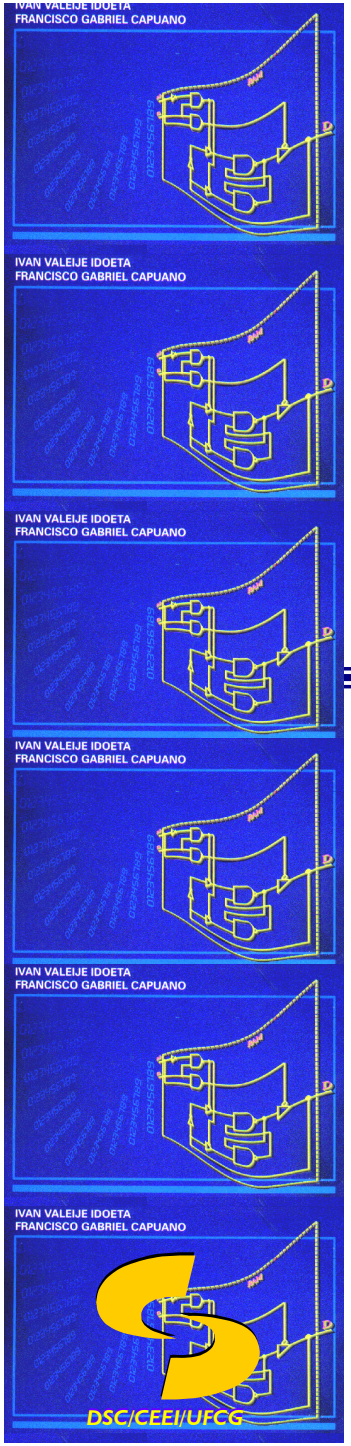


Universidade Federal de Campina Grande

Unidade Acadêmica de Sistemas e Computação

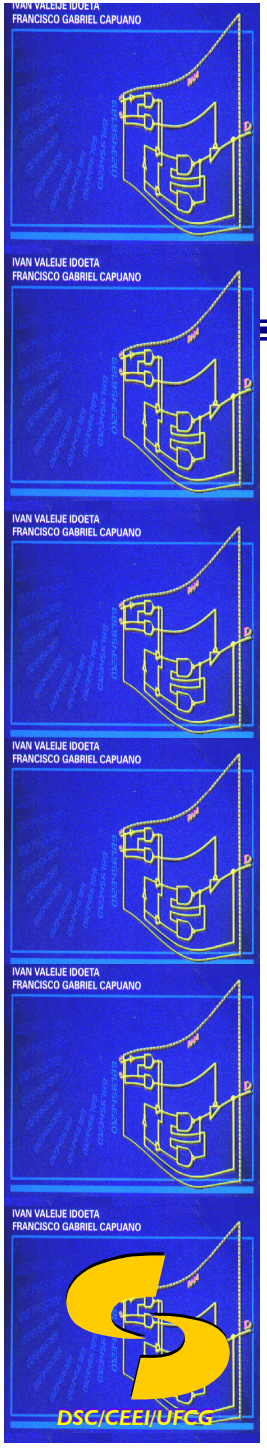
Introdução à Computação

Conceitos Básicos de Eletrônica Digital (Parte I)

Prof.ª Joseana Macêdo Fechine Régis de Araújo

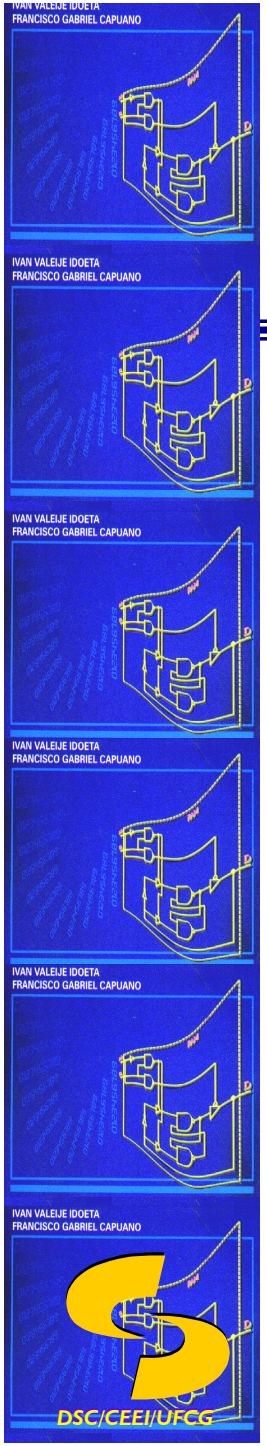
joseana@computacao.ufcg.edu.br

Carga Horária: 60 horas



Conceitos Básicos de Eletrônica Digital

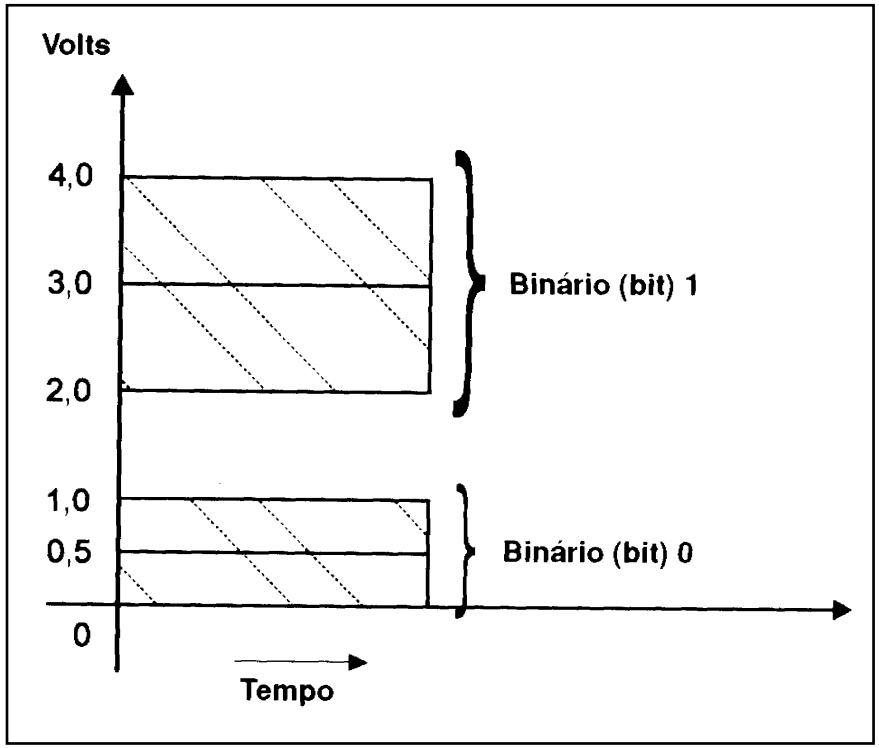
- Tópicos
  - Álgebra de Boole
  - Portas Lógicas
  - Circuitos combinacionais
  - Exemplos de aplicação

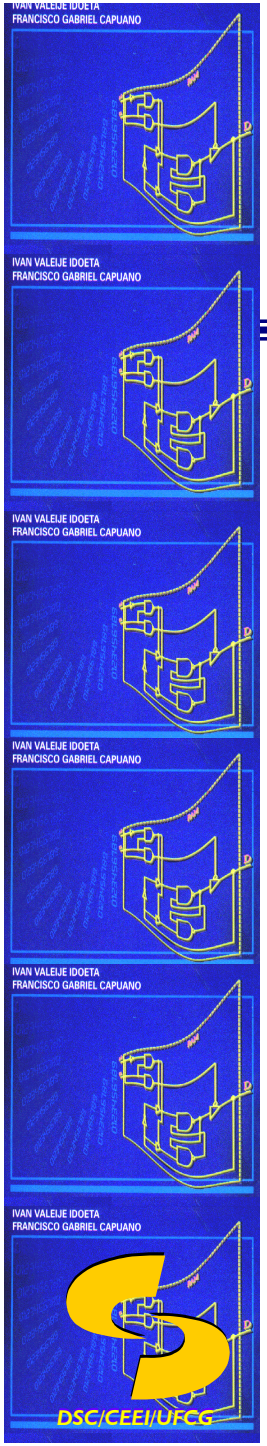


Conceitos Básicos de Eletrônica Digital

- **Computador digital** - máquina projetada para armazenar e manipular informações representadas apenas por algarismos (ou dígitos) e que só podem assumir dois valores distintos, 0 e 1.

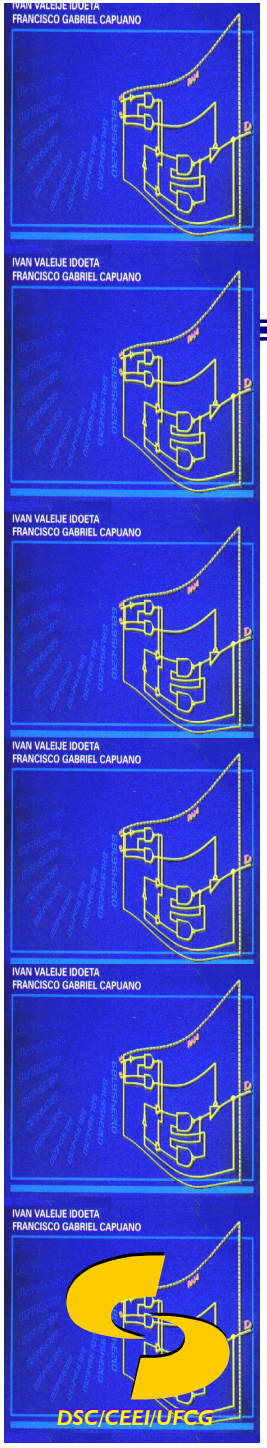
Informação binária (0 ou 1) - representada em um sistema digital por quantidades físicas (sinais elétricos).





Conceitos Básicos de Eletrônica Digital

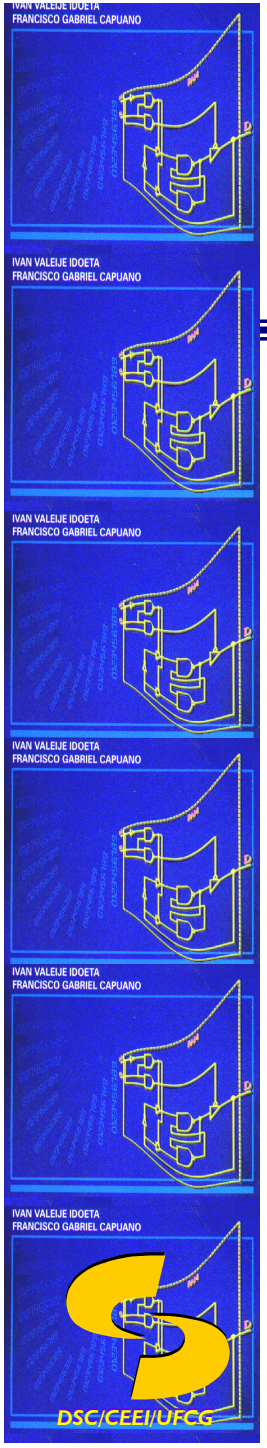
- Operações de um computador digital - combinações de simples operações aritméticas e lógicas básicas: somar bits, complementar bits (para fazer subtrações), comparar bits, mover bits.
- As operações são fisicamente realizadas por circuitos eletrônicos, chamados **circuitos digitais**.
- Componentes básicos dos circuitos digitais - **"portas" (*gates*) lógicas**, por permitirem ou não a passagem dos sinais.
- **Circuitos lógicos** - circuitos que contêm as portas lógicas.



Conceitos Básicos de Eletrônica Digital

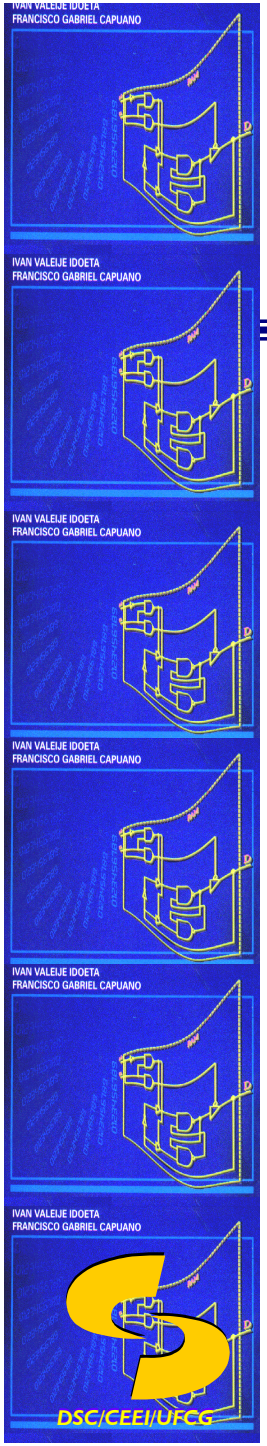
- **Computadores digitais** (binários) - construídos com circuitos eletrônicos digitais - as portas lógicas (circuitos lógicos).

- Um computador digital é construído, então, contendo circuitos lógicos (ou portas), convenientemente distribuídos e organizados, de modo que:
  - alguns servirão para armazenamento de valores,
  - outros permitirão e controlarão o fluxo de sinais entre componentes e
  - outros serão utilizados para realizar operações matemáticas.



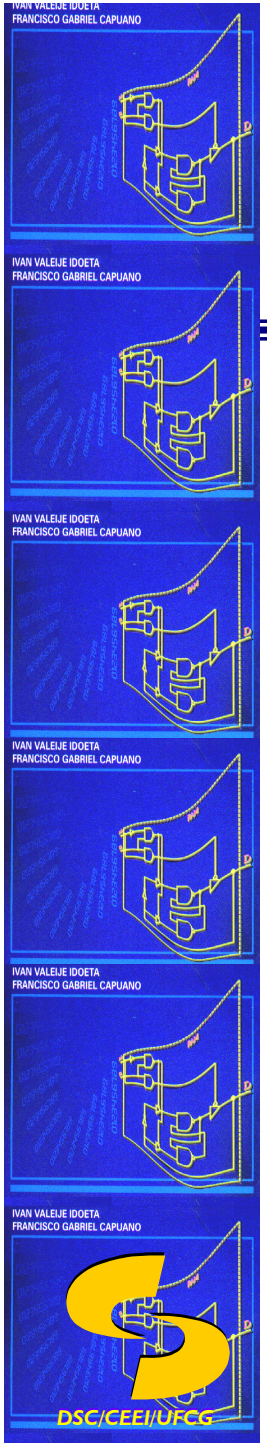
Conceitos Básicos de Eletrônica Digital

- O projeto de circuitos digitais e a análise de seu comportamento podem ser realizados através do emprego de conceitos e regras estabelecidas pela **álgebra de chaveamentos**, um ramo da álgebra moderna ou **álgebra de Boole**, conceituada pelo matemático inglês George Boole (1815 - 1864).



Conceitos Básicos de Eletrônica Digital

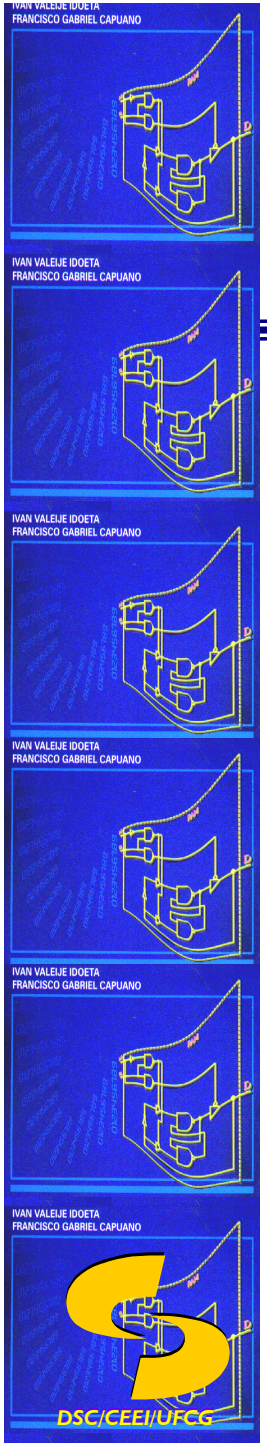
É importante entender o significado dos seguintes conceitos: **Lógica e Álgebra de Boole** e como estes conceitos podem ser empregados para a implementação das **portas lógicas** e, conseqüentemente, dos **circuitos lógicos** (digitais) **e computadores digitais.**



Conceitos Básicos de Eletrônica Digital

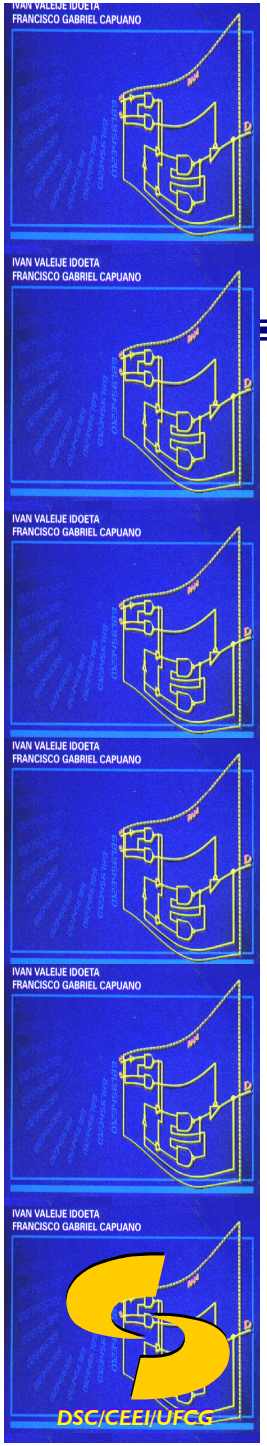
- A **lógica** é a base da eletrônica digital e da informática. Esta surgiu na Grécia antiga com a contribuição de três filósofos: **Sócrates**, **Platão** e **Aristóteles**.
  - Sócrates não deixou seus ensinamentos por escrito.
  - Platão (seguidor de Sócrates) escreveu vários de seus diálogos e desenvolveu sua filosofia abrangendo a ética, a política e o conhecimento, tendo como princípio o método da investigação.
  - Aristóteles, baseado nos diálogos escritos por Platão, observou que a linguagem deve ter uma estrutura lógica, para que leve, necessariamente, a uma verdade.
  - Pelo método de investigação de Sócrates, se duas verdades são alcançadas individualmente, ao juntá-las tem-se uma única verdade.

Sócrates, considerado um dos homens mais sábios da humanidade, notabilizou-se por afirmar que era sábio justamente por "**saber que nada sabia**".



Conceitos Básicos de Eletrônica Digital

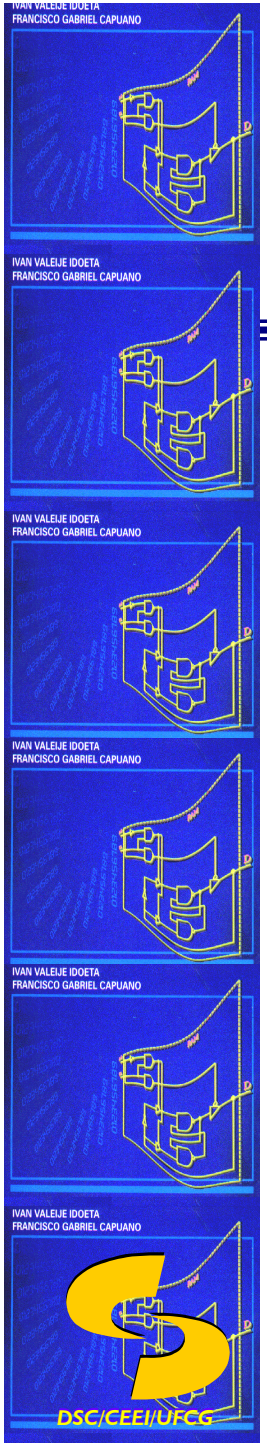
- No século XIX, a teoria de Aristóteles foi sintetizada em forma de álgebra, ganhando o nome de Álgebra Booleana.
- A Álgebra de Boole permite que **uma afirmação** (lógica) possa ser expressa matematicamente.
- Boole construiu sua lógica a partir de símbolos, representando as expressões por letras e ligando-as através de conectivos - **símbolos algébricos**.
- Boole, através de seu livro "*An investigation of the laws of thought*" (Uma investigação das leis do pensamento) apresentou a **lógica binária.**



Conceitos Básicos de Eletrônica Digital

- A **lógica** teve como objetivo modelar o raciocínio humano.
- Partindo de frases declarativas (*proposições*), que podem ser ***verdadeiras*** ou ***falsas***, estuda-se o processo de construção e a veracidade de outras proposições usando conectivos.
- Na lógica proposicional associa-se a cada proposição um valor lógico: ou verdade (1) ou falso (0).

Da **Lógica** nasceu a **Lógica Matemática** e, dentro desta, várias filosofias da lógica que interpretam os cálculos simbólicos e sua sistematização axiomática.

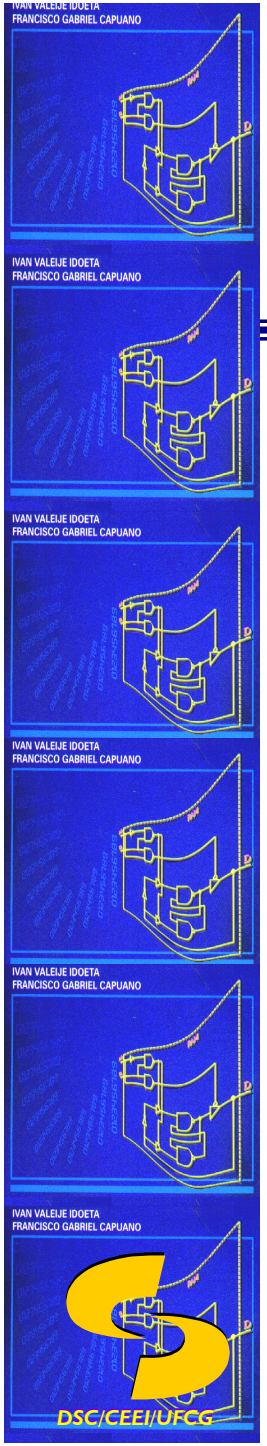


Álgebra de Boole

- **Operação lógica –** realizada sobre um ou mais valores lógicos para produzir um certo resultado (também um valor lógico).

- Assim como na álgebra comum, é necessário definir símbolos matemáticos e gráficos para representar as operações lógicas (e os operadores lógicos).

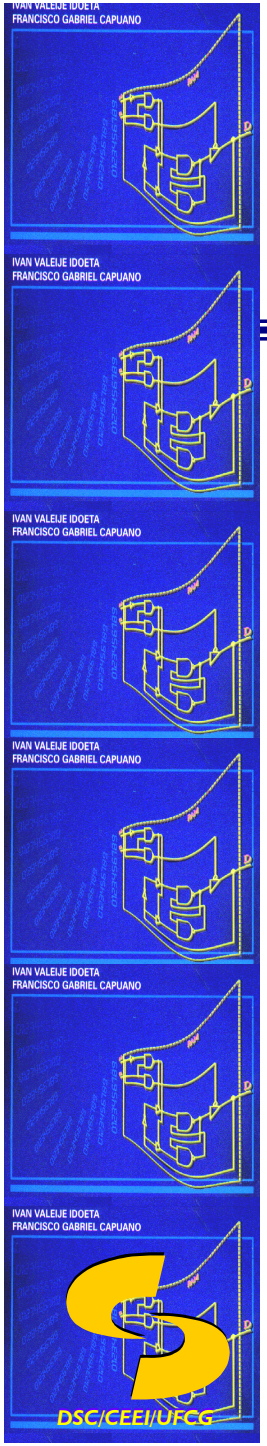
- Resultados possíveis de uma operação lógica:
  - 0 (FALSO, **F**= bit 0) - nível baixo
  - 1 (VERDADEIRO, **V** = bit 1) - nível alto (Lógica Positiva)



Álgebra de Boole

OPERADORES LÓGICOS BÁSICOS

- Os conectivos ou OPERADORES LÓGICOS ou FUNÇÕES LÓGICAS são:
  - **E (ou AND)** - uma sentença é verdadeira **SE - e somente se** - todos os termos forem verdadeiros.
  - **OU (ou OR)** - uma sentença resulta verdadeira se **QUALQUER UM** dos termos for verdadeiro.
  - **NÃO (ou NOT)** - este operador **INVERTE** um termo.

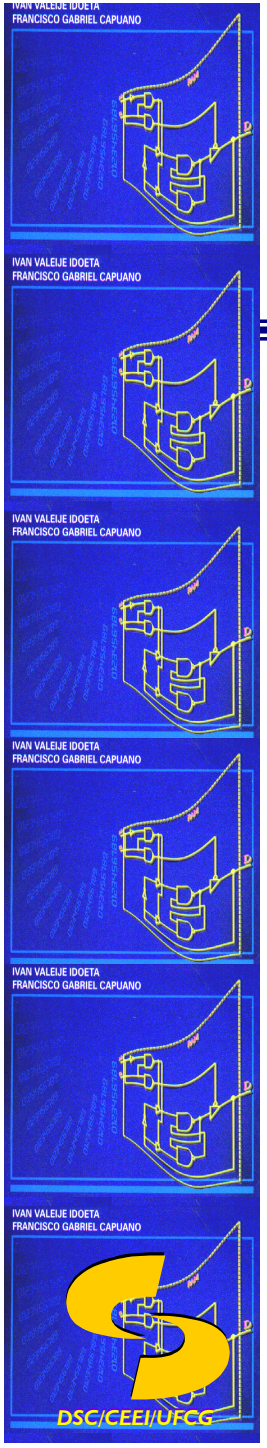


Álgebra de Boole

OPERADORES LÓGICOS BÁSICOS

- Os operadores lógicos são representados por:
  - **E** → • (um ponto, como se fosse uma multiplicação)
  - **OU** → **+** (o sinal de soma)
  - **NOT** → $\overline{}$ (ou ’) (uma barra horizontal sobre o termo a ser invertido ou negado).

Simbologia definida pela ANSI



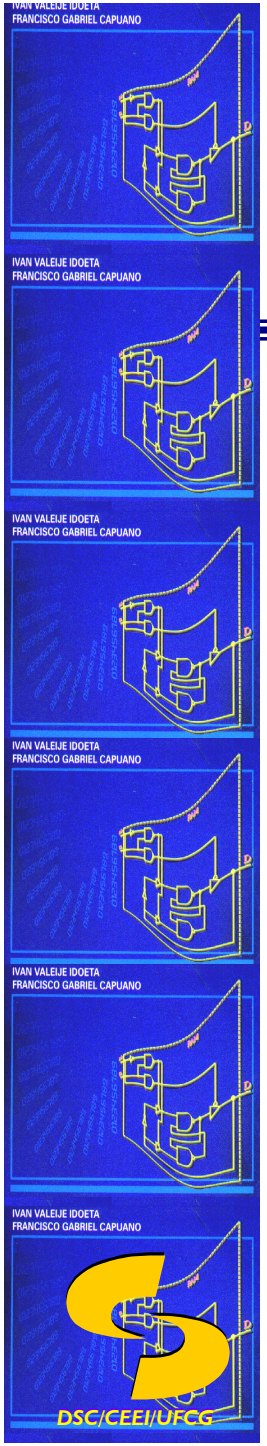
Álgebra de Boole

FUNÇÕES LÓGICAS

- Operadores que possuem como entrada pelo menos uma variável lógica e uma saída.
- Dada uma variável lógica (**A**), é possível construir uma função desta variável, **f**(**A**).
- Operações da álgebra booleana aplicadas a uma ou mais variáveis lógicas.

- **Funções básicas**: E, OU e INVERSORA (AND, OR e NOT ou INVERTER)

- **Derivadas**: (NAND, NOR, XOR e XNOR).

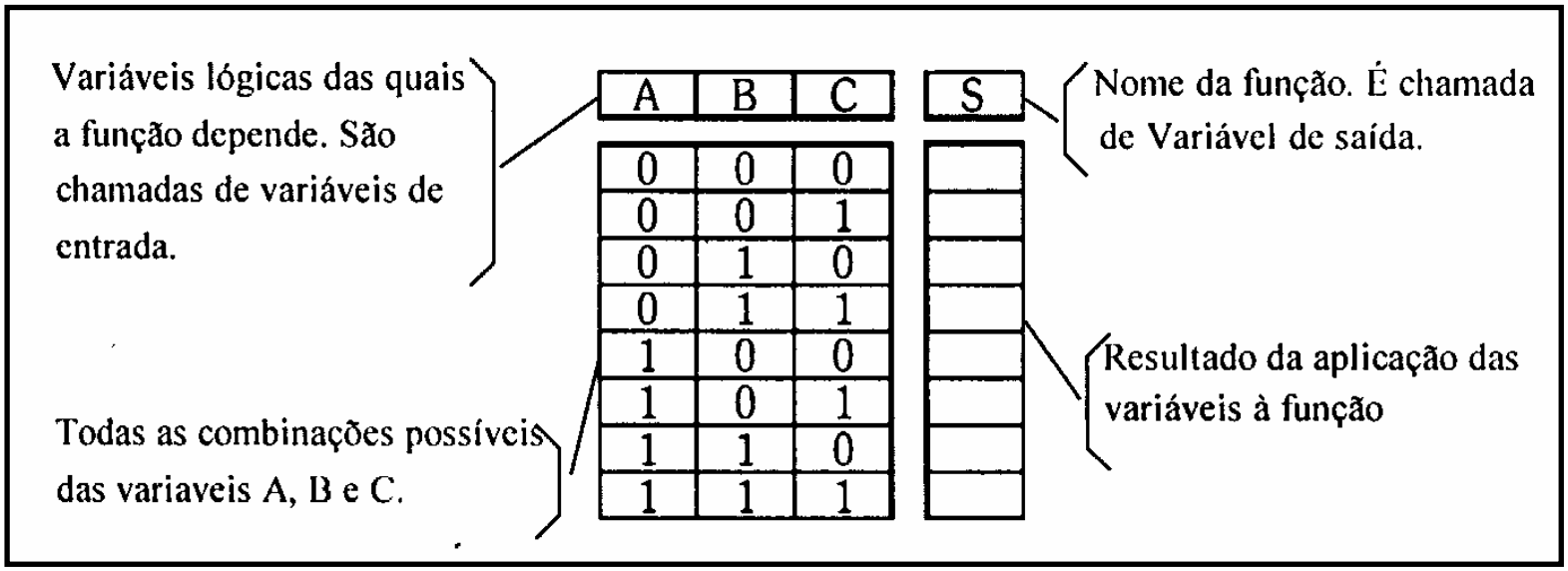


Álgebra de Boole

- A partir das combinações dos valores de entrada, determina-se todos os valores possíveis de resultado de uma dada operação lógica.
- Essas possibilidades podem ser representadas de forma tabular, e o conjunto se chama **TABELA VERDADE**.
- **TABELA VERDADE** - tabela que representa todas as possíveis combinações das variáveis de entrada de uma função, e os seus respectivos valores de saída.

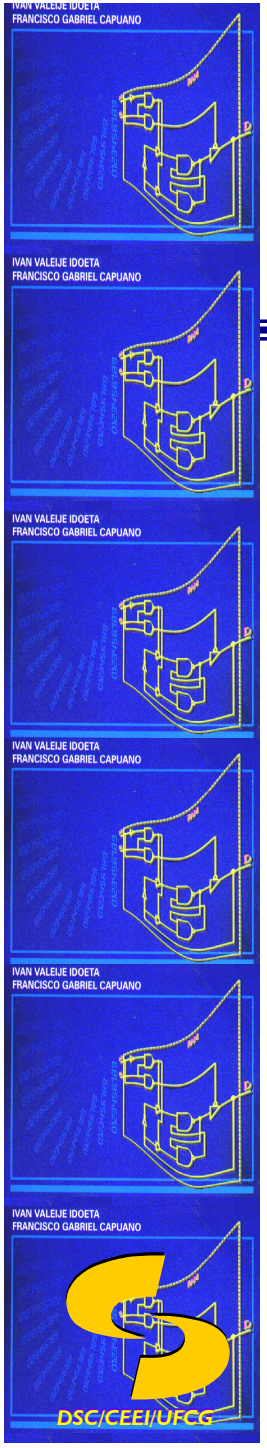
Álgebra de Boole

Tabela-verdade



A	B	C	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Cada operação lógica possui sua própria tabela verdade, estabelecida de acordo com a regra que define a respectiva operação lógica.



Álgebra de Boole

FUNÇÃO AND (E)

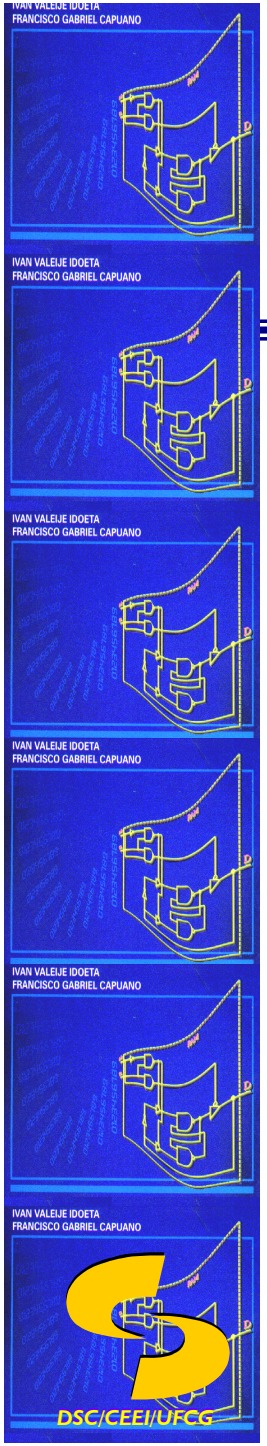
$$S = A \cdot B$$

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

FUNÇÃO OR (OU)

$$S = A + B$$

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

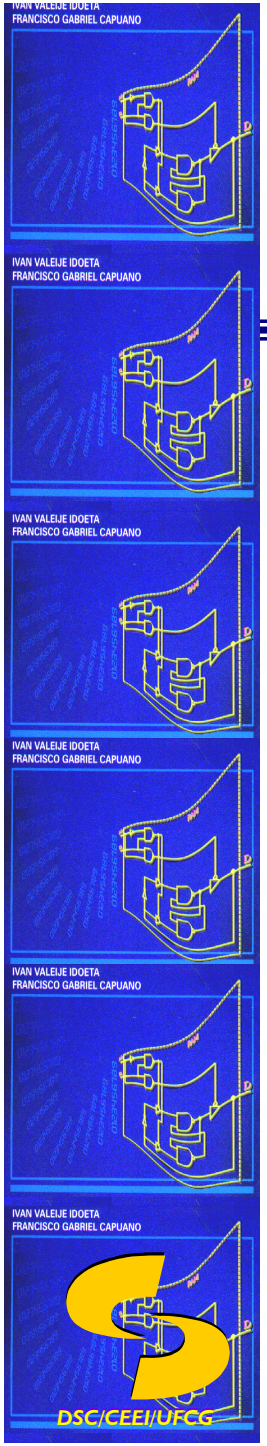


Álgebra de Boole

FUNÇÃO NOT
(INVERTER OU NÃO)

$$S = \overline{A}$$

A	S
0	1
1	0



Álgebra de Boole

FUNÇÃO NAND (NÃO E)

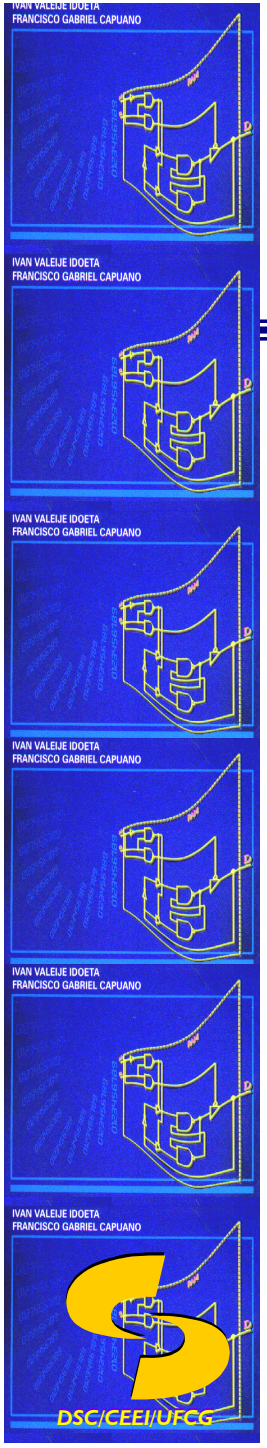
$$S = \overline{A \cdot B}$$

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

FUNÇÃO NOR (NÃO OU)

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

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



Álgebra de Boole

FUNÇÃO XOR
(OU EXCLUSIVO)

$S = A \oplus B$

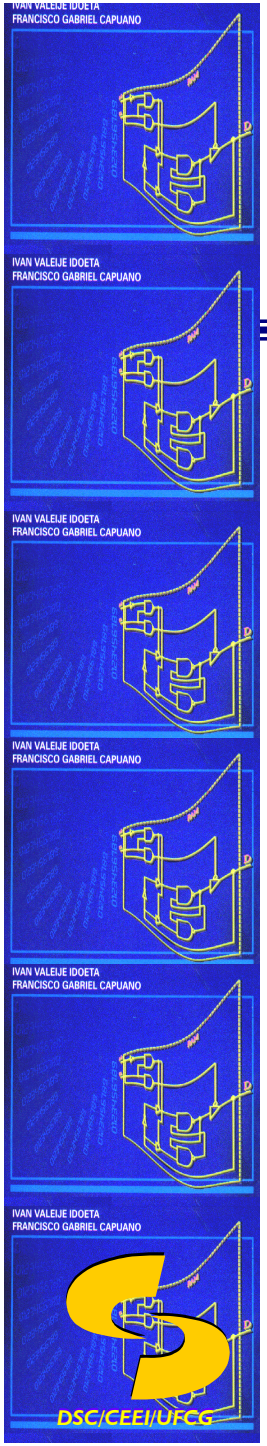
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

FUNÇÃO XNOR
(OU COINCIDÊNCIA)

$S = A \otimes B$

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

XOR - a saída será verdade se **exclusivamente uma ou outra entrada** for verdade. (XNOR - inverso da XOR). Isto só se aplica se houver apenas 2 entradas.

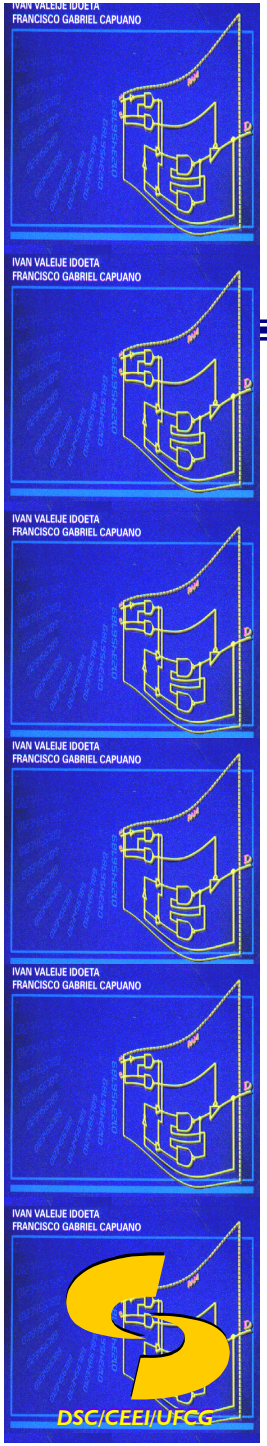


Álgebra de Boole e Computadores Digitais

- O projeto de elementos digitais está relacionado com a **conversão de idéias em *hardware* real**, e os elementos encontrados na **álgebra booleana** permitem que uma idéia, uma afirmação, possa ser expressa matematicamente.

- A **álgebra booleana** permite também que a expressão resultante da formulação matemática da idéia possa ser simplificada e, finalmente, **convertida no mundo real do *hardware* de portas lógicas e outros elementos digitais**.

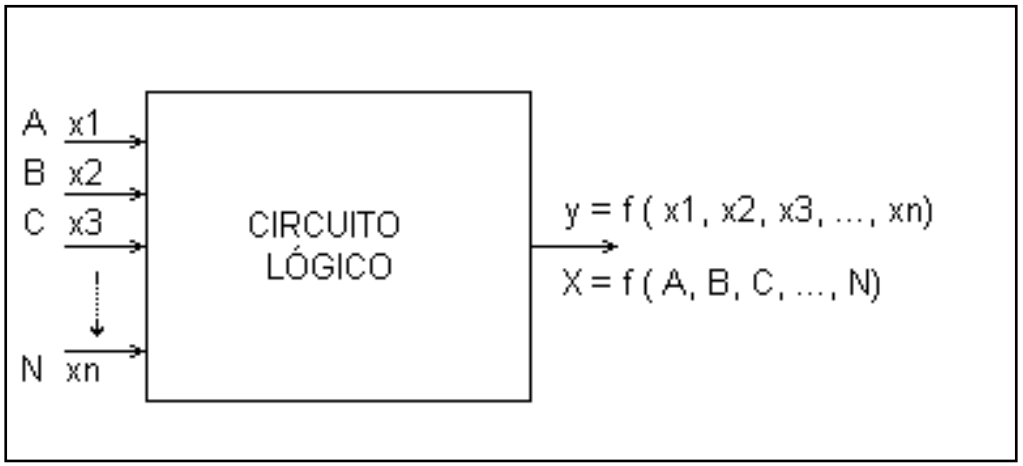
O que são exatamente?

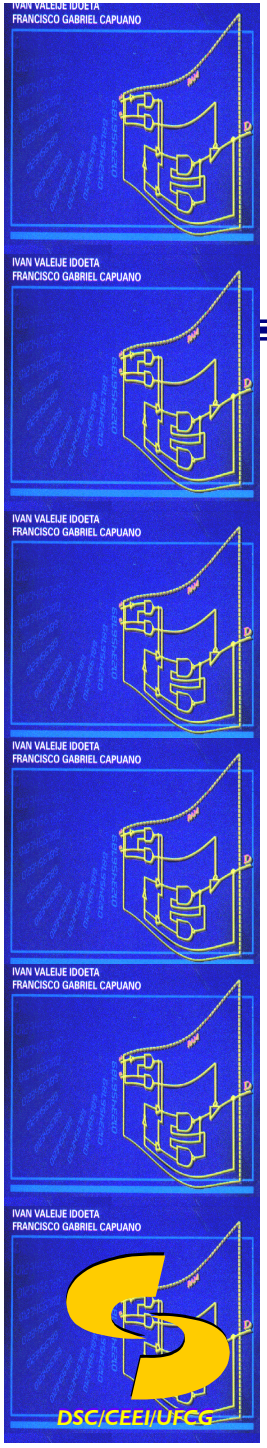


Álgebra de Boole

- **Portas lógicas**: dispositivos dos circuitos digitais - implementam funções lógicas.

- São dispositivos ou circuitos lógicos que operam um ou mais sinais lógicos de entrada para produzir uma (e somente uma) saída, a qual é dependente da função implementada no circuito.

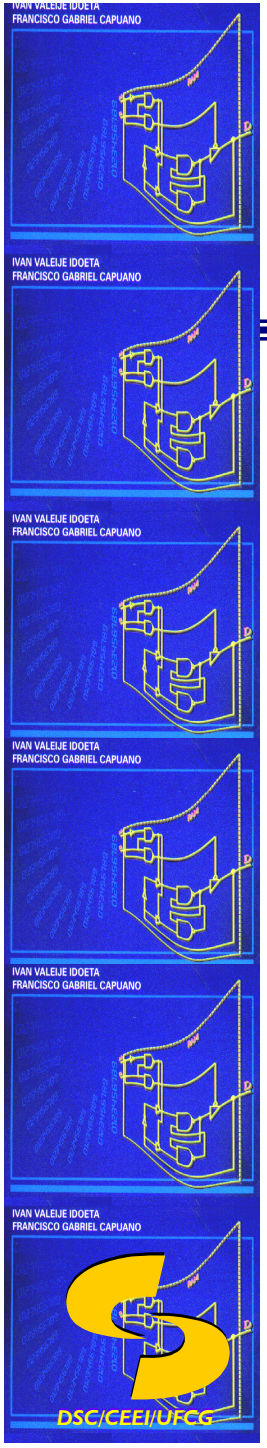




Álgebra de Boole e Computadores Digitais

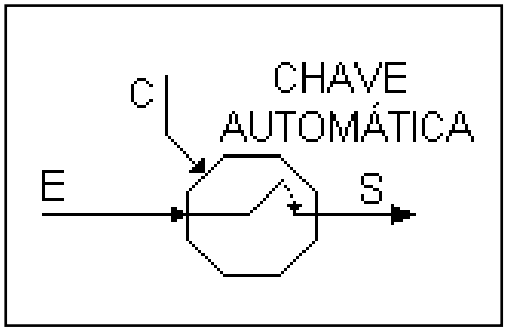
Como os conceitos da álgebra de chaveamentos (ramo da álgebra do Boole) são aplicados ao projeto dos computadores digitais?

- **Primeiros computadores fabricados** (Ex.: ENIAC) - trabalhavam em DECIMAL - grande complexidade ao projeto e construção dos computadores, tendo por conseqüência um custo muito elevado.
- **Aplicação da álgebra de Boole** – uso de apenas dois algarismos 0 (F) e 1(V) → simplificação do projeto e construção dos computadores.



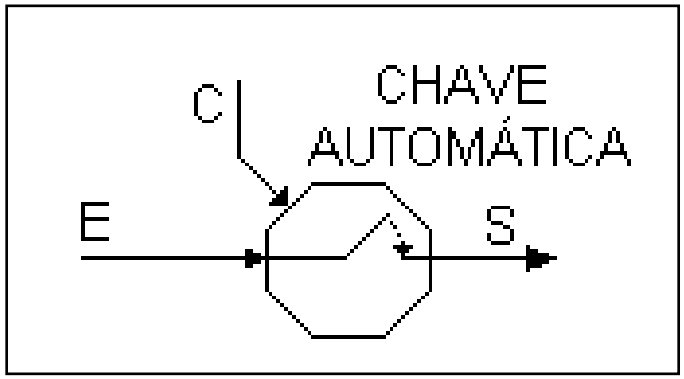
Álgebra de Boole e Computadores Digitais

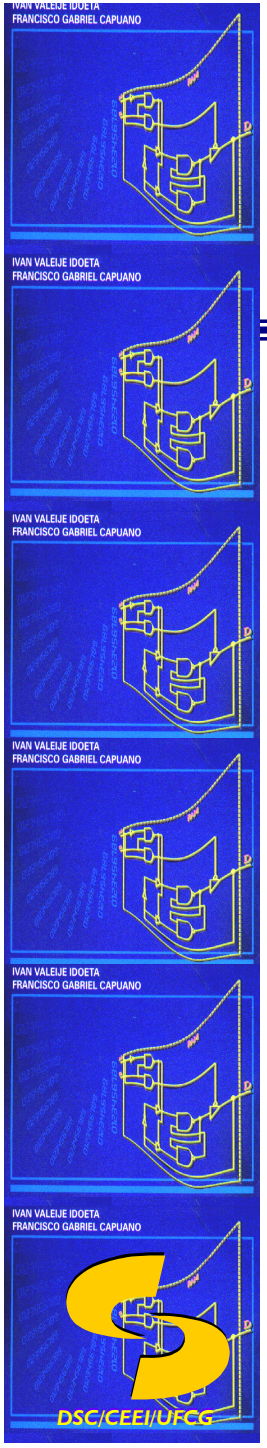
- A chave de tudo é um circuito eletrônico chamado **CHAVE AUTOMÁTICA**.
- **Como funciona uma chave automática?**
- Considerar um circuito chaveador com as seguintes entradas:
  - uma fonte de alimentação (fornece energia para o circuito)
  - um fio de controle (comanda a operação do circuito)
  - um fio de saída (conduz o resultado)



Álgebra de Boole e Computadores Digitais

- Sinal **C = 0** (ou F) ⇒ **S = 0** (ou Falso). A chave permanece aberta.
- Sinal **C = 1** (ou V) ⇒ **S = 1** (ou V). A chave muda de posição.
- A posição da chave se manterá enquanto não ocorrer um novo sinal na entrada.





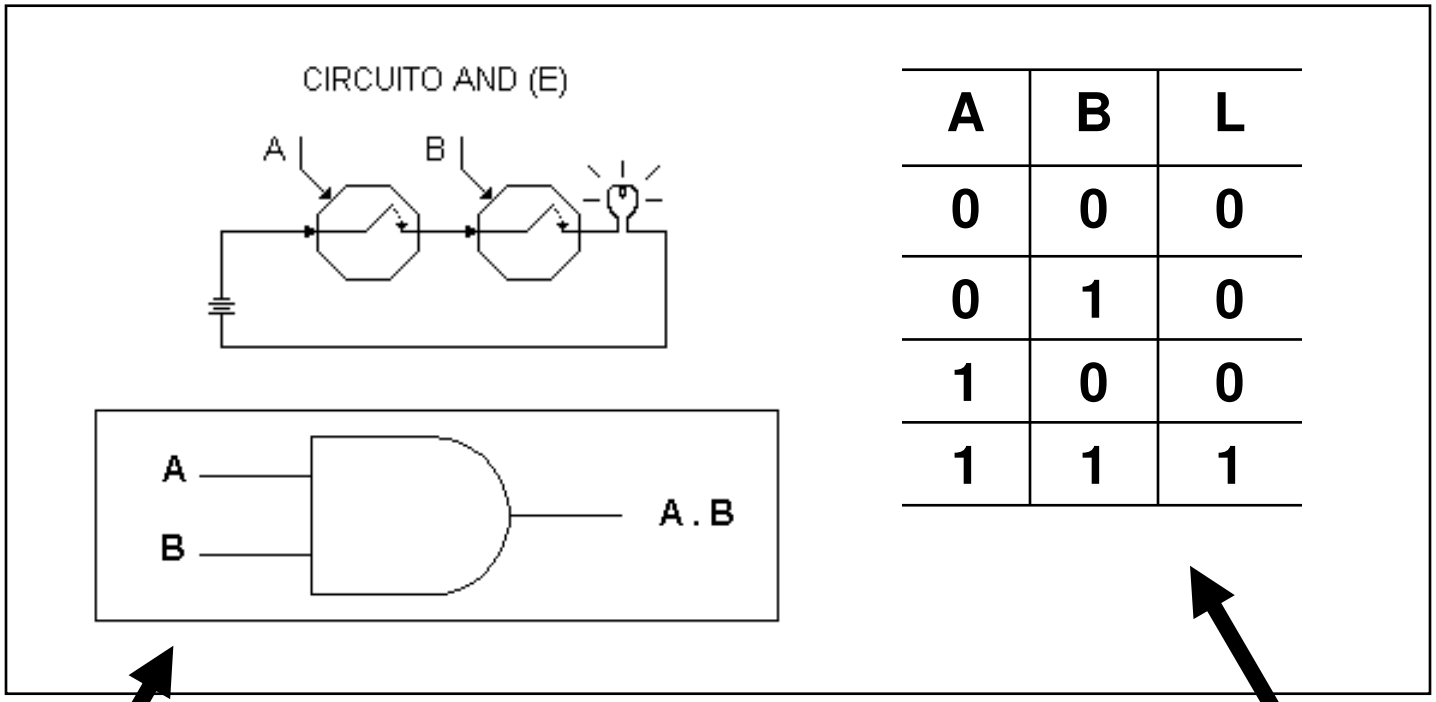
Álgebra de Boole e Computadores Digitais

- A chave automática foi inicialmente implementada com relés eletromecânicos e depois com válvulas eletrônicas.
- A partir da metade da década de 50, passaram a ser utilizados dispositivos em estado sólido - os TRANSISTORES, inventados em Stanford em 1947.

- Modernos Circuitos Integrados e microprocessadores são implementados com milhões de transistores "impressos" em minúsculas pastilhas.

Álgebra de Boole e Computadores Digitais

- **Ligação em SÉRIE** de duas chaves automáticas (com uma lâmpada ligada ao circuito).

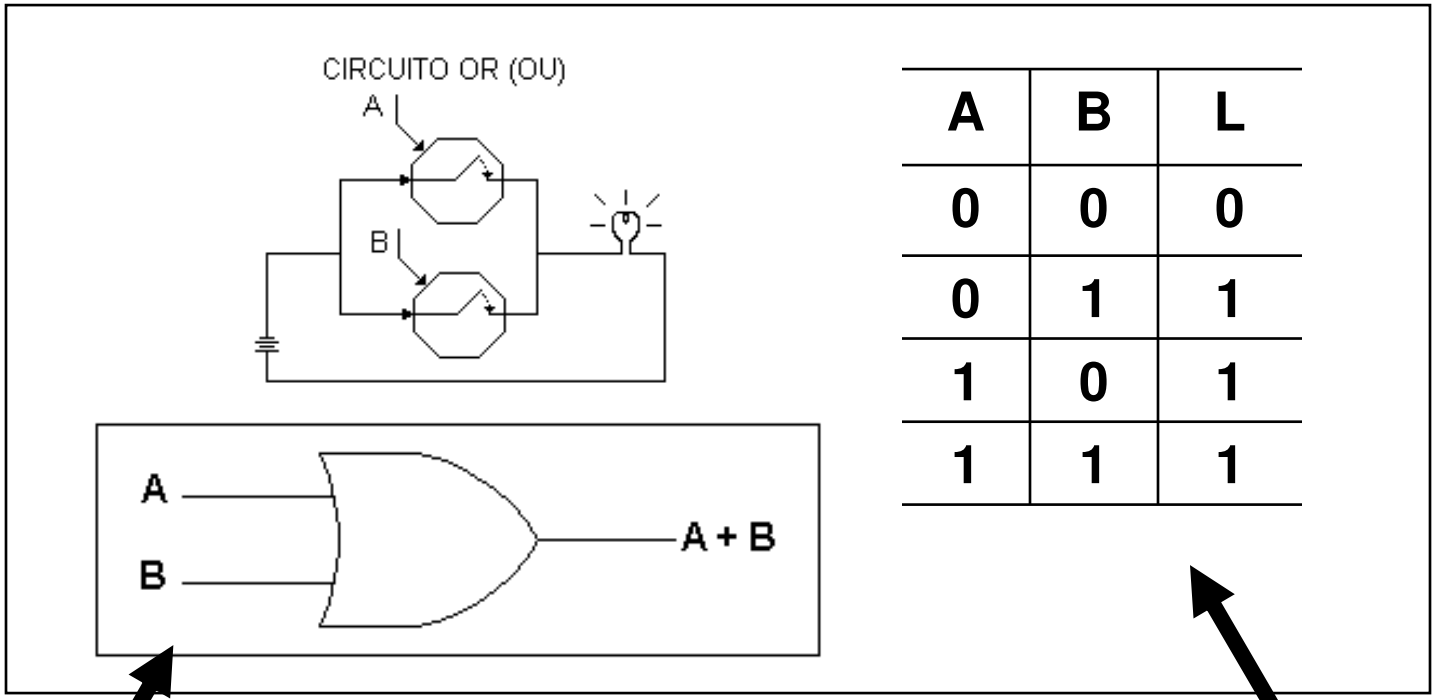


A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

PORTA E (AND GATE) - circuito que implementa a **função E**.

Álgebra de Boole e Computadores Digitais

- **Ligação em PARALELO** de duas chaves automáticas (com uma lâmpada ligada ao circuito).



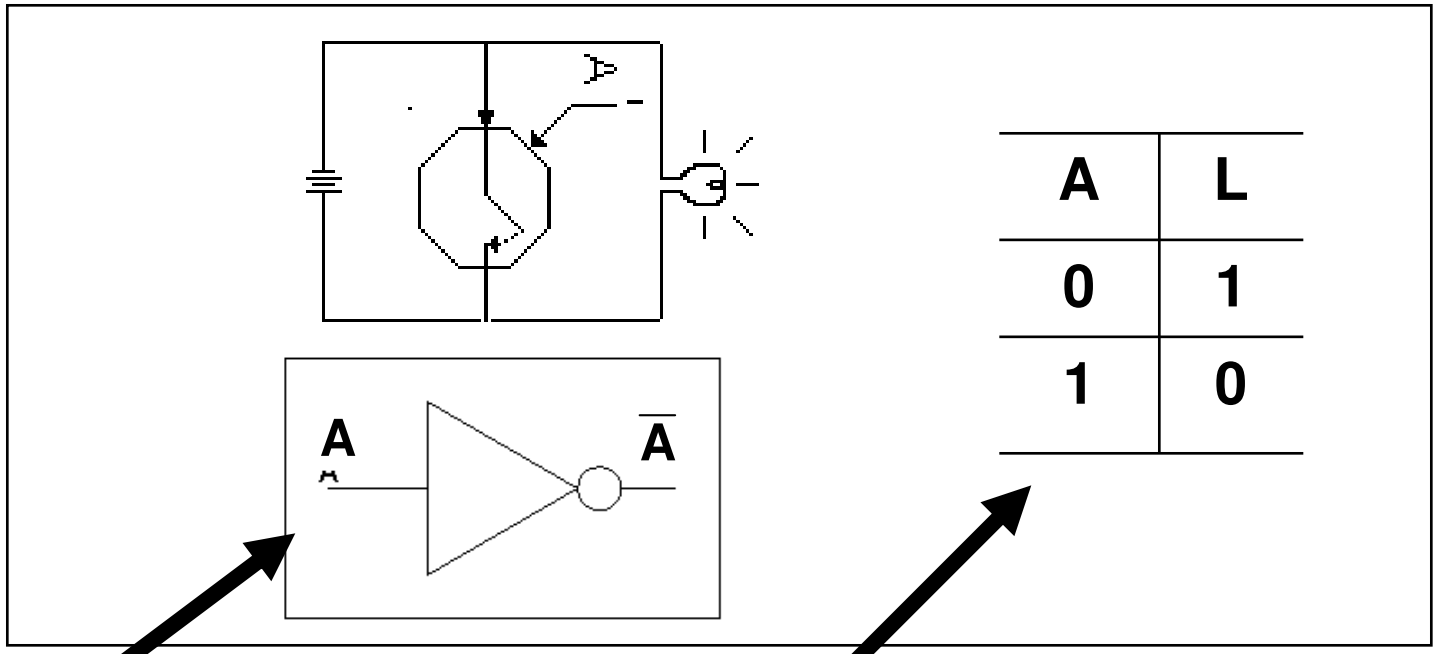
A	B	L
0	0	0
0	1	1
1	0	1
1	1	1

PORTA OU (OR GATE) - circuito que implementa a **função OR**.

IVAN VALEIJE IDOETA
FRANCISCO GABRIEL CAPUANO

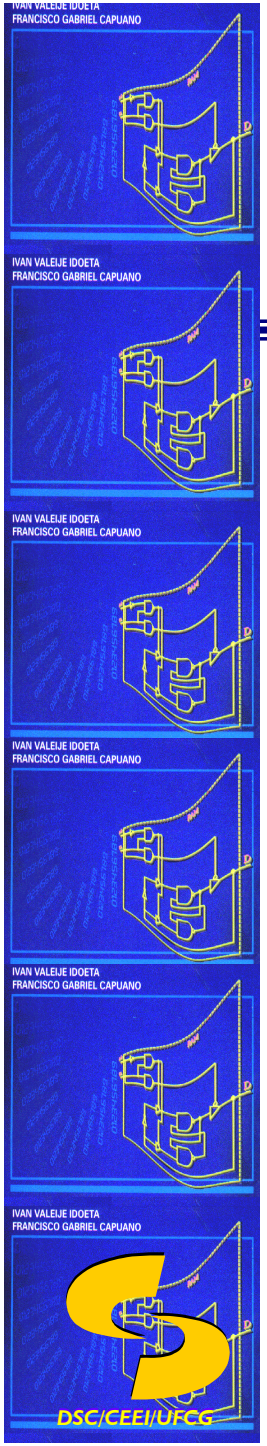
Álgebra de Boole e Computadores Digitais

- **Ligação** de uma chave automática (com uma lâmpada ligada ao circuito).



A	L
0	1
1	0

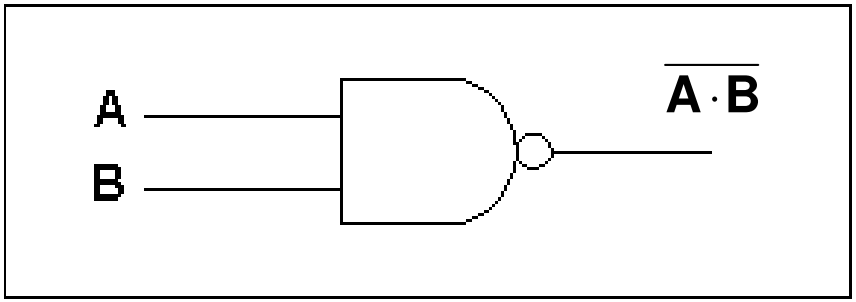
PORTA NÃO (NOT GATE ou INVERTER GATE) - circuito que implementa a **função NÃO**.



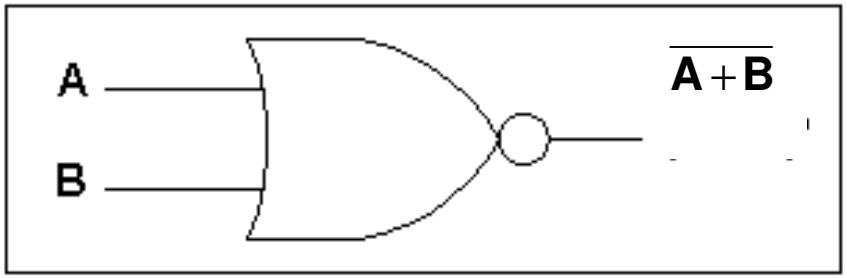
Álgebra de Boole e Computadores Digitais

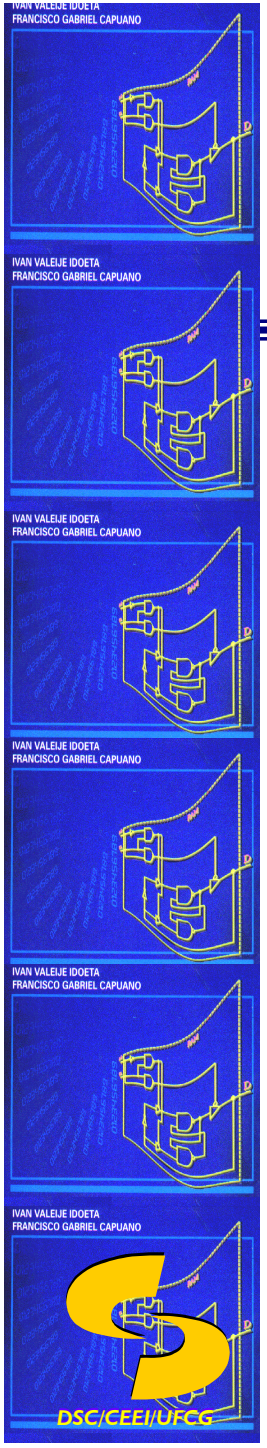
- **Demais portas lógicas:**

PORTA NAND (NAND GATE) - circuito que implementa a **função NAND**.



PORTA NOR (NOR GATE) - circuito que implementa a **função NOR**.

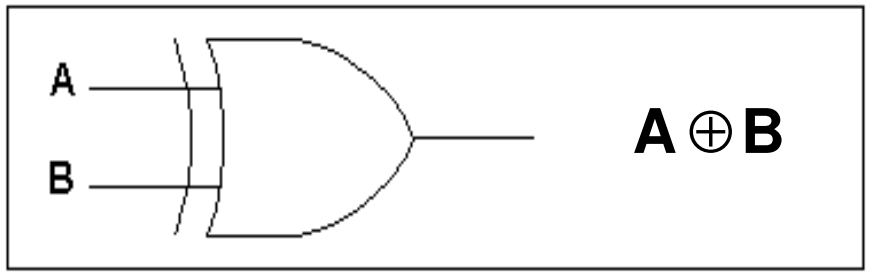




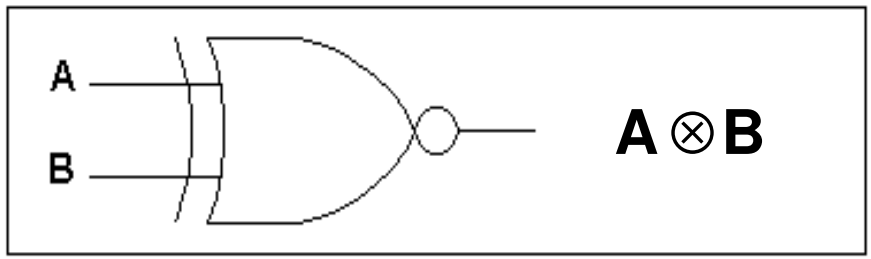
Álgebra de Boole e Computadores Digitais

- **Demais portas lógicas:**

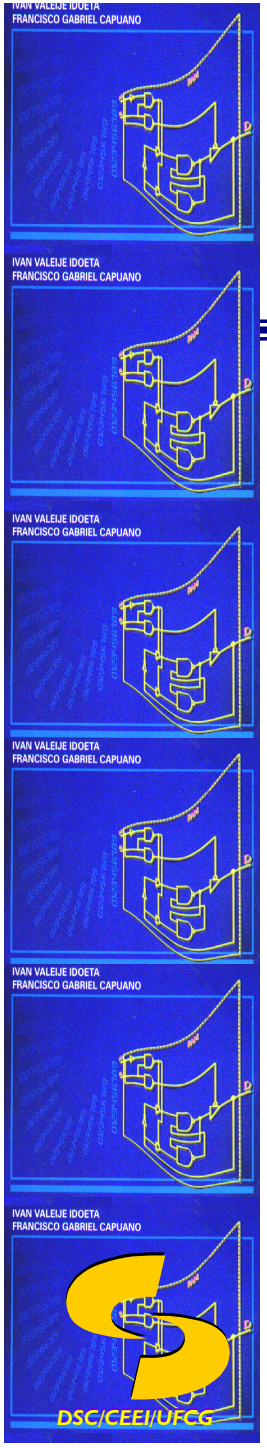
PORTA XOR (XOR GATE) - circuito que implementa a **função XOR**.



PORTA XNOR (XNOR GATE) - circuito que implementa a **função XNOR**.



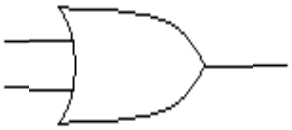
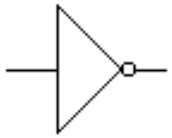
Número par de entradas - portas XOR e XNOR possuem **saídas complementares** entre si. **Número ímpar de entradas**, as saídas das portas XOR e XNOR são **iguais entre si**.



Álgebra de Boole e Computadores Digitais

Quadro Resumo

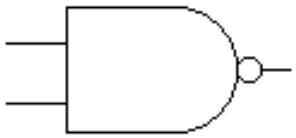
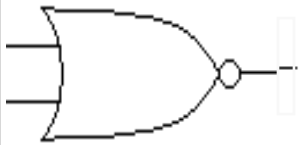
BLOCOS LÓGICOS BÁSICOS

Porta	Símbolo usual	Tabela Verdade	Função Lógica	Expressão
E AND		A B S 0 0 0 0 1 0 1 0 0 1 1 1	**Função E**: assume 1 quando todas as variáveis forem 1 e 0 nos outros casos.	$S = A \cdot B$
OU OR		A B S 0 0 0 0 1 1 1 0 1 1 1 1	**Função OU**: assume 0 quando todas as variáveis forem 0 e 1 nos outros casos.	$S = A + B$
NÃO NOT		A S 0 1 1 0	**Função NOT**: inverte a variável aplicada a sua entrada.	$S = \overline{A}$

IVAN VALEIJE IDOETA
FRANCISCO GABRIEL CAPUANO

Álgebra de Boole e Computadores Digitais

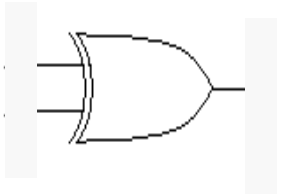
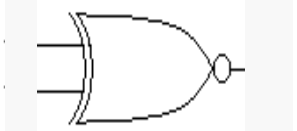
Quadro Resumo

BLOCOS LÓGICOS BÁSICOS				
Porta	**Símbolo usual**	**Tabela Verdade**	**Função Lógica**	**Expressão**
NÃO E NAND		A B S 0 0 1 0 1 1 1 0 1 1 1 0	**Função NÃO E**: Inverso da função E	$S = \overline{A \cdot B}$
NÃO OU NOR		A B S 0 0 1 0 1 0 1 0 0 1 1 0	**Função NÃO OU**: Inverso da função OU	$S = \overline{A + B}$

Álgebra de Boole e Computadores Digitais

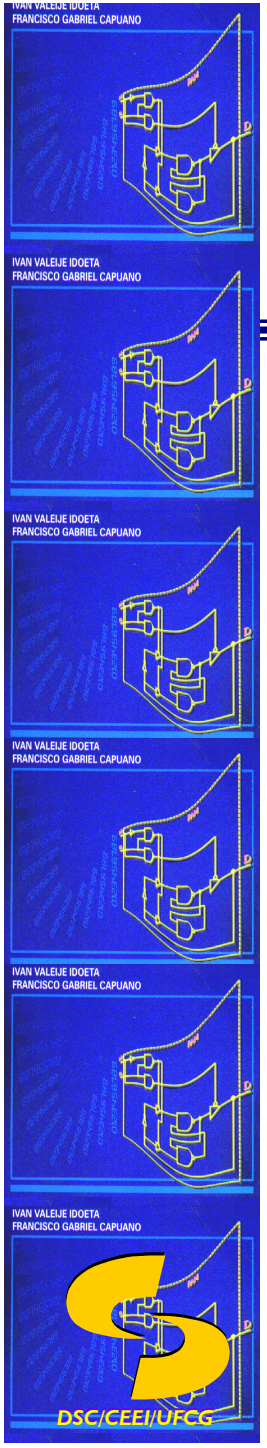
Quadro Resumo

BLOCOS LÓGICOS BÁSICOS

Porta	Símbolo usual	Tabela Verdade	Função Lógica	Expressão
OU EXCLUSIVO XOR		A B S 0 0 0 0 1 1 1 0 1 1 1 0	**Função XOR**: Assume 1 quando as duas variáveis assumirem valores diferentes entre si.	$S = A \oplus B$
COINCIDÊNCIA XNOR		A B S 0 0 1 0 1 0 1 0 0 1 1 1	**Função XNOR**: Assume 1 quando houver coincidência entre os valores das variáveis.	$S = A \otimes B$

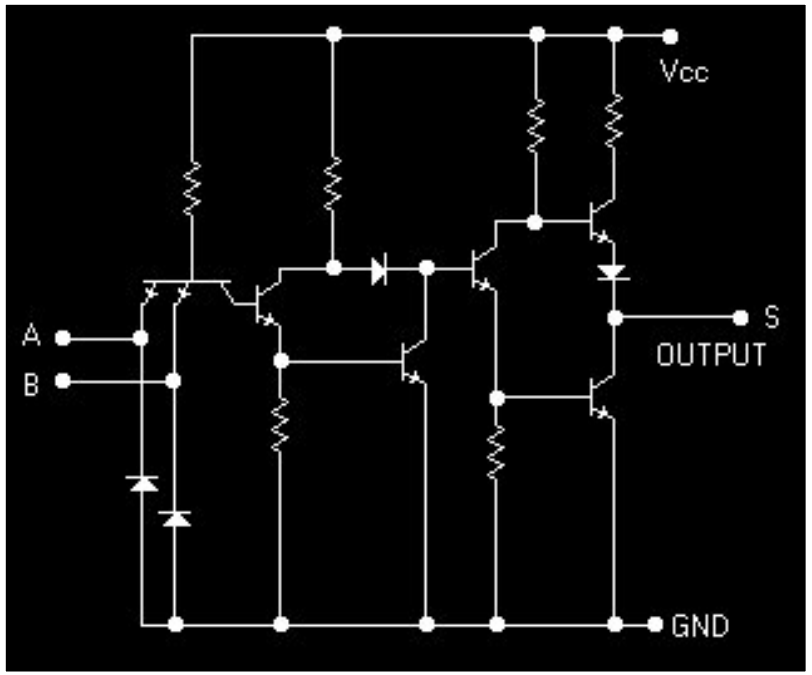
As Portas lógicas **XOR e XNOR** são na verdade circuitos obtidos de portas lógicas básicas.

$$S = A \oplus B = \overline{A} \cdot B + A \cdot \overline{B} \qquad S = A \otimes B = A \cdot B + \overline{A} \cdot \overline{B}$$

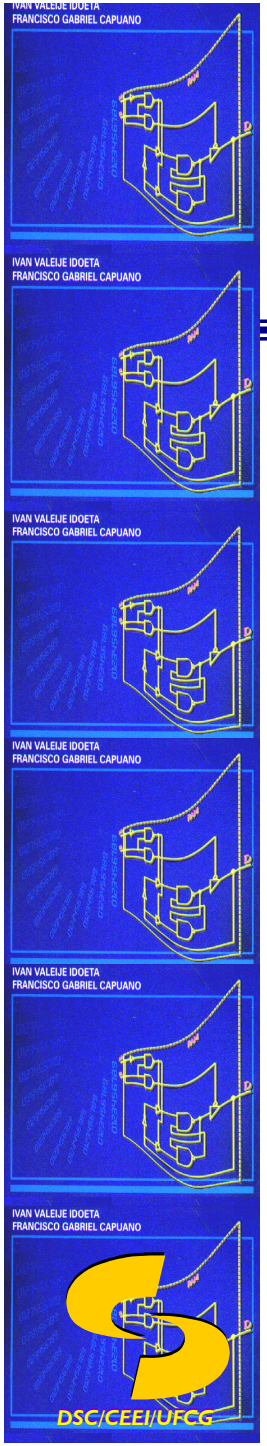


Álgebra de Boole e Computadores Digitais

- **Obs.**: O circuito elétrico da porta lógica que implementa a função **AND** é :



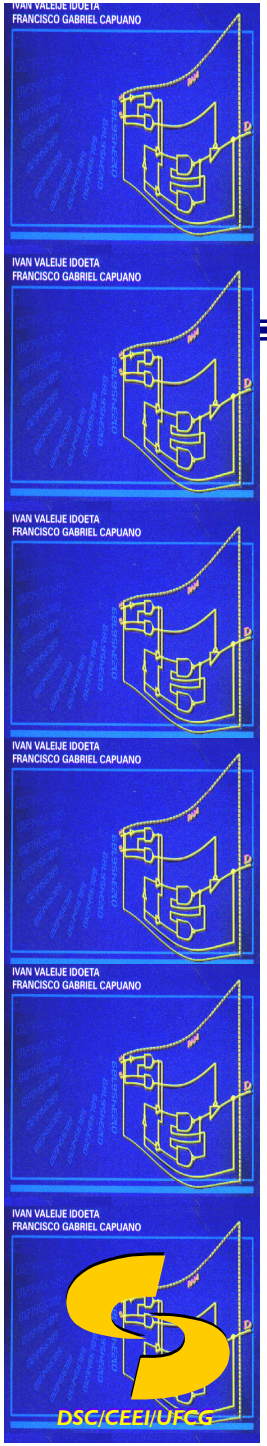
- Torna-se difícil desenhar o esquema elétrico de um projeto composto por várias portas lógicas representadas desta forma.
- **Solução**: uso de uma **SIMBOLOGIA**.



Álgebra de Boole e Computadores Digitais

- **Operações lógicas** podem ser realizadas para
  - satisfazer um determinado **requisito de *hardware*** (visto adiante) ou
  - para atender a uma **especificação de um programador** em um programa.

- Para tanto, a maioria dos processadores possui uma instrução de máquina correspondente a uma função lógica em seu conjunto de instruções, bem como muitas linguagens de programação de alto nível implementam essa função.



Exemplos – Operações Lógicas

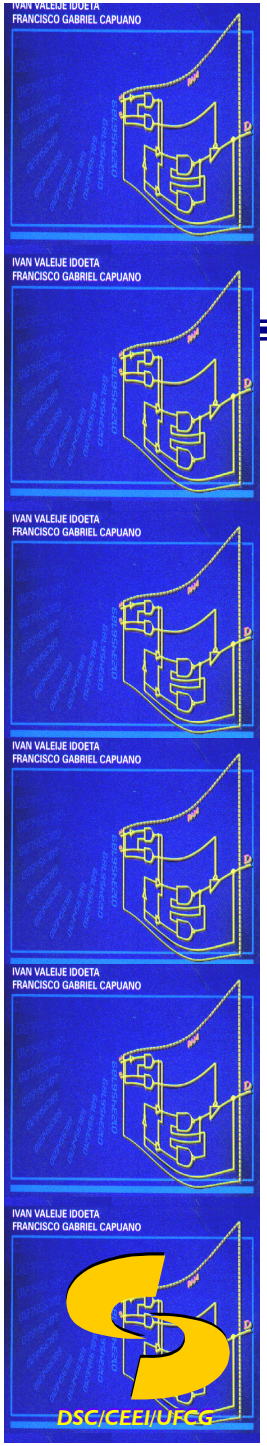
- Lógica de um determinado programa.

Exemplo 1:

- Ler X, Y e Z
- T=X+Y
- R=Z+X
- SE (T>6 E(AND) R <10)
  - ENTÃO IMPRIMIR T
  - ENTÃO IMPRIMIR R

Exemplo 2:

- Ler X, Y e Z
- T=X+Y
- R=Z+X
- SE (T>6 OU(OR) R <10)
  - ENTÃO IMPRIMIR T
  - ENTÃO IMPRIMIR R

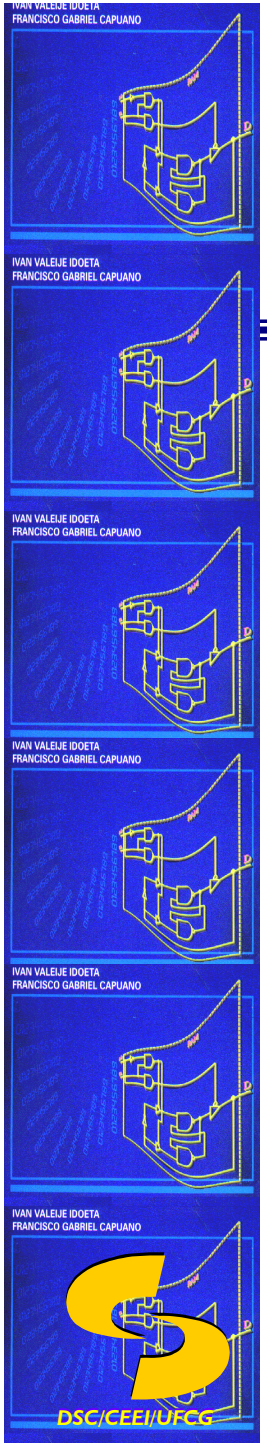


Exemplos – Operações Lógicas

- Operações lógicas também podem ser realizadas com valores constituídos de vários algarismos (A Unidade Lógica e Aritmética (ULA) realiza tal tipo de operação) – operação "bit a bit".

Exemplos:

$$\mathbf{A = 0110 \ \ e \ \ B = 1010}$$
$$\Rightarrow \mathbf{A \cdot B = 0010}$$
$$\Rightarrow \mathbf{A + B = 1110}$$
$$\Rightarrow \mathbf{\overline{A} = 1001}$$
$$\Rightarrow \mathbf{\overline{(A \cdot B)} = 1101}$$
$$\Rightarrow \mathbf{\overline{(A + B)} = 0001}$$
$$\Rightarrow \mathbf{A \oplus B = 1100}$$
$$\Rightarrow \mathbf{\overline{(A \otimes B)} = 0011}$$



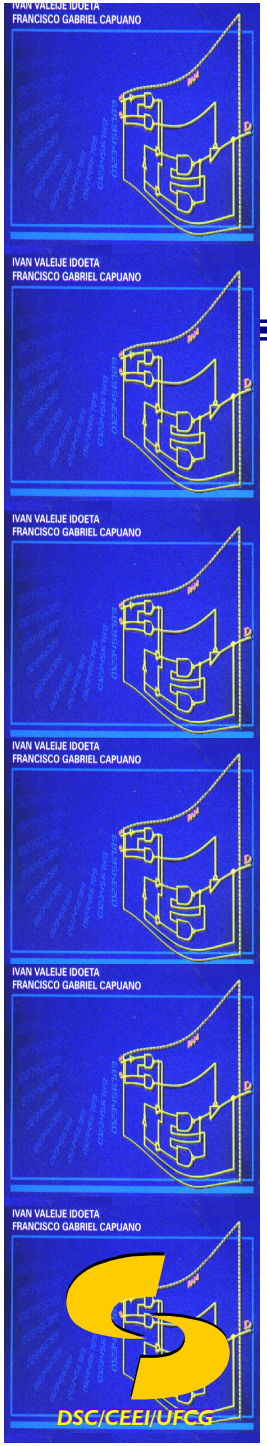
Exemplos - Circuitos utilizando portas lógicas

- Detector de incêndio com vários sensores (entradas) e uma campainha para alarme (saída). Se QUALQUER UM dos sensores for acionado (significando que um dos sensores detectou sinal de incêndio), a campainha é ACIONADA.

Sensor 1	Sensor 2	Alarme
0	0	0
0	1	1
1	0	1
1	1	1

Solução:

Porta OR



Exemplos - Circuitos utilizando portas lógicas

- Uma campainha que toca (saída) se o motorista der a partida no motor do carro (entrada) sem estar com o cinto de segurança afivelado (entrada).

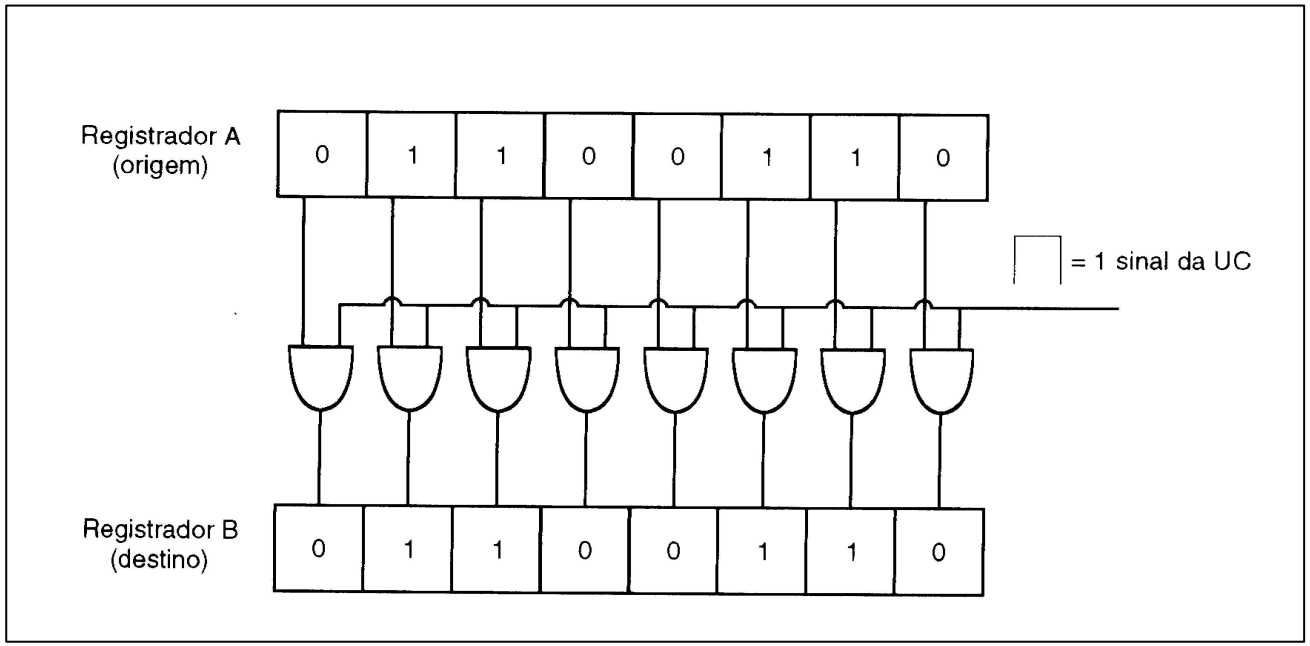
ignição	Cinto desafi-velado	campainha
0	0	0
0	1	0
1	0	0
1	1	1

Solução:

Porta AND

Exemplos – Circuitos utilizando portas lógicas

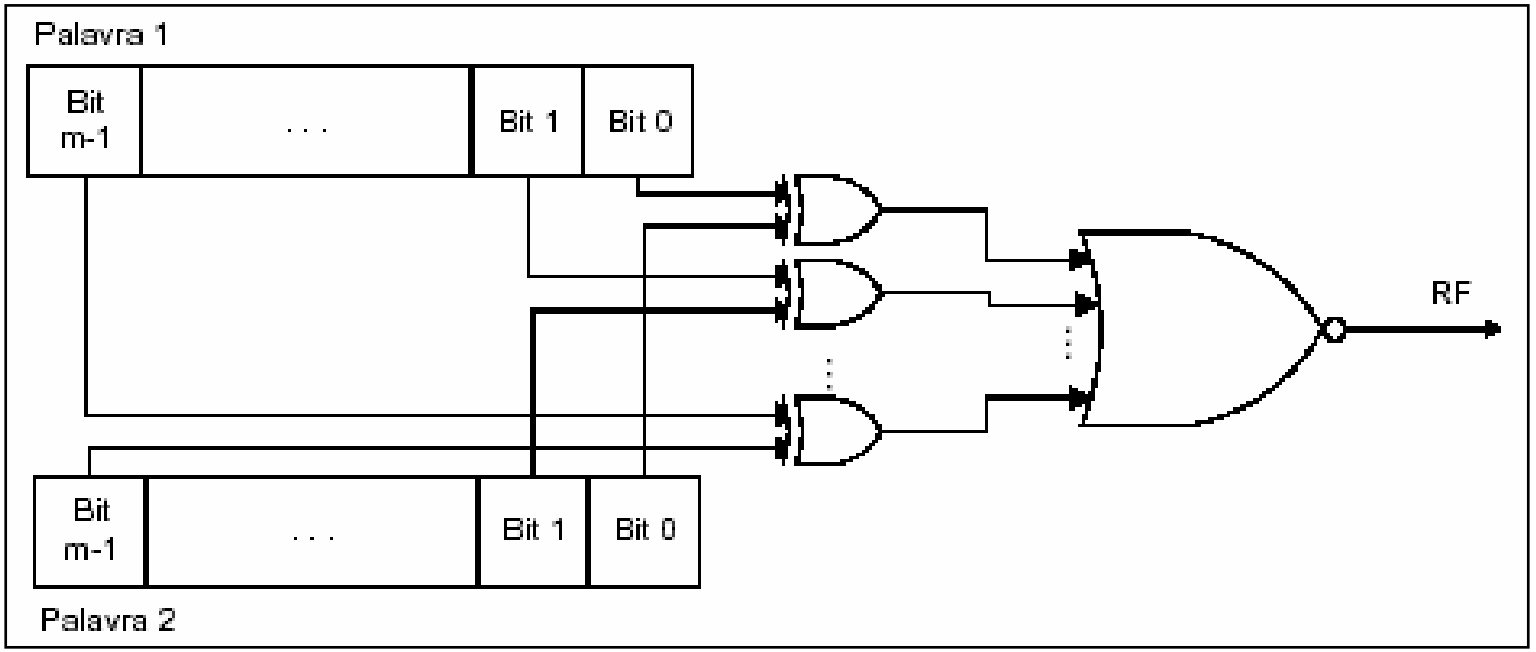
- Circuito de ativação de uma linha de dados para movimentar bits de um registrador (ou células) para outro (uso de um bit como sinal de controle da Unidade de Controle (UC)).



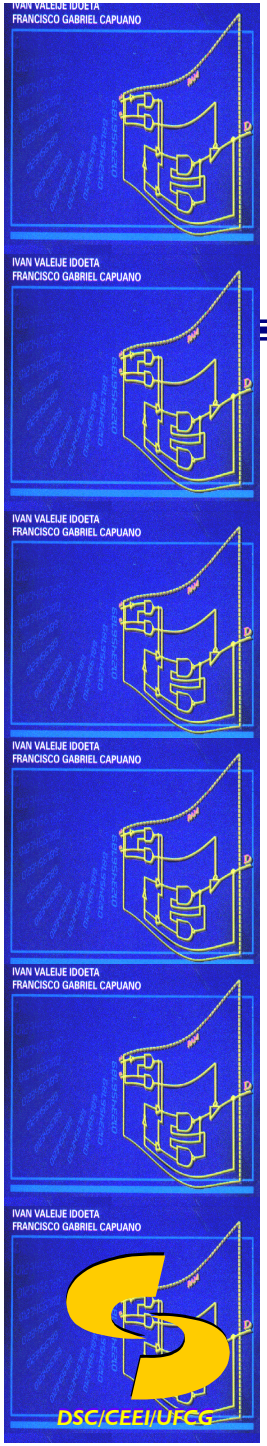
Solução: Porta AND

Exemplos - Circuitos utilizando portas lógicas

- Circuito para testar a igualdade entre valores, por exemplo, para testar de modo rápido se duas palavras são iguais.



Solução: **Porta XOR** e **porta NOR**

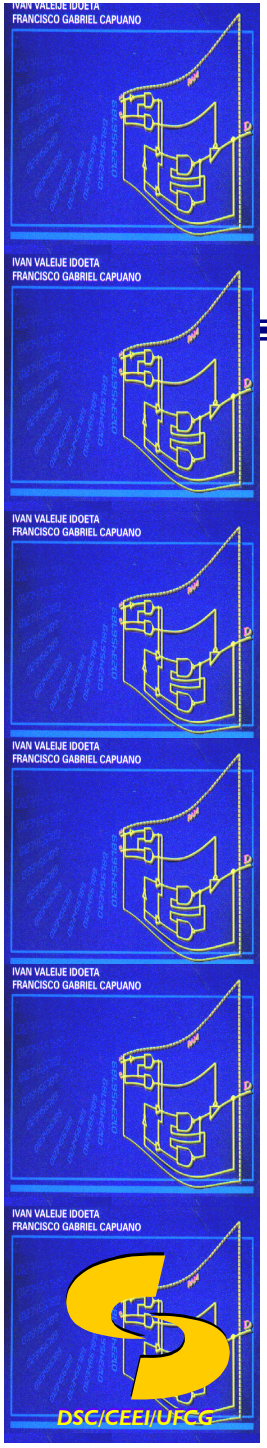


Exemplos - Circuitos utilizando portas lógicas

- Circuito para implementação de operação aritmética em ponto fixo, quando se usa aritmética de complemento (complemento de 1 ou complemento de 2).

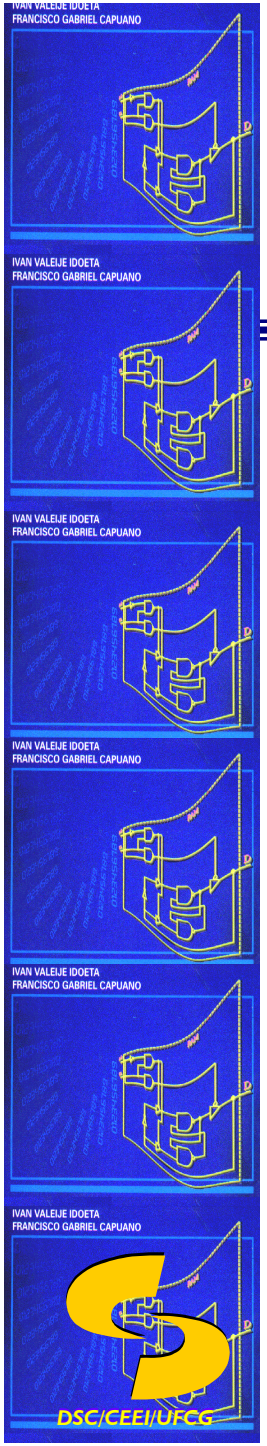
  Solução: Porta NOT

- É muito comum encontrar a porta **NAND** (**ou NOR**) em circuitos lógicos complexos, visto que é possível simplificar a fabricação de circuitos lógicos e reduzir a quantidade de componentes eletrônicos usando-se apenas circuitos **NAND** (**NOR**).



Portas Lógicas - Fabricação

- As portas lógicas são fornecidas em dispositivos denominados **circuitos integrados** ou CI´s.
- Um CI (ou **chip**) é um cristal semicondutor, habitualmente de silício.
- Cada CI´s comporta um certo número de portas lógicas, sendo este número limitado pelas características físicas do componente como, por exemplo, o número de terminais.
- A partir do surgimento do transistor procurou-se padronizar os sinais elétricos correspondentes aos níveis lógicos $\Rightarrow$ surgimento de **famílias de componentes digitais**.



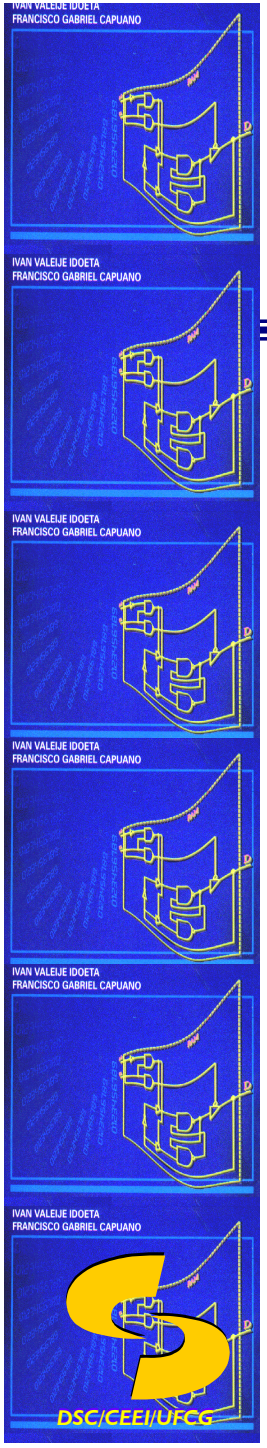
Portas Lógicas - Fabricação

- As famílias lógicas diferem basicamente pelo componente principal utilizado por cada uma em seus circuitos (Ex.: TTL e CMOS).

- Família **TTL** (***T**ransistor-**T**ransistor **L**ogic*) - transistores bipolares.

  TTL $\rightarrow$ 0 V a 0.8 V = nível lógico 0,
  2 V a 5 V = nível lógico 1.

- Famílias: tecnologia **MOS** (***M**etal **O**xide **S**emicondutor*) - transistores unipolares MOSFET (transistor por efeito de campo - técnica MOS).
  - CMOS (MOS complementar) opera com fontes de 3 a 18 V (baixíssimo consumo).



Portas Lógicas - Fabricação

- TTL e CMOS constituíram as alternativas principais durante muitos anos, mas a evolução tecnológica permitiu o aparecimento regular de outras soluções de compromisso entre a velocidade e o consumo.
  - Em **TTL** tem-se as variantes **L** (*low power*), **S** (*Schottky*), **LS** (*low-power Schottky*), etc.
  - Em **CMOS**, temos as variantes **HC** (*high-speed CMOS*) e **HCT** (compatível pino a pino com os TTL).

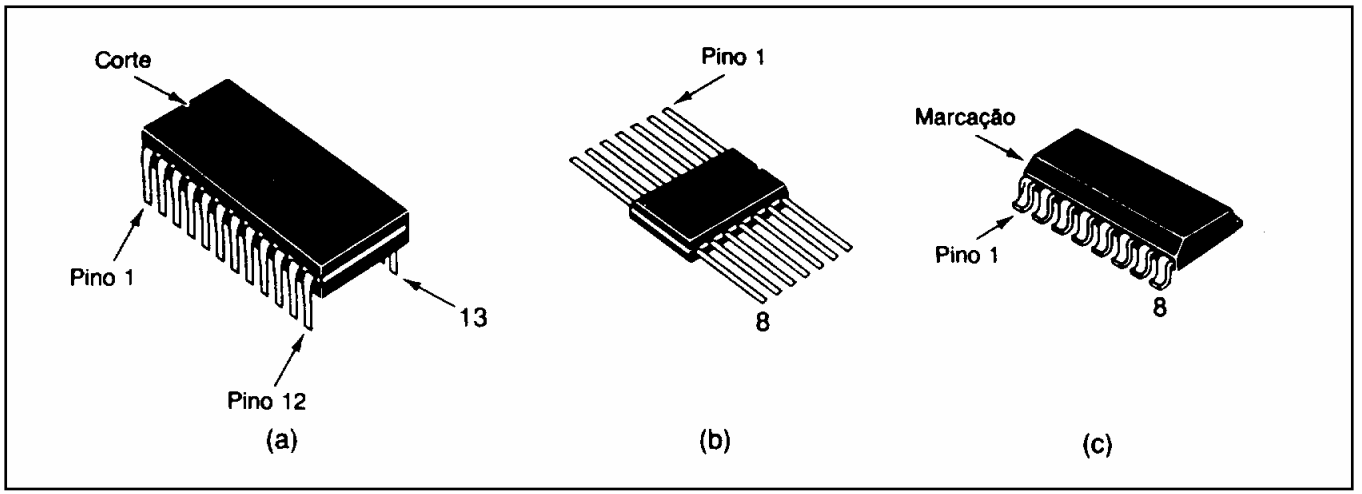
Portas Lógicas - Fabricação

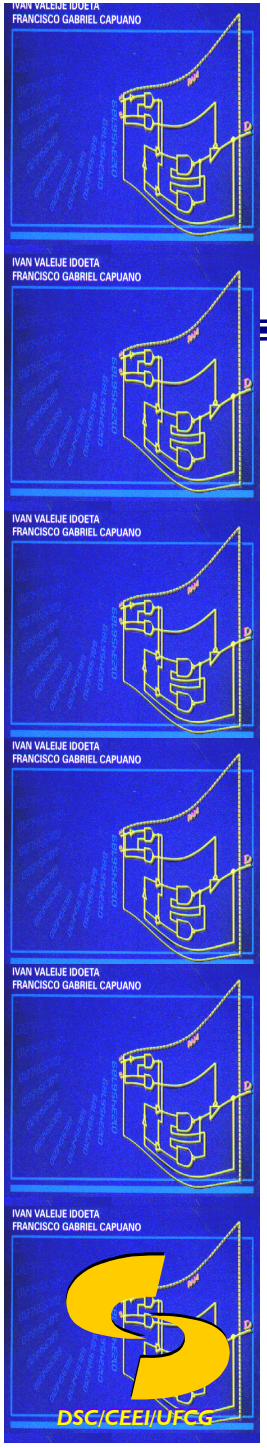
- O chip é montado dentro de um empacotamento cerâmico ou plástico e são construídas ligações do chip para os pinos externos do integrado.
- Encapsulamentos comuns para CIs:

  (a) DIP (*dual-in-line package*) de 24 pinos;

  (b) envoltório de cerâmica flexível de 14 pinos;

  (c) envoltório montado sobre a superfície (*surface-mount)*.

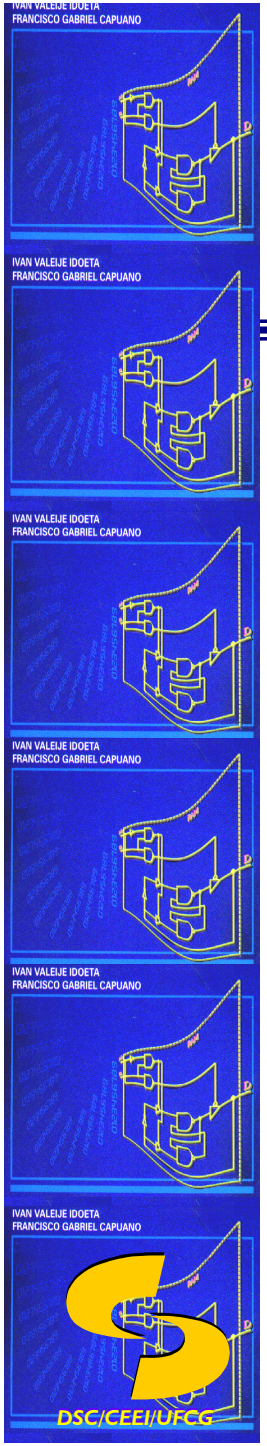




Portas Lógicas - Fabricação

- As famílias podem ser classificadas quanto ao seu grau de integração em **SSI, MSI, LSI, VLSI e ULSI.**

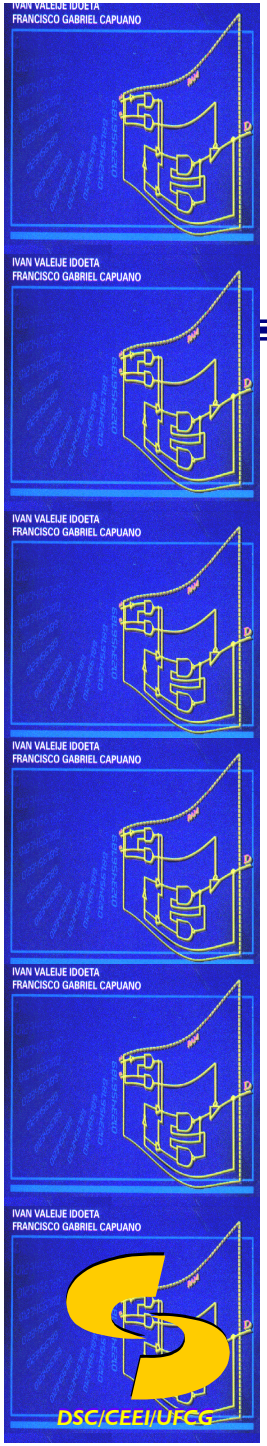
Nível de integração	Número de Portas	Aplicação
SSI (*Small-Scale Integration*) - Integração em pequena escala	Menos de 12	portas básicas simples.
MSI (*Medium-Scale Integration*) - Integração em média escala	Menos de 100	funções elementares, somadores, etc
LSI (*Large-Scale Integration*) - Integração em larga escala	até alguns milhares	pequenos processadores, etc.
VLSI (*Very Large-Scale Integration*)- Integração em escala muito larga.	a partir de alguns milhares	microprocessadores, etc.



Portas Lógicas - Fabricação

Circuitos Integrados Comerciais:

- As portas lógicas AND, OR, NAND e NOR podem ser encontradas comercialmente com duas, três, quatro ou oito entradas.
- A porta inversora, sempre possui uma entrada.



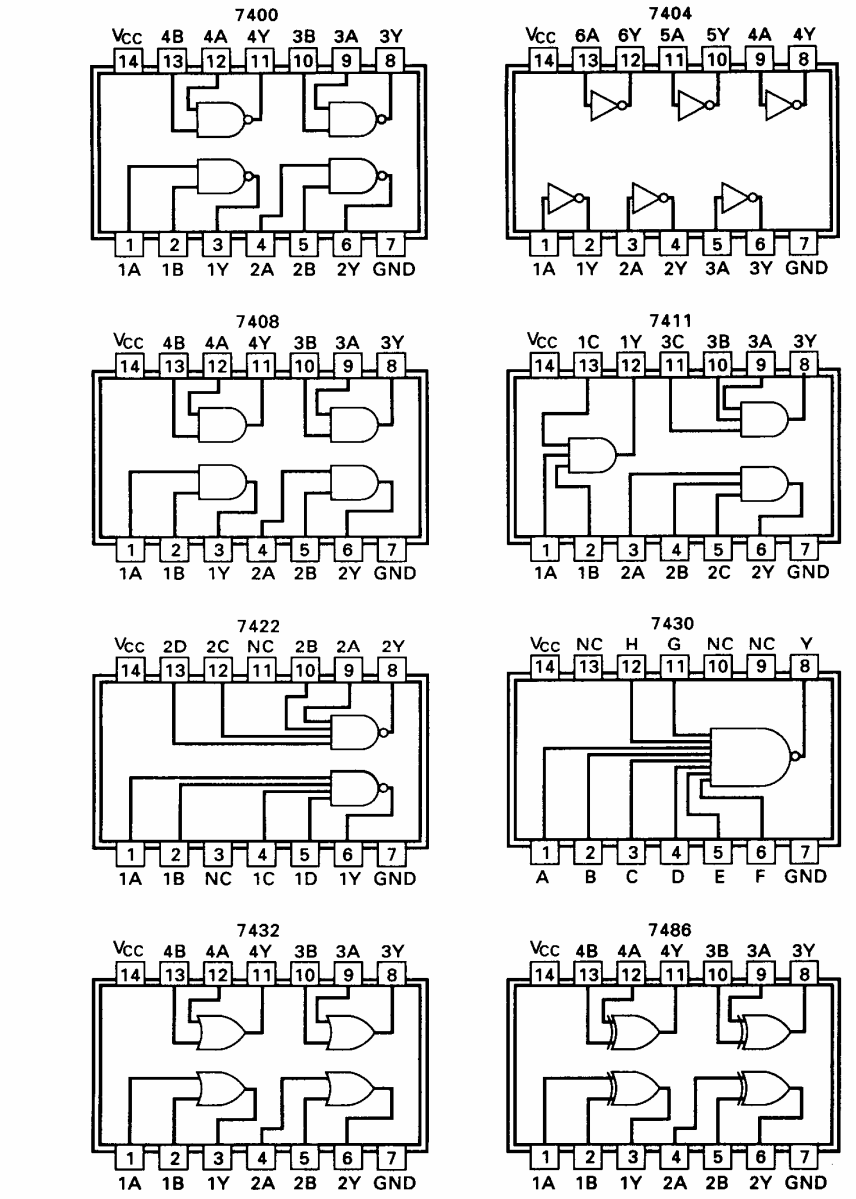
Portas Lógicas - Fabricação

Circuitos Integrados Comerciais:

Exemplos:

TTL	CMOS	Especificações
7400	4011	4 portas NAND de 2 entradas
7402	4001	4 portas NOR de 2 entradas
7404	4009	6 portas INVERSORAS
7408	4081	4 portas AND de 2 entradas
7432	4071	4 portas OR de 2 entradas
7486	4030	4 portas XOR de 2 entradas
7410	4023	3 portas NAND de 3 entradas
7427	4002	2 portas NOR de 4 entradas
7430	74C30	1 porta NAND de 8 entradas

Exemplos de CIs - TTL



Algumas pastilhas SSI. *Layouts* de pinos de *The TTL Data Book for Design Engineers*, direitos reservados de Texas Instruments Incorporated, 1976.