

SSC0101 - ICC1 – Teórica

Introdução à Ciência da Computação I

**Conceitos básicos
sobre computadores
(continuação)**

Prof. Vanderlei Bonato
Prof. Cláudio Fabiano Motta Toledo

Sumário

- ✓ O que é um computador e onde podemos encontrá-los?
 - ✓ Divisão: Hardware (HW) e Software (SW)
 - ✓ Sistema Operacional (SO)
 - ✓ Representação Interna de dados
 - ✓ Linguagem de programação
 - ✓ Compiladores
-

Representação interna de dados

- Toda a informação é representada por níveis lógicos '0' e '1', sendo que no dispositivo eletrônico esses níveis são armazenados por meio de transistores em estado aberto ou fechado, ou seja, operando em modo "chave". Por exemplo, '0' = aberto e '1' = fechado.
 - O valor lógico 0 ou 1 é definido como bit
 - Oito bits formam um byte
 - Assim, qualquer informação armazenada no computador é representada por valores binários
-

Conversão de decimal para binário

Convert $(857)_{10}$

				Remainder	
$857 \div 2$	$=$	428	1		LSB
$428 \div 2$	$=$	214	0		
$214 \div 2$	$=$	107	0		
$107 \div 2$	$=$	53	1		
$53 \div 2$	$=$	26	1		
$26 \div 2$	$=$	13	0		
$13 \div 2$	$=$	6	1		
$6 \div 2$	$=$	3	0		
$3 \div 2$	$=$	1	1		
$1 \div 2$	$=$	0	1		MSB

Result is $(1101011001)_2$

Conversão de binário para decimal

$$K = k_{n-1} k_{n-2} \dots k_1 k_0$$

$$V(k) = \sum_{i=0}^{n-1} k_i \times r^i$$

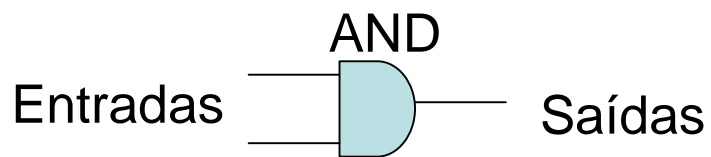
Onde:

K é o valor em binário, $r = 2$ e V o valor decimal

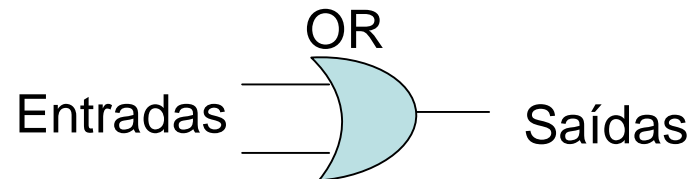
Representação interna de dados

- Internamente, o computador manipula esses bits por meio de portas lógicas
 - Porta lógica é um dispositivo que fornece a saída de uma operação booleana a partir de suas entradas
 - Portas lógicas são implementadas por transistores
-

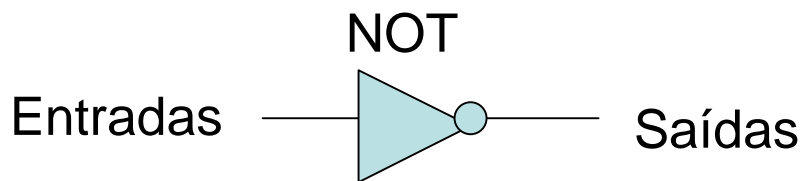
Portas lógicas: AND, OR, XOR, NOT



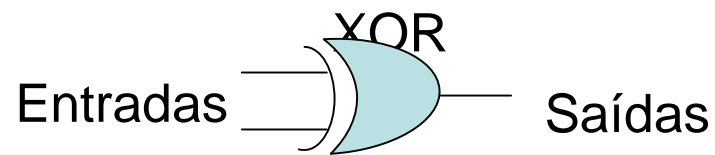
Entradas	Saídas
0 0	0
0 1	0
1 0	0
1 1	1



Entradas	Saídas
0 0	0
0 1	1
1 0	1
1 1	1

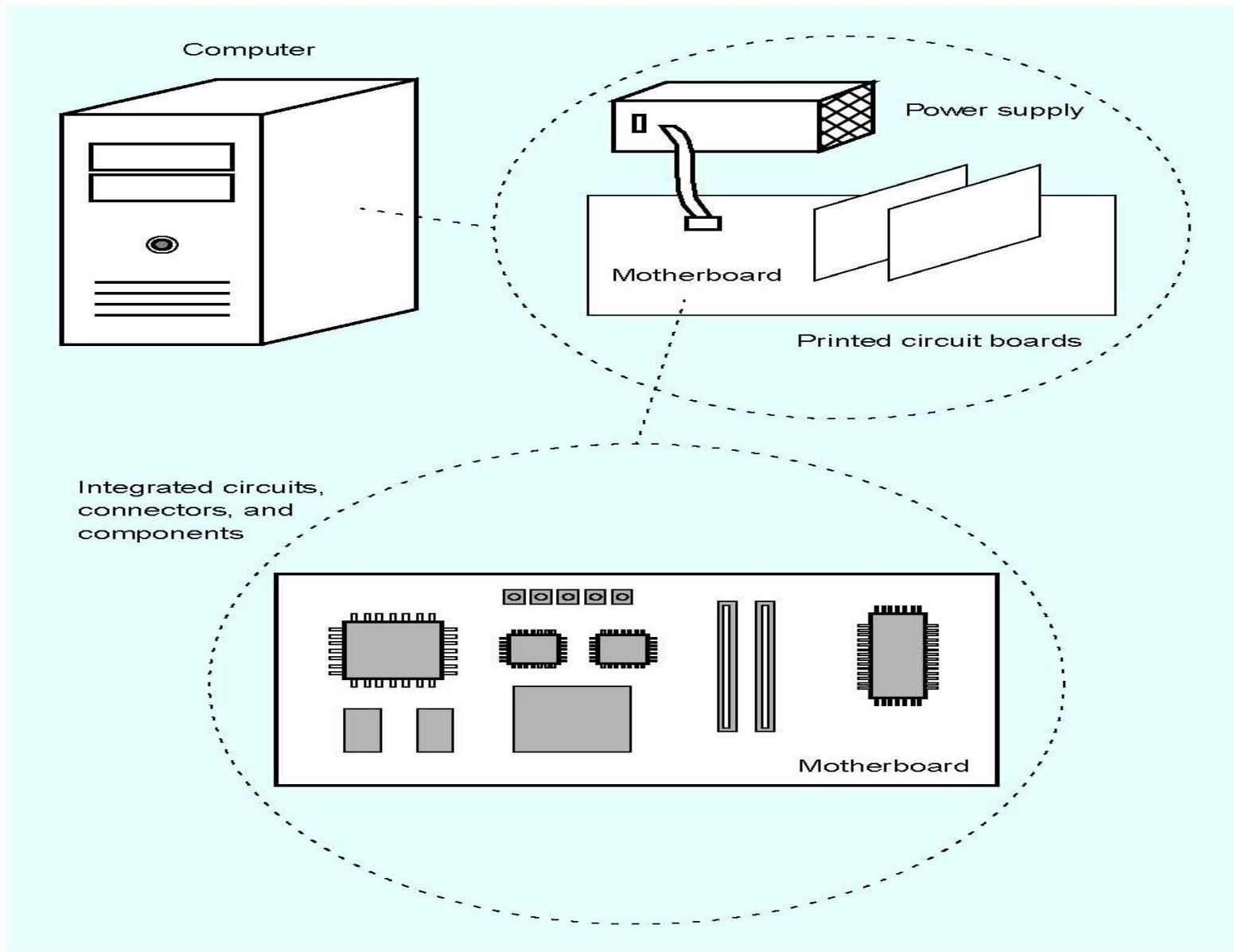


Entradas	Saídas
0	1
1	0

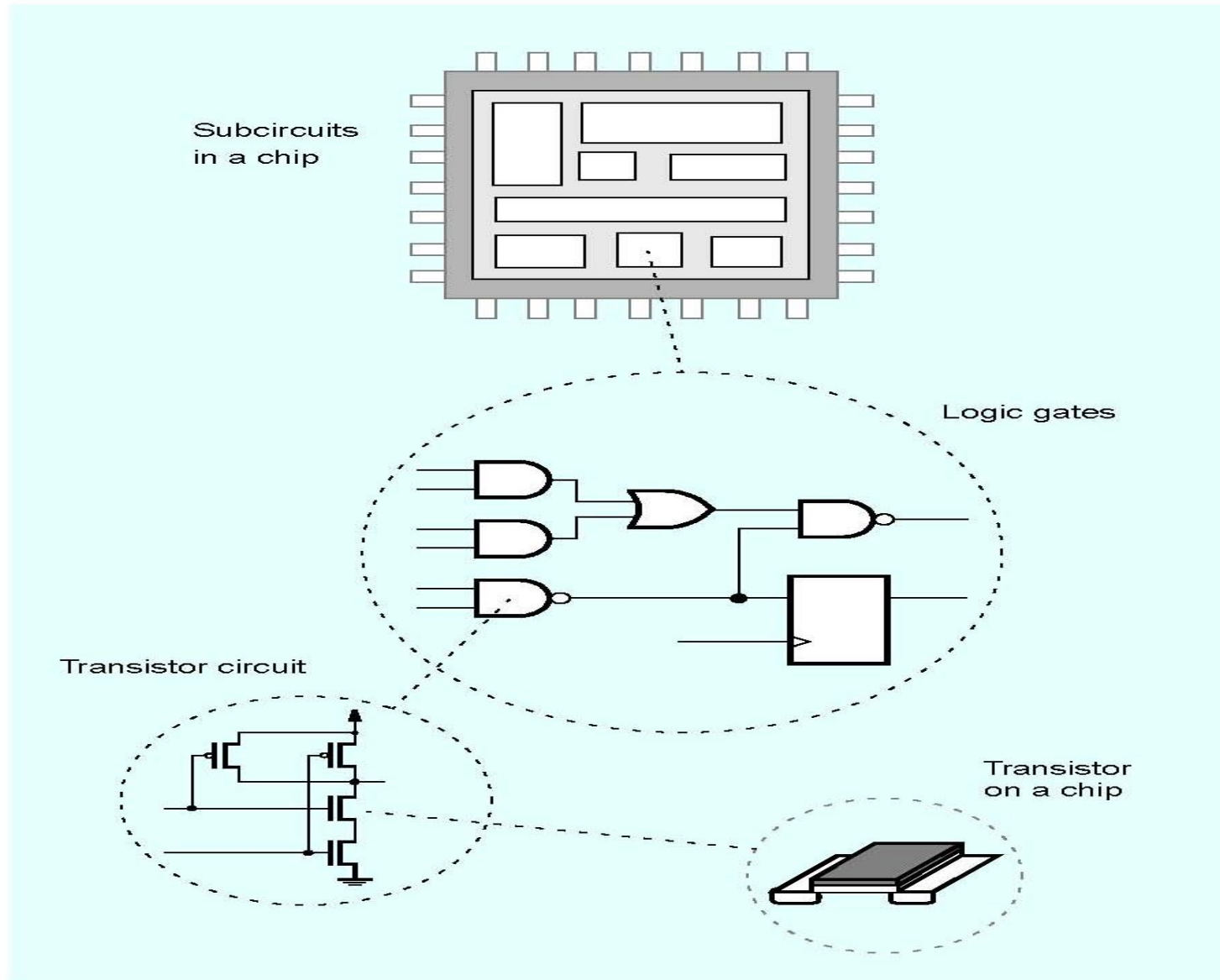


Entradas	Saídas
0 0	0
0 1	1
1 0	1
1 1	0

Sistema de hardware digital de um computador (parte 1:2)



Sistema de hardware digital de um computador (parte 2:2)



Dispositivo de armazenamento

- Células ou posições de memórias são as unidades manipuláveis em que estão organizados os circuitos de armazenamento da memória principal
- Cada célula tem um byte ou oito bits
- 1 Mega Byte: $2^{20} = 1.048.576$ bytes.

Célula 0	Célula 1	Célula 2	Célula 3	...	Célula n
10111010	00111101	11110001	01010010	...	00000111

Dispositivo de armazenamento

- Se a memória estiver dividida em células de tamanho de 1 byte, uma cadeia de 16 bits é armazenada em duas células consecutivas
 - Cada célula é individualmente referenciada, acessada e modificada
 - As células de uma memória do tipo RAM (Random Access Memory) são referenciadas por endereços, os quais podem ser apresentados em qualquer ordem
-

Linguagem de programação

- ✓ Um programa pode ser executado por qualquer dispositivo capaz de interpretar e executar as instruções de que é formado
 - ✓ Linguagens de programação foram projetadas para se aproximar das linguagens usadas por seres humanos
 - ✓ Raramente linguagem de máquina é usada para desenvolver um programa, porém, algumas vezes, para aumentar o desempenho, parte de um programa pode ser desenvolvido dessa forma
-

Evolução das linguagens de programação

- Classificadas em cinco gerações (GL)
- 1GL: linguagem de máquina
- 2GL: assembly

Linguagem de 1a geração (1GL)

- Representadas pelas linguagens de máquina
 - Programadores escreviam em papel os programas e depois traduziam para utilização em máquinas. Posteriormente, o processo de tradução foi identificado como um procedimento a ser executado pela máquina
-

Linguagem de 2a geração (2GL)

- Representadas pelas linguagens de montagem
 - O uso de mnemônicos foi formalizado como uma **linguagem de montagem (*assembly language*)**
 - Um **programa montador (*assembler*)** foi desenvolvido para **traduzir** os programas escritos em linguagem de montagem para linguagem de máquina
 - O programa recebeu o nome de montador porque sua tarefa era a de montar instruções de máquinas a partir de códigos de operação obtidos da conversão de mnemônicos e identificadores
-

Linguagem de 3a geração (3GL)

- Linguagens de alto nível, tais como Cobol e C
 - Linguagem independente da arquitetura da máquina
 - Necessidade do uso de tradutores/compiladores mais sofisticados. Uma única instrução de alto nível pode gerar várias instruções *assembly*
 - Necessidade de compiladores capazes de identificar erros e de otimizar o código automaticamente
-

Linguagem de 4a geração (4GL)

- São linguagens não procedimentais
 - O foco está no que fazer e não em como fazer
 - Ex.: SQL (Structured Query Language) para acessar banco de dados
-

Linguagem de 5a geração (5GL)

- Usada em referência ao conceito de programação declarativa com ênfase em programação lógica
 - A idéia é permitir ao usuário do computador resolver um problema concentrando-se apenas na análise de suas características, em vez de determinar a forma como ele deve ser resolvido
 - Ex. PROLOG
-

Compilador

- ✓ O compilador traduz um programa em linguagem de alto nível (código fonte) para a linguagem *assembly* dependente da máquina
 - ✓ O código *assembly* gerado pelo compilador também é conhecido como **programa-objeto**
 - ✓ O **programa-objeto** é, de fato, um programa que contém algumas “pontas soltas”, as quais devem ser devidamente conectadas a outros programas-objetos, antes de se conseguir um programa executável
 - ✓ O programa ligador (*linker*) executa a tarefa de construir tais conexões. Ele une programas-objetos, rotinas do sistema operacional e outros softwares utilitários, para produzir um programa completo (módulo de carga). Esse programa é armazenado na forma de arquivo, no sistema de armazenamento de massa do computador
-

Compilador

- ✓ O programa carregador (loader) transfere para a memória o módulo de carga para que ele possa ser executado
 - ✓ Uma vez realizados os passos de tradução e ligação, o programa pode ser repetidamente carregado e executado, sem a necessidade de retornar à versão original
 - ✓ No entanto, se for preciso executar alguma alteração no programa, esta deve ser feita no programa-fonte, o qual, por sua vez, é novamente traduzido e ligado, resultando um novo módulo de carga, que incorpora as alterações efetuadas
-

Compilador

- ✓ Há também compiladores que traduzem um código fonte para outro código fonte, conhecidos como “source to source”
 - ✓ Note que independente do tipo ou nível do código gerado pelo compilador, a semântica do mesmo deve ser mantida
-

Referências

- Fedeli, R.D.; Polloni, E.G.F.; Peres, F.E. Introdução à Ciência da Computação. 2ª Edição. São Paulo: Cengage Learning, 2010. 250p.
- Brookshear, J.G. *Ciência da Computação: uma visão abrangente*. 5ª Edição. Porto Alegre: Bookman, 2000. 500p.
- BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with VHDL Design, McGraw Hill, 2005.
-