UNIVASSOURAS

CAMPUS UNIVERSITÁRIO
DE SAQUAREMA

UNIVASSOURAS

Curso de Graduação em Engenharia de Software

ENGENHARIA DE SOFTWARE
UNIVASSOURAS SAQUAREMA

Infraestrutura de TI para Engenharia de Software

Aula 3 parte II

Prof. Dr. André Saraiva
Doutor em Ciência da Computação
Mestre em sistemas Compurtacionais
Especialista em Arquitetura e Projeto de Cloud Computing

Lógica Digital

Álgebra Boolena

- **Disciplina Matemática -** utilizada para projetar e analisar comportamento de circuitos digitais em computadores e sistemas digitais
- Homenageia George Boole
  - Propôs os princípios básicos em 1854
- Análise: Modo econômico de descrever Circuito Digital
- Projeto: Dada uma função, a álgebra booleana pode ser aplicada para desenvolver uma implementação simplificada dessa função.

Lógica Digital

Variáveis Booleanas e Operações

- Uso de variáveis e operações
  - Uma variável pode assumir os valores:
    - 1 – TRUE
    - 0 – FALSE
  - Operações Lógicas básicas:
    - AND
    - OR
    - NOT

Lógica Digital

Variáveis Booleanas e Operações

- AND
  - O resultado é TRUE se ambos os operadores são TRUE.
  - Na ausência de parênteses tem precedência sobre OR.
  - Representação:
    - $\mathrm{A\ AND\ B} = \mathrm{A\ .\ B}$
    - $\mathrm{A\ AND\ B} = \mathrm{AB}$

Lógica Digital

Variáveis Booleanas e Operações

- OR
  - O resultado é TRUE se um ou ambos os operadores são TRUE.
  - Representação:
    - $\mathrm{A\ AND\ B} = \mathrm{A} + \mathrm{B}$
  - NOT
    - Inverte o valor do operando
    - $\mathrm{A} = \overline{\mathrm{A}}$

Lógica Digital

Variáveis Booleanas e Operações

- NAND
  - É o complemento (NOT) da função AND
  - $\mathrm{A\ NAND\ B} = \mathrm{NOT(A\ AND\ B)} = \overline{\mathrm{AB}}$
- NOR
  - É o complemento (NOT) da função OR
  - $\mathrm{A\ NOR\ B} = \mathrm{NOT(A\ OR\ B)} = \overline{\mathrm{A + B}}$

Lógica Digital

Operadores Booleanos

P	Q	NOT P ($\overline{P}$)	P AND Q ($P \cdot Q$)	P OR Q ($P + Q$)	P NAND Q ($\overline{P \cdot Q}$)	P NOR Q ($\overline{P + Q}$)	P XOR Q ($P \oplus Q$)
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

- Tabela Verdade dos Operadores Básicos

Lógica Digital

Operadores Booleanos

Postulados básicos		
$A \cdot B = B \cdot A$	$A + B = B + A$	Propriedade comutativa
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Propriedade distributiva
$1 \cdot A = A$	$0 + A = A$	Elementos de identidade
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$	Elementos de inversão
Outras identidades		
$0 \cdot A = 0$	$1 + A = 1$	
$A \cdot A = A$	$A + A = A$	
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$	Propriedade associativa
$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$	Teorema de DeMorgan

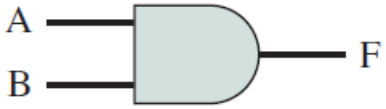
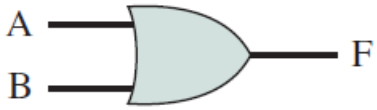
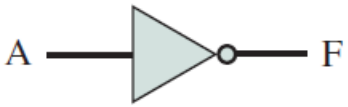
Lógica Digital

Portas Lógicas

- Uma porta lógica é um circuito eletrônico em que a saída é uma operação booleana
- As portas lógicas usadas na lógica digital são:
  - AND, OR, NOT, NAND, NOR, XOR
- Cada porta lógica tem:
  - Símbolo Gráfico
  - Função Algébrica
  - Tabela verdade

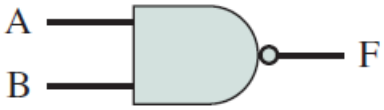
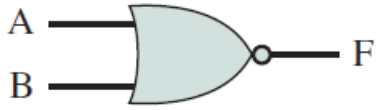
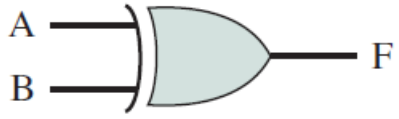
Lógica Digital

Portas Lógicas

Nome	Símbolo gráfico	Função algébrica	Tabela verdade
AND		$F = A \bullet B$ 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

Lógica Digital

Portas Lógicas

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		$F = A \oplus B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0

Lógica Digital

Portas Lógicas

- Conjuntos funcionalmente completos:
  - AND, OR, NOT
  - AND, NOT
  - OR, NOT
  - NAND
  - NOR

Lógica Digital

Portas Lógicas

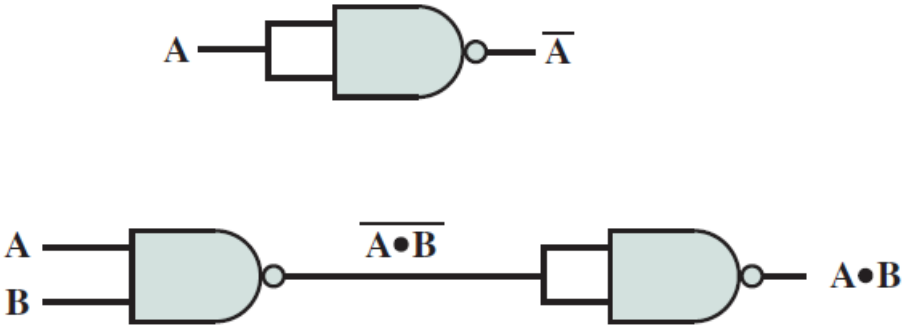
- Para as portas AND e NOT formarem um conjunto funcionalmente completo deve ser possível representar o OR usando apenas AND e NOT

- $A + B = \overline{\overline{A} + \overline{B}}$
- $A \text{ OR } B = \text{NOT } ((\text{NOT } A) \text{ AND } (\text{NOT } B))$

- Exercício para casa: representar a função AND utilizando as portas OR e NOT

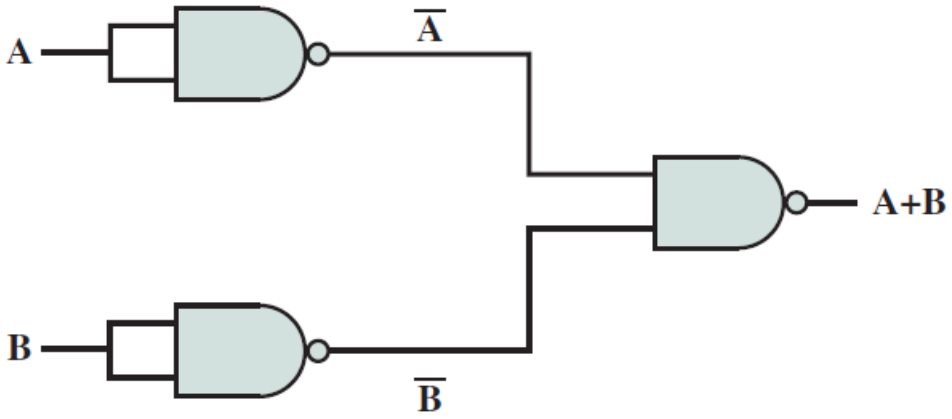
Lógica Digital

Uso de Portas NAND



Lógica Digital

Uso de Portas NAND



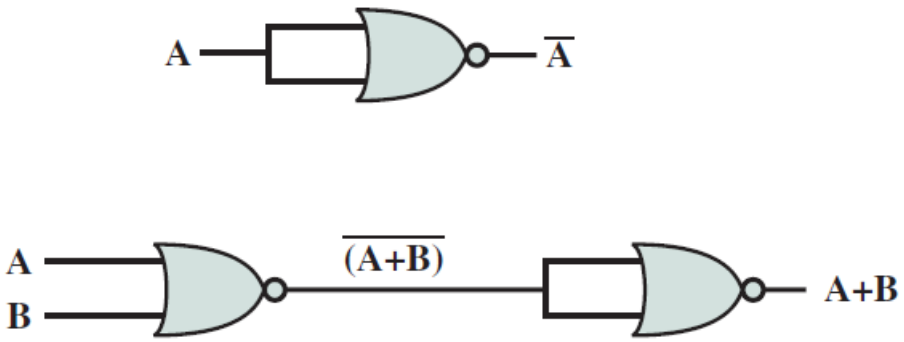
Lógica Digital

Portas Lógicas

- Esses desenhos demonstram o uso das portas NAND para as operações NOT, AND e OR
- Similarmente os próximos slides irão apresentar o uso de NOR para as operações NOT, AND e OR

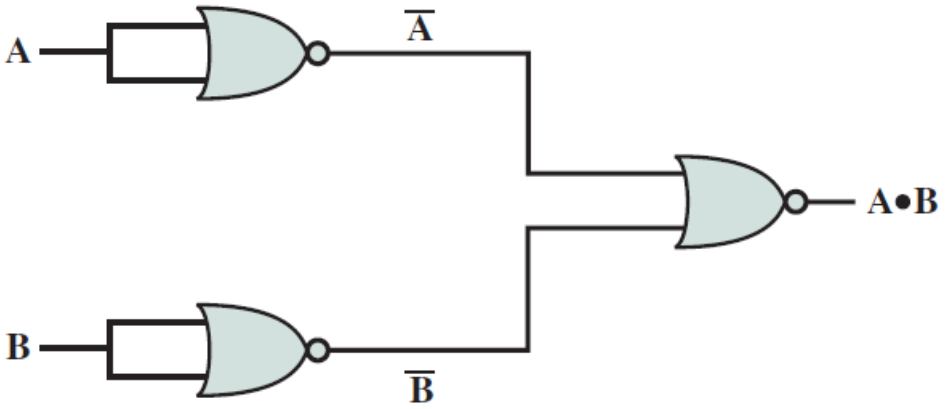
Lógica Digital

Uso de Portas NOR



Lógica Digital

Uso de Portas NOR



Lógica Digital

Função Booleana de 3 Variáveis

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

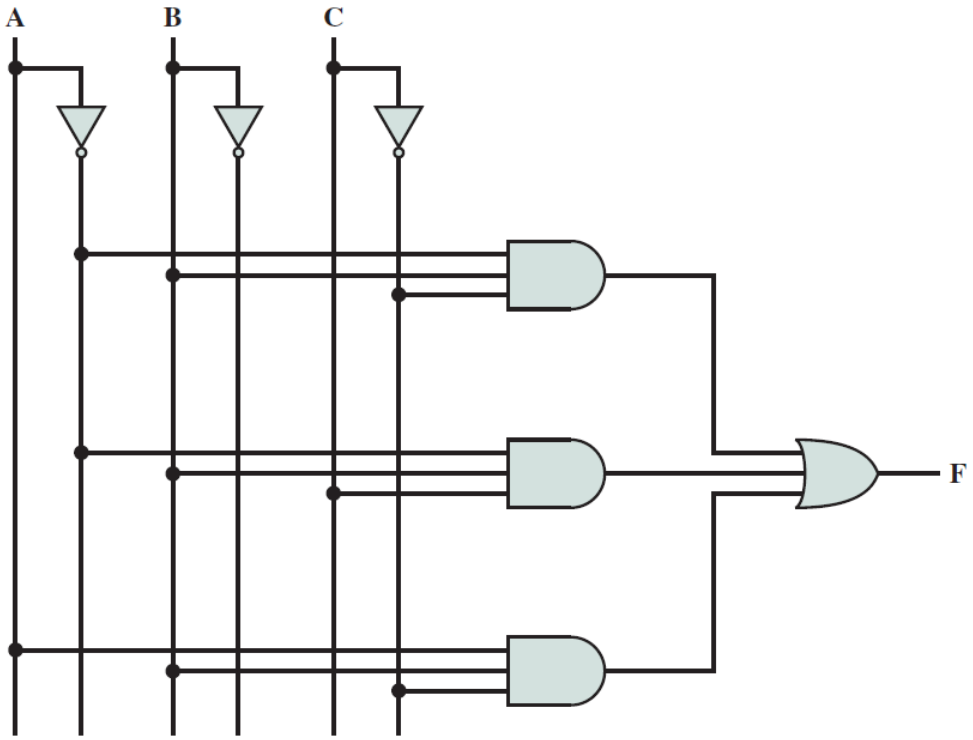
Lógica Digital

Portas Lógicas

- Nessa Tabela Verdade existem 3 combinações que fazem com que F seja 1.
- Pode ser representada por:
  - $F = \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$
- Forma de expressão conhecida como **Soma dos Produtos (SOP)**

Lógica Digital

Implementação da Soma dos Produtos



Lógica Digital

Produto das Somas (POS)

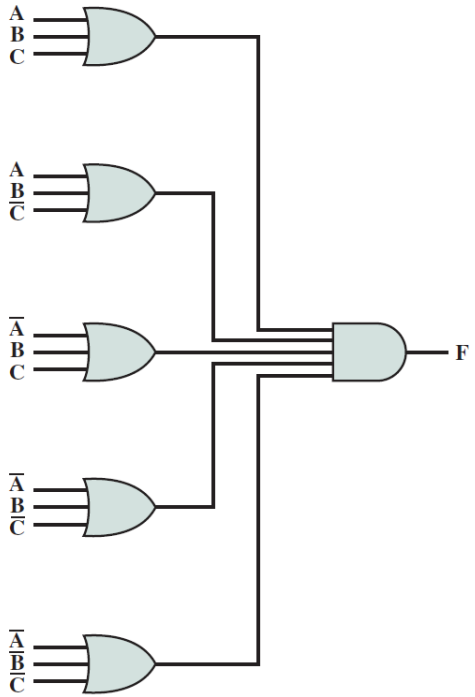
- Mesma expressão F pode ser representada por:

- $F = (A + B + C).(A + B + \bar{C}).(\bar{A} + B + C).(\bar{A} + B + \bar{C}).\ (\bar{A} + \bar{B} + \bar{C})$

- SOP tem um termo para cada 1
- POS tem um termo para cada 0

Lógica Digital

Implementação do Produto das Somas



Lógica Digital

Circuitos Combinacionais

- Mesma expressão pode ter mais de 1 representação
- Interessante usar a forma mais simples
- Alguns casos, pode ser interessante utilizar apenas 1 tipo de porta

- Métodos de Simplificação:
  - Simplificação Algébrica
  - Mapa de Karnaugh

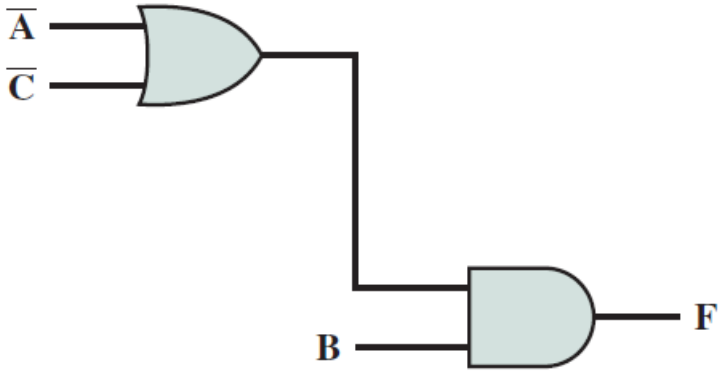
Lógica Digital

Simplificação Algébrica

- A mesma expressão F anterior pode ser expressa por:
  - $F = B(\bar{A} + \overline{C})$

Lógica Digital

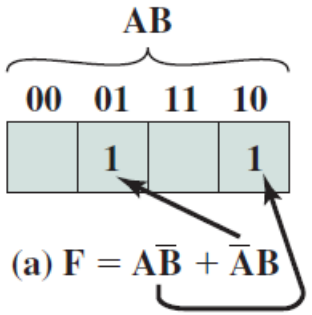
Implementação Simplificada



Lógica Digital

Mapa de Karnaugh

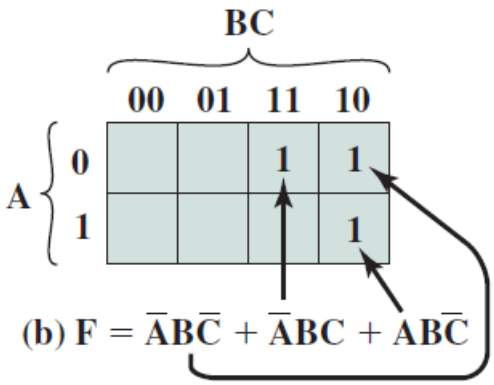
- Maneira conveniente de representar uma função booleana
- Mapa é uma matriz de n^2 quadrados
  - Todas as combinações possíveis de n variáveis



Lógica Digital

Mapa de Karnaugh

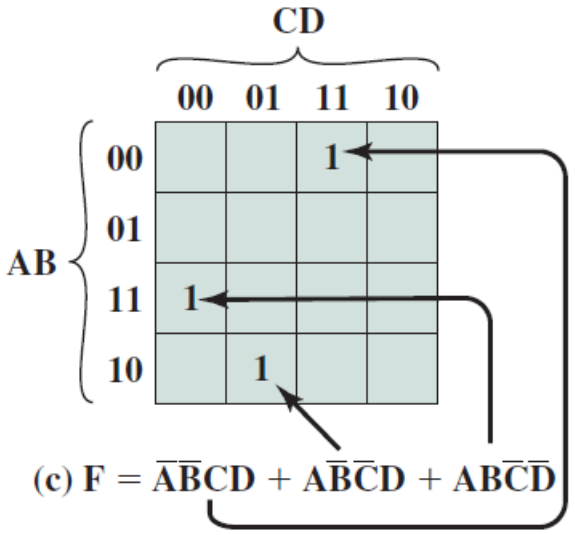
- 3 variáveis = 8 quadrados



Lógica Digital

Mapa de Karnaugh

- 4 variáveis = 16 quadrados



ROM – Read Only Memory

- Circuitos combinacionais são conhecido como circuitos sem memória.
- Saída depende apenas da entrada
- Exceção é a **memória somente de leitura**
- Informação armazenada em ROM é permanente.
- Criada durante o processo de fabricação.
- Pode ser implementada com um decodificador e conjunto de portas OR

ROM – Tabela Verdade

Entrada				Saída			
X_1	X_2	X_3	X_4	Z_1	Z_2	Z_3	Z_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Circuitos Sequenciais

- Circuitos combinacionais, com exceção da ROM, não proporcionam memória ou informação de estado

- Dessa forma, em alguns casos, é necessário a utilização de circuitos sequenciais

- A saída atual de um circuito sequencial depende da entrada e do estado atual do circuito

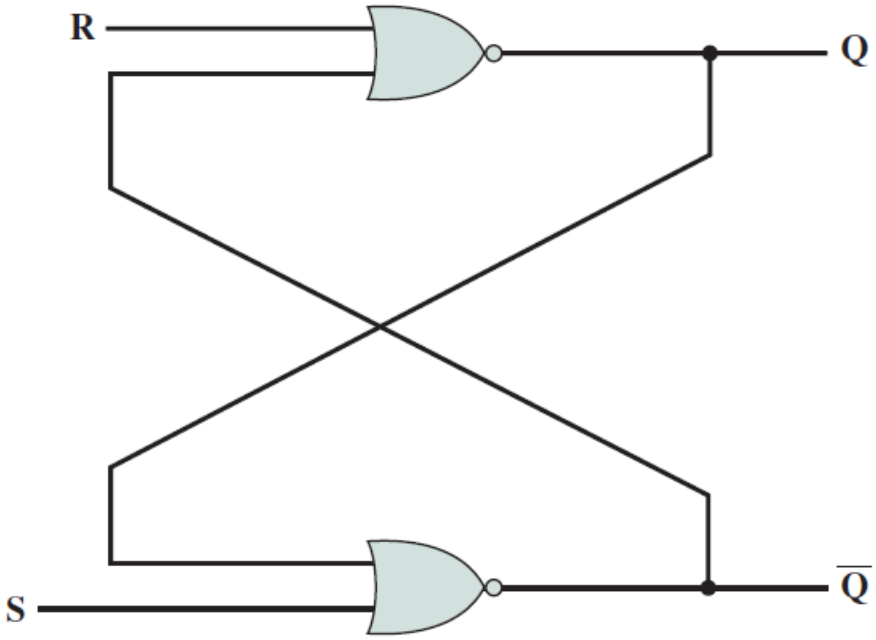
Flip-flops

- Circuito sequencial mais simples
- Dispositivo biestável
- Existe em dois estados e na ausência de entrada permanece no estado
- Funciona como memória de 1 bit
- Tem duas saídas : uma é sempre o complemento da outra.

Flip-flop S-R

- Duas entradas:
  - S (set)
  - R (reset)
- Duas saídas:
  - Q e $\bar{Q}$

Flip-flop S-R



Flip-flop S-R

- Demonstração da estabilidade:
  - Supondo S e R iguais a 0.
  - Entradas porta NOR inferior são Q = 0 e S = 0
  - Dessa maneira saída $\bar{Q} = 1$
  - Entradas porta NOR superior são e $\bar{Q} = 1$ e R = 0
  - A saída será Q = 0
  - Ou seja, permanece estável para R = S = 0
- Usando raciocínio análogo para Q = 1, vemos que continua R = S = 0

Flip-flop S-R

- Alterando o valor
  - Supondo S e R iguais a 0, Q = 0 e $\bar{Q} = 1$.
  - Mudando S para 1.
  - As entradas NOR inferior são S = 1 e Q =0.
  - Após Δt saída da porta NOR inferior será $\bar{Q} = 0$.
  - Nesse instante as entradas de NOR superior será
  - R = 0, $\bar{Q} = 0$
  - Após outro Δt a saída Q se torna 1.
  - Já sabemos que é estável.

✓ ***Atividades***

- ✓ Ler o capítulo 11 da bibliografia a seguir:

STALLINGS, W. Arquitetura e Organização de Computadores. 10ª edição. São Paulo: Pearson, 2010.

Motivação: Cai na prova. ☺

Contato

Professor:
André Saraiva, DSc

E-mail:
andre.saraiva@univassouras.edu.br