

Computação

Introdução á Computação

Marcial Porto Fernandez
Mariela Inés Cortés



Química



Ciências
Biológicas



Artes
Plásticas



Computação



Física



Matemática



Pedagogia



Computação

Introdução á Computação

Marcial Porto Fernandez
Mariela Inés Cortés

3ª edição
Fortaleza - Ceará



2015



Química



Biologia



Artes



Computação



Física



Matemática



Pedagogia

Copyright © 2015. Todos os direitos reservados desta edição à UAB/UECE. Nenhuma parte deste material poderá ser reproduzida, transmitida e gravada, por qualquer meio eletrônico, por fotocópia e outros, sem a prévia autorização, por escrito, dos autores.

Editora Filiada à



Presidenta da República

Dilma Vana Rousseff

Ministro da Educação

Renato Janine Ribeiro

Presidente da CAPES

Carlos Afonso Nobre

Diretor de Educação a Distância da CAPES

Jean Marc Georges Mutzig

Governador do Estado do Ceará

Camilo Sobreira de Santana

Reitor da Universidade Estadual do Ceará

José Jackson Coelho Sampaio

Vice-Reitor

Hidelbrando dos Santos Soares

Pró-Reitora de Graduação

Marcília Chagas Barreto

Coordenador da SATE e UAB/UECE

Francisco Fábio Castelo Branco

Coordenadora Adjunta UAB/UECE

Eloisa Maia Vidal

Diretor do CCT/UECE

Luciano Moura Cavalcante

Coordenador da Licenciatura em Informática

Francisco Assis Amaral Bastos

Coordenadora de Tutoria e Docência em Informática

Maria Wilda Fernandes

Editor da EdUECE

Erasmo Miessa Ruiz

Coordenadora Editorial

Rocylânia Isídio de Oliveira

Projeto Gráfico e Capa

Roberto Santos

Diagramador

Francisco José da Silva Saraiva

Conselho Editorial

Antônio Luciano Pontes

Eduardo Diatahy Bezerra de Menezes

Emanuel Ângelo da Rocha Fragoso

Francisco Horácio da Silva Frota

Francisco José Camelo Parente

Gisafran Nazareno Mota Jucá

José Ferreira Nunes

Liduina Farias Almeida da Costa

Lucili Grangeiro Cortez

Luiz Cruz Lima

Manfredo Ramos

Marcelo Gurgel Carlos da Silva

Marcony Silva Cunha

Maria do Socorro Ferreira Osterne

Maria Salete Bessa Jorge

Silvia Maria Nóbrega-Therrien

Conselho Consultivo

Antônio Torres Montenegro (UFPE)

Eliane P. Zamith Brito (FGV)

Homero Santiago (USP)

Ieda Maria Alves (USP)

Manuel Domingos Neto (UFF)

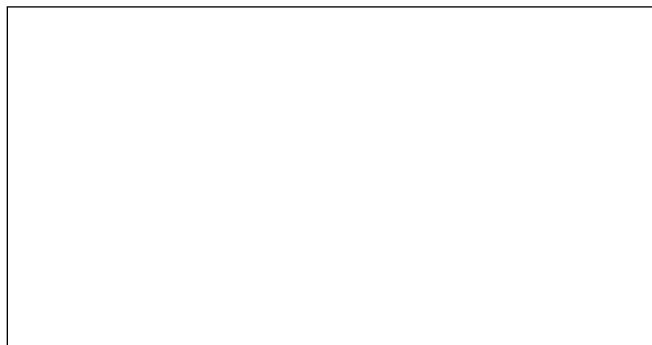
Maria do Socorro Silva Aragão (UFC)

Maria Lírída Callou de Araújo e Mendonça (UNIFOR)

Pierre Salama (Universidade de Paris VIII)

Romeu Gomes (FIOCRUZ)

Túlio Batista Franco (UFF)



Editora da Universidade Estadual do Ceará – EdUECE

Av. Dr. Silas Munguba, 1700 – Campus do Itaperi – Reitoria – Fortaleza – Ceará

CEP: 60714-903 – Fone: (85) 3101-9893

Internet: www.uece.br – E-mail: eduece@uece.br

Secretaria de Apoio às Tecnologias Educacionais

Fone: (85) 3101-9962

Sumário

Parte 1 - Histórico, aplicações e conceitos da Tecnologia da Informação	5
Capítulo 1 - História e evolução dos computadores	7
Introdução	7
1. Precursores do computador	8
2. As máquinas de primeira geração (1930-1958)	15
3. Computadores de segunda geração (1955-1965)	17
4. Computadores de terceira geração (1965-1980)	18
5. Computadores de quarta geração (1980 - ...)	19
Capítulo 2 - Tecnologias de computadores e suas aplicações	23
1. Computador descartável	24
2. Microcontrolador	25
3. Computador de jogos	27
4. Computador pessoal (PC)	28
5. Servidor	28
6. Conjunto de estações de trabalho	29
7. <i>Mainframe</i> - Supercomputadores	30
Capítulo 3 - Introdução à ciência da computação	31
Capítulo 4 - Tecnologia da Informação	35
Parte 2 - Representação e processamento da informação	39
Capítulo 5 - Representação e processamento da informação	41
Introdução	41
1. Identificando os bits	42
2. <i>Strings</i> e vetores	42
3. Representação de dados	43
Capítulo 6 - Sistemas de Numeração	45
1. O sistema binário	45
2. O sistema hexadecimal	47
Capítulo 7 - Aritmética binária	51
1. Aritmética de Números Inteiros Positivos	51
2. Aritmética de Números Inteiros Negativos	54
3. Números de ponto flutuante	56
Capítulo 8 - Portas lógicas	59
1. Portas básicas	59
2. Portas compostas	61

Parte 3 - Arquitetura de computadores	67
Capítulo 9 - Arquitetura de computadores	69
Introdução	69
Capítulo 10 - Unidade central de processamento e memória	75
1. Processador.....	75
2. Barramento de dados.....	78
3. Memória	78
Capítulo 11 - Dispositivos de armazenamento	81
1. Disco rígido	81
2. Disquete flexível	82
3. Cartões da memória Flash	82
4. CD-ROM.....	83
Capítulo 12 - Sistemas de entrada e saída	85
1. Dispositivos de Entrada	85
2. Dispositivos Apontadores	86
3. Tablet gráfico	87
4. Scanner.....	88
5. Microfone	88
6. Dispositivos de saída	88
7. Dispositivos de Entrada e Saída	91
Parte 4 - Conjunto de instruções e endereçamento	95
Introdução	97
Capítulo 13 - Instruções	101
Capítulo 14 - Instruction Set Architecture - ISA	105
1. Numero de operandos e localização	106
Capítulo 15 - Modos de endereçamento	111
Capítulo 16 - CISC versus RISC	117
1. Princípios de projeto para computadores modernos.....	118
Sobre os autores	121

Parte

1

**Histórico,
aplicações e conceitos da
Tecnologia da Informação**

História e evolução dos computadores

Introdução

Neste capítulo será apresentada uma resenha histórica da evolução dos computadores, partindo do seu surgimento no formato mecânico até os sistemas usados na atualidade, destacando as suas principais características estruturais. Em sincronia com a rápida evolução da área, o desenvolvimento de novas tecnologias de apoio e sua permanente evolução impulsionou a imediata difusão e adoção nas mais diversas áreas de aplicação.

Hoje em dia a utilização de sistemas baseados em computador é uma necessidade básica para o desenvolvimento de qualquer atividade. O foco principal destes sistemas é a produção, tratamento, e comunicação de informação. A partir desta nova realidade social, surge o conceito de Tecnologia da Informação (TI) englobando o conjunto de todas as atividades que, baseadas em recursos computacionais, participam da geração, processamento e uso da informação.

O computador é uma máquina capaz de receber, armazenar, tratar e produzir informações de forma automática. Ao longo da sua evolução, que teve início no século 16, muito foi exigido em relação ao desempenho dos computadores, atingindo atualmente grande rapidez e precisão no desempenho das operações.

Para que uma máquina possa executar uma tarefa é preciso criar um **algoritmo**¹ definindo os passos em que a tarefa pode ser executada. No entanto, para que uma máquina possa executar a tarefa descrita pelo algoritmo, esse algoritmo deve ser traduzido para uma linguagem de programação para que possa ser compreendido pela máquina.

De uma forma geral, programas são coletivamente denominados de software, enquanto que os equipamentos físicos necessários para a execução de tais programas, hardware.

Originalmente estudados no contexto da matemática, os algoritmos eram projetados para descrever o raciocínio necessário à resolução de problemas de um determinado tipo, de forma unificada. Com o passar do tem-

¹ Mais precisamente, um algoritmo determina um conjunto ordenado e não-ambíguo de passos que definem uma atividade finita.

po, e a partir da sua popularização nos diferentes âmbitos da sociedade, as máquinas se tornaram responsáveis pela execução de tarefas cada vez mais complexas, propiciando o surgimento da disciplina Ciência da Computação com o objetivo de elaborar, nas mais diversas áreas de aplicação, algoritmos cada vez mais complexos e eficientes.

A complexidade dos algoritmos pode encontrar limitações na própria tecnologia, ou seja, na própria capacidade da máquina. Algoritmos complexos podem demandar recursos da máquina que não estejam disponíveis como, por exemplo, maior capacidade de memória do que a que a máquina pode oferecer. Esta necessidade impulsiona a pesquisa e o avanço tecnológico que também são áreas de estudo da Ciência da Computação.

1. Precursores do computador

A história da tecnologia se desenvolveu paralelamente e é quase tão antiga quanto a história da evolução humana. Um marco considerado fundamental neste contexto é o início das atividades agrícolas, a partir do desenvolvimento de melhores técnicas de domínio do homem sobre a natureza, em relação aos seus antecessores. Este marco estabelece uma mudança radical nos hábitos até então nômades do homem pré-histórico, para se tornar sedentário e garantir seu território para a agricultura e a criação de animais. Surgem assim aldeias e povoados e inicia-se uma era de grandes descobertas.

Com o desenvolvimento da lavoura tornou-se necessário um método para a contagem do tempo, delimitando as épocas de plantio e colheita. O primeiro passo do homem rumo à ciência e à tecnologia é a concepção da idéia de número. Inicialmente era utilizado um mecanismo de comparação entre conjuntos de elementos, passando a comparar por agrupamentos, na medida que os conjuntos cresciam em tamanho. No entanto, esta abordagem ainda não era muito prática, pois a contagem era feita utilizando-se os dedos das próprias mãos, mas foi importante por dar origem ao sistema de numeração de base decimal.

A compreensão do conceito de número como uma idéia abstrata, e não somente como uma qualidade de um determinado objeto, não foi um processo imediato. Conforme Bertrand Russell, “foram necessários muitos anos para se descobrir que um par de faisões e um par de dias eram ambos instâncias do número dois”. Esta dissociação representa o primeiro passo em direção a um sistema de notação, e daí para a aritmética. Para auxílio deste método, eram usados gravetos, contas ou marcas na parede.

Com a evolução da humanidade se fizeram necessárias novas invenções na tentativa de criar dispositivos mecânicos para auxiliar a fazer cálculos. Apesar dos computadores eletrônicos terem efetivamente aparecido somente na década de 40, os fundamentos em que se baseiam remontam a centenas ou até mesmo milhares de anos.

1.1. O Ábaco (aprox. 3500 a.C.)

A palavra *cálculo* tem sua origem no termo latino *calculus*, que há milhares de anos servia para denominar pequenas pedras que eram usadas para contar deslizando-se por sulcos cavados no chão. A partir desse elemento de cálculo, outros similares apareceram em diversos lugares do mundo, sendo chamados de ábaco. O mais antigo data de aproximadamente 3500 a.C. e surgiu provavelmente no Mediterrâneo. Utilizado pelos mercadores para contar e calcular, o ábaco era tão eficiente que se propagou por toda parte, antes do século XVII, e nenhum outro instrumento de cálculo podia competir com ele.

Por volta do ano 2600 a.C. apareceu o ábaco chinês que evoluiu rapidamente e foi chamado em sua forma final de Suan-Pan. Foi muito utilizado pelas civilizações egípcia, grega, chinesa e romana, tendo sido encontrado no Japão, ao término da segunda guerra mundial, chamado de Soroban.

O ábaco constituiu, portanto, o primeiro dispositivo manual de cálculo, e é utilizado ainda nos dias de hoje. Essa máquina é bastante simples e serve para representar números no sistema decimal e realizar operações com eles.

A figura 1 apresenta um ábaco, que consiste numa moldura retangular dividida em 2 partes; possui uma vareta vertical para cada dígito, sendo que cada vareta tem em sua parte inferior 5 anéis que, em repouso, ficam para baixo, e na parte superior, 2 anéis que, em repouso, ficam para cima.

Os anéis são movidos de um lado para outro sobre as varetas, e suas posições representam valores numéricos. Cada unidade acrescentada a um dos dígitos do número é representada pelo movimento para cima de um dos anéis da parte inferior da vareta. Quando os 5 estão na parte de cima devem ser movidos para baixo.

O mesmo deve ser feito com os mesmos anéis na parte superior da mesma vareta. Se os dois anéis da parte superior estão para baixo, devem ser movidos para cima acrescentando-se uma unidade à vareta seguinte, à esquerda dessa vareta. O maior número que pode ser calculado depende do número de varetas.

Fazendo uma analogia com o computador, pode-se dizer que é por meio das posições dos anéis que este dispositivo representa e armazena dados. Os dados são ingressados quando os anéis são posicionados e a saída dos resultados consiste em observar as posições dos anéis.

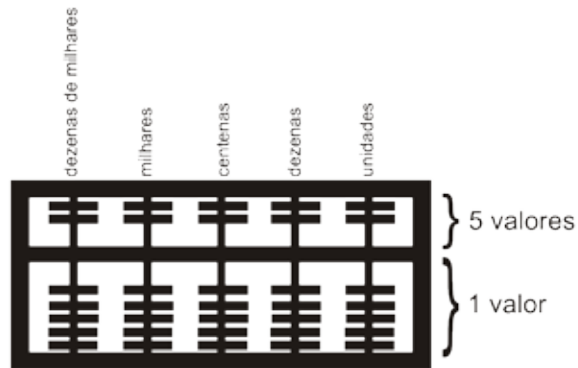


Figura 1 - Modelo ilustrativo de um ábaco

1.2. Logaritmos e os primeiros dispositivos mecânicos de cálculo

Passaram-se séculos sem que qualquer invenção ficasse registrada até que fossem criados tais bastões. John Napier (1550-1617), Barão de Merchiston, nobre escocês, teólogo e matemático, é bastante conhecido pela descoberta dos logaritmos. Também passou grande parte de sua vida inventando instrumentos para ajudar no cálculo aritmético, principalmente para o uso de sua primeira tabela de logaritmo, estabelecendo uma relação entre uma série geométrica e uma série aritmética.

Essas descobertas ajudaram a tornar mais simples os cálculos aritméticos, transformando operações de multiplicação em operações de soma, e operações de divisão em operações de subtração. A importância dos logaritmos não se resumia somente à simplificação dos cálculos, mas teve contribuição também na ciência e na tecnologia.

1.3. Auxílios mecânicos para cálculo

John Napier generalizou o procedimento tabular e construiu em 1617 um dispositivo simples e barato constituído de bastões de ossos (figura 2). Eram tabelas móveis de multiplicação e divisão feitas de marfim. Um conjunto completo de bastões de Napier consiste em 9 peças: uma para cada dígito de 1 a 9.

Cada uma de suas hastes é essencialmente uma coluna de uma tabela de multiplicação contendo os algarismos de 1 a 9 no quadrado superior e os oito quadrados restantes do bastão continham o produto deste número por 2,

3, 4, 5, 6, 7, 8 e 9 respectivamente. Para obter o produto, os dígitos de cada diagonal são somados da direita para a esquerda.

Apesar de dispositivos semelhantes aos bastões terem sido usados desde o final do século XVI, eles só apareceram documentados a partir de 1614.

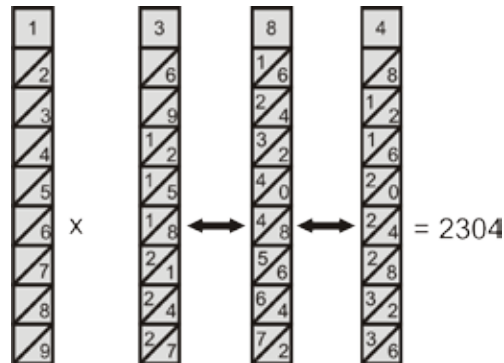


Figura 2 - Modelo ilustrativo dos bastões de Napier

1.4. Régua de cálculo

A partir das descobertas de Napier surgiu uma outra grande invenção, desenvolvida pelo brilhante matemático inglês Willian Oughtred, e tornada pública em 1638: a régua de cálculo. Ganhou sua forma atual, de uma régua que se move entre dois outros blocos fixos, por volta do ano de 1650, tendo sido esquecida por duzentos anos, para se tornar, no século XX, um grande símbolo de avanço tecnológico, com uso extremamente difundido.

Uma regra de cálculo é considerada como o primeiro dispositivo mecânico-analógico de computação. Com um formato tipicamente circular, este instrumento permite a realização de cálculos de forma aproximada, por meio de guias graduadas deslizantes. Existem diversas variantes da regra de cálculo dependendo da área de aplicação específica da engenharia. Embora todas elas realizem as operações básicas, para o caso da regra para uso em engenharia elétrica, por exemplo, podem existir escalas para conversão entre unidades de potência (kW), cálculo de tensão em condutores (V) e outras.

A régua de cálculo e as calculadoras mecânicas foram largamente utilizadas por engenheiros até 1970, quando surgiram as calculadoras eletrônicas cuja utilização era muito mais simples e seus resultados mais precisos. Um cálculo como 1345×3442 , por exemplo, é resolvido em poucos segundos com uma régua de cálculo. No entanto, o resultado é apenas uma indicação de que o valor é próximo de 4.650.000, quando o valor exato é 4.629.490.

1.5. Geração zero (século XVII): a evolução da calculadora

A calculadora decimal foi construída entre 1642 e 1644 por Blaise Pascal. Trata-se de um aparelho mecânico consistindo de uma caixa com rodas e engrenagens interligados que possibilitava a execução de operações aritméticas de soma e subtração. O resultado era apresentado numa seqüência de janelinhas.

Embora o aparelho se mostrasse muito útil, não obteve sucesso comercial, pois seu preço foi muito alto.

Em 1671, o filósofo e matemático alemão de Leipzig, Gottfried Wilhelm von Leibnitz (1646 - 1716) introduziu uma melhoria em relação à Calculadora de Pascal considerando o conceito de realizar multiplicações e divisões através de adições e subtrações sucessivas. A máquina, capaz de executar as quatro operações básicas da matemática, foi construída em 1694. No entanto sua operação apresentou-se muito deficiente e sujeita a erros, tendo sido, portanto, abandonada.

O sucesso comercial veio 1820 com a Calculadora a Arithmometer, projetada por Charles Xavier Thomas (1785-1870). A máquina era capaz de efetuar as 4 operações aritméticas básicas utilizando o mesmo princípio da calculadora de Leibnitz, precisando de assistência do usuário para a realização de cálculos de dividir. Até 1850 venderam-se cerca de 1500 Arithmometers.

1.6. Máquina diferencial de Babbage (1823)

Entre 1802 e 1822, Charles Babbage (1792-1871), um matemático e engenheiro britânico, previu que grande parte da informação poderia ser manipulada por máquina, caso fosse possível antes converter a informação em números e projetou um engenho para isso. Em particular, Babbage tinha a intenção de reduzir o trabalho de calculo e corrigir erros nas tabelas matemáticas da sua época.

Neste contexto, a chamada Máquina de diferenças foi projetada para computar e imprimir um conjunto de tabelas de funções (logaritmos, funções trigonométricas, etc.) sem a intervenção de um operador humano. A máquina seria movida a vapor, era baseada também no princípio de discos giratórios e operada por uma simples manivela. O projeto de Babbage é considerado o precursor dos modernos computadores eletrônicos digitais.

Ao operador cabia somente iniciar a cadeia de operações, e a seguir a máquina tomava seu curso de cálculos, preparando totalmente a tabela prevista. Em 1823, o governo britânico financiou a construção de uma nova versão, mas não obteve resultado satisfatório, devido aos limites do ferramental industrial da época. Embora tenham sido contratados especialistas em máquinas, o projeto exigia mecanismos altamente precisos e movimentos complexos.

Neste contexto, Babbage se viu obrigado a destinar parte dos seus recursos e tempo para desenhar peças e ferramentas no intuito de avançar no estado da arte da tecnologia, retardando assim o desenvolvimento do projeto. Após 10 anos de trabalho, tudo que Babbage havia conseguido era uma pequena máquina de 3 registros e 6 caracteres, sendo que deveria ser, de acordo com o projeto, uma máquina de 7 registros e 20 caracteres cada. Finalmente, em 1833, quando o governo inglês desistiu do financiamento, Charles Babbage parou de trabalhar em sua máquina.

O projeto de Babbage, no entanto, é considerado o precursor dos modernos computadores eletrônicos digitais.

1.7. Máquina analítica

Após vários anos de tentativas, Babbage concentrou seus esforços no projeto de uma máquina de cálculo universal capaz de executar qualquer tipo de cálculo (1833).

Babbage despertou para seu novo projeto observando a revolução produzida pelos teares de Jacquard. Utilizando a mesma estratégia, a Máquina Analítica seria programada através de um conjunto mutável de instruções, para calcular várias funções diferentes, processando padrões algébricos da mesma maneira que o tear processava padrões de desenhos. Este evento determina um marco na evolução do computador: surge o conceito de “programação”.

Saiba Mais



Teares de Jacquard (1801)

O francês Joseph-Marie Jacquard (1752-1834) introduziu a primeira máquina para substituir o trabalho humano. Era uma máquina que automatizava os processos mecânicos de fabricação de tecido. Para executar um determinado trançado, a fiandeira deveria ter um plano ou programa que lhe dissesse quais fios deveria passar por cima ou por baixo, em que momento repetir o desenho, etc. A evolução chave propiciada pela máquina de Jacquard era o uso de uma série de cartões cujas perfurações estavam configuradas para descrever o modelo a ser produzido.

A abordagem utilizada por Jacquard implementa o conceito de armazenamento de informações em placas perfuradas com o objetivo de controlar uma máquina de tecelagem. Este conceito será posteriormente estendido para sua aplicação no processamento de dados.

O sucesso foi total e em 1812 havia na França 11.000 teares de Jacquard, e despertou na sociedade daquela época, temor pelo desemprego, provocando uma grande reação popular contra essa espécie de pré-automatização.

A máquina era constituída de unidade de controle de memória aritmética e de entrada e saída, e sua operação era governada por um conjunto de cartões perfurados, de modo que, de acordo com os resultados dos cálculos intermediários, a máquina poderia saltar os cartões, modificando dessa forma o curso dos cálculos.

Babbage investiu toda sua fortuna pessoal e de seu filho, que com ele trabalhou durante anos, na construção de sua máquina Analítica, vindo a falecer em 1871, sem findar a construção.

1.8. A primeira programadora da história da computação

Babbage percebeu que para criar as instruções responsáveis pela execução dos cálculos precisaria de um tipo inteiramente novo de linguagem e a imaginou como números, flechas e outros símbolos. Neste cenário surge a importante contribuição de Ada Augusta Byron, condessa de Lovelace (1851 - 1852), filha do famoso poeta britânico Lorde Byron.

Ada Byron é uma das poucas mulheres a figurar na história do computador. Matemática talentosa, compreendeu o funcionamento da Máquina Analítica e escreveu os mais completos relatos sobre o processo. Ada compreendeu o alcance das novas invenções de Babbage, que diferentemente dos projetos anteriores, de funcionamento analógico (execução dos cálculos usando medidas), a Máquina de Babbage era digital (execução dos cálculos usando formulas numéricas).

Antes mesmo da máquina ser construída, Ada aproveitou seus conhecimentos em matemática para criar programas combinando funções lógicas e aritméticas para a máquina, tornando-se a primeira programadora de computador do mundo. Alguns dos conceitos fundamentais na programação de computadores largamente utilizados até os dias de hoje, tais como sub-rotinas, laços e saltos, já foram utilizados por Ada na programação para a Máquina analítica.

Ada também colaborou com a documentação do projeto da Máquina de Babbage, através da publicação de uma coleção de notas, redescobertas em 1953. Em 1980, o Departamento de Defesa dos EUA registrou a linguagem de programação ADA, em sua homenagem.

1.9. Máquina de Hollerith

Preocupado com a lentidão na análise dos dados obtidos do censo anual, cuja demora era de aproximadamente 10 anos, o jovem funcionário do *Unites States Census Office*, Herman Hellerith, idealizou uma máquina capaz de tabular tais

informações. Baseado na técnica de cartões perfurados de Jacquard, aplicou o mesmo conceito, mas para o armazenamento dos dados coletados no censo.

O sistema era capaz de registrar números como buracos nos cartões em locais específicos os quais eram classificados através de pinos. A Máquina de Hollerith ou Máquina Tabuladora era composta das seguintes unidades:

- Unidade de controle, que dirigiria a seqüência das operações de toda a máquina através de furos em cartões perfurados.
- Entrada de dados, que utilizava também cartões perfurados.
- Saída, que perfuraria os resultados em cartões para uso posterior como entrada, aumentando assim a memória interna com armazenamento externo, indefinidamente grande.
- Saída impressa utilizada na apresentação dos resultados finais, tais como tabelas matemáticas, a qual de uma linotipo automática acoplada ao sistema.

Com a utilização da Máquina de Hollerith fizeram muito sucesso no censo de 1890 nos EUA, conseguindo que o tempo de processamento dos dados do censo diminuísse de 8 para 3 anos. A tecnologia de cartões perfurados foi adotada rapidamente por diversos países da Europa, difundindo a utilização das máquinas Hollerith a nível mundial e por bastante tempo.

Posteriormente, Hollerith fundou a bem sucedida companhia *Tabulating Machine Company*. Em 1924, esta firma mudou de nome, tornando-se a *International Business Machines Corporation*, hoje mais conhecida como IBM. No início, as vendas da IBM eram baseadas na linha de equipamentos de escritório e, em particular, máquinas tabulares.

Com o desenvolvimento dos primeiros dispositivos mecânicos para cálculo automático, começa efetivamente a vertente tecnológica que levará à construção dos primeiros computadores.

Para refletir:

Quando chegou o primeiro computador no Brasil? Quais eram suas características?

2. As máquinas de primeira geração (1930-1958)

O automatismo completo foi alcançado no século XX, quando um grande número de projetos foi implementado. Embora a concepção da máquina de Babbage foi o alicerce dos computadores modernos, a principal diferença era a utilização de circuitos eletrônicos (relés² e válvulas³) em lugar de mecânicos (rodas e engrenagens).

² Relé: eletroímã cuja função é abrir ou fechar contatos elétricos com o intuito de interromper ou estabelecer circuitos.

³ Válvula: dispositivo que conduz a corrente elétrica num só sentido

Este novo marco na evolução dos computadores determina o surgimento dos computadores chamados de Primeira Geração.

A principal vantagem das máquinas a relé sobre as máquinas mecânicas era, sem dúvida, a maior velocidade de processamento (milissegundos). Ainda, um outro aspecto positivo era a possibilidade de funcionamento contínuo, apresentando uma menor taxa de erros de cálculo.

Os computadores da primeira geração contavam com dispositivos de entrada/saída primitivos. A grande utilidade dessas máquinas era no processamento de dados, os quais eram ingressados e armazenados com base na utilização de cartões perfurados.

Por outro lado existia uma série de desvantagens como: custo elevado, relativa lentidão, pouca confiabilidade, grande quantidade de energia consumida e necessitavam de grandes instalações de ar condicionado para dissipar o calor gerado por um grande número de válvulas (cerca de 20 mil). Os computadores mais representativos desta geração foram: o MARK I, ABC (*Atanasoff Berry Computer*), ENIAC (*Electronic Numeric Integrator and Calculator*).

Juntamente com o ENIAC, ocorreu também o desenvolvimento na área de periféricos de computador com o aparecimento de equipamentos tais como as unidades de fita magnética, impressoras, etc.

2.1. Arquitetura Von Neuman

Uma contribuição importantíssima desta época foi o conceito de programa armazenado, introduzida pelo matemático húngaro, naturalizado americano John Von Neuman por volta de 1940.

A partir de sua experiência como consultor no projeto ENIAC, von Neuman conhecia os problemas da programação destas máquinas em relação à introdução dos programas. Os programas para os computadores da época eram construídos com base em modificações nos circuitos. Esta técnica era muito trabalhosa e demorada.

A solução proposta por von Neuman foi a utilização de cartões perfurados para o ingresso do programa, analogamente a como era feito com os dados. Para sustentar esta abordagem, John von Neuman assim desenvolveu a lógica dos circuitos, os conceitos de programa e operações com números binários.

Nesse modelo, um elemento Processador segue as instruções de programas armazenados em uma Memória de Programas, para ler canais de Entrada, enviar comandos sobre canais de Saída e alterar as informações contidas em uma Memória de Dados.

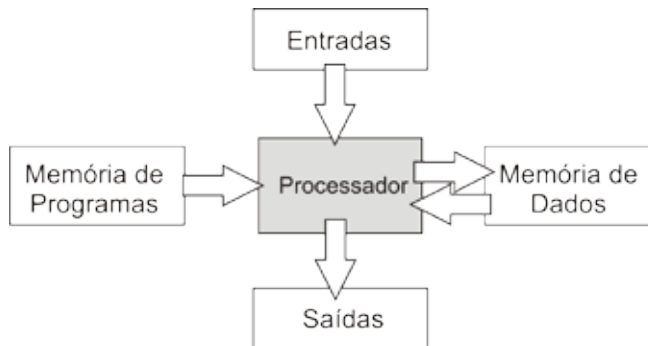


Figura 3 - Modelo proposto por Von Neuman

Estes conceitos, adotados nos computadores atuais, revolucionou o conceito de programação de computadores da época, tornando os programas muito mais flexíveis e versáteis.

A arquitetura de von Neuman inicial evoluiu para uma estrutura em barramento, que é a base dos computadores modernos. Nessa estrutura, as memórias de dados e de programa são fundidas em uma memória única facilitando a sua manipulação, e as comunicações entre elementos são efetuadas através de uma via comum de alta velocidade:

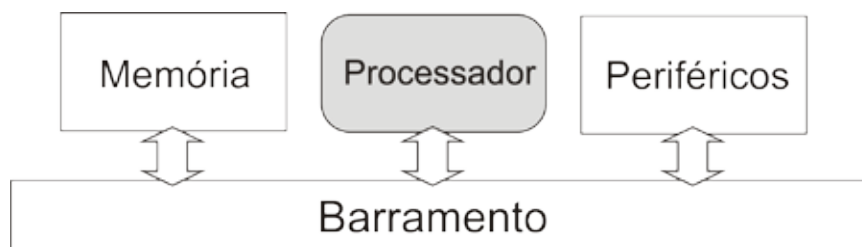


Figura 4 - Evolução do modelo de Von Neuman

O novo conceito de programação introduzido por Von Neuman deu origem a muitos outros projetos nos quais ele próprio esteve envolvido, como por exemplo, o EDVAC (*Electronic Discrete Variable Automatic Computer*), o IBM 650 (o primeiro computador da IBM), e o UNIVAC (Universal Automatic Computer), que foi o primeiro computador a ser fabricado em linha.

3. Computadores de segunda geração (1955-1965)

O novo marco que caracteriza a segunda geração de computadores é a invenção do transistor. Considerada uma das maiores descobertas da história moderna, o transistor é responsável pela revolução eletrônica da década de 1960. Esta nova tecnologia veio substituir a utilização de válvulas: em 1956 já se produziam computadores com esta tecnologia fabricados pela IBM e logo

depois a DEC (*Digital Equipment Corporation*), tornando estas empresas líderes mundiais na indústria de computadores. Com os transistores, os computadores passaram a ocupar um menor espaço físico, trabalhando a temperatura mais baixa e consumindo menos energia elétrica em relação às válvulas.

Com a segunda geração apareceram as memórias com anéis ferromagnéticos, evoluindo para as fitas magnéticas, as que se tornaram a forma dominante de armazenamento secundário. Algumas vantagens das fitas magnéticas sobre as perfuradas é que possuíam capacidade muito maior de armazenamento e o ingresso dos dados mais rápido. Posteriormente surgiram os discos magnéticos (1962), o que permitiu o acesso direto à arquivos muito grandes.

Também nesse período houve avanços no que se refere às unidades de memória principal, como por exemplo, a substituição do sistema de tubos de raios catódicos pelo de núcleos magnéticos, utilizado até hoje nos *chips* de memória RAM. Os dispositivos de memória auxiliar introduzidos na primeira geração continuam a ser utilizados.

Apareceram também os modernos dispositivos periféricos, tais como as impressoras, as fitas magnéticas, os discos para armazenamento, etc. Os computadores passaram a ter um desenvolvimento rápido, impulsionados principalmente por dois fatores essenciais: os sistemas operacionais e as linguagens de programação.

Exemplos de computadores dessa geração são o IBM 1401 e o Honeywell 800. O primeiro minicomputador foi o PDP-1, criado em 1959 e instalado em 1961. O primeiro produzido comercialmente foi o PDP-5. Um dos computadores mais comercializados nesta época foi o IBM 7090.

Para refletir

Pesquise sobre os computadores que foram os principais representantes desta geração e descreva suas características.

4. Computadores de terceira geração (1965-1980)

A substituição dos transistores pela tecnologia dos circuitos integrados de silício caracterizou os computadores da terceira geração. Os circuitos integrados foram criados por Robert Noyce, que foi o inventor do microchip e co-fundador da *Intel Corporation*. A tecnologia entrou no mercado em 1961 pela *Fairchild Semiconductor* e pela *Texas Instruments*, localizadas no Vale do Silício na região de Palo Alto e Stanford, na Califórnia.

A tecnologia dos circuitos integrados permitiu que dezenas de transistores fossem colocados em um único chip. Desta forma, surgiram computa-

dores de menores dimensões, mais rápidos (nanossegundos), confiáveis e baratos do que as máquinas das gerações anteriores.

A tecnologia utilizada na época era a de pequena escala de integração (SSI – *Small Scale of Integration*) integrando aproximadamente mil transistores no circuito de uma pastilha.

Dentre os computadores mais representativos dessa geração temos a linha *System/360* desenvolvido pela IBM, empresa líder da área de computadores da época (1965). O *System/360* foi projetado pra computação científica e comercial e apresentava muitas inovações:

- Família de máquinas com a mesma linguagem de montagem com tamanho e capacidades crescentes, projetada para propiciar compatibilidade entre um grande número de modelos.
- Multiprogramação: vários programas simultaneamente em memória, o que resulta em um maior aproveitamento da CPU.
- Primeira máquina capaz de emular (simular) outros computadores. Com isso, podiam ser executados programas antigos sem necessidade de modificar ou traduzir.
- Maior espaço de endereçamento de memória.

5. Computadores de quarta geração (1980 - ...)

Esta geração é marcada por uma drástica redução do tamanho dos computadores, porém com desempenho muito melhor: mais velozes e poderosos. Este avanço foi possível graças à tecnologia de circuitos integrados, possibilitando a integração de milhares de componentes eletrônicos em um pequeno espaço ou chip, dando origem aos microprocessadores.

Esta evolução teve início na década de 70, com a tecnologia de grande escala ou LSI (*Large Scale Integration*), que integrava até 65 mil componentes em um único chip. Nos anos 80 a técnica foi aperfeiçoada, atingindo a faixa de milhares de transistores numa pastilha VLSI (*Very Large Scale Integration*).

A redução de tamanho e custo desta nova geração de computadores possibilitou a sua aquisição para uso pessoal, privilégio que antes era somente reservado para organizações que poderiam arcar com grandes custos e amplos locais para a acomodação. A partir da abertura de um novo mercado de uso doméstico para os computadores, surgiram novas necessidades e desafios para as companhias. De forma geral, os computadores pessoais eram utilizados para processamento de texto, manipulação de planilhas e aplicações interativas como jogos.

No início dessa geração nasceu a empresa americana Intel, que começou a desenvolver o primeiro microprocessador, o Intel 4004 que manipulava palavras binárias de 4 bits, um circuito integrado com 2250 transistores, equivalente ao ENIAC.

O 4004 foi seguido pelo Intel 8008 de 8 bits e, mais tarde, pelo Intel 8080. O primeiro microcomputador da história foi o Altair 8800, que usava o chip Intel 8088, tornou-se padrão mundial da época para os microcomputadores de uso pessoal, abrindo uma nova era na história da informática.

Stephen Wozniak e Steve Jobs formaram em 1976 uma pequena empresa, a Apple, onde construíram, segundo conta a lenda, numa garagem de fundo de quintal, o Apple I. Um ano depois, com um novo e melhor projeto, surge o Apple II, primeiro microcomputador com grande sucesso comercial seguido, mais tarde, pelo Apple III.

A IBM decidiu entrar no mercado de computadores pessoais, introduzindo o PC (IBM *Personal Computer*), um microcomputador com tecnologia de 16 bits, estabelecendo o Intel 8088 como sua CPU. Lançado em 1981, o PC se tornou campeão de vendas da história dos computadores. Um PC tem a seguinte configuração típica – 256 a 640 K de memória RAM na placa principal, duas unidades de disco flexível de 360 K, controlador de vídeo gráfico, monitor monocromático e interface serial ou paralela para a impressora. Seu clock era de 4,77 MHz.

Com o sucesso impulsionado pela IBM PC, a Intel continuou fazendo versões cada vez mais poderosas. Os principais modelos de PC são: PC-XT, PC-XT 286, PC-AT386, PC 486.

O IBM PC vinha equipada com o sistema operacional MS-DOS, fornecido pela novata *Microsoft Corporation*.

Em 1993 chegou ao mercado o *Pentium*, cuja versão *Pentium III* possui cerca de nove milhões de transistores. O *Pentium* trouxe um novo fôlego às chamadas estações de trabalho. Uma característica comum às máquinas das gerações até então, é a existência de uma única CPU para executar o processamento.

Uma das novidades que o *Pentium* trouxe é que possibilita a simulação de dois processadores, ou seja, um princípio de paralelização antes possível apenas em supercomputadores e que agora está ao alcance dos usuários de microcomputadores. Mais recentemente, já existem computadores funcionando com mais de uma CPU.

Os atuais avanços em pesquisa e o projeto de novas tecnologias para os computadores estão possibilitando o surgimento da quinta geração. Dois avanços que configuram um divisor de águas são o processamento paralelo, que quebrou o paradigma de von Neumann, e a tecnologia dos supercondutores.

Atividades de avaliação



1. Pesquise sobre os computadores que foram os principais representantes desta geração e descreva suas características.

Tecnologias de computadores e suas aplicações

Na tecnologia atual, podemos classificar os computadores de acordo com a suas capacidades e finalidades de uso para o qual foram desenvolvidos. Possivelmente os computadores pessoais sejam os mais amplamente utilizados e conhecidos, no entanto existem outras variantes que serão analisadas nesta seção.

A quarta geração de computadores cujo domínio se estende até nossos dias, é impulsionada principalmente pela capacidade dos engenheiros e fabricantes de chips em empacotar cada vez mais transistores por pastilha, a cada ano. A cada incremento, são gerados computadores com memórias maiores e processadores mais poderosos.

A evolução tecnológica observada nas últimas décadas comprova que o número de transistores dobra a cada 18 meses, e a previsão indica que essa taxa será mantida no futuro. Esta constatação é conhecida como Lei de Moore, que dá fundamento ao chamado círculo virtuoso onde:

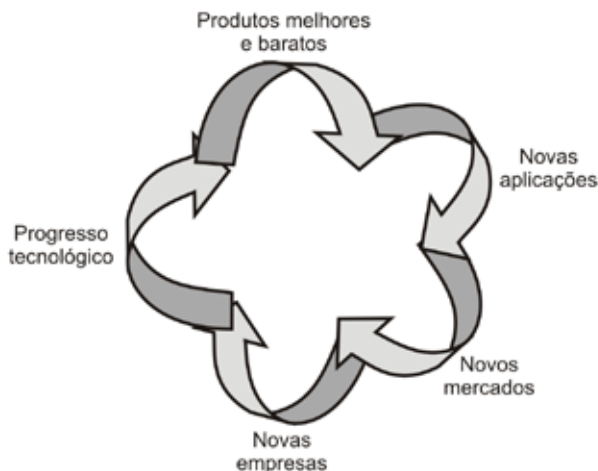


Figura 5 - Representação do círculo virtuoso relacionado ao desenvolvimento de computadores

Os benefícios alcançados pela lei de Moore podem ser mapeados para duas atitudes possíveis. Por um lado construir computadores cada vez mais poderosos mantendo o preço constante. Por outro lado, uma postura mais conservadora seria continuar a desenvolver o mesmo computador por um preço cada vez mais baixo. A indústria de computadores adotou as duas posturas, o que resultou na ampla variedade de computadores que existem atualmente e que podem ser utilizados para uma grande variedade de aplicações.

1. Computador descartável

Quem não recebeu alguma vez um cartão de congratulações musical? Ou deu de presente para a mãe? Pois é, esta tal vez seja a aplicação mais simples de computador descartável. Trata-se de um único chip sonoro descartável, porem com capacidade computacional maior do que o maior computador do mundo de 1955.

No entanto, utilidades mais interessantes sugeriram a partir do desenvolvimento do chip RFID (*Radio Frequency IDentification* ou identificação por radiofrequencia). Estes chips de baixíssimo custo (centavos), não possuem bateria e a sua espessura é de menos de 0,5 mm., possuem um minúsculo radiotransponder e um único numero de 128 bits embutido.

Esta tecnologia é utilizada para a identificação automática através de sinais de rádio, recuperando e armazenando dados remotamente através de dispositivos chamados de *tags* RFID: quando pulsados por uma antena transmissora externa, são alimentados pelo sinal de radio de entrada por tempo suficiente para transmitir seu numero de volta à antena.

Uma tag pode ser colocado em uma pessoa, animal, equipamento, embalagem ou produto, dentre outros. Além das tags passivas, que respondem ao sinal enviado pela base transmissora, existem ainda as tags ativas, dotadas de bateria, que lhes permite enviar o próprio sinal. São bem mais caras que as tags passivas.

A principal funcionalidade dos tags RFID é a identificação de todo tipo de objeto, inclusive seres vivos. No caso de objetos, estes tags viriam substituir os populares códigos de barras.

- Implantes em animais. Este tipo de sistema usado na identificação dos animais ajuda no gerenciamento dos mesmos entre as companhias, no controle de epidemias e garantia de qualidade e procedência. A identificação animal por sistemas de RFID pode ser feita de quatro maneiras diferentes: colares, brincos, injetáveis ou ingeríveis (bolus).

- Implantes humanos. Implantes de chips RFID usados em animais agora estão sendo usados em humanos também. Em 1998, o professor britânico de cibernética Kevin Warwick implantou um chip no seu braço. A empresa Applied Digital Solutions propôs seus chips “formato único para debaixo da pele” como uma solução para identificar fraude, segurança em acesso a determinados locais, computadores, banco de dados de medicamento, iniciativas anti-sequestro, identificação e monitoração de pacientes, entre outros.
- Hoje em dia, a maioria dos sistemas que gerencia recipiente é baseada em código de barras, porém no meio industrial o uso deste tipo de sistema não é confiável o suficiente, e os transponders de um sistema RFID pode guardar mais informações úteis posteriormente,
- No setor industrial os sistemas de RFID têm várias aplicações. Uma delas é na identificação de ferramentas, que no caso de grandes indústrias facilita o processo tanto de manutenção, quanto de substituição e administração das mesmas. Mas outro campo que sistemas RFID podem tanto melhorar a rapidez e qualidade do serviço, como também ter um papel de segurança nas indústrias é na identificação de recipientes, embalagens e garrafas, principalmente em produtos químicos e gases, onde um erro na hora de embalar pode causar sérios danos. Em algumas cidades do mundo, já existem sistemas RFID para controle de lixo, onde as tags são colocadas nas latas de lixo e os caminhões de coleta tendo leitores específicos.

A tecnologia usada em RFID se desenvolve rapidamente. Os menores são passivos (sem alimentação interna) e podem apenas transmitir seus números exclusivos quando consultados. Os maiores, são ativos, podem conter uma pequena bateria e um computador primitivo, sendo capazes de fazer alguns cálculos. Nesta classificação entram os Smart Cards usados em transações financeiras. (www.rfid.org).

Para refletir

Descreva outras aplicações possíveis para dispositivos RFID.

2. Microcontrolador

A grande maioria de aparelhos utilizados diariamente, inclusive para uso doméstico, possuem computadores embutidos ou microcontroladores. Estes dispositivos são utilizados para gerenciar as funcionalidades do aparelho e manipular a sua interface com o usuário. Podemos encontrar microcontroladores em uma grande variedade de aparelhos, por exemplo:

- Eletrodomésticos
- Aparelhos de comunicação
- Periféricos de computadores
- Equipamento de entretenimento
- Aparelhos de reprodução de imagens
- Equipamentos médicos
- Armamentos militares
- Dispositivos de vendas
- Brinquedos

Um carro moderno, por exemplo, pode conter até 50 microcomputadores, que executam diferentes atividades tais como injeção, rádio, GPS, etc. A previsão é que dentro de poucos anos, quase tudo o que depende de energia elétrica ou baterias para funcionar, possua um microcontrolador.

Um microcontrolador (também denominado MCU ou μC) é um pequeno computador-num-chip, mas completos. Cada microprocessador contém um processador, memória e funções de entrada/saída. Dentre as funções de entrada/saída temos: controle de botões do aparelho, controle de luzes e sons, etc. Os microcontroladores podem ser de propósito geral ou de propósito específico.

A tecnologia de microcontroladores é regida principalmente pela variável de custo, de forma a tornar viável economicamente o controle eletrônico de muitos processos. Microcontroladores geralmente são encomendados por milhões de unidades. Neste caso, um centavo a mais ou a menos pode determinar a escolha por um determinado fornecedor.

Desta forma, a escolha por uma arquitetura é decidida com base principalmente no custo de fabricação. Outros fatores que influenciam o custo dos microcontroladores é a quantidade de bits, quantidade de memória e tipo. Por outro lado, o aumento na capacidade de processamento é sempre necessário para conseguir atender às necessidades dos produtos de tecnologia mais avançada.

Outra característica dos microcontroladores é o funcionamento em tempo real, ou seja, que devem ser capazes de detectar um evento e dar uma resposta imediata.

Finalmente, uma característica inerente de sistemas embutidos (como é o caso de microcontroladores) é a restrição física em termos de tamanho, peso, consumo, etc. Estas restrições podem ser contornadas colocando os

dispositivos para “hibernar” enquanto nenhum evento interessante for percebido pelos periféricos, reduzindo o consumo de energia diminui. As duas últimas características fazem com que a arquitetura destes dispositivos precise ser projetada cuidadosamente.

Microcontroladores são usados com frequência em produtos e dispositivos de controle automático, tais como sistemas de controle em motores de automóveis, controles remotos, máquinas de escritório, ferramentas e brinquedos. Ao reduzir o tamanho, custo e consumo de energia em comparação com um projeto que usa microprocessador, memória e dispositivos de E/S separados, microcontroladores tornam viável economicamente o controle eletrônico de muitos processos.

Cerca de 50% dos microcontroladores vendidos são controladores “simples”, outros 20% são processadores de sinais digitais (DSPs) mais especializados. Os microcontroladores podem ser encontrados em praticamente todos os dispositivos eletrônicos digitais que nos cercam: Teclado do computador, dentro do monitor, Harddisk, relógio de pulso, rádio relógio, máquinas de lavar, forno de microondas, telefones, etc. Você está certamente cercado de dezenas deles agora.

Certamente eles foram tão ou mais importantes para a revolução dos produtos eletrônicos que os computadores. Eles permitiram a evolução de equipamentos que a anos não evoluíam, como os motores a combustão, que agora com o novo controle eletrônico podem funcionar com sistema bi-combustível e poluindo menos e as máquinas fotográficas, que migraram de processos químico/mecânico a circuitos com microcontroladores+Sensores Digitais+Memória.

3. Computador de jogos

Os computadores de jogos são computadores normais, com recursos gráficos especiais e capacidade de som, mas *software* limitado e pouca capacidade de extensão. Inicialmente projetados para jogos simples, eram utilizados CPU de baixo valor. Com o passar dos anos evoluíram se tornando poderosos e capazes de competir em desempenho com computadores pessoais.

Os produtos de tecnologia vieram se tornar populares conhecidos genericamente como videogame. O primeiro foi lançado em 3 de dezembro de 1994 no Japão e quase um ano depois nos Estados Unidos, pela Sony. O *PlayStation* foi um sucesso de vendas, chegando a 100 milhões de unidades até 2006 quando sua produção foi encerrada. Posteriormente a tecnologia evoluiu para o *PlayStation 2* e *3*. O *PlayStation* consistia em um console independente de jogos composta de uma unidade leitora de CD-ROM e ranhuras

para *memory cards* (cartões de memória que armazenam informações dos jogos) e entradas para joysticks.

4. Computador pessoal (PC)

Nesta categoria podemos considerar os computadores de mesa que hoje em dia são utilizados em muitos lares como ferramenta para as mais diversas atividades, e os notebooks.

O coração de um computador pessoal está na placa de circuito impresso, por exemplo, da Intel. Geralmente na placa é contida a CPU, memória e os vários dispositivos de E/S (som, modem), interfaces para os periféricos (teclado, mouse, disco), e encaixes de expansão.

Um *notebook* pode ser considerado um PC, pois segue a mesma especificação de *software* e *hardware*, com a diferença de serem projetados para serem portáteis.

Ambos os computadores possuem sistemas operacionais elaborados e uma grande variedade de *software* disponível.

5. Servidor

Com o surgimento das redes de computadores e *clusters* de máquinas, e com o posterior aumento na complexidade de tais estruturas de comunicação e compartilhamento, aparece a figura do servidor de rede. O servidor de rede é um computador dedicado responsável por prestar algum serviço à rede de computadores (*clientes*⁴). Os serviços prestados podem ser diversos: arquivos, correio eletrônico, impressão, etc. Foi o crescimento e popularização da internet que impulsionou o desenvolvimento e aperfeiçoamento de tecnologias para servidores.

O servidor de rede (Intranet ou Internet) é um computador geralmente reforçado cujo diferencial é a capacidade de trabalho em rede a alta velocidade. De uma forma geral, os computadores servidores são mais rápidos, possuem mais espaço em disco e possuem uma conexão de rede mais rápida, principalmente se são dedicados. Os servidores executam os mesmos sistemas operacionais do que PC. Os servidores podem ser projetados de acordo a uma configuração com um único processador ou com múltiplos processadores.

Dentre os diversos tipos de servidores, os mais conhecidos são:

- **Servidor de fax:** servidor para transmissão e recepção automatizada de fax pela Internet, e através de todas as estações da rede.
- **Servidor de arquivos:** servidor que armazena arquivos de diversos usuários.

⁴ Cliente-servidor é um modelo computacional que separa clientes e servidores, sendo interligados entre si geralmente utilizando-se uma rede de computadores. Cada instância de um cliente pode enviar requisições de dado para algum dos servidores conectados e esperar pela resposta. Por sua vez, algum dos servidores disponíveis pode aceitar tais requisições, processá-las e retornar o resultado para o cliente. Apesar do conceito ser aplicado em diversos usos e aplicações, a arquitetura é praticamente a mesma.

- **Servidor web:** servidor responsável pelo armazenamento de páginas de um determinado site, requisitados pelos clientes através de browsers.
- **Servidor de e-mail:** servidor responsável pelo armazenamento, envio e recebimento de mensagens de correio eletrônico.
- **Servidor de impressão:** servidor responsável por controlar pedidos de impressão de arquivos dos diversos clientes.

Um computador pode atuar em mais de um tipo diferente de servidor. Computadores que atuem como um único tipo de servidor é chamado de servidor dedicado. Os servidores dedicados possuem a vantagem de atender a uma requisição de um cliente mais rapidamente.

Servidores dedicados, que possuem uma alta requisição de dados por partes dos clientes e que atuam em aplicações críticas utilizam *hardware* específico para servidores. Já servidores que não possuam essas atuações podem utilizar *hardware* de um computador comum, não necessitando ser um supercomputador.

Devido a operar com muitas entradas e saídas de informações, os servidores necessitam de processadores de alta velocidade, algumas vezes alguns servidores são multi-processados, ou seja, possuem mais de um processador.

Por ter de operar por muito tempo (as vezes de maneira ininterrupta), alguns servidores são ligados a geradores elétricos. Outros utilizam sistemas de alimentação (por exemplo, o UPS) que continuam a alimentar o servidor caso haja alguma queda de tensão.

6. Conjunto de estações de trabalho

Considerando a boa relação custo/desempenho dos computadores pessoais, nos últimos anos tem se tornado uma prática comum a interconexão por rede de gigabytes, de um grande número de PCs para formar *Clusters of Workstations* (COWs).

Através da execução de *software* especial, todas as máquinas interconectadas podem trabalhar juntas num mesmo problema, atingindo um maior poder de processamento e alta velocidade desde que o processamento pode ser realizado em paralelo.

É este poder de processamento a um custo mais baixo, que começou a tornar a utilização de supercomputadores e *mainframes* desnecessária levando esta categoria de computadores caminho à extinção. A quantidade de máquinas pode variar de forma indefinida, limitada pelos recursos disponíveis.

7. Mainframe - Supercomputadores

Os *mainframes* são computadores de grande porte físico, desde que ocupam uma sala inteira, da mesma forma que os seus antecessores dos anos 60. A sua principal característica é a sua capacidade de E/S, equipadas com coleções de discos para armazenamento de dados.

Esta classe de computadores geralmente hospeda *software* legado há algumas décadas, desenvolvidos especialmente para esses sistemas. É por conta do grande risco em reprogramar e migrar essas aplicações para máquinas menores, que faz com que muitas organizações continuem a manter por milhões de dólares esses sistemas, em lugar de comprar um novo.

A história dos supercomputadores começa, de fato, no final de 1975 com o Cray-1. As aplicações para esses computadores são muito especiais e incluem laboratórios e centros de pesquisa aeroespaciais, empresas de altíssima tecnologia, previsão do tempo e a produção de efeitos e imagens computadorizadas de alta qualidade. Os supercomputadores são os mais poderosos, mais rápidos e de maior custo. Utilizam o conceito de processamento paralelo e são máquinas vetoriais, isto é, podem executar a mesma operação em diversas variáveis simultaneamente.

Atividades de avaliação



1. Pesquise quais as características e responsabilidades do Servidor de banco de dados e Servidor FTP.
2. Pesquise mais algum outro tipo de servidor que não tinha sido mencionado.

Introdução à ciência da computação

Ciência da Computação é o estudo dos algoritmos e suas aplicações, bem como das estruturas matemáticas indispensáveis à formulação precisa dos conceitos fundamentais da teoria da computabilidade e da computação aplicada.

Os estudos oriundos da Ciência da Computação podem ser aplicados em qualquer área do conhecimento humano em que seja possível definir métodos de resolução de problemas baseados em repetições previamente observadas. Desempenha por isso um papel importante na área de ciência da computação a formalização matemática de algoritmos, como forma de representar problemas decidíveis, i.e., os que são susceptíveis de redução a operações elementares básicas, capazes de serem reproduzidas através de um dispositivo mecânico/eletrônico capaz de armazenar e manipular dados.

Um destes dispositivos é o computador digital, de uso generalizado, nos dias de hoje, pelo custo reduzido dos componentes eletrônicos que formam o seu *hardware*.

A Ciência da Computação constitui uma base científica para diversos tópicos voltados à programação de computadores, processamento da informação e estudo de soluções algorítmicas propriamente dito.

Um dos conceitos fundamentais da Ciência da Computação é o algoritmo. Como citado nas primeiras seções desta unidade, um algoritmo determina um conjunto ordenado e não-ambíguo de passos que definem uma atividade finita. De forma geral, esta atividade finita visa a resolução de um problema.

Os algoritmos, na forma de programas, são executados em computadores. Para que um algoritmo seja executado no computador precisa ser representado de forma apropriada para que possa ser interpretado por uma máquina ou por um ser humano de forma correta e sem ambiguidades.

Estudos baseados em lingüística e gramática se preocupam pela tradução do algoritmo conceitual em um conjunto de comandos com uma sintaxe

⁵ Pseudocódigos: Sistema notacional nos quais idéias podem ser informalmente expressas durante o desenvolvimento de um algoritmo.

e semântica bem definidos através representação fornecida pelas linguagens de programação.

As linguagens de programação possibilitam escrever algoritmos num formato de fácil tradução para a linguagem da máquina, mas mantendo a familiaridade com a linguagem de **pseudocódigo**⁵ utilizada na descrição do algoritmo.

O desenvolvimento histórico das linguagens de programação modernas visa, a cada nova geração, elevar o nível de abstração de forma a propiciar cada vez mais independência da máquina. Nos primeiros dispositivos computacionais, os programas eram codificados no *hardware*, no interior da unidade de controle. Esta abordagem torna estes dispositivos muito inflexíveis, dada a dificuldade de alterar os programas.

Neste contexto surge o conceito de programa armazenado na memória principal, da mesma forma que os dados. Para isso, os programadores começaram a utilizar linguagem de máquina na escrita dos seus algoritmos. Este método era muito laborioso e complexo, uma vez que os códigos de operação e operandos eram representados através de dígitos numéricos, o que tornava a programação muito propensa a erros. O primeiro passo para melhorar a situação foi a utilização de nomes descritivos para as posições de memória e mnemônicos (em lugar de códigos hexadecimais), para expressar os códigos de operação.

A partir da utilização desta técnica, surgiu a necessidade de tradução dos programas escritos nesta notação para a linguagem de máquina. Surgiram assim as linguagens de montagem. Fica assim como responsabilidade do sistema operacional, a utilização da linguagem de montagem para traduzir o programa e, posteriormente, armazenando a versão traduzida para um uso posterior.

No entanto, o problema da dependência com a máquina continuava existindo, agora em relação à linguagem de montagem. Além disso, existia a dificuldade de programar em um nível muito baixo de abstração (ou seja muito detalhado), o que prejudicava o desenvolvimento do projeto da solução.

A necessidade de primitivas de alto nível propiciou o surgimento das linguagens de terceira geração, para o desenvolvimento de *software*, as quais por sua vez, foram projetadas com base nas primitivas de baixo nível já conhecidas. Por exemplo:

```
assign Total the value Valor + Imposto
```

Tendo como potencial primitiva:

```
assign <identificador> the value <expressão>
```

O desenvolvimento das linguagens de terceira geração conseguiu atender o objetivo de independência de plataforma de uma forma muito mais próxima ao raciocínio utilizado na análise e resolução de problemas.

Na medida em que um maior nível de abstração se torna uma necessidade surgem os chamados ambientes de programação, os quais possibilitam a utilização de conceitos mais abstratos na programação dos algoritmos.

A perspectiva nestes ambientes é que a máquina se adapta ao raciocínio humano, em contrapartida das linguagens de gerações anteriores, onde o ser humano precisava se adaptar à máquina. Uma propriedade inerente destes ambientes é a facilidade de utilização e a possibilidade de customização, sem auxílio de especialistas.

De forma geral, a programação nestes ambientes é facilitada pela utilização de menus e seleção de ícones na tela os quais representam atalhos de alto nível para a execução de instruções de nível mais baixo. Estes pacotes ou ambientes de programação são denominados de linguagens de quarta geração.

Finalmente, as linguagens classificadas como de quinta geração, tem como base principal a programação declarativa, e geralmente possuem um enfoque mais especializado (programação lógica).

Saiba Mais



Um Sistema de Informação (SI), coleta, processa, armazena, analisa e dissemina informações com o propósito específico. Como qualquer outro sistema, um sistema de informação abrange entradas (dados) saídas (relatórios, cálculos), processa essas entradas e saídas e gera saídas que são enviadas para o usuário ou outros sistemas. É possível incluir um mecanismo de resposta feedback – que controle a operação. E como qualquer outro sistema, um sistema de informação opera dentro de um ambiente.

Atividades de avaliação



1. O que significa dizer que uma linguagem de programação é independente de máquina?
2. O que significa que uma linguagem possui um nível mais alto que as gerações anteriores?
3. Escreva um algoritmo em pseudocódigo para realizar alguma atividade doméstica.

Tecnologia da Informação

Antes mesmo da popularização dos computadores, existia uma noção da importância do cuidado e tratamento da informação como uma forma de preservar o histórico, para que, eventualmente, possa ser utilizado na análise de fatores do passado para o entendimento do presente e, principalmente, na tomada de decisão de ações futuras. Deste este ponto de vista, a informação pode ser interpretada como um elemento que pode adicionar certo grau de competitividade às organizações.

Antigamente, os sistemas de informação nas organizações se baseavam basicamente em técnicas de arquivamento e recuperação de informações de grandes arquivos de forma manual, sob a responsabilidade de um *arquivador*. O processo era relativamente simples, porém muito trabalhoso, tornando difícil e propensa a erros a recuperação da informação e principalmente a atualização dos dados.

Com o avanço da tecnologia, no século XX a informação começou a fluir e se multiplicar a velocidade assustadora. Era o advento da Era da Informação. Um marco desta nova fase é a invenção do telégrafo elétrico em 1837, e a partir dali o desenvolvimento de meios de comunicação de massivos, e até mais recentemente, o surgimento da grande rede de comunicação de dados que é a Internet. Esta globalização propiciou um crescimento exponencial do volume de dados disponíveis, tornando um desafio a coleta, organização, catalogação e posterior recuperação de forma eficiente e segura da informação.

No ambiente complexo e turbulento que nos encontramos é essencial para as organizações o estímulo ao resgate do conhecimento organizacional como auxílio no processo de decisão e na busca de vantagens competitivas. Como consequência do processo de globalização, o contexto mundial, em permanente mudança, vive permanentemente submetido às implicações decorrentes de grandes transformações políticas, econômicas e tecnológicas. Neste cenário, a informação passa a ser considerada um recurso essencial nas tomadas de decisão. Para Beck (2007), “atualmente, toda empresa está envolta com amplos e diversos tipos de informação e, para competir neste

contexto dinâmico, o segredo do sucesso é a agregação de valor a partir do acesso, do tratamento, da utilização e da disseminação da informação”.

A partir desta premissa surge a necessidade nas empresas de gerenciar o conhecimento com base num novo modelo de tecnologia de informação e gestão de forma a propiciar a distribuição da informação com o intuito de atingir competitividade.

Um SI é um sistema tem seu foco na informação. Seu objetivo é armazenar, tratar e fornecer informações de tal modo a apoiar as funções ou processos de uma organização.

Geralmente, um SI é estruturado em dois sub-sistemas:

- Sub-sistema social que inclui as pessoas, processos, informações e documentos.
- Sub-sistema automatizado, que consiste dos meios automatizados (máquinas, computadores, redes de comunicação) que interligam os elementos do sub-sistema social.

O sub-sistema automatizado consiste em um conjunto de recursos tecnológicos e computacionais para geração e uso da informação de forma a viabilizar a [Gestão do Conhecimento](#)⁶ para a tomada de decisão e a elaboração de estratégias a partir de novos bancos de dados originados dos bancos de dados operacionais da empresa como, por exemplo, o *data warehouse*, o *data mining*, entre outros.

Essas ferramentas, além de dar subsídios para a tomada de decisão, estão voltadas à disseminação do conhecimento na organização, seja através de melhores práticas de trabalho, seja através do registro de discussões para a disseminação de conhecimento, como por exemplo o *Groupware* e a *Intranet*.

Esse conjunto de recursos é chamado de Tecnologia da Informação (TI). As aplicações de TI são diversas e estão ligadas às mais diversas áreas, de forma a automatizar os processos organizacionais. Através da TI, pequenas e grandes empresas podem alcançar maior produtividade e competitividade, alcançando o sucesso e alavancando maiores rendimentos.

A TI exerce um importante papel no âmbito da gestão do conhecimento reduzindo distâncias e propiciando a transferência do conhecimento de forma *on-line*. No domínio da gestão do conhecimento, o benefício principal da utilização de recursos computacionais é a sua capacidade de armazenamento e comunicação de computadores ligados em rede fazendo deles propiciadores e disseminadores de conhecimento (DAVENPORT e PRUSAK, 1998).

⁶ Gestão de Conhecimento: envolve as atividades de coleta, seleção, disponibilização e disseminação de informações.

A TI envolve a organização de recursos computacionais tais como *hardware*, *software* e mecanismos de comunicação, em um sistema capaz de executar um conjunto de tarefas. Além dos recursos computacionais, tecnologias relativas ao planejamento e desenvolvimento de sistemas, suporte ao *software*, e aos processos de produção e operação, também são considerados elementos de TI.

Por outro lado, tecnologias intelectuais tais como métodos, teorias e metodologias utilizadas para lidar com o ciclo da informação, também são consideradas sustento da TI. Neste conjunto podem ser consideradas, por exemplo, técnicas de classificação, que não necessariamente requerem uso de máquinas, mas apenas um esquema para serem aplicadas.

Técnicas que se mostraram eficientes geralmente são automatizadas pela elaboração de um algoritmo ou *software*, de forma a facilitar a sua aplicação sistemática. Seguindo o exemplo das tecnologias de classificação e organização de informações, a sua origem se remonta ao surgimento das primeiras bibliotecas. Hoje em dia praticamente toda biblioteca tem seu acervo gerenciado através de um sistema computacional.

O sucesso e difusão dos sistemas de informação alcançados através da rápida evolução e surgimento de novas Tecnologias de Informação fazem com que praticamente todas as atividades desenvolvidas na sociedade atual sejam sustentadas eficientemente através dos recursos da informática, e com custos cada vez mais reduzidos.

O processamento de informação é uma atividade central em diversas áreas, tais como economia, transportes, produção, imprensa, comunicação. A necessidade pela difusão social da informação de domínio público ou privado em larga escala de transmissão é possível somente através de sistemas tecnológicos inteligentes.

Uma consequência do desenvolvimento tecnológico é a intensificação da comunicação pessoal e institucional, através de programas de processamento de texto, de formação de bancos de dados, de editoração eletrônica, bem de tecnologias que permitem a transmissão de documentos, envio de mensagens e arquivos, assim como consultas a computadores remotos (via rede mundial de computadores, como a internet).

Em contrapartida aos enormes benefícios atingidos pela utilização da TI nos diversas atividades, alguns problemas podem ser apontados em relação à privacidade dos indivíduos e ao seu direito à informação, pois os cidadãos geralmente não tem acesso a grande quantidade de informação sobre eles, coletadas por instituições particulares ou públicas.

Embora a TI sustente os processos de Gestão do Conhecimento, a eficácia final destes processos depende da interpretação deste conhecimento pelos homens, e sua transformação em ações. Sendo assim, a tecnologia da informação, isoladamente, não pode ser a base da criação e gestão de conhecimento organizacional, mas o suporte dentro desse processo.

Síntese do capítulo



Neste capítulo foi apresentado o histórico da evolução dos computadores, a partir da descrição dos dispositivos mecânicos precursores dos computadores utilizados atualmente. Foi feita uma breve retrospectiva das gerações que determinam os estágios da evolução dos computadores até os dias de hoje.

Neste capítulo também foram apresentados os principais conceitos que definem a Ciência da Computação, apresentando brevemente as principais disciplinas que a compõem. Para concluir foi apresentada a importância da Tecnologia de informação (TI) no processo de construção e disseminação do conhecimento nas organizações, de forma a aumentar o alcance e a velocidade da transferência do conhecimento.

Leituras, filmes e sites



<http://www.cotianet.com.br/bit/hist/Default.htm>

Charles Babbage Institute <http://www.cbi.umn.edu/>

<http://www.daniel.prof.ufu.br/icc.html>

Referências



TANENBAUM, ANDREW S. **Organização Estruturada de Computadores**. 5. Edição. São Paulo: Editora Pearson. 2007.

STALLINGS WILLIAM. **Estrutura e Organização de Computadores**. 5. Edição. São Paulo: Editora Pearson. 2004.

BROOKSHEAR GLENN J. **Ciência da Computação – Uma visão abrangente**. 5. Edição. São Paulo: Editora Bookman. 2000.

Parte

2

Representação e processamento da informação

Representação e processamento da informação

Introdução

Neste capítulo serão apresentadas as formas de representação das informações nos computadores. Assim como o ser humano tem a fala e a escrita para representar as informações, o computador tem a sua própria representação para que possa tratá-las e processá-las. Iniciamos apresentando a unidade básica de representação de informação no computador, o *bit*, e mostramos também a representação de bytes (bloco de bits), de letras e números.

Em seguida apresentamos uma introdução ao sistema binário e a forma como os números são representados e mostramos como realizamos as operações matemáticas em um computador. Finalmente apresentamos uma introdução à lógica digital mostrando os tipos mais comuns de portas lógicas, que estabelecem a forma de como um computador pode processar as informações.

Para um computador qualquer informação deve ser codificada com uma linguagem apropriada para seu mundo, que é diferente do mundo dos humanos. Para o mundo eletrônico o ideal é que essa linguagem tenha dois estados, ligado e desligado, ou simplesmente, “0” e “1”. Por isso a base mais importante para o computador é a base 2, daí o nome binário.

Vamos começar com algumas definições sobre a representação de informações em um computador.

- **bit**: unidade básica de representação digital.
- **byte**: conjunto de 8 bits.
- **nybble**: um bloco de 4 bits.
- **palavra** ou **word**: conjunto de 32 bits
- **meia-palavra** ou **halfword**: um bloco de 16 bits
- **palavra-dupla** ou **doubleword**: conjunto de 64 bits

A definição de “palavra” não está claramente definida. Ela geralmente se refere ao número de bits usados simultaneamente em um computador.

Atualmente a maioria dos computadores usa uma palavra de 32 bits. No entanto, cada vez mais, arquiteturas de 64 bits estão sendo construídas o que muda o tamanho da palavra. Entretanto, por ser o mais comum, ao longo desse texto usaremos palavra com 32 bits.

1. Identificando os bits

É conveniente identificarmos cada bit individualmente para que possamos referenciá-los pelo nome. Daremos um rótulo para cada bit, da direita para a esquerda, sendo 0 (zero) o índice mais à direita do bloco.

Por exemplo, suponha que você tenha um número com n -bits. Poderíamos identificá-los como: b_{n-1} , b_{n-2} , ..., b_1 , b_0 .

Também é comum definir mais dois termos para um bloco com n -bits:

- **bit menos significativo** ou *least significant bit*. Bit b_0 . É comum abreviar como LSb com o 'b', em minúsculas para representar bit. (Quando estiver em letras maiúsculas, isso significará byte menos significativo).
- **bit mais significativo** ou *most significant bit*. Bit b_{n-1} . É comum abreviar como MSb com o 'b', em minúsculas para representar bit. (Quando estiver em letras maiúsculas, isso significará byte mais significativo).

Vale lembrar que o termo "menos significativo" não significa que ele seja "menos importante". Na verdade essa denominação está relacionada com o menor índice de sua posição. Todos temos ciência que alterar esse bit (ou seja, mudá-lo de 0 para 1, ou 1 para 0) iria afetar o valor global assim como qualquer outro *bit*. Da mesma forma, o *bit* "mais significativos" não significa "mais importante", mas apenas que seu índice é o maior do bloco.

2. Strings e vetores

Embora existam diferentes padrões para definir os tamanhos dos blocos de bits (*bits*, *bytes*, *nybbles*, etc), às vezes é necessário fazer referência a um conjunto de *bits* mais arbitrário. Iremos fazer referência a um conjunto de " N " bits como um *string*⁷ de bits ou um vetor.

Pode parecer estranho para se referir a um número como uma *string*, mas não é tão incomum. Quando escrevemos números, nós costumamos alterar o número de dígitos, conforme necessário. Assim, você escreve 10, em vez de 0010 porque os zeros adicionais não têm significância. Porém, em *hardware*, temos um número fixo de *bits* para a palavra como, por exemplo, 32 *bits* ou 64 *bits*. Nesse caso acrescentamos zeros a direita para preencher o número necessário para completar de número total de *bits*.

⁷ String: conjunto de bits de tamanho arbitrário que representam números ou palavras.

Então, o nome de *string de bits* pode ser bom, porque pode ser usado para qualquer comprimento de uma string. São igualmente *string de bits* os chamados *vetores de bit*. Neste caso, um vetor geralmente significa uma "coordenada" com n argumentos. Por exemplo, quando você trabalha com gráficos em 2 ou 3 dimensões, as coordenadas são escritas como (2,3) ou (2,3,1), então, 0010 poderia significar (0,0,1,0), ou seja, uma coordenada em quatro dimensões.

Claro que, quando as pessoas falam de vetores eles geralmente não pensam nisso como uma coordenada, mas sim em uma sequência de valores com posições numeradas. Em todo o caso, vamos chamá-lo simplesmente de *strings*.

2.1. Quantos diferentes *strings* podemos escrever?

Suponha que você tenha um *string* de n -bits. Quantas combinações possíveis de 1 e 0 existem? E por que isso interessa?

Para calcular o número de combinações de string de n -bits, percebem que temos 2 opções para cada bit. Simplesmente basta multiplicar 2, n vezes. Isto é, 2^n .

Este é um fato muito importante para se lembrar, pois o número de possibilidades de um string limita o número de possíveis valores que ele pode representar. Por exemplo, se você tem 3 bits, então você tem 8 possíveis valores para ser representado pelo *string* ($8 = 2^3$). Isso limita a um máximo de 8 valores diferentes.

3. Representação de dados

Imagine que você está em um teclado, no entanto, é um teclado horrível. Esse teclado tem apenas duas teclas para se usar: 'a' e 'b'. É impossível escrever uma mensagem para alguém usando apenas duas teclas, afinal, quantas palavras só usam as letras 'a' e 'b'? Não muitas.

No entanto, ainda é possível enviar mensagens. Como? Você pode codificar a mensagem utilizando um código, por exemplo, o código ASCII⁸. O código ASCII atribui um número a cada letra ou símbolo do teclado. Por exemplo, 'A' é codificado para o valor numérico 65 (na base 10) ou 0100 0001 (na base 2). Computadores codificam em binário, o que significa, que todas as letras podem ser codificados em 0's e 1's, mais simples para um computador entender. Se considerarmos 'a' como 0 e 'b' como 1, então você pode codificar a letra A como "abaa aab" ou "0100 0001" em binário.

⁸ ASCII: acrônimo de American Standard Code for Information Interchange, padrão de representação binária de letras e números.

Uma das coisas mais importantes para aprender sobre o *hardware* são as formas de como os dados são codificados em 0 e 1. Isto é chamado de representação de dados.

Além de teclas ou caracteres, ainda existem outras informações básicas que têm de ser codificadas: os números. Aparentemente parece trivial, basta converter o número decimal no seu equivalente binário, e aí, efetuar todas as operações. Para representar números inteiros positivos isso funciona, mas inteiros negativos, frações, números ponto flutuante?

Um dado pode ser representado nas seguintes formas:

- Inteiros sem sinal (inteiros positivos)
- Inteiros com sinal (inteiros positivos, negativos e zero)
- Números de ponto flutuante (aproximações de números reais positivos e negativos)
- Caracteres (ASCII, Unicode)

É importante lembrar que é impossível representar qualquer número real no computador porque há a limitação no tamanho do número representado, por exemplo, tamanho da mantissa e do expoente, obrigando a armazenar uma aproximação do valor. Esse “arredondamento” pode causar imprecisões nos cálculos realizados em um computador.

Podemos dizer que, de todas as coisas que você aprender sobre o computador, a representação de dados será provavelmente a mais importante. A maioria dos formandos em Computação provavelmente nunca irão projetar um *hardware*, mas todos deverão ter conhecimento de como os dados serão armazenados.

Atividades de avaliação



1. Pesquise outros tipos de representação de números e caracteres.

Sistemas de Numeração

Um sistema de numeração é um conjunto de caracteres e regras matemáticas usadas para representar um número.

Os sistemas numéricos nasceram com o início de nossa civilização. Alguns exemplos são o sistema numérico romano (numerais romanos), o sistema numérico chinês, o sistema numérico grego, e o sistema numérico árabe. Nossos números contemporâneos são baseados nesse último.

O sistema mais usado é a base 10 ou o sistema decimal, que usa uma base 10 para operações e é conseqüentemente muito simples se usar (que nós temos usado desde o pré-escolar!). Os outros sistemas são o sistema hexadecimal e o sistema binário. Estes sistemas usam a base 16 e a base 2, respectivamente. Os sistemas hexadecimais e binários serão discutidos a seguir.

Por que nós usamos a base 10? Olhe suas próprias duas mãos e conte seus dedos! A palavra “dígito” se origina de *digitus*, palavra latina que significa “dedo”. Nós todos aprendemos na escola que cada dígito em um número corresponde ao valor de seu lugar. Por exemplo, 245 seria $(2 \times 100) + (4 \times 10) + (5 \times 1) = 245$. Graficamente temos:

100	10	1
2	4	5

Como nós observamos, cada valor da coluna representa 10 vezes o valor da coluna à direita dele. Por isso chamamos base 10.

1. O sistema binário

O sistema binário⁹ tem uma base 2, significando que do valor de cada coluna é 2 vezes o valor da coluna à direita, começando com o dígito mais a direita, com o valor igual a 1. Usamos apenas os dígitos 0 e 1. Por exemplo, o número 111101101 representado graficamente temos:

⁹ Sistema binário: sistema numérico posicional com base 2.

256	128	64	32	16	8	4	2	1
1	1	1	1	0	1	1	0	1

Assim nós temos:

$$(1 \times 256) + (1 \times 128) + (1 \times 64) + (1 \times 32) + (0 \times 16) + (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) = 4096 + 256 + 128 + 64 + 32 + 8 + 4 + 1 = 493$$

Assim, 111101101 no sistema binário é igual a 493 no sistema decimal.

O sistema binário é útil nos computadores porque nos circuitos eletrônicos temos dois estados distintos, ligado ou desligado, representando 1 ou 0. Outra vantagem do sistema binário é porque o transistor (que executa a função de chave no circuito eletrônico) consome um mínimo de energia quando ele funciona nos estados de corte e saturação, respectivamente 0 ou 1. Isso permite circuitos mais densos e ocupando menos espaços, possibilitando dispositivos mais compactos, mais velozes e mais econômicos (usando menos pilha, por exemplo).

1.2. Convertendo binário-decimal-binário

O procedimento acima pode ser utilizado para converter um número binário em número decimal. Vamos mostrar o exemplo do número 1100110 que é igual a $64 + 32 + 0 + 0 + 4 + 2 + 0 = 102$. Graficamente temos:

512	256	128	64	32	16	8	4	2	1
0	0	0	1	1	0	0	1	1	0

Apresentamos o método para converter números de binário ao decimal. Como podemos fazer o inverso? Vamos mostrar a conversão do número 500 em binário. Utilizando nosso diagrama de potências de dois, devemos escrever os valores nos lugares da esquerda para a direita até completar o valor desejado.

Como 500 é menor que 512 o primeiro dígito será 0. Escrevemos então, 0 sob o valor 512.

512	256	128	64	32	16	8	4	2	1
0									

A partir de agora, o esquema é simples: testar o lugar imediatamente à direita. Se é menor ou igual ao número, escrever um 1 sob ele, subtrair o valor do número e usar o resultado na posição seguinte. Se é maior, escrever embaixo um 0, não subtrair nada e ir para a posição seguinte. Na tabela acima,

256 é menor que 500, assim escrevemos um 1 sob 256 e subtraímos 256 de 500. Agora vamos usar o resultado (244) na posição seguinte. Assim temos:

512	256	128	64	32	16	8	4	2	1
0	1								

128 é menor que 244, assim escrevemos um 1 sob 128 e subtraímos 128 de 244. Usamos o resultado (116) na posição seguinte.

512	256	128	64	32	16	8	4	2	1
0	1	1							

E assim por diante até zerar o número:

512	256	128	64	32	16	8	4	2	1
0	1	1	1	1	1	0	1	0	0

Simple não? Com esse algoritmo simples, podemos converter um número decimal em um número binário. 500 em decimal é igual a 111110100 em binário.

2. O sistema hexadecimal

O sistema hexadecimal¹⁰ têm uma base 16, significando que cada coluna é igual a 16 vezes a coluna a sua direita, começando com o valor 1 na coluna mais a direita. Com isso temos:

65536	4096	256	16	1
0	4	A	3	8

O sistema hexadecimal, além dos dígitos 1 a 9, usa as letras A, B, C, D, E, F; que representam 10, 11, 12, 13, 14 e 15, respectivamente. No exemplo acima, o número 04A32 vale $(4 \times 4096) + (10 \times 256) + (3 \times 16) + (8 \times 1) = 19000$.

Por que usar um número hexadecimal? É muito complicado para um humano trabalhar com números binários, pois são muito longos e fáceis de se confundir. A conversão de hexadecimal para binário é direta (16 é potência de 2), assim, cada 4 dígitos binários representam um dígito hexadecimal conforme mostramos abaixo:

¹⁰ Sistema hexadecimal: Sistema numérico com base 16.

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Veja como é fácil converter o valor hexadecimal acima em binário:

65536	4096	256	16	1
0	4	A	3	8
0000	0100	1010	0011	0100

O número hexadecimal 4A38 é igual a 0100101000110100 em binário.

2.1. Convertendo hexadecimal-decimal-hexadecimal

Vamos converter o número hexadecimal 7EF. Recorde que o dígito mais a direita é igual a um e cada dígito à esquerda é igual a dezesseis vezes o dígito imediatamente a sua direita. Colocando na tabela temos:

256	16	1
7	E	F

Multiplicamos agora cada dígito hexadecimal pelo número em seu cabeçalho (lembre-se que o A-F representa 10-15 respectivamente)

$$256 \times 7 + 16 \times 14 (E) + 1 \times 15 (F)$$

Totalizando o resultado, temos:

$$1792 + 224 + 15 = 2031$$

Assim chegamos ao valor final: 7EF em hexadecimal é igual a 2031 em decimal.

Converter do decimal para hexadecimal é um pouquinho mais complicado. Para este exemplo usaremos 293. Inicialmente escrevemos a tabela com os valores em decimal na ordem da direita para a esquerda. Cada dígito a esquerda deve ter o valor de 16 vezes maior que o da direita, começando com 1.

4096	256	16	1
0			

Colocamos um 0 sob o valor 4096 da tabela que é maior do que o número a ser convertido, 2031. Dividimos o número a ser convertido por 256, que é o valor mais elevado da tabela e menor do que o número a ser convertido. Neste caso, 293 dividido por 256 igual a 1, com um resto 37. Em seguida, escrevemos o resultado (1) e usamos o resto (37) para o cálculo seguinte.

4096	256	16	1
0	1		

Dividir agora o resto do último cálculo (37) pelo valor seguinte (16). 37 por 16 igual a 2 com um resto 5. Escrevemos o resultado (2) sob 16 e dividimos o resto (5) pelo valor seguinte (1)

4096	256	16	1
0	1	2	

5 dividido por 1 igual a 5 sem resto, assim colocamos o valor (5) sob 1 e está concluído. Assim:

4096	256	16	1
0	1	2	5

Então 293 decimal é igual a 125 em hexadecimal.

Se por acaso em alguns dos cálculos o resultado não tiver nenhum resto, colocamos simplesmente o resultado sob o respectivo valor e colocamos 0 sob todos os valores restantes à direita.

Atividades de avaliação



1. Converta o número 10110011 binário em hexadecimal e decimal.
2. Converta o número 863 decimal em binário e hexadecimal.
3. Converta o número 37D2 hexadecimal em binário e decimal.

Aritmética binária

As operações aritméticas¹¹ são possíveis em números binários assim como nos números decimais. De fato os procedimentos são completamente similares em ambos os sistemas, assim como em qualquer base de numeração. A multiplicação e a divisão não são realmente difíceis, mas como a não familiaridade com os números binários causa bastante dificuldade para os humanos. Entretanto, como iremos mostrar, para os computadores é extremamente fácil.

¹¹ Aritmética binária: operações com dois ou mais números na representação binária sem necessidade de conversão para outra base

1. Aritmética de Números Inteiros Positivos

1.1. Adição

Quando somamos números decimais, somamos os algarismos de mesma ordem e caso ultrapasse o valor da base, nesse caso 10, acrescentamos uma unidade à ordem seguinte. No caso de adição binária, devemos considerar a base 2.

A adição é quase tão fácil quanto “um mais um é igual a dois”. Na verdade, em binário, “um mais um igual a zero e vai um”. Se convertermos “10” para decimal teremos o valor “2”, que está correto. Também temos que “zero mais zero igual a zero” e “um mais zero é igual a um”.

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 1 &= 10 \end{aligned}$$

Observar que ao adicionar dois “1” produz um resultado com dois dígitos, significa que nós temos um “vai um” para a posição seguinte, da mesma forma como fazemos na adição decimal. Vejam o exemplo abaixo de uma soma binária e a mesma soma em decimal.

$$\begin{array}{r} 11 \\ + 101 \\ \hline 1000 \end{array} \quad \begin{array}{r} 3 \\ + 5 \\ \hline 8 \end{array}$$

As regras de adição binária são mostradas abaixo:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0, \text{ e soma } 1 \text{ ao bit seguinte} \end{aligned}$$

Mais alguns exemplos:

$\begin{array}{r} 11 \\ 00011010 \\ +00001100 \\ \hline 00100110 \end{array}$	$\begin{array}{r} 26 \\ + 12 \\ \hline 38 \end{array}$	$\begin{array}{r} 11111 \\ 00010011 \\ + 00111110 \\ \hline 01010001 \end{array}$	$\begin{array}{r} 19 \\ + 62 \\ \hline 81 \end{array}$
---	--	---	--

1.2. Subtração

A subtração é semelhante à adição, sendo que agora, se não houver quantidade para subtrair fazemos um “empréstimo” da ordem acima.

$\begin{array}{r} 1 \\ 101 \\ - 11 \\ \hline 010 \end{array}$	$\begin{array}{r} 5 \\ - 3 \\ \hline 2 \end{array}$
--	---

As regras de subtração binária são mostradas abaixo:

$$\begin{aligned} 0 - 0 &= 0 \\ 0 - 1 &= 1, \text{ e pede-se } 1 \text{ ao bit seguinte} \\ 1 - 0 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

Mais alguns exemplos:

$\begin{array}{r} 1 \\ 00100101 \\ - 00010001 \\ \hline 00010100 \end{array}$	$\begin{array}{r} 37 \\ - 17 \\ \hline 20 \end{array}$
$\begin{array}{r} 1 \\ 00110011 \\ - 00010110 \\ \hline 00011101 \end{array}$	$\begin{array}{r} 51 \\ - 22 \\ \hline 29 \end{array}$

1.3. Multiplicação

A multiplicação consiste em uma série de somas. Operamos da mesma forma que uma multiplicação na base 10.

$$\begin{array}{r}
 101 \\
 \times 10 \\
 \hline
 000 \\
 101 \\
 \hline
 1010
 \end{array}
 \qquad
 \begin{array}{r}
 5 \\
 \times 2 \\
 \hline
 10
 \end{array}$$

As regras de multiplicação binária são mostradas abaixo:

$$\begin{array}{l}
 0 \times 0 = 0 \\
 0 \times 1 = 0 \\
 1 \times 0 = 0 \\
 1 \times 1 = 1, \text{ e soma } 1 \text{ ao bit seguinte}
 \end{array}$$

Mais alguns exemplos:

$$\begin{array}{r}
 00101001 \\
 \times 00000110 \\
 \hline
 00000000 \\
 00101001 \\
 00101001 \\
 \hline
 0011110110
 \end{array}
 \qquad
 \begin{array}{r}
 41 \\
 \times 6 \\
 \hline
 246
 \end{array}$$

$$\begin{array}{r}
 00010111 \\
 \times 00000011 \\
 \hline
 00010111 \\
 00010111 \\
 \hline
 001000101
 \end{array}
 \qquad
 \begin{array}{r}
 23 \\
 \times 3 \\
 \hline
 69
 \end{array}$$

Podemos observar que a primeira parcela da multiplicação é repetida em toda as linhas quando houver 1 na outra parcela e zero quando o algarismo é 0. Isso sugere que podemos realizar uma multiplicação através de uma série de somas de uma parcela deslocada conforme a segunda parcela. Podemos fazer um circuito eletrônico de multiplicação juntando um somador e um deslocador de bits.

1.4. Divisão

A divisão consiste em uma série de subtrações binárias, semelhante a uma divisão na base 10.

$$\begin{array}{r}
 110 \qquad \qquad \qquad 6 \\
 10 \qquad \qquad \qquad 2 \\
 \hline
 110 \qquad \qquad \qquad 3 \\
 - 10 \quad (10) \\
 \hline
 010 \\
 - 10 \quad (01) \\
 \hline
 000 \quad (11)
 \end{array}$$

2. Aritmética de Números Inteiros Negativos

As coisas são fáceis enquanto fazemos operações com inteiros positivos. Mas as coisas tornam-se mais complicada quando nós queremos fazer operações com números inteiros positivos e negativos.

Inicialmente podemos pensar em fazer como na aritmética decimal, indicando com um sinal especial (– ou +) se o número é negativo ou positivo. Na aritmética binária, nós poderíamos simplesmente reservar um bit para determinar o sinal. Nos circuitos para a adição, nós teríamos um circuito para adicionar dois números e outro para subtrair dois números. A combinação de sinais nas duas entradas determinaria que circuito a se usar com os valores absolutos, e também o sinal da saída.

Apesar deste método funcionar, existe uma forma muito mais fácil de operar aritmética com números negativos em circuitos eletrônicos. Este método é chamado “método do complemento a dois”. Com esse método não é necessário um circuito especial para tratar os sinais e podemos usar um único circuito somador padrão para operar números positivos e negativos. A fim de explicar este método vamos mostrar inicialmente como se trabalharia na aritmética decimal para depois mostrar como trabalhar com aritmética binária.

2.1. Complemento de dez

Imagine o odômetro de um automóvel. Ele tem um determinado número de rodas, cada uma com os dez dígitos gravados. Quando uma roda se mover de 9 a 0, a roda imediatamente à esquerda avança uma posição. Vamos supor que o automóvel ande de ré. Então o reverso acontece, isto é quando uma

roda vai de 0 a 9, a roda a sua esquerda diminui uma posição. Quando todas as rodas são 0, nós interpretamos o valor como o inteiro 0 (zero).

Vamos supor agora que nós temos um circuito de adição que possa armazenar números inteiros com 4 dígitos (imagine um odômetro com 4 dígitos). Vamos realizar a soma $(-2) + 5 = 3$. Vamos considerar que a sequência 0000 seja igual a 0. Nesse circuito vamos representar o valor (-2) com a sequência 9998, isto é, diminuimos 2 unidades de 0000, e o valor (5) com 0005, para então, realizamos a soma tradicional. Primeiramente adicionamos 8 e 5 que dá 3 e vai 1. Em seguida, adiciona 9 e 1, dando 0 e vai 1. Para os demais, o valor será 0 com vai 1, assim que o resultado final será 0003 positivo. Este resultado é correto, tanto pelo valor como pelo sinal.

É importante observar que na representação de complemento de 10 com uma quantidade limitada de dígitos, não podemos usar toda a faixa de valores. Nesse exemplo, os valores 0001 a 4999 representam números inteiros positivos, enquanto os valores 5000 a 9999 representam números negativos, respectivamente de -4999 a -1 . Outro problema é o estouro do registro, se somarmos 1 ao número 4999 não teremos 5000, mas -4999 . Um circuito extra pode detectar essa ocorrência e corrigir o valor.

Vamos terminar esta seção dando um método simples para computar o valor absoluto de um inteiro nessa representação. Basta tomar cada dígito individual, substituí-lo por 9 menos o seu valor original, e então no final, adiciona 1 ao número obtido. Assim por exemplo, o número 9998 transforma-se em 0001 que somando 1 fica 0002 negativo, que é a resposta correta.

2.2. Complemento de dois¹²

Até agora, nós estudamos a representação de números negativos usando o complemento de dez. Em um computador, nós usamos a base dois. Felizmente, o método descrito na seção anterior também funciona para a base dois. Para um somador de n -bits (n é geralmente 32 ou 64), nós podemos representar números positivos com o dígito na posição mais a esquerda seja 0, que dá valores entre 0 e $2^{(n-1)} - 1$, e números negativos com o dígito na posição mais a esquerda seja 1, que dá valores entre $-2^{(n-1)}$ e -1 . A vantagem do complemento a dois é que o dígito a esquerda pode também indicar o sinal do número, 0 para números negativos e 1 para números positivos.

Da mesma forma como no caso complemento de dez, usamos o mesmo princípio para a detecção do estouro. Se, ao adicionar dois números positivos, nós encontramos um resultado negativo (isto é, com o bit mais à esquerda 1), nós teremos um estouro de valor (*overflow*). Da mesma forma, se, ao

¹² Complemento de dois: Técnica para possibilitar a soma e subtração de números positivos e negativos na representação binária

adicionar dois números negativos, encontramos um resultado positivo (isto é com seu bit mais à esquerda 0), temos um sub-valor (*underflow*).

3. Números de ponto flutuante

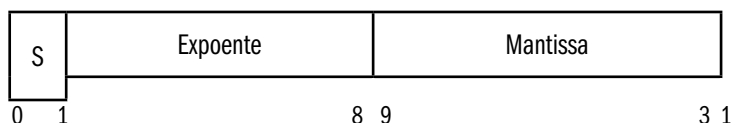
A representação numérica que ainda falta são dos números racionais. Apesar de ser possível a representação de números racionais (por exemplo, representar um número com o seu numerador e denominador), a maioria dos computadores utiliza a representação de um subconjunto dos números racionais, que atende a maioria dos casos. Nós chamamos estes números de *ponto flutuante*.

A idéia básica atrás dos números de ponto flutuante é representar um número com a sua *mantissa* e um *expoente*, cada um com um número fixo de bits referentes a sua precisão. Se nós denotamos a mantissa com m e o expoente com e , o número representado será equivalente a $m \cdot 2^e$.

O uso de ponto flutuante é mais importante para aplicações científicas ou aplicações que exijam muitos cálculos matemáticos como computação gráfica. Nessas aplicações precisamos *multiplicar* ou *dividir* dois números mais frequentemente. A multiplicação de dois números de ponto flutuante é fácil, basta multiplicar as mantissas e somar os expoentes. A divisão é somente um pouco mais complicada. Observe que a imprecisão no resultado de uma multiplicação ou de uma divisão é somente devido à imprecisão nos operandos originais. Nenhuma imprecisão adicional é introduzida pela operação própria (exceto possivelmente 1 unidade em menos dígito significativo).

Para somar dois números de ponto flutuante, o expoente menor deve primeiramente ter sua mantissa deslocada para a direita para igualar os expoentes. Se o deslocamento for maior do que o número de bits na representação da mantissa, o segundo número será tratado como 0.

Geralmente, os computadores manipulam dados em blocos de **8**, **16**, **32**, **64**, ou **128** bits. É conseqüentemente útil caber um único número de ponto flutuante com a mantissa e o expoente em tal bloco. Em tal bloco precisamos ter um bit para o sinal, a mantissa, e o expoente.



Desde 1980 os computadores adotam uma representação padrão para números em ponto flutuante chamada IEEE 754 para facilitar a portabilidade de programas. Os números podem ter precisão simples, constituído com 1 bit de sinal, 8 bits de expoente e 23 bits de mantissa mais 1 implícito, $t_o = 24$,

precisão dupla = 1 bit de sinal, 11 bits de expoente e 52 bits de mantissa + 1 implícito = 53).

Precisão	Sinal	Expoente	Mantissa	Implícito	Total (bits)
Simplex	1	8	23	1	32
Dupla	1	11	52	1	64

Atividades de avaliação



1. Converta os números decimais 19 e 32 em binários, calcule a soma deles em binário e converta novamente para decimal para conferir o resultado.
2. Converta os números decimais 14 e -8 em binários complemento de dois, calcule a soma deles em binário e converta novamente para decimal para conferir o resultado. (dica: use um número de 5 bits).

Portas lógicas

Os blocos básicos de um computador são chamados *portas lógicas* ou apenas *portas*.

As portas são os circuitos básicos que têm pelo menos uma (e geralmente mais) *entradas* e exatamente uma *saída*. Os valores da entrada e da saída são os valores lógicos verdadeiro ou falso. Na convenção de computação é comum usar 0 para falso e 1 para verdadeiro. As portas não têm nenhuma *memória*. O valor da saída depende somente do valor atual das entradas. Este fato torna possível usar uma *tabela de verdade* para descrever inteiramente o comportamento de uma porta.

Nós consideramos geralmente três tipos básicos das portas, as portas E, as portas OU, e as portas NÃO (ou inversores). Também é usual chamar as portas pelo termo em inglês, respectivamente, AND, OR e NOT.

1. Portas básicas¹³

Portas básicas são portas fundamentais que podem formar outras portas com funções especiais. As portas básicas são: E, OU e NÃO.

1.1. Porta E (AND)

Uma porta E pode ter um número arbitrário de entradas. O valor da saída é 1 se e somente se todas as entradas são 1. Se não o valor da saída é 0. O nome foi escolhido porque a saída é 1 se e somente se a primeira entrada e a segunda entrada, e, ..., e a n-ésima entrada são todas 1.

É frequentemente útil desenhar diagramas das portas e das suas interconexões. Em tais diagramas, a porta E é desenhada como esta:



Figura 6 - Ilustração da porta E

¹³ Portas básicas: dispositivo que implementa funções lógicas básicas para todas as funções lógicas.

A tabela de verdade para uma porta E com duas entradas é mostrada abaixo:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

1.2. Porta OU (OR)

Como a porta E, a porta OU pode ter um número arbitrário de entradas. O valor da saída é 1, se e somente se, pelo menos de um dos valores da entrada é 1. Caso contrário, a saída é 0. Ou seja o valor da saída é 0 somente se todas as entradas são 0. O nome foi escolhido porque a saída é 1, se e somente se, a primeira entrada ou a segunda entrada, ou, ..., ou a n-ésima entrada são 1.

Em esquemas de circuito, nós desenhamos a porta OU como este:



Figura 7 - Ilustração da porta OU

A tabela de verdade para uma porta OU com duas entradas olha como esta:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

1.3. Porta NÃO (NOT)

A porta NÃO também é conhecida como *inversor* e tem exatamente uma entrada e uma saída. O valor da saída é 1 se e somente se a entrada é 0. Se não, a saída é 0. Ou seja o valor da saída é exatamente o oposto do valor da entrada.

Em esquemas de circuito, nós desenhamos a porta NÃO como este:



Figura 8 - Ilustração da porta NÃO

A tabela de verdade para um *inversor* olha como esta:

x	y
0	1
1	0

2. Portas compostas¹⁴

Às vezes, é prático combinar funções das portas básicas em portas mais complexas a fim reduzir o espaço em diagramas de circuito. Nesta seção, nós mostraremos algumas portas compostas junto com suas tabelas de verdade.

¹⁴ Portas Compostas: estrutura de lógica complexa implementado através da associação de várias portas básicas.

2.1. Porta NÃO-E (NAND)

A porta NÃOE é uma porta E com um inversor na saída. Assim em vez de desenhar diversas portas com esta:



Figura 9 - Ilustração da porta-E

Nós desenhamos uma única porta-E com um pequeno anel na saída como este:

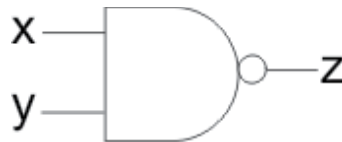


Figura 10 - Ilustração da porta NÃOE

A porta-nãoe, assim como a *porta-E* pode ter um número arbitrário de entradas.

A tabela de verdade para a porta-nãoe é semelhante à tabela para porta-e, exceto que todos os valores da saída são invertidos:

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

2.2. Porta NÃO-OU (NOR)

A porta NÃO-OU é uma porta OU com um inversor na saída. Assim em vez de desenhar diversas portas como esta:



Figura 11 - Ilustração da porta OU

Nós desenhamos uma única porta OU com um pequeno anel na saída como este:

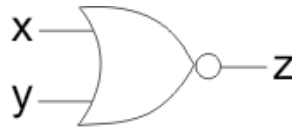


Figura 12 - Ilustração da porta NÃO-OU

A *porta não-ou*, assim como a *porta ou* pode ter um número arbitrário de entradas.

A tabela de verdade para a *porta não-ou* é semelhante à tabela para *porta ou*, exceto que todos os valores da saída são invertidos:

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

2.3. Porta OU-EXCLUSIVO (XOR)

A porta OU-EXCLUSIVO é similar a uma porta OU. Apesar de poder ter um número arbitrário de entradas normalmente representamos apenas duas entradas. O valor da saída é 1 se somente uma das entradas é 1 e a outra é 0, por isso se diz OU-EXCLUSIVO. Se não, a saída é 0. Assim, em vez de desenhar diversas portas como esta:

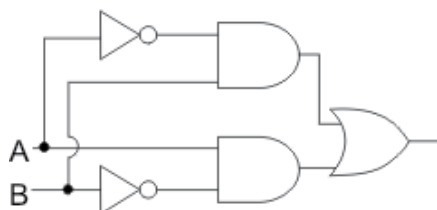


Figura 13 - Ilustração da porta OU-EXCLUSIVO

Nós desenhamos uma porta OU-EXCLUSIVO como mostrada abaixo:



Figura 14 - Ilustração da porta OU-EXCLUSIVO

A tabela de verdade para uma porta OU-EXCLUSIVO com duas entradas é mostrada abaixo:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

Mas para que serve uma porta OU-EXCLUSIVO? Ela é um bloco essencial para a construção de um circuito muito útil em computadores, o SOMADOR.

2.4. Meio Somador¹⁵

O Circuito somador realiza a soma de um bit tendo como resultado a valor da soma e o bit de vai-um conforme diagrama mostrado abaixo:

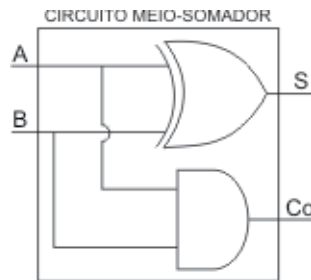


Figura 15 - Ilustração do circuito meio somador

A tabela de verdade de um meio-somador é mostrado abaixo:

x	y	z	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

¹⁵ Meio-somador: Circuito que realiza a soma de um bit utilizando portas lógicas básicas.

¹⁶ Somador completo: Circuito somador que realiza a soma considerando vai-um. O somador completo pode ser cascadeado para formar um somador de número contendo vários bits.

Somador Completo¹⁶

Apesar do meio somador informar o bit de vai um, ele não considera o vai um do bit anterior, assim ele só pode ser usado para somar o bit menos significativo.

Para realizar uma soma completa precisamos utilizar o circuito somador completo que tanto informa o vai-um para o próximo bit, como considera o vai-um do bit anterior da soma. O diagrama do somador completo é mostrado abaixo:

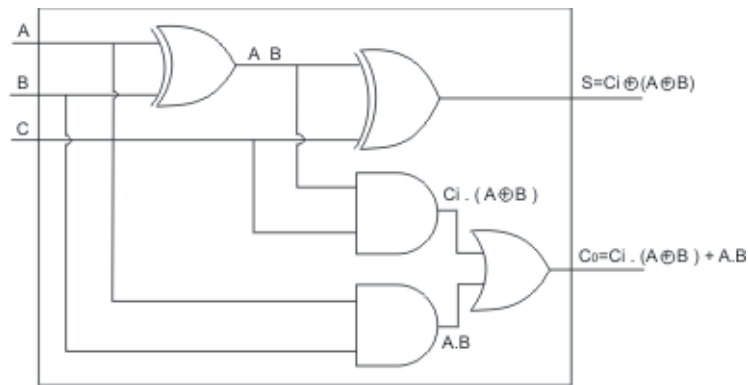


Figura 16 - Ilustração do circuito somador completo

2.6. Fazendo qualquer porta composta usando somente um tipo da porta

Por questões de processo de fabricação de circuitos integrados algumas tecnologias de circuito, são mais fáceis construir uma porta NÃO-E do que outras portas. Menor ocupação de espaço e menor consumo de energia são fatores considerados. Nesse caso, a porta NÃO-E é considerada o bloco mais básico, e todos os outros são construídos a partir dele.

É possível construir qualquer tipo da porta usando somente portas NÃO-E. Primeiramente podemos observar que um inversor é apenas uma porta NÃO-E com somente uma entrada. Em seguida, uma PORTA-E pode ser construída como uma porta NÃO-E com um inversor em sua saída. Uma porta OU pode ser construída com uma porta NÃO-E com um inversor em cada entrada.

Da mesma forma, todas as portas podem ser construídas usando somente portas NÃO-OU. E a partir dela podemos construir as demais.

Síntese do capítulo

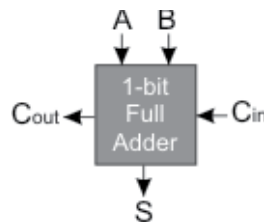


Este capítulo apresentou as formas de representação de informações em um computador. Foi mostrado a unidade básica de informação, o bit, e as representações de dados em byte, caracteres e números. Foi apresentada uma introdução ao sistema binário e a representação de números e operações matemáticas. Finalmente mostramos uma introdução à lógica digital e as principais portas lógicas existentes assim como a composição de portas mais complexas.

Atividades de avaliação



1. Construa um somador de 4 bits utilizando blocos somadores-completos mostrado na figura abaixo:



2. Observando as tabelas verdades mostradas nessa seção, desenhe o esquemático de uma porta OU-EXCLUSIVO utilizando apenas portas NÃO-E.

Leituras, filmes e sites



Como Tudo Funciona: Informática

<http://informatica.hsw.uol.com.br/>

Introdução à Computação

<http://www.cic.unb.br/~jhcf/MyBooks/ic/index.html>

Introdução à Ciência da Computação

http://pt.wikiversity.org/wiki/Introdução_à_Ciência_da_Computação

Referências



THOMAS FLOYD. **Sistemas Digitais**. 9ª Ed. Editora: Bookman, 2007.

RONALD J. TOCCI, NEAL S. WIDMER, GREGORY L. MOSS. **Sistemas Digitais: Princípios e Aplicações**. 10ª Ed. São Paulo: Editora Pearson Prentice-Hall, 2007.

ANDREW S. TANENBAUM. **Organização Estruturada de Computadores**. 5ª Ed. São Paulo: Editora Pearson Prentice-Hall, 2006.

JOHN L. HENNESSY, DAVID A. PATTERSON. **Organização e Projeto de Computadores**. 3ª Ed. São Paulo: Editora Campus, 2005.

Parte

3

Arquitectura de computadores

Arquitetura de computadores

Introdução

Neste capítulo será apresentada a arquitetura de um computador genérico com a função de cada um dos seus componentes principais. Será apresentada a relação entre *hardware*¹⁷ e *software*¹⁸ mostrando os fatores relevantes que influenciam o desempenho.

Serão apresentados a estrutura da Unidade Central de Processamento (CPU) com seus principais componentes, além das características que determinam a sua capacidade de processamento.

Também serão mostrados as características dos diversos tipos de memória e a interação de cada uma com os demais componentes do computador. Finalmente, serão mostrados os diversos dispositivos de entrada e saída existentes que permitem a utilização dos computadores pelo ser humano de forma confortável e eficaz.

Os primeiros computadores do mundo eram as pessoas. Por isso, para chamar os primeiros os computadores mecânicos ou eletrônicos foram dados nomes das pessoas que executavam o trabalho de “computar”. O “computador” era uma profissão (exercida predominantemente por mulheres) cujo trabalho era executar os cálculos repetitivos exigidos para computar tabelas de navegação, cartas de marés e posições planetárias.

Imagine que você tenha um trabalho de ficar hora após hora, dia após dia, fazendo nada além de computar multiplicações. Com o tempo o cansaço iria conduzir à negligência e ao erro. Por isso, durante vários séculos os inventores têm procurado por uma maneira que mecanize (isto é, encontrar um mecanismo que pudesse executar) esta tarefa.

A máquina que você pensa ser um “computador” é chamado mais precisamente “computador eletrônico digital de uso geral.” É de uso geral porque pode ser programado para executar uma grande variedade de aplicações (que fazem a diferença de um computador de propósito específico, projetado para executar somente uma função).

¹⁷. Hardware: conjunto de componentes físicos de um computador como processador, memória, disco, etc.

¹⁸. Software: conjunto de comandos lógicos que fazem um computador funcionar como programas, sistema operacional, etc.

Digital significa que o computador guarda e manipula todos os dados internamente sob a forma dos números (todos os dados numéricos, todos os dados de texto, e mesmo os sons e as figuras são armazenadas como números). O dígito significa originalmente “dedo” e desde que os povos começaram a contar com seus dedos, o nome dígito veio ser aplicado aos números.

Os computadores modernos são totalmente eletrônicos porque manipulam dados usando circuitos de chaveamento eletrônico mas algumas máquinas de computação mais velhas, eram mecânicas, usando rodas, alavancas, etc.

Um computador é um dispositivo que executa quatro funções: ele recebe dados de entradas (converte dados do mundo exterior para o universo eletrônico); armazena os dados (de forma eletrônica); processa dados (executa operações matemáticas e lógicas); e exibe os dados de saídas (mostra os resultados para os usuários através de uma tela).

Um computador consiste no *hardware* e no *software*. O *hardware* é o equipamento físico: o próprio computador e os periféricos conectados. Os periféricos são todos os dispositivos ligados ao computador para finalidades de entrada, saída, e armazenamento dos dados (tais como um teclado, um monitor de vídeo ou um disco rígido externo).

O *software* consiste nos programas e nos dados associados (informação) armazenados no computador. Um programa é uma sequência de instruções que o computador segue com o objetivo de manipular dados. A possibilidade de incluir ou excluir programas diferentes é a fonte de versatilidade de um computador.

Sem programas, um computador é apenas *hardware* de alta tecnologia que não faz qualquer coisa. Mas com a sequência de instruções detalhadas, descrevendo cada passo do programa (escrito por seres humanos) o computador pode ser usado para muitas tarefas que variam do processamento de texto a simulação de padrões de tempo globais.

Como um usuário, você irá interagir com os programas que funcionam em seu computador através dos dispositivos de entrada conectados a ele, tal como um mouse e um teclado. Você usa esses dispositivos para fornecer a entrada (tal como o texto de um relatório que você está trabalhando) e para dar comandos ao programa (tal como a definição de que uma frase do texto vai aparecer com formato negrito). O programa fornecerá a saída (os dados resultantes das manipulações dentro do computador) através de vários dispositivos de saída (tal como um monitor ou uma impressora).

Os computadores pessoais (geralmente conhecidos como PC) são usados de uma maneira muito interativa, com o usuário inserindo continuamente dados e comandos (para escolher as várias funções do programa)

e monitorando a saída com os resultados das operações realizadas. Isto é muito diferente da maneira que os grandes computadores mais antigos eram usados (o usuário fornecia a entrada e recebia a saída mais tarde, o que é chamado processamento em lote).

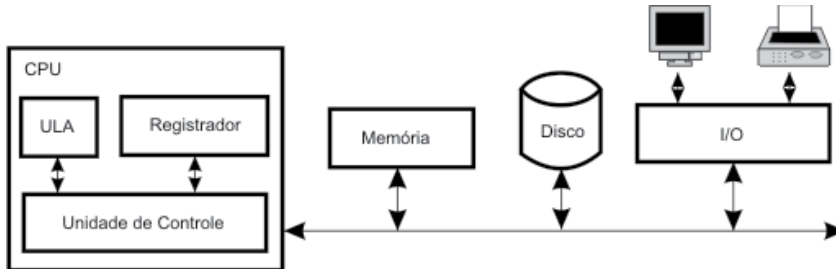


Figura 17 - Arquitetura de um computador

A Figura 17 mostra a arquitetura simplificada de um Computador. Os computadores modernos tem uma arquitetura bem mais complexa do que esta, mas os blocos principais estão aqui representados. O principal módulo é a Unidade Central de Processamento, usualmente conhecida como CPU¹⁹ (*Central Processing Unit*). A CPU é o “cérebro” do computador onde todo o processamento é realizado. Ela é constituída por três submódulos: a Unidade Lógica e Aritmética (ULA), os Registradores e a Unidade de Controle. Nos computadores modernos a CPU é construída em um único Circuito Integrado (chip).

A Unidade Lógica e Aritmética (ULA), também conhecida como *Arithmetic and Logic Unit* (ALU), é responsável por realizar as operações do computador como soma e subtração assim como as funções lógicas, OU, E, NÃO e OU Exclusivo. Algumas ULAs mais sofisticadas realizam também multiplicações e divisões.

Os registradores são *memórias*²⁰ utilizadas para realizar as operações na CPU. Essas memórias são muito rápidas e de tamanho pequeno, geralmente suficiente para guardar menos de uma dezena de valores. Além dos registradores é comum haver uma memória de rascunho dentro da CPU, chamada de Cache. Ela é utilizada para aumentar a velocidade de processamento reduzindo o tempo de acesso à memória externa.

A Unidade de Controle é responsável por controlar todo o funcionamento da CPU e também de todo o computador. Ela controla o processamento entre ULA e Registrador e interage com o barramento externo onde ficam os periféricos. A Unidade de Controle tem uma função chave que é a interpretação do código do programa que irá nortear os comandos do processamento.

Continuando na Figura 17 podemos observar um barramento e três módulos externos à CPU: Memória, Disco e Interface de Entrada e Saída (I/O).

¹⁹. CPU: acrônimo de Central Processing Unit ou Unidade Central de Processamento. Unidade responsável por realizar o processamento no computador.

²⁰. Memória: componente construído com material semi-condutor ou magnético responsável por armazenar, temporariamente ou permanentemente, os dados em um computador.

O barramento serve para interligar todos os componentes internos do computador. Para se obter altas velocidades esse barramento é geralmente paralelo (os dados são enviados paralelamente). Nessa figura representamos apenas um barramento, mas geralmente ele é dividido em barramento de dados (onde os dados trafegam), barramento de endereço (onde indicamos a localização de cada dado) e o barramento de controle (onde indicamos o comando, por exemplo, ler, escrever, copiar, etc). Em um computador real existem vários níveis de barramento com velocidades diferentes, quanto mais próximo da CPU mais rápido ele é, e quanto mais próximo dos periféricos mais lento ele é.

A Memória é constituída por um conjunto de memória semicondutora utilizada para armazenar os dados temporariamente. Podemos dizer que essa é a memória de trabalho do computador e geralmente fica localizada na placa mãe do computador. Essa memória é mais lenta que os Registradores mas é mais rápida que as unidades de armazenamento em disco. A memória semicondutora geralmente perde os dados quando o computador é desligado, por isso deve-se usar uma unidade de armazenamento permanente, o disco.

O disco consiste em uma unidade eletro-mecânica que armazena os dados em um disco magnético que mantém as informações mesmo quando o computador é desligado. Essa unidade tem grande capacidade de armazenamento mas tem velocidade de acesso significativamente menor em relação às memórias semicondutoras.

O módulo de Entrada e Saída, ou Input/Output (I/O), estabelece a ligação do computador com o mundo externo, usando equipamentos periféricos. Essa interface permite a ligação de teclados e mouses (para entrada de dados), monitores ou impressoras (para exibição dos dados) e placas de comunicação (para trocar dados a longa distância). Essa interface é significativamente mais lenta que os demais componentes do computador devido a natureza dos periféricos e a incapacidade humana de processar informações na velocidade dos computadores.

Os próximos capítulos irão detalhar cada um dos módulos que constituem um computador.

Atividades de avaliação



1. Observando a Figura 17 avalie os componentes de uma arquitetura de computadores que podem limitar o desempenho. Proponha soluções para resolver esse gargalo.
2. Pesquise sobre a história do computador e liste as invenções mais importantes.
3. Qual módulo de uma arquitetura de computadores deve ser mais rápido para melhorar o desempenho de um sistema de processamento numérico (muitas operações matemáticas)?
4. Qual módulo de uma arquitetura de computadores deve ser mais rápido para melhorar o desempenho de um sistema de armazenamento de dados (muitos dados guardados e consultados)?

Unidade central de processamento e memória

1. Processador

O processador é o componente do computador que faz o processamento real dos dados. Um microprocessador é uma unidade do processador central (CPU) encapsulada dentro de um microchip. É o “cérebro” do computador, mas devemos considerar ser um termo um pouco exagerado pois ele realmente é apenas um circuito eletrônico constituído por interruptores que executam instruções simples mas com muita rapidez.

O circuito integrado do microprocessador contém uma placa de silício que contém milhões de transistores e outros componentes fabricados no próprio silício. Como os transistores no *chip* são muito minúsculos, mesmo uma pequena descarga de corrente, como a eletricidade estática da mão do técnico, pode destruí-lo. Eis porque todos os circuitos integrados devem ser manipulados de forma que minimize a possibilidade de descarga estática através de acessórios apropriados (pulseira aterrada).

Por causa da grande quantidade de circuitos encapsulados em uma área tão minúscula, a energia necessária para seu funcionamento gera muito calor e exige sistemas de resfriamento para evitar o superaquecimento do chip. Na placa mãe de um computador o chip do processador central é coberta por um grande dissipador de calor em metal com aletas para permitir que o fluxo de ar dos ventiladores de refrigeração retirem o calor gerado pelo chip.

1.1. Velocidade de relógio

Os diversos chips em uma placa mãe precisam de uma sincronização única para manter o fluxo de dados coerente. Para isso um circuito de relógio (*clock*) distribui um sinal único para todos os chips que passarão a se comunicar de forma coerente. Você pode imaginar o relógio como uma “pulsção do coração” do computador. Quanto mais rápidos os impulso de relógio, mais rapidamente

o computador funciona; mas, o pulso de disparo não pode funcionar mais rapidamente do que a capacidade de velocidade dos chips. Com a melhoria da tecnologia de construção dos chips, a velocidade em que eles podem funcionar aumentou, aumentando a velocidade de processamento. Em razão disso, atualmente o processador central funciona mais rapidamente do que o barramento da placa mãe, que geralmente usa uma fração da taxa do processador central.

A velocidade de relógio é medida em unidades de ciclos por segundo, que é chamado de **Hertz**²¹ (Hz). As placas e os processadores centrais do computador funcionam em taxas de milhões e bilhões de Hertz, **megahertz**²² (MHz) e **gigahertz**²³ (GHz). Uma boa velocidade para um microprocessador do PC em 2008 era 4 gigahertz. Obviamente os componentes em velocidades mais altas são mais críticos e tem um custo geralmente maior.

²¹ Hertz (Hz): unidade de medida de frequência, significando quantidade de ciclos por segundo.

²² Megahertz (MHz): unidade de medida de frequência, significando quantidade de milhões de ciclos por segundo.

²³ Gigahertz (GHz): unidade de medida de frequência, significando quantidade de bilhões de ciclos por segundo.

Você pode reconhecer os termos megahertz e gigahertz das emissões de rádio. O rádio de FM e a transmissão de TV ficam na escala do megahertz, e os telefones móveis transmitem na escala do gigahertz. Assim, os circuitos do computador produzem interferências na frequência que poderia causar problemas nos dispositivos próximos. Para impedir isso, um computador geralmente contém uma caixa metálica onde ficam a placa mãe e os demais componentes.

1.2. Velocidade do processador central

A velocidade de relógio é somente um aspecto que contribui à velocidade total de processamento de um microprocessador. A arquitetura do microprocessador também é relevante. Um fator importante é o tamanho da palavra do microprocessador, que é quantos bits de entrada/saída que ele pode processar simultaneamente em um determinado instante. Os microprocessadores mais antigos usavam o tamanho de palavra de 8 bits; os microprocessadores mais novos usam o tamanho de palavra de 64-bits.

O projeto do microprocessador pode incluir a memória cache de alta velocidade. A memória cache é uma memória instalada entre o processador e a memória RAM que permite ao processador guardar instruções ou dados mais recentemente usados para o caso de precisá-los outra vez, de modo que não precise ir buscá-los na memória RAM principal, muito mais lenta que a memória cache.

Os computadores gastam muito tempo nos laços (loops), repetindo a mesma sequência de instruções várias vezes, assim se esse código estiver na memória cache pode melhorar significativamente o desempenho. Dependendo se a memória cache está situada no chip do processador central, ou em chips externos próximos com uma ligação de alta velocidade, é chamado cache L1 ou cache L2.

Outros aspectos da arquitetura dos microprocessadores que afetam a velocidade incluem a habilidade de alguns processadores centrais de trabalhar ao mesmo tempo com instruções múltiplas em paralelo. Também, alguns processadores centrais são CISC²⁴ (conjunto de instrução complexa), enquanto outros são RISC (conjunto de instrução reduzida). Os microprocessadores RISC²⁵ têm um conjunto de instruções mais simples e precisam de mais instruções para executar uma mesma ação que um microprocessador CISC faz com uma instrução, mas os processadores RISC são mais rápidos e podem terminar a operação antes.

Até hoje se discute a eficiência dos processadores RISC e CISC, mas na verdade o desempenho é dependente do tipo de instrução executada e do tipo de aplicação usada. O resultado é que você não pode simplesmente comparar processadores diferentes olhando apenas sua velocidade de relógio. Um processador PowerPC com o dobro da velocidade de relógio de um processador *Pentium* tem a velocidade de processamento aproximadamente equivalente.

1.3. Tipos de microprocessadores

O microprocessador mais usado nos PC é feito pela Intel. Desde que a IBM escolheu o processador Intel 8088 para o PC original da IBM, a maioria de clone do PC usam um processador da série Intel: 8088, 80286, 80386, 80486, *Pentium*, *Pentium II*, *Pentium III* e *Pentium 4*. A Figura 18 mostra um processador Intel.



Figura 18 - Processador Intel (Intel Corp)

Outro fabricante dos microprocessadores para o PC é AMD (*Advanced Micro Devices, Inc.*). Sua linha de processadores Athlon foi bem sucedida e tomou uma fração do mercado do processador da Intel. O processador AMD é compatível com o Intel, isto é, possuem o mesmo conjunto de instruções e qualquer *software* funciona em ambos os processadores sem alteração.

A série de computadores Macintosh da Apple usou originalmente o microprocessador Motorola 68000. Os processadores da Motorola usam um conjunto de instrução diferente do que os processadores da Intel, por isso o *software* do PC não funciona em um Mac e vice-versa. A Apple usou mais tarde o processador RISC PowerPC, desenvolvido pela Apple, Motorola e IBM. Os Mac a partir de 2004 usaram processador PowerPC G4 ou o PowerPC G5. Em 2006 a Apple passou a usar processadores Intel em seus Mac, tornando possível o funcionamento de *software* Windows diretamente no Mac.

^{24.} C I S C : processador com conjunto de instrução complexa.

^{25.} R I S C : processador com conjunto de instrução reduzida.

²⁶ FSB (Front Side Bus): Barramento principal de um computador PC.

2. Barramento de dados

O barramento de dados é a estrada elétrica com várias vias que ligam o processador central aos outros componentes da placa mãe, tal como memória RAM, disco rígido e controladores de I/O. Esse barramento é chamado *Front Side Bus (FSB)*²⁶. O tamanho da palavra do barramento de dados determina quantos bits podem ser movidos simultaneamente ao longo dele, tipicamente nos computadores modernos usa-se 32 bits ou 64 bits.

A velocidade de relógio dos outros componentes ligados ao barramento de dados na placa mãe é mais lenta do que a velocidade de relógio do processador central. Por isso é comum na especificação de um computador diferenciar a velocidade do processador, tipicamente na ordem de gigahertz, e a velocidade do barramento, tipicamente centenas de megahertz.

3. Memória

A memória de um microcomputador é o lugar onde os programas e os dados são armazenados quando estão sendo usados. Existem vários tipos de memórias em um microcomputador que serão comentadas a seguir.

3.1. Memória RAM

O RAM é o espaço de memória principal do computador. O termo RAM significa memória de acesso aleatório (*Random Access Memory*), e o termo vem dos primeiros computadores quando as unidades centrais tinham dois tipos de memória: Acesso aleatório, em que todo o bit da memória poderia ser endereçado a qualquer instante; e memória seqüencial (tal como os dados armazenados em fita) onde os bits poderiam somente ser alcançados de forma sequencial. Toda memória em um computador moderno é de acesso aleatório, assim não se preocupe com a memória seqüencial.

O RAM é a área de trabalho do computador. Se o computador tem mais RAM, pode abrir mais programas e mais arquivos simultaneamente. É como você tivesse uma grande mesa de trabalho onde pudesse espalhar muito mais papéis de trabalho sobre ela em vez de uma pequena mesa. Um arquivo que você está editando (e o programa que o seu computador está usando para editar) é armazenado na RAM.

O RAM consiste em bancos de transistores do circuito integrado onde podem estar ligados ou desligados (representando um ou zero). Os chips de RAM precisam energia constante para manter o que é armazenado nele; um desligamento da energia, mesmo que seja uma fração de segundo (por exem-

plo, causado por um relâmpago na rede elétrica) pode fazer com que o RAM perca seus dados. Por esse motivo, a RAM é chamada de volátil (ou aquilo que se evapora facilmente) e por isso é importante salvar frequentemente seu trabalho em um armazenamento mais permanente, tal como um disco rígido.

O tipo específico de RAM usado em um computador poderia ser DRAM (RAM Dinâmica), ou SDRAM (RAM Dinâmica Síncrona) ou RDRAM (RAM Dinâmico Rambus), mas não se preocupe sobre os detalhes. Apenas tenha cuidado para escolher o modelo apropriado para seu sistema quando você compra mais memória RAM.

O RAM é instalado geralmente na placa mãe em soquetes como DIMMs (*Dual Inline Memory Module*), pequena placa de circuito impresso com chips de memória RAM (Figura 19). A arquitetura da placa mãe limitará a quantidade de memória RAM extra você pode adicionar. O tamanho da memória RAM é medido em bytes, quilo-bytes, megabytes, etc.



Figura 19 - Módulo DIMM de Memória RAM (Spansion)

3.2. Memória ROM

Os computadores também têm alguma memória ROM (*Read Only Memory* ou Memória apenas para leitura) na placa mãe. A ROM, apesar de ser construída por transistores, não precisa de energia para guardar suas informações mas também não podem ser re-escritas.

Quando o computador é ligado, o programa armazenado na ROM é executado, dando os primeiros comandos para o computador funcionar. Esse programa, chamado de BIOS (*Basic Input/Output System* ou Sistema Básico de Entrada/Saída) executa todas as funções iniciais de um computador, por exemplo, verifica se o *hardware* está em ordem e procura os dispositivos de armazenamento onde pode encontrar uma cópia do Sistema Operacional que será carregado na memória RAM (que encontra-se apagada pois o computador estava desligado) e a seguir entrega o controle do computador ao Sistema Operacional.

Devemos lembrar que o código de BIOS nos computadores modernos é gravado em dispositivos regraváveis como EEPROM (ROM Programável e Apagável Eletricamente). A placa mãe inclui circuitos especiais que permitem que os arquivos de BIOS “permanentes” sejam atualizados quando necessário, mas que é feito raramente (EEPROM pode somente ser re-escrita um número limitado de vezes, da ordem de centenas a milhares de vezes).

3.3. Memória CMOS

A memória CMOS de um computador é uma pequena quantidade de memória “semi-permanente” onde os dados podem ser armazenados e re-escritos quantas vezes quiser mas que permanecer disponíveis quando o computador for desligado. Uma bateria pequena na placa mãe mantém a memória CMOS ‘viva’ quando não há energia.

A memória do CMOS pode armazenar parâmetros de configuração de um computador como características de disco e interfaces, etc. A BIOS irá ler essas informações e usá-las durante o carregamento do sistema operacional. A memória do CMOS pode também guardar a data e a hora permitindo que o computador mantenha a data e hora correta mesmo estando desligado. O computador tem um relógio alimentado pela pequena bateria.

A memória CMOS é um tipo de memória de transistor que exige a potência muito pequena para armazenar dados, com uma pequena bateria os dados são mantidos por um longo tempo. Entretanto, quase todas os chips de memória em um computador hoje em dia (como a RAM) usam a tecnologia CMOS, e ironicamente, “a memória semi-permanente CMOS” nos PCs modernos não é armazenada em memórias CMOS, mas em uma memória Flash, que é um tipo de memória EEPROM com tamanho maior e com capacidade de re-escrita.

3.4. Memória Virtual

A memória virtual não é necessariamente um novo componente de um computador, mas apenas uma técnica que os sistemas operacionais usam para ultrapassar os limites de memória no computador.

A quantidade de RAM limita o tamanho e quantidade de arquivos de dados e programas podem ser aberto simultaneamente. Você não poderia abrir simultaneamente dois programas que exigem 80 MB de RAM cada um em um computador que tenha somente 128 MB do RAM.

Entretanto, os sistemas operacionais modernos usam a memória virtual para permitir a ampliação dessa capacidade. Se tudo não couber na memória RAM disponível, o sistema operacional pode automaticamente escrever os dados não utilizados nesse momento no disco rígido, e coloca na memória os dados que está utilizando. Mas isso tem um preço! Os disco rígidos (dispositivos mecânicos) são muito mais lentos do que a memória RAM (dispositivo eletrônico), assim o computador funcionará muito mais lento se usar a memória virtual.

Dispositivos de armazenamento

Os dispositivos de armazenamento, tais como unidades de disco, armazenam programas e arquivos de dados quando não estão sendo usados. Ao contrário das memórias RAM, os dados armazenados nestes dispositivos não desaparecem quando a energia é desligada.

Os tipos principais de dispositivos de armazenamento são: discos magnéticos, disco de estado-sólido e disco ótico.

1. Disco rígido

Um disco rígido é um dispositivo mecânico constituído por discos de metal e revestidos com óxido de metal que permite ser magnetizado. Uma cabeça pequena de leitura/gravação eletromagnética na extremidade de um braço móvel magnetiza pontos minúsculos no disco para armazenar dados. Todos esse componente são lacrados em uma caixa hermeticamente fechada. A distância entre o disco e a cabeça de leitura/gravação é comparável ao de um grão de poeira, caso exista alguma sujeira na superfície do disco poderá danificar definitivamente a unidade de disco.

Os pontos magnéticos magnetizados em um sentido representam o valor binário “um” e os pontos magnetizados no sentido oposto representam um “zero”. A mesma cabeça eletromagnética pode mais tarde detectar o sentido do campo magnético dos pontos quando o disco passa debaixo da cabeça, identificando se o dado é 0 ou 1 conforme o sentido de magnetização. A Figura 20 mostra um disco rígido com seus principais componentes.



Figura 20 - Disco Rígido IBM

Os discos rígidos são avaliadas por sua capacidade de armazenamento, tipicamente dezenas ou centenas de gigabyte. Outra métrica de avaliação é quanto rapidamente os discos giram (em RPM, Rotações por Minuto), que é

tipicamente milhares de RPM. Uma outra métrica de avaliação de um disco rígido é o tempo de acesso médio (medido em milissegundos), que diz em média quanto tempo levaria a movimentação do braço para recuperar um dado no disco. Os tempos de busca típicos ao redor 6 ms.

A eletrônica que controla o disco rígido incorpora geralmente alguma memória cache. O movimento do braço lê diversos setores dos dados de uma só vez e se o processador central pedir os setores seguintes, a memória cache pode responder imediatamente sem ter que esperar o disco girar e posicionar o braço no local correto.

Existem vários padrões de controlador para interface com disco rígido como: IDE, ATA, SATA, SCSI, ou outros. Não vamos entrar em detalhes sobre esses padrões aqui, apenas você deve ter em mente que deve se usar o mesmo padrão interface para se ligar o disco na placa mãe.

2. Disquete flexível

Um disquete flexível é um disco plástico flexível de *mylar* (tipo de plástico) coberto com óxido de metal que é magnetizado através de uma cabeça magnética, de forma semelhante ao disco rígido. As disquetes flexíveis tem 3.5 “ de diâmetro e dispõe de uma capacidade de 1.44 MB para armazenamento.

O disquete dispõe de um obturador que protege a superfície de disco da sujeira e das impressões digitais, o obturador desliza expondo a mídia magnética quando o disco é introduzido na unidade de leitura de modo que as cabeças de leitura/gravação possam alcançar o disco magnetizável.

Uma trava plástica pode ser deslizado para desbloquear um furo no canto do disquete para proteger o disco de escrita (assim dados não podem ser acidentalmente apagados). O tempo de acesso é muito mais lento do que um disco rígido, e a vida útil é significativamente mais curta.

As unidades de disquete flexível estão caindo em desuso. A maioria dos computadores novos não dispõe dessa interface. O motivo desse abandono é a redução de custo de outras mídias com maior capacidade, como CD Regravável, DVD Regravável e Memória Flash (*pen drive*), com capacidades de centenas de megabytes a vários gigabytes.

3. Cartões da memória Flash

Memória Flash é um tipo especial de memória RAM desenvolvida nos anos 80 que apresenta a característica de não ser volátil, isto é, não perde os dados quando a energia é desligada. Apesar de ser um pouco mais lenta que as memórias RAM voláteis ela é muito mais rápida que os discos rígidos.

A memória *Flash* se popularizou com a disponibilidade de interface USB, que permitiu a construção de memórias não voláteis portáteis que podem se ligar em qualquer porta USB com uma velocidade bastante alta. Esse dispositivo chamado *pen drive* ou *memory key* tem se tornado cada vez mais popular em virtude da praticidade e baixo preço. A Figura 21 mostra um *pen drive* USB.

O mesmo tipo da memória Flash pode ser usado em pequenos cartões de memória como cartão SD, *Compact Flash*, *Memory Stick*. Esses cartões são usados em telefones celulares, PDAs, câmeras digitais, tocadores MP3, e outros dispositivos digitais. Você pode ler um cartão desses em um leitor de cartão da memória instalado no computador. Estes cartões de memória tem capacidades de armazenamento de centenas de megabytes a dezenas de gigabyte.



Figura 21 - Pendrive Sandisk

4. CD-ROM

Um CD-ROM é um dispositivo de memória ótica com capacidade de até 700 MB. O ótico significa que a luz é usada para ler os dados do disco (não é uma mídia magnética). Os CD-ROM são muito baratos se produzidos em grandes quantidades e a maioria de *software* são distribuídos em CD-ROM.

Os dados são armazenados em um CD-ROM através de pequenos poços gravados na camada plástica interna que é aluminizada e recoberta com uma outra camada plástica transparente para proteção. Um raio laser projetado sobre o disco girando reflete uma seqüências dos poços e não-poços que converte em dados zero e um.

O desempenho de CD-ROM são avaliadas pela velocidade, tal como 32x, que significa 32 vezes mais rápido do que a primeira geração de CD-ROM.

4.1. CD-R e CD-RW

Os CD-R (CD Gravável) e CD-RW (CD Regravável) são os CD que podem ser escritos e re-escritos, respectivamente.

Os discos dos CD-R têm uma camada de tinta que é alterada por um laser de potência mais elevada que escreve os zeros e uns no disco. Na leitura se usa um laser mais fraco que reflete os poços mas não altera os dados. Os CD-R podem ter sua superfície escrita apenas uma vez, após o qual pode-se apenas realizar leitura. Assim como os CD, os CD-R pode armazenar até 700 MB dos dados.

Os discos CD-RW contêm um material sensível especial e usa dois raios laser diferentes, um que escreve (com potência alta) e lê (com potência baixa) e um outro laser que “apaga” o conteúdo, deixando o disco pronto para ser regravado. Assim esses discos podem ser usados muitas vezes (mas devem ser apagados antes da re-escrita).

4.2. DVD

Os discos de DVD-ROM são mídias óticas similares aos CD-ROM, mas com uma capacidade de armazenamento maior. Os DVD permitem gravação de 4.7 GB de dados em mídia de camada única ou 8.54 GB em camada dupla.

Como um CD, os DVD tem igualmente tem variações de gravação como DVD-R para o gravável e DVD-RW para o regravável. As velocidades de leitura de DVD são avaliadas nos termos de quantas vezes ele é mais rápido que os drives originais de DVD, por exemplo, um disco 6x equivale a dizer que ele é 6 vezes mais rápido que os primeiros DVD.

Atividades de avaliação



1. Relacione os principais componentes de uma CPU e descreva a função de cada um.
2. Por que é importante para um computador ter um relógio único?
3. Diga a diferença entre memória RAM e memória ROM?
4. Qual a forma de codificação de dados que podemos usar em discos rígidos e flexíveis e unidade de CD ou DVD?

Sistemas de entrada e saída

Em computação, o sistema de entrada/saída, em inglês *Input/Output*, também conhecido pela sigla *I/O*, referem a comunicação entre um sistema de processamento de dados (como um computador) e o mundo exterior, seja ele um ser humano ou um outro sistema de processamento de dados. As entradas são os sinais ou os dados recebidos pelo sistema, e as saídas são os sinais ou os dados emitidos por ele.

Os dispositivos de *I/O* são usados por uma pessoa (ou por outro sistema) para comunicar-se com um computador. Por exemplo, teclados e mouses são dispositivos de entrada de um computador, enquanto o monitores e impressoras são considerados dispositivos de saída de um computador. Os dispositivos para comunicação entre computadores, tais como modem e placas de rede, são tipicamente para entrada e saída.

Em um computador a combinação do processador central e da memória central (isto é, a memória que o processador pode ler e escrever a diretamente) é considerada o cérebro de um computador. Qualquer transferência de informação entre esse bloco e qualquer outra unidade é considerada também *I/O*, como por exemplo, uma unidade de disco magnético ou ótico.

Abaixo vamos discutir a variedade de dispositivos usados para a entrada e a saída do computador.

1. Dispositivos de Entrada

1.1. Teclado

O teclado de computador é utilizado para incluir informações de texto no computador, como se faz na datilografia. O teclado pode também ser usado para inserir os comandos que controlam o computador para executar determinadas ações. Os comandos são escolhidos tipicamente de um menu na tela usando um mouse, mas geralmente há alguns atalhos de teclado para esses comandos.

Além das teclas do teclado principal (usado para datilografar o texto), os teclados geralmente têm um teclado numérico (para inserir dados numéricos eficientemente), um conjunto de teclas de edição (usadas para operações de edição do texto), e uma fila de chaves de função na parte superior (para chamar determinadas funções dos programas). Os computadores portáteis, que não têm espaço para teclados grandes, incluem frequentemente uma tecla “fn” de forma que outras teclas possam realizar duas funções (por exemplo, ter um teclado numérico encaixado dentro do teclado principal).

A maioria dos teclados padrão PC se conectam ao computador através de um conector PS/2 ou da porta do USB.

2. Dispositivos Apontadores

As interfaces gráficas (chamadas GUI) exigem algum tipo do dispositivo para posicionar o cursor na tela. Os dispositivos apontadores típicos são: mouse, almofada de toque e trackball. Os dispositivos apontadores, tais como mouse, se conectaram ao PC através de portas seriais (mais antigos), porta PS/2 ou a porta USB (os mais novos).

2.1. Mouse

O dispositivo apontador mouse apoia-se em sua superfície de trabalho, como uma mesa, e é movido com sua mão. Nos mouses mais velhos uma esfera na parte inferior rola na superfície enquanto você move o mouse, e os rolos internos detetam o movimento da esfera e transmitem a informação ao computador através do cabo. O nome *mouse* (rato em inglês) vem da semelhança do dispositivo em forma de concha com um cabo. A Figura 22 mostra um mouse tradicional.



Figura 22 - Mouse com fio (Clone)

O mouse óptico (mais recente) não usa uma esfera de rolamento, mas uma luz e um sensor óptico para detetar por reflexão o movimento do mouse seguindo a rugosidade da superfície da mesa. Os mouses óticos evitam o problema de uma esfera suja no mouse, que faz com que os mouses antigos percam a precisão se a esfera e os rolos internos não são limpos frequentemente.

Um mouse pode ser sem fio, comunicando-se com o computador através das ondas de rádio (geralmente usando o protocolo Bluetooth), tornando desnecessário o cabo para ligar ao computador (no entanto, ele vai precisar de baterias internas).

Um mouse inclui uma ou várias teclas (às vezes uma roda) para permitir que os usuários interajam com a interface gráfica. O mouse tradicional de PC tem duas teclas para acionar comandos, e a roda serve para deslizar a imagem na tela do computador.

2.2. Almofada de toque (touch pad)

A maioria de computadores portáteis têm hoje um dispositivo apontador tipo almofada de toque (*touch pad*). Você move o cursor na tela deslizando o dedo pela superfície da almofada plástica. As teclas ficam situadas abaixo da almofada, mas a maioria das almofadas de toque permitem que você execute o “clique do mouse” batendo com o dedo na própria almofada.

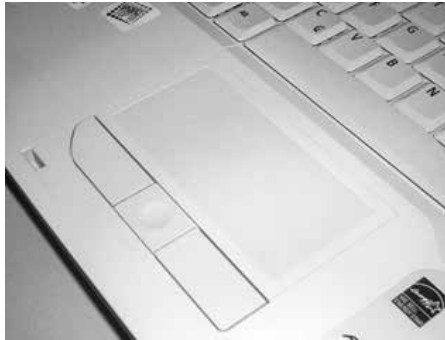


Figura 23 - Touchpad de notebook (Acer)

As almofadas de toque têm a vantagem sobre os mouses porque não tem partes mecânicas que se danifiquem e não pegam sujeira. A Figura 23 mostra uma almofada de toque.

2.3. Trackball

O *trackball* é como um mouse de cabeça para baixo, com a esfera situada na parte superior. Você usa seus dedos para rolar o trackball, e os rolos internos (similares ao de um mouse) detetam o movimento que é transmitido ao computador. Os *trackballs* têm a vantagem sobre mouses porque o corpo do trackball permanece estacionário em sua mesa, assim que você não precisa tanto espaço para usar o trackball.

3. Tablet gráfico

Um *tablet* gráfico consiste em uma área eletrônica de escrita e uma pena especial que “escreve” sobre ela. Os *tablets* gráficos permitem que artistas criem imagens gráficas com movimentos e as ações similares aos de ferramentas de desenho tradicionais. A pena do *tablet* gráfico é sensível a pressão, assim que pressionando obtemos um traço mais grosso ou mais fino.

4. Scanner

Um *scanner* é um dispositivo que lê imagens em uma página impressa em papel produzindo uma imagem feita de pixels minúsculos dos valores diferentes de brilho e da cor que são representados numericamente e transmitidos ao computador digitalmente. Os scanners fazem a leitura de figuras, mas podem também fazer a leitura de páginas do texto que, através de *software* OCR (reconhecimento de caráter ótico) que vai identificar as formas individuais das letras e criar uma linha de texto da página em formato digital.

5. Microfone

Um microfone pode ser ligados a um computador através de uma entrada de som. O som digitalizado é convertido em números digitais que representam o som analógico original podendo ser armazenado no computador e reproduzido mais tarde.

6. Dispositivos de saída

6.1. Monitor CRT

O dispositivo de saída tradicional de um computador para um ser humano é um monitor CRT (tubo de raio de catódico). Assim como um aparelho de televisão, o monitor CRT contém um grande tubo de raio de catódico que projeta um feixe de elétron em direção à tela gerando uma imagem dos pontos coloridos no interior da tela. Os monitores do CRT são pesados e consomem mais energia que as telas LCD.

O tamanho de tela do monitor é medido diagonalmente em polegadas. Nem toda a área da tela pode ser útil para mostrar imagem, assim muitas vezes também é indicada a área visível. A resolução do monitor é o número máximo de pixels que pode mostrar horizontalmente e verticalmente (como 800 x 600, ou 1024 x 768, ou 1600 x 1200).

A maioria de monitores podem indicar diversas resoluções abaixo de seu valor máximo. Quanto maior a quantidade de pixels possíveis maior é a qualidade do monitor. Os pixels são os pontos pequenos que fazem a imagem na tela. O afastamento dos pontos minúsculos do fósforo da tela é chamado distância entre pontos, também conhecido como dot pitch (dp). Uma tela com distância entre pontos menor produz umas imagens mais definida.

6.2. Monitor de tela plana LCD

Um monitor de tela plana usa geralmente uma tela do LCD (tela de cristal líquido) para indicar a saída do computador. O LCD consiste em diversas camadas finas que polarizam a luz que passa por elas. A polarização de uma camada, contendo as moléculas de cristais líquidos, pode ser controlada eletronicamente em cada pixel, obstruindo quantidades variadas de luz fazendo um pixel mais claro ou mais escuro. Existem outros tipos de tecnologia do tela plana como as telas de plasma mas os LCDs são os mais comuns. A figura 24 mostra um monitor LCD.



Figura 24 - Monitor LCD Samsung

Como no CRT, o tamanho de uma tela plana é expressa pelo tamanho diagonal em polegadas, e a resolução é definida pelo número de pixels horizontais e verticais da tela.

Os LCDs mais antigos tinham tempos de resposta lentos e baixo contraste, mas as telas LCD de matriz ativa têm um transistor de película fina transparente (conhecido pela sigla TFT) controlando cada pixel, assim o tempo de resposta, o contraste, e o ângulo de visão melhoram muito.

As telas planas são menos volumosas e consomem muito menos energia que monitores CRT. Ainda são mais caros do que os CRT mas a diferença de preço está cada vez menor, tornando-se cada vez mais comuns.

6.3. Impressoras Jato de tinta

Para fazer uma cópia definitiva de um trabalho, você precisa algum tipo da impressora ligada ao seu computador. O tipo o mais comum de impressora é do tipo jato de tinta colorida. Estas impressoras dão forma à imagem na página pulverizando gotas minúsculas da tinta a partir de uma cabeça de impressão na folha de papel. A impressora precisa diversas cores de tintas básicas (ciano, amarelo, magenta, e preto) de fazer imagens coloridas.

As impressoras jato de tinta são baratas, mas o custo dos materiais de consumo (cartuchos de tinta e papel especial) podem tornar seu custo relativamente alto.

6.4. Impressora laser

Uma impressora laser produz imagens de boa qualidade usando a mesma tecnologia que as fotocopiadoras usam. Um cilindro revestido com o material fotossensível é carregado, a seguir uma imagem é desenhada nela por um laser (ou por diodo emissor de luz) que faz aquelas áreas perderem a carga. O cilindro rola então através do tonalizador (partículas minúsculas de pigmento) que é atraído apenas pelas áreas carregadas do cilindro. O tonalizador é depositado no papel e fundido através de calor.

A maioria de impressoras laser são monocromáticas (somente uma cor, geralmente preto), mas algumas impressoras laser possuem cartuchos de tonalizador de várias cores podendo produzir impressões coloridas.

As impressoras laser são mais rápidas do que as impressoras jato de tinta. Sua velocidade é avaliada em páginas por o minuto (ppm). As impressoras de laser são mais caras do que as jato de tinta, mas tem um custo operacional mais baixo.

6.5. Impressoras matriciais

As impressoras matriciais usam pequenos pinos eletromagnéticos ativados na cabeça de impressão, em uma fita entintada para produzir imagens pelo impacto. Essas impressoras são lentas e ruidosas e atualmente não são muito usadas em computadores pessoais, no entanto tem o menor custo operacional e podem imprimir os formulários carbonados (várias cópias), coisa que nem as impressoras jato de tinta nem laser podem fazer.

6.6. Saída de som

Os computadores igualmente produzem uma saída de som, variando dos sinais acústicos simples que alertam o usuário, aos efeitos sonoros de jogos, à música com qualidade de concerto. Os circuitos para produzir o som geralmente estão localizados na placa mãe mas podem ser instalados em cartão de expansão. Para ouvir o som é necessário um conjunto de alto-falantes externos ou de fones de ouvido.

7. Dispositivos de Entrada e Saída

7.1. Tela sensível ao toque

Alguns computadores, especialmente os PDA e telefones celulares, têm telas sensíveis ao toque. O usuário pode fazer escolhas e pressionar imagens da tecla na tela. Também é comum usar um estilete permitindo que se “escreva” na superfície de uma tela sensível ao toque. Uma tela sensível ao toque tem a função de uma tela plana e de um teclado, por isso consideramos um dispositivo de entrada/saída. A Figura 25 mostra uma tela sensível ao toque.



Figura 25 - Tela sensível ao toque (Aramedia)

7.2. Modem

Modem vem do acrônimo Modulador/Demodulador e consiste em uma interface que serializa os dados e transmite através de uma linha telefônica. Um modem é usado tanto para transmitir como para receber, e caso a linha telefônica seja de boa qualidade pode alcançar distâncias bastante grandes. A velocidade máxima é 56 Kbps, e cada vez mais tem sido substituído por dispositivos de alta velocidade.

7.3. Interface Ethernet

A interface Ethernet é utilizada para ligar o computador a outros computadores, sejam eles locais (rede local) ou longa distância (Internet). O protocolo Ethernet cada vez mais tem sido usado como interface padrão dos computadores PCs. Uma interface Ethernet funciona em 10, 100 MBPS, 1 e 10 GBPS de modo auto adaptável, isto é, uma interface tenta se conectar na velocidade mais alta e caso não seja possível vai baixando a velocidade até sincronizar com o outro equipamento e conseguir se comunicar. O fato principal de popularidade das interfaces Ethernet é a alta velocidade e o baixo custo.

Síntese do capítulo



Este capítulo apresentou a arquitetura de um computador genérico e a função dos diversos componentes que o compõe. Foi mostrado a arquitetura da Unidade Central de Processamento (CPU), seus respectivos componentes e suas funções. Foram mostrados os diversos tipos de memórias detalhando suas funções conforme suas características, além de mostrar suas aplicações na arquitetura proposta. Finalmente, foram mostrados os diversos dispositivos de entrada e saída que facilitam a utilização do computador pelos seres humanos.

Atividades de avaliação



1. Qual o dispositivo de entrada e saída de maior velocidade? Por que ele pode ter essa velocidade?
2. Para melhorar a velocidade na comunicação entre computador e dispositivos de entrada/saída usa-se um mecanismo chamado DMA (*Direct Memory Access*). Pesquise e explique suas características e funcionamento.
3. Comente as vantagens e desvantagens de se transmitir dados em paralelo ou serial entre dispositivos de entrada e saída e computador.

Leituras, filmes e sites



Introdução à Arquitetura de Computadores

http://pt.wikibooks.org/wiki/Introdução_à_Arquitetura_de_Computadores

Introdução à Ciência da Computação

http://pt.wikiversity.org/wiki/Introdução_à_Ciência_da_Computação

Museu do Computador

<http://www.museudocomputador.com.br/>

O Computador

<http://pt.wikipedia.org/wiki/Computador>

Referências



FERNANDO DE CASTRO VELLOSO. **Informática: Conceitos Básicos**. 7ª Ed. Editora: Campus, 2004.

JOHN L. HENNESSY, DAVID A. PATTERSON. **Arquitetura de Computadores: Uma Abordagem Quantitativa**. 4ª Ed. Editora: Campus, 2008.

WILLIAN STALLINGS. **Arquitetura e Organização de Computadores**. 5ª Ed. Editora: Pearson, 2002.

Parte

4

Conjunto de instruções e endereçamento

Introdução

O termo *software* faz parte cada vez mais do vocabulário das pessoas com os mais diversos interesses, seja no contexto do profissional (*software* de escritório), ou de entretenimento (*software* de jogos). De forma geral, *software* é o conjunto de produtos que inclui principalmente programas de computador, geralmente acompanhados de manuais, especificações e outros documentos.

Por outro lado, o conjunto de instruções que compõem o programa a ser seguido e executado pelo computador, é também chamado de *software*, assim como o comportamento observado a partir da sua execução.

De forma geral, as instruções que compõem os programas são escritas utilizando uma linguagem de programação de alto nível de abstração. Uma linguagem classificada como sendo de alto nível indica que possui características que a torna mais próxima da linguagem natural humana. Em contrapartida, uma linguagem dita de baixo nível, é mais próxima da linguagem compreendida pelo *hardware*. Como exemplo de linguagens de programação de alto nível podemos citar: Pascal, Fortran, C.

As instruções de alto nível precisam ser traduzidas ou interpretadas, para posteriormente serem executadas por um *processador* ou *máquina virtual*. Para que o processador possa interpretar e executar um programa, seu conjunto de instruções precisa ser escrito em *linguagem de máquina*²⁷ ou *assembly language*. O resultado deste processo é o programa em linguagem de montagem ou linguagem de máquina. Cada arquitetura possui uma linguagem de montagem particular.

Finalmente, o programa em linguagem de montagem é convertido para um programa em código objeto pelo montador (*assembler*). O montador traduz diretamente uma instrução da forma textual para a forma de código binário. Neste formato final a instrução é carregada na memória e interpretada pelo processador.

²⁷ Linguagem de Máquina: representação numérica utilizada diretamente pelo processador

Saiba Mais



A linguagem *Assembly* é uma linguagem de baixo nível para programação de computadores. Esta linguagem implementa uma representação simbólica de códigos de máquina numéricos e considera outras restrições para uma arquitetura de CPU particular. Esta representação usualmente é definida pelo fabricante de *hardware*, e é baseada em abreviaturas das palavras em inglês (mnemônicos), de forma a facilitar o trabalho dos programadores. A linguagem de máquina é específica a uma determinada arquitetura de computador física e virtual, diferentemente das linguagens de alto nível, as quais são portáteis.

Desde o ponto de vista do *software*, um computador pode ser estruturado como uma série (ou níveis) de abstrações, onde a série superior representa um incremento em relação à série que a precede. Em cada nível é utilizada uma linguagem com propriedades diferenciadas, onde cada linguagem usa a sua antecessora (nível imediatamente inferior), como base. A linguagem ou nível que fica mais embaixo é mais simples de forma a ser entendido pelo computador, enquanto que a que fica mais acima é a mais sofisticada e complexa, porém, mais próxima do entendimento humano.

O nível mais baixo é o nível lógico digital, composto por circuitos eletrônicos estudados no capítulo 2.

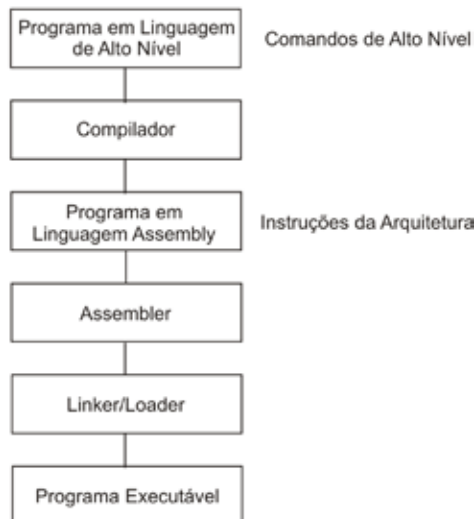


Figura 26 - O conjunto de instruções dentro do contexto de *software*.

Nesta unidade será abordado o nível de Arquitetura do Conjunto de Instruções (ISA). Este conjunto de instruções determina as instruções executadas que podem ser interpretadas por uma máquina específica. Cada máquina tem uma linguagem consistindo em todas as instruções que a máquina é capaz de executar.

Praticando

1. Que significa que uma linguagem de programação é classificada como sendo de “baixo ou alto” nível?
2. Quais linguagens de alto nível você conhece? De exemplos de instruções de uma linguagem de alto nível.
3. Um programa em uma linguagem de alto nível pode ser traduzido ou interpretado pelo processador. Pesquise qual é a diferença entre esses dois conceitos.

Instruções

De forma geral, uma única instrução em linguagem de alto nível é equivalente a várias instruções em linguagem de máquina. Por exemplo, observe a seguinte instrução de linguagem de alto nível em Pascal: $X = X + Y$, onde as variáveis X e Y correspondem às posições de memória 312 e 311. Esse comando alto nível pode ser implementado através do seguinte conjunto de instruções de máquina:

1. Carregar um registrador com o conteúdo da posição de memória 312.
2. Adicionar o conteúdo da posição de memória 311.
3. Armazenar o conteúdo do registrador na posição 312.

Em linguagem de máquina, as instruções são representadas através de mnemônicos, que vinculam o nome da instrução à sua função. Alguns exemplos comuns são:

- **ADD** Adição
- **SUB** Subtração
- **MPY** Multiplicação
- **DIV** Divisão
- **LOAD** Carregar dados na memória
- **STOR** Armazenar dados na memória

Tudo o que a máquina pode fazer é controlado pelo seu conjunto de instruções. As instruções permitem movimentar dados, realizar operações aritméticas e lógicas, e de comparação, a partir das quais é possível desviar a execução do programa segundo os resultados obtidos.

Cada instrução deve conter toda a informação necessária para que a CPU possa executá-la. Uma instrução de máquina normalmente é composta pelos seguintes elementos:

- **Código de operação:** código binário que especifica a operação a ser executada.

- **Referencia a operando fonte** que constitui o dado de entrada para a operação. Podem existir mais de um.
- **Referencia a operando de destino** indicando o destino do resultado eventualmente gerado pela operação.
- **Endereço da próxima instrução** indica onde a CPU deve procurar a próxima instrução a ser executada, que pode estar localizada na memória principal ou virtual. Este campo fica implícito no caso em que a próxima instrução segue imediatamente à atual.

Cada instrução é dividida em campos de acordo aos elementos que a compõem, e cada campo é representado por um conjunto de bits. A seguir é apresentado um formato de instrução simples.



Os passos envolvidos na execução de uma instrução são os seguintes:

1. Cálculo do endereço da instrução
2. Busca da instrução
3. Alterar o contador do programa para indicar a próxima instrução.
4. Decodificação da operação da instrução
5. Cálculo do endereço de operando
6. Busca de operando
7. Operação sobre os operandos
8. Armazenamento de resultado
9. Cálculo de endereço de operando

Os passos 4 e 5 podem se repetir no caso de múltiplos operandos. Da mesma forma os passos 7 e 8 para múltiplos resultados. A seqüência de passos recomeça na busca de uma nova instrução, uma vez que a anterior for completada.

Essa seqüência de etapas, denominada de ciclo **buscar-decodificar-executar**, é de fundamental importância para a operação de todos os computadores.

Para realizar este ciclo, a Unidade Lógica e Aritmética é auxiliada por registradores (normalmente de 1 a 32), onde os dados são armazenados. Os registradores alimentam os registradores de entrada da ALU, os quais vão ser utilizados para executar a operação. Efetuada a operação o resultado é

temporariamente armazenado no registrador de saída da ALU, para depois ser armazenado em um registrador. Este processo é chamado de **ciclo do caminho** de dados. Outros registradores importantes que contribuem neste ciclo são o Contador de Programa (PC), que indica a próxima instrução a ser executada. Uma vez recuperada, a instrução sendo atualmente executada é armazenada no Registrador de Instrução.

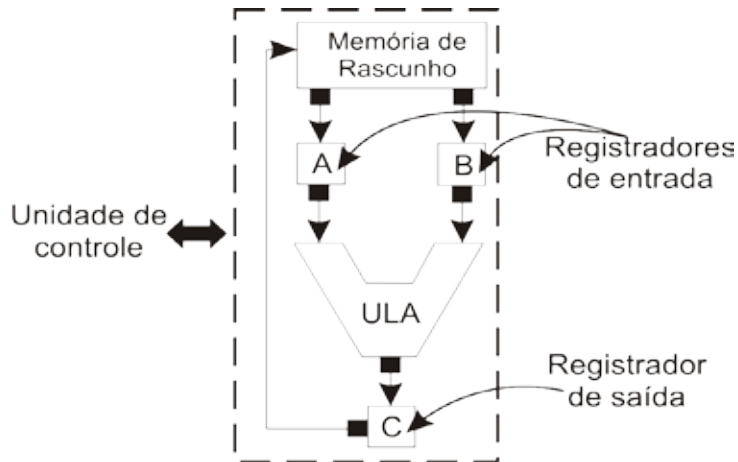


Figura 27 - Arquitetura básica de um computador.

Em algumas máquinas, todas as instruções tem o mesmo comprimento; em outras podem existir comprimentos diferentes. No primeiro caso, a decodificação fica mais fácil, mas pode acarretar desperdício de espaço.

Atividades de avaliação



1. Quais seriam as vantagens de um computador estruturado em níveis possuir mais ou menos níveis?
2. Explique os diferentes papéis dos registradores no contexto do ciclo de execução de uma instrução.
3. Considere o conjunto de passos seguidos para a execução de uma instrução. Qual é a finalidade do passo 3? O que aconteceria se este passo fosse omitido?
4. Considere as seguintes tarefas: (1) somar 100 números armazenados na memória principal e (2) somar um número a ele mesmo 100 vezes. Apesar de ambos os programas precisarem de 100 adições, o segundo poderá ser executado muito mais rapidamente do que o primeiro? Por que?

Instruction Set Architecture - ISA

Esse nível está posicionado entre o nível da microarquitetura²⁸ e o sistema operacional e faz às vezes de interface entre o *software* (compiladores) e o *hardware*, facilitando a portabilidade de programas escritos em diversas linguagens entre diferentes computadores.

A estratégia utilizada pelos projetistas para reduzir esse gap semântico é criar uma representação intermediária – o nível ISA – e construir *hardware* que possa executar programas de nível ISA diretamente. Desta forma, a ISA é projetada de forma a tornar compatíveis as necessidades de ambas as camadas.

Alem disso, desde o ponto de vista do cliente, é de suma importância que a ISA seja compatível entre os diversos modelos de computadores, de forma a suportar a portabilidade dos programas antigos para uma nova arquitetura, sem que sejam necessárias alterações. Desta forma surge o conceito de arquitetura para uma família de computadores, os quais compartilham um conjunto de propriedades e características em comum.

O conjunto de instruções do processador é uma questão central para a definição e implementação da sua arquitetura (por exemplo: operações realizadas pela unidade lógica e aritmética, número e função dos registradores, etc.).

O conjunto de instruções é caracterizado pelos tipos de instruções e operandos, número e localização dos operandos em instruções aritméticas e lógicas, modos de endereçamento para acesso aos dados na memória, e o formato dos códigos de instrução.

Consequentemente existe uma dependência direta entre o projeto do conjunto de instruções e o projeto da arquitetura do computador, e vice versa. Analogamente, os tipos de operando que podem ser diretamente manipulados por uma arquitetura dependem, é claro, dos tipos de instruções oferecidas.

As instruções oferecidas por uma arquitetura podem ser classificadas em três categorias diferentes, de acordo com o tipo de operação que realizam. Uma arquitetura de uso geral fornece pelo menos três categorias de instruções básicas a seguir.

²⁸ Nível da Microarquitetura: composto normalmente de 8 a 32 registradores que compõem uma memória local, e um circuito capaz de executar operações aritméticas simples (ALU).

- **Instruções aritméticas e lógicas:** são as instruções que realizam operações aritméticas sobre números inteiros (adição, subtração) e operações lógicas bit-a-bit (AND, OR);
- **Instruções de movimentação de dados:** são instruções que transferem dados entre os registradores ou entre os registradores e a memória principal; E/S.
- **Instruções de transferência de controle:** são instruções de desvio e de chamada de rotina, que transferem o controle da execução para uma instrução específica dentro do código do programa.
- **Instruções de armazenamento de dados:** são instruções de memória.

Arquiteturas destinadas para uma aplicação específica ou domínio específico podem oferecer outros tipos de instruções, especializadas para aquela aplicação. A Unidade 2 apresenta os principais tipos de dados normalmente representados em uma arquitetura de uso geral.

1. Numero de operandos e localização

Uma instrução, seja aritmética ou lógica, é composta por um identificador de operação e um conjunto de operandos referenciados. Por exemplo, operações aritméticas e lógicas são unárias ou binárias. Considerando que o resultado da operação precisa ser armazenado, se faz necessário um terceiro operando. De forma geral, o endereço da próxima instrução é implícito (contido no contador do programa).

Exemplo: computar o comando $Y = (A - B) \div (C + D \times E)$, utilizando instruções típicas de um, dois, e três endereços.

Instruções com três endereços		
	Instrução	Comentário
SUB	Y, A, B	$Y \leftarrow A - B$
MPY	T, D, E	$T \leftarrow D \times E$
ADD	T, T, C	$T \leftarrow T + C$
DIV	Y, Y, T	$Y \leftarrow Y \div T$

Instruções com dois endereços		
Instrução		Comentário
MOVE	Y, A	$Y \leftarrow A$
SUB	Y, B	$Y \leftarrow T - B$
MOVE	T, D	$T \leftarrow D$
MPY	T, E	$T \leftarrow T \times E$
ADD	T, C	$T \leftarrow T + C$
DIV	Y, T	$Y \leftarrow Y \div T$

Instruções com um endereço		
Instrução		Comentário
LOAD	D	$AC \leftarrow D$
MPY	E	$AC \leftarrow AC \times E$
ADD	C	$AC \leftarrow AC + C$
STOR	Y	$Y \leftarrow AC$
LOAD	A	$AC \leftarrow A$
SUB	B	$AC \leftarrow AC - B$
DIV	Y	$AC \leftarrow AC \div Y$
STOR	Y	$Y \leftarrow AC$

As instruções com três operandos não são muito comuns por conta do tamanho relativamente grande das instruções. No caso de instruções com dois operandos e operações binárias, um dos endereços referencia um operando e o resultado, ou seja, que o resultado é armazenado no lugar do próprio operando. Um cuidado especial tem que ser tomado para não alterar o valor de algum operando utilizando a instrução MOVE.

Embora o tamanho das instruções é reduzido, o número de instruções requeridas aumenta de 4 para 6. Finalmente, no caso de instruções com um operando o segundo operando fica implícito, utilizando-se um registrador da CPU (acumulador – AC) para auxiliar na resolução da operação. Neste caso são necessárias oito instruções para realizar a operação.

O número de endereços por instrução resulta em uma CPU menos complexa, uma vez que as instruções têm uma extensão menor e são mais primitivas. Em contrapartida, a quantidade de instruções por programa é maior o que geralmente resulta em programas mais complexos e maior tempo de execução.

O número de endereços também implica na utilização de múltiplos registradores, tornando a execução de operações mais rápidas. Desta forma, a definição do número de endereços por instrução constitui uma decisão de projeto importante.

Praticando

1. Analise o exemplo apresentado e descreva em português a seqüência de instruções de máquina utilizadas em cada caso.
2. Computar o seguinte comando utilizando instruções de um, dois e três operandos:

$$Z = (C \times B) \div (A \times E - D)$$

A partir do número de operandos explicitamente referenciados e da localização destes operandos, é possível classificar as arquiteturas nos seguintes tipos:

- **Arquiteturas memória-memória:** as instruções aritméticas/lógicas usam três operandos e todos os operandos podem estar na memória;
- **Arquiteturas registrador-memória:** as instruções aritméticas/lógicas usam dois operandos, sendo que apenas um deles pode residir na memória;
- **Arquiteturas registrador-registrador:** as instruções aritméticas/lógicas usam três operandos, todos em registradores²⁹. Neste caso, apenas duas instruções acessam diretamente a memória: LOAD e STORE. A instrução LOAD carrega em um registrador um dado armazenado na memória e instrução STORE armazena na memória o conteúdo de um registrador.

As arquiteturas também podem ser classificadas em relação à localização dos operandos:

- **Arquiteturas memória-memória:** as instruções aritméticas/lógicas usam três operandos e todos os operandos podem estar na memória;
- **Arquiteturas registrador-memória:** as instruções aritméticas/lógicas usam dois operandos, sendo que apenas um deles pode residir na memória;
- **Arquiteturas registrador-registrador:** as instruções aritméticas/lógicas usam três operandos, todos em registradores. Neste caso, apenas duas instruções acessam diretamente a memória: LOAD e STORE. A instrução LOAD carrega em um registrador um dado armazenado na memória e instrução STORE armazena na memória o conteúdo de um registrador.

²⁹ Registradores: Os registradores são memórias utilizadas para realizar as operações na CPU. Essas memórias são muito rápidas e de tamanho pequeno, geralmente suficiente para guardar menos de uma dezena de valores.

Atividades de avaliação



1. As linguagens de máquina são específicas de uma dada máquina. Que significa que cada tipo de computador tem sua própria linguagem de máquina? Explique por que isto acontece.

Modos de endereçamento

Grande parte das instruções tem operandos, os quais devem ser localizados para realizar a operação. Os operandos podem encontrar-se em registradores, na memória principal ou embutidos implicitamente na própria instrução. O modo de endereçamento determina a forma em que uma instrução especifica a localização dos seus operandos. Existem três modos de endereçamento básicos:

- **Modo registrador:** a instrução indica explicitamente o registrador de dados onde se encontra o operando (fonte ou destino);
- **Modo imediato:** o operando se encontra dentro do próprio código da instrução;
- **Modo implícito:** a localização do operando não é indicada explicitamente na instrução. Por exemplo, nas chamadas arquiteturas acumulador, um dos operandos-fonte e o operando-destino nas instruções aritméticas/lógicas encontra-se sempre em um registrador especial, o acumulador.

Existem diferentes arquiteturas que podem ser utilizadas para o endereçamento de dados armazenados na memória principal. Um modo de endereçamento referente à memória indica como deve ser obtido o endereço da locação de memória onde se encontra o dado que será acessado. Este endereço é chamado endereço efetivo.

Apesar da variedade mencionada, é possível identificar alguns modos de endereçamento referentes à memória que são oferecidos pela maioria das arquiteturas. Por outro lado, quase todas as arquiteturas fornecem mais de um modo de endereçamento. Todos estes modos envolvem decisões em relação a quantidade de posições de memória endereçáveis e/ou flexibilidade de endereçamento e/ou complexidade do cálculo. Estes modos de endereçamento mais comuns estão relacionados na tabela a seguir.

A forma mais simples de endereçamento é o **endereçamento imediato** onde o valor do operando é especificado diretamente na instrução, sem necessidade de acesso à memória.

OPERANDO = A

Este modo geralmente é utilizado na definição e uso de constantes e inicialização de variáveis. Este modo tem como desvantagem que o tamanho do operando fica limitado pelo tamanho do campo endereço na instrução.

No modo **endereçamento direto**, o endereço efetivo é um valor imediato contido no campo endereço da instrução. Por exemplo, na instrução

ADD (100), R1

um dos operandos encontra-se na locação de memória com endereço 100. O significado da instrução acima é equivalente a:

$$R1 \leftarrow M [100] + R1$$

Este modo de endereçamento é usado principalmente no acesso às variáveis estáticas de um programa, cujo endereço em memória pode ser determinado durante a compilação do programa em um único acesso a memória. Por outro lado a faixa de endereços é limitada por conta do tamanho do campo.

No **modo indireto via registrador**, o endereço efetivo encontra-se em um registrador. Por exemplo, na instrução:

ADD (R1), R2

Um dos operandos encontra-se na locação de memória cujo endereço se encontra no registrador R1. Ou seja, existe uma indireção para chegar ao endereço efetivo do operando na memória, através de um registrador que contém essa informação. A instrução acima é equivalente à seguinte expressão:

$$R2 \leftarrow M [R1] + R2$$

Este modo de endereçamento é usado no acesso a variáveis dinâmicas, cujo endereço na memória é conhecido apenas durante a execução do programa. O acesso a uma variável dinâmica é realizado através de um ponteiro, que nada mais é do que o endereço da variável. Para realizar o acesso à variável dinâmica, o ponteiro é carregado em um registrador, e a instrução que acessa a variável usa este registrador com o modo de endereçamento indireto.

No **modo relativo à base**, o endereço efetivo é obtido a partir da soma do conteúdo de um registrador, chamado endereço-base, com um valor imediato contido na instrução, chamado deslocamento. Este modo de endereça-

mento é bastante poderoso e combina as capacidades do endereçamento direto e do indireto via registrador. Por exemplo, na instrução:

ADD 100(R1), R2

O registrador R1 contém o endereço-base e 100 é o deslocamento. O endereço efetivo do operando na memória é calculado a partir da soma do conteúdo de R1 com o valor 100.

No caso da instrução acima, ela pode ser expressa como:

$R2 \leftarrow M[100 + R1] + R2$

O modo relativo à base é usado no acesso a componentes de variáveis dinâmicas estruturadas (por exemplo, *record* em Pascal ou *struct* em C). A figura 28 a seguir mostra como é realizado o cálculo do endereço efetivo no modo de endereçamento relativo à base.

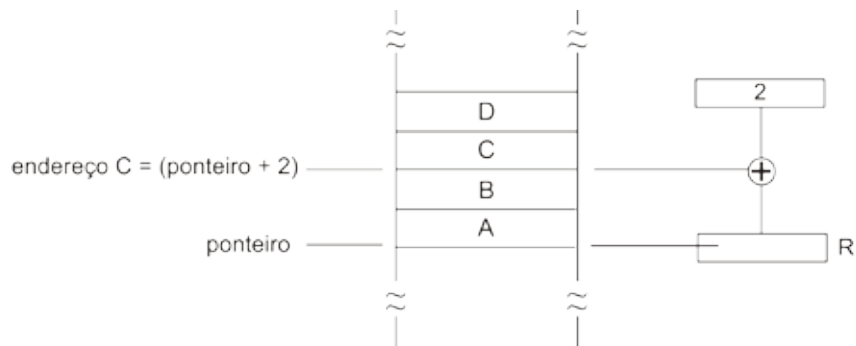


Figura 28 - Modelo de realização do cálculo do endereço

A figura mostra a localização na memória de uma estrutura com quatro campos A, B, C e D. O endereço inicial da estrutura é indicado por um ponteiro, que torna-se conhecido apenas durante a execução do programa. No entanto, a posição de cada campo em relação ao início da estrutura é fixo, sendo conhecido durante a compilação. O endereço de um campo é obtido somando-se a posição do campo (o deslocamento) ao ponteiro que indica o início da estrutura (o endereço-base). Por exemplo, na figura, para somar um valor ao campo C, o compilador pode usar a instrução:

ADD 2(R1), R2

Precedida de uma instrução para carregar em R1 o endereço-base da estrutura.

No **modo indexado**, o endereço efetivo é calculado somando dois registradores mais um deslocamento (opcional). Um registrador é a base e o outro o índice. Por exemplo, na instrução:

ADD (R1 + R2), R3

R1 contém o endereço-base, e R2 o índice. A interpretação para este modo de endereçamento é o seguinte:

$$R3 \leftarrow M [R1 + R2] + R3$$

O modo indexado é normalmente usado eficientemente para implementar operações iterativas, como por exemplo, no acesso aos elementos de um vetor. A figura embaixo mostra como é calculado o endereço efetivo no modo de endereçamento indexado.

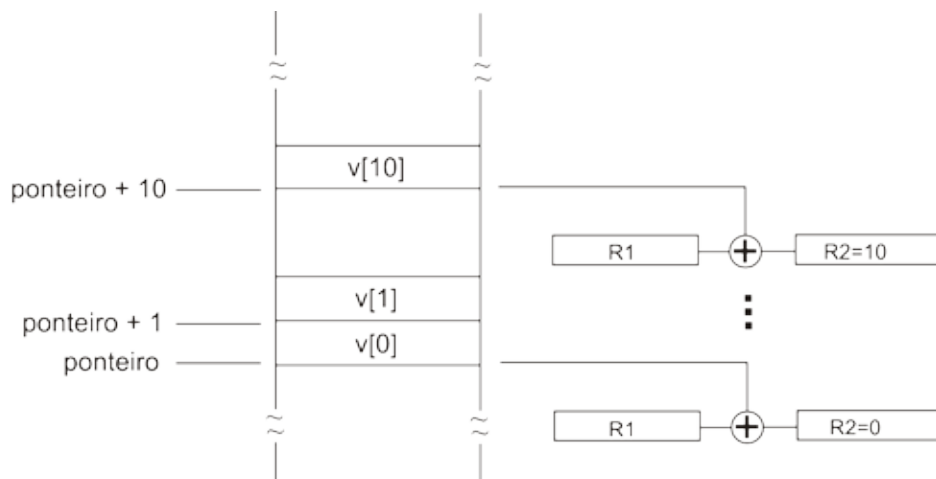


Figura 29 - Modelo de realização de cálculo do endereço efetivo

Na figura é representada a localização de um vetor V na memória. Um ponteiro indica o endereço-base do vetor indicando a posição do primeiro elemento. Considerando que o vetor é uma estrutura homogênea, cada elemento do vetor possui o mesmo tamanho e estrutura. Desta forma é possível calcular a posição na memória do próximo elemento do vetor somando o índice do elemento ao endereço-base.

Para realizar o acesso seqüencialmente os elementos do vetor, o índice é inicialmente carregado no registrador com o valor 0. O índice é então incrementado dentro de um laço após o acesso a cada elemento. Por exemplo, para somar um valor em registrador aos elementos do vetor, o compilador pode usar as seguintes instruções em um laço:

ADD R1, (R2 + R3)

ADD 1, R3

Onde R1 contém o valor a ser somado, R2 contém o ponteiro para o vetor e R3 é o registrador com o índice, com valor inicial 0.

Atividades de avaliação



1. Quais são os modos de endereçamento mais comuns? Apresente um exemplo de uso para cada caso (diferente do apresentado no texto).
2. Por que é dito que o modo de endereçamento relativo à base combina as capacidades do endereçamento direto e do indireto via registrador? Explique.
3. O processo de decodificação da instrução é crítico. Como você justificaria esta afirmação?

CISC versus RISC

Os primeiros computadores eram baseados em conjuntos de instruções simples e pequenos. No entanto, a procura por maior desempenho fez com que as instruções se tornassem mais complexas e poderosas.

Com o objetivo de diminuir a grande lacuna semântica entre as necessidades das pessoas expressas em linguagens de alto nível e o que o computador pode fazer, os projetistas de computadores investiram na definição de instruções complexas.

Na medida em que as instruções de máquina ficaram mais próximas das instruções de linguagens de alto nível, aparentemente a tarefa dos compiladores ficou simplificada. No entanto, a utilização de um conjunto de instruções mais rico requer de uma unidade de controle mais complexa capaz de interpretar³⁰ e executar as instruções, aumentando conseqüentemente o tempo de execução de cada instrução individual. Por outro lado, outras atividades do compilador tais como otimização do código gerado pode ser dificultada. Um exemplo de arquitetura CISC é *Pentium* da Intel.

Desta forma, considerando que os benefícios na utilização de conjuntos de instruções complexos não se apresentaram muito claros, novos estudos sobre o comportamento da execução de programas em linguagens de alto nível orientaram o projeto de um novo tipo de arquitetura: o computador com um conjunto reduzido de instruções (RISC).

Arquiteturas RISC compartilham as seguintes características:

- Conjunto limitado de instruções simples com formato fixo, facilitando a decodificação e acesso aos operandos.
- Uma instrução por ciclo pode ser implementada diretamente pelo hardware, reduzindo o tempo de execução (não precisavam de interpretação).
- Grande número de registradores ou o uso de um compilador que otimize o uso dos mesmos.
- Operações de registrador para registrador.

³⁰ Interpretar: no processo de interpretação, cada instrução é examinada e decodificada individualmente, para ser imediatamente executada. Diferentemente do que na tradução, nenhum programa traduzido é gerado.

- Método de endereçamento simples de registrador.
- Enfoque na otimização do uso da pipeline de instruções.

Exemplos de computadores com arquitetura RISC são: **Sun** e **UltraSparc**.

Esses novos processadores tinham diferenças significativas em relação aos processadores existentes. Esta incompatibilidade se tornou favorável para o desenvolvimento de um novo conjunto de instruções de forma a otimizar o desempenho do sistema. Os processadores denominados RISC possuíam um conjunto de instruções em torno de 50, enquanto que nos computadores CISC existentes na época, o conjunto de instruções era entre 200 e 300.

A estratégia por trás do funcionamento de processadores RISC é que embora fosse necessário de 4 a 5 instruções simples para fazer o equivalente a uma instrução CISC (*Complex Instruction Set Computer*), se as instruções RISC fossem suficientemente rápidas o desempenho global do sistema seria maior, uma vez que estas instruções não eram interpretadas.

Houve dois principais motivos que impediram a predominância das máquinas RISC, mesmo com um desempenho maior:

1. O grande investimento das organizações em tecnologia Intel, incompatível com os novos processadores.
2. A Intel desenvolveu uma arquitetura híbrida combinando o melhor das duas abordagens: o núcleo de instruções mais comuns funcionam de modo RISC, enquanto as instruções mais complicadas e menos frequentes, no modo CISC. Isto foi implementado a partir da 486.

A solução da Intel tornou os computadores competitivos a nível de desempenho global, e ainda manteve a compatibilidade com *softwares* antigos.

1. Princípios de projeto para computadores modernos

Dado o estado atual da tecnologia de *hardware*, alguns princípios de projeto para computadores (CPUs de uso geral), passaram a ser aceitos. Muitos desses princípios têm sua origem no projeto de arquitetura RISC:

- Todas as instruções são executadas diretamente pelo *hardware* eliminando desta forma um nível de interpretação. Este princípio propicia maior velocidade na execução das instruções.
- De forma a minimizar o problema do gargalo no acesso à memória e aumentar a velocidade de execução das instruções, a técnica de busca antecipada ou *prefetch* consiste na busca antecipada de instruções da memória, antes mesmo de ser utilizada, de modo a encontrá-las no processador quando chegar o momento de executá-las.

Tais instruções eram armazenadas em um conjunto de registradores chamados de *buffer* de pré-busca. Deste modo o processo de busca pode ser sobreposto à execução normal.

- Maximização da taxa de execução das instruções. O paralelismo³¹ na execução das instruções pode representar uma grande melhoria no desempenho.
- Instruções fáceis de decodificar. Formatos regulares, de comprimento fixo e com poucos campos facilita a decodificação, aumentando a taxa de emissão de instruções.
- Referencia a memória somente através de LOAD e STORE. Uma vez que o acesso a memória tem um alto custo em tempo, deve ser priorizado o uso de registradores para o armazenamento de operandos durante a execução das instruções.
- Utilização de vários registradores (no mínimo 32) de forma a evitar ao máximo o processo de descarregamento de memória.

³¹. Paralelismo: o paralelismo tem duas formas gerais: paralelismo no nível de instrução e paralelismo no nível do processador. No primeiro caso, uma única CPU é capaz de executar um maior número de instruções por segundo. No segundo caso, várias CPUs trabalham juntas no mesmo problema.

Síntese do capítulo



Neste capítulo foi apresentado o conceito central de instrução, suas características, componentes, tipos e modos de endereçamento. Foi apresentada a noção de máquina multinível, e a forma de funcionamento.

Com base no conceito de instrução e linguagem de máquina, foi discutida a importância da ISA -Arquitetura do Conjunto de Instruções, no contexto do projeto da arquitetura do computador, sua influência e restrições. Foram analisadas as propriedades das duas principais arquiteturas: RISC e CISC, fazendo uma análise de vantagens e desvantagens de cada uma delas, as quais serviram de base para a elicitação dos princípios do projeto dos computadores modernos.

Atividades de avaliação



1. Especifique as características das arquiteturas RISC e CISC fazendo um comparativo de vantagens e desvantagens de cada uma delas.
2. Qual é a causa principal do alto desempenho da arquitetura RISC?

3. Por que, mesmo atingindo um desempenho melhor, a arquitetura RISC não atingiu as expectativas do mercado?
4. Para que serve o buffer de pré-busca?
5. Qual é a maior vantagem de usar paralelismo?
6. De forma a ilustrar as abordagens de paralelismo, pesquise os conceitos de *Pipelining* (paralelismo no nível de instrução) e Computadores Matriciais (paralelismo no nível de processador).

Leituras, filmes e sites



Intel Corporation. <http://www.intel.com/>

Processadores Intel <http://www.intel.com/portugues/products/processor/index.htm>

Sparc International <http://www.sparc.org/>

Sun Microsystems Brasil <http://br.sun.com/>

Referências



TANENBAUM, ANDREW S. **Organização Estruturada de Computadores**. 5 ta. Edição. Editora Pearson. 2007.

STALLINGS WILLIAM. **Estrutura e Organização de Computadores**. 5 ta. Edição. Editora Pearson. 2004.

BROOKSHEAR GLENN J. **Ciência da Computação – Uma visão abrangente**. 5 ta. Edição. Editora Bookman. 2000.

Sobre os autores

Marcial Porto Fernandez: possui graduação em Engenharia Eletrônica e Computação pela Universidade Federal do Rio de Janeiro (1988), mestrado em Engenharia Elétrica pela Universidade Federal do Rio de Janeiro (1998) e doutorado em Engenharia Elétrica pela Universidade Federal do Rio de Janeiro (2002). Atualmente é professor adjunto da Universidade Estadual do Ceará e assessor técnico da Empresa de Tecnologia da Informação do Ceará. Tem experiência na área de Ciência da Computação, com ênfase em Teleinformática, atuando principalmente nos seguintes temas: gerenciamento de redes, redes óticas, redes móveis e sem fio e arquitetura de computadores.

Mariela Inés Cortés: atualmente é Professora Adjunta de da Universidade Estadual do Ceará (UECE). Possui doutorado em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio, 2003), mestrado em Sistemas e Computação pelo Instituto Militar de Engenharia (IME, 1998) e graduação pela Universidad Nacional de La Plata, Argentina - Analista de Computación (1996). Tem experiência na área de Ciência da Computação, com ênfase em Evolução de software e gerência de projetos de software, atuando principalmente nos seguintes temas: evolução de *framework*, *refactorings*, *agent societies* e *multi-agent systems*, gerencia de projetos de desenvolvimento de software.



Computação

Fiel a sua missão de interiorizar o ensino superior no estado Ceará, a UECE, como uma instituição que participa do Sistema Universidade Aberta do Brasil, vem ampliando a oferta de cursos de graduação e pós-graduação na modalidade de educação a distância, e gerando experiências e possibilidades inovadoras com uso das novas plataformas tecnológicas decorrentes da popularização da internet, funcionamento do cinturão digital e massificação dos computadores pessoais.

Comprometida com a formação de professores em todos os níveis e a qualificação dos servidores públicos para bem servir ao Estado, os cursos da UAB/UECE atendem aos padrões de qualidade estabelecidos pelos normativos legais do Governo Federal e se articulam com as demandas de desenvolvimento das regiões do Ceará.





Computação

Introdução á Computação

Marcial Porto Fernandez
Mariela Inés Cortés

3ª edição
Fortaleza - Ceará



2015



Química



Biologia



Artes



Computação



Física



Matemática



Pedagogia

Copyright © 2015. Todos os direitos reservados desta edição à UAB/UECE. Nenhuma parte deste material poderá ser reproduzida, transmitida e gravada, por qualquer meio eletrônico, por fotocópia e outros, sem a prévia autorização, por escrito, dos autores.

Editora Filiada à



Presidenta da República

Dilma Vana Rousseff

Ministro da Educação

Renato Janine Ribeiro

Presidente da CAPES

Carlos Afonso Nobre

Diretor de Educação a Distância da CAPES

Jean Marc Georges Mutzig

Governador do Estado do Ceará

Camilo Sobreira de Santana

Reitor da Universidade Estadual do Ceará

José Jackson Coelho Sampaio

Vice-Reitor

Hidelbrando dos Santos Soares

Pró-Reitora de Graduação

Marcília Chagas Barreto

Coordenador da SATE e UAB/UECE

Francisco Fábio Castelo Branco

Coordenadora Adjunta UAB/UECE

Eloisa Maia Vidal

Diretor do CCT/UECE

Luciano Moura Cavalcante

Coordenador da Licenciatura em Informática

Francisco Assis Amaral Bastos

Coordenadora de Tutoria e Docência em Informática

Maria Wilda Fernandes

Editor da EdUECE

Erasmo Miessa Ruiz

Coordenadora Editorial

Rocylânia Isídio de Oliveira

Projeto Gráfico e Capa

Roberto Santos

Diagramador

Francisco José da Silva Saraiva

Conselho Editorial

Antônio Luciano Pontes

Eduardo Diatahy Bezerra de Menezes

Emanuel Ângelo da Rocha Fragoso

Francisco Horácio da Silva Frota

Francisco José Camelo Parente

Gisafran Nazareno Mota Jucá

José Ferreira Nunes

Liduina Farias Almeida da Costa

Lucili Grangeiro Cortez

Luiz Cruz Lima

Manfredo Ramos

Marcelo Gurgel Carlos da Silva

Marcony Silva Cunha

Maria do Socorro Ferreira Osterne

Maria Salete Bessa Jorge

Silvia Maria Nóbrega-Therrien

Conselho Consultivo

Antônio Torres Montenegro (UFPE)

Eliane P. Zamith Brito (FGV)

Homero Santiago (USP)

Ieda Maria Alves (USP)

Manuel Domingos Neto (UFF)

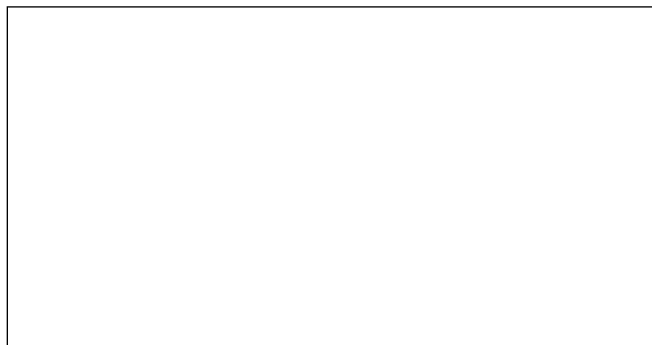
Maria do Socorro Silva Aragão (UFC)

Maria Lírída Callou de Araújo e Mendonça (UNIFOR)

Pierre Salama (Universidade de Paris VIII)

Romeu Gomes (FIOCRUZ)

Túlio Batista Franco (UFF)



Editora da Universidade Estadual do Ceará – EdUECE

Av. Dr. Silas Munguba, 1700 – Campus do Itaperi – Reitoria – Fortaleza – Ceará

CEP: 60714-903 – Fone: (85) 3101-9893

Internet: www.uece.br – E-mail: eduece@uece.br

Secretaria de Apoio às Tecnologias Educacionais

Fone: (85) 3101-9962

Sumário

Parte 1 - Histórico, aplicações e conceitos da Tecnologia da Informação	5
Capítulo 1 - História e evolução dos computadores	7
Introdução	7
1. Precursores do computador	8
2. As máquinas de primeira geração (1930-1958)	15
3. Computadores de segunda geração (1955-1965)	17
4. Computadores de terceira geração (1965-1980)	18
5. Computadores de quarta geração (1980 - ...)	19
Capítulo 2 - Tecnologias de computadores e suas aplicações	23
1. Computador descartável	24
2. Microcontrolador	25
3. Computador de jogos	27
4. Computador pessoal (PC)	28
5. Servidor	28
6. Conjunto de estações de trabalho	29
7. <i>Mainframe</i> - Supercomputadores	30
Capítulo 3 - Introdução à ciência da computação	31
Capítulo 4 - Tecnologia da Informação	35
Parte 2 - Representação e processamento da informação	39
Capítulo 5 - Representação e processamento da informação	41
Introdução	41
1. Identificando os bits	42
2. <i>Strings</i> e vetores	42
3. Representação de dados	43
Capítulo 6 - Sistemas de Numeração	45
1. O sistema binário	45
2. O sistema hexadecimal	47
Capítulo 7 - Aritmética binária	51
1. Aritmética de Números Inteiros Positivos	51
2. Aritmética de Números Inteiros Negativos	54
3. Números de ponto flutuante	56
Capítulo 8 - Portas lógicas	59
1. Portas básicas	59
2. Portas compostas	61

Parte 3 - Arquitetura de computadores	67
Capítulo 9 - Arquitetura de computadores	69
Introdução	69
Capítulo 10 - Unidade central de processamento e memória	75
1. Processador.....	75
2. Barramento de dados.....	78
3. Memória	78
Capítulo 11 - Dispositivos de armazenamento	81
1. Disco rígido	81
2. Disquete flexível	82
3. Cartões da memória Flash	82
4. CD-ROM.....	83
Capítulo 12 - Sistemas de entrada e saída	85
1. Dispositivos de Entrada	85
2. Dispositivos Apontadores	86
3. Tablet gráfico	87
4. Scanner.....	88
5. Microfone	88
6. Dispositivos de saída	88
7. Dispositivos de Entrada e Saída	91
Parte 4 - Conjunto de instruções e endereçamento	95
Introdução	97
Capítulo 13 - Instruções	101
Capítulo 14 - Instruction Set Architecture - ISA	105
1. Numero de operandos e localização	106
Capítulo 15 - Modos de endereçamento	111
Capítulo 16 - CISC versus RISC	117
1. Princípios de projeto para computadores modernos.....	118
Sobre os autores	121

Parte

1

**Histórico,
aplicações e conceitos da
Tecnologia da Informação**

História e evolução dos computadores

Introdução

Neste capítulo será apresentada uma resenha histórica da evolução dos computadores, partindo do seu surgimento no formato mecânico até os sistemas usados na atualidade, destacando as suas principais características estruturais. Em sincronia com a rápida evolução da área, o desenvolvimento de novas tecnologias de apoio e sua permanente evolução impulsionou a imediata difusão e adoção nas mais diversas áreas de aplicação.

Hoje em dia a utilização de sistemas baseados em computador é uma necessidade básica para o desenvolvimento de qualquer atividade. O foco principal destes sistemas é a produção, tratamento, e comunicação de informação. A partir desta nova realidade social, surge o conceito de Tecnologia da Informação (TI) englobando o conjunto de todas as atividades que, baseadas em recursos computacionais, participam da geração, processamento e uso da informação.

O computador é uma máquina capaz de receber, armazenar, tratar e produzir informações de forma automática. Ao longo da sua evolução, que teve início no século 16, muito foi exigido em relação ao desempenho dos computadores, atingindo atualmente grande rapidez e precisão no desempenho das operações.

Para que uma máquina possa executar uma tarefa é preciso criar um **algoritmo**¹ definindo os passos em que a tarefa pode ser executada. No entanto, para que uma máquina possa executar a tarefa descrita pelo algoritmo, esse algoritmo deve ser traduzido para uma linguagem de programação para que possa ser compreendido pela máquina.

De uma forma geral, programas são coletivamente denominados de software, enquanto que os equipamentos físicos necessários para a execução de tais programas, hardware.

Originalmente estudados no contexto da matemática, os algoritmos eram projetados para descrever o raciocínio necessário à resolução de problemas de um determinado tipo, de forma unificada. Com o passar do tem-

¹ Mais precisamente, um algoritmo determina um conjunto ordenado e não-ambíguo de passos que definem uma atividade finita.

po, e a partir da sua popularização nos diferentes âmbitos da sociedade, as máquinas se tornaram responsáveis pela execução de tarefas cada vez mais complexas, propiciando o surgimento da disciplina Ciência da Computação com o objetivo de elaborar, nas mais diversas áreas de aplicação, algoritmos cada vez mais complexos e eficientes.

A complexidade dos algoritmos pode encontrar limitações na própria tecnologia, ou seja, na própria capacidade da máquina. Algoritmos complexos podem demandar recursos da máquina que não estejam disponíveis como, por exemplo, maior capacidade de memória do que a que a máquina pode oferecer. Esta necessidade impulsiona a pesquisa e o avanço tecnológico que também são áreas de estudo da Ciência da Computação.

1. Precursores do computador

A história da tecnologia se desenvolveu paralelamente e é quase tão antiga quanto a história da evolução humana. Um marco considerado fundamental neste contexto é o início das atividades agrícolas, a partir do desenvolvimento de melhores técnicas de domínio do homem sobre a natureza, em relação aos seus antecessores. Este marco estabelece uma mudança radical nos hábitos até então nômades do homem pré-histórico, para se tornar sedentário e garantir seu território para a agricultura e a criação de animais. Surgem assim aldeias e povoados e inicia-se uma era de grandes descobertas.

Com o desenvolvimento da lavoura tornou-se necessário um método para a contagem do tempo, delimitando as épocas de plantio e colheita. O primeiro passo do homem rumo à ciência e à tecnologia é a concepção da idéia de número. Inicialmente era utilizado um mecanismo de comparação entre conjuntos de elementos, passando a comparar por agrupamentos, na medida que os conjuntos cresciam em tamanho. No entanto, esta abordagem ainda não era muito prática, pois a contagem era feita utilizando-se os dedos das próprias mãos, mas foi importante por dar origem ao sistema de numeração de base decimal.

A compreensão do conceito de número como uma idéia abstrata, e não somente como uma qualidade de um determinado objeto, não foi um processo imediato. Conforme Bertrand Russell, “foram necessários muitos anos para se descobrir que um par de faisões e um par de dias eram ambos instâncias do número dois”. Esta dissociação representa o primeiro passo em direção a um sistema de notação, e daí para a aritmética. Para auxílio deste método, eram usados gravetos, contas ou marcas na parede.

Com a evolução da humanidade se fizeram necessárias novas invenções na tentativa de criar dispositivos mecânicos para auxiliar a fazer cálculos. Apesar dos computadores eletrônicos terem efetivamente aparecido somente na década de 40, os fundamentos em que se baseiam remontam a centenas ou até mesmo milhares de anos.

1.1. O Ábaco (aprox. 3500 a.C.)

A palavra *cálculo* tem sua origem no termo latino *calculus*, que há milhares de anos servia para denominar pequenas pedras que eram usadas para contar deslizando-se por sulcos cavados no chão. A partir desse elemento de cálculo, outros similares apareceram em diversos lugares do mundo, sendo chamados de ábaco. O mais antigo data de aproximadamente 3500 a.C. e surgiu provavelmente no Mediterrâneo. Utilizado pelos mercadores para contar e calcular, o ábaco era tão eficiente que se propagou por toda parte, antes do século XVII, e nenhum outro instrumento de cálculo podia competir com ele.

Por volta do ano 2600 a.C. apareceu o ábaco chinês que evoluiu rapidamente e foi chamado em sua forma final de Suan-Pan. Foi muito utilizado pelas civilizações egípcia, grega, chinesa e romana, tendo sido encontrado no Japão, ao término da segunda guerra mundial, chamado de Soroban.

O ábaco constituiu, portanto, o primeiro dispositivo manual de cálculo, e é utilizado ainda nos dias de hoje. Essa máquina é bastante simples e serve para representar números no sistema decimal e realizar operações com eles.

A figura 1 apresenta um ábaco, que consiste numa moldura retangular dividida em 2 partes; possui uma vareta vertical para cada dígito, sendo que cada vareta tem em sua parte inferior 5 anéis que, em repouso, ficam para baixo, e na parte superior, 2 anéis que, em repouso, ficam para cima.

Os anéis são movidos de um lado para outro sobre as varetas, e suas posições representam valores numéricos. Cada unidade acrescentada a um dos dígitos do número é representada pelo movimento para cima de um dos anéis da parte inferior da vareta. Quando os 5 estão na parte de cima devem ser movidos para baixo.

O mesmo deve ser feito com os mesmos anéis na parte superior da mesma vareta. Se os dois anéis da parte superior estão para baixo, devem ser movidos para cima acrescentando-se uma unidade à vareta seguinte, à esquerda dessa vareta. O maior número que pode ser calculado depende do número de varetas.

Fazendo uma analogia com o computador, pode-se dizer que é por meio das posições dos anéis que este dispositivo representa e armazena dados. Os dados são ingressados quando os anéis são posicionados e a saída dos resultados consiste em observar as posições dos anéis.

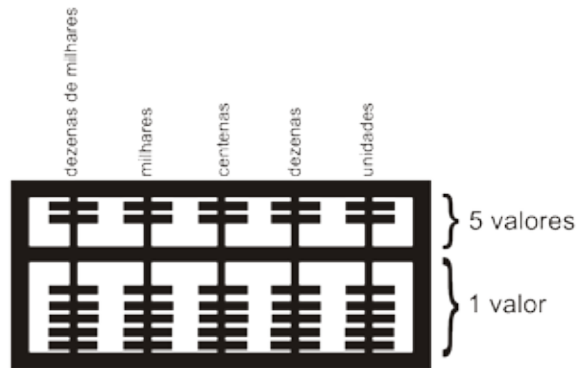


Figura 1 - Modelo ilustrativo de um ábaco

1.2. Logaritmos e os primeiros dispositivos mecânicos de cálculo

Passaram-se séculos sem que qualquer invenção ficasse registrada até que fossem criados tais bastões. John Napier (1550-1617), Barão de Merchiston, nobre escocês, teólogo e matemático, é bastante conhecido pela descoberta dos logaritmos. Também passou grande parte de sua vida inventando instrumentos para ajudar no cálculo aritmético, principalmente para o uso de sua primeira tabela de logaritmo, estabelecendo uma relação entre uma série geométrica e uma série aritmética.

Essas descobertas ajudaram a tornar mais simples os cálculos aritméticos, transformando operações de multiplicação em operações de soma, e operações de divisão em operações de subtração. A importância dos logaritmos não se resumia somente à simplificação dos cálculos, mas teve contribuição também na ciência e na tecnologia.

1.3. Auxílios mecânicos para cálculo

John Napier generalizou o procedimento tabular e construiu em 1617 um dispositivo simples e barato constituído de bastões de ossos (figura 2). Eram tabelas móveis de multiplicação e divisão feitas de marfim. Um conjunto completo de bastões de Napier consiste em 9 peças: uma para cada dígito de 1 a 9.

Cada uma de suas hastes é essencialmente uma coluna de uma tabela de multiplicação contendo os algarismos de 1 a 9 no quadrado superior e os oito quadrados restantes do bastão continham o produto deste número por 2,

3, 4, 5, 6, 7, 8 e 9 respectivamente. Para obter o produto, os dígitos de cada diagonal são somados da direita para a esquerda.

Apesar de dispositivos semelhantes aos bastões terem sido usados desde o final do século XVI, eles só apareceram documentados a partir de 1614.

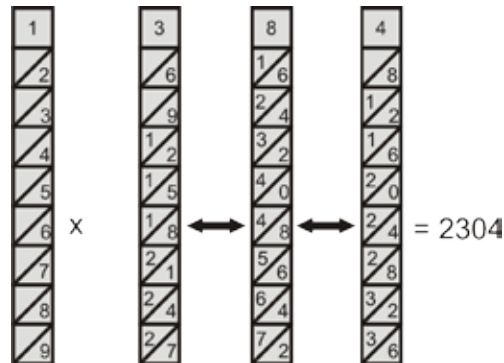


Figura 2 - Modelo ilustrativo dos bastões de Napier

1.4. Régua de cálculo

A partir das descobertas de Napier surgiu uma outra grande invenção, desenvolvida pelo brilhante matemático inglês Willian Oughtred, e tornada pública em 1638: a régua de cálculo. Ganhou sua forma atual, de uma régua que se move entre dois outros blocos fixos, por volta do ano de 1650, tendo sido esquecida por duzentos anos, para se tornar, no século XX, um grande símbolo de avanço tecnológico, com uso extremamente difundido.

Uma regra de cálculo é considerada como o primeiro dispositivo mecânico-analógico de computação. Com um formato tipicamente circular, este instrumento permite a realização de cálculos de forma aproximada, por meio de guias graduadas deslizantes. Existem diversas variantes da regra de cálculo dependendo da área de aplicação específica da engenharia. Embora todas elas realizem as operações básicas, para o caso da regra para uso em engenharia elétrica, por exemplo, podem existir escalas para conversão entre unidades de potência (kW), cálculo de tensão em condutores (V) e outras.

A régua de cálculo e as calculadoras mecânicas foram largamente utilizadas por engenheiros até 1970, quando surgiram as calculadoras eletrônicas cuja utilização era muito mais simples e seus resultados mais precisos. Um cálculo como 1345×3442 , por exemplo, é resolvido em poucos segundos com uma régua de cálculo. No entanto, o resultado é apenas uma indicação de que o valor é próximo de 4.650.000, quando o valor exato é 4.629.490.

1.5. Geração zero (século XVII): a evolução da calculadora

A calculadora decimal foi construída entre 1642 e 1644 por Blaise Pascal. Trata-se de um aparelho mecânico consistindo de uma caixa com rodas e engrenagens interligados que possibilitava a execução de operações aritméticas de soma e subtração. O resultado era apresentado numa seqüência de janelinhas.

Embora o aparelho se mostrasse muito útil, não obteve sucesso comercial, pois seu preço foi muito alto.

Em 1671, o filósofo e matemático alemão de Leipzig, Gottfried Wilhelm von Leibnitz (1646 - 1716) introduziu uma melhoria em relação à Calculadora de Pascal considerando o conceito de realizar multiplicações e divisões através de adições e subtrações sucessivas. A máquina, capaz de executar as quatro operações básicas da matemática, foi construída em 1694. No entanto sua operação apresentou-se muito deficiente e sujeita a erros, tendo sido, portanto, abandonada.

O sucesso comercial veio 1820 com a Calculadora a Arithmometer, projetada por Charles Xavier Thomas (1785-1870). A máquina era capaz de efetuar as 4 operações aritméticas básicas utilizando o mesmo princípio da calculadora de Leibnitz, precisando de assistência do usuário para a realização de cálculos de dividir. Até 1850 venderam-se cerca de 1500 Arithmometers.

1.6. Máquina diferencial de Babbage (1823)

Entre 1802 e 1822, Charles Babbage (1792-1871), um matemático e engenheiro britânico, previu que grande parte da informação poderia ser manipulada por máquina, caso fosse possível antes converter a informação em números e projetou um engenho para isso. Em particular, Babbage tinha a intenção de reduzir o trabalho de calculo e corrigir erros nas tabelas matemáticas da sua época.

Neste contexto, a chamada Máquina de diferenças foi projetada para computar e imprimir um conjunto de tabelas de funções (logaritmos, funções trigonométricas, etc.) sem a intervenção de um operador humano. A máquina seria movida a vapor, era baseada também no princípio de discos giratórios e operada por uma simples manivela. O projeto de Babbage é considerado o precursor dos modernos computadores eletrônicos digitais.

Ao operador cabia somente iniciar a cadeia de operações, e a seguir a máquina tomava seu curso de cálculos, preparando totalmente a tabela prevista. Em 1823, o governo britânico financiou a construção de uma nova versão, mas não obteve resultado satisfatório, devido aos limites do ferramental industrial da época. Embora tenham sido contratados especialistas em máquinas, o projeto exigia mecanismos altamente precisos e movimentos complexos.

Neste contexto, Babbage se viu obrigado a destinar parte dos seus recursos e tempo para desenhar peças e ferramentas no intuito de avançar no estado da arte da tecnologia, retardando assim o desenvolvimento do projeto. Após 10 anos de trabalho, tudo que Babbage havia conseguido era uma pequena máquina de 3 registros e 6 caracteres, sendo que deveria ser, de acordo com o projeto, uma máquina de 7 registros e 20 caracteres cada. Finalmente, em 1833, quando o governo inglês desistiu do financiamento, Charles Babbage parou de trabalhar em sua máquina.

O projeto de Babbage, no entanto, é considerado o precursor dos modernos computadores eletrônicos digitais.

1.7. Máquina analítica

Após vários anos de tentativas, Babbage concentrou seus esforços no projeto de uma máquina de cálculo universal capaz de executar qualquer tipo de cálculo (1833).

Babbage despertou para seu novo projeto observando a revolução produzida pelos teares de Jacquard. Utilizando a mesma estratégia, a Máquina Analítica seria programada através de um conjunto mutável de instruções, para calcular várias funções diferentes, processando padrões algébricos da mesma maneira que o tear processava padrões de desenhos. Este evento determina um marco na evolução do computador: surge o conceito de “programação”.

Saiba Mais



Teares de Jacquard (1801)

O francês Joseph-Marie Jacquard (1752-1834) introduziu a primeira máquina para substituir o trabalho humano. Era uma máquina que automatizava os processos mecânicos de fabricação de tecido. Para executar um determinado trançado, a fiandeira deveria ter um plano ou programa que lhe dissesse quais fios deveria passar por cima ou por baixo, em que momento repetir o desenho, etc. A evolução chave propiciada pela máquina de Jacquard era o uso de uma série de cartões cujas perfurações estavam configuradas para descrever o modelo a ser produzido.

A abordagem utilizada por Jacquard implementa o conceito de armazenamento de informações em placas perfuradas com o objetivo de controlar uma máquina de tecelagem. Este conceito será posteriormente estendido para sua aplicação no processamento de dados.

O sucesso foi total e em 1812 havia na França 11.000 teares de Jacquard, e despertou na sociedade daquela época, temor pelo desemprego, provocando uma grande reação popular contra essa espécie de pré-automatização.

A máquina era constituída de unidade de controle de memória aritmética e de entrada e saída, e sua operação era governada por um conjunto de cartões perfurados, de modo que, de acordo com os resultados dos cálculos intermediários, a máquina poderia saltar os cartões, modificando dessa forma o curso dos cálculos.

Babbage investiu toda sua fortuna pessoal e de seu filho, que com ele trabalhou durante anos, na construção de sua máquina Analítica, vindo a falecer em 1871, sem findar a construção.

1.8. A primeira programadora da história da computação

Babbage percebeu que para criar as instruções responsáveis pela execução dos cálculos precisaria de um tipo inteiramente novo de linguagem e a imaginou como números, flechas e outros símbolos. Neste cenário surge a importante contribuição de Ada Augusta Byron, condessa de Lovelace (1851 - 1852), filha do famoso poeta britânico Lorde Byron.

Ada Byron é uma das poucas mulheres a figurar na história do computador. Matemática talentosa, compreendeu o funcionamento da Máquina Analítica e escreveu os mais completos relatos sobre o processo. Ada compreendeu o alcance das novas invenções de Babbage, que diferentemente dos projetos anteriores, de funcionamento analógico (execução dos cálculos usando medidas), a Máquina de Babbage era digital (execução dos cálculos usando formulas numéricas).

Antes mesmo da máquina ser construída, Ada aproveitou seus conhecimentos em matemática para criar programas combinando funções lógicas e aritméticas para a máquina, tornando-se a primeira programadora de computador do mundo. Alguns dos conceitos fundamentais na programação de computadores largamente utilizados até os dias de hoje, tais como sub-rotinas, laços e saltos, já foram utilizados por Ada na programação para a Máquina analítica.

Ada também colaborou com a documentação do projeto da Máquina de Babbage, através da publicação de uma coleção de notas, redescobertas em 1953. Em 1980, o Departamento de Defesa dos EUA registrou a linguagem de programação ADA, em sua homenagem.

1.9. Máquina de Hollerith

Preocupado com a lentidão na análise dos dados obtidos do censo anual, cuja demora era de aproximadamente 10 anos, o jovem funcionário do *Unites States Census Office*, Herman Hellerith, idealizou uma máquina capaz de tabular tais

informações. Baseado na técnica de cartões perfurados de Jacquard, aplicou o mesmo conceito, mas para o armazenamento dos dados coletados no censo.

O sistema era capaz de registrar números como buracos nos cartões em locais específicos os quais eram classificados através de pinos. A Máquina de Hollerith ou Máquina Tabuladora era composta das seguintes unidades:

- Unidade de controle, que dirigiria a seqüência das operações de toda a máquina através de furos em cartões perfurados.
- Entrada de dados, que utilizava também cartões perfurados.
- Saída, que perfuraria os resultados em cartões para uso posterior como entrada, aumentando assim a memória interna com armazenamento externo, indefinidamente grande.
- Saída impressa utilizada na apresentação dos resultados finais, tais como tabelas matemáticas, a qual de uma linotipo automática acoplada ao sistema.

Com a utilização da Máquina de Hollerith fizeram muito sucesso no censo de 1890 nos EUA, conseguindo que o tempo de processamento dos dados do censo diminuísse de 8 para 3 anos. A tecnologia de cartões perfurados foi adotada rapidamente por diversos países da Europa, difundindo a utilização das máquinas Hollerith a nível mundial e por bastante tempo.

Posteriormente, Hollerith fundou a bem sucedida companhia *Tabulating Machine Company*. Em 1924, esta firma mudou de nome, tornando-se a *International Business Machines Corporation*, hoje mais conhecida como IBM. No início, as vendas da IBM eram baseadas na linha de equipamentos de escritório e, em particular, máquinas tabulares.

Com o desenvolvimento dos primeiros dispositivos mecânicos para cálculo automático, começa efetivamente a vertente tecnológica que levará à construção dos primeiros computadores.

Para refletir:

Quando chegou o primeiro computador no Brasil? Quais eram suas características?

2. As máquinas de primeira geração (1930-1958)

O automatismo completo foi alcançado no século XX, quando um grande número de projetos foi implementado. Embora a concepção da máquina de Babbage foi o alicerce dos computadores modernos, a principal diferença era a utilização de circuitos eletrônicos (relés² e válvulas³) em lugar de mecânicos (rodas e engrenagens).

² Relé: eletroímã cuja função é abrir ou fechar contatos elétricos com o intuito de interromper ou estabelecer circuitos.

³ Válvula: dispositivo que conduz a corrente elétrica num só sentido

Este novo marco na evolução dos computadores determina o surgimento dos computadores chamados de Primeira Geração.

A principal vantagem das máquinas a relé sobre as máquinas mecânicas era, sem dúvida, a maior velocidade de processamento (milissegundos). Ainda, um outro aspecto positivo era a possibilidade de funcionamento contínuo, apresentando uma menor taxa de erros de cálculo.

Os computadores da primeira geração contavam com dispositivos de entrada/saída primitivos. A grande utilidade dessas máquinas era no processamento de dados, os quais eram ingressados e armazenados com base na utilização de cartões perfurados.

Por outro lado existia uma série de desvantagens como: custo elevado, relativa lentidão, pouca confiabilidade, grande quantidade de energia consumida e necessitavam de grandes instalações de ar condicionado para dissipar o calor gerado por um grande número de válvulas (cerca de 20 mil). Os computadores mais representativos desta geração foram: o MARK I, ABC (*Atanasoff Berry Computer*), ENIAC (*Electronic Numeric Integrator and Calculator*).

Juntamente com o ENIAC, ocorreu também o desenvolvimento na área de periféricos de computador com o aparecimento de equipamentos tais como as unidades de fita magnética, impressoras, etc.

2.1. Arquitetura Von Neuman

Uma contribuição importantíssima desta época foi o conceito de programa armazenado, introduzida pelo matemático húngaro, naturalizado americano John Von Neuman por volta de 1940.

A partir de sua experiência como consultor no projeto ENIAC, von Neuman conhecia os problemas da programação destas máquinas em relação à introdução dos programas. Os programas para os computadores da época eram construídos com base em modificações nos circuitos. Esta técnica era muito trabalhosa e demorada.

A solução proposta por von Neuman foi a utilização de cartões perfurados para o ingresso do programa, analogamente a como era feito com os dados. Para sustentar esta abordagem, John von Neuman assim desenvolveu a lógica dos circuitos, os conceitos de programa e operações com números binários.

Nesse modelo, um elemento Processador segue as instruções de programas armazenados em uma Memória de Programas, para ler canais de Entrada, enviar comandos sobre canais de Saída e alterar as informações contidas em uma Memória de Dados.

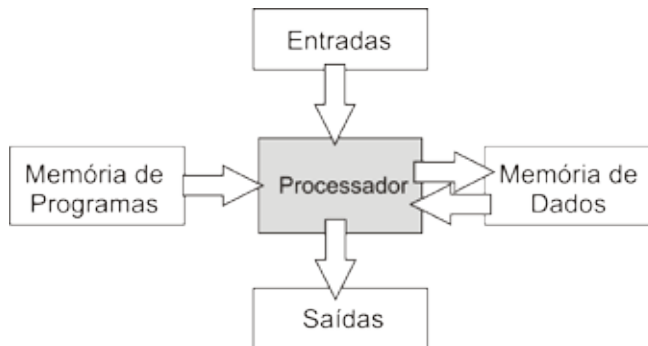


Figura 3 - Modelo proposto por Von Neuman

Estes conceitos, adotados nos computadores atuais, revolucionou o conceito de programação de computadores da época, tornando os programas muito mais flexíveis e versáteis.

A arquitetura de von Neuman inicial evoluiu para uma estrutura em barramento, que é a base dos computadores modernos. Nessa estrutura, as memórias de dados e de programa são fundidas em uma memória única facilitando a sua manipulação, e as comunicações entre elementos são efetuadas através de uma via comum de alta velocidade:

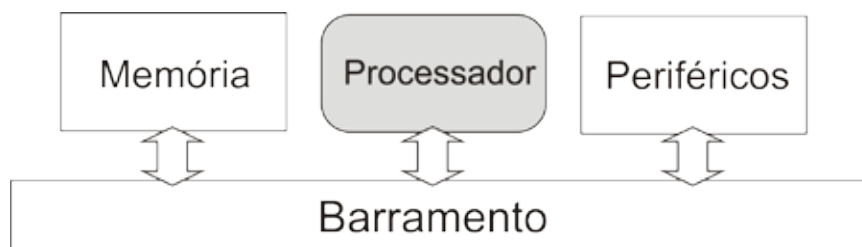


Figura 4 - Evolução do modelo de Von Neuman

O novo conceito de programação introduzido por Von Neuman deu origem a muitos outros projetos nos quais ele próprio esteve envolvido, como por exemplo, o EDVAC (*Electronic Discrete Variable Automatic Computer*), o IBM 650 (o primeiro computador da IBM), e o UNIVAC (Universal Automatic Computer), que foi o primeiro computador a ser fabricado em linha.

3. Computadores de segunda geração (1955-1965)

O novo marco que caracteriza a segunda geração de computadores é a invenção do transistor. Considerada uma das maiores descobertas da história moderna, o transistor é responsável pela revolução eletrônica da década de 1960. Esta nova tecnologia veio substituir a utilização de válvulas: em 1956 já se produziam computadores com esta tecnologia fabricados pela IBM e logo

depois a DEC (*Digital Equipment Corporation*), tornando estas empresas líderes mundiais na indústria de computadores. Com os transistores, os computadores passaram a ocupar um menor espaço físico, trabalhando a temperatura mais baixa e consumindo menos energia elétrica em relação às válvulas.

Com a segunda geração apareceram as memórias com anéis ferromagnéticos, evoluindo para as fitas magnéticas, as que se tornaram a forma dominante de armazenamento secundário. Algumas vantagens das fitas magnéticas sobre as perfuradas é que possuíam capacidade muito maior de armazenamento e o ingresso dos dados mais rápido. Posteriormente surgiram os discos magnéticos (1962), o que permitiu o acesso direto à arquivos muito grandes.

Também nesse período houve avanços no que se refere às unidades de memória principal, como por exemplo, a substituição do sistema de tubos de raios catódicos pelo de núcleos magnéticos, utilizado até hoje nos *chips* de memória RAM. Os dispositivos de memória auxiliar introduzidos na primeira geração continuam a ser utilizados.

Apareceram também os modernos dispositivos periféricos, tais como as impressoras, as fitas magnéticas, os discos para armazenamento, etc. Os computadores passaram a ter um desenvolvimento rápido, impulsionados principalmente por dois fatores essenciais: os sistemas operacionais e as linguagens de programação.

Exemplos de computadores dessa geração são o IBM 1401 e o Honeywell 800. O primeiro minicomputador foi o PDP-1, criado em 1959 e instalado em 1961. O primeiro produzido comercialmente foi o PDP-5. Um dos computadores mais comercializados nesta época foi o IBM 7090.

Para refletir

Pesquise sobre os computadores que foram os principais representantes desta geração e descreva suas características.

4. Computadores de terceira geração (1965-1980)

A substituição dos transistores pela tecnologia dos circuitos integrados de silício caracterizou os computadores da terceira geração. Os circuitos integrados foram criados por Robert Noyce, que foi o inventor do microchip e co-fundador da *Intel Corporation*. A tecnologia entrou no mercado em 1961 pela *Fairchild Semiconductor* e pela *Texas Instruments*, localizadas no Vale do Silício na região de Palo Alto e Stanford, na Califórnia.

A tecnologia dos circuitos integrados permitiu que dezenas de transistores fossem colocados em um único chip. Desta forma, surgiram computa-

dores de menores dimensões, mais rápidos (nanossegundos), confiáveis e baratos do que as máquinas das gerações anteriores.

A tecnologia utilizada na época era a de pequena escala de integração (SSI – *Small Scale of Integration*) integrando aproximadamente mil transistores no circuito de uma pastilha.

Dentre os computadores mais representativos dessa geração temos a linha *System/360* desenvolvido pela IBM, empresa líder da área de computadores da época (1965). O *System/360* foi projetado pra computação científica e comercial e apresentava muitas inovações:

- Família de máquinas com a mesma linguagem de montagem com tamanho e capacidades crescentes, projetada para propiciar compatibilidade entre um grande número de modelos.
- Multiprogramação: vários programas simultaneamente em memória, o que resulta em um maior aproveitamento da CPU.
- Primeira máquina capaz de emular (simular) outros computadores. Com isso, podiam ser executados programas antigos sem necessidade de modificar ou traduzir.
- Maior espaço de endereçamento de memória.

5. Computadores de quarta geração (1980 - ...)

Esta geração é marcada por uma drástica redução do tamanho dos computadores, porém com desempenho muito melhor: mais velozes e poderosos. Este avanço foi possível graças à tecnologia de circuitos integrados, possibilitando a integração de milhares de componentes eletrônicos em um pequeno espaço ou chip, dando origem aos microprocessadores.

Esta evolução teve início na década de 70, com a tecnologia de grande escala ou LSI (*Large Scale Integration*), que integrava até 65 mil componentes em um único chip. Nos anos 80 a técnica foi aperfeiçoada, atingindo a faixa de milhares de transistores numa pastilha VLSI (*Very Large Scale Integration*).

A redução de tamanho e custo desta nova geração de computadores possibilitou a sua aquisição para uso pessoal, privilégio que antes era somente reservado para organizações que poderiam arcar com grandes custos e amplos locais para a acomodação. A partir da abertura de um novo mercado de uso doméstico para os computadores, surgiram novas necessidades e desafios para as companhias. De forma geral, os computadores pessoais eram utilizados para processamento de texto, manipulação de planilhas e aplicações interativas como jogos.

No início dessa geração nasceu a empresa americana Intel, que começou a desenvolver o primeiro microprocessador, o Intel 4004 que manipulava palavras binárias de 4 bits, um circuito integrado com 2250 transistores, equivalente ao ENIAC.

O 4004 foi seguido pelo Intel 8008 de 8 bits e, mais tarde, pelo Intel 8080. O primeiro microcomputador da história foi o Altair 8800, que usava o chip Intel 8088, tornou-se padrão mundial da época para os microcomputadores de uso pessoal, abrindo uma nova era na história da informática.

Stephen Wozniak e Steve Jobs formaram em 1976 uma pequena empresa, a Apple, onde construíram, segundo conta a lenda, numa garagem de fundo de quintal, o Apple I. Um ano depois, com um novo e melhor projeto, surge o Apple II, primeiro microcomputador com grande sucesso comercial seguido, mais tarde, pelo Apple III.

A IBM decidiu entrar no mercado de computadores pessoais, introduzindo o PC (IBM *Personal Computer*), um microcomputador com tecnologia de 16 bits, estabelecendo o Intel 8088 como sua CPU. Lançado em 1981, o PC se tornou campeão de vendas da história dos computadores. Um PC tem a seguinte configuração típica – 256 a 640 K de memória RAM na placa principal, duas unidades de disco flexível de 360 K, controlador de vídeo gráfico, monitor monocromático e interface serial ou paralela para a impressora. Seu clock era de 4,77 MHz.

Com o sucesso impulsionado pela IBM PC, a Intel continuou fazendo versões cada vez mais poderosas. Os principais modelos de PC são: PC-XT, PC-XT 286, PC-AT386, PC 486.

O IBM PC vinha equipada com o sistema operacional MS-DOS, fornecido pela novata *Microsoft Corporation*.

Em 1993 chegou ao mercado o *Pentium*, cuja versão *Pentium III* possui cerca de nove milhões de transistores. O *Pentium* trouxe um novo fôlego às chamadas estações de trabalho. Uma característica comum às máquinas das gerações até então, é a existência de uma única CPU para executar o processamento.

Uma das novidades que o *Pentium* trouxe é que possibilita a simulação de dois processadores, ou seja, um princípio de paralelização antes possível apenas em supercomputadores e que agora está ao alcance dos usuários de microcomputadores. Mais recentemente, já existem computadores funcionando com mais de uma CPU.

Os atuais avanços em pesquisa e o projeto de novas tecnologias para os computadores estão possibilitando o surgimento da quinta geração. Dois avanços que configuram um divisor de águas são o processamento paralelo, que quebrou o paradigma de von Neumann, e a tecnologia dos supercondutores.

Atividades de avaliação



1. Pesquise sobre os computadores que foram os principais representantes desta geração e descreva suas características.

Tecnologias de computadores e suas aplicações

Na tecnologia atual, podemos classificar os computadores de acordo com a suas capacidades e finalidades de uso para o qual foram desenvolvidos. Possivelmente os computadores pessoais sejam os mais amplamente utilizados e conhecidos, no entanto existem outras variantes que serão analisadas nesta seção.

A quarta geração de computadores cujo domínio se estende até nossos dias, é impulsionada principalmente pela capacidade dos engenheiros e fabricantes de chips em empacotar cada vez mais transistores por pastilha, a cada ano. A cada incremento, são gerados computadores com memórias maiores e processadores mais poderosos.

A evolução tecnológica observada nas últimas décadas comprova que o número de transistores dobra a cada 18 meses, e a previsão indica que essa taxa será mantida no futuro. Esta constatação é conhecida como Lei de Moore, que dá fundamento ao chamado círculo virtuoso onde:

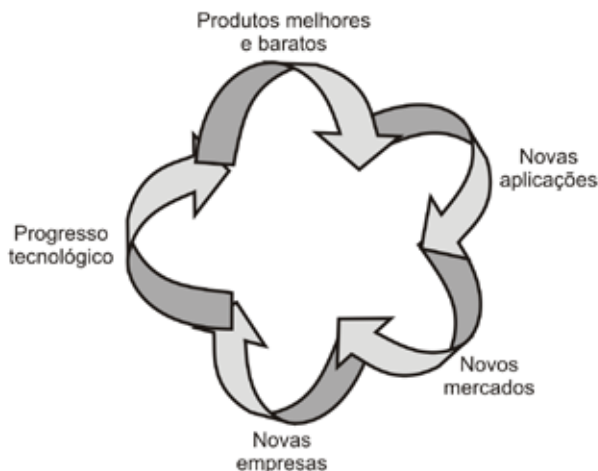


Figura 5 - Representação do círculo virtuoso relacionado ao desenvolvimento de computadores

Os benefícios alcançados pela lei de Moore podem ser mapeados para duas atitudes possíveis. Por um lado construir computadores cada vez mais poderosos mantendo o preço constante. Por outro lado, uma postura mais conservadora seria continuar a desenvolver o mesmo computador por um preço cada vez mais baixo. A indústria de computadores adotou as duas posturas, o que resultou na ampla variedade de computadores que existem atualmente e que podem ser utilizados para uma grande variedade de aplicações.

1. Computador descartável

Quem não recebeu alguma vez um cartão de congratulações musical? Ou deu de presente para a mãe? Pois é, esta tal vez seja a aplicação mais simples de computador descartável. Trata-se de um único chip sonoro descartável, porem com capacidade computacional maior do que o maior computador do mundo de 1955.

No entanto, utilidades mais interessantes sugeriram a partir do desenvolvimento do chip RFID (*Radio Frequency IDentification* ou identificação por radiofrequencia). Estes chips de baixíssimo custo (centavos), não possuem bateria e a sua espessura é de menos de 0,5 mm., possuem um minúsculo radiotransponder e um único numero de 128 bits embutido.

Esta tecnologia é utilizada para a identificação automática através de sinais de rádio, recuperando e armazenando dados remotamente através de dispositivos chamados de *tags* RFID: quando pulsados por uma antena transmissora externa, são alimentados pelo sinal de radio de entrada por tempo suficiente para transmitir seu numero de volta à antena.

Uma tag pode ser colocado em uma pessoa, animal, equipamento, embalagem ou produto, dentre outros. Além das tags passivas, que respondem ao sinal enviado pela base transmissora, existem ainda as tags ativas, dotadas de bateria, que lhes permite enviar o próprio sinal. São bem mais caras que as tags passivas.

A principal funcionalidade dos tags RFID é a identificação de todo tipo de objeto, inclusive seres vivos. No caso de objetos, estes tags viriam substituir os populares códigos de barras.

- Implantes em animais. Este tipo de sistema usado na identificação dos animais ajuda no gerenciamento dos mesmos entre as companhias, no controle de epidemias e garantia de qualidade e procedência. A identificação animal por sistemas de RFID pode ser feita de quatro maneiras diferentes: colares, brincos, injetáveis ou ingeríveis (bolus).

- Implantes humanos. Implantes de chips RFID usados em animais agora estão sendo usados em humanos também. Em 1998, o professor britânico de cibernética Kevin Warwick implantou um chip no seu braço. A empresa Applied Digital Solutions propôs seus chips “formato único para debaixo da pele” como uma solução para identificar fraude, segurança em acesso a determinados locais, computadores, banco de dados de medicamento, iniciativas anti-sequestro, identificação e monitoração de pacientes, entre outros.
- Hoje em dia, a maioria dos sistemas que gerencia recipiente é baseada em código de barras, porém no meio industrial o uso deste tipo de sistema não é confiável o suficiente, e os transponders de um sistema RFID pode guardar mais informações úteis posteriormente,
- No setor industrial os sistemas de RFID têm várias aplicações. Uma delas é na identificação de ferramentas, que no caso de grandes indústrias facilita o processo tanto de manutenção, quanto de substituição e administração das mesmas. Mas outro campo que sistemas RFID podem tanto melhorar a rapidez e qualidade do serviço, como também ter um papel de segurança nas indústrias é na identificação de recipientes, embalagens e garrafas, principalmente em produtos químicos e gases, onde um erro na hora de embalar pode causar sérios danos. Em algumas cidades do mundo, já existem sistemas RFID para controle de lixo, onde as tags são colocadas nas latas de lixo e os caminhões de coleta tendo leitores específicos.

A tecnologia usada em RFID se desenvolve rapidamente. Os menores são passivos (sem alimentação interna) e podem apenas transmitir seus números exclusivos quando consultados. Os maiores, são ativos, podem conter uma pequena bateria e um computador primitivo, sendo capazes de fazer alguns cálculos. Nesta classificação entram os Smart Cards usados em transações financeiras. (www.rfid.org).

Para refletir

Descreva outras aplicações possíveis para dispositivos RFID.

2. Microcontrolador

A grande maioria de aparelhos utilizados diariamente, inclusive para uso doméstico, possuem computadores embutidos ou microcontroladores. Estes dispositivos são utilizados para gerenciar as funcionalidades do aparelho e manipular a sua interface com o usuário. Podemos encontrar microcontroladores em uma grande variedade de aparelhos, por exemplo:

- Eletrodomésticos
- Aparelhos de comunicação
- Periféricos de computadores
- Equipamento de entretenimento
- Aparelhos de reprodução de imagens
- Equipamentos médicos
- Armamentos militares
- Dispositivos de vendas
- Brinquedos

Um carro moderno, por exemplo, pode conter até 50 microcomputadores, que executam diferentes atividades tais como injeção, rádio, GPS, etc. A previsão é que dentro de poucos anos, quase tudo o que depende de energia elétrica ou baterias para funcionar, possua um microcontrolador.

Um microcontrolador (também denominado MCU ou μC) é um pequeno computador-num-chip, mas completos. Cada microprocessador contém um processador, memória e funções de entrada/saída. Dentre as funções de entrada/saída temos: controle de botões do aparelho, controle de luzes e sons, etc. Os microcontroladores podem ser de propósito geral ou de propósito específico.

A tecnologia de microcontroladores é regida principalmente pela variável de custo, de forma a tornar viável economicamente o controle eletrônico de muitos processos. Microcontroladores geralmente são encomendados por milhões de unidades. Neste caso, um centavo a mais ou a menos pode determinar a escolha por um determinado fornecedor.

Desta forma, a escolha por uma arquitetura é decidida com base principalmente no custo de fabricação. Outros fatores que influenciam o custo dos microcontroladores é a quantidade de bits, quantidade de memória e tipo. Por outro lado, o aumento na capacidade de processamento é sempre necessário para conseguir atender às necessidades dos produtos de tecnologia mais avançada.

Outra característica dos microcontroladores é o funcionamento em tempo real, ou seja, que devem ser capazes de detectar um evento e dar uma resposta imediata.

Finalmente, uma característica inerente de sistemas embutidos (como é o caso de microcontroladores) é a restrição física em termos de tamanho, peso, consumo, etc. Estas restrições podem ser contornadas colocando os

dispositivos para “hibernar” enquanto nenhum evento interessante for percebido pelos periféricos, reduzindo o consumo de energia diminui. As duas últimas características fazem com que a arquitetura destes dispositivos precise ser projetada cuidadosamente.

Microcontroladores são usados com frequência em produtos e dispositivos de controle automático, tais como sistemas de controle em motores de automóveis, controles remotos, máquinas de escritório, ferramentas e brinquedos. Ao reduzir o tamanho, custo e consumo de energia em comparação com um projeto que usa microprocessador, memória e dispositivos de E/S separados, microcontroladores tornam viável economicamente o controle eletrônico de muitos processos.

Cerca de 50% dos microcontroladores vendidos são controladores “simples”, outros 20% são processadores de sinais digitais (DSPs) mais especializados. Os microcontroladores podem ser encontrados em praticamente todos os dispositivos eletrônicos digitais que nos cercam: Teclado do computador, dentro do monitor, Harddisk, relógio de pulso, rádio relógio, máquinas de lavar, forno de microondas, telefones, etc. Você está certamente cercado de dezenas deles agora.

Certamente eles foram tão ou mais importantes para a revolução dos produtos eletrônicos que os computadores. Eles permitiram a evolução de equipamentos que a anos não evoluíam, como os motores a combustão, que agora com o novo controle eletrônico podem funcionar com sistema bi-combustível e poluindo menos e as máquinas fotográficas, que migraram de processos químico/mecânico a circuitos com microcontroladores+Sensores Digitais+Memória.

3. Computador de jogos

Os computadores de jogos são computadores normais, com recursos gráficos especiais e capacidade de som, mas *software* limitado e pouca capacidade de extensão. Inicialmente projetados para jogos simples, eram utilizados CPU de baixo valor. Com o passar dos anos evoluíram se tornando poderosos e capazes de competir em desempenho com computadores pessoais.

Os produtos de tecnologia vieram se tornar populares conhecidos genericamente como videogame. O primeiro foi lançado em 3 de dezembro de 1994 no Japão e quase um ano depois nos Estados Unidos, pela Sony. O *PlayStation* foi um sucesso de vendas, chegando a 100 milhões de unidades até 2006 quando sua produção foi encerrada. Posteriormente a tecnologia evoluiu para o *PlayStation 2* e *3*. O *PlayStation* consistia em um console independente de jogos composta de uma unidade leitora de CD-ROM e ranhuras

para *memory cards* (cartões de memória que armazenam informações dos jogos) e entradas para joysticks.

4. Computador pessoal (PC)

Nesta categoria podemos considerar os computadores de mesa que hoje em dia são utilizados em muitos lares como ferramenta para as mais diversas atividades, e os notebooks.

O coração de um computador pessoal está na placa de circuito impresso, por exemplo, da Intel. Geralmente na placa é contida a CPU, memória e os vários dispositivos de E/S (som, modem), interfaces para os periféricos (teclado, mouse, disco), e encaixes de expansão.

Um *notebook* pode ser considerado um PC, pois segue a mesma especificação de *software* e *hardware*, com a diferença de serem projetados para serem portáteis.

Ambos os computadores possuem sistemas operacionais elaborados e uma grande variedade de *software* disponível.

5. Servidor

Com o surgimento das redes de computadores e *clusters* de máquinas, e com o posterior aumento na complexidade de tais estruturas de comunicação e compartilhamento, aparece a figura do servidor de rede. O servidor de rede é um computador dedicado responsável por prestar algum serviço à rede de computadores (*clientes*⁴). Os serviços prestados podem ser diversos: arquivos, correio eletrônico, impressão, etc. Foi o crescimento e popularização da internet que impulsionou o desenvolvimento e aperfeiçoamento de tecnologias para servidores.

O servidor de rede (Intranet ou Internet) é um computador geralmente reforçado cujo diferencial é a capacidade de trabalho em rede a alta velocidade. De uma forma geral, os computadores servidores são mais rápidos, possuem mais espaço em disco e possuem uma conexão de rede mais rápida, principalmente se são dedicados. Os servidores executam os mesmos sistemas operacionais do que PC. Os servidores podem ser projetados de acordo a uma configuração com um único processador ou com múltiplos processadores.

Dentre os diversos tipos de servidores, os mais conhecidos são:

- **Servidor de fax:** servidor para transmissão e recepção automatizada de fax pela Internet, e através de todas as estações da rede.
- **Servidor de arquivos:** servidor que armazena arquivos de diversos usuários.

⁴ Cliente-servidor é um modelo computacional que separa clientes e servidores, sendo interligados entre si geralmente utilizando-se uma rede de computadores. Cada instância de um cliente pode enviar requisições de dado para algum dos servidores conectados e esperar pela resposta. Por sua vez, algum dos servidores disponíveis pode aceitar tais requisições, processá-las e retornar o resultado para o cliente. Apesar do conceito ser aplicado em diversos usos e aplicações, a arquitetura é praticamente a mesma.

- **Servidor web:** servidor responsável pelo armazenamento de páginas de um determinado site, requisitados pelos clientes através de browsers.
- **Servidor de e-mail:** servidor responsável pelo armazenamento, envio e recebimento de mensagens de correio eletrônico.
- **Servidor de impressão:** servidor responsável por controlar pedidos de impressão de arquivos dos diversos clientes.

Um computador pode atuar em mais de um tipo diferente de servidor. Computadores que atuem como um único tipo de servidor é chamado de servidor dedicado. Os servidores dedicados possuem a vantagem de atender a uma requisição de um cliente mais rapidamente.

Servidores dedicados, que possuem uma alta requisição de dados por partes dos clientes e que atuam em aplicações críticas utilizam *hardware* específico para servidores. Já servidores que não possuam essas atuações podem utilizar *hardware* de um computador comum, não necessitando ser um supercomputador.

Devido a operar com muitas entradas e saídas de informações, os servidores necessitam de processadores de alta velocidade, algumas vezes alguns servidores são multi-processados, ou seja, possuem mais de um processador.

Por ter de operar por muito tempo (as vezes de maneira ininterrupta), alguns servidores são ligados a geradores elétricos. Outros utilizam sistemas de alimentação (por exemplo, o UPS) que continuam a alimentar o servidor caso haja alguma queda de tensão.

6. Conjunto de estações de trabalho

Considerando a boa relação custo/desempenho dos computadores pessoais, nos últimos anos tem se tornado uma prática comum a interconexão por rede de gigabytes, de um grande número de PCs para formar *Clusters of Workstations* (COWs).

Através da execução de *software* especial, todas as máquinas interconectadas podem trabalhar juntas num mesmo problema, atingindo um maior poder de processamento e alta velocidade desde que o processamento pode ser realizado em paralelo.

É este poder de processamento a um custo mais baixo, que começou a tornar a utilização de supercomputadores e *mainframes* desnecessária levando esta categoria de computadores caminho à extinção. A quantidade de máquinas pode variar de forma indefinida, limitada pelos recursos disponíveis.

7. Mainframe - Supercomputadores

Os *mainframes* são computadores de grande porte físico, desde que ocupam uma sala inteira, da mesma forma que os seus antecessores dos anos 60. A sua principal característica é a sua capacidade de E/S, equipadas com coleções de discos para armazenamento de dados.

Esta classe de computadores geralmente hospeda *software* legado há algumas décadas, desenvolvidos especialmente para esses sistemas. É por conta do grande risco em reprogramar e migrar essas aplicações para máquinas menores, que faz com que muitas organizações continuem a manter por milhões de dólares esses sistemas, em lugar de comprar um novo.

A história dos supercomputadores começa, de fato, no final de 1975 com o Cray-1. As aplicações para esses computadores são muito especiais e incluem laboratórios e centros de pesquisa aeroespaciais, empresas de altíssima tecnologia, previsão do tempo e a produção de efeitos e imagens computadorizadas de alta qualidade. Os supercomputadores são os mais poderosos, mais rápidos e de maior custo. Utilizam o conceito de processamento paralelo e são máquinas vetoriais, isto é, podem executar a mesma operação em diversas variáveis simultaneamente.

Atividades de avaliação



1. Pesquise quais as características e responsabilidades do Servidor de banco de dados e Servidor FTP.
2. Pesquise mais algum outro tipo de servidor que não tinha sido mencionado.

Introdução à ciência da computação

Ciência da Computação é o estudo dos algoritmos e suas aplicações, bem como das estruturas matemáticas indispensáveis à formulação precisa dos conceitos fundamentais da teoria da computabilidade e da computação aplicada.

Os estudos oriundos da Ciência da Computação podem ser aplicados em qualquer área do conhecimento humano em que seja possível definir métodos de resolução de problemas baseados em repetições previamente observadas. Desempenha por isso um papel importante na área de ciência da computação a formalização matemática de algoritmos, como forma de representar problemas decidíveis, i.e., os que são susceptíveis de redução a operações elementares básicas, capazes de serem reproduzidas através de um dispositivo mecânico/eletrônico capaz de armazenar e manipular dados.

Um destes dispositivos é o computador digital, de uso generalizado, nos dias de hoje, pelo custo reduzido dos componentes eletrônicos que formam o seu *hardware*.

A Ciência da Computação constitui uma base científica para diversos tópicos voltados à programação de computadores, processamento da informação e estudo de soluções algorítmicas propriamente dito.

Um dos conceitos fundamentais da Ciência da Computação é o algoritmo. Como citado nas primeiras seções desta unidade, um algoritmo determina um conjunto ordenado e não-ambíguo de passos que definem uma atividade finita. De forma geral, esta atividade finita visa a resolução de um problema.

Os algoritmos, na forma de programas, são executados em computadores. Para que um algoritmo seja executado no computador precisa ser representado de forma apropriada para que possa ser interpretado por uma máquina ou por um ser humano de forma correta e sem ambiguidades.

Estudos baseados em lingüística e gramática se preocupam pela tradução do algoritmo conceitual em um conjunto de comandos com uma sintaxe

⁵ Pseudocódigos: Sistema notacional nos quais idéias podem ser informalmente expressas durante o desenvolvimento de um algoritmo.

e semântica bem definidos através representação fornecida pelas linguagens de programação.

As linguagens de programação possibilitam escrever algoritmos num formato de fácil tradução para a linguagem da máquina, mas mantendo a familiaridade com a linguagem de **pseudocódigo**⁵ utilizada na descrição do algoritmo.

O desenvolvimento histórico das linguagens de programação modernas visa, a cada nova geração, elevar o nível de abstração de forma a propiciar cada vez mais independência da máquina. Nos primeiros dispositivos computacionais, os programas eram codificados no *hardware*, no interior da unidade de controle. Esta abordagem torna estes dispositivos muito inflexíveis, dada a dificuldade de alterar os programas.

Neste contexto surge o conceito de programa armazenado na memória principal, da mesma forma que os dados. Para isso, os programadores começaram a utilizar linguagem de máquina na escrita dos seus algoritmos. Este método era muito laborioso e complexo, uma vez que os códigos de operação e operandos eram representados através de dígitos numéricos, o que tornava a programação muito propensa a erros. O primeiro passo para melhorar a situação foi a utilização de nomes descritivos para as posições de memória e mnemônicos (em lugar de códigos hexadecimais), para expressar os códigos de operação.

A partir da utilização desta técnica, surgiu a necessidade de tradução dos programas escritos nesta notação para a linguagem de máquina. Surgiram assim as linguagens de montagem. Fica assim como responsabilidade do sistema operacional, a utilização da linguagem de montagem para traduzir o programa e, posteriormente, armazenando a versão traduzida para um uso posterior.

No entanto, o problema da dependência com a máquina continuava existindo, agora em relação à linguagem de montagem. Além disso, existia a dificuldade de programar em um nível muito baixo de abstração (ou seja muito detalhado), o que prejudicava o desenvolvimento do projeto da solução.

A necessidade de primitivas de alto nível propiciou o surgimento das linguagens de terceira geração, para o desenvolvimento de *software*, as quais por sua vez, foram projetadas com base nas primitivas de baixo nível já conhecidas. Por exemplo:

```
assign Total the value Valor + Imposto
```

Tendo como potencial primitiva:

```
assign <identificador> the value <expressão>
```

O desenvolvimento das linguagens de terceira geração conseguiu atender o objetivo de independência de plataforma de uma forma muito mais próxima ao raciocínio utilizado na análise e resolução de problemas.

Na medida em que um maior nível de abstração se torna uma necessidade surgem os chamados ambientes de programação, os quais possibilitam a utilização de conceitos mais abstratos na programação dos algoritmos.

A perspectiva nestes ambientes é que a máquina se adapta ao raciocínio humano, em contrapartida das linguagens de gerações anteriores, onde o ser humano precisava se adaptar à máquina. Uma propriedade inerente destes ambientes é a facilidade de utilização e a possibilidade de customização, sem auxílio de especialistas.

De forma geral, a programação nestes ambientes é facilitada pela utilização de menus e seleção de ícones na tela os quais representam atalhos de alto nível para a execução de instruções de nível mais baixo. Estes pacotes ou ambientes de programação são denominados de linguagens de quarta geração.

Finalmente, as linguagens classificadas como de quinta geração, tem como base principal a programação declarativa, e geralmente possuem um enfoque mais especializado (programação lógica).

Saiba Mais



Um Sistema de Informação (SI), coleta, processa, armazena, analisa e dissemina informações com o propósito específico. Como qualquer outro sistema, um sistema de informação abrange entradas (dados) saídas (relatórios, cálculos), processa essas entradas e saídas e gera saídas que são enviadas para o usuário ou outros sistemas. É possível incluir um mecanismo de resposta feedback – que controle a operação. E como qualquer outro sistema, um sistema de informação opera dentro de um ambiente.

Atividades de avaliação



1. O que significa dizer que uma linguagem de programação é independente de máquina?
2. O que significa que uma linguagem possui um nível mais alto que as gerações anteriores?
3. Escreva um algoritmo em pseudocódigo para realizar alguma atividade doméstica.

Tecnologia da Informação

Antes mesmo da popularização dos computadores, existia uma noção da importância do cuidado e tratamento da informação como uma forma de preservar o histórico, para que, eventualmente, possa ser utilizado na análise de fatores do passado para o entendimento do presente e, principalmente, na tomada de decisão de ações futuras. Deste este ponto de vista, a informação pode ser interpretada como um elemento que pode adicionar certo grau de competitividade às organizações.

Antigamente, os sistemas de informação nas organizações se baseavam basicamente em técnicas de arquivamento e recuperação de informações de grandes arquivos de forma manual, sob a responsabilidade de um *arquivador*. O processo era relativamente simples, porém muito trabalhoso, tornando difícil e propensa a erros a recuperação da informação e principalmente a atualização dos dados.

Com o avanço da tecnologia, no século XX a informação começou a fluir e se multiplicar a velocidade assustadora. Era o advento da Era da Informação. Um marco desta nova fase é a invenção do telégrafo elétrico em 1837, e a partir dali o desenvolvimento de meios de comunicação de massivos, e até mais recentemente, o surgimento da grande rede de comunicação de dados que é a Internet. Esta globalização propiciou um crescimento exponencial do volume de dados disponíveis, tornando um desafio a coleta, organização, catalogação e posterior recuperação de forma eficiente e segura da informação.

No ambiente complexo e turbulento que nos encontramos é essencial para as organizações o estímulo ao resgate do conhecimento organizacional como auxílio no processo de decisão e na busca de vantagens competitivas. Como consequência do processo de globalização, o contexto mundial, em permanente mudança, vive permanentemente submetido às implicações decorrentes de grandes transformações políticas, econômicas e tecnológicas. Neste cenário, a informação passa a ser considerada um recurso essencial nas tomadas de decisão. Para Beck (2007), “atualmente, toda empresa está envolta com amplos e diversos tipos de informação e, para competir neste

contexto dinâmico, o segredo do sucesso é a agregação de valor a partir do acesso, do tratamento, da utilização e da disseminação da informação”.

A partir desta premissa surge a necessidade nas empresas de gerenciar o conhecimento com base num novo modelo de tecnologia de informação e gestão de forma a propiciar a distribuição da informação com o intuito de atingir competitividade.

Um SI é um sistema tem seu foco na informação. Seu objetivo é armazenar, tratar e fornecer informações de tal modo a apoiar as funções ou processos de uma organização.

Geralmente, um SI é estruturado em dois sub-sistemas:

- Sub-sistema social que inclui as pessoas, processos, informações e documentos.
- Sub-sistema automatizado, que consiste dos meios automatizados (máquinas, computadores, redes de comunicação) que interligam os elementos do sub-sistema social.

O sub-sistema automatizado consiste em um conjunto de recursos tecnológicos e computacionais para geração e uso da informação de forma a viabilizar a [Gestão do Conhecimento](#)⁶ para a tomada de decisão e a elaboração de estratégias a partir de novos bancos de dados originados dos bancos de dados operacionais da empresa como, por exemplo, o *data warehouse*, o *data mining*, entre outros.

Essas ferramentas, além de dar subsídios para a tomada de decisão, estão voltadas à disseminação do conhecimento na organização, seja através de melhores práticas de trabalho, seja através do registro de discussões para a disseminação de conhecimento, como por exemplo o *Groupware* e a *Intranet*.

Esse conjunto de recursos é chamado de Tecnologia da Informação (TI). As aplicações de TI são diversas e estão ligadas às mais diversas áreas, de forma a automatizar os processos organizacionais. Através da TI, pequenas e grandes empresas podem alcançar maior produtividade e competitividade, alcançando o sucesso e alavancando maiores rendimentos.

A TI exerce um importante papel no âmbito da gestão do conhecimento reduzindo distâncias e propiciando a transferência do conhecimento de forma *on-line*. No domínio da gestão do conhecimento, o benefício principal da utilização de recursos computacionais é a sua capacidade de armazenamento e comunicação de computadores ligados em rede fazendo deles propiciadores e disseminadores de conhecimento (DAVENPORT e PRUSAK, 1998).

⁶ Gestão de Conhecimento: envolve as atividades de coleta, seleção, disponibilização e disseminação de informações.

A TI envolve a organização de recursos computacionais tais como *hardware*, *software* e mecanismos de comunicação, em um sistema capaz de executar um conjunto de tarefas. Além dos recursos computacionais, tecnologias relativas ao planejamento e desenvolvimento de sistemas, suporte ao *software*, e aos processos de produção e operação, também são considerados elementos de TI.

Por outro lado, tecnologias intelectuais tais como métodos, teorias e metodologias utilizadas para lidar com o ciclo da informação, também são consideradas sustento da TI. Neste conjunto podem ser consideradas, por exemplo, técnicas de classificação, que não necessariamente requerem uso de máquinas, mas apenas um esquema para serem aplicadas.

Técnicas que se mostraram eficientes geralmente são automatizadas pela elaboração de um algoritmo ou *software*, de forma a facilitar a sua aplicação sistemática. Seguindo o exemplo das tecnologias de classificação e organização de informações, a sua origem se remonta ao surgimento das primeiras bibliotecas. Hoje em dia praticamente toda biblioteca tem seu acervo gerenciado através de um sistema computacional.

O sucesso e difusão dos sistemas de informação alcançados através da rápida evolução e surgimento de novas Tecnologias de Informação fazem com que praticamente todas as atividades desenvolvidas na sociedade atual sejam sustentadas eficientemente através dos recursos da informática, e com custos cada vez mais reduzidos.

O processamento de informação é uma atividade central em diversas áreas, tais como economia, transportes, produção, imprensa, comunicação. A necessidade pela difusão social da informação de domínio público ou privado em larga escala de transmissão é possível somente através de sistemas tecnológicos inteligentes.

Uma consequência do desenvolvimento tecnológico é a intensificação da comunicação pessoal e institucional, através de programas de processamento de texto, de formação de bancos de dados, de editoração eletrônica, bem de tecnologias que permitem a transmissão de documentos, envio de mensagens e arquivos, assim como consultas a computadores remotos (via rede mundial de computadores, como a internet).

Em contrapartida aos enormes benefícios atingidos pela utilização da TI nos diversas atividades, alguns problemas podem ser apontados em relação à privacidade dos indivíduos e ao seu direito à informação, pois os cidadãos geralmente não tem acesso a grande quantidade de informação sobre eles, coletadas por instituições particulares ou públicas.

Embora a TI sustente os processos de Gestão do Conhecimento, a eficácia final destes processos depende da interpretação deste conhecimento pelos homens, e sua transformação em ações. Sendo assim, a tecnologia da informação, isoladamente, não pode ser a base da criação e gestão de conhecimento organizacional, mas o suporte dentro desse processo.

Síntese do capítulo



Neste capítulo foi apresentado o histórico da evolução dos computadores, a partir da descrição dos dispositivos mecânicos precursores dos computadores utilizados atualmente. Foi feita uma breve retrospectiva das gerações que determinam os estágios da evolução dos computadores até os dias de hoje.

Neste capítulo também foram apresentados os principais conceitos que definem a Ciência da Computação, apresentando brevemente as principais disciplinas que a compõem. Para concluir foi apresentada a importância da Tecnologia de informação (TI) no processo de construção e disseminação do conhecimento nas organizações, de forma a aumentar o alcance e a velocidade da transferência do conhecimento.

Leituras, filmes e sites



<http://www.cotianet.com.br/bit/hist/Default.htm>

Charles Babbage Institute <http://www.cbi.umn.edu/>

<http://www.daniel.prof.ufu.br/icc.html>

Referências



TANENBAUM, ANDREW S. **Organização Estruturada de Computadores**. 5. Edição. São Paulo: Editora Pearson. 2007.

STALLINGS WILLIAM. **Estrutura e Organização de Computadores**. 5. Edição. São Paulo: Editora Pearson. 2004.

BROOKSHEAR GLENN J. **Ciência da Computação – Uma visão abrangente**. 5. Edição. São Paulo: Editora Bookman. 2000.

Parte

2

Representação e processamento da informação

Representação e processamento da informação

Introdução

Neste capítulo serão apresentadas as formas de representação das informações nos computadores. Assim como o ser humano tem a fala e a escrita para representar as informações, o computador tem a sua própria representação para que possa tratá-las e processá-las. Iniciamos apresentando a unidade básica de representação de informação no computador, o *bit*, e mostramos também a representação de bytes (bloco de bits), de letras e números.

Em seguida apresentamos uma introdução ao sistema binário e a forma como os números são representados e mostramos como realizamos as operações matemáticas em um computador. Finalmente apresentamos uma introdução à lógica digital mostrando os tipos mais comuns de portas lógicas, que estabelecem a forma de como um computador pode processar as informações.

Para um computador qualquer informação deve ser codificada com uma linguagem apropriada para seu mundo, que é diferente do mundo dos humanos. Para o mundo eletrônico o ideal é que essa linguagem tenha dois estados, ligado e desligado, ou simplesmente, “0” e “1”. Por isso a base mais importante para o computador é a base 2, daí o nome binário.

Vamos começar com algumas definições sobre a representação de informações em um computador.

- **bit**: unidade básica de representação digital.
- **byte**: conjunto de 8 bits.
- **nybble**: um bloco de 4 bits.
- **palavra** ou **word**: conjunto de 32 bits
- **meia-palavra** ou **halfword**: um bloco de 16 bits
- **palavra-dupla** ou **doubleword**: conjunto de 64 bits

A definição de “palavra” não está claramente definida. Ela geralmente se refere ao número de bits usados simultaneamente em um computador.

Atualmente a maioria dos computadores usa uma palavra de 32 bits. No entanto, cada vez mais, arquiteturas de 64 bits estão sendo construídas o que muda o tamanho da palavra. Entretanto, por ser o mais comum, ao longo desse texto usaremos palavra com 32 bits.

1. Identificando os bits

É conveniente identificarmos cada bit individualmente para que possamos referenciá-los pelo nome. Daremos um rótulo para cada bit, da direita para a esquerda, sendo 0 (zero) o índice mais à direita do bloco.

Por exemplo, suponha que você tenha um número com n -bits. Poderíamos identificá-los como: $b_{n-1}, b_{n-2}, \dots, b_1, b_0$.

Também é comum definir mais dois termos para um bloco com n -bits:

- **bit menos significativo** ou *least significant bit*. Bit b_0 . É comum abreviar como LSb com o 'b', em minúsculas para representar bit. (Quando estiver em letras maiúsculas, isso significará byte menos significativo).
- **bit mais significativo** ou *most significant bit*. Bit b_{n-1} . É comum abreviar como MSb com o 'b', em minúsculas para representar bit. (Quando estiver em letras maiúsculas, isso significará byte mais significativo).

Vale lembrar que o termo "menos significativo" não significa que ele seja "menos importante". Na verdade essa denominação está relacionada com o menor índice de sua posição. Todos temos ciência que alterar esse bit (ou seja, mudá-lo de 0 para 1, ou 1 para 0) iria afetar o valor global assim como qualquer outro *bit*. Da mesma forma, o *bit* "mais significativos" não significa "mais importante", mas apenas que seu índice é o maior do bloco.

2. Strings e vetores

Embora existam diferentes padrões para definir os tamanhos dos blocos de bits (*bits, bytes, nybbles, etc*), às vezes é necessário fazer referência a um conjunto de *bits* mais arbitrário. Iremos fazer referência a um conjunto de " N " bits como um *string*⁷ de bits ou um vetor.

Pode parecer estranho para se referir a um número como uma *string*, mas não é tão incomum. Quando escrevemos números, nós costumamos alterar o número de dígitos, conforme necessário. Assim, você escreve 10, em vez de 0010 porque os zeros adicionais não têm significância. Porém, em *hardware*, temos um número fixo de *bits* para a palavra como, por exemplo, 32 *bits* ou 64 *bits*. Nesse caso acrescentamos zeros a direita para preencher o número necessário para completar de número total de *bits*.

⁷ String: conjunto de bits de tamanho arbitrário que representam números ou palavras.

Então, o nome de *string de bits* pode ser bom, porque pode ser usado para qualquer comprimento de uma string. São igualmente *string de bits* os chamados *vetores de bit*. Neste caso, um vetor geralmente significa uma "coordenada" com n argumentos. Por exemplo, quando você trabalha com gráficos em 2 ou 3 dimensões, as coordenadas são escritas como (2,3) ou (2,3,1), então, 0010 poderia significar (0,0,1,0), ou seja, uma coordenada em quatro dimensões.

Claro que, quando as pessoas falam de vetores eles geralmente não pensam nisso como uma coordenada, mas sim em uma sequência de valores com posições numeradas. Em todo o caso, vamos chamá-lo simplesmente de *strings*.

2.1. Quantos diferentes *strings* podemos escrever?

Suponha que você tenha um *string* de n -bits. Quantas combinações possíveis de 1 e 0 existem? E por que isso interessa?

Para calcular o número de combinações de string de n -bits, percebem que temos 2 opções para cada bit. Simplesmente basta multiplicar 2, n vezes. Isto é, 2^n .

Este é um fato muito importante para se lembrar, pois o número de possibilidades de um string limita o número de possíveis valores que ele pode representar. Por exemplo, se você tem 3 bits, então você tem 8 possíveis valores para ser representado pelo *string* (8 = 2³). Isso limita a um máximo de 8 valores diferentes.

3. Representação de dados

Imagine que você está em um teclado, no entanto, é um teclado horrível. Esse teclado tem apenas duas teclas para se usar: 'a' e 'b'. É impossível escrever uma mensagem para alguém usando apenas duas teclas, afinal, quantas palavras só usam as letras 'a' e 'b'? Não muitas.

No entanto, ainda é possível enviar mensagens. Como? Você pode codificar a mensagem utilizando um código, por exemplo, o código ASCII⁸. O código ASCII atribui um número a cada letra ou símbolo do teclado. Por exemplo, 'A' é codificado para o valor numérico 65 (na base 10) ou 0100 0001 (na base 2). Computadores codificam em binário, o que significa, que todas as letras podem ser codificados em 0's e 1's, mais simples para um computador entender. Se considerarmos 'a' como 0 e 'b' como 1, então você pode codificar a letra A como "abaa aab" ou "0100 0001" em binário.

⁸ ASCII: acrônimo de American Standard Code for Information Interchange, padrão de representação binária de letras e números.

Uma das coisas mais importantes para aprender sobre o *hardware* são as formas de como os dados são codificados em 0 e 1. Isto é chamado de representação de dados.

Além de teclas ou caracteres, ainda existem outras informações básicas que têm de ser codificadas: os números. Aparentemente parece trivial, basta converter o número decimal no seu equivalente binário, e aí, efetuar todas as operações. Para representar números inteiros positivos isso funciona, mas inteiros negativos, frações, números ponto flutuante?

Um dado pode ser representado nas seguintes formas:

- Inteiros sem sinal (inteiros positivos)
- Inteiros com sinal (inteiros positivos, negativos e zero)
- Números de ponto flutuante (aproximações de números reais positivos e negativos)
- Caracteres (ASCII, Unicode)

É importante lembrar que é impossível representar qualquer número real no computador porque há a limitação no tamanho do número representado, por exemplo, tamanho da mantissa e do expoente, obrigando a armazenar uma aproximação do valor. Esse “arredondamento” pode causar imprecisões nos cálculos realizados em um computador.

Podemos dizer que, de todas as coisas que você aprender sobre o computador, a representação de dados será provavelmente a mais importante. A maioria dos formandos em Computação provavelmente nunca irão projetar um *hardware*, mas todos deverão ter conhecimento de como os dados serão armazenados.

Atividades de avaliação



1. Pesquise outros tipos de representação de números e caracteres.

Sistemas de Numeração

Um sistema de numeração é um conjunto de caracteres e regras matemáticas usadas para representar um número.

Os sistemas numéricos nasceram com o início de nossa civilização. Alguns exemplos são o sistema numérico romano (numerais romanos), o sistema numérico chinês, o sistema numérico grego, e o sistema numérico árabe. Nossos números contemporâneos são baseados nesse último.

O sistema mais usado é a base 10 ou o sistema decimal, que usa uma base 10 para operações e é conseqüentemente muito simples se usar (que nós temos usado desde o pré-escolar!). Os outros sistemas são o sistema hexadecimal e o sistema binário. Estes sistemas usam a base 16 e a base 2, respectivamente. Os sistemas hexadecimais e binários serão discutidos a seguir.

Por que nós usamos a base 10? Olhe suas próprias duas mãos e conte seus dedos! A palavra “dígito” se origina de *digitus*, palavra latina que significa “dedo”. Nós todos aprendemos na escola que cada dígito em um número corresponde ao valor de seu lugar. Por exemplo, 245 seria $(2 \times 100) + (4 \times 10) + (5 \times 1) = 245$. Graficamente temos:

100	10	1
2	4	5

Como nós observamos, cada valor da coluna representa 10 vezes o valor da coluna à direita dele. Por isso chamamos base 10.

1. O sistema binário

O sistema binário⁹ tem uma base 2, significando que do valor de cada coluna é 2 vezes o valor da coluna à direita, começando com o dígito mais a direita, com o valor igual a 1. Usamos apenas os dígitos 0 e 1. Por exemplo, o número 111101101 representado graficamente temos:

⁹ Sistema binário: sistema numérico posicional com base 2.

256	128	64	32	16	8	4	2	1
1	1	1	1	0	1	1	0	1

Assim nós temos:

$$(1 \times 256) + (1 \times 128) + (1 \times 64) + (1 \times 32) + (0 \times 16) + (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) = 4096 + 256 + 128 + 64 + 32 + 8 + 4 + 1 = 493$$

Assim, 111101101 no sistema binário é igual a 493 no sistema decimal.

O sistema binário é útil nos computadores porque nos circuitos eletrônicos temos dois estados distintos, ligado ou desligado, representando 1 ou 0. Outra vantagem do sistema binário é porque o transistor (que executa a função de chave no circuito eletrônico) consome um mínimo de energia quando ele funciona nos estados de corte e saturação, respectivamente 0 ou 1. Isso permite circuitos mais densos e ocupando menos espaços, possibilitando dispositivos mais compactos, mais velozes e mais econômicos (usando menos pilha, por exemplo).

1.2. Convertendo binário-decimal-binário

O procedimento acima pode ser utilizado para converter um número binário em número decimal. Vamos mostrar o exemplo do número 1100110 que é igual a $64 + 32 + 0 + 0 + 4 + 2 + 0 = 102$. Graficamente temos:

512	256	128	64	32	16	8	4	2	1
0	0	0	1	1	0	0	1	1	0

Apresentamos o método para converter números de binário ao decimal. Como podemos fazer o inverso? Vamos mostrar a conversão do número 500 em binário. Utilizando nosso diagrama de potências de dois, devemos escrever os valores nos lugares da esquerda para a direita até completar o valor desejado.

Como 500 é menor que 512 o primeiro dígito será 0. Escrevemos então, 0 sob o valor 512.

512	256	128	64	32	16	8	4	2	1
0									

A partir de agora, o esquema é simples: testar o lugar imediatamente à direita. Se é menor ou igual ao número, escrever um 1 sob ele, subtrair o valor do número e usar o resultado na posição seguinte. Se é maior, escrever embaixo um 0, não subtrair nada e ir para a posição seguinte. Na tabela acima,

256 é menor que 500, assim escrevemos um 1 sob 256 e subtraímos 256 de 500. Agora vamos usar o resultado (244) na posição seguinte. Assim temos:

512	256	128	64	32	16	8	4	2	1
0	1								

128 é menor que 244, assim escrevemos um 1 sob 128 e subtraímos 128 de 244. Usamos o resultado (116) na posição seguinte.

512	256	128	64	32	16	8	4	2	1
0	1	1							

E assim por diante até zerar o número:

512	256	128	64	32	16	8	4	2	1
0	1	1	1	1	1	0	1	0	0

Simple não? Com esse algoritmo simples, podemos converter um número decimal em um número binário. 500 em decimal é igual a 111110100 em binário.

2. O sistema hexadecimal

O sistema hexadecimal¹⁰ têm uma base 16, significando que cada coluna é igual a 16 vezes a coluna a sua direita, começando com o valor 1 na coluna mais a direita. Com isso temos:

65536	4096	256	16	1
0	4	A	3	8

O sistema hexadecimal, além dos dígitos 1 a 9, usa as letras A, B, C, D, E, F; que representam 10, 11, 12, 13, 14 e 15, respectivamente. No exemplo acima, o número 04A32 vale $(4 \times 4096) + (10 \times 256) + (3 \times 16) + (8 \times 1) = 19000$.

Por que usar um número hexadecimal? É muito complicado para um humano trabalhar com números binários, pois são muito longos e fáceis de se confundir. A conversão de hexadecimal para binário é direta (16 é potência de 2), assim, cada 4 dígitos binários representam um dígito hexadecimal conforme mostramos abaixo:

¹⁰ Sistema hexadecimal: Sistema numérico com base 16.

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Veja como é fácil converter o valor hexadecimal acima em binário:

65536	4096	256	16	1
0	4	A	3	8
0000	0100	1010	0011	0100

O número hexadecimal 4A38 é igual a 0100101000110100 em binário.

2.1. Convertendo hexadecimal-decimal-hexadecimal

Vamos converter o número hexadecimal 7EF. Recorde que o dígito mais a direita é igual a um e cada dígito à esquerda é igual a dezesseis vezes o dígito imediatamente a sua direita. Colocando na tabela temos:

256	16	1
7	E	F

Multiplicamos agora cada dígito hexadecimal pelo número em seu cabeçalho (lembre-se que o A-F representa 10-15 respectivamente)

$$256 \times 7 + 16 \times 14 (E) + 1 \times 15 (F)$$

Totalizando o resultado, temos:

$$1792 + 224 + 15 = 2031$$

Assim chegamos ao valor final: 7EF em hexadecimal é igual a 2031 em decimal.

Converter do decimal para hexadecimal é um pouquinho mais complicado. Para este exemplo usaremos 293. Inicialmente escrevemos a tabela com os valores em decimal na ordem da direita para a esquerda. Cada dígito a esquerda deve ter o valor de 16 vezes maior que o da direita, começando com 1.

4096	256	16	1
0			

Colocamos um 0 sob o valor 4096 da tabela que é maior do que o número a ser convertido, 2031. Dividimos o número a ser convertido por 256, que é o valor mais elevado da tabela e menor do que o número a ser convertido. Neste caso, 293 dividido por 256 igual a 1, com um resto 37. Em seguida, escrevemos o resultado (1) e usamos o resto (37) para o cálculo seguinte.

4096	256	16	1
0	1		

Dividir agora o resto do último cálculo (37) pelo valor seguinte (16). 37 por 16 igual a 2 com um resto 5. Escrevemos o resultado (2) sob 16 e dividimos o resto (5) pelo valor seguinte (1)

4096	256	16	1
0	1	2	

5 dividido por 1 igual a 5 sem resto, assim colocamos o valor (5) sob 1 e está concluído. Assim:

4096	256	16	1
0	1	2	5

Então 293 decimal é igual a 125 em hexadecimal.

Se por acaso em alguns dos cálculos o resultado não tiver nenhum resto, colocamos simplesmente o resultado sob o respectivo valor e colocamos 0 sob todos os valores restantes à direita.

Atividades de avaliação



1. Converta o número 10110011 binário em hexadecimal e decimal.
2. Converta o número 863 decimal em binário e hexadecimal.
3. Converta o número 37D2 hexadecimal em binário e decimal.

Aritmética binária

As operações aritméticas¹¹ são possíveis em números binários assim como nos números decimais. De fato os procedimentos são completamente similares em ambos os sistemas, assim como em qualquer base de numeração. A multiplicação e a divisão não são realmente difíceis, mas como a não familiaridade com os números binários causa bastante dificuldade para os humanos. Entretanto, como iremos mostrar, para os computadores é extremamente fácil.

¹¹ Aritmética binária: operações com dois ou mais números na representação binária sem necessidade de conversão para outra base

1. Aritmética de Números Inteiros Positivos

1.1. Adição

Quando somamos números decimais, somamos os algarismos de mesma ordem e caso ultrapasse o valor da base, nesse caso 10, acrescentamos uma unidade à ordem seguinte. No caso de adição binária, devemos considerar a base 2.

A adição é quase tão fácil quanto “um mais um é igual a dois”. Na verdade, em binário, “um mais um igual a zero e vai um”. Se convertermos “10” para decimal teremos o valor “2”, que está correto. Também temos que “zero mais zero igual a zero” e “um mais zero é igual a um”.

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 1 &= 10 \end{aligned}$$

Observar que ao adicionar dois “1” produz um resultado com dois dígitos, significa que nós temos um “vai um” para a posição seguinte, da mesma forma como fazemos na adição decimal. Vejam o exemplo abaixo de uma soma binária e a mesma soma em decimal.

$$\begin{array}{r} 11 \\ + 101 \\ \hline 1000 \end{array} \quad \begin{array}{r} 3 \\ + 5 \\ \hline 8 \end{array}$$

As regras de adição binária são mostradas abaixo:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 0, \text{ e soma } 1 \text{ ao bit seguinte} \end{aligned}$$

Mais alguns exemplos:

$\begin{array}{r} 11 \\ 00011010 \\ +00001100 \\ \hline 00100110 \end{array}$	$\begin{array}{r} 26 \\ + 12 \\ \hline 38 \end{array}$	$\begin{array}{r} 11111 \\ 00010011 \\ + 00111110 \\ \hline 01010001 \end{array}$	$\begin{array}{r} 19 \\ + 62 \\ \hline 81 \end{array}$
---	--	---	--

1.2. Subtração

A subtração é semelhante à adição, sendo que agora, se não houver quantidade para subtrair fazemos um “empréstimo” da ordem acima.

$\begin{array}{r} -1 \\ 101 \\ 11 \\ \hline 010 \end{array}$	$\begin{array}{r} 5 \\ - 3 \\ \hline 2 \end{array}$
--	---

As regras de subtração binária são mostradas abaixo:

$$\begin{aligned} 0 - 0 &= 0 \\ 0 - 1 &= 1, \text{ e pede-se } 1 \text{ ao bit seguinte} \\ 1 - 0 &= 1 \\ 1 - 1 &= 0 \end{aligned}$$

Mais alguns exemplos:

$\begin{array}{r} -1 \\ 00100101 \\ - 00010001 \\ \hline 00010100 \end{array}$	$\begin{array}{r} 37 \\ - 17 \\ \hline 20 \end{array}$
$\begin{array}{r} -1 \\ 00110011 \\ 00010110 \\ \hline 00011101 \end{array}$	$\begin{array}{r} 51 \\ - 22 \\ \hline 29 \end{array}$

1.3. Multiplicação

A multiplicação consiste em uma série de somas. Operamos da mesma forma que uma multiplicação na base 10.

$$\begin{array}{r}
 101 \\
 \times 10 \\
 \hline
 000 \\
 101 \\
 \hline
 1010
 \end{array}
 \qquad
 \begin{array}{r}
 5 \\
 \times 2 \\
 \hline
 10
 \end{array}$$

As regras de multiplicação binária são mostradas abaixo:

$$\begin{array}{l}
 0 \times 0 = 0 \\
 0 \times 1 = 0 \\
 1 \times 0 = 0 \\
 1 \times 1 = 1, \text{ e soma } 1 \text{ ao bit seguinte}
 \end{array}$$

Mais alguns exemplos:

$$\begin{array}{r}
 00101001 \\
 \times 00000110 \\
 \hline
 00000000 \\
 00101001 \\
 00101001 \\
 \hline
 0011110110
 \end{array}
 \qquad
 \begin{array}{r}
 41 \\
 \times 6 \\
 \hline
 246
 \end{array}$$

$$\begin{array}{r}
 00010111 \\
 \times 00000011 \\
 \hline
 00010111 \\
 00010111 \\
 \hline
 001000101
 \end{array}
 \qquad
 \begin{array}{r}
 23 \\
 \times 3 \\
 \hline
 69
 \end{array}$$

Podemos observar que a primeira parcela da multiplicação é repetida em toda as linhas quando houver 1 na outra parcela e zero quando o algarismo é 0. Isso sugere que podemos realizar uma multiplicação através de uma série de somas de uma parcela deslocada conforme a segunda parcela. Podemos fazer um circuito eletrônico de multiplicação juntando um somador e um deslocador de bits.

1.4. Divisão

A divisão consiste em uma série de subtrações binárias, semelhante a uma divisão na base 10.

$$\begin{array}{r}
 110 \qquad \qquad \qquad 6 \\
 10 \qquad \qquad \qquad 2 \\
 \hline
 110 \qquad \qquad \qquad 3 \\
 -10 \quad (10) \\
 \hline
 010 \\
 -10 \quad (01) \\
 \hline
 000 \quad (11)
 \end{array}$$

2. Aritmética de Números Inteiros Negativos

As coisas são fáceis enquanto fazemos operações com inteiros positivos. Mas as coisas tornam-se mais complicada quando nós queremos fazer operações com números inteiros positivos e negativos.

Inicialmente podemos pensar em fazer como na aritmética decimal, indicando com um sinal especial (– ou +) se o número é negativo ou positivo. Na aritmética binária, nós poderíamos simplesmente reservar um bit para determinar o sinal. Nos circuitos para a adição, nós teríamos um circuito para adicionar dois números e outro para subtrair dois números. A combinação de sinais nas duas entradas determinaria que circuito a se usar com os valores absolutos, e também o sinal da saída.

Apesar deste método funcionar, existe uma forma muito mais fácil de operar aritmética com números negativos em circuitos eletrônicos. Este método é chamado “método do complemento a dois”. Com esse método não é necessário um circuito especial para tratar os sinais e podemos usar um único circuito somador padrão para operar números positivos e negativos. A fim de explicar este método vamos mostrar inicialmente como se trabalharia na aritmética decimal para depois mostrar como trabalhar com aritmética binária.

2.1. Complemento de dez

Imagine o odômetro de um automóvel. Ele tem um determinado número de rodas, cada uma com os dez dígitos gravados. Quando uma roda se mover de 9 a 0, a roda imediatamente à esquerda avança uma posição. Vamos supor que o automóvel ande de ré. Então o reverso acontece, isto é quando uma

roda vai de 0 a 9, a roda a sua esquerda diminui uma posição. Quando todas as rodas são 0, nós interpretamos o valor como o inteiro 0 (zero).

Vamos supor agora que nós temos um circuito de adição que possa armazenar números inteiros com 4 dígitos (imagine um odômetro com 4 dígitos). Vamos realizar a soma $(-2) + 5 = 3$. Vamos considerar que a sequência 0000 seja igual a 0. Nesse circuito vamos representar o valor (-2) com a sequência 9998, isto é, diminuimos 2 unidades de 0000, e o valor (5) com 0005, para então, realizamos a soma tradicional. Primeiramente adicionamos 8 e 5 que dá 3 e vai 1. Em seguida, adiciona 9 e 1, dando 0 e vai 1. Para os demais, o valor será 0 com vai 1, assim que o resultado final será 0003 positivo. Este resultado é correto, tanto pelo valor como pelo sinal.

É importante observar que na representação de complemento de 10 com uma quantidade limitada de dígitos, não podemos usar toda a faixa de valores. Nesse exemplo, os valores 0001 a 4999 representam números inteiros positivos, enquanto os valores 5000 a 9999 representam números negativos, respectivamente de -4999 a -1 . Outro problema é o estouro do registro, se somarmos 1 ao número 4999 não teremos 5000, mas -4999 . Um circuito extra pode detectar essa ocorrência e corrigir o valor.

Vamos terminar esta seção dando um método simples para computar o valor absoluto de um inteiro nessa representação. Basta tomar cada dígito individual, substituí-lo por 9 menos o seu valor original, e então no final, adiciona 1 ao número obtido. Assim por exemplo, o número 9998 transforma-se em 0001 que somando 1 fica 0002 negativo, que é a resposta correta.

2.2. Complemento de dois¹²

Até agora, nós estudamos a representação de números negativos usando o complemento de dez. Em um computador, nós usamos a base dois. Felizmente, o método descrito na seção anterior também funciona para a base dois. Para um somador de n -bits (n é geralmente 32 ou 64), nós podemos representar números positivos com o dígito na posição mais a esquerda seja 0, que dá valores entre 0 e $2^{(n-1)} - 1$, e números negativos com o dígito na posição mais a esquerda seja 1, que dá valores entre $-2^{(n-1)}$ e -1 . A vantagem do complemento a dois é que o dígito a esquerda pode também indicar o sinal do número, 0 para números negativos e 1 para números positivos.

Da mesma forma como no caso complemento de dez, usamos o mesmo princípio para a detecção do estouro. Se, ao adicionar dois números positivos, nós encontramos um resultado negativo (isto é, com o bit mais à esquerda 1), nós teremos um estouro de valor (*overflow*). Da mesma forma, se, ao

¹² Complemento de dois: Técnica para possibilitar a soma e subtração de números positivos e negativos na representação binária

adicionar dois números negativos, encontramos um resultado positivo (isto é com seu bit mais à esquerda 0), temos um sub-valor (*underflow*).

3. Números de ponto flutuante

A representação numérica que ainda falta são dos números racionais. Apesar de ser possível a representação de números racionais (por exemplo, representar um número com o seu numerador e denominador), a maioria dos computadores utiliza a representação de um subconjunto dos números racionais, que atende a maioria dos casos. Nós chamamos estes números de *ponto flutuante*.

A idéia básica atrás dos números de ponto flutuante é representar um número com a sua *mantissa* e um *expoente*, cada um com um número fixo de bits referentes a sua precisão. Se nós denotamos a mantissa com m e o expoente com e , o número representado será equivalente a $m \cdot 2^e$.

O uso de ponto flutuante é mais importante para aplicações científicas ou aplicações que exijam muitos cálculos matemáticos como computação gráfica. Nessas aplicações precisamos *multiplicar* ou *dividir* dois números mais frequentemente. A multiplicação de dois números de ponto flutuante é fácil, basta multiplicar as mantissas e somar os expoentes. A divisão é somente um pouco mais complicada. Observe que a imprecisão no resultado de uma multiplicação ou de uma divisão é somente devido à imprecisão nos operandos originais. Nenhuma imprecisão adicional é introduzida pela operação própria (exceto possivelmente 1 unidade em menos dígito significativo).

Para somar dois números de ponto flutuante, o expoente menor deve primeiramente ter sua mantissa deslocada para a direita para igualar os expoentes. Se o deslocamento for maior do que o número de bits na representação da mantissa, o segundo número será tratado como 0.

Geralmente, os computadores manipulam dados em blocos de **8**, **16**, **32**, **64**, ou **128** bits. É conseqüentemente útil caber um único número de ponto flutuante com a mantissa e o expoente em tal bloco. Em tal bloco precisamos ter um bit para o sinal, a mantissa, e o expoente.



Desde 1980 os computadores adotam uma representação padrão para números em ponto flutuante chamada IEEE 754 para facilitar a portabilidade de programas. Os números podem ter precisão simples, constituído com 1 bit de sinal, 8 bits de expoente e 23 bits de mantissa mais 1 implícito, $t_o = 24$,

precisão dupla = 1 bit de sinal, 11 bits de expoente e 52 bits de mantissa + 1 implícito = 53).

Precisão	Sinal	Expoente	Mantissa	Implícito	Total (bits)
Simplex	1	8	23	1	32
Dupla	1	11	52	1	64

Atividades de avaliação



1. Converta os números decimais 19 e 32 em binários, calcule a soma deles em binário e converta novamente para decimal para conferir o resultado.
2. Converta os números decimais 14 e -8 em binários complemento de dois, calcule a soma deles em binário e converta novamente para decimal para conferir o resultado. (dica: use um número de 5 bits).

Portas lógicas

Os blocos básicos de um computador são chamados *portas lógicas* ou apenas *portas*.

As portas são os circuitos básicos que têm pelo menos uma (e geralmente mais) *entradas* e exatamente uma *saída*. Os valores da entrada e da saída são os valores lógicos verdadeiro ou falso. Na convenção de computação é comum usar 0 para falso e 1 para verdadeiro. As portas não têm nenhuma *memória*. O valor da saída depende somente do valor atual das entradas. Este fato torna possível usar uma *tabela de verdade* para descrever inteiramente o comportamento de uma porta.

Nós consideramos geralmente três tipos básicos das portas, as portas E, as portas OU, e as portas NÃO (ou inversores). Também é usual chamar as portas pelo termo em inglês, respectivamente, AND, OR e NOT.

1. Portas básicas¹³

Portas básicas são portas fundamentais que podem formar outras portas com funções especiais. As portas básicas são: E, OU e NÃO.

1.1. Porta E (AND)

Uma porta E pode ter um número arbitrário de entradas. O valor da saída é 1 se e somente se todas as entradas são 1. Se não o valor da saída é 0. O nome foi escolhido porque a saída é 1 se e somente se a primeira entrada e a segunda entrada, e, ..., e a n-ésima entrada são todas 1.

É frequentemente útil desenhar diagramas das portas e das suas interconexões. Em tais diagramas, a porta E é desenhada como esta:



Figura 6 - Ilustração da porta E

¹³ Portas básicas: dispositivo que implementa funções lógicas básicas para todas as funções lógicas.

A tabela de verdade para uma porta E com duas entradas é mostrada abaixo:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

1.2. Porta OU (OR)

Como a porta E, a porta OU pode ter um número arbitrário de entradas. O valor da saída é 1, se e somente se, pelo menos de um dos valores da entrada é 1. Caso contrário, a saída é 0. Ou seja o valor da saída é 0 somente se todas as entradas são 0. O nome foi escolhido porque a saída é 1, se e somente se, a primeira entrada ou a segunda entrada, ou, ..., ou a n-ésima entrada são 1.

Em esquemas de circuito, nós desenhamos a porta OU como este:



Figura 7 - Ilustração da porta OU

A tabela de verdade para uma porta OU com duas entradas olha como esta:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

1.3. Porta NÃO (NOT)

A porta NÃO também é conhecida como *inversor* e tem exatamente uma entrada e uma saída. O valor da saída é 1 se e somente se a entrada é 0. Se não, a saída é 0. Ou seja o valor da saída é exatamente o oposto do valor da entrada.

Em esquemas de circuito, nós desenhamos a porta NÃO como este:



Figura 8 - Ilustração da porta NÃO

A tabela de verdade para um *inversor* olha como esta:

x	y
0	1
1	0

2. Portas compostas¹⁴

Às vezes, é prático combinar funções das portas básicas em portas mais complexas a fim reduzir o espaço em diagramas de circuito. Nesta seção, nós mostraremos algumas portas compostas junto com suas tabelas de verdade.

¹⁴ Portas Compostas: estrutura de lógica complexa implementado através da associação de várias portas básicas.

2.1. Porta NÃO-E (NAND)

A porta NÃOE é uma porta E com um inversor na saída. Assim em vez de desenhar diversas portas com esta:



Figura 9 - Ilustração da porta-E

Nós desenhamos uma única porta-E com um pequeno anel na saída como este:

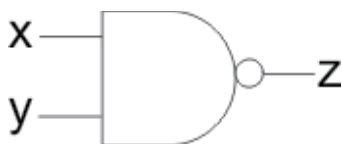


Figura 10 - Ilustração da porta NÃOE

A porta-nãoe, assim como a *porta-E* pode ter um número arbitrário de entradas.

A tabela de verdade para a porta-nãoe é semelhante à tabela para porta-e, exceto que todos os valores da saída são invertidos:

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

2.2. Porta NÃO-OU (NOR)

A porta NÃO-OU é uma porta OU com um inversor na saída. Assim em vez de desenhar diversas portas como esta:

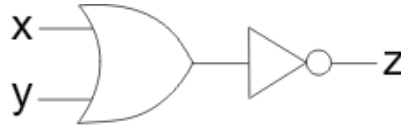


Figura 11 - Ilustração da porta OU

Nós desenhamos uma única porta OU com um pequeno anel na saída como este:

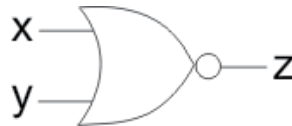


Figura 12 - Ilustração da porta NÃO-OU

A *porta não-ou*, assim como a *porta ou* pode ter um número arbitrário de entradas.

A tabela de verdade para a *porta não-ou* é semelhante à tabela para *porta ou*, exceto que todos os valores da saída são invertidos:

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

2.3. Porta OU-EXCLUSIVO (XOR)

A porta OU-EXCLUSIVO é similar a uma porta OU. Apesar de poder ter um número arbitrário de entradas normalmente representamos apenas duas entradas. O valor da saída é 1 se somente uma das entradas é 1 e a outra é 0, por isso se diz OU-EXCLUSIVO. Se não, a saída é 0. Assim, em vez de desenhar diversas portas como esta:

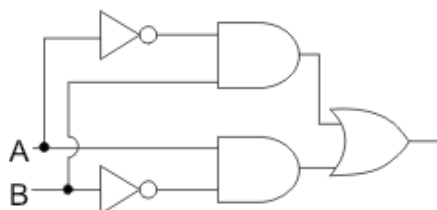


Figura 13 - Ilustração da porta OU-EXCLUSIVO

Nós desenhamos uma porta OU-EXCLUSIVO como mostrada abaixo:



Figura 14 - Ilustração da porta OU-EXCLUSIVO

A tabela de verdade para uma porta OU-EXCLUSIVO com duas entradas é mostrada abaixo:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

Mas para que serve uma porta OU-EXCLUSIVO? Ela é um bloco essencial para a construção de um circuito muito útil em computadores, o SOMADOR.

2.4. Meio Somador¹⁵

O Circuito somador realiza a soma de um bit tendo como resultado a valor da soma e o bit de vai-um conforme diagrama mostrado abaixo:

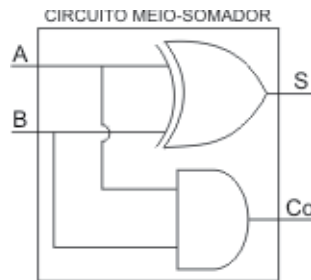


Figura 15 - Ilustração do circuito meio somador

A tabela de verdade de um meio-somador é mostrado abaixo:

x	y	z	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

¹⁵ Meio-somador: Circuito que realiza a soma de um bit utilizando portas lógicas básicas.

¹⁶ Somador completo: Circuito somador que realiza a soma considerando vai-um. O somador completo pode ser cascadeado para formar um somador de número contendo vários bits.

Somador Completo¹⁶

Apesar do meio somador informar o bit de vai um, ele não considera o vai um do bit anterior, assim ele só pode ser usado para somar o bit menos significativo.

Para realizar uma soma completa precisamos utilizar o circuito somador completo que tanto informa o vai-um para o próximo bit, como considera o vai-um do bit anterior da soma. O diagrama do somador completo é mostrado abaixo:

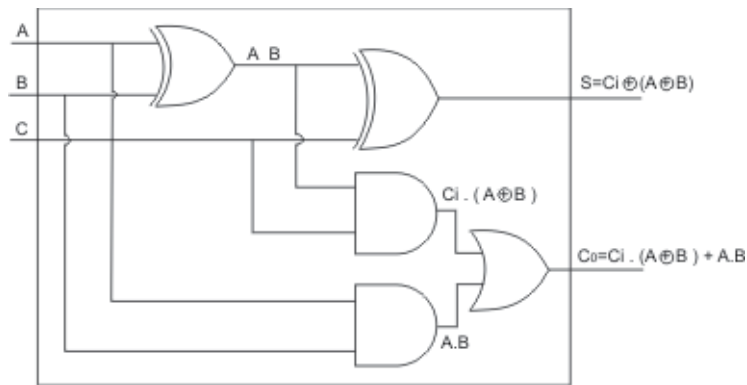


Figura 16 - Ilustração do circuito somador completo

2.6. Fazendo qualquer porta composta usando somente um tipo da porta

Por questões de processo de fabricação de circuitos integrados algumas tecnologias de circuito, são mais fáceis construir uma porta NÃO-E do que outras portas. Menor ocupação de espaço e menor consumo de energia são fatores considerados. Nesse caso, a porta NÃO-E é considerada o bloco mais básico, e todos os outros são construídos a partir dele.

É possível construir qualquer tipo da porta usando somente portas NÃO-E. Primeiramente podemos observar que um inversor é apenas uma porta NÃO-E com somente uma entrada. Em seguida, uma PORTA-E pode ser construída como uma porta NÃO-E com um inversor em sua saída. Uma porta OU pode ser construída com uma porta NÃO-E com um inversor em cada entrada.

Da mesma forma, todas as portas podem ser construídas usando somente portas NÃO-OU. E a partir dela podemos construir as demais.

Síntese do capítulo

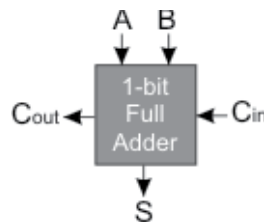


Este capítulo apresentou as formas de representação de informações em um computador. Foi mostrado a unidade básica de informação, o bit, e as representações de dados em byte, caracteres e números. Foi apresentada uma introdução ao sistema binário e a representação de números e operações matemáticas. Finalmente mostramos uma introdução à lógica digital e as principais portas lógicas existentes assim como a composição de portas mais complexas.

Atividades de avaliação



1. Construa um somador de 4 bits utilizando blocos somadores-completos mostrado na figura abaixo:



2. Observando as tabelas verdades mostradas nessa seção, desenhe o esquemático de uma porta OU-EXCLUSIVO utilizando apenas portas NÃO-E.

Leituras, filmes e sites



Como Tudo Funciona: Informática

<http://informatica.hsw.uol.com.br/>

Introdução à Computação

<http://www.cic.unb.br/~jhcf/MyBooks/ic/index.html>

Introdução à Ciência da Computação

http://pt.wikiversity.org/wiki/Introdução_à_Ciência_da_Computação

Referências



THOMAS FLOYD. **Sistemas Digitais**. 9ª Ed. Editora: Bookman, 2007.

RONALD J. TOCCI, NEAL S. WIDMER, GREGORY L. MOSS. **Sistemas Digitais: Princípios e Aplicações**. 10ª Ed. São Paulo: Editora Pearson Prentice-Hall, 2007.

ANDREW S. TANENBAUM. **Organização Estruturada de Computadores**. 5ª Ed. São Paulo: Editora Pearson Prentice-Hall, 2006.

JOHN L. HENNESSY, DAVID A. PATTERSON. **Organização e Projeto de Computadores**. 3ª Ed. São Paulo: Editora Campus, 2005.

Parte

3

Arquitectura de computadores

Arquitetura de computadores

Introdução

Neste capítulo será apresentada a arquitetura de um computador genérico com a função de cada um dos seus componentes principais. Será apresentada a relação entre *hardware*¹⁷ e *software*¹⁸ mostrando os fatores relevantes que influenciam o desempenho.

Serão apresentados a estrutura da Unidade Central de Processamento (CPU) com seus principais componentes, além das características que determinam a sua capacidade de processamento.

Também serão mostrados as características dos diversos tipos de memória e a interação de cada uma com os demais componentes do computador. Finalmente, serão mostrados os diversos dispositivos de entrada e saída existentes que permitem a utilização dos computadores pelo ser humano de forma confortável e eficaz.

Os primeiros computadores do mundo eram as pessoas. Por isso, para chamar os primeiros os computadores mecânicos ou eletrônicos foram dados nomes das pessoas que executavam o trabalho de “computar”. O “computador” era uma profissão (exercida predominante por mulheres) cujo trabalho era executar os cálculos repetitivos exigidos para computar tabelas de navegação, cartas de marés e posições planetárias.

Imagine que você tenha um trabalho de ficar hora após hora, dia após dia, fazendo nada além de computar multiplicações. Com o tempo o cansaço iria conduzir à negligência e ao erro. Por isso, durante vários séculos os inventores têm procurado por uma maneira que mecanize (isto é, encontrar um mecanismo que pudesse executar) esta tarefa.

A máquina que você pensa ser um “computador” é chamado mais precisamente “computador eletrônico digital de uso geral.” É de uso geral porque pode ser programado para executar uma grande variedade de aplicações (que fazem a diferença de um computador de propósito específico, projetado para executar somente uma função).

¹⁷. Hardware: conjunto de componentes físicos de um computador como processador, memória, disco, etc.

¹⁸. Software: conjunto de comandos lógicos que fazem um computador funcionar como programas, sistema operacional, etc.

Digital significa que o computador guarda e manipula todos os dados internamente sob a forma dos números (todos os dados numéricos, todos os dados de texto, e mesmo os sons e as figuras são armazenadas como números). O dígito significa originalmente “dedo” e desde que os povos começaram a contar com seus dedos, o nome dígito veio ser aplicado aos números.

Os computadores modernos são totalmente eletrônicos porque manipulam dados usando circuitos de chaveamento eletrônico mas algumas máquinas de computação mais velhas, eram mecânicas, usando rodas, alavancas, etc.

Um computador é um dispositivo que executa quatro funções: ele recebe dados de entradas (converte dados do mundo exterior para o universo eletrônico); armazena os dados (de forma eletrônica); processa dados (executa operações matemáticas e lógicas); e exibe os dados de saídas (mostra os resultados para os usuários através de uma tela).

Um computador consiste no *hardware* e no *software*. O *hardware* é o equipamento físico: o próprio computador e os periféricos conectados. Os periféricos são todos os dispositivos ligados ao computador para finalidades de entrada, saída, e armazenamento dos dados (tais como um teclado, um monitor de vídeo ou um disco rígido externo).

O *software* consiste nos programas e nos dados associados (informação) armazenados no computador. Um programa é uma sequência de instruções que o computador segue com o objetivo de manipular dados. A possibilidade de incluir ou excluir programas diferentes é a fonte de versatilidade de um computador.

Sem programas, um computador é apenas *hardware* de alta tecnologia que não faz qualquer coisa. Mas com a sequência de instruções detalhadas, descrevendo cada passo do programa (escrito por seres humanos) o computador pode ser usado para muitas tarefas que variam do processamento de texto a simulação de padrões de tempo globais.

Como um usuário, você irá interagir com os programas que funcionam em seu computador através dos dispositivos de entrada conectados a ele, tal como um mouse e um teclado. Você usa esses dispositivos para fornecer a entrada (tal como o texto de um relatório que você está trabalhando) e para dar comandos ao programa (tal como a definição de que uma frase do texto vai aparecer com formato negrito). O programa fornecerá a saída (os dados resultantes das manipulações dentro do computador) através de vários dispositivos de saída (tal como um monitor ou uma impressora).

Os computadores pessoais (geralmente conhecidos como PC) são usados de uma maneira muito interativa, com o usuário inserindo continuamente dados e comandos (para escolher as várias funções do programa)

e monitorando a saída com os resultados das operações realizadas. Isto é muito diferente da maneira que os grandes computadores mais antigos eram usados (o usuário fornecia a entrada e recebia a saída mais tarde, o que é chamado processamento em lote).

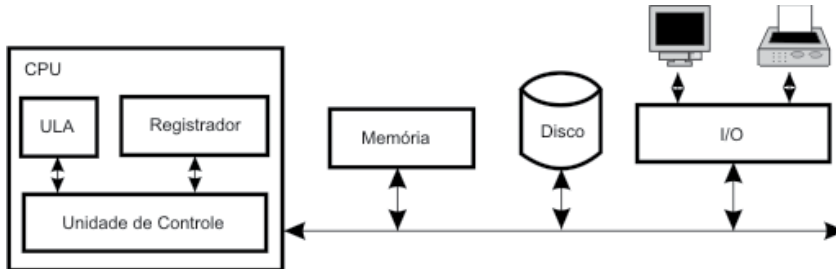


Figura 17 - Arquitetura de um computador

A Figura 17 mostra a arquitetura simplificada de um Computador. Os computadores modernos tem uma arquitetura bem mais complexa do que esta, mas os blocos principais estão aqui representados. O principal módulo é a Unidade Central de Processamento, usualmente conhecida como CPU¹⁹ (*Central Processing Unit*). A CPU é o “cérebro” do computador onde todo o processamento é realizado. Ela é constituída por três submódulos: a Unidade Lógica e Aritmética (ULA), os Registradores e a Unidade de Controle. Nos computadores modernos a CPU é construída em um único Circuito Integrado (chip).

A Unidade Lógica e Aritmética (ULA), também conhecida como *Arithmetic and Logic Unit* (ALU), é responsável por realizar as operações do computador como soma e subtração assim como as funções lógicas, OU, E, NÃO e OU Exclusivo. Algumas ULAs mais sofisticadas realizam também multiplicações e divisões.

Os registradores são *memórias*²⁰ utilizadas para realizar as operações na CPU. Essas memórias são muito rápidas e de tamanho pequeno, geralmente suficiente para guardar menos de uma dezena de valores. Além dos registradores é comum haver uma memória de rascunho dentro da CPU, chamada de Cache. Ela é utilizada para aumentar a velocidade de processamento reduzindo o tempo de acesso à memória externa.

A Unidade de Controle é responsável por controlar todo o funcionamento da CPU e também de todo o computador. Ela controla o processamento entre ULA e Registrador e interage com o barramento externo onde ficam os periféricos. A Unidade de Controle tem uma função chave que é a interpretação do código do programa que irá nortear os comandos do processamento.

Continuando na Figura 17 podemos observar um barramento e três módulos externos à CPU: Memória, Disco e Interface de Entrada e Saída (I/O).

¹⁹. CPU: acrônimo de Central Processing Unit ou Unidade Central de Processamento. Unidade responsável por realizar o processamento no computador.

²⁰. Memória: componente construído com material semi-condutor ou magnético responsável por armazenar, temporariamente ou permanentemente, os dados em um computador.

O barramento serve para interligar todos os componentes internos do computador. Para se obter altas velocidades esse barramento é geralmente paralelo (os dados são enviados paralelamente). Nessa figura representamos apenas um barramento, mas geralmente ele é dividido em barramento de dados (onde os dados trafegam), barramento de endereço (onde indicamos a localização de cada dado) e o barramento de controle (onde indicamos o comando, por exemplo, ler, escrever, copiar, etc). Em um computador real existem vários níveis de barramento com velocidades diferentes, quanto mais próximo da CPU mais rápido ele é, e quanto mais próximo dos periféricos mais lento ele é.

A Memória é constituída por um conjunto de memória semicondutora utilizada para armazenar os dados temporariamente. Podemos dizer que essa é a memória de trabalho do computador e geralmente fica localizada na placa mãe do computador. Essa memória é mais lenta que os Registradores mas é mais rápida que as unidades de armazenamento em disco. A memória semicondutora geralmente perde os dados quando o computador é desligado, por isso deve-se usar uma unidade de armazenamento permanente, o disco.

O disco consiste em uma unidade eletro-mecânica que armazena os dados em um disco magnético que mantém as informações mesmo quando o computador é desligado. Essa unidade tem grande capacidade de armazenamento mas tem velocidade de acesso significativamente menor em relação às memórias semicondutoras.

O módulo de Entrada e Saída, ou Input/Output (I/O), estabelece a ligação do computador com o mundo externo, usando equipamentos periféricos. Essa interface permite a ligação de teclados e mouses (para entrada de dados), monitores ou impressoras (para exibição dos dados) e placas de comunicação (para trocar dados a longa distância). Essa interface é significativamente mais lenta que os demais componentes do computador devido a natureza dos periféricos e a incapacidade humana de processar informações na velocidade dos computadores.

Os próximos capítulos irão detalhar cada um dos módulos que constituem um computador.

Atividades de avaliação



1. Observando a Figura 17 avalie os componentes de uma arquitetura de computadores que podem limitar o desempenho. Proponha soluções para resolver esse gargalo.
2. Pesquise sobre a história do computador e liste as invenções mais importantes.
3. Qual módulo de uma arquitetura de computadores deve ser mais rápido para melhorar o desempenho de um sistema de processamento numérico (muitas operações matemáticas)?
4. Qual módulo de uma arquitetura de computadores deve ser mais rápido para melhorar o desempenho de um sistema de armazenamento de dados (muitos dados guardados e consultados)?

Unidade central de processamento e memória

1. Processador

O processador é o componente do computador que faz o processamento real dos dados. Um microprocessador é uma unidade do processador central (CPU) encapsulada dentro de um microchip. É o “cérebro” do computador, mas devemos considerar ser um termo um pouco exagerado pois ele realmente é apenas um circuito eletrônico constituído por interruptores que executam instruções simples mas com muita rapidez.

O circuito integrado do microprocessador contém uma placa de silício que contém milhões de transistores e outros componentes fabricados no próprio silício. Como os transistores no *chip* são muito minúsculos, mesmo uma pequena descarga de corrente, como a eletricidade estática da mão do técnico, pode destruí-lo. Eis porque todos os circuitos integrados devem ser manipulados de forma que minimize a possibilidade de descarga estática através de acessórios apropriados (pulseira aterrada).

Por causa da grande quantidade de circuitos encapsulados em uma área tão minúscula, a energia necessária para seu funcionamento gera muito calor e exige sistemas de resfriamento para evitar o superaquecimento do chip. Na placa mãe de um computador o chip do processador central é coberta por um grande dissipador de calor em metal com aletas para permitir que o fluxo de ar dos ventiladores de refrigeração retirem o calor gerado pelo chip.

1.1. Velocidade de relógio

Os diversos chips em uma placa mãe precisam de uma sincronização única para manter o fluxo de dados coerente. Para isso um circuito de relógio (*clock*) distribui um sinal único para todos os chips que passarão a se comunicar de forma coerente. Você pode imaginar o relógio como uma “pulsção do coração” do computador. Quanto mais rápidos os impulso de relógio, mais rapidamente

o computador funciona; mas, o pulso de disparo não pode funcionar mais rapidamente do que a capacidade de velocidade dos chips. Com a melhoria da tecnologia de construção dos chips, a velocidade em que eles podem funcionar aumentou, aumentando a velocidade de processamento. Em razão disso, atualmente o processador central funciona mais rapidamente do que o barramento da placa mãe, que geralmente usa uma fração da taxa do processador central.

A velocidade de relógio é medida em unidades de ciclos por segundo, que é chamado de **Hertz**²¹ (Hz). As placas e os processadores centrais do computador funcionam em taxas de milhões e bilhões de Hertz, **megahertz**²² (MHz) e **gigahertz**²³ (GHz). Uma boa velocidade para um microprocessador do PC em 2008 era 4 gigahertz. Obviamente os componentes em velocidades mais altas são mais críticos e tem um custo geralmente maior.

²¹ Hertz (Hz): unidade de medida de frequência, significando quantidade de ciclos por segundo.

²² Megahertz (MHz): unidade de medida de frequência, significando quantidade de milhões de ciclos por segundo.

²³ Gigahertz (GHz): unidade de medida de frequência, significando quantidade de bilhões de ciclos por segundo.

Você pode reconhecer os termos megahertz e gigahertz das emissões de rádio. O rádio de FM e a transmissão de TV ficam na escala do megahertz, e os telefones móveis transmitem na escala do gigahertz. Assim, os circuitos do computador produzem interferências na frequência que poderia causar problemas nos dispositivos próximos. Para impedir isso, um computador geralmente contém uma caixa metálica onde ficam a placa mãe e os demais componentes.

1.2. Velocidade do processador central

A velocidade de relógio é somente um aspecto que contribui à velocidade total de processamento de um microprocessador. A arquitetura do microprocessador também é relevante. Um fator importante é o tamanho da palavra do microprocessador, que é quantos bits de entrada/saída que ele pode processar simultaneamente em um determinado instante. Os microprocessadores mais antigos usavam o tamanho de palavra de 8 bits; os microprocessadores mais novos usam o tamanho de palavra de 64-bits.

O projeto do microprocessador pode incluir a memória cache de alta velocidade. A memória cache é uma memória instalada entre o processador e a memória RAM que permite ao processador guardar instruções ou dados mais recentemente usados para o caso de precisá-los outra vez, de modo que não precise ir buscá-los na memória RAM principal, muito mais lenta que a memória cache.

Os computadores gastam muito tempo nos laços (loops), repetindo a mesma sequência de instruções várias vezes, assim se esse código estiver na memória cache pode melhorar significativamente o desempenho. Dependendo se a memória cache está situada no chip do processador central, ou em chips externos próximos com uma ligação de alta velocidade, é chamado cache L1 ou cache L2.

Outros aspectos da arquitetura dos microprocessadores que afetam a velocidade incluem a habilidade de alguns processadores centrais de trabalhar ao mesmo tempo com instruções múltiplas em paralelo. Também, alguns processadores centrais são CISC²⁴ (conjunto de instrução complexa), enquanto outros são RISC (conjunto de instrução reduzida). Os microprocessadores RISC²⁵ têm um conjunto de instruções mais simples e precisam de mais instruções para executar uma mesma ação que um microprocessador CISC faz com uma instrução, mas os processadores RISC são mais rápidos e podem terminar a operação antes.

Até hoje se discute a eficiência dos processadores RISC e CISC, mas na verdade o desempenho é dependente do tipo de instrução executada e do tipo de aplicação usada. O resultado é que você não pode simplesmente comparar processadores diferentes olhando apenas sua velocidade de relógio. Um processador PowerPC com o dobro da velocidade de relógio de um processador *Pentium* tem a velocidade de processamento aproximadamente equivalente.

1.3. Tipos de microprocessadores

O microprocessador mais usado nos PC é feito pela Intel. Desde que a IBM escolheu o processador Intel 8088 para o PC original da IBM, a maioria de clone do PC usam um processador da série Intel: 8088, 80286, 80386, 80486, *Pentium*, *Pentium II*, *Pentium III* e *Pentium 4*. A Figura 18 mostra um processador Intel.



Figura 18 - Processador Intel (Intel Corp)

Outro fabricante dos microprocessadores para o PC é AMD (*Advanced Micro Devices, Inc.*). Sua linha de processadores Athlon foi bem sucedida e tomou uma fração do mercado do processador da Intel. O processador AMD é compatível com o Intel, isto é, possuem o mesmo conjunto de instruções e qualquer *software* funciona em ambos os processadores sem alteração.

A série de computadores Macintosh da Apple usou originalmente o microprocessador Motorola 68000. Os processadores da Motorola usam um conjunto de instrução diferente do que os processadores da Intel, por isso o *software* do PC não funciona em um Mac e vice-versa. A Apple usou mais tarde o processador RISC PowerPC, desenvolvido pela Apple, Motorola e IBM. Os Mac a partir de 2004 usaram processador PowerPC G4 ou o PowerPC G5. Em 2006 a Apple passou a usar processadores Intel em seus Mac, tornando possível o funcionamento de *software* Windows diretamente no Mac.

^{24.} CISC: processador com conjunto de instrução complexa.

^{25.} RISC: processador com conjunto de instrução reduzida.

²⁶ FSB (Front Side Bus): Barramento principal de um computador PC.

2. Barramento de dados

O barramento de dados é a estrada elétrica com várias vias que ligam o processador central aos outros componentes da placa mãe, tal como memória RAM, disco rígido e controladores de I/O. Esse barramento é chamado *Front Side Bus (FSB)*²⁶. O tamanho da palavra do barramento de dados determina quantos bits podem ser movidos simultaneamente ao longo dele, tipicamente nos computadores modernos usa-se 32 bits ou 64 bits.

A velocidade de relógio dos outros componentes ligados ao barramento de dados na placa mãe é mais lenta do que a velocidade de relógio do processador central. Por isso é comum na especificação de um computador diferenciar a velocidade do processador, tipicamente na ordem de gigahertz, e a velocidade do barramento, tipicamente centenas de megahertz.

3. Memória

A memória de um microcomputador é o lugar onde os programas e os dados são armazenados quando estão sendo usados. Existem vários tipos de memórias em um microcomputador que serão comentadas a seguir.

3.1. Memória RAM

O RAM é o espaço de memória principal do computador. O termo RAM significa memória de acesso aleatório (*Random Access Memory*), e o termo vem dos primeiros computadores quando as unidades centrais tinham dois tipos de memória: Acesso aleatório, em que todo o bit da memória poderia ser endereçado a qualquer instante; e memória seqüencial (tal como os dados armazenados em fita) onde os bits poderiam somente ser alcançados de forma sequencial. Toda memória em um computador moderno é de acesso aleatório, assim não se preocupe com a memória seqüencial.

O RAM é a área de trabalho do computador. Se o computador tem mais RAM, pode abrir mais programas e mais arquivos simultaneamente. É como você tivesse uma grande mesa de trabalho onde pudesse espalhar muito mais papéis de trabalho sobre ela em vez de uma pequena mesa. Um arquivo que você está editando (e o programa que o seu computador está usando para editar) é armazenado na RAM.

O RAM consiste em bancos de transistores do circuito integrado onde podem estar ligados ou desligados (representando um ou zero). Os chips de RAM precisam energia constante para manter o que é armazenado nele; um desligamento da energia, mesmo que seja uma fração de segundo (por exem-

plo, causado por um relâmpago na rede elétrica) pode fazer com que o RAM perca seus dados. Por esse motivo, a RAM é chamada de volátil (ou aquilo que se evapora facilmente) e por isso é importante salvar frequentemente seu trabalho em um armazenamento mais permanente, tal como um disco rígido.

O tipo específico de RAM usado em um computador poderia ser DRAM (RAM Dinâmica), ou SDRAM (RAM Dinâmica Síncrona) ou RDRAM (RAM Dinâmico Rambus), mas não se preocupe sobre os detalhes. Apenas tenha cuidado para escolher o modelo apropriado para seu sistema quando você compra mais memória RAM.

O RAM é instalado geralmente na placa mãe em soquetes como DIMMs (*Dual Inline Memory Module*), pequena placa de circuito impresso com chips de memória RAM (Figura 19). A arquitetura da placa mãe limitará a quantidade de memória RAM extra você pode adicionar. O tamanho da memória RAM é medido em bytes, quilo-bytes, megabytes, etc.



Figura 19 - Módulo DIMM de Memória RAM (Spansion)

3.2. Memória ROM

Os computadores também têm alguma memória ROM (*Read Only Memory* ou Memória apenas para leitura) na placa mãe. A ROM, apesar de ser construída por transistores, não precisa de energia para guardar suas informações mas também não podem ser re-escritas.

Quando o computador é ligado, o programa armazenado na ROM é executado, dando os primeiros comandos para o computador funcionar. Esse programa, chamado de BIOS (*Basic Input/Output System* ou Sistema Básico de Entrada/Saída) executa todas as funções iniciais de um computador, por exemplo, verifica se o *hardware* está em ordem e procura os dispositivos de armazenamento onde pode encontrar uma cópia do Sistema Operacional que será carregado na memória RAM (que encontra-se apagada pois o computador estava desligado) e a seguir entrega o controle do computador ao Sistema Operacional.

Devemos lembrar que o código de BIOS nos computadores modernos é gravado em dispositivos regraváveis como EEPROM (ROM Programável e Apagável Eletricamente). A placa mãe inclui circuitos especiais que permitem que os arquivos de BIOS “permanentes” sejam atualizados quando necessário, mas que é feito raramente (EEPROM pode somente ser re-escrita um número limitado de vezes, da ordem de centenas a milhares de vezes).

3.3. Memória CMOS

A memória CMOS de um computador é uma pequena quantidade de memória “semi-permanente” onde os dados podem ser armazenados e re-escritos quantas vezes quiser mas que permanecer disponíveis quando o computador for desligado. Uma bateria pequena na placa mãe mantém a memória CMOS ‘viva’ quando não há energia.

A memória do CMOS pode armazenar parâmetros de configuração de um computador como características de disco e interfaces, etc. A BIOS irá ler essas informações e usá-las durante o carregamento do sistema operacional. A memória do CMOS pode também guardar a data e a hora permitindo que o computador mantenha a data e hora correta mesmo estando desligado. O computador tem um relógio alimentado pela pequena bateria.

A memória CMOS é um tipo de memória de transistor que exige a potência muito pequena para armazenar dados, com uma pequena bateria os dados são mantidos por um longo tempo. Entretanto, quase todas os chips de memória em um computador hoje em dia (como a RAM) usam a tecnologia CMOS, e ironicamente, “a memória semi-permanente CMOS” nos PCs modernos não é armazenada em memórias CMOS, mas em uma memória Flash, que é um tipo de memória EEPROM com tamanho maior e com capacidade de re-escrita.

3.4. Memória Virtual

A memória virtual não é necessariamente um novo componente de um computador, mas apenas uma técnica que os sistemas operacionais usam para ultrapassar os limites de memória no computador.

A quantidade de RAM limita o tamanho e quantidade de arquivos de dados e programas podem ser aberto simultaneamente. Você não poderia abrir simultaneamente dois programas que exigem 80 MB de RAM cada um em um computador que tenha somente 128 MB do RAM.

Entretanto, os sistemas operacionais modernos usam a memória virtual para permitir a ampliação dessa capacidade. Se tudo não couber na memória RAM disponível, o sistema operacional pode automaticamente escrever os dados não utilizados nesse momento no disco rígido, e coloca na memória os dados que está utilizando. Mas isso tem um preço! Os disco rígidos (dispositivos mecânicos) são muito mais lentos do que a memória RAM (dispositivo eletrônico), assim o computador funcionará muito mais lento se usar a memória virtual.

Dispositivos de armazenamento

Os dispositivos de armazenamento, tais como unidades de disco, armazenam programas e arquivos de dados quando não estão sendo usados. Ao contrário das memórias RAM, os dados armazenados nestes dispositivos não desaparecem quando a energia é desligada.

Os tipos principais de dispositivos de armazenamento são: discos magnéticos, disco de estado-sólido e disco ótico.

1. Disco rígido

Um disco rígido é um dispositivo mecânico constituído por discos de metal e revestidos com óxido de metal que permite ser magnetizado. Uma cabeça pequena de leitura/gravação eletromagnética na extremidade de um braço móvel magnetiza pontos minúsculos no disco para armazenar dados. Todos esse componente são lacrados em uma caixa hermeticamente fechada. A distância entre o disco e a cabeça de leitura/gravação é comparável ao de um grão de poeira, caso exista alguma sujeira na superfície do disco poderá danificar definitivamente a unidade de disco.

Os pontos magnéticos magnetizados em um sentido representam o valor binário “um” e os pontos magnetizados no sentido oposto representam um “zero”. A mesma cabeça eletromagnética pode mais tarde detectar o sentido do campo magnético dos pontos quando o disco passa debaixo da cabeça, identificando se o dado é 0 ou 1 conforme o sentido de magnetização. A Figura 20 mostra um disco rígido com seus principais componentes.



Figura 20 - Disco Rígido IBM

Os discos rígidos são avaliadas por sua capacidade de armazenamento, tipicamente dezenas ou centenas de gigabyte. Outra métrica de avaliação é quanto rapidamente os discos giram (em RPM, Rotações por Minuto), que é

tipicamente milhares de RPM. Uma outra métrica de avaliação de um disco rígido é o tempo de acesso médio (medido em milissegundos), que diz em média quanto tempo levaria a movimentação do braço para recuperar um dado no disco. Os tempos de busca típicos ao redor 6 ms.

A eletrônica que controla o disco rígido incorpora geralmente alguma memória cache. O movimento do braço lê diversos setores dos dados de uma só vez e se o processador central pedir os setores seguintes, a memória cache pode responder imediatamente sem ter que esperar o disco girar e posicionar o braço no local correto.

Existem vários padrões de controlador para interface com disco rígido como: IDE, ATA, SATA, SCSI, ou outros. Não vamos entrar em detalhes sobre esses padrões aqui, apenas você deve ter em mente que deve se usar o mesmo padrão interface para se ligar o disco na placa mãe.

2. Disquete flexível

Um disquete flexível é um disco plástico flexível de *mylar* (tipo de plástico) coberto com óxido de metal que é magnetizado através de uma cabeça magnética, de forma semelhante ao disco rígido. As disquetes flexíveis tem 3.5 “ de diâmetro e dispõe de uma capacidade de 1.44 MB para armazenamento.

O disquete dispõe de um obturador que protege a superfície de disco da sujeira e das impressões digitais, o obturador desliza expondo a mídia magnética quando o disco é introduzido na unidade de leitura de modo que as cabeças de leitura/gravação possam alcançar o disco magnetizável.

Uma trava plástica pode ser deslizado para desbloquear um furo no canto do disquete para proteger o disco de escrita (assim dados não podem ser acidentalmente apagados). O tempo de acesso é muito mais lento do que um disco rígido, e a vida útil é significativamente mais curta.

As unidades de disquete flexível estão caindo em desuso. A maioria dos computadores novos não dispõe dessa interface. O motivo desse abandono é a redução de custo de outras mídias com maior capacidade, como CD Regravável, DVD Regravável e Memória Flash (*pen drive*), com capacidades de centenas de megabytes a vários gigabytes.

3. Cartões da memória Flash

Memória Flash é um tipo especial de memória RAM desenvolvida nos anos 80 que apresenta a característica de não ser volátil, isto é, não perde os dados quando a energia é desligada. Apesar de ser um pouco mais lenta que as memórias RAM voláteis ela é muito mais rápida que os discos rígidos.

A memória *Flash* se popularizou com a disponibilidade de interface USB, que permitiu a construção de memórias não voláteis portáteis que podem se ligar em qualquer porta USB com uma velocidade bastante alta. Esse dispositivo chamado *pen drive* ou *memory key* tem se tornado cada vez mais popular em virtude da praticidade e baixo preço. A Figura 21 mostra um *pen drive* USB.

O mesmo tipo da memória Flash pode ser usado em pequenos cartões de memória como cartão SD, *Compact Flash*, *Memory Stick*. Esses cartões são usados em telefones celulares, PDAs, câmeras digitais, tocadores MP3, e outros dispositivos digitais. Você pode ler um cartão desses em um leitor de cartão da memória instalado no computador. Estes cartões de memória tem capacidades de armazenamento de centenas de megabytes a dezenas de gigabyte.



Figura 21 - Pendrive Sandisk

4. CD-ROM

Um CD-ROM é um dispositivo de memória ótica com capacidade de até 700 MB. O ótico significa que a luz é usada para ler os dados do disco (não é uma mídia magnética). Os CD-ROM são muito baratos se produzidos em grandes quantidades e a maioria de *software* são distribuídos em CD-ROM.

Os dados são armazenados em um CD-ROM através de pequenos poços gravados na camada plástica interna que é aluminizada e recoberta com uma outra camada plástica transparente para proteção. Um raio laser projetado sobre o disco girando reflete uma seqüências dos poços e não-poços que converte em dados zero e um.

O desempenho de CD-ROM são avaliadas pela velocidade, tal como 32x, que significa 32 vezes mais rápido do que a primeira geração de CD-ROM.

4.1. CD-R e CD-RW

Os CD-R (CD Gravável) e CD-RW (CD Regravável) são os CD que podem ser escritos e re-escritos, respectivamente.

Os discos dos CD-R têm uma camada de tinta que é alterada por um laser de potência mais elevada que escreve os zeros e uns no disco. Na leitura se usa um laser mais fraco que reflete os poços mas não altera os dados. Os CD-R podem ter sua superfície escrita apenas uma vez, após o qual pode-se apenas realizar leitura. Assim como os CD, os CD-R pode armazenar até 700 MB dos dados.

Os discos CD-RW contêm um material sensível especial e usa dois raios laser diferentes, um que escreve (com potência alta) e lê (com potência baixa) e um outro laser que “apaga” o conteúdo, deixando o disco pronto para ser regravado. Assim esses discos podem ser usados muitas vezes (mas devem ser apagados antes da re-escrita).

4.2. DVD

Os discos de DVD-ROM são mídias óticas similares aos CD-ROM, mas com uma capacidade de armazenamento maior. Os DVD permitem gravação de 4.7 GB de dados em mídia de camada única ou 8.54 GB em camada dupla.

Como um CD, os DVD tem igualmente tem variações de gravação como DVD-R para o gravável e DVD-RW para o regravável. As velocidades de leitura de DVD são avaliadas nos termos de quantas vezes ele é mais rápido que os drives originais de DVD, por exemplo, um disco 6x equivale a dizer que ele é 6 vezes mais rápido que os primeiros DVD.

Atividades de avaliação



1. Relacione os principais componentes de uma CPU e descreva a função de cada um.
2. Por que é importante para um computador ter um relógio único?
3. Diga a diferença entre memória RAM e memória ROM?
4. Qual a forma de codificação de dados que podemos usar em discos rígidos e flexíveis e unidade de CD ou DVD?

Sistemas de entrada e saída

Em computação, o sistema de entrada/saída, em inglês *Input/Output*, também conhecido pela sigla *I/O*, referem a comunicação entre um sistema de processamento de dados (como um computador) e o mundo exterior, seja ele um ser humano ou um outro sistema de processamento de dados. As entradas são os sinais ou os dados recebidos pelo sistema, e as saídas são os sinais ou os dados emitidos por ele.

Os dispositivos de *I/O* são usados por uma pessoa (ou por outro sistema) para comunicar-se com um computador. Por exemplo, teclados e mouses são dispositivos de entrada de um computador, enquanto o monitores e impressoras são considerados dispositivos de saída de um computador. Os dispositivos para comunicação entre computadores, tais como modem e placas de rede, são tipicamente para entrada e saída.

Em um computador a combinação do processador central e da memória central (isto é, a memória que o processador pode ler e escrever a diretamente) é considerada o cérebro de um computador. Qualquer transferência de informação entre esse bloco e qualquer outra unidade é considerada também *I/O*, como por exemplo, uma unidade de disco magnético ou ótico.

Abaixo vamos discutir a variedade de dispositivos usados para a entrada e a saída do computador.

1. Dispositivos de Entrada

1.1. Teclado

O teclado de computador é utilizado para incluir informações de texto no computador, como se faz na datilografia. O teclado pode também ser usado para inserir os comandos que controlam o computador para executar determinadas ações. Os comandos são escolhidos tipicamente de um menu na tela usando um mouse, mas geralmente há alguns atalhos de teclado para esses comandos.

Além das teclas do teclado principal (usado para datilografar o texto), os teclados geralmente têm um teclado numérico (para inserir dados numéricos eficientemente), um conjunto de teclas de edição (usadas para operações de edição do texto), e uma fila de chaves de função na parte superior (para chamar determinadas funções dos programas). Os computadores portáteis, que não têm espaço para teclados grandes, incluem frequentemente uma tecla “fn” de forma que outras teclas possam realizar duas funções (por exemplo, ter um teclado numérico encaixado dentro do teclado principal).

A maioria dos teclados padrão PC se conectam ao computador através de um conector PS/2 ou da porta do USB.

2. Dispositivos Apontadores

As interfaces gráficas (chamadas GUI) exigem algum tipo do dispositivo para posicionar o cursor na tela. Os dispositivos apontadores típicos são: mouse, almofada de toque e trackball. Os dispositivos apontadores, tais como mouse, se conectaram ao PC através de portas seriais (mais antigos), porta PS/2 ou a porta USB (os mais novos).

2.1. Mouse

O dispositivo apontador mouse apoia-se em sua superfície de trabalho, como uma mesa, e é movido com sua mão. Nos mouses mais velhos uma esfera na parte inferior rola na superfície enquanto você move o mouse, e os rolos internos detetam o movimento da esfera e transmitem a informação ao computador através do cabo. O nome *mouse* (rato em inglês) vem da semelhança do dispositivo em forma de concha com um cabo. A Figura 22 mostra um mouse tradicional.



Figura 22 - Mouse com fio (Clone)

O mouse óptico (mais recente) não usa uma esfera de rolamento, mas uma luz e um sensor óptico para detetar por reflexão o movimento do mouse seguindo a rugosidade da superfície da mesa. Os mouses óticos evitam o problema de uma esfera suja no mouse, que faz com que os mouses antigos percam a precisão se a esfera e os rolos internos não são limpos frequentemente.

Um mouse pode ser sem fio, comunicando-se com o computador através das ondas de rádio (geralmente usando o protocolo Bluetooth), tornando desnecessário o cabo para ligar ao computador (no entanto, ele vai precisar de baterias internas).

Um mouse inclui uma ou várias teclas (às vezes uma roda) para permitir que os usuários interajam com a interface gráfica. O mouse tradicional de PC tem duas teclas para acionar comandos, e a roda serve para deslizar a imagem na tela do computador.

2.2. Almofada de toque (touch pad)

A maioria de computadores portáteis têm hoje um dispositivo apontador tipo almofada de toque (*touch pad*). Você move o cursor na tela deslizando o dedo pela superfície da almofada plástica. As teclas ficam situadas abaixo da almofada, mas a maioria das almofadas de toque permitem que você execute o “clique do mouse” batendo com o dedo na própria almofada.

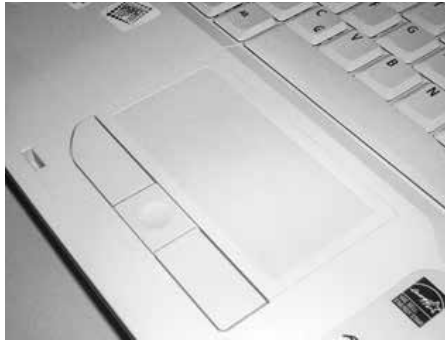


Figura 23 - Touchpad de notebook (Acer)

As almofadas de toque têm a vantagem sobre os mouses porque não tem partes mecânicas que se danifiquem e não pegam sujeira. A Figura 23 mostra uma almofada de toque.

2.3. Trackball

O *trackball* é como um mouse de cabeça para baixo, com a esfera situada na parte superior. Você usa seus dedos para rolar o trackball, e os rolos internos (similares ao de um mouse) detetam o movimento que é transmitido ao computador. Os *trackballs* têm a vantagem sobre mouses porque o corpo do trackball permanece estacionário em sua mesa, assim que você não precisa tanto espaço para usar o trackball.

3. Tablet gráfico

Um *tablet* gráfico consiste em uma área eletrônica de escrita e uma pena especial que “escreve” sobre ela. Os *tablets* gráficos permitem que artistas criem imagens gráficas com movimentos e as ações similares aos de ferramentas de desenho tradicionais. A pena do *tablet* gráfico é sensível a pressão, assim que pressionando obtemos um traço mais grosso ou mais fino.

4. Scanner

Um *scanner* é um dispositivo que lê imagens em uma página impressa em papel produzindo uma imagem feita de pixels minúsculos dos valores diferentes de brilho e da cor que são representados numericamente e transmitidos ao computador digitalmente. Os scanners fazem a leitura de figuras, mas podem também fazer a leitura de páginas do texto que, através de *software* OCR (reconhecimento de caráter ótico) que vai identificar as formas individuais das letras e criar uma linha de texto da página em formato digital.

5. Microfone

Um microfone pode ser ligados a um computador através de uma entrada de som. O som digitalizado é convertido em números digitais que representam o som analógico original podendo ser armazenado no computador e reproduzido mais tarde.

6. Dispositivos de saída

6.1. Monitor CRT

O dispositivo de saída tradicional de um computador para um ser humano é um monitor CRT (tubo de raio de catódico). Assim como um aparelho de televisão, o monitor CRT contém um grande tubo de raio de catódico que projeta um feixe de elétron em direção à tela gerando uma imagem dos pontos coloridos no interior da tela. Os monitores do CRT são pesados e consomem mais energia que as telas LCD.

O tamanho de tela do monitor é medido diagonalmente em polegadas. Nem toda a área da tela pode ser útil para mostrar imagem, assim muitas vezes também é indicada a área visível. A resolução do monitor é o número máximo de pixels que pode mostrar horizontalmente e verticalmente (como 800 x 600, ou 1024 x 768, ou 1600 x 1200).

A maioria de monitores podem indicar diversas resoluções abaixo de seu valor máximo. Quanto maior a quantidade de pixels possíveis maior é a qualidade do monitor. Os pixels são os pontos pequenos que fazem a imagem na tela. O afastamento dos pontos minúsculos do fósforo da tela é chamado distância entre pontos, também conhecido como dot pitch (dp). Uma tela com distância entre pontos menor produz umas imagens mais definida.

6.2. Monitor de tela plana LCD

Um monitor de tela plana usa geralmente uma tela do LCD (tela de cristal líquido) para indicar a saída do computador. O LCD consiste em diversas camadas finas que polarizam a luz que passa por elas. A polarização de uma camada, contendo as moléculas de cristais líquidos, pode ser controlada eletronicamente em cada pixel, obstruindo quantidades variadas de luz fazendo um pixel mais claro ou mais escuro. Existem outros tipos de tecnologia do tela plana como as telas de plasma mas os LCDs são os mais comuns. A figura 24 mostra um monitor LCD.



Figura 24 - Monitor LCD Samsung

Como no CRT, o tamanho de uma tela plana é expressa pelo tamanho diagonal em polegadas, e a resolução é definida pelo número de pixels horizontais e verticais da tela.

Os LCDs mais antigos tinham tempos de resposta lentos e baixo contraste, mas as telas LCD de matriz ativa têm um transistor de película fina transparente (conhecido pela sigla TFT) controlando cada pixel, assim o tempo de resposta, o contraste, e o ângulo de visão melhoram muito.

As telas planas são menos volumosas e consomem muito menos energia que monitores CRT. Ainda são mais caros do que os CRT mas a diferença de preço está cada vez menor, tornando-se cada vez mais comuns.

6.3. Impressoras Jato de tinta

Para fazer uma cópia definitiva de um trabalho, você precisa algum tipo da impressora ligada ao seu computador. O tipo o mais comum de impressora é do tipo jato de tinta colorida. Estas impressoras dão forma à imagem na página pulverizando gotas minúsculas da tinta a partir de uma cabeça de impressão na folha de papel. A impressora precisa diversas cores de tintas básicas (ciano, amarelo, magenta, e preto) de fazer imagens coloridas.

As impressoras jato de tinta são baratas, mas o custo dos materiais de consumo (cartuchos de tinta e papel especial) podem tornar seu custo relativamente alto.

6.4. Impressora laser

Uma impressora laser produz imagens de boa qualidade usando a mesma tecnologia que as fotocopiadoras usam. Um cilindro revestido com o material fotossensível é carregado, a seguir uma imagem é desenhada nela por um laser (ou por diodo emissor de luz) que faz aquelas áreas perderem a carga. O cilindro rola então através do tonalizador (partículas minúsculas de pigmento) que é atraído apenas pelas áreas carregadas do cilindro. O tonalizador é depositado no papel e fundido através de calor.

A maioria de impressoras laser são monocromáticas (somente uma cor, geralmente preto), mas algumas impressoras laser possuem cartuchos de tonalizador de várias cores podendo produzir impressões coloridas.

As impressoras laser são mais rápidas do que as impressoras jato de tinta. Sua velocidade é avaliada em páginas por o minuto (ppm). As impressoras de laser são mais caras do que as jato de tinta, mas tem um custo operacional mais baixo.

6.5. Impressoras matriciais

As impressoras matriciais usam pequenos pinos eletromagnéticos ativados na cabeça de impressão, em uma fita entintada para produzir imagens pelo impacto. Essas impressoras são lentas e ruidosas e atualmente não são muito usadas em computadores pessoais, no entanto tem o menor custo operacional e podem imprimir os formulários carbonados (várias cópias), coisa que nem as impressoras jato de tinta nem laser podem fazer.

6.6. Saída de som

Os computadores igualmente produzem uma saída de som, variando dos sinais acústicos simples que alertam o usuário, aos efeitos sonoros de jogos, à música com qualidade de concerto. Os circuitos para produzir o som geralmente estão localizados na placa mãe mas podem ser instalados em cartão de expansão. Para ouvir o som é necessário um conjunto de alto-falantes externos ou de fones de ouvido.

7. Dispositivos de Entrada e Saída

7.1. Tela sensível ao toque

Alguns computadores, especialmente os PDA e telefones celulares, têm telas sensíveis ao toque. O usuário pode fazer escolhas e pressionar imagens da tecla na tela. Também é comum usar um estilete permitindo que se “escreva” na superfície de uma tela sensível ao toque. Uma tela sensível ao toque tem a função de uma tela plana e de um teclado, por isso consideramos um dispositivo de entrada/saída. A Figura 25 mostra uma tela sensível ao toque.



Figura 25 - Tela sensível ao toque (Aramedia)

7.2. Modem

Modem vem do acrônimo Modulador/Demodulador e consiste em uma interface que serializa os dados e transmite através de uma linha telefônica. Um modem é usado tanto para transmitir como para receber, e caso a linha telefônica seja de boa qualidade pode alcançar distâncias bastante grandes. A velocidade máxima é 56 Kbps, e cada vez mais tem sido substituído por dispositivos de alta velocidade.

7.3. Interface Ethernet

A interface Ethernet é utilizada para ligar o computador a outros computadores, sejam eles locais (rede local) ou longa distância (Internet). O protocolo Ethernet cada vez mais tem sido usado como interface padrão dos computadores PCs. Uma interface Ethernet funciona em 10, 100 MBPS, 1 e 10 GBPS de modo auto adaptável, isto é, uma interface tenta se conectar na velocidade mais alta e caso não seja possível vai baixando a velocidade até sincronizar com o outro equipamento e conseguir se comunicar. O fato principal de popularidade das interfaces Ethernet é a alta velocidade e o baixo custo.

Síntese do capítulo



Este capítulo apresentou a arquitetura de um computador genérico e a função dos diversos componentes que o compõe. Foi mostrado a arquitetura da Unidade Central de Processamento (CPU), seus respectivos componentes e suas funções. Foram mostrados os diversos tipos de memórias detalhando suas funções conforme suas características, além de mostrar suas aplicações na arquitetura proposta. Finalmente, foram mostrados os diversos dispositivos de entrada e saída que facilitam a utilização do computador pelos seres humanos.

Atividades de avaliação



1. Qual o dispositivo de entrada e saída de maior velocidade? Por que ele pode ter essa velocidade?
2. Para melhorar a velocidade na comunicação entre computador e dispositivos de entrada/saída usa-se um mecanismo chamado DMA (*Direct Memory Access*). Pesquise e explique suas características e funcionamento.
3. Comente as vantagens e desvantagens de se transmitir dados em paralelo ou serial entre dispositivos de entrada e saída e computador.

Leituras, filmes e sites



Introdução à Arquitetura de Computadores

http://pt.wikibooks.org/wiki/Introdução_à_Arquitetura_de_Computadores

Introdução à Ciência da Computação

http://pt.wikiversity.org/wiki/Introdução_à_Ciência_da_Computação

Museu do Computador

<http://www.museudocomputador.com.br/>

O Computador

<http://pt.wikipedia.org/wiki/Computador>

Referências



FERNANDO DE CASTRO VELLOSO. **Informática: Conceitos Básicos**. 7ª Ed. Editora: Campus, 2004.

JOHN L. HENNESSY, DAVID A. PATTERSON. **Arquitetura de Computadores: Uma Abordagem Quantitativa**. 4ª Ed. Editora: Campus, 2008.

WILLIAN STALLINGS. **Arquitetura e Organização de Computadores**. 5ª Ed. Editora: Pearson, 2002.

Parte

4

Conjunto de instruções e endereçamento

Introdução

O termo *software* faz parte cada vez mais do vocabulário das pessoas com os mais diversos interesses, seja no contexto do profissional (*software* de escritório), ou de entretenimento (*software* de jogos). De forma geral, *software* é o conjunto de produtos que inclui principalmente programas de computador, geralmente acompanhados de manuais, especificações e outros documentos.

Por outro lado, o conjunto de instruções que compõem o programa a ser seguido e executado pelo computador, é também chamado de *software*, assim como o comportamento observado a partir da sua execução.

De forma geral, as instruções que compõem os programas são escritas utilizando uma linguagem de programação de alto nível de abstração. Uma linguagem classificada como sendo de alto nível indica que possui características que a torna mais próxima da linguagem natural humana. Em contrapartida, uma linguagem dita de baixo nível, é mais próxima da linguagem compreendida pelo *hardware*. Como exemplo de linguagens de programação de alto nível podemos citar: Pascal, Fortran, C.

As instruções de alto nível precisam ser traduzidas ou interpretadas, para posteriormente serem executadas por um *processador* ou *máquina virtual*. Para que o processador possa interpretar e executar um programa, seu conjunto de instruções precisa ser escrito em *linguagem de máquina*²⁷ ou *assembly language*. O resultado deste processo é o programa em linguagem de montagem ou linguagem de máquina. Cada arquitetura possui uma linguagem de montagem particular.

Finalmente, o programa em linguagem de montagem é convertido para um programa em código objeto pelo montador (*assembler*). O montador traduz diretamente uma instrução da forma textual para a forma de código binário. Neste formato final a instrução é carregada na memória e interpretada pelo processador.

²⁷ Linguagem de Máquina: representação numérica utilizada diretamente pelo processador

Saiba Mais



A linguagem *Assembly* é uma linguagem de baixo nível para programação de computadores. Esta linguagem implementa uma representação simbólica de códigos de máquina numéricos e considera outras restrições para uma arquitetura de CPU particular. Esta representação usualmente é definida pelo fabricante de *hardware*, e é baseada em abreviaturas das palavras em inglês (mnemônicos), de forma a facilitar o trabalho dos programadores. A linguagem de máquina é específica a uma determinada arquitetura de computador física e virtual, diferentemente das linguagens de alto nível, as quais são portáteis.

Desde o ponto de vista do *software*, um computador pode ser estruturado como uma série (ou níveis) de abstrações, onde a série superior representa um incremento em relação à série que a precede. Em cada nível é utilizada uma linguagem com propriedades diferenciadas, onde cada linguagem usa a sua antecessora (nível imediatamente inferior), como base. A linguagem ou nível que fica mais embaixo é mais simples de forma a ser entendido pelo computador, enquanto que a que fica mais acima é a mais sofisticada e complexa, porém, mais próxima do entendimento humano.

O nível mais baixo é o nível lógico digital, composto por circuitos eletrônicos estudados no capítulo 2.

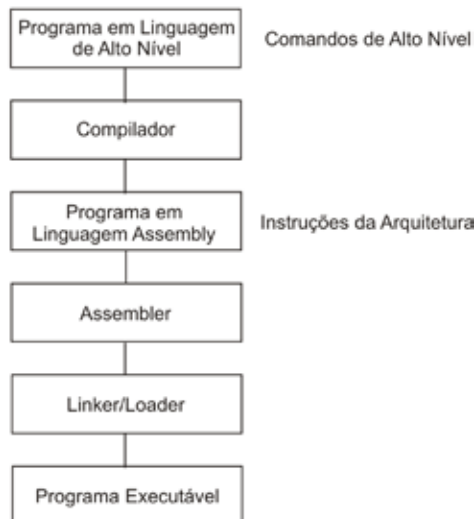


Figura 26 - O conjunto de instruções dentro do contexto de *software*.

Nesta unidade será abordado o nível de Arquitetura do Conjunto de Instruções (ISA). Este conjunto de instruções determina as instruções executadas que podem ser interpretadas por uma máquina específica. Cada máquina tem uma linguagem consistindo em todas as instruções que a máquina é capaz de executar.

Praticando

1. Que significa que uma linguagem de programação é classificada como sendo de “baixo ou alto” nível?
2. Quais linguagens de alto nível você conhece? De exemplos de instruções de uma linguagem de alto nível.
3. Um programa em uma linguagem de alto nível pode ser traduzido ou interpretado pelo processador. Pesquise qual é a diferença entre esses dois conceitos.

Instruções

De forma geral, uma única instrução em linguagem de alto nível é equivalente a várias instruções em linguagem de máquina. Por exemplo, observe a seguinte instrução de linguagem de alto nível em Pascal: $X = X + Y$, onde as variáveis X e Y correspondem às posições de memória 312 e 311. Esse comando alto nível pode ser implementado através do seguinte conjunto de instruções de máquina:

1. Carregar um registrador com o conteúdo da posição de memória 312.
2. Adicionar o conteúdo da posição de memória 311.
3. Armazenar o conteúdo do registrador na posição 312.

Em linguagem de máquina, as instruções são representadas através de mnemônicos, que vinculam o nome da instrução à sua função. Alguns exemplos comuns são:

- **ADD** Adição
- **SUB** Subtração
- **MPY** Multiplicação
- **DIV** Divisão
- **LOAD** Carregar dados na memória
- **STOR** Armazenar dados na memória

Tudo o que a máquina pode fazer é controlado pelo seu conjunto de instruções. As instruções permitem movimentar dados, realizar operações aritméticas e lógicas, e de comparação, a partir das quais é possível desviar a execução do programa segundo os resultados obtidos.

Cada instrução deve conter toda a informação necessária para que a CPU possa executá-la. Uma instrução de máquina normalmente é composta pelos seguintes elementos:

- **Código de operação:** código binário que especifica a operação a ser executada.

- **Referencia a operando fonte** que constitui o dado de entrada para a operação. Podem existir mais de um.
- **Referencia a operando de destino** indicando o destino do resultado eventualmente gerado pela operação.
- **Endereço da próxima instrução** indica onde a CPU deve procurar a próxima instrução a ser executada, que pode estar localizada na memória principal ou virtual. Este campo fica implícito no caso em que a próxima instrução segue imediatamente à atual.

Cada instrução é dividida em campos de acordo aos elementos que a compõem, e cada campo é representado por um conjunto de bits. A seguir é apresentado um formato de instrução simples.



Os passos envolvidos na execução de uma instrução são os seguintes:

1. Cálculo do endereço da instrução
2. Busca da instrução
3. Alterar o contador do programa para indicar a próxima instrução.
4. Decodificação da operação da instrução
5. Cálculo do endereço de operando
6. Busca de operando
7. Operação sobre os operandos
8. Armazenamento de resultado
9. Cálculo de endereço de operando

Os passos 4 e 5 podem se repetir no caso de múltiplos operandos. Da mesma forma os passos 7 e 8 para múltiplos resultados. A seqüência de passos recomeça na busca de uma nova instrução, uma vez que a anterior for completada.

Essa seqüência de etapas, denominada de ciclo **buscar-decodificar-executar**, é de fundamental importância para a operação de todos os computadores.

Para realizar este ciclo, a Unidade Lógica e Aritmética é auxiliada por registradores (normalmente de 1 a 32), onde os dados são armazenados. Os registradores alimentam os registradores de entrada da ALU, os quais vão ser utilizados para executar a operação. Efetuada a operação o resultado é

temporariamente armazenado no registrador de saída da ALU, para depois ser armazenado em um registrador. Este processo é chamado de **ciclo do caminho** de dados. Outros registradores importantes que contribuem neste ciclo são o Contador de Programa (PC), que indica a próxima instrução a ser executada. Uma vez recuperada, a instrução sendo atualmente executada é armazenada no Registrador de Instrução.

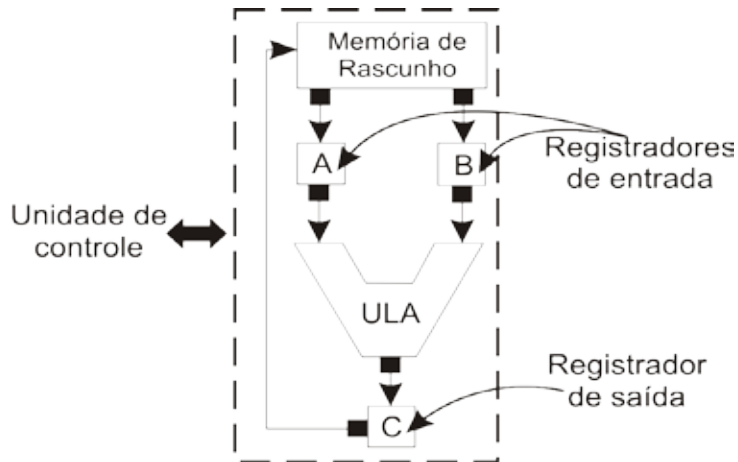


Figura 27 - Arquitetura básica de um computador.

Em algumas máquinas, todas as instruções tem o mesmo comprimento; em outras podem existir comprimentos diferentes. No primeiro caso, a decodificação fica mais fácil, mas pode acarretar desperdício de espaço.

Atividades de avaliação



1. Quais seriam as vantagens de um computador estruturado em níveis possuir mais ou menos níveis?
2. Explique os diferentes papéis dos registradores no contexto do ciclo de execução de uma instrução.
3. Considere o conjunto de passos seguidos para a execução de uma instrução. Qual é a finalidade do passo 3? O que aconteceria se este passo fosse omitido?
4. Considere as seguintes tarefas: (1) somar 100 números armazenados na memória principal e (2) somar um número a ele mesmo 100 vezes. Apesar de ambos os programas precisarem de 100 adições, o segundo poderá ser executado muito mais rapidamente do que o primeiro? Por que?

Instruction Set Architecture - ISA

Esse nível está posicionado entre o nível da microarquitetura²⁸ e o sistema operacional e faz às vezes de interface entre o *software* (compiladores) e o *hardware*, facilitando a portabilidade de programas escritos em diversas linguagens entre diferentes computadores.

A estratégia utilizada pelos projetistas para reduzir esse gap semântico é criar uma representação intermediária – o nível ISA – e construir *hardware* que possa executar programas de nível ISA diretamente. Desta forma, a ISA é projetada de forma a tornar compatíveis as necessidades de ambas as camadas.

Alem disso, desde o ponto de vista do cliente, é de suma importância que a ISA seja compatível entre os diversos modelos de computadores, de forma a suportar a portabilidade dos programas antigos para uma nova arquitetura, sem que sejam necessárias alterações. Desta forma surge o conceito de arquitetura para uma família de computadores, os quais compartilham um conjunto de propriedades e características em comum.

O conjunto de instruções do processador é uma questão central para a definição e implementação da sua arquitetura (por exemplo: operações realizadas pela unidade lógica e aritmética, número e função dos registradores, etc.).

O conjunto de instruções é caracterizado pelos tipos de instruções e operandos, número e localização dos operandos em instruções aritméticas e lógicas, modos de endereçamento para acesso aos dados na memória, e o formato dos códigos de instrução.

Consequentemente existe uma dependência direta entre o projeto do conjunto de instruções e o projeto da arquitetura do computador, e vice versa. Analogamente, os tipos de operando que podem ser diretamente manipulados por uma arquitetura dependem, é claro, dos tipos de instruções oferecidas.

As instruções oferecidas por uma arquitetura podem ser classificadas em três categorias diferentes, de acordo com o tipo de operação que realizam. Uma arquitetura de uso geral fornece pelo menos três categorias de instruções básicas a seguir.

²⁸ Nível da Microarquitetura: composto normalmente de 8 a 32 registradores que compõem uma memória local, e um circuito capaz de executar operações aritméticas simples (ALU).

- **Instruções aritméticas e lógicas:** são as instruções que realizam operações aritméticas sobre números inteiros (adição, subtração) e operações lógicas bit-a-bit (AND, OR);
- **Instruções de movimentação de dados:** são instruções que transferem dados entre os registradores ou entre os registradores e a memória principal; E/S.
- **Instruções de transferência de controle:** são instruções de desvio e de chamada de rotina, que transferem o controle da execução para uma instrução específica dentro do código do programa.
- **Instruções de armazenamento de dados:** são instruções de memória.

Arquiteturas destinadas para uma aplicação específica ou domínio específico podem oferecer outros tipos de instruções, especializadas para aquela aplicação. A Unidade 2 apresenta os principais tipos de dados normalmente representados em uma arquitetura de uso geral.

1. Numero de operandos e localização

Uma instrução, seja aritmética ou lógica, é composta por um identificador de operação e um conjunto de operandos referenciados. Por exemplo, operações aritméticas e lógicas são unárias ou binárias. Considerando que o resultado da operação precisa ser armazenado, se faz necessário um terceiro operando. De forma geral, o endereço da próxima instrução é implícito (contido no contador do programa).

Exemplo: computar o comando $Y = (A - B) \div (C + D \times E)$, utilizando instruções típicas de um, dois, e três endereços.

Instruções com três endereços		
	Instrução	Comentário
SUB	Y, A, B	$Y \leftarrow A - B$
MPY	T, D, E	$T \leftarrow D \times E$
ADD	T, T, C	$T \leftarrow T + C$
DIV	Y, Y, T	$Y \leftarrow Y \div T$

Instruções com dois endereços		
Instrução		Comentário
MOVE	Y, A	$Y \leftarrow A$
SUB	Y, B	$Y \leftarrow T - B$
MOVE	T, D	$T \leftarrow D$
MPY	T, E	$T \leftarrow T \times E$
ADD	T, C	$T \leftarrow T + C$
DIV	Y, T	$Y \leftarrow Y \div T$

Instruções com um endereço		
Instrução		Comentário
LOAD	D	$AC \leftarrow D$
MPY	E	$AC \leftarrow AC \times E$
ADD	C	$AC \leftarrow AC + C$
STOR	Y	$Y \leftarrow AC$
LOAD	A	$AC \leftarrow A$
SUB	B	$AC \leftarrow AC - B$
DIV	Y	$AC \leftarrow AC \div Y$
STOR	Y	$Y \leftarrow AC$

As instruções com três operandos não são muito comuns por conta do tamanho relativamente grande das instruções. No caso de instruções com dois operandos e operações binárias, um dos endereços referencia um operando e o resultado, ou seja, que o resultado é armazenado no lugar do próprio operando. Um cuidado especial tem que ser tomado para não alterar o valor de algum operando utilizando a instrução MOVE.

Embora o tamanho das instruções é reduzido, o número de instruções requeridas aumenta de 4 para 6. Finalmente, no caso de instruções com um operando o segundo operando fica implícito, utilizando-se um registrador da CPU (acumulador – AC) para auxiliar na resolução da operação. Neste caso são necessárias oito instruções para realizar a operação.

O número de endereços por instrução resulta em uma CPU menos complexa, uma vez que as instruções têm uma extensão menor e são mais primitivas. Em contrapartida, a quantidade de instruções por programa é maior o que geralmente resulta em programas mais complexos e maior tempo de execução.

O número de endereços também implica na utilização de múltiplos registradores, tornando a execução de operações mais rápidas. Desta forma, a definição do número de endereços por instrução constitui uma decisão de projeto importante.

Praticando

1. Analise o exemplo apresentado e descreva em português a seqüência de instruções de máquina utilizadas em cada caso.
2. Computar o seguinte comando utilizando instruções de um, dois e três operandos:

$$Z = (C \times B) \div (A \times E - D)$$

A partir do número de operandos explicitamente referenciados e da localização destes operandos, é possível classificar as arquiteturas nos seguintes tipos:

- **Arquiteturas memória-memória:** as instruções aritméticas/lógicas usam três operandos e todos os operandos podem estar na memória;
- **Arquiteturas registrador-memória:** as instruções aritméticas/lógicas usam dois operandos, sendo que apenas um deles pode residir na memória;
- **Arquiteturas registrador-registrador:** as instruções aritméticas/lógicas usam três operandos, todos em registradores²⁹. Neste caso, apenas duas instruções acessam diretamente a memória: LOAD e STORE. A instrução LOAD carrega em um registrador um dado armazenado na memória e instrução STORE armazena na memória o conteúdo de um registrador.

As arquiteturas também podem ser classificadas em relação à localização dos operandos:

- **Arquiteturas memória-memória:** as instruções aritméticas/lógicas usam três operandos e todos os operandos podem estar na memória;
- **Arquiteturas registrador-memória:** as instruções aritméticas/lógicas usam dois operandos, sendo que apenas um deles pode residir na memória;
- **Arquiteturas registrador-registrador:** as instruções aritméticas/lógicas usam três operandos, todos em registradores. Neste caso, apenas duas instruções acessam diretamente a memória: LOAD e STORE. A instrução LOAD carrega em um registrador um dado armazenado na memória e instrução STORE armazena na memória o conteúdo de um registrador.

²⁹ Registradores: Os registradores são memórias utilizadas para realizar as operações na CPU. Essas memórias são muito rápidas e de tamanho pequeno, geralmente suficiente para guardar menos de uma dezena de valores.

Atividades de avaliação



1. As linguagens de máquina são específicas de uma dada máquina. Que significa que cada tipo de computador tem sua própria linguagem de máquina? Explique por que isto acontece.

Modos de endereçamento

Grande parte das instruções tem operandos, os quais devem ser localizados para realizar a operação. Os operandos podem encontrar-se em registradores, na memória principal ou embutidos implicitamente na própria instrução. O modo de endereçamento determina a forma em que uma instrução especifica a localização dos seus operandos. Existem três modos de endereçamento básicos:

- **Modo registrador:** a instrução indica explicitamente o registrador de dados onde se encontra o operando (fonte ou destino);
- **Modo imediato:** o operando se encontra dentro do próprio código da instrução;
- **Modo implícito:** a localização do operando não é indicada explicitamente na instrução. Por exemplo, nas chamadas arquiteturas acumulador, um dos operandos-fonte e o operando-destino nas instruções aritméticas/lógicas encontra-se sempre em um registrador especial, o acumulador.

Existem diferentes arquiteturas que podem ser utilizadas para o endereçamento de dados armazenados na memória principal. Um modo de endereçamento referente à memória indica como deve ser obtido o endereço da locação de memória onde se encontra o dado que será acessado. Este endereço é chamado endereço efetivo.

Apesar da variedade mencionada, é possível identificar alguns modos de endereçamento referentes à memória que são oferecidos pela maioria das arquiteturas. Por outro lado, quase todas as arquiteturas fornecem mais de um modo de endereçamento. Todos estes modos envolvem decisões em relação a quantidade de posições de memória endereçáveis e/ou flexibilidade de endereçamento e/ou complexidade do cálculo. Estes modos de endereçamento mais comuns estão relacionados na tabela a seguir.

A forma mais simples de endereçamento é o **endereçamento imediato** onde o valor do operando é especificado diretamente na instrução, sem necessidade de acesso à memória.

OPERANDO = A

Este modo geralmente é utilizado na definição e uso de constantes e inicialização de variáveis. Este modo tem como desvantagem que o tamanho do operando fica limitado pelo tamanho do campo endereço na instrução.

No modo **endereçamento direto**, o endereço efetivo é um valor imediato contido no campo endereço da instrução. Por exemplo, na instrução

ADD (100), R1

um dos operandos encontra-se na locação de memória com endereço 100. O significado da instrução acima é equivalente a:

$$R1 \leftarrow M [100] + R1$$

Este modo de endereçamento é usado principalmente no acesso às variáveis estáticas de um programa, cujo endereço em memória pode ser determinado durante a compilação do programa em um único acesso a memória. Por outro lado a faixa de endereços é limitada por conta do tamanho do campo.

No **modo indireto via registrador**, o endereço efetivo encontra-se em um registrador. Por exemplo, na instrução:

ADD (R1), R2

Um dos operandos encontra-se na locação de memória cujo endereço se encontra no registrador R1. Ou seja, existe uma indireção para chegar ao endereço efetivo do operando na memória, através de um registrador que contém essa informação. A instrução acima é equivalente à seguinte expressão:

$$R2 \leftarrow M [R1] + R2$$

Este modo de endereçamento é usado no acesso a variáveis dinâmicas, cujo endereço na memória é conhecido apenas durante a execução do programa. O acesso a uma variável dinâmica é realizado através de um ponteiro, que nada mais é do que o endereço da variável. Para realizar o acesso à variável dinâmica, o ponteiro é carregado em um registrador, e a instrução que acessa a variável usa este registrador com o modo de endereçamento indireto.

No **modo relativo à base**, o endereço efetivo é obtido a partir da soma do conteúdo de um registrador, chamado endereço-base, com um valor imediato contido na instrução, chamado deslocamento. Este modo de endereça-

mento é bastante poderoso e combina as capacidades do endereçamento direto e do indireto via registrador. Por exemplo, na instrução:

ADD 100(R1), R2

O registrador R1 contém o endereço-base e 100 é o deslocamento. O endereço efetivo do operando na memória é calculado a partir da soma do conteúdo de R1 com o valor 100.

No caso da instrução acima, ela pode ser expressa como:

$R2 \leftarrow M [100 + R1] + R2$

O modo relativo à base é usado no acesso a componentes de variáveis dinâmicas estruturadas (por exemplo, *record* em Pascal ou *struct* em C). A figura 28 a seguir mostra como é realizado o cálculo do endereço efetivo no modo de endereçamento relativo à base.

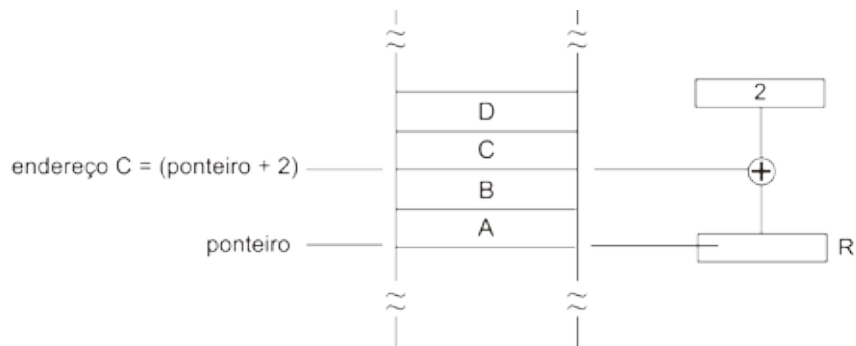


Figura 28 - Modelo de realização do cálculo do endereço

A figura mostra a localização na memória de uma estrutura com quatro campos A, B, C e D. O endereço inicial da estrutura é indicado por um ponteiro, que torna-se conhecido apenas durante a execução do programa. No entanto, a posição de cada campo em relação ao início da estrutura é fixo, sendo conhecido durante a compilação. O endereço de um campo é obtido somando-se a posição do campo (o deslocamento) ao ponteiro que indica o início da estrutura (o endereço-base). Por exemplo, na figura, para somar um valor ao campo C, o compilador pode usar a instrução:

ADD 2(R1), R2

Precedida de uma instrução para carregar em R1 o endereço-base da estrutura.

No **modo indexado**, o endereço efetivo é calculado somando dois registradores mais um deslocamento (opcional). Um registrador é a base e o outro o índice. Por exemplo, na instrução:

ADD (R1 + R2), R3

R1 contém o endereço-base, e R2 o índice. A interpretação para este modo de endereçamento é o seguinte:

$$R3 \leftarrow M [R1 + R2] + R3$$

O modo indexado é normalmente usado eficientemente para implementar operações iterativas, como por exemplo, no acesso aos elementos de um vetor. A figura embaixo mostra como é calculado o endereço efetivo no modo de endereçamento indexado.

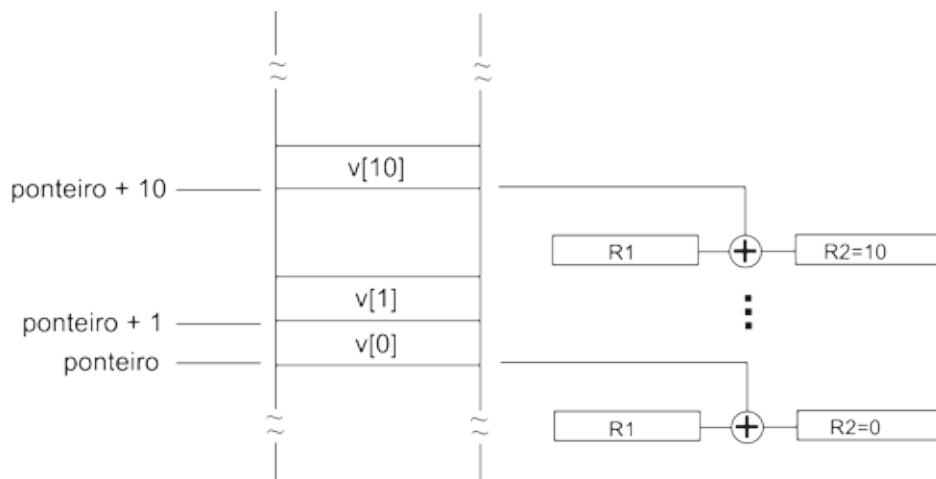


Figura 29 - Modelo de realização de cálculo do endereço efetivo

Na figura é representada a localização de um vetor V na memória. Um ponteiro indica o endereço-base do vetor indicando a posição do primeiro elemento. Considerando que o vetor é uma estrutura homogênea, cada elemento do vetor possui o mesmo tamanho e estrutura. Desta forma é possível calcular a posição na memória do próximo elemento do vetor somando o índice do elemento ao endereço-base.

Para realizar o acesso seqüencialmente os elementos do vetor, o índice é inicialmente carregado no registrador com o valor 0. O índice é então incrementado dentro de um laço após o acesso a cada elemento. Por exemplo, para somar um valor em registrador aos elementos do vetor, o compilador pode usar as seguintes instruções em um laço:

ADD R1, (R2 + R3)

ADD 1, R3

Onde R1 contém o valor a ser somado, R2 contém o ponteiro para o vetor e R3 é o registrador com o índice, com valor inicial 0.

Atividades de avaliação



1. Quais são os modos de endereçamento mais comuns? Apresente um exemplo de uso para cada caso (diferente do apresentado no texto).
2. Por que é dito que o modo de endereçamento relativo à base combina as capacidades do endereçamento direto e do indireto via registrador? Explique.
3. O processo de decodificação da instrução é crítico. Como você justificaria esta afirmação?

CISC versus RISC

Os primeiros computadores eram baseados em conjuntos de instruções simples e pequenos. No entanto, a procura por maior desempenho fez com que as instruções se tornassem mais complexas e poderosas.

Com o objetivo de diminuir a grande lacuna semântica entre as necessidades das pessoas expressas em linguagens de alto nível e o que o computador pode fazer, os projetistas de computadores investiram na definição de instruções complexas.

Na medida em que as instruções de máquina ficaram mais próximas das instruções de linguagens de alto nível, aparentemente a tarefa dos compiladores ficou simplificada. No entanto, a utilização de um conjunto de instruções mais rico requer de uma unidade de controle mais complexa capaz de interpretar³⁰ e executar as instruções, aumentando conseqüentemente o tempo de execução de cada instrução individual. Por outro lado, outras atividades do compilador tais como otimização do código gerado pode ser dificultada. Um exemplo de arquitetura CISC é *Pentium* da Intel.

Desta forma, considerando que os benefícios na utilização de conjuntos de instruções complexos não se apresentaram muito claros, novos estudos sobre o comportamento da execução de programas em linguagens de alto nível orientaram o projeto de um novo tipo de arquitetura: o computador com um conjunto reduzido de instruções (RISC).

Arquiteturas RISC compartilham as seguintes características:

- Conjunto limitado de instruções simples com formato fixo, facilitando a decodificação e acesso aos operandos.
- Uma instrução por ciclo pode ser implementada diretamente pelo hardware, reduzindo o tempo de execução (não precisavam de interpretação).
- Grande número de registradores ou o uso de um compilador que otimize o uso dos mesmos.
- Operações de registrador para registrador.

³⁰ Interpretar: no processo de interpretação, cada instrução é examinada e decodificada individualmente, para ser imediatamente executada. Diferentemente do que na tradução, nenhum programa traduzido é gerado.

- Método de endereçamento simples de registrador.
- Enfoque na otimização do uso da pipeline de instruções.

Exemplos de computadores com arquitetura RISC são: **Sun** e **UltraSparc**.

Esses novos processadores tinham diferenças significativas em relação aos processadores existentes. Esta incompatibilidade se tornou favorável para o desenvolvimento de um novo conjunto de instruções de forma a otimizar o desempenho do sistema. Os processadores denominados RISC possuíam um conjunto de instruções em torno de 50, enquanto que nos computadores CISC existentes na época, o conjunto de instruções era entre 200 e 300.

A estratégia por trás do funcionamento de processadores RISC é que embora fosse necessário de 4 a 5 instruções simples para fazer o equivalente a uma instrução CISC (*Complex Instruction Set Computer*), se as instruções RISC fossem suficientemente rápidas o desempenho global do sistema seria maior, uma vez que estas instruções não eram interpretadas.

Houve dois principais motivos que impediram a predominância das máquinas RISC, mesmo com um desempenho maior:

1. O grande investimento das organizações em tecnologia Intel, incompatível com os novos processadores.
2. A Intel desenvolveu uma arquitetura híbrida combinando o melhor das duas abordagens: o núcleo de instruções mais comuns funcionam de modo RISC, enquanto as instruções mais complicadas e menos frequentes, no modo CISC. Isto foi implementado a partir da 486.

A solução da Intel tornou os computadores competitivos a nível de desempenho global, e ainda manteve a compatibilidade com *softwares* antigos.

1. Princípios de projeto para computadores modernos

Dado o estado atual da tecnologia de *hardware*, alguns princípios de projeto para computadores (CPUs de uso geral), passaram a ser aceitos. Muitos desses princípios têm sua origem no projeto de arquitetura RISC:

- Todas as instruções são executadas diretamente pelo *hardware* eliminando desta forma um nível de interpretação. Este princípio propicia maior velocidade na execução das instruções.
- De forma a minimizar o problema do gargalo no acesso à memória e aumentar a velocidade de execução das instruções, a técnica de busca antecipada ou *prefetch* consiste na busca antecipada de instruções da memória, antes mesmo de ser utilizada, de modo a encontrá-las no processador quando chegar o momento de executá-las.

Tais instruções eram armazenadas em um conjunto de registradores chamados de *buffer* de pré-busca. Deste modo o processo de busca pode ser sobreposto à execução normal.

- Maximização da taxa de execução das instruções. O paralelismo³¹ na execução das instruções pode representar uma grande melhoria no desempenho.
- Instruções fáceis de decodificar. Formatos regulares, de comprimento fixo e com poucos campos facilita a decodificação, aumentando a taxa de emissão de instruções.
- Referencia a memória somente através de LOAD e STORE. Uma vez que o acesso a memória tem um alto custo em tempo, deve ser priorizado o uso de registradores para o armazenamento de operandos durante a execução das instruções.
- Utilização de vários registradores (no mínimo 32) de forma a evitar ao máximo o processo de descarregamento de memória.

³¹. Paralelismo: o paralelismo tem duas formas gerais: paralelismo no nível de instrução e paralelismo no nível do processador. No primeiro caso, uma única CPU é capaz de executar um maior número de instruções por segundo. No segundo caso, várias CPUs trabalham juntas no mesmo problema.

Síntese do capítulo



Neste capítulo foi apresentado o conceito central de instrução, suas características, componentes, tipos e modos de endereçamento. Foi apresentada a noção de máquina multinível, e a forma de funcionamento.

Com base no conceito de instrução e linguagem de máquina, foi discutida a importância da ISA -Arquitetura do Conjunto de Instruções, no contexto do projeto da arquitetura do computador, sua influência e restrições. Foram analisadas as propriedades das duas principais arquiteturas: RISC e CISC, fazendo uma análise de vantagens e desvantagens de cada uma delas, as quais serviram de base para a elicitação dos princípios do projeto dos computadores modernos.

Atividades de avaliação



1. Especifique as características das arquiteturas RISC e CISC fazendo um comparativo de vantagens e desvantagens de cada uma delas.
2. Qual é a causa principal do alto desempenho da arquitetura RISC?

3. Por que, mesmo atingindo um desempenho melhor, a arquitetura RISC não atingiu as expectativas do mercado?
4. Para que serve o buffer de pré-busca?
5. Qual é a maior vantagem de usar paralelismo?
6. De forma a ilustrar as abordagens de paralelismo, pesquise os conceitos de *Pipelining* (paralelismo no nível de instrução) e Computadores Matriciais (paralelismo no nível de processador).

Leituras, filmes e sites



Intel Corporation. <http://www.intel.com/>

Processadores Intel <http://www.intel.com/portugues/products/processor/index.htm>

Sparc International <http://www.sparc.org/>

Sun Microsystems Brasil <http://br.sun.com/>

Referências



TANENBAUM, ANDREW S. **Organização Estruturada de Computadores**. 5 ta. Edição. Editora Pearson. 2007.

STALLINGS WILLIAM. **Estrutura e Organização de Computadores**. 5 ta. Edição. Editora Pearson. 2004.

BROOKSHEAR GLENN J. **Ciência da Computação – Uma visão abrangente**. 5 ta. Edição. Editora Bookman. 2000.

Sobre os autores

Marcial Porto Fernandez: possui graduação em Engenharia Eletrônica e Computação pela Universidade Federal do Rio de Janeiro (1988), mestrado em Engenharia Elétrica pela Universidade Federal do Rio de Janeiro (1998) e doutorado em Engenharia Elétrica pela Universidade Federal do Rio de Janeiro (2002). Atualmente é professor adjunto da Universidade Estadual do Ceará e assessor técnico da Empresa de Tecnologia da Informação do Ceará. Tem experiência na área de Ciência da Computação, com ênfase em Teleinformática, atuando principalmente nos seguintes temas: gerenciamento de redes, redes óticas, redes móveis e sem fio e arquitetura de computadores.

Mariela Inés Cortés: atualmente é Professora Adjunta de da Universidade Estadual do Ceará (UECE). Possui doutorado em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio, 2003), mestrado em Sistemas e Computação pelo Instituto Militar de Engenharia (IME, 1998) e graduação pela Universidad Nacional de La Plata, Argentina - Analista de Computación (1996). Tem experiência na área de Ciência da Computação, com ênfase em Evolução de software e gerência de projetos de software, atuando principalmente nos seguintes temas: evolução de *framework*, *refactorings*, *agent societies* e *multi-agent systems*, gerencia de projetos de desenvolvimento de software.



Computação

Fiel a sua missão de interiorizar o ensino superior no estado Ceará, a UECE, como uma instituição que participa do Sistema Universidade Aberta do Brasil, vem ampliando a oferta de cursos de graduação e pós-graduação na modalidade de educação a distância, e gerando experiências e possibilidades inovadoras com uso das novas plataformas tecnológicas decorrentes da popularização da internet, funcionamento do cinturão digital e massificação dos computadores pessoais.

Comprometida com a formação de professores em todos os níveis e a qualificação dos servidores públicos para bem servir ao Estado, os cursos da UAB/UECE atendem aos padrões de qualidade estabelecidos pelos normativos legais do Governo Federal e se articulam com as demandas de desenvolvimento das regiões do Ceará.

