

INSTITUTO FEDERAL

Triângulo Mineiro

Campus Uberlândia Centro

APOSTILA DE
INTRODUÇÃO À INFORMÁTICA

Prof. Walteno Martins Parreira Júnior

www.waltenomartins.com.br

waltenomartins@iftm.edu.br

2019

SUMÁRIO

| | | |
|----------|--|----|
| 1 | Sistema de Computação Digital | 2 |
| 1.1 | Conceitos | 2 |
| 1.2 | Componentes de um Sistema de Computação..... | 7 |
| 2 | História da Computação | 9 |
| 2.1 | O Computador | 9 |
| 2.2 | A História dos Computadores..... | 10 |
| 3 | SISTEMAS DE NUMERAÇÃO | 32 |
| 3.1 | Sistemas de Numeração Posicionais..... | 33 |
| 3.2 | Conversões entre Bases | 36 |
| 3.3 | Aritmética em Binário | 40 |
| 3.4 | Aritmética em outras bases..... | 43 |
| 4 | Lógica Booleana | 45 |
| 4.1 | Proposição | 45 |
| 4.2 | As três leis do pensamento | 45 |
| 4.3 | Proposições | 45 |
| 4.4 | Tabelas-verdade e conectivos..... | 47 |
| 5 | OPERAÇÃO DO MICROPROCESSADOR | 54 |
| 4.1 | Arquivos | 54 |
| 4.2 | Programas | 55 |
| 4.3 | Sistema Operacional | 56 |
| 6 | COMPONENTES DO MICROPROCESSADOR | 59 |
| 5.1 | Introdução | 59 |
| 5.2 | Unidade Lógica e Aritmética (ULA)..... | 60 |
| 5.3 | Unidade de Controle (UC)..... | 61 |
| 5.4 | Relógio..... | 61 |
| 5.5 | Registradores | 62 |
| 5.6 | Decodificador de Instrução..... | 64 |
| 5.7 | Barramento | 65 |
| 5.8 | Memória..... | 68 |
| 5.9 | Dispositivos de entrada e saída..... | 72 |
| 7 | CISC x RISC | 74 |
| 7.1 | Apresentação | 74 |
| 7.2 | A arquitetura RISC | 75 |
| 7.3 | Pipeline | 76 |
| 8 | Instruções de Máquina | 79 |
| 7.1 | Tipos de Instruções | 79 |
| 7.2 | Formato das Instruções | 79 |
| 7.3 | Modos de Endereçamento | 80 |

1 Sistema de Computação Digital

Um computador é uma máquina formada por um conjunto de peças eletrônicas e eletromecânicas capaz de coletar, manipular e fornecer resultados da manipulação destas informações para um ou mais objetivos.

Processamento é a manipulação das informações coletadas, enquanto as informações coletadas são denominadas de dados. Assim, dado é a matéria prima obtida e a informação é o resultado do processamento dos dados coletados.

Todos os computadores modernos possuem uma parte importante, denominada de Unidade Central de Processamento (UCP). Em inglês, usa-se a sigla CPU, que é a abreviatura de Central Processing Unit. Nos computadores de grande porte, a CPU é formada por uma ou várias placas. Cada uma dessas placas contém vários chips. Nos microcomputadores a CPU nada mais é que o próprio processador. É também comum chamar a placa que contém o processador de placa de CPU ou placa mãe. Não quer dizer que a CPU seja a placa inteira. A placa de CPU é a placa que contém a CPU, ou seja, que contém o processador.

O microprocessador, também conhecido como CPU ou UCP, é uma pastilha de silício denominado de chip que mantém as funções de processamento e controle de instruções, localizado sobre a placa mãe do computador. Esse chip sofreu transformações tecnológicas ao longo dos anos, proporcionando aos computadores um aumento considerável em seu poder computacional e na sua flexibilidade de uso. Paralelamente à evolução das CPU, os computadores passaram a ser utilizados por um número cada vez maior de pessoas, pois na medida em que as máquinas passaram a ter uma alta demanda, o preço teve considerável redução, sendo essa uma tendência seguida até os dias atuais.

Dessa forma, os microprocessadores têm conduzido a evolução tecnológica da computação, pois assim que novos chips são lançados no mercado, são também lançados softwares e dispositivos mais poderosos, com o intuito de proporcionar maior rapidez, flexibilidade e confiabilidade na execução de tarefas.

1.1 Conceitos

Bit - Binário, (contração das palavras binary digit), sistema usado em computação que consiste em apenas dois valores, 1 e 0. Todas as operações que são possíveis no sistema decimal, assim como processamento de texto, sons, imagens e outros dados analógicos podem ser feitos usando o sistema digital. É a menor unidade de informação armazenável em um computador.

Byte - É uma unidade de armazenamento de dados. Cada byte é formado por 8 bits e é suficiente para 256 combinações diferentes. É por isso que no sistema de caracteres ASCII é usado um byte para representar cada caractere. Com 256 combinações é possível incluir todas as letras, números e ainda um punhado de caracteres especiais e símbolos. No conjunto Unicode, onde são incluídos também caracteres do Japonês, Chinês, Árabe e outras línguas são usados 2 bytes para cada caractere, o que permite 65 mil caracteres diferentes.

BIOS - A BIOS (Basic Input output system) é uma série de instruções gravadas na

ROM que quando o computador é inicializado essas instruções são interpretadas e executadas. Também pode ser definida como um programa de computador pré-gravado em memória permanente, são extremamente relacionadas à configuração do hardware e é executado toda vez que o pc é ligado. Tem como funções:

- Dar suporte básico de acesso ao hardware do pc;
- Fazer a carga do Sistema Operacional na memória;
- Realizar verificação funcional dos circuitos do PC
- Responsável por carregar o programa de boot, durante a inicialização do sistema.

Ciclo de instrução - Também chamado de ciclo de busca e execução ou ciclo busca-execução é o período de tempo no qual um computador lê e processa uma instrução em linguagem de máquina da sua memória ou a seqüência de ações que a CPU realiza para executar cada instrução em código de máquina num programa.

Compactadores de Arquivos – são softwares (programas) capazes de compactar e descompactar arquivos, em formato único ou variado. Basicamente estes programas fazem um processo de reduzir o tamanho físico de um bloco de informação, ou basicamente arquivos. O processo de compactação está associado ao de compressão e o processo de descompactação está associado ao de descompressão, ambas ligadas a fases distintas e contrárias.

Compilador - Os computadores não entendem nada além de comandos, dados e endereços escritos em linguagem binária. Mas, qualquer ser humano que se disponha a tentar desenvolver um programa complexo programando diretamente em linguagem de máquina terá muita dificuldade antes de concluir seu trabalho. Para resolver este problema, surgiram as linguagens de programação, que permitem escrever programas usando comandos fáceis de lembrar e funções prontas. Logo, o compilador é um programa que permite transformar este código escrito na linguagem de programação em linguagem de máquina, gerando o binário que pode ser executado

CPU - CPU é a sigla para Central Process Unit, ou Unidade Central de Processamento. Ele é o principal item de hardware do computador, que também é conhecido como processador. A CPU é responsável por calcular e realizar tarefas determinadas pelo usuário e é considerado o cérebro do PC.

Disco rígido ou Hard Disk (HD) é um sistema de armazenamento de alta capacidade, que por não ser volátil, é destinado ao armazenamento de arquivos e programas. Apesar de não parece, o HD é um dos componentes que compõe um PC, que envolve mais tecnologia. Todos os programas e arquivos são armazenados no disco rígido, também chamado de Winchester.

Dispositivos de Entrada e Saída – São também chamados de periféricos. Utilizando estes dispositivos o computador pode armazenar, ler, transmitir e receber dados. O termo “Entrada e Saída” é abreviado por E/S, ou para quem preferir na língua inglesa, I/O (Input/Output). Dentre os diversos dispositivos de E/S, existem alguns que são especializados apenas em entrada, outros especializados apenas em saída e outros em ambas as ações. Maiores detalhes no item 3.9.

Drive - Em Portugal ao invés do termo drive é usado o termo "acionador", que explica bem o significado do termo. Os drives de disquetes, drives de CD-ROM, etc. são os mecanismos que permitem ler estas mídias. O HD também é um drive, a única diferença é que ao contrário de um disquete ou CD-ROM, os discos são lacrados.

Driver (de dispositivo) - Conjunto de rotinas que permitem ao sistema operacional acessar um periférico. O driver funciona como uma espécie de tradutor entre o dispositivo, uma placa de vídeo por exemplo e o sistema operacional ou programas que o estejam utilizando. Por ser especializado, o driver funcionará adequadamente apenas junto com o qual foi escrito. Em geral existe uma versão diferente do driver par cada sistema operacional.

Emulador - Um programa que simula o ambiente de um outro computador. Existem atualmente emuladores para PC que emulam quase todos os equipamentos antigos, desde vídeo games até sistemas como o MSX e Amiga. O problema dos emuladores é que por trabalharem traduzindo instruções, sempre há perda de desempenho.

Endereço IP - Identificam cada micro na rede. A regra básica é que cada micro deve ter um endereço IP diferente e todos devem usar endereços dentro da mesma faixa. Um endereço IP é composto de uma seqüência de 32 bits, divididos em 4 grupos de 8 bits cada. Cada grupo de 8 bits recebe o nome de octeto. Veja que 8 bits permitem 256 combinações diferentes. Para facilitar a configuração dos endereços, usamos números de 0 a 255 para representar cada octeto, formando endereços como 224.45.10.222, 231.175.34.7 etc. O endereço IP é dividido em duas partes. A primeira identifica a rede à qual o computador está conectado (necessário, pois numa rede TCP/IP podemos ter várias redes conectadas entre sí, veja o caso da Internet) e a segunda identifica o computador (chamado de host) dentro da rede. Quando um computador conecta na Internet ele recebe um endereço IP válido, emprestado pelo provedor de acesso, algo como por exemplo "242.221.223.34". É através deste número que outros computadores na Internet podem enviar informações e arquivos para o computador.

FCC ID - O código de identificação fornecido pela FCC (Federal Communications Commission) que vem decalcado em quase todos os periféricos, modems, placas de som, placas mãe etc. o formato mais comum é FCCID: XXX-#####-YYY onde o "XXX" é o código do fabricante e o "#####-YYY" é o código do produto. Este número pode ser usado para identificar uma placa quando estiver em busca de drivers ou manuais. A busca pode ser efetuada através do site <http://www.fcc.gov/oet/fccid/> .

Firmware - Conjunto de instruções essenciais para o funcionamento de um dispositivo. Atualmente encontramos firmwares em vários dispositivos, tais como modems, gravadores de CD, etc. O firmware é armazenado em um chip de memória ROM ou memória Flash no próprio dispositivo. Em muitos, casos, o firmware pode ser atualizado via software, por exemplo com o objetivo de destravar um drive de DVD ou atualizar um modem de 33.6 para 56 K.

Freeware - Programa que pode ser usado gratuitamente. Não é a mesma coisa que de domínio público. O programa continua pertencendo ao seu criador. Também não é a mesma coisa que software livre, já que o programa não possui código aberto.

Hardware - É a parte física de um computador, é formado pelos componentes eletrônicos, como por exemplo, circuitos de fios e luz, placas, utensílios, correntes, e qualquer outro material em estado físico, que seja necessário para fazer com o que computador funcione. O hardware é basicamente utilizado por computadores e elementos eletrônicos. Qualquer equipamento físico como chaves, fechaduras, correntes e peças do próprio computador. O hardware não se limita apenas a computadores pessoais, também está disponível em automóveis, celulares, tablets e etc. A impressora, o scanner, o monitor, mouse e o teclado de um computador é considerado hardware deste PC. Em suma, os hardwares são todos os periféricos conectados em um sistema operacional.

HDTV - High-definition TV, um padrão de TV's de alta resolução, com 1080 linhas de resolução horizontal e um formato de tela de 16:9, o mesmo formato retangular das telas de cinema. Apesar de ainda serem caros, estes aparelhos vão substituir os aparelhos atuais nos próximos anos. Muitas emissoras de TV já estão produzindo no novo formato.

Instrução de Máquina - As instruções correspondem a seqüências muito simples de operações, como transferir um dado em memória para a CPU ou somar dois valores e são normalmente interpretadas por micro-código. Todo computador possui um conjunto de instruções que seu processador é capaz de executar. Essas instruções, chamadas de código de máquina, são representadas por seqüências de bits, normalmente limitadas pelo número de bits do registrador principal da CPU.

Interface - É um meio de comunicação ou de transmissão de informações. É um termo bastante genérico, que pode ser usado em relação a uma interface gráfica, que facilita o uso de um programa, a uma porta de transmissão de dados, como por exemplo as interfaces IDE da placa mãe, que permitem a troca de dados entre o processador e os discos rígidos, as interfaces seriais e paralelas, que permitem a conexão de mouses e impressoras, entre outros dispositivos, ou ainda a um joystick ou teclado, que novamente formam um tipo de interface entre o usuário e o programa ou jogo que está sendo executado.

IP - O endereçamento IP é importante, já que é ele que permite que as redes e hosts que formam a Internet sejam capazes de se comunicar entre si. Existem duas versões do protocolo IP: o IPV4 que é a versão atual, que é utilizado na maioria das situações, enquanto o IPV6 é a versão atualizada, que prevê um número muito maior de endereços e deve se popularizar a partir de 2012, quando os endereços IPV4 começarem a se esgotar. No IPV4, os endereços IP são compostos por 4 blocos de 8 bits (32 bits no total), que são representados através de números de 0 a 255, como "200.156.23.43" ou "204.245.32.11". Os grupos de 8 bits que formam o endereço são chamados de "octetos", o que dá origem a expressões como "o primeiro octeto do endereço". De qualquer forma, a divisão dos endereços em octetos e o uso de números decimais serve apenas para facilitar a configuração para nós, seres humanos. Quando processados, os endereços são transformados em binários, tais como "11001000100110010001011100101011".

Kernel - Em qualquer sistema operacional, o Kernel é o núcleo, encarregado de controlar o acesso à memória de demais componentes de Hardware, gerenciar os programas abertos, dividir o tempo de processador entre eles, etc. É a base, sobre a qual rodam as demais partes do sistema operacional, drives de dispositivo e programas.

Linguagem de Máquina – São instruções escritas em código binário, que dizem à CPU (Unidade Central de Processamento) quais circuitos exatamente devem ser ligados e desligados.

Memória – São todos os dispositivos que permitem a um computador guardar dados, temporariamente ou permanentemente. Existem vários tipos de memórias, tais como: flash, RAM, ROM, etc. Local do computador (hardware) onde se armazenam temporária ou definitivamente dados (números, caracteres e instruções).

Memória Flash - Diferentemente da memória RAM e também das SRAM, a memória Flash permite armazenar dados por longos períodos, sem precisar de alimentação elétrica. Graças a isso, a memória Flash se tornou rapidamente a tecnologia dominante

em cartões de memória, pendrives, HDs de estado sólido (SSDs), memória de armazenamento em câmeras, celulares e palmtops e assim por diante. As memórias Flash NOR chegaram a ser utilizadas nos primeiros cartões de memória PCMCIA e CompactFlash, mas elas desapareceram desse ramo quando foram introduzidas as memórias NAND, que são de longe o tipo mais usado atualmente. Nelas, cada célula é composta por dois transistores, com uma fina camada de óxido de silício precisamente posicionada entre os dois, que armazena cargas negativas. Isso cria uma espécie de armadilha de elétrons, que permite manter os dados por longos períodos de tempo, sem que seja necessário manter a alimentação elétrica (como nas memórias SRAM), ou muito menos fazer um refresh periódico (como na memória DRAM). Isso simplifica muito o design dos cartões, pendrives e outros dispositivos, pois eles precisam incluir apenas os chips de memória Flash NAND, um chip controlador e as trilhas necessárias. Nada de baterias, circuitos de refresh ou qualquer coisa do gênero.

Malware - Malicious Software, software malicioso. Corresponde a programas criados com a intenção de invadir sistemas, ou causar algum tipo de dano. Exemplos são trojans, vírus, backdoors, etc.

Palavra – Um conjunto de bits que representa uma informação útil para o computador. É um valor fixo e constante para um determinado processador, por exemplo 32 bits ou 64 bits. Na maioria dos computadores é a unidade de transferência e processamento e que normalmente possui um número múltiplo de 1 byte.

Periférico - Os periféricos são componentes que podem ser adquiridos e adicionados posteriormente, como placas de som, modems, impressoras, etc. É usado em relação a todos os componentes fora do trio placa-mãe, memória e processador.

Programa – É um conjunto de instruções ou declarações para ser executado por um computador e desenvolvido em uma linguagem de programação. Um programa em código de máquina consiste de uma seqüência de números que significam uma seqüência de instruções a serem executadas. É normal a representação da linguagem de máquina por meio de números (opcodes) constantes e variáveis em sistema binário ou sistema hexadecimal. Alguns computadores também têm seu opcodes representados no sistema octal. Ver item 2.2 para maiores informações.

Protocolo TCP/IP - Especifica o formato dos dados e das confirmações que os dois computadores trocam para oferecer uma transferência confiável e, também, os procedimentos de que se valem os computadores para assegurar que os dados cheguem corretamente. No caso do TCP ele apresenta como principais características transferência de dados confiável fim-a-fim (todo pacote transmitido requer um Ack que é um bit de reconhecimento, há recuperação de dados perdidos, descarte de dados duplicados e reorganização dos dados recebidos fora de ordem), comunicação bidirecional (full-duplex) entre cliente servidor, o sequenciamento: bytes de segmentos são numerados, de forma a garantir a entrega em ordem e a detecção e eliminação de duplicatas e é voltado para atuar sobre redes heterogêneas com tamanhos máximos de pacotes variáveis, faixas de passagem variáveis topologias distintas. O ponto fraco atual deste padrão é a adaptação a taxas de erros grandes, comum em comunicação sem fio (wireless). Quando cada segmento for transmitido é adicionado um checksum e quando estes são recebidos eles são verificados, se danificados os pacotes são descartados, como se tivessem se perdido pela rede e retransmitidos pela origem.

RAM - Random Access Memory, ou memória de acesso aleatório, indicando a principal característica da memória RAM, que é o fato de permitir o acesso direto a qualquer um dos endereços disponíveis e de forma bastante rápida. É a memória principal de um

computador.

Registadores - são pequenos espaços de memória utilizados pelo processador para armazenar informações que serão usadas para processar a instrução. Eles estão muito próximos das unidades de execução do processador e por isso as informações podem ser acessadas imediatamente, mas em troca eles são incrivelmente limitados em tamanho. A arquitetura x86 prevê o uso de apenas 8 registradores, com 32 bits cada um. Ou seja, todos os registradores somados armazenam apenas 8 bytes de dados.

ROM - Read Only Memory, memória apenas para leitura. Existem vários tipos, usada para guardar dados que não serão alterados. Exemplos de memória ROM são as EPROM. As memórias Flash são atualmente muito usadas para substituir chips de memória ROM, como por exemplo para armazenar o BIOS da placa mãe. Entretanto, memória Flash pode ser regravada, por isso não pode ser considerada memória ROM, apesar de substituí-la com vantagens.

Shareware - Programa que tem todas as funções da versão completa, mas só funciona por um certo tempo. Serve para que os interessados possam testar o programa antes de comprá-lo.

Software - É uma sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas. Também pode ser definido como os programas que comandam o funcionamento de um computador. Em um computador, o software é classificado como a parte lógica cuja função é fornecer instruções para o hardware. O software é constituído por todos os programas que existem para um referido sistema, quer sejam produzidos pelo próprio utente ou pelo fabricante do computador.

Spyware - Um software "espião", que uma vez instalado em um computador, reúne informações sobre os hábitos de navegação do usuário, ou mesmo dados mais pessoais, enviando-os ao dono quando o usuário se conectar à Internet.

Voz sobre IP - VoIP ou Voice Over IP. Este é um termo bastante divulgado atualmente, já que é um dos caminhos da telefonia. O sistema Voz sobre IP utiliza a Internet para realizar as chamadas de voz ao invés do sistema telefônico comutado. Com isso, as chamadas telefônicas sobretudo os interurbanos e chamadas internacionais tornam-se mais baratos.

1.2 Componentes de um Sistema de Computação

O processamento eletrônico de dados foi desenvolvido a partir dos conceitos da teoria de sistemas e por isto mesmo é chamado de sistema de computação. É sistema porque é um conjunto de partes que se coordenam, tais como o processador, a memória e os periféricos para atender um objetivo que é o de computar.

Computar é calcular. Os computadores foram desenvolvidos para realizar cálculos matemáticos, logo são máquinas de computar, de calcular e de realizar operações matemáticas.. Mesmo quando um processador está sendo utilizado para processar um texto, ele o faz por meio de cálculos matemáticos e o mesmo acontece quando realiza outros tipos de processamento, tais como tocar música ou vídeo e navegar na Internet.

Sistemas de processamento de dados são os responsáveis pela coleta, armazenamento, processamento e recuperação, usando equipamentos de

processamento eletrônico, dos dados necessários para o funcionamento do sistema de informações. O sistema de informações de uma organização pode ser conceituado como um conjunto de métodos, processos e equipamentos necessários para se obter, processar e utilizar informações nesta organização.

Em uma organização, os sistemas de informações são desenvolvidos em duas dimensões. Primeiramente os componentes da organização, isto é, os seus setores funcionais e o segundo, que é o nível de decisão, que segue uma hierarquia em três níveis:

- Nível operacional, que é de execução cotidiana e imediata, de competência dos escalões inferiores;
- Nível gerencial, de nível intermediário, de competência da gerencia setorial;
- Alto nível, de ação estratégica, de competência da diretoria.

A decisão em cada nível requer uma quantidade diferente de informações e em consequência disto, diferentes tipos de relatórios e disponibilidade de informações deve ser operacionalizado. Assim, um Sistema de Informações Gerenciais (SIG) é um sistema que engloba todos os componentes e todos os níveis de decisão de uma organização.

2 História da Computação

A Informática engloba toda atividade relacionada ao desenvolvimento e uso de equipamentos computacionais que permitem aprimorar e automatizar tarefas em qualquer área da sociedade. Pode-se definir a informática como a ciência do tratamento das informações. Muito mais que visar simplesmente à programação de computadores para executar tarefas específicas, a informática estuda a estrutura e o tratamento das informações sob suas mais variadas formas: números, textos, gráficos, imagens, sons, etc.

O computador em si intervém apenas como um instrumento para agilizar o tratamento da informação, e não como seu objetivo final. A informática busca criar uma abstração da realidade dentro de um sistema de computação, com o objetivo de reproduzi-la mais fielmente possível e assim poder substituí-la ou melhorar sua compreensão.

O profissional de Informática atua basicamente no desenvolvimento do que se pode chamar de um Sistema Computacional, o qual abrange a combinação de hardware (circuitos), software (programas) e outros elementos essenciais. A crescente evolução na área de Informática, particularmente no que diz respeito ao desenvolvimento de equipamentos de informática, tais como processadores cada vez mais velozes, novas tecnologias de armazenamento de dados e novos periféricos, aliada a queda nos preços do hardware, possibilitou um avanço das atividades relacionadas à informática em muitas atividades humanas, começando pelas Engenharias e atingindo as diversas áreas tais como a Medicina, as Artes, o Entretenimento, a Economia, etc...

E como consequência, o atendimento da necessidade de cada área quanto a informatização, os profissionais desenvolvem um conhecimento da tecnologia de Informática que seja útil na solução dos problemas relacionados com o contexto profissional da área.

2.1 O Computador

O computador é uma máquina composta de um conjunto de partes eletrônicas e eletromecânicas capaz de receber, armazenar, tratar e produzir informações de forma automática, com grande rapidez e precisão. A evolução dos sistemas de computação teve seu início no século 16, mas somente mostraram-se úteis no século 20, e sua popularização se deu graças a recente evolução na microeletrônica.

As famílias de computadores eram classificadas em 5 grupos distintos: os computadores pessoais (PCs), os minicomputadores, os superminicomputadores, os computadores de grande porte (mainframes) e os supercomputadores.

2.2 A História dos Computadores

A disputa pelo título de primeiro computador do mundo é acirrada. Isto porque devemos levar em consideração que os precursores dos primeiros computadores eram máquinas mecânicas. Aos poucos, com a utilização de eletricidade com chaves mecânicas e eletromagnéticas, chegou-se a utilização de computadores eletromecânicos. Mas está aí o ponto. As máquinas puramente mecânicas são ou não são computadores? E as eletromecânicas? Podemos considerá-las, como computadores?

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.

Se levarmos em conta que o termo COMPUTAR, significa fazer cálculos, contar, efetuar operações aritméticas, COMPUTADOR seria então o mecanismo ou máquina que auxilia essa tarefa, com vantagens no tempo gasto e na precisão. Lembrando que até o século XIX tinha-se a profissão de Computador, que era o profissional que fazia cálculos.

Inicialmente o homem utilizou seus próprios dedos para essa tarefa, dando origem ao sistema DECIMAL e aos termos DIGITAL e DIGITO . Para auxílio deste método, eram usados gravetos, contas ou marcas na parede, entre outros métodos de cálculo e armazenamento.

Com a evolução da humanidade, novas invenções para auxiliar os cálculos tornaram-se de fundamental importância e necessidade para o homem.

2.2.1 Os precursores

Ábaco (aprox. 3500 a.C.)

A palavra “cálculo” tem sua origem no termo latino cálculos que a milhares de anos servia para denominar pequenas pedras que eram usadas para contar deslizando-se por sulcos cavados no chão. Essa espécie de Ábaco foi descoberta em recentes escavações arqueológicas.

A partir do momento que o homem pré-histórico trocou seus hábitos nômades por aldeias e tribos fixas, desenvolvendo a lavoura e a criação de animais domésticos, tornou-se necessário um método para a contagem do tempo, delimitando as épocas de plantio e colheita, assim como os animais.

A partir deste elemento de cálculo, outros similares apareceram em diversos lugares do mundo, sendo chamados de ábaco. O mais antigo data de aproximadamente 3500 a.C., no vale entre os rios Tigre e Eufrates. Por volta do ano 2600 a.C. apareceu o ábaco chinês que evoluiu rapidamente e foi chamado em sua forma final de Suan-Pan, e de modo semelhante apareceu no Japão, o Soroban.

O ábaco (figura 1) constituiu portanto o primeiro dispositivo manual de cálculo,

que servia para representar números no sistema decimal e realizar operações com eles. O ábaco consiste numa moldura 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. 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 a vareta seguinte, à esquerda dessa vareta. O maior número que pode ser calculado pelo ábaco depende do número de varetas.

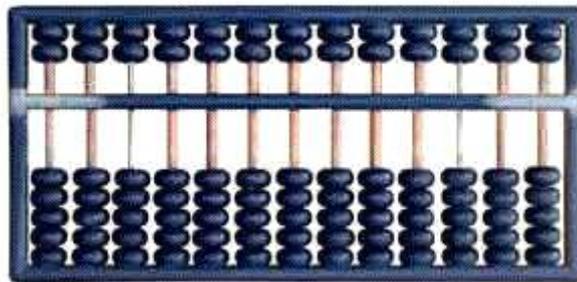


Figura 1 – Ábaco

Bastões de Napier (1610 -1614)

Passaram-se séculos sem que qualquer invenção fosse registrada até que fossem criados tais bastões. Eram tabelas moveis de multiplicação e divisão feitas de marfim. O responsável foi um nobre escocês chamado John Napier, inventor também dos logaritmos. Apesar de dispositivos semelhantes aos bastões terem sido usados desde o final do século XVI, eles só apareceram documentados a partir de 1614. Um conjunto completo de bastões de Napier consiste em 9 peças: uma para cada dígito de 1 a 9. Cada uma destas hastes é essencialmente uma coluna de uma tabela de multiplicação. Para obter o produto, os dígitos de cada diagonal são somados da direita para a esquerda.

Régua de Calculo (1621)

As tabelas de Napier influenciaram diretamente a invenção da régua de calculo, concretizada pelo matemático inglês William Oughtred com uma forma circular considerada como um dos primeiros dispositivos analógicos de computação. A Régua de Calculo e as calculadoras mecânicas foram largamente utilizadas até 1970, quando surgiram as calculadoras eletrônicas. E ainda são utilizadas em algumas regiões atrasadas do planeta.

2.2.2 Geração Zero (século XVII)

Os primeiros computadores, ou de geração zero, apareceram no século XVII e eram compostos exclusivamente por elementos mecânicos. Além disso,

caracterizavam-se por uma grande rigidez no que diz respeito aos programas a executar, uma grande parte deles são o que se chama hoje de máquinas dedicadas.

Calculadora de Pascal (1642)

Dos trabalhos conhecidos deste período, destaca-se o trabalho de Blaise Pascal, que em 1642 desenvolveu uma máquina de calcular totalmente mecânica. A máquina, também chamada de Pascaline (Fig. 2), era baseada na existência de um disco para cada potência de 10, cada disco sendo dotado de 10 dígitos (de 0 a 9). Embora fosse capaz de realizar apenas adições e subtrações, outras operações, como multiplicações e divisões podiam ser realizadas através da combinação das primeiras. Pascal esperava comercializar sua máquina, mas foi um fracasso comercial, apesar de ser uma importante invenção.



Figura 2: Pascaline

Calculadora de Leibnitz (1671)

Em 1671, o filósofo e matemático alemão de Leipzig, Gottfried Wilhelm von Leibnitz (21/06/1646 -14/11/1716) introduziu o conceito de realizar multiplicações e divisões através de adições e subtrações sucessivas. Em 1694, a máquina foi construída (figura 3) e apresentava uma certa evolução em relação à Calculadora de Pascal. Através de somas repetidas era capaz de efetuar multiplicações, também era capaz de realizar divisões, assim sendo capaz de executar as quatro operações básicas da matemática. Sua operação apresentou-se muito deficiente e sujeita a erros, tendo sido abandonada.

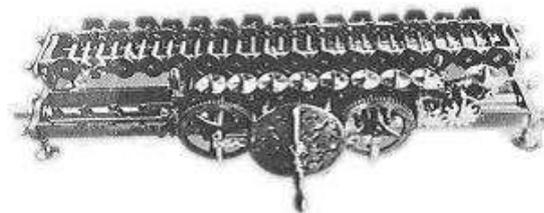


Figura 3: Calculadora de Leibnitz.

Máquina de Jacquard de Placa Perfurada (1801)

Joseph Marie Jacquard introduziu o conceito de armazenamento de informações em placas perfuradas, que não eram usadas especificamente em processamento de dados, mas para controlar uma máquina de tecelagem. Esse processo despertou, já nessa época, temor pelo desemprego, provocando uma grande reação popular contra essa espécie de pré-automação.

Arithmometer (1820)

Em 1820, Charles Xavier Thomas (1785-1870, conhecido como Thomas de Colmar, Paris-França) projetou e construiu uma máquina capaz de efetuar as 4 operações aritméticas básicas: a Arithmometer (figura 4). Esta foi a primeira calculadora realmente comercializada com sucesso: até 1850 vendeu-se cerca de 1500 Arithmometers. Ela fazia multiplicações com o mesmo princípio da calculadora de Leibnitz e com a assistência do usuário efetuava as divisões.

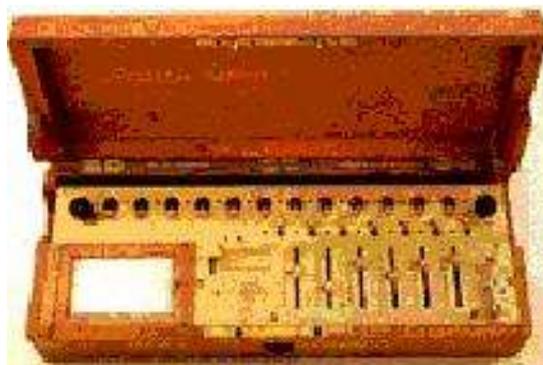


Figura 4: Arithmometer.

Maquina Diferencial de Babbage (1823)

Entre 1802 e 1822, Charles Babbage (1792-1871), um matemático e engenheiro britânico, construiu uma máquina (a máquina de diferenças) que baseava-se também no princípio de discos giratórios e era operada por uma simples manivela. Babbage é considerado o precursor dos modernos computadores eletrônicos digitais, sendo, considerado por muitos como o pai dos computadores modernos.

Esta máquina de diferenças surgiu devido à preocupação de Babbage com os erros contidos nas tabelas matemáticas de sua época. Esta máquina permite calcular tabelas de funções (logaritmos, funções trigonométricas, etc.) sem a intervenção de um operador humano. 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 os limites do ferramental industrial da época. Babbage se viu obrigado a desenhar peças e ferramentas, retardando 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, além de apresentar seus

resultados impressos.



Figura 5: Máquina de Diferenças.

Máquina Analítica (1833)

Em 1833, Babbage projetou uma máquina bastante aperfeiçoada (com o auxílio de Ada Lovelace), que chamou de Máquina Analítica. Ada é uma das poucas mulheres a figurar na história do computador. Matemática talentosa, compreendeu o funcionamento da Máquina Analítica e escreveu os melhores relatos sobre o processo. Ela criou programas para a máquina, tornando-se a primeira programadora de computador do mundo. A Máquina Analítica poderia ser programada para calcular várias funções diferentes, era constituída de unidade de controle de memória aritmética e de entrada e saída. Sua operação era governada por 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 concluir a construção.

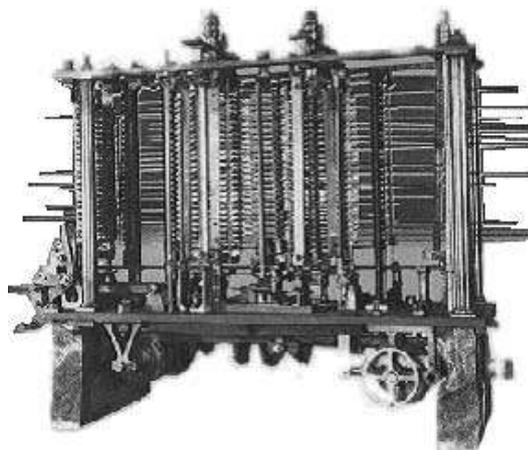


Figura 6: Máquina Analítica.

Máquina de Hollerith (1886)

Aproximadamente em 1885, o Departamento de Recenseamento dos E.U.A., percebeu que a conclusão do censo realizado em 1880 poderia demorar cerca de 8 anos para ser concluído e lançou um desafio para o censo de 1890 e Herman Hollerith que era funcionário do censo observou que a maioria das perguntas tinha como resposta sim ou não. Em 1886 idealizou um cartão perfurado que guardaria as informações coletadas no censo e uma máquina capaz de tabular essas informações e venceu o desafio. Construiu então a Máquina de Recenseamento ou Máquina Tabuladora. A máquina Tabuladora (figura 7) 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.



Figura 7: Máquina Tabuladora.

Foi Herman Hollerith, que concebeu a idéia de processar dados a partir de cartões perfurados (o problema a resolver era a computação de dados do censo dos Estados Unidos). Com esta solução, Hollerith conseguiu que o tempo de processamento dos dados do censo baixasse 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.

Dez anos mais tarde, Hollerith fundou uma empresa, a 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 de tabulação de dados. Com isso a empresa orientou suas atividades para o mercado externo, abrindo sua primeira filial fora dos Estados Unidos, no Canadá em 1917.

2.2.3 As máquinas de primeira geração (1930-1958)

Já no século XX, um grande número de projetos foram implementados, baseados na utilização de reles e válvulas eletrônicas (Fig. 8) para a realização de cálculos automaticamente, eram os computadores de primeira geração. Reles são eletroímãs cuja função é abrir ou fechar contatos elétricos com o intuito de interromper ou estabelecer circuitos. Válvula é um dispositivo que conduz a corrente elétrica num só sentido.

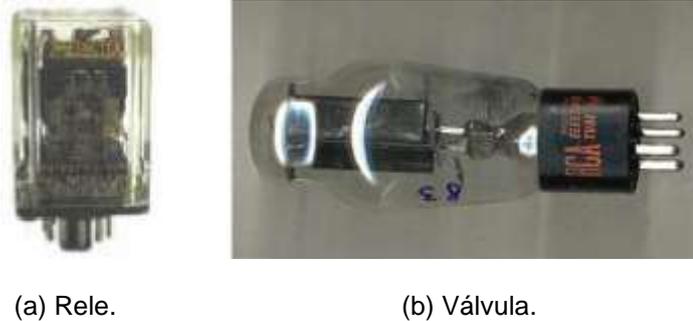


Figura 8: Dispositivos utilizados em computadores eletromecânicos.

Uma das grandes vantagens das máquinas a rele sobre as máquinas de calcular mecânicas era a maior velocidade de processamento (milésimos de segundos). Ainda, um outro aspecto positivo era a possibilidade de funcionamento contínuo, apresentando poucos erros de cálculo e pequeno tempo de manutenção.

Os computadores da primeira geração são todos baseados em tecnologias de válvulas eletrônicas. Normalmente queimavam após não muitas horas de uso. Tinham dispositivos de entrada / saída primitivos e calculavam com uma velocidade de milissegundos (milésimos de segundo). Os cartões perfurados foram o principal meio usado para armazenar os arquivos de dados e para dar entrada no computador. A grande utilidade dessas máquinas era no processamento de dados. No entanto tinham uma série de desvantagens como: custo elevado, relativa lentidão, pouca confiabilidade, quilômetros de fios (ocupando verdadeiros edifícios), 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). A seguir serão apresentados alguns destes computadores.



Figura 9: Válvula.

MARK I

O Mark I (fig. 10) foi criado entre 1937 e 1944, durante a II Guerra Mundial. Uma calculadora eletromecânica muito grande, idealizada por H. Aiken na Universidade de Harvard, foi considerado o primeiro projeto de computador.

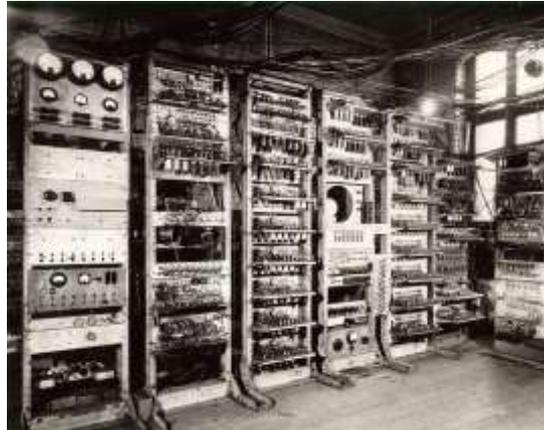


Figura 10: Mark I.

Utilizava muitas válvulas, as operações internas eram controladas por reles e os cálculos eram realizados mecanicamente. Integrava conceitos de computadores digitais e analógicos, pois tinha sistema eletrônico e mecânico na mesma máquina. Media 2,5 m de altura e 18 m de comprimento e pesava 70 toneladas. Era constituído por 7 milhões de peças móveis e sua fiação alcançava 800 Km. Com o apoio da IBM e da Marinha dos Estados Unidos, Howard Aiken, o pesquisador que desenvolveu Mark I, construiu outras versões deste computador (Mark II a Mark IV).

ABC (Atanasoff Berry Computer)

Criado em 1939. Foi o primeiro a usar válvulas para circuitos lógicos e o primeiro a ter memória para armazenar dados, princípio no qual se baseiam os computadores digitais. Atanasoff levou 4 princípios em consideração em seu projeto de computador:

- Usar eletricidade e eletrônica como meio;
- Recorrer à lógica binária para as operações;
- Usar um condensador para memória que pudesse ser regenerado para evitar intervalos;
- Calcular por ação lógica direta, não por via convencional de numeração.

Computadores no esforço de guerra

Em 1941, o computador Z3 (figura 11a) foi construído pelos alemães e tinha como principal função a codificação de mensagens. Porém foi destruído em Berlim nos deixando pouquíssimas informações.

Em 1943, assim como os alemães, os ingleses também foram em busca de

tecnologias para decifrar códigos secretos alemães. Foi construído o Colossus (figura 11b) por solicitação do Serviço de Inteligência Britânico. Possuindo dimensões gigantescas, o Colossus funcionava por meio de válvulas chegando a processar cerca de 5 mil caracteres por segundo ele foi inventado por Alan Turing.



Figura 11 – Os computadores Z3 (a) e Colossus (b)

ENIAC (Electronic Numeric Integrator and Calculator)

O ENIAC (fig. 12) foi criado entre 1943 e 1946 por John W. Mauchly e J. Prester Eckert Jr., juntamente com cientistas da Universidade da Pensilvânia. Foi considerado o primeiro grande computador digital. Não usava um programa de armazenamento interno. Os programas eram introduzidos por meio de cabos, o que fazia sua preparação para cálculos demorar semanas. Ocupava 170 m², pesava 30 toneladas, funcionava com 18 mil válvulas e 10 mil capacitores, além de milhares de resistores a relé, consumindo uma potência de 150 Kwatts. Como tinha vários componentes discretos, não funcionava por muitos minutos seguidos sem que um deles quebrasse. Chega a ser, em algumas operações, mil vezes mais rápido que o MARK I.

A entrada de dados no ENIAC era baseada na tecnologia de cartões perfurados e os programas eram modificados através de reconfigurações no circuito. Apesar das dúvidas com relação a sua confiabilidade, o ENIAC permaneceu operacional por mais de 10 anos.

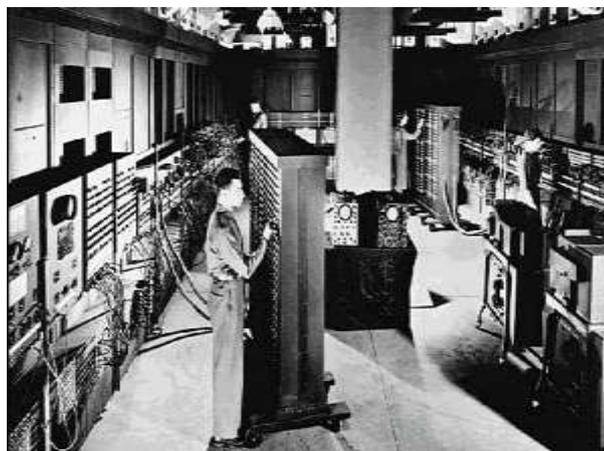


Figura 12: ENIAC.

Outra contribuição importante desta época foi o conceito de programa

armazenado, introduzida por John Von Neuman. Von Neuman tinha sido consultor no projeto ENIAC e conhecia os problemas da programação destas máquinas. Os programas para os computadores da época eram feitos através de modificações nos circuitos, o correspondia a um trabalho de dias para um programa relativamente simples. A proposta de Von Neuman foi inspirada na tecnologia de entrada de dados utilizada na época, fazendo com que os programas fossem introduzidos através de cartões perfurados como se fazia com os dados. John Von Neuman assim desenvolveu a lógica dos circuitos, os conceitos de programa e operações com números binários. Estes conceitos, adotados nos computadores atuais, revolucionou o conceito de programação de computadores da época, tornando muito mais flexíveis e versáteis.

Em 1947, Presper Eckert e John Mauchly, pioneiros na história da informática, fundaram a Cia. Eckert-Mauchly Computer Corporation, com o objetivo de fabricar máquinas baseadas em suas experiências como o ENIAC e o EDVAC.

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) e o UNIVAC (Universal Automatic Computer), que foi o primeiro computador a ser fabricado em linha. 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. O UNIVAC (figura 13) executava 1905 operações por segundo e seu preço chegou a US\$ 1 milhão.

Em 1953, A International Business Machines (IBM) lança seu primeiro computador digital, o IBM 701. Como primeiro computador da marca comercializado, foram vendidas 19 máquinas em três anos.

Em 1961 chegou o primeiro computador no Brasil: um UNIVAC 1105, ainda com válvulas, para o IBGE.



Figura 13 - UNIVAC

2.2.4 Computadores de segunda geração (1955-1965)

Com a invenção do transistor (fig. 14) em 1948, o mundo dos computadores é tomado de assalto por uma onda de novos projetos que dá origem, na década de 60 a empresas mundialmente conhecidas no que diz respeito à fabricação destas máquinas - DEC e IBM.

Com a segunda geração apareceram as memórias com anéis ferromagnéticos. As

fitas magnéticas foram a forma dominante de armazenamento secundário: permitiam capacidade muito maior de armazenamento e o ingresso mais rápido de dados que as fitas perfuradas.

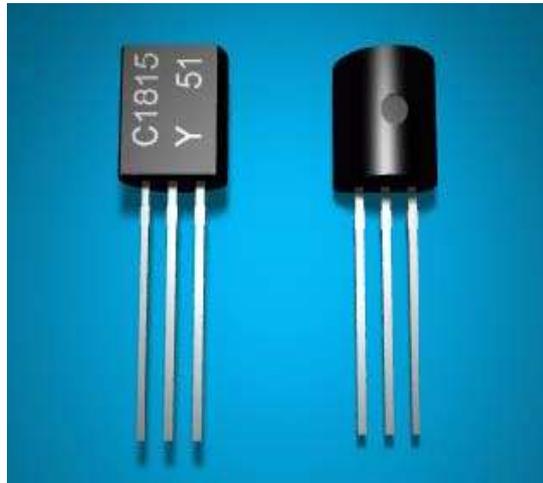


Figura 14: Transistor.

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.

Esses computadores, além de menores e mais baratos, consumiam menos energia, possuíam maior confiabilidade, eram mais rápidos (a velocidade passou para milionésimos de segundos) e eliminavam quase que por completo o problema do desprendimento de calor, característico da geração anterior.

Exemplos de computadores dessa geração são o IBM 1401 e o Honeywell 800. O IBM 1401 apareceu na década de 60 e com ele a IBM assumiu uma posição dominante na indústria de computadores.

A Digital Equipment Corporation (DEC) tinha então uma posição proeminente no setor com sua linha PDP. 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, que eram comercializados a um custo de três milhões de dólares. Já no início dos anos 60, a IBM passou a produzir os computadores da linha IBM 7040, que eram menos poderosos que seus predecessores, mas de custo bastante inferior.

2.2.5 Computadores de terceira geração (1965-1980)

Essa geração é marcada pela substituição dos transistores pela tecnologia dos circuitos integrados (transistores e outros componentes eletrônicos miniaturizados e montados numa única pastilha de silício - o chip). Em 1958, Jack Kilby criou o primeiro circuito integrado na Texas Instrument para provar que resistores e capacitores poderiam existir num mesmo pedaço de material semicondutor. Seu circuito era formado por uma lasca de germânio e cinco componentes ligados por fios. 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, que permitiu a substituição de dezenas de transistores numa única peça de silício, permitiu o surgimento de computadores de menores dimensões, mais rápidos e mais baratos. Com esses circuitos integrados o tempo passou a ser medido em nanossegundos (bilionésimos de segundos).

A tecnologia utilizada na época era a de pequena escala de integração (SSI - Small Scale of Integration) com a qual ao redor de mil transistores podiam-se integrar no circuito de uma pastilha. Com isso os computadores eram menores, mais confiáveis, com maior velocidade de operação e um custo bem mais baixo do que as máquinas das gerações anteriores. Também eram usados discos magnéticos para armazenamento, o que permitiu o acesso direto a arquivos grandes.

O exemplo típico dessa geração foi o IBM 360, série que introduziu o conceito de família de computadores compatíveis, facilitando a migração dos sistemas quando é necessário mudar para um computador mais potente. Esta estratégia permitiu que a IBM se posicionasse, já neste período, como líder do mercado de computadores. Essa família era composta por seis modelos básicos e várias opções de expansão que realizava mais de 2 milhões de adições por segundo e cerca de 500 mil multiplicações. Outra novidade introduzida por esta classe de computadores foi o conceito de multiprogramação, na qual diversos programas poderiam estar residentes na memória da máquina. No caso em que um programa entrasse em espera para uma operação de entrada/saída de dados, a unidade central passava a executar a parte de um outro programa.

Em 1965, a DEC introduziu o PDP-8, o primeiro minicomputador comercializado com sucesso. Ele era vendido a US\$ 18 mil.

Um outro computador desta geração que conheceu grande sucesso, particularmente nas universidades e centros de pesquisa foram os minicomputadores da série PDP-11 (DEC), apresentado na Figura 15.

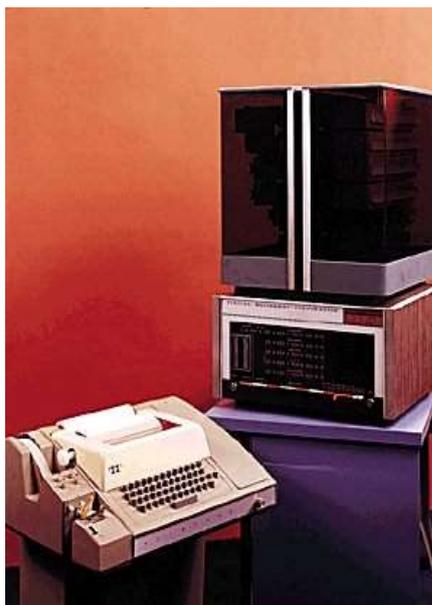


Figura 15: PDP 11.

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

Durante a década de 70, com a tecnologia da alta escala de integração (LSI - Large Scale of Integration) pode-se combinar até 65 mil componentes em uma só pastilha de silício (chip). Os anos 80, com o grande desenvolvimento da tecnologia de circuitos integrados, o número de transistores podendo ser integrados numa pastilha de silício atingiu a faixa dos milhares e, logo em seguida, dos milhões. Foi assim que surgiram os novos computadores, ainda menores, mais velozes e mais poderosos que aqueles da geração anterior. Na segunda metade da década de 90, houve a passagem da LSI para a VLSI (Very Large Scale of Integration - muito alta escala de integração). As máquinas de todas as gerações têm como característica comum a existência de uma única CPU para executar o processamento. Porém, mais recentemente, já existem computadores funcionando com mais de uma CPU. A velocidade dos computadores passou a atingir os trilionesimos de segundos, quase a velocidade da luz.

Os microcomputadores

Os pesquisadores da Xerox no centro de pesquisa em Palo Alto, projetaram o ALTO em 1974, a primeira estação de trabalho com uma entrada interna para mouse.

Desde o início da década de 80 os preços haviam caído de tal maneira que já começava a ser possível a uma pessoa ter o seu próprio computador -começava então a era da informática pessoal. Os computadores pessoais passaram então a ser utilizados de uma maneira relativamente distinta dos grandes computadores de então.

Esta geração caracteriza-se, portanto, pela crescente miniaturização e refinamento dos circuitos integrados. Nesta última geração surgiu o microcomputador, graças ao desenvolvimento de um novo circuito integrado: o microprocessador. O primeiro destes microprocessadores, o 8080, foi produzido pela Intel Corporation. A partir disto, o microcomputador teve uma grande evolução, gerando grandes transformações nos mais diversos ramos de atividades (das operações com cartão de crédito ao projeto de automóveis).

No início dessa geração nasceu a Intel, que começou a desenvolver o primeiro microprocessador, o Intel 4004 de 4 bits, um circuito integrado com 2250 transistores, equivalente ao ENIAC, apresentado na Figura 16.



Figura 16: Intel 4004.

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, ilustrado na Figura 17. Ele 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.



Figura 17: Altair.

Steve Wozniak e Steve Jobs formaram em 1976 uma pequena empresa, a Apple, onde construíram, numa garagem de fundo de quintal, o Apple I (Fig. 18). Um ano depois, com um novo e melhor projeto, surge o Apple II em 1977, primeiro microcomputador com grande sucesso comercial, pelas suas características: circuito impresso em sua placa-mãe, fonte de alimentação, teclado e cartuchos para jogos e, mais tarde, o Apple III. Em 1983 entra no mercado o Lisa e em 1984 o Macintosh, com tecnologia de 32 bits, o primeiro computador com mouse e interface gráfica, com valor comercial de US\$ 1,5 milhões de dólares. O diskette de 3 ½" foi amplamente aceito, ajudado pela decisão da Apple Computer de integrar seu uso no novo Macintosh.

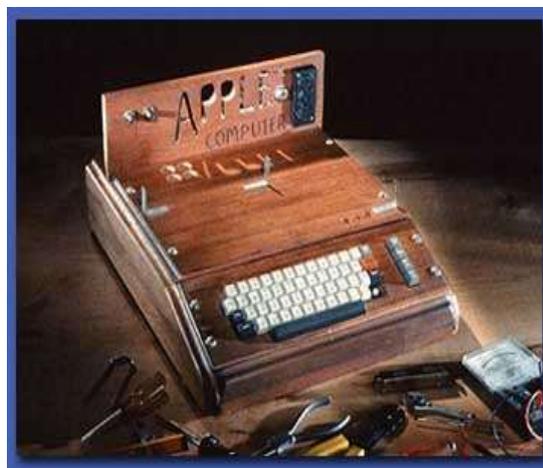


Figura 18: Apple I.

O computador pessoal Tandy Radio Shack's, o TRS-80, vendeu no primeiro ano (1977) 10 mil unidades, mais do que as 3 mil projetadas pela companhia.

Em 1980, a Seagate Technology desenvolveu o primeiro Hard Disk Drive para microcomputadores. O disco armazenou cinco megabytes de dados, cinco vezes mais que a maioria dos discos comuns da época. Desenvolvido pela Philips, o primeiro disco óptico de armazenamento de dados tinha uma capacidade de armazenamento 60 vezes maior do que um disco flexível de 5 ¼".

Em 1981, a IBM entrou no mercado de micros, introduzindo o PC (figura 19), um microcomputador com tecnologia de 16 bits (Intel 8088) que em pouco tempo se tornou um padrão. Os principais modelos de PC são:

- **PC:** possui cinco slots, dos quais dois são ocupados na configuração mínima - um para o controlador de disco flexível e o outro para a placa de vídeo e impressora, 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.
- **PX-XT:** possui oito slots, sendo dois ou três ocupados na configuração inicial - placa controladora de vídeo mais uma ou duas placas para controlar discos (flexível e disco rígido). A configuração típica de um XT é 512 a 768 K de memória RAM na placa principal, um drive de 360 K, um disco rígido de 10, 20 ou 30 Mb, placa controladora de vídeo gráfica, monitor monocromático e interface paralela ou serial. Seu clock era de 8, 10 até 12 MHz.
- **PC-XT 286:** modelo intermediário entre o PC-XT e o PC-AT ou, como era chamado, um AT simplificado, uma vez que usa o microprocessador do AT o Intel 80286. Este era três vezes mais rápido que o XT e podia executar várias tarefas ao mesmo tempo. É um PC-XT com o processador 80286.
- **PC-AT:** usa o microprocessador da Intel 80286 de 32 bits e possui maior capacidade de processamento, com memória principal de até 4 Mbytes. Sua configuração inicial típica é: 1 Mbyte de RAM, um drive de 5,25 polegadas de alta capacidade, disco rígido de 20 ou 30 Mbytes com alta velocidade de acesso, interface paralela e serial RS-232, controlador de vídeo e monitor monocromático. Sua velocidade de processamento alcançava entre 16 e 20 Mhz. A grande importância do AT está na maior capacidade do 80286, que resulta em um desempenho duas a três vezes maior que os XT.
- **PC-386:** É um PC-AT com o microprocessador da Intel, o 80386. Com isso adquiriram grande velocidade de processamento e era capaz da multitarefa em 32 bits. O processador 80386 foi o grande marco da popularização da computação pessoal.
- **PC 486:** utiliza o microprocessador Intel 80486, com um co-processador aritmético embutido e mais de 1,2 milhão de transistores encolhidos em um chip.



Figura 19 – Computador Pessoal IBM (PC)

O Pentium

Em 1993 chegou ao mercado o processador Pentium (figura 20). Os microprocessadores Pentium contêm mais de três milhões de transistores e já incluem co-processador matemático e memória cache. Opera com 5 volts, e apresenta muito aquecimento, mas com melhorias no projeto, a Intel permitiu a operação com 3,5 volts, resultando num aquecimento bem menor. Novas versões foram lançadas como a de 75, 90, 100, 120, 133, 155, 166 e 200 MHz. O Pentium é um microprocessador de 32 bits, mas com várias características de 64 bits.



Figura 20: Processador Pentium

O Pentium Pro foi criado para ser o sucessor do Pentium. Inicialmente foi lançado nas versões 150, 180 e 200 MHz. Opera com 32 bits e utiliza memória de 64 bits, da mesma forma como ocorre com o Pentium. Seu projeto foi otimizado para realizar processamento de 32 bits, sendo neste tipo de aplicação mais rápido que o Pentium comum, só que ao realizar processamento de 16 bits perde para o Pentium comum.

Pentium MMX (P55c) foi desenvolvido visando aumentar o desempenho de programas que fazem processamento de gráficos, imagens e sons, a Intel adicionou ao microprocessador Pentium, 57 novas instruções específicas para a execução rápida deste tipo de processamento, elas são chamadas de instruções MMX (MMX = Multimedia Extensions). Uma única instrução MMX realiza o processamento equivalente ao de várias instruções comuns. Essas instruções realizam cálculos que aparecem nos processamentos de sons e imagens. O Pentium MMX possui uma memória cache interna de 32 KB e trabalha com níveis duplos de voltagem: externamente a 3,3 volts enquanto o núcleo do processador opera a 2,8 volts. A

conexão é feita através do Soquete 7, ou seja, possui o mesmo conjunto de sinais digitais que o Pentium comum. Teve as versões: 166 MMX, 200 MMX, 233 MMX MHz.

Pentium II (i440Bx) foi o sucessor do Pentium MMX, com velocidades de 300, 333, 350, 400 MHz. Possui barramento de 100 MHz, e é encapsulado em um envólucro (cartucho) que engloba o processador e a memória cache externa (L2), este envólucro metálico facilita a dissipação do calor.

Pentium III (440Bx) foi projetado para a Internet, o processador Pentium III vem com clock de 450 e 500 MHz, e com 70 novas instruções que habilita aplicativos de processamento avançados de imagens, 3D, áudio e vídeo, e reconhecimento de voz. Seu barramento é de 100 MHz, com memória cache secundária de 512 KB, e possui cerca de nove milhões de transistores.

Um Pentium 4 (Prescott) tem 125 milhões de transistores e chegou aos 3.8 GHz. Em 2003 a Intel lançou o Pentium M, um chip derivado da antiga arquitetura do Pentium III, que consome pouca energia, esquenta pouco e mesmo assim oferece um excelente desempenho. Um Pentium M de 1.4 GHz chega a superar um Pentium 4 de 2.6 GHz em diversas aplicações. O Pentium M foi desenvolvido originalmente para ser usado em notebooks, mas se mostrou tão eficiente que acabou sendo usado como base para o desenvolvimento da plataforma Core, usada nos processadores Core 2 Duo fabricados atualmente pela Intel. Quando as possibilidades de aumento de clock do Pentium 4 se esgotaram, a Intel lançou o Pentium D (figura 21a), uma versão dual-core do Pentium 4. O Pentium 4 acabou se revelando um beco sem saída, descontinuado e condenado ao esquecimento.

Ainda temos o Celeron, uma versão mais barata, mas com um desempenho um pouco inferior, por ter menos cache ou outras limitações. O Celeron não é uma família separada de chips, mas apenas um nome comercial usado nas versões mais baratas de vários processadores Intel. Existem Celerons baseados nas várias versões de Pentium e também o Celeron 4xx, que é uma versão single-core (e com menos cache) do Core 2 Duo.

O Pentium trouxe um novo fôlego às chamadas estações de trabalho (microcomputadores poderosos usados em tarefas pesadas, como computação gráfica e aplicações científicas). Uma das novidades dele é que possibilita a simulação de dois processadores, ou seja, um princípio de paralelização antes possível apenas em supercomputadores e que agora esta ao alcance dos usuários de microcomputadores.



Figura 21 – Processadores Athlon X2 (a) e Pentium D (b)

O Athlon

O Athlon, que foi o primeiro grande processador (tanto em desempenho, quanto em tamanho) da AMD. A primeira versão usava um formato de cartucho (slot A) similar ao Pentium II, mas incompatível com as placas para ele. Ele foi sucedido pelo Athlon Thunderbird, que passou a usar o formato de soquete utilizado (com atualizações) até os dias de hoje.

Por volta de 2000, surgiram as primeiras notícias do processador de 64 bits, que foi finalmente lançado em versão doméstica na forma do Athlon 64, que passou a ser o topo de linha da AMD. Apesar das mudanças internas, o Athlon 64 continua sendo compatível com os programas de 32 bits, da mesma forma que os processadores atuais são capazes de rodar softwares da época do 386, muito embora tenham incorporado diversos novos recursos.

O fato de ser um processador de 64 bits não torna o Athlon 64 gritantemente mais rápido, mesmo em aplicativos otimizados. A principal vantagem dos processadores de 64 bits é derrubar uma limitação inerente a todos os processadores de 32 bits, que são capazes de acessar apenas 4 GB de memória RAM, um limite que está se tornando cada vez mais uma limitação grave em várias áreas. Os 4 GB de memória podem não parecer um obstáculo imediato, mas lembre-se de que há duas décadas os PCs eram vendidos com 128 KB de memória, há uma década já vinham com 4 ou 8 MB, e hoje são vendidos com 512 MB ou mais.

O Athlon 64 deu origem ao Athlon X2 (figura 21b), o primeiro processador dual-core da AMD, onde temos dois processadores Athlon 64 no mesmo encapsulamento, dividindo a carga de processamento e também o Turion, que é uma versão de baixo custo do Athlon 64, destinado a notebooks.

Supercomputadores

A história dos supercomputadores começa, de fato, no final de 1975 como Cray-1 (figura 22). 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. Como exemplos de supercomputadores podemos citar: Cray-1, Cyber 205, Fujitsu FACON-APU, Hitachi M200HIAP, Galaxy, Cray-2, Cray-3.



Figura 22: Cray-1.

2.2.7 Quinta Geração?

Alguns consideram, ainda, uma quinta geração de computadores. Nesta geração, os computadores utilizariam o ainda em estudo, Biochips. Os computadores serão então capazes de entender a linguagem natural do homem, e a inteligência Artificial será a fonte de diversos avanços.

2.2.8 O software

Como pode ser observado, o hardware foi o que sofreu as maiores transformações. Para avaliar esta evolução, pode-se ter como base o primeiro computador eletrônico ENIAC, apresentado em 1946. Este computador ocupava uma área superior a 170 m², pesava 30 toneladas, utilizando 18.000 válvulas e 10.000 capacitores, consumindo 150.000 watts de energia e custando vários milhões de dólares. Além de todo este tamanho, sua preparação demorava semanas, pois a programação era realizada pela ligação de fios. Atualmente, os microcomputadores pesam alguns quilos e possuem capacidade infinitamente superior ao ENIAC. Mas o software também evoluiu, como pode é descrito abaixo:

- 1957: Primeiro compilador Fortran, que permitia ao computador executar tarefas repetidas a partir de um conjunto de instruções;
- 1959: Comitê Codasyl é formado para criar o Cobol (Common Business Oriented Language), a primeira linguagem voltada para o uso em programação. A linguagem Lisp é criada para aplicações de inteligência artificial;
- 1960: É desenvolvido o padrão Algol 60;
- 1962: Os estudantes do MIT Slug Russel, Shag Graetz e Alan Kotok escreveram o SpaceWar!, considerado o primeiro jogo interativo de computador. O jogo oferecia gráficos interativos que inspiraram os vídeo games futuros;

- 1963: ASCII - American Standard Code for Information Interchange - permitiu que máquinas de diferentes fabricantes trocassem dados entre si;
- 1964: Linguagem Basic (Begginer's All-purpose Symbolic Instruction Codes) é criada por Thomas Kurtz e John Kemeny, professores do DartMouth College;
- 1967: Criação da linguagem Simula, a primeira linguagem orientada a objetos. Seymour Papert desenhou o LOGO como se fosse uma linguagem de computação para crianças;
- 1969: Os programadores dos laboratórios AT&T Bell, Ken Thompson e Denis Richie desenvolvem o UNIX, primeiro sistema operacional que poderia ser aplicado em qualquer máquina. O exército americano interligou as máquinas da Arpanet, formando a rede que originaria a Internet.
- 1970: A primeira comunicação computer-to-computer expandiu-se quando o Departamento de Defesa Americano estabeleceu quatro pontos para comunicação no ARPANET: Universidade de California-Santa Barbara, UCLA, SRI internacional, e Universidade de Utah;
- 1971: A linguagem Pascal é apresentada;
- 1972: A linguagem C é criada no Bell Labs; A Linguagem Prolog é criada na Universidade de Marseille;
- 1975: A Telenet, a primeira rede comercial, equivalente a ARPANET, foi implantada;
- 1976: Gary Kildall desenvolveu o sistema operacional CP/M, para computadores pessoais;
- 1979: Os programadores Daniel Bricklin e Robert Frankston, da Universidade Harvard, desenvolveram o VisiCalc, programa que tornou os computadores comerciais em computadores pessoais;
- 1980: A linguagem Ada é criada;
- 1981: O sistema operacional MS-DOS (Microsoft Disk Operating System) foi o software básico lançado para o PC da IBM, estabelecendo uma longa parceria entre a IBM e a Microsoft;
- 1982: Mitch Kapor desenvolveu o Lotus 1-2-3, software direcionado para o IBM PC. O uso de gráficos gerados em computadores para filmes deu grande passo através da realização do filme "Tron", lançado pela Disney;
- 1983: A Linguagem C++ é desenvolvida; A Microsoft anunciou o processador de textos Word, chamado anteriormente de Multi-Tool Word e também anunciou o lançamento do S.O. Windows. O MIDI (Musical Instrument Digital Interface) foi apresentado na primeira mostra North American Music Manufactures, em Los Angeles.

- 1984: Apple lança o Sistema Operacional Macintosh System 1, que utiliza janelas;
- 1985: A Internet ganhou suporte quando o National Science Foundation estruturou o NSFNET ligando cinco supercomputadores nas Universidades de Princeton, Pittsburgh, Califórnia, Illinois e Cornell. Iniciam a distribuição do S.O. Macintosh System e do S.O. Windows 1.0;
- 1987: William Atkinson, engenheiro da Apple, projetou HyperCard, um software que simplificava o desenvolvimento de aplicações domésticas. São lançados os S.O. Macintosh System 5 e Windows 2.0;
- 1988: Pixar's "Tin Toy" tornou-se o primeiro filme feito em computador a ganhar um prêmio da Academia, o Oscar, por melhor desenho animado de curta metragem. Robert Morris enviou um vírus através da Internet, que causou problemas para aproximadamente 10% do 60 mil usuários da rede;
- 1989: A Maxis lançou o SimCity, um jogo de video game que utilizava uma série de simuladores. A cidade de Sim é usada frequentemente em ambientes educacionais. O conceito da realidade virtual foi o tópico principal na convenção de Siggraph's, realizada em Boston, Massachussets;
- 1990: A Microsoft anunciou o Windows 3.0, compatível com o DOS, a primeira versão do Windows oferecia satisfação e performance aos usuários de PC. A World Wide Web nasceu quando Tim Berners-Lee, um pesquisador do CERN, desenvolveu o HTML (HiperText Markup Language);
- 1992: Após muita expectativa, o vírus Michelangelo resulta em um estrago de pequenas proporções. É disponibilizado o Macintosh System 7.5 e o Windows 3.2;
- 1993: A equipe do National Center for Supercomputing Applications da Universidade de Illinois desenvolvem uma interface gráfica para navegação pela Internet chamada NCSA Mosaic. A Microsoft disponibiliza o sistema operacional Windows 3.11, que é, na verdade, uma interface gráfica de usuário (GUI) aplicada sobre o real controlador: o MS-DOS.
- 1994: O primeiro browser da Netscape torna-se disponível. Organizados por David Filo e Jerry Yang, nasce o "Jerry and David's Guide to the World Wide Web", que, ainda naquele ano passou a se chamar "Yet Another Hierarchical Officious Oracle" ou, simplesmente, Yahoo!.
- 1995: A linguagem Java é lançada pela Sun. Criado pelos sócios brasileiros Gustavo Viberti e Fábio Oliveira, o Cadê? o primeiro buscador brasileiro de sucesso.
- 1997: É lançado o sistema operacional MAC OS 8. Larry Page e Sergey Brin registram o domínio de Internet **www.google.com** e apresentam sua

tecnologia ao Escritório de Licenciamento Tecnológico da Universidade Stanford. O Yahoo lança um serviço gratuito de e-mails públicos;

- 1998: É lançado o sistema operacional Windows 98. A empresa Google é formalmente criada na Califórnia (EUA);
- 1999: O sistema operacional Linux é lançado. O Yahoo lança a versão brasileira do seu portal;
- 2001: Lançado os sistemas operacionais Windows XP e Mac OS X 10.1.
- 2007: Lançamento do iPhone OS (hoje iOS), sistema operacional móvel da Apple.
- 2008: Lançamento da primeira versão comercial do sistema operacional Androide pela Google

3 SISTEMAS DE NUMERAÇÃO

Os sistemas de numeração tem por objetivo prover símbolos e convenções para representar quantidades, de forma a registrar a informação quantitativa e poder processá-la. A representação de quantidades se faz com os **números**. Na antiguidade, duas formas de representar quantidades foram inventadas. Inicialmente, os egípcios, criaram um sistema em que cada dezena (uma mão-cheia) era representada por um símbolo diferente. Usando por exemplo os símbolos # para representar uma centena, & para representar uma dezena e @ representando uma unidade (símbolos escolhidos ao acaso), teríamos que ###&&@ representaria 321.

Sistema, o sistema de numeração romano. Eram usados símbolos (letras) que representavam as quantidades, como por exemplo: I (valendo 1), V (valendo 5), X (valendo 10), C (valendo 100), etc. A regra de posicionamento determinava que as letras que representavam quantidades menores e precediam as que representavam quantidades maiores, seriam somadas; se o inverso ocorresse, o menor valor era subtraído do maior (e não somado). Assim, a quantidade 128 era representada por CXXVIII = $100 + 10 + 10 + 5 + 1 + 1 + 1 = 128$. Por outro lado, a quantidade 94 era representada por XCIV = $(-10 + 100) + (-1 + 5) = 94$.

Nesses sistemas, os símbolos tinham um valor intrínseco, independente da posição que ocupavam na representação (sistema numérico não-posicional). Um grande problema desse sistema é a dificuldade de realizar operações com essa representação. Experimente multiplicar CXXVIII por XCIV! Assim, posteriormente foram criados sistemas em que a posição dos algarismos no número passou a alterar seu valor (sistemas de numeração posicionais).

Nos sistemas de numeração posicionais (ver adiante **Sistemas de Numeração Posicionais**), o valor representado pelo algarismo no número depende da posição em que ele aparece na representação. O primeiro sistema desse tipo foi inventado pelos chineses. Eram usados palitos, sendo 1 a 5 palitos dispostos na vertical para representar os números 1 a 5; de 6 a 9 eram representados por 1 a 4 palitos na vertical, mais um palito na horizontal (valendo 5) sobre os demais. Cada número era então representado por uma pilha de palitos, sendo uma pilha de palitos para as unidades, outra para as dezenas, outra para as centenas, etc. Esse sistema, com as pilhas de palitos dispostas em um tabuleiro, permitia a realização das quatro operações aritméticas. Não existia representação para o zero (o espaço relativo ficava vazio). O tabuleiro aritmético (chamado *swan-pan*), além das quatro operações, era usado na álgebra e na solução de equações. Essa técnica era chamada de *Método do Elemento Celestial*.

O Alfabeto e o Ábaco No Oriente Médio, por esses tempos, criou-se uma das mais importantes invenções da humanidade: o **alfabeto**. Na antiguidade, usava-se um símbolo para representar cada conceito ou palavra. Assim, eram necessários milhares de símbolos para representar todos os objetos, ações, sentimentos, etc - como são ainda hoje algumas linguagens. Como decorar todos? O grande achado foi decompor a linguagem em alguns poucos símbolos e regras básicas. Uma consequência de fundamental importância para nossos estudos de informática foi possibilitar a **ordenação alfabética** (essa é uma tarefa típica dos computadores).

Nessa época, foi também criado o **ábaco** - uma calculadora decimal manual.

Os Algarismos e o Zero Por volta do ano de 650, os hindus inventaram um método de produzir papel (que antes já fora inventado pelos chineses) e seus matemáticos criaram uma representação para os números em que existiam diferentes símbolos para os as unidades, incluindo um símbolo para representar o zero. Essa simples criação permitiu que se processasse a aritmética decimal e se fizesse contas - no papel! Bom, depois de milhares de anos em que todos os cálculos eram feitos com calculadoras (ábacos, swan-pan, etc) finalmente era possível calcular sem auxílio mecânico, usando um instrumento de escrita e papel. A matemática criada pelos hindus foi aprendida pelos árabes (que depois foram copiados pelos europeus). Por volta de 830, um matemático persa (chamado Al-khwarismi, que inspirou o nome algarismo) escreveu um livro (Al-gebr we'l Mukabala, ou álgebra) em que apresentava os algarismos hindus. E esse livro, levado para a Europa e traduzido, foi a base da matemática do Renascimento.

3.1 Sistemas de Numeração Posicionais

Desde quando se começou a registrar informações sobre quantidades, foram criados diversos métodos de representar as quantidades.

O método ao qual estamos acostumados usa um sistema de numeração posicional. Isso significa que a posição ocupada por cada algarismo em um número altera seu valor de uma potência de 10 (na base 10) para cada casa à esquerda.

Por exemplo, no sistema decimal (base 10), no número 125 o algarismo 1 representa 100 (uma centena ou 10^2), o algarismo 2 representa 20 (duas dezenas ou 2×10^1) e o algarismo 5 representa 5 mesmo (5 unidades ou 5×10^0). Assim, em nossa notação,

$$125 = 1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$

Base de um Sistema de Numeração A base de um sistema é a quantidade de algarismos disponível na representação. A base 10 é hoje a mais usualmente empregada, embora não seja a única utilizada. No comércio pedimos uma dúzia de rosas ou uma grossa de parafusos (base 12) e também marcamos o tempo em minutos e segundos (base 60).

Os computadores utilizam a base 2 (sistema binário) e os programadores, por facilidade, usam em geral uma base que seja uma potência de 2, tal como 2^4 (base 16 ou sistema hexadecimal) ou eventualmente ainda 2^3 (base 8 ou sistema octal).

Na base 10, dispomos de 10 algarismos para a representação do número: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9. Na base 2, seriam apenas 2 algarismos: 0 e 1. Na base 16, seriam 16: os 10 algarismos aos quais estamos acostumados, mais os símbolos A, B, C, D, E e F, representando respectivamente 10, 11, 12, 13, 14 e 15 unidades. Generalizando, temos que uma base b qualquer disporá de b algarismos, variando entre 0 e (b-1).

A representação 125,3810 (base 10) significa $1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2}$:

Generalizando, representamos uma quantidade N qualquer, numa dada base b, com um número tal como segue:

$$N^b = a_{n-1}.b^{n-1} + \dots + a_2.b^2 + a_1.b^1 + a_0.b^0 + a_{-1}.b^{-1} + a_{-2}.b^{-2} + \dots + a_{-n}.b^{-n} \text{ sendo que}$$
$$a_{n-1}.b^{n-1} + \dots + a_2.b^2 + a_1.b^1 + a_0.b^0 \text{ é a parte inteira e}$$

$a_{-1}.b^{-1} + a_{-2}.b^{-2} + \dots + a_{-n}.b^{-n}$ é a parte fracionária.

Intuitivamente, sabemos que o maior número que podemos representar, com n algarismos na base b , será o número composto n vezes pelo maior algarismo disponível naquela base (ou seja, $b-1$). Por exemplo, o maior número que pode ser representado na base 10 usando 3 algarismos será 999 (ou seja, $10^3 - 1 = 999$).

Generalizando, podemos ver que o maior número inteiro N que pode ser representado, em uma dada base b , com n algarismos (n "casas"), será $N = b^n - 1$. Assim, o maior número de 2 algarismos na base 16 será FF_{16} que, na base 10, equivale a $255_{10} = 16^2 - 1$.

Representação Binária Os computadores modernos utilizam apenas o sistema binário, isto é, todas as informações armazenadas ou processadas no computador usam apenas DUAS grandezas, representadas pelos algarismos 0 e 1. Essa decisão de projeto deve-se à maior facilidade de representação interna no computador, que é obtida através de dois diferentes níveis de tensão. Havendo apenas dois algarismos, portanto dígitos binários, o elemento mínimo de informação nos computadores foi apelidado de bit (uma contração do inglês *binary digit*).

Na base 2, o número "10" vale dois. Mas se $10_2 = 2_{10}$, então dez é igual a dois?

Não, dez não é e nunca será igual a dois!

Na realidade, "10" não significa necessariamente "dez". Nós estamos acostumados a associar "10" a "dez" porque estamos acostumados a usar o sistema de numeração decimal. O número 10_2 seria lido "um-zero" na base 2 e vale 2_{10} (convertido para "dois" na base dez), 10_5 seria lido "um-zero" na base 5 e vale 5_{10} (convertido para "cinco" na base dez), 10_{10} pode ser lido como "um-zero" na base 10 ou então como "dez" na base dez, 10_{16} seria lido "um-zero" na base 16 e vale 16_{10} (convertido para "dezesesseis" na base dez), etc.

Portanto, 10 só será igual a dez se - e somente se - o número estiver representado na base dez!

Uma curiosidade: o número " 10_b " vale sempre igual à base, porque em uma dada base b os algarismos possíveis vão sempre de 0 a $(b - 1)$! Como o maior algarismo possível em uma dada base b é igual a $(b-1)$, o próximo número será $(b - 1 + 1 = b)$ e, portanto, será sempre 10 e assim, numa dada base qualquer, o valor da base será sempre representado por "10"!

Obs: Toda vez que um número for apresentado sem que seja indicado em qual sistema de numeração ele está representado, estenderemos que a base é dez. Sempre que outra base for utilizada, a base será obrigatoriamente indicada.

Um dia pode ser que os computadores se tornem obrigatórios e sejamos todos forçados por lei a estudar a aritmética em binário! Mas, mesmo antes disso, quem programa computadores precisa conhecer a representação em binário! Vamos começar entendendo as potências de dois:

| Repr.Binária | Potência | Repr.Decimal |
|--------------|----------|--------------|
| 1 | 2^0 | 1 |
| 10 | 2^1 | 2 |
| 100 | 2^2 | 4 |
| 1000 | 2^3 | 8 |
| 10000 | 2^4 | 16 |
| 100000 | 2^5 | 32 |
| 1000000 | 2^6 | 64 |
| 10000000 | 2^7 | 128 |
| 100000000 | 2^8 | 256 |
| 1000000000 | 2^9 | 512 |
| 10000000000 | 2^{10} | 1.024 |

Depois (e só depois) de compreender bem a tabela acima, fazendo a devida correlação com a representação decimal, é conveniente decorar os valores da tabela. As conversões entre base dois e base dez e as potências de dois são utilizadas a todo momento e seria perda de tempo estar toda hora convertendo. Da mesma forma que, uma vez entendido o mecanismo da multiplicação, decoramos a tabuada, é muito mais efetivo saber de cor a tabela acima que fazer as contas de conversão toda vez que for necessário.

A representação binária é perfeitamente adequada para utilização pelos computadores. No entanto, um número representado em binário apresenta muitos bits, ficando longo e passível de erros quando manipulado por seres humanos normais como, por exemplo, os programadores, analistas e engenheiros de sistemas. Para facilitar a visualização e manipulação por programadores de grandezas processadas em computadores, são usualmente adotadas as representações octal (base 8) e principalmente hexadecimal (base 16). Ressaltamos mais uma vez que o computador opera apenas na base 2 e as representações octal e hexadecimal não são usadas no computador, elas se destinam apenas à manipulação de grandezas pelos programadores.

Representação em Octal e em Hexadecimal Em projetos de informática (isto é, nos trabalhos realizados pelos programadores, analistas e engenheiros de sistemas), é usual representar quantidades usando sistemas em potências do binário (octal e principalmente hexadecimal), para reduzir o número de algarismos da representação e conseqüentemente facilitar a compreensão da grandeza e evitar erros. No sistema octal (base 8), cada tres bits são representados por apenas um algarismo octal (de 0 a 7). No sistema hexadecimal (base 16), cada quatro bits são representados por apenas um algarismo hexadecimal (de 0 a F).

A seguir, apresentamos uma tabela com os números em decimal e sua representação correspondente em binário, octal e hexadecimal:

| Base 10 | Base 2 | Base 8 | Base 16 |
|---------|--------|--------|---------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |

Nota: a base 16 ou sistema hexadecimal pode ser indicada também por um "H" ou "h" após o número; por exemplo: FFH significa que o número FF (ou 255 em decimal) está em hexadecimal. Não confundir o "H" ou "h" com mais um dígito, mesmo porque em hexadecimal só temos algarismos até "F" e portanto não existe um algarismo "H".

Exemplo: Como seria a representação do número 16_{10} em binário, octal e hexadecimal? Solução: Seria respectivamente 10000_2 , 20_8 e 10_{16} .

3.2 Conversões entre Bases

Vamos analisar agora as regras gerais para converter números entre duas bases quaisquer.

3.2.1 Conversão entre bases 2, 8 e 16

As conversões mais simples são as que envolvem bases que são potências entre si. Vamos exemplificar com a conversão entre a base 2 e a base 8. Como $2^3 = 8$, separando os bits de um número binário em grupos de tres bits (começando sempre da direita para a esquerda) e convertendo cada grupo de tres bits para seu equivalente em octal, teremos a representação do número em octal. Por exemplo:

$10101001_2 = 10.101.001_2$ (separando em grupos de 3, sempre começando da direita para a esquerda) Sabemos que $010_2 = 2_8$; $101_2 = 5_8$; $001_2 = 1_8$ portanto $10101001_2 = 251_8$

Se você ainda não sabe de cor, faça a conversão utilizando a regra geral. Vamos agora exemplificar com uma conversão entre as bases 2 e 16. Como $2^4 = 16$,

basta separarmos em grupos de 4 bits (começando sempre da direita para a esquerda) e converter. Por exemplo:

$11010101101_2 = 110.1010.1101_2$ (separando em grupos de 4 bits, sempre começando da direita para a esquerda) Sabemos que $110_2 = 6_{16}$; $1010_2 = A_{16}$; $1101_2 = D_{16}$; portanto $11010101101_2 = 6AD_{16}$

Vamos agora exercitar a conversão inversa. Quanto seria 3F5h (lembrar que o H está designando "hexadecimal") em octal? O método mais prático seria converter para binário e em seguida para octal. Por exemplo:

$3F5_h = 11.1111.0101_2$ (convertendo cada dígito hexadecimal em 4 dígitos binários) = $1.111.110.101_2$ (agrupando de três em três bits) = 1765_8 (convertendo cada grupo de três bits para seu valor equivalente em octal).

3.2.2 Conversão de Números em uma base b qualquer para a base 10

Vamos lembrar a expressão geral já apresentada:

$$N_b = a_{n-1} \cdot b^{n-1} + \dots + a_2 \cdot b^2 + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-n} \cdot b^{-n}$$

A melhor forma de fazer a conversão é usando essa expressão. Tomando como exemplo o número 101101_2 , vamos calcular seu valor representado na base dez. Usando a expressão acima, fazemos:

$$101101_2 = 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}$$

Podemos fazer a conversão de números em qualquer base para a base 10 usando o algoritmo acima. Exemplos:

a) Converter 4F5H para a base 10. Solução:

Lembramos que o H significa que a representação é hexadecimal (base 16). Sabemos ainda que $F_{16}=15_{10}$. Então:

$$4x16^2 + 15x16^1 + 5x16^0 = 4x256 + 15x16 + 5 = 1024 + 240 + 5 = 1269_{10}$$

b) Converter 3485_9 para a base 10. Solução:

$$3x9^3 + 4x9^2 + 8x9^1 + 5x9^0 = 3x729 + 4x81 + 8x9 + 5 = 2187 + 324 + 72 + 5 = 2588_{10}$$

c) Converter $7G_{16}$ para a base 10. Solução: Uma base **b** dispõe dos algarismos entre 0 e (b-1). Assim, a base 16 dispõe dos algarismos 0 a F e portanto o símbolo G não pertence à representação hexadecimal.

d) Converter $1001,01_2$ para a base 10. Solução:

$$1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} = 8 + 0 + 0 + 1 + 0 + 0,25 = 9,25_{10}$$

e) Converter $34,3_5$ para a base 10. Solução:

$$3x5^1 + 4x5^0 + 3x5^{-1} = 15 + 4 + 0,6 = 19,6_{10}$$

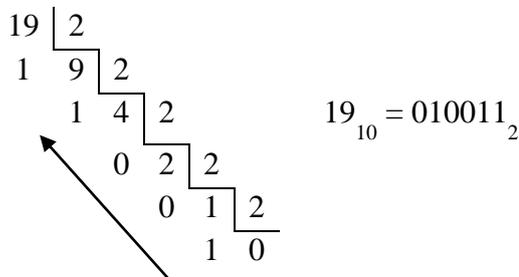
f) Converter $38,3_8$ para a base 10. Solução:

Uma base **b** dispõe dos algarismos entre 0 e (b-1). Assim, a base 8 dispõe dos algarismos 0 a 7 e portanto o algarismo 8 não existe nessa base. A representação 38,3 não existe na base 8.

3.2.3 Conversão de Números da Base 10 para uma Base b qualquer

A conversão de números da base dez para uma base qualquer emprega algoritmos que serão o inverso dos acima apresentados. O algoritmo será entendido melhor pelo exemplo que por uma descrição formal. Vamos a seguir apresentar os algoritmos para a parte inteira e para a parte fracionária:

Parte Inteira: O número decimal será dividido sucessivas vezes pela base; o resto de cada divisão ocupará sucessivamente as posições de ordem 0, 1, 2 e assim por diante até que o resto da última divisão (que resulta em quociente zero) ocupe a posição de mais alta ordem. Veja o exemplo da conversão do número 19_{10} para a base 2:



Experimente fazer a conversão contrária (retornar para a base 10) e ver se o resultado está correto.

Parte Fracionária Se o número for fracionário, a conversão se fará em duas etapas distintas: primeiro a parte inteira e depois a parte fracionária. Os algoritmos de conversão são diferentes. O algoritmo para a parte fracionária consiste de uma série de multiplicações sucessivas do número fracionário a ser convertido pela base; a parte inteira do resultado da primeira multiplicação será o valor da primeira casa fracionária e a parte fracionária será de novo multiplicada pela base; e assim por diante, até o resultado dar zero ou até encontrarmos o número de casas decimais desejado. Por exemplo, vamos converter $15,65_{10}$ para a base 2, com 5 e com 10 algarismos fracionários:

| | |
|--|---|
| $ \begin{array}{r} 15 \quad \quad 2 \\ 1 \quad \quad 7 \quad \quad 2 \\ \quad \quad \quad 1 \quad \quad 3 \quad \quad 2 \\ \quad \quad \quad \quad 1 \quad \quad 1 \quad \quad 2 \\ \quad \quad \quad \quad \quad 1 \quad \quad 0 \end{array} $ | $15_{10} = 01111_2$ |
| $ \begin{array}{l} 0,65 \times 2 = 1,30 \\ 0,30 \times 2 = 0,60 \\ 0,60 \times 2 = 1,20 \\ 0,20 \times 2 = 0,40 \\ 0,40 \times 2 = 0,80 \\ 0,80 \times 2 = 1,60 \\ 0,60 \times 2 = 1,20 \\ 0,20 \times 2 = 0,40 \\ 0,40 \times 2 = 0,80 \\ 0,80 \times 2 = 1,60 \end{array} $ | $0,65_{10} = 10100_2$ $0,65_{10} = 1010011001_2$ $15,65_{10} = 1111,1010011001_2$ |

Obs.: Em ambos os casos, a conversão foi interrompida quando encontramos o número de algarismos fracionários solicitados no enunciado. No entanto, como não encontramos resultado 0 em nenhuma das multiplicações, poderíamos continuar efetuando multiplicações indefinidamente até encontrar (se encontrarmos) resultado zero. No caso de interrupção por chegarmos ao número de dígitos especificado sem encontramos resultado zero, o resultado encontrado é aproximado e essa aproximação será função do número de algarismos que calcularmos. Fazendo a conversão inversa, encontraremos:

a) Com 5 algarismos fracionários:

Parte inteira: $1111_2 = 15_{10}$

Parte fracionária: $0,10100_2 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5} = 0,5 + 0,125 = 0,625_{10}$

b) Com 10 algarismos fracionários:

Parte inteira: $1111_2 = 15_{10}$

Parte fracionária: $0,1010011001_2 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6} + 1 \times 2^{-7} + 0 \times 2^{-8} + 0 \times 2^{-9} + 1 \times 2^{-10} = 1/2 + 1/8 + 1/64 + 1/128 + 1/1024 = 0,5 + 0,125 + 0,015625 + 0,0078125 + 0,0009765625 = 0,6494140625_{10}$

Ou seja, podemos verificar (sem nenhuma surpresa) que, quanto maior for a quantidade de algarismos considerados, melhor será a aproximação.

Resumindo

Quanto a conversão de valores com parte inteira e parte fracionária, deve-se

realizar as operações de forma separada. A conversão da parte inteira é obtida dividindo-se o número decimal pelo valor da base desejada; o resto encontrado é o algarismo menos significativo do valor na base B (mais à direita). Em seguida, divide-se o quociente encontrado pela base B; o resto é o algarismo significativo seguinte (à esquerda); e assim sucessivamente, vão-se dividindo os quocientes pelo valor da base até se obter quociente de valor zero. Em cada divisão, o resto encontrado é um algarismo significativo do número na nova base; o primeiro resto encontrado é o valor do algarismo menos significativo (mais à direita), e o último resto será o algarismo mais significativo (mais à esquerda).

Para a parte fracionária do número a ser convertido, seleciona a parte fracionária e multiplica pela base a ser convertida (B), e o valor encontrado é constituído de duas partes (a parte inteira, mesmo que seja zero e a parte fracionária). A parte inteira é o primeiro algarismo procurado (o primeiro algarismo a direita da virgula) e a parte fracionária é novamente multiplicada pela base e obtém-se um novo resultado (em duas partes distintas, a inteira e a fracionária), onde a inteira será o segundo algarismo a direita da virgula e a parte fracionária é novamente multiplicada pela base e assim sucessivamente.

3.2.4 Conversão de Números entre duas Bases quaisquer

Para converter números de uma base b para uma outra base b' quaisquer (isso é, que não sejam os casos particulares anteriormente estudados), o processo prático utilizado é converter da base b dada para a base 10 e depois da base 10 para a base b' pedida.

Exemplo: Converter 43_5 para $()_9$.

$$43_5 = (4 \times 5^1 + 3 \times 5^0)_{10} = 23_{10} \implies 23/9 = 2 \text{ (resto 5) logo } 43_5 = 23_{10} = 25_9$$

Números com parte fracionária

Quando há uma parte fracionária no número a ser convertido, a melhor opção é fazer a conversão do número (parte inteira e fracionária) para a base 10 e depois fazer a conversão da base 10 para a outra base, seguindo as instruções apresentadas anteriormente.

3.3 Aritmética em Binário

3.3.1 Operação direta

A taboada da soma aritmética em binário é muito simples.

São poucas regras:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ (e "vai 1" para o dígito de ordem superior)}$$

$$1 + 1 + 1 = 1 \text{ (e "vai 1" para o dígito de ordem superior)}$$

Exemplo: Efetuar $011100 + 011010$

Observações:

a) Lembre-se: soma-se as colunas da direita para a esquerda, tal como uma soma em decimal.

b) No exemplo, são usadas, em seqüência, da direita para a esquerda, todas as regrinhas acima.

c) Na primeira linha, é indicado o resultado do "vai um".

d) Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

```
11-----> "vai um"
011100
011010 +
-----
110110
```

Vamos ver agora a tabuada da subtração:

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ ("vem um do próximo")}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Obs.: Como é impossível tirar 1 de zero, o artifício é "pedir emprestado" 1 da casa de ordem superior. Ou seja, na realidade o que se faz é subtrair 1 de 10 e encontramos 1 como resultado, devendo então subtrair 1 do dígito de ordem superior (aquele 1 que se "pediu emprestado"). Vamos lembrar que esse algoritmo é exatamente o mesmo da subtração em decimal a que já estamos acostumados desde o curso primário.

Exemplo: Efetuar $111100 - 011010$

Observações:

a) Lembre-se: subtrai-se as colunas da direita para a esquerda, tal como uma subtração em decimal.

b) No exemplo, são usadas, em seqüência, da direita para a esquerda, todas as regrinhas acima.

c) Na primeira linha, é indicado o resultado do "vem um".

d) Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

```
----02-> "vem um"
11100
01010 -
-----
10010
```

3.3.2 Complemento a Base

A implementação do algoritmo da subtração em computadores é complexa, requerendo vários testes. Assim, em computadores a subtração em binário é feita por um artifício. O método utilizado é o "Método do Complemento a Base" que consiste em encontrar o complemento do número em relação à base e depois somar os números. Os computadores funcionam sempre na base 2, portanto o complemento à base será complemento a dois. Computadores encontram o complemento a dois de um número através de um algoritmo que pode ser assim descrito:

a) Se o número é positivo, mantenha o número (o complemento de um número positivo é o próprio número)

b) Se o número é negativo:

- inverta o número negativo ou o subtraendo na subtração (todo 1 vira zero, todo zero vira um);
- some 1 ao número em complemento ;
- some as parcelas (na subtração, some o minuendo ao subtraendo) ;
- se a soma em complemento acarretar "vai-um" ao resultado, ignore o transporte final).

Como exemplo, vamos usar o algoritmo acima na subtração $1101 - 1100 = 0001$

| | | | |
|----------------------------|-----|------|---|
| mantém o minuendo | --> | 101 |) |
| inverte o subtraendo | --> | 011 | (|
| soma minuendo e subtraendo | --> | 0000 |) |
| soma 1 | --> | 0001 |) |
| ignora o "vai-um" | --> | 001 | (|

3.3.3 Multiplicação em binário

Vamos ver agora a tabuada da multiplicação:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

No entanto, também a multiplicação em computadores é feita por um artifício: para multiplicar um número A por n, basta somar A com A, n vezes.

Por exemplo, $4 \times 3 = 4 + 4 + 4$.

E a divisão também pode ser feita por subtrações sucessivas! O que

concluimos? Que qualquer operação aritmética pode ser realizada em computadores apenas através de somas (diretas ou em complemento).

3.4 Aritmética em outras bases

3.4.1 Operação direta

A tabuada da soma aritmética em uma base qualquer é simples.

Por exemplo, na base 4, as regras são:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 2$$

$$2 + 0 = 2$$

...

$$3 + 2 = 1 \text{ (} 5 - 4 \text{ (base) = 1 e "vai 1" para o dígito de ordem superior)}$$

$$3 + 3 = 2 \text{ (} 6 - 4 \text{ (base) = 2 e "vai 1" para o dígito de ordem superior)}$$

Exemplo: Efetuar $02323 + 02110$

Observações:

a) Lembre-se: soma-se as colunas da direita para a esquerda, tal como uma soma em decimal.

b) No exemplo, são usadas, em seqüência, da direita para a esquerda, todas as regrinhas acima.

c) Na primeira linha, é indicado o resultado do "vai um".

d) Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

11-----> "vai um"

02323

02110 +

11033

Vamos ver agora a tabuada da subtração:

$$0 - 0 = 0$$

$$0 - 1 = 3 \text{ ("vem um do próximo", que é igual a base: 4, logo } 4 + 0 = 4, 4 - 1 = 3)$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

...

$$2 - 3 = 3 \text{ ("vem um do próximo", que é igual a base: 4, logo } 4 + 2 = 6, 6 - 3 = 3)$$

...

$$3 - 2 = 1$$

$$3 - 3 = 0$$

Obs.: Como é impossível tirar 1 de zero, o artifício é "pedir emprestado" 1 da casa de ordem superior. Ou seja, na realidade o que se faz é subtrair 1 de 10 e

encontramos 3 como resultado (1 de ordem superior é do tamanho da base, logo igual a 4), devendo então subtrair 1 do dígito de ordem superior (aquele 1 que se "pediu emprestado" e que vale 4). Vamos lembrar que esse algoritmo é exatamente o mesmo da subtração em decimal a que já estamos acostumados desde o curso primário.

Exemplo: Efetuar $21321 - 02310$

Observações:

a) Lembre-se: subtrai-se as colunas da direita para a esquerda, tal como uma subtração em decimal.

b) No exemplo, são usadas, em seqüência, da direita para a esquerda, as regrinhas acima.

c) Na primeira linha, é indicado o resultado do "vem um".

d) Por simplicidade, no exemplo estamos considerando os dois números positivos. Solução:

```

15-----> "vem um"
21321
02310 -
-----
13011
    
```

As mesmas regras usada para as operações em base 4 valem para as outras bases como 8, 16, 5, 7. Observando que o maior algarismo é a base - 1.

Outros Exemplos:

```

11
3AC7816
26 44116 +
-----
610B916
    
```

```

11
347238
233348 +
-----
602578
    
```

```

2 10 12 11 10
3 1 3 2 04
0 2 3 3 34 -
-----
2 2 3 2 14
    
```

4 Lógica Booleana

4.1 Proposição

Definição: todo conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo.

As proposições transmitem pensamentos, isto é, afirmam fatos ou exprimem juízos que formamos a respeito de determinados entes.

Exemplos:

- ✓ Japão está situado no continente africano.
- ✓ A lâmpada da sala está acesa.
- ✓ A cidade de Recife é a capital de Pernambuco.

4.2 As três leis do pensamento

A lógica adota como regras fundamentais do pensamento os seguintes princípios (ou axiomas):

a) Princípio da Identidade

Se qualquer proposição é verdadeira, então ela é verdadeira.

b) Princípio da Não-Contradição

Uma proposição não pode ser verdadeira e falsa ao mesmo tempo.

c) Princípio do terceiro Excluído

Toda proposição ou é verdadeira ou é falsa, isto é, verifica-se sempre um destes casos e nunca um terceiro caso.

Com base nesses princípios as proposições simples são ou verdadeiras ou falsas - sendo mutuamente exclusivos os dois casos; daí dizer que a lógica clássica é bivalente

Exemplos de proposições falsas:

- ✓ O navegador Vasco da Gama descobriu o Brasil.
- ✓ O escritor francês Dante escreveu Os Lusíadas.
- ✓ número π é um número racional.
- ✓ O México está localizado na América do Sul.

4.3 Proposições

4.3.1 Valores lógicos das proposições

Definição: chama-se **valor lógico** de uma proposição a **verdade** se a proposição é

verdadeira e a **falsidade** se a proposição é falsa.

Os valores lógicos verdade e falsidade de uma proposição designam-se abreviadamente pelas letras **V** e **F** respectivamente. Assim, o que os princípios (axiomas) afirmam é que:

“Toda proposição tem um, e um só, dos valores V e F.”

O valor lógico de uma proposição P é a verdade (V) se P é verdadeira, escrevendo:

$v(P) = V$ e lê-se: “o valor lógico de P é V”

Exemplos:

| | |
|--|------------|
| P: O mercúrio é mais pesado que a água. | $v(P) = V$ |
| Q: O sol gira em torno da Terra. | $v(Q) = F$ |
| R: A lua é um satélite natural da Terra. | $v(R) = V$ |

4.3.2 Sentenças abertas

Definição: quando em uma proposição substituimos alguns (ou todos os) componentes por variáveis, obtemos uma sentença (proposição) aberta.

Seja a proposição “Magda é Uberlandense”, se substituimos o nome **Magda** pela variável **X**, obteremos a sentença aberta “X é Uberlandense”, que não é necessariamente verdadeira e nem falsa.

Exemplos:

P: X é filho de Y.
Q: $x - y = 12$
R: Se x é sobrinho de y, então, x é primo de z.

4.3.3 Proposições simples (atômicas)

Definição: chama-se **proposição simples**, a proposição que não contém nenhuma outra proposição como parte integrante de si.

Exemplos:

P: Carlos é careca.
Q: Pedro é estudante.
R: O galo põe ovos.

4.3.4 Conectivos lógicos

Definição: são palavras ou frases que são usadas para formar novas proposição a partir de outras proposições.

Os conectivos usuais em lógica matemática são:

| Conectivo | Palavras | Símbolo |
|---------------|--------------------------|-------------------------|
| Negação | Não | \sim ou \neg |
| Conjunção | E | \wedge |
| Disjunção | Ou | \vee |
| Condicional | Se ... então ... | \rightarrow |
| Bicondicional | ... Se, e somente se ... | \leftrightarrow |

4.3.5 Proposições compostas

Definição: são as proposições formadas por duas ou mais proposições simples e ligadas pelos conectivos lógicos.

Exemplos:

P: Carlos é careca **e** Pedro é estudante.

Q: Carlos é careca **ou** Pedro é estudante.

R: **Se** Carlos é careca, **então** Carlos é infeliz.

S: Carlos é careca **se, e somente se** Pedro é estudante.

4.4 Tabelas-verdade e conectivos.

Segundo o Princípio do terceiro excluído, toda proposição simples P é verdadeira ou é falsa, isto é, tem valor lógico V (verdade) ou o valor lógico F (falsidade). Em uma proposição composta, a determinação do seu valor lógico é feito segundo o princípio: “O valor lógico de qualquer proposição composta depende unicamente dos valores lógicos das proposições simples componentes, ficando por eles univocamente determinado.”

Para aplicar este princípio na prática, recorre-se ao uso do dispositivo denominado **Tabela-verdade**, que apresenta todos os possíveis valores lógicos da proposição composta correspondentes a todas as possíveis atribuições de valores lógicos às proposições simples correspondentes.

O número de linhas da tabela-verdade de uma proposição composta está em função do número de proposições simples que a compõem. A tabela-verdade de uma proposição composta com **n** proposições simples contém **2ⁿ** linhas.

Portanto, podemos observar:

- a) Para uma proposição simples P, o numero de linhas da tabela-verdade será: $2^1 = 2$, representando na tabela-verdade:

| |
|----------|
| P |
| V |
| F |

- b) Para uma proposição composta cujas proposições simples componentes são P e Q, o número de linhas da tabela-verdade será: $2^2 = 4$, que será representada:

| P | Q |
|---|---|
| V | V |
| V | F |
| F | V |
| F | F |

- c) Para uma proposição composta cujas proposições simples componentes são P, Q e R, o número de linhas da tabela-verdade será: $2^3 = 8$, que será representada:

| P | Q | R |
|---|---|---|
| V | V | V |
| V | V | F |
| V | F | V |
| V | F | F |
| F | V | V |
| F | V | F |
| F | F | V |
| F | F | F |

Observe que os valores lógicos V e F se alternam de quatro (4) em quatro para a primeira proposição (P), de dois (2) em dois para a Segunda proposição (Q) e de um (1) em um para a terceira proposição (R).

4.4.1 Operação Lógica Negação

Definição: chama-se negação de uma proposição P a proposição representada por “Não P”, cujo valor lógico é a verdade (V) quando P é falsa e a falsidade (F) quando P é verdadeira.

Simbolicamente, a negação de P indica-se com a notação “ $\sim P$ ”, que se lê: “**NÃO P**”

A tabela-verdade:

| P | $\sim P$ |
|---|----------|
| V | F |
| F | V |

$$v(\sim P) = \sim v(P)$$

Para uma proposição P, podemos formar a sua negação de qualquer um dos seguintes modos:

- ✓ “não é verdade que P”
- ✓ “é falso que P”
- ✓ “não” em P

Exemplo:

P: Lídia é estudiosa.

~P: Não é verdade que Lídia é estudiosa.

~P: É falso que Lídia é estudiosa.

~P: Lídia não é estudiosa.

Q: João não foi ao cinema.

~Q: É falso que João não foi ao cinema.

Observações: Pode-se ter algumas variações, por necessidades da língua portuguesa, por exemplo:

P: Todos os homens são elegantes.

~P: Nem todos os homens são elegantes.

Q: Nenhum homem é elegante.

~Q: Algum homem é elegante.

4.4.2 Operação Lógica Conjunção

Definição: chama-se conjunção de duas proposições P e Q a proposição representada por “P e Q”, cujo valor lógico é a verdade (V) quando as proposições P e Q são ambas verdadeiras e a falsidade (F) nos demais casos.

Simbolicamente, a conjunção de duas proposições P e Q indica-se com a notação “ $P \wedge Q$ ”, que se lê “**P E Q**”. A tabela-verdade:

| P | Q | $P \wedge Q$ |
|---|---|--------------|
| V | V | V |
| V | F | F |
| F | V | F |
| F | F | F |

$$v(P \wedge Q) = v(P) \wedge v(Q)$$

Exemplo:

P: Pitágoras era Grego

Q: Descartes era Francês

$P \wedge Q$: Pitágoras era Grego e Descartes era Francês.

$$v(P) = V \text{ e } v(Q) = V \Rightarrow v(P \wedge Q) = V$$

4.4.3 Operação Lógica Disjunção

Definição: chama-se disjunção de duas proposições P e Q a proposição representada por “P ou Q”, cujo valor lógico é a verdade (V) quando ao menos uma das proposições P e Q é verdadeira e a falsidade (F) quando as proposições P e Q são ambas falsas.

Simbolicamente, a conjunção de duas proposições P e Q indica-se com a notação “ $P \vee Q$ ”, que se lê “**P OU Q**”.

A tabela-verdade:

| P | Q | $P \vee Q$ |
|---|---|------------|
| V | V | V |
| V | F | V |
| F | V | V |
| F | F | F |

$$v(P \vee Q) = v(P) \vee v(Q)$$

Exemplo:

P: A cidade de Paris é a capital da França

Q: O sol é um satélite artificial da terra

$P \vee Q$: A cidade de Paris é a capital da França ou o sol é um satélite artificial da terra.

$$v(P) = V \text{ e } v(Q) = F \Rightarrow v(P \vee Q) = V$$

Na linguagem coloquial a palavra ou tem dois sentidos. Exemplifiquemos:

P: Amilton é bombeiro ou electricista.

Q: Rosa é mineira ou goiana.

Na proposição Q (Rosa é mineira ou goiana), está afirmando que somente uma das proposições é verdadeira, pois não é possível ocorrer as duas coisas: Rosa ser mineira e goiana ao mesmo tempo.

Na proposição P, diz-se que o ou é inclusivo, já na proposição Q diz-se que o ou é exclusivo. Logo a proposição P é uma disjunção inclusiva ou simplesmente disjunção; e a proposição Q é a disjunção exclusiva.

A disjunção exclusiva de duas proposições P e Q é a proposição composta “ $P \underline{\vee} Q$ ”, que se lê: “ou P ou Q” ou se lê “P ou Q, mas não ambos”. É a falsidade (F) quando o valor lógico das proposições P e Q forem ambos verdadeiros ou ambos falsos e a verdade (V) quando P e Q tem valores lógicos diferentes.

A tabela-verdade:

| P | Q | $P \underline{\vee} Q$ |
|---|---|------------------------|
| V | V | F |
| V | F | V |
| F | V | V |
| F | F | F |

$$v(P \underline{\vee} Q) = v(P) \underline{\vee} v(Q)$$

4.4.4 Operação Lógica Condicional

Definição: chama-se proposição condicional de duas proposições P e Q a proposição representada por “se P então Q”, cujo valor lógico é a falsidade (F) no caso em que a proposição P é verdadeira e Q é falsa e a verdade (V) nos demais casos.

Simbolicamente, a condicional de duas proposições P e Q indica-se com a notação “ $P \rightarrow Q$ ”, que se lê:

- **“P SOMENTE SE Q”**
- “p é condição suficiente para q”
- “q é condição necessária para p”
- “p implica (ou acarreta) q”

Na condicional “ $P \rightarrow Q$ ”, diz-se que P é o antecedente e Q é o conseqüente. O símbolo “ \rightarrow ” é chamado de símbolo de implicação.

A tabela-verdade:

| P | Q | $P \rightarrow Q$ |
|---|---|-------------------|
| V | V | V |
| V | F | F |
| F | V | V |
| F | F | V |

$$v(P \rightarrow Q) = v(P) \rightarrow v(Q)$$

Observação: Uma condicional $P \rightarrow Q$ não afirma que o conseqüente Q se deduz ou é conseqüência do antecedente P; o que uma condicional afirma é unicamente uma relação entre os valores lógicos do antecedente e do conseqüente de acordo com a tabela-verdade apresentada acima.

Exemplo:

P: Paris é a capital da França

Q: O sol é um satélite natural da terra

$P \rightarrow Q$: Se Paris é a capital da França então o sol é um satélite natural da terra.

$v(P) = V$ e $v(Q) = F \Rightarrow v(P \rightarrow Q) = F$

4.4.5 Operação Lógica Bi-condicional

Definição: chama-se proposição bi-condicional de duas proposições P e Q a proposição representada por “P se, e somente se Q”, cujo valor lógico é a verdade (V) quando as proposições P e Q são ambas verdadeiras ou ambas falsas e a falsidade (F) nos demais casos.

A conjunção da sentença “ $P \rightarrow Q$ ” com a sentença “ $Q \rightarrow P$ ” resulta na sentença “ $P \leftrightarrow Q$ ”, assim temos $(P \rightarrow Q) \wedge (Q \rightarrow P)$ equivale a $(Q \leftrightarrow P)$.

Simbolicamente, a bi-condicional de duas proposições P e Q indica-se com a notação “ $P \leftrightarrow Q$ ”, que se lê:

- **“P SE, E SOMENTE SE Q”**
- “p é equivalente a q”
- “q se, e somente se p”

- “q é equivalente a p”
- “p é condição necessária e suficiente para q”
- “q é condição necessária e suficiente para p”

A tabela-verdade:

| | | |
|---|---|-----------------------|
| P | Q | $P \leftrightarrow Q$ |
| V | V | V |
| V | F | F |
| F | V | F |
| F | F | V |

$$v(P \leftrightarrow Q) = v(P) \leftrightarrow v(Q)$$

Exemplo:

P: A cidade de Paris é a capital da França

Q: O sol é um satélite natural da terra

$P \leftrightarrow Q$: A cidade de Paris é a capital da França se, e somente se o sol é um satélite natural da terra.

$$v(P) = V \text{ e } v(Q) = F \Rightarrow v(P \leftrightarrow Q) = F$$

4.4.6 Tabela Verdade dos conectores

Resumindo, tem-se a seguinte tabela verdade:

| p | q | $p \wedge q$ | $p \vee q$ | $p \underline{\vee} q$ | $p \rightarrow q$ | $p \leftrightarrow q$ |
|----------|----------|--------------------------------|------------------------------|--|-------------------------------------|---|
| V | V | V | V | F | V | V |
| V | F | F | V | V | F | F |
| F | V | F | V | V | V | F |
| F | F | F | F | F | V | V |

| p | $\sim p$ |
|----------|----------------------------|
| V | F |
| F | V |

4.4.7 Ordem De Precedência Dos Conectores

Para reduzir o número de parêntesis necessários em uma proposição composta (fórmula lógica proposicional), estipula-se uma ordem na qual os conectores são aplicados. A ordem de precedência é:

maior



- a) conectivos dentro de parêntesis, do mais interno para o mais externo
- b) \sim
- c) \wedge
- d) \vee ou $\underline{\vee}$

- e) \rightarrow
 - f) \leftrightarrow
- menor

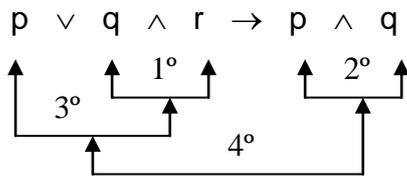
Quando há dois ou mais conectivos de mesma ordem de precedência, o conectivo mais a esquerda na fórmula proposicional tem prioridade sobre o conectivo à direita.

Exemplos:

a) “ $p \vee \sim q$ ” equivale a “ $(p \vee (\sim q))$ ”

b) “ $p \vee q \rightarrow r$ ” equivale a “ $((p \vee q) \rightarrow r)$ ”

c) “ $p \vee q \wedge r \rightarrow p \wedge q$ ” equivale a “ $((p \vee (q \wedge r)) \rightarrow (p \wedge q))$ ”



5 OPERAÇÃO DO MICROPROCESSADOR

A primeira característica a considerar num computador é sua unidade central de processamento (UCP), que poderá fornecer uma série de indicações sobre o equipamento. A UCP ou CPU (Central Processing Unit), também pode ser chamada de processador ou microprocessador, os quatro termos são equivalentes. Tudo o que acontece num computador provém da UCP, que gerencia todos os recursos disponíveis no sistema. Seu funcionamento é coordenado pelos programas, que indicam o que deve ser feito e quando. Basicamente, a UCP executa cálculos muito simples como somas e comparações entre números, mas com uma característica muito especial: a velocidade extremamente elevada.

A função da UCP é sempre a mesma. O que as diferenciam é sua estrutura interna e, o mais importante, o fato de cada uma ter seu conjunto de instruções próprio. Ou seja, um programa escrito para uma UCP dificilmente poderá ser executado diretamente em outra, sendo esse é um dos principais motivos da incompatibilidade entre os computadores.

A UCP trabalha diretamente com a memória principal. O conteúdo da memória principal é uma combinação de informações e instruções. As instruções que o processador pode executar diretamente estão inseridos nele mesmo e possui uma linguagem de máquina da UCP.

O processamento é feito pela Unidade Central de Processamento utilizando o ciclo busca-execução regulado pelo clock (relógio). A seqüência desse ciclo é:

- Buscar (cópia) instrução na memória principal;
- Executar a instrução;
- Buscar a instrução seguinte;
- Executar a instrução seguinte;
- E assim por diante (milhões de vezes por segundo).

As instruções em linguagem de máquina são muito primitivas. Por exemplo:

- Ler (copiar) conteúdo de um endereço de memória no registrador do processador central;
- Comparar duas informações;
- Adicionar ou subtrair dois números;
- Escrever palavra na memória ou dispositivo de saída.

Estas etapas compõem o que se denomina ciclo de instrução. Este ciclo se repete indefinidamente até que o sistema seja desligado, ou ocorra algum tipo de erro, ou seja, encontrada uma instrução de parada.

4.1 Arquivos

A palavra **Arquivo** é um dos conceitos mais importantes em computação. Em inglês é "File". Arquivo nada mais é que um conjunto de dados gravados na memória secundária (disco rígido, disquete, fita magnética, CD-ROM, etc). Os arquivos são uma

forma de organizar os dados dentro da memória secundária. Se os dados estivessem todos espalhados, por exemplo, ao longo de um disquete, seu acesso seria extremamente complicado. Podemos fazer uma analogia entre dados, arquivos, casas e ruas. Os dados corresponderiam às casas, enquanto que os arquivos corresponderiam às ruas. Seria difícil localizar uma casa, sabendo apenas os nomes de seus moradores. Sabendo o nome da rua, o acesso é bem mais imediato. Por essa razão, os dados são agrupados em arquivos. Sabendo o nome do arquivo, fica mais fácil localizar os dados.

Os nomes completos dos arquivos são tradicionalmente divididos em duas partes (Nome e Extensão), separadas por um ponto. Por exemplo: CARTA.DOC

Neste caso, o “nome” é CARTA, e a extensão é DOC. É comum nesse caso, dizer que o nome do arquivo é CARTA, ou então CARTA.DOC, leia “carta ponto doc”. A extensão serve para indicar o tipo de arquivo. Por exemplo, DOC significa que trata-se de um documento de texto; JPG é um arquivo gráfico, EXE é um arquivo de programa, WAV é um arquivo de som, e assim por diante. No passado, os arquivos podiam usar no máximo 8 caracteres para o nome e 3 caracteres para a extensão (formato 8.3). A partir do lançamento do sistema operacional Windows 95 da Microsoft e do OS/2, da IBM, os PCs passaram a utilizar nomes longos. Hoje podemos usar para os arquivos, nomes como:

Trabalho de Computação.DOC
Foto da Cascata.JPG
Canto do Galo.MP3

4.2 Programas

Os computadores passam o tempo todo executando programas. Os programas nada mais são que grupos de instruções e dados. Por exemplo, quando você está executando um jogo no computador, está na verdade executando um programa. Se você deixa o computador “parado”, por exemplo, na hora do almoço, pode pensar que nessa hora o computador não está executando nenhum programa. Engana-se. Na verdade, mesmo que você não tenha dado nenhum comando, o computador está na verdade executando um programa interno. O que este programa faz é monitorar continuamente o teclado e o mouse, para checar se você envia um novo comando. A única hora em que o computador não está executando programa nenhum é quando está desligado ou em estado de espera.

Para que um programa possa ser executado, é preciso que seja transferido para a memória RAM. A maioria dos programas ficam armazenados em disco (disco rígido, CD-ROM, etc.), mas a CPU não pode executar nenhum programa diretamente a partir do disco. O programa precisa ser antes lido do disco e carregado na RAM. Por exemplo, para executar o programa Bloco de Notas (pequeno editor de textos que acompanha o Windows), é preciso que você clique os menus do Windows na seqüência (por exemplo):

Iniciar / Programas / Acessórios / Bloco de Notas

Deve-se observar que os nomes dos programas que aparecem nos menus do Windows não são necessariamente iguais aos nomes com os quais esses programas estão armazenados no disco. O programa que aparece nos menus com o nome de Bloco de Notas é na verdade o arquivo NOTEPAD.EXE.

Uma vez fornecida a seqüência de cliques do mouse, o NOTEPAD.EXE é lido do disco rígido e carregado na memória RAM. A CPU pode então executar o programa. A figura 23 simboliza a leitura do programa NOTEPAD.EXE a partir do disco rígido para a memória RAM (essa operação é chamada de carga), e seu processamento pela CPU (essa operação é chamada de execução).

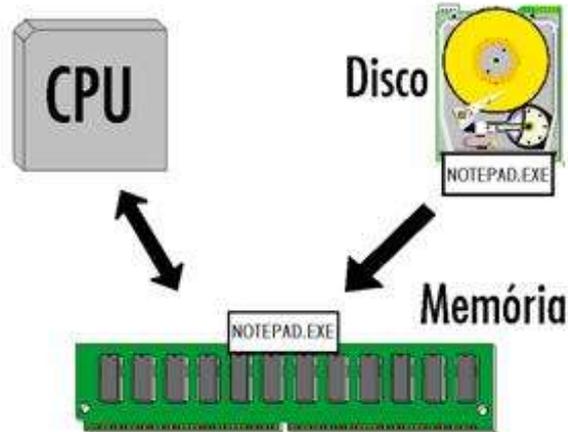


Figura 23 – Leitura e Execução de um Programa

4.3 Sistema Operacional

Pode-se pensar que é realizada uma mágica com a leitura do arquivo NOTEPAD.EXE do disco rígido para a memória, e a seguir sua execução pela CPU. Na verdade, quem leu o arquivo NOTEPAD.EXE e providenciou sua execução foi um outro programa. Trata-se de um programa que fica o tempo todo na memória, chamado de sistema operacional. Neste exemplo, trata-se do sistema operacional Windows. Uma das várias funções do sistema operacional é ficar o tempo todo ativo na memória RAM, esperando que o usuário comande a execução de algum programa.

Um sistema operacional é um grande conjunto de programas e arquivos auxiliares. O próprio Bloco de Notas é um programa que faz parte do sistema operacional Windows. Outro programa importante do Windows é o “Windows Explorer” em que o arquivo tem o nome de EXPLORER.EXE. Este programa é o responsável por, entre outras ações, receber os comandos que o usuário envia para o computador, através do teclado e do mouse. Por exemplo quando clicamos em Iniciar e aparece um menu, no qual consta a opção Programas, depois Acessórios e finalmente Bloco de Notas, é o EXPLORER.EXE que está recebendo os comandos do mouse e apresentando os menus na tela. Portanto, quando é fornecido um comando ou programa, como o Bloco de Notas, o que ocorre é o seguinte:

- 1) Inicialmente o EXPLORER.EXE está checando se algum comando foi dado pelo teclado ou pelo mouse.
- 2) Se clicar na seqüência: Iniciar / Programas / Acessórios / Bloco de Notas.
- 3) O programa EXPLORER.EXE identifica que o programa denominado de Bloco de Notas é na verdade o arquivo NOTEPAD.EXE. Ele envia comandos para outros componentes do sistema operacional para que procurem no disco o arquivo NOTEPAD.EXE e para que façam sua carga na memória RAM.

4) O Windows é um sistema operacional que permite que vários programas possam ser executados ao mesmo tempo. Na verdade o processador dedica uma fração do seu tempo para cada um dos programas em execução.

5) Ao finalizar o programa NOTEPAD.EXE, fechando a sua janela ou usando o comando Arquivo / Sair o sistema operacional retira este arquivo da memória.

Pode-se entender que nenhum programa chega até a memória por mágica, e sim, através do controle feito pelo sistema operacional. Alguém mais observador pode então ficar com a seguinte dúvida: “Se é o sistema operacional quem lê para a RAM todos os programas a serem executados, como é então que o próprio sistema operacional chegou na RAM?”.

No instante em que é ligado o computador, a RAM não contém programa algum (os dados existentes na RAM são apagados quando o computador é desligado). Nesse instante, o sistema operacional está armazenado no disco rígido e precisa ser carregado na memória. Quem faz a carga do sistema operacional para a memória é um programa chamado BIOS, que fica gravado na memória ROM. A memória ROM não perde seus dados quando o computador é desligado.

Portanto, no instante que é ligado o computador, o BIOS já está na memória, e é imediatamente processado pela CPU. O processamento do BIOS começa com uma contagem de memória, seguido de alguns testes rápidos no hardware, e finalmente a leitura do sistema operacional do disco para a memória RAM. Esse processo, ou seja, a carga do sistema operacional na memória RAM, é chamado de boot. A figura 24 mostra o processo de boot:

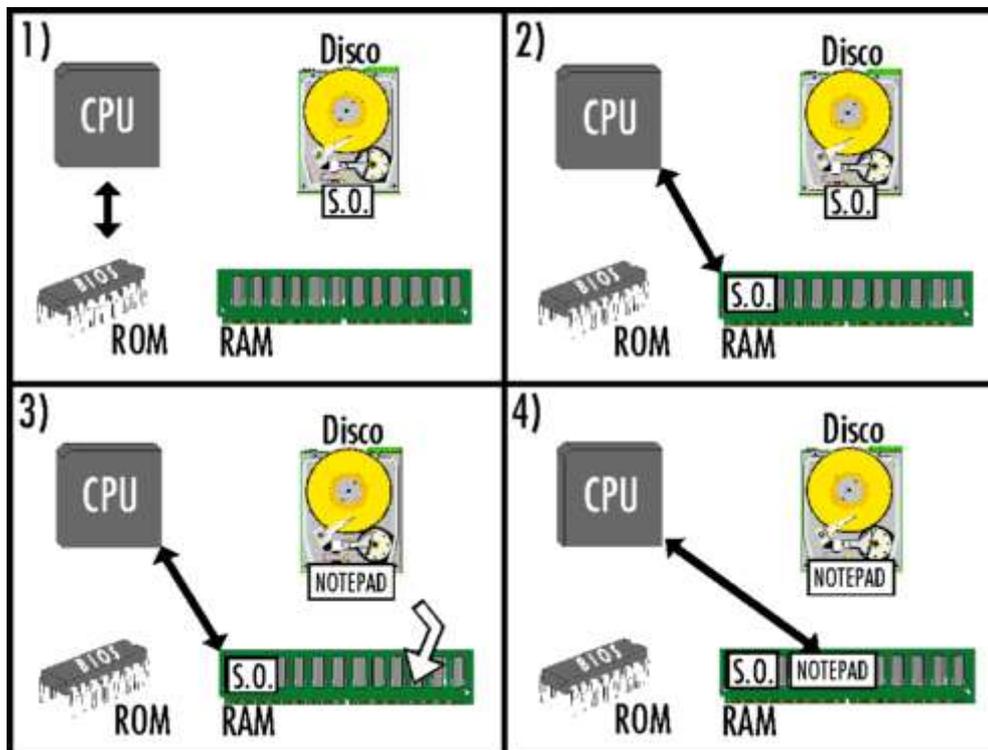


Figura 24 – Boot e Carga do Sistema Operacional na memória

Uma das atribuições do sistema operacional é fazer a carga e providenciar a execução dos programas que o usuário solicita. Mesmo quando um programa qualquer está em execução, o sistema operacional continua ajudando. Por exemplo, muitos

programas precisam realizar acesso ao teclado, vídeo e impressora, assim como acessos ao disco rígido para ler e gravar arquivos. Todos esses acessos são realizados pelo sistema operacional, que fica o tempo todo ativo, prestando serviços aos programas que estão sendo executados.

O sistema operacional também faz um gerenciamento dos recursos do computador, para evitar que os programas entrem em conflito. Por exemplo, o sistema operacional evita que dois programas simultaneamente acessem a mesma área da memória, o que poderia causar grandes problemas. O sistema operacional funciona como um “maestro”, providenciando para que todos os programas e todos os componentes do computador funcionem de forma harmônica.

6 COMPONENTES DO MICROPROCESSADOR

5.1 Introdução

Um computador digital (figura 25) pode ser descrito de uma forma simplificada segundo o diagrama de blocos:

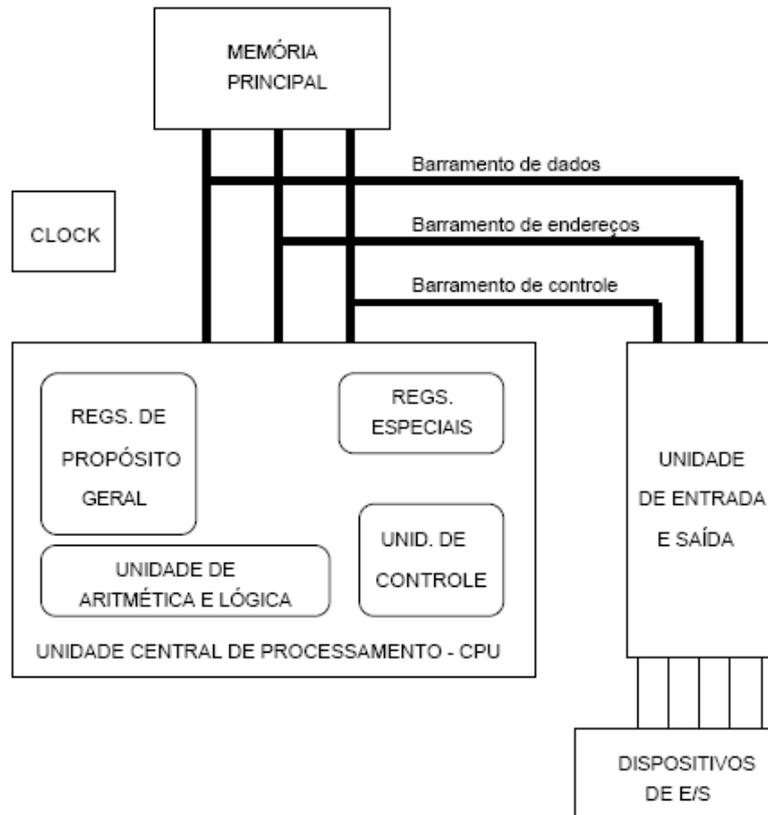


Figura 25 – Esquema simplificado da CPU utilizando diagrama de blocos

A Unidade Central de Processamento é composta de Unidade Lógica e Aritmética (ULA), da Unidade de Controle (UC), de um conjunto de registradores e um barramento interno que interliga os componentes. O barramento interno tem o tamanho da palavra da CPU.

A figura 26 apresenta um esquema simplificado dos componentes de um computador. Pode-se dizer de forma simplificada que um computador possui CPU, Memória, Dispositivos de Entrada/Saída e o Barramento que interliga todos.

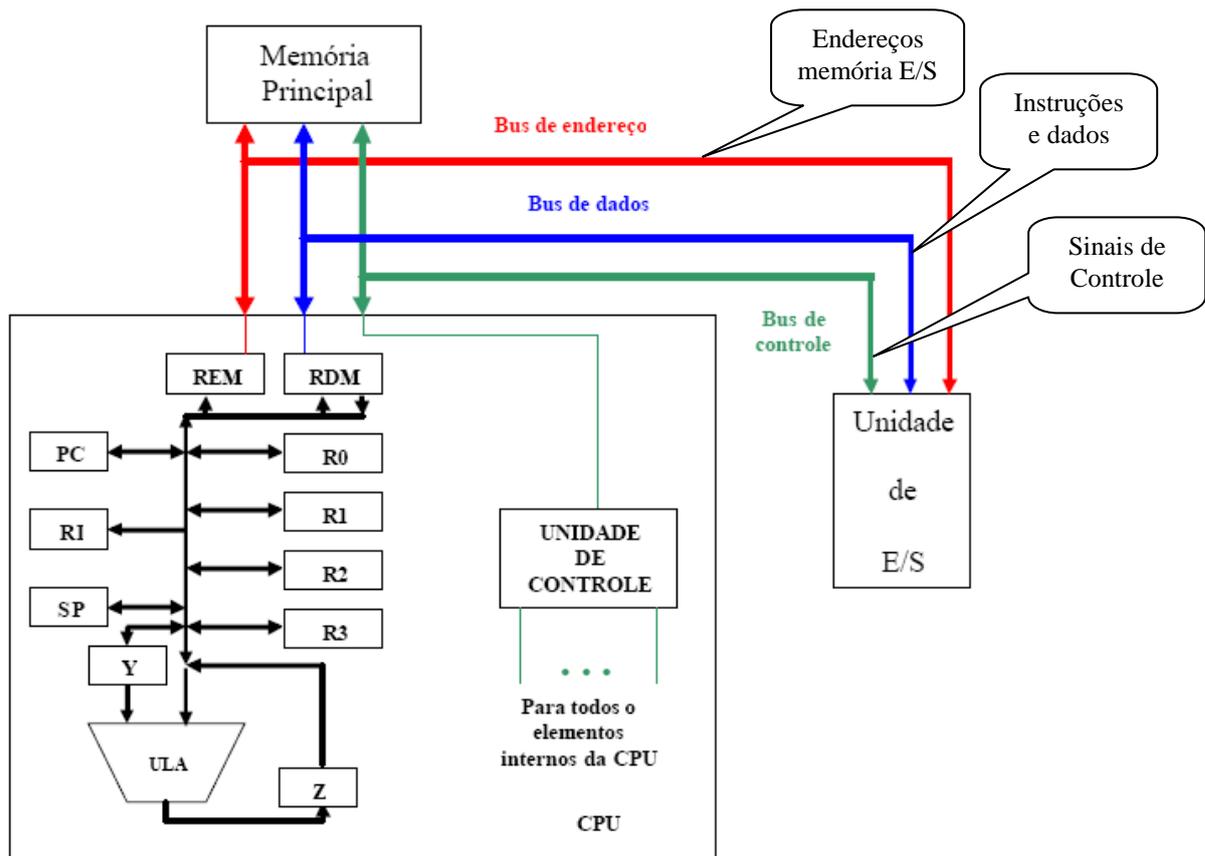


Figura 26 – Esquema simplificado do Computador

5.2 Unidade Lógica e Aritmética (ULA)

A ULA (figura 27) é o dispositivo da UCP que executa realmente as operações matemáticas com os dados.

A ULA é um aglomerado de circuitos lógicos e componentes eletrônicos simples que, integrados, realizam as operações já mencionadas. Ela pode ser uma parte pequena da pastilha do processador, usada em pequenos sistemas, ou pode compreender um considerável conjunto de componentes lógicos de alta velocidade. Apesar da grande variação de velocidade, tamanho e complexidade, as operações aritméticas e lógicas realizadas por uma ULA seguem sempre os mesmos princípios fundamentais.

Realiza operações lógicas e aritméticas (soma, and, etc) com operandos armazenados em seus registradores temporários. Opera com o conjunto de bits do tamanho da palavra. Os operandos, antes de serem transferidos para os registradores temporários, podem estar na memória ou nos registradores de uso geral.

Exemplo: Suponha que a CPU acabou de buscar a seguinte instrução: Some R1 com R2 o coloque o resultado em R1, veja esta operação na figura 27.

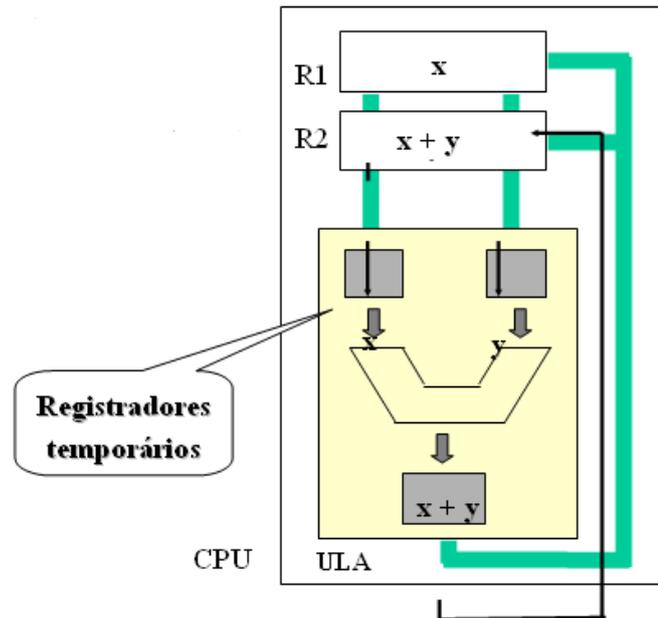


Figura 27 – Exemplo de operação da ULA

5.3 Unidade de Controle (UC)

A Unidade de Controle (figura 4) é o dispositivo mais complexo da UCP. Além de possuir a lógica necessária para realizar a movimentação de dados e instruções de e para a UCP, através dos sinais de controle que emite em instantes de tempo programados, esse dispositivo controla a ação da ULA. Os sinais de controle emitidos pela UC ocorrem em vários instantes durante o período de realização de um ciclo de instrução e, de modo geral, todos possuem uma duração fixa e igual, originada em um gerador de sinais usualmente conhecido como relógio.

Ao contrário de circuitos integrados mais comuns, cuja função é limitada pelo hardware, a unidade de controle é mais flexível. Ela recebe instruções da unidade de E/S, as converte em um formato que pode ser entendido pela unidade de aritmética e lógica, e controla qual etapa do programa está sendo executado.

5.4 Relógio

É o dispositivo gerador de pulsos cuja duração é chamada de ciclo, também conhecido como clock (ver figura 25). A quantidade de vezes em que este pulso básico se repete em um segundo define a unidade de medida do relógio, denominada frequência, a qual também usamos para definir velocidade na UCP.

A unidade de medida usual para a frequência dos relógios de UCP é o Hertz (Hz), que significa um ciclo por segundo. Como se trata de frequências elevadas, abreviam-se os valores usando-se milhões de Hertz, ou de ciclos por segundo (Megahertz ou simplesmente, MHz). Assim, por exemplo, se um determinado processador funciona como seu relógio oscilando 25 milhões de vezes por segundo, sua frequência de operação é de 25 MHz. E como a duração de um ciclo, seu período, é o inverso da frequência, então cada ciclo, neste exemplo, será igual ao inverso de

25.000.000 ou $1/25.000.000=0,00000004$ ou 40 nanossegundos.

5.5 Registradores

Para que um dado possa ser transferido para a ULA, é necessário que ele permaneça, mesmo que por um breve instante, armazenado em um registrador (figura 4). Além disso, o resultado de uma operação aritmética ou lógica realizada na ULA deve ser armazenado temporariamente, de modo que possa ser utilizado mais adiante ou apenas para ser, em seguida, transferido para a memória.

Para atender a estes propósitos, a UCP é fabricada com uma certa quantidade de registradores, destinados ao armazenamento de dados. Servem, pois, de memória auxiliar da ULA. São extremamente rápidos. A gravação ou a leitura de uma informação é realizada de forma muito rápida, se comparada com estas operações nas memórias.

Diz-se que um registrador é de x bits se ele puder armazenar uma informação de x bits.

Na CPU existem registradores de uso geral e registradores especiais.

5.7.1 Registradores de uso geral

Armazenam dados temporários ou muito utilizados (figura 28). Ex: usar uma variável armazenada em um registrador para efetuar um cálculo. São acessíveis pelo programador para utilização na programação. Possuem o tamanho da palavra da CPU. As CPU's possuem cerca de 10 a 50 registradores, dependendo do projeto.

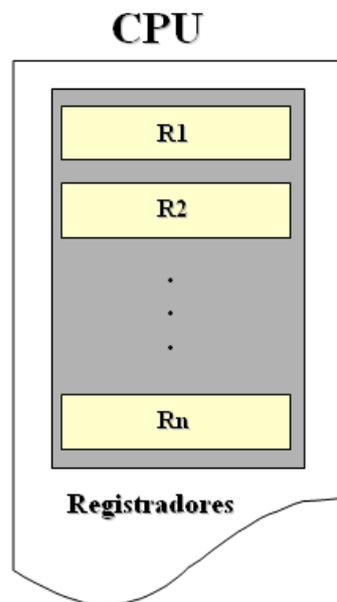


Figura 28 – Esquema dos registradores de uso geral

5.7.2 Registradores especiais

São registradores que possuem uma função específica (ver figura 4). Estes registradores são transparentes em termos de funcionalidade em processamento de

dados. Sua função é tipicamente eletrônica. Possuem tamanho variado, dependendo da função que exerce. Os mais conhecidos: ACC, CI, RDM, REM, RI.

5.7.3 Registrador de Instrução (RI)

É o registrador especial que tem a função específica de armazenar a instrução a ser executada pela UCP. Ao se iniciar um ciclo de instrução, a UC emite o sinal de controle que acarretará a realização de um ciclo de leitura para buscar a instrução na memória, e que, via barramento de dados e RDM, será armazenada no RI. Possui o tamanho de uma instrução.

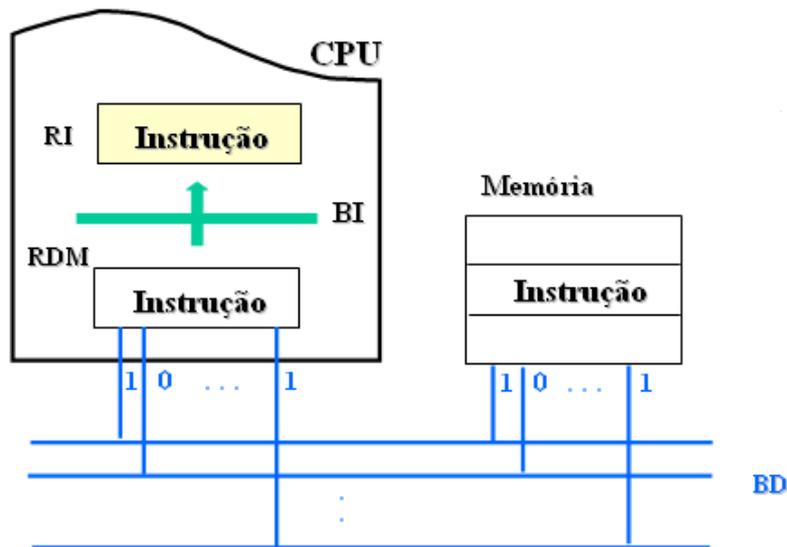


Figura 29 – Esquema de funcionamento do RI

5.7.4 Contador de Instrução (CI)

É o registrador cuja função específica é armazenar o endereço da próxima instrução a ser executada, também conhecido como Program Counter (PC) em alguns livros. Tão logo a instrução que vai ser executada seja buscada (lida) da memória para a UCP, o sistema providencia a modificação do conteúdo do CI de modo que ele passe a armazenar o endereço da próxima instrução na seqüência. Por isso, é comum definir a função do CI como sendo a de "armazenar o endereço da próxima instrução", que é o que realmente ele faz durante a maior parte da realização de um ciclo de instrução. Possui o tamanho do barramento de endereço.

5.7.5 Registrador de Dados de Memória (RDM)

É o registrador específico utilizado pela UCP e os periféricos para comunicação e transferência de dados (informações). Em geral o RDM (figura 30) possui um tamanho igual ao da palavra do barramento de dados, Toda comunicação da UCP com o barramento de dados (BD) é feita através do RDM.

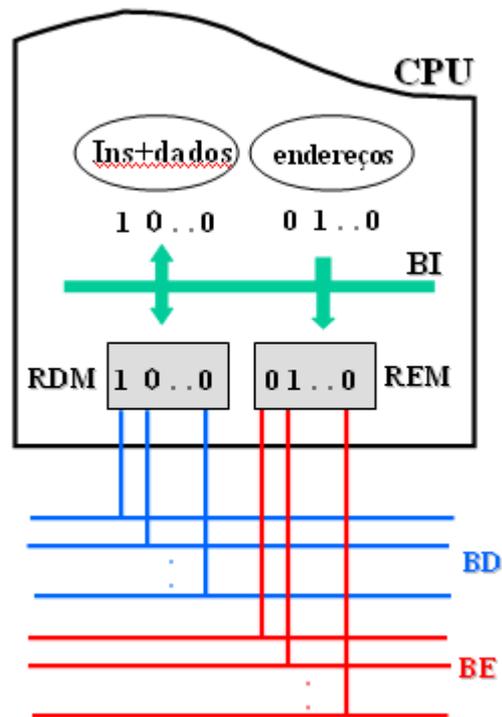


Figura 30 – Esquema de uso do RDM e REM

5.7.6 Registrador de Endereços de Memória (REM)

É o registrador específico utilizado pela UCP e os periféricos para comunicação e transferência de um endereço. Em geral o REM (figura 30) possui um tamanho igual ao dos endereços da memória. Toda comunicação da UCP com o barramento de endereço (BE) é feita através do REM.

5.7.7 Registrador Acumulador (ACC)

É utilizado para guardar inicialmente um dos valores e depois armazenar o resultado do cálculo efetuado.

5.7.8 Registrador Apontador de Pilha (SP)

Também conhecido como stack pointer, é utilizado em conjunto com o registrador SS, para acessar a área de pilha na memória; aponta para o topo da pilha.

5.6 Decodificador de Instrução

É um dispositivo utilizado para identificar as operações a serem realizadas, que estão correlacionadas à instrução em execução. Em outras palavras, cada instrução é uma ordem para que a UCP realize uma determinada operação. Como são muitas instruções, é necessário que cada uma possua uma identificação própria e única. A unidade de controle está, por sua vez, preparada para sinalizar adequadamente aos diversos dispositivos da UCP, conforme ela tenha identificado a instrução a ser executada.

O decodificador recebe na entrada um conjunto de bits previamente escolhido e específico para identificar uma instrução de máquina e possui 2^N saídas, sendo N a quantidade de algarismos binários do valor de entrada.

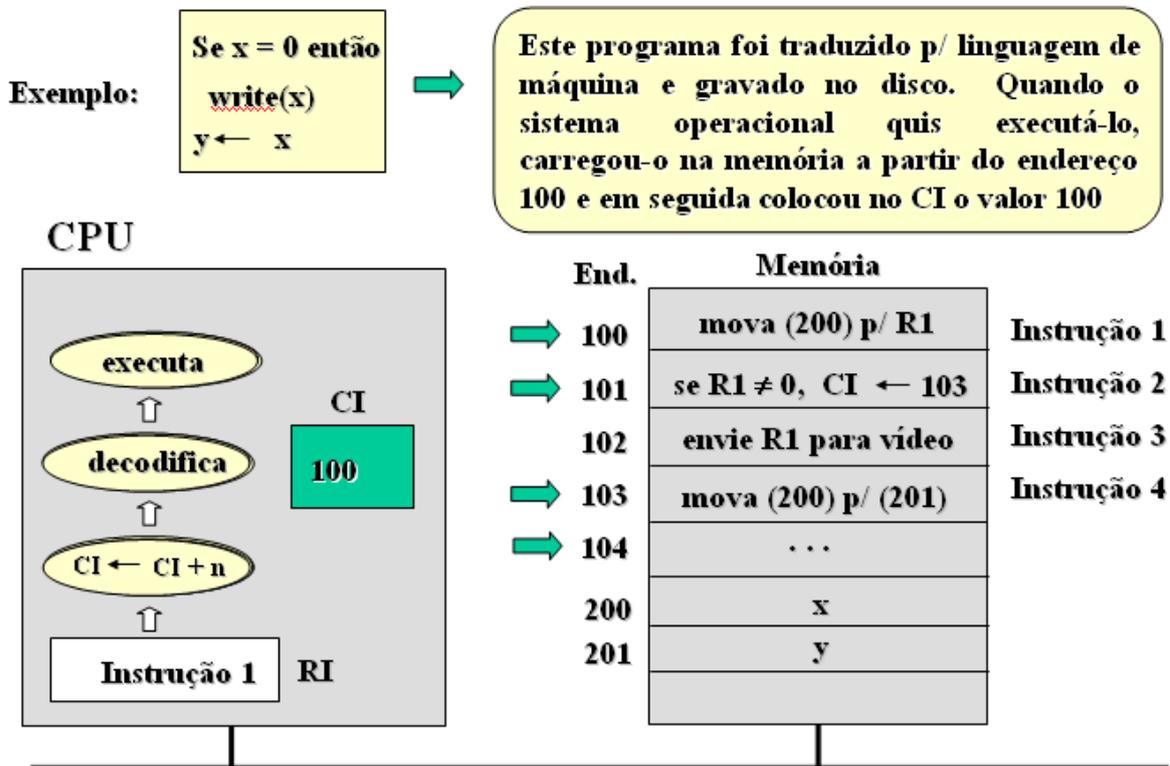


Figura 31 – exemplo de execução de um conjunto de instruções

5.7 Barramento

Para cada tipo de informação (dados, endereços e controle) a CPU possui pinos/conectores específicos destinados a enviar/receber informações dos componentes (figura 26).

Através do barramento o processador faz a comunicação com o seu exterior. Nele trafegam os dados lidos da memória, escritos na memória, enviados para interfaces e recebidos de interfaces. Pode ser dividido em três grupos:

- Barramento de dados
- Barramento de endereços
- Barramento de controle

Exemplo: Se o processador quiser armazenar o dado 10101100b no endereço 5h:

- Colocar, no barramento de dados, o valor do dado (100101100b);
- No barramento de endereços, o valor do endereço (5h);
- Barramento de controle, confirmar que se trata de uma operação de escrita em memória.

Tudo isso é feito simultaneamente e é transparente ao usuário. Observando que todos os circuitos recebem o mesmo dado e o mesmo endereço simultaneamente.

Através do barramento de controle, os demais circuitos souberam que o dado era destinado à memória, portanto deveriam ficar inoperantes.

5.7.1 Definindo os componentes do barramento

Através do barramento de endereços o processador pode especificar qual a placa ou interface através da qual quer transmitir ou receber dados, e também especificar o endereço de memória no qual deseja ler ou armazenar dados. A maioria dos processadores modernos têm barramento de endereços com 36 bits, podendo assim endereçar até 64 GB de memória física.

O barramento de dados tem 64 bits na maioria dos processadores modernos. O barramento de endereços é sempre unidirecional, ou seja, os bits são gerados pelo processador. O barramento de dados é bidirecional, ou seja, os dados são ora transmitidos, ora recebidos pelo processador.

O barramento de controle contém vários sinais que são necessários ao funcionamento do processador, bem como controlar o tráfego do barramento de dados. Alguns dos seus sinais são de saída, outros são de entrada, outros são bidirecionais. Existem sinais para indicação do tipo de operação (leitura ou escrita), sinais de especificação de destino/origem de dados (memória ou E/S), sinais de sincronismo, sinais de interrupção, sinais que permitem a outro dispositivo tomar o controle do barramento, sinais de clock, sinais de programação e diversos outros.

| CPU | BD | BE | CE (endereços) |
|------------|-----------|-----------|-----------------------|
| 8086 | 16 | 20 | 1 M |
| 8088 | 8 | 20 | 1 M |
| 80286 | 16 | 24 | 16 M |
| 80386-SX | 16 | 24 | 16 M |
| 80386-DX | 32 | 32 | 4 G |
| 80486 | 32 | 32 | 4 G |
| Pentium | 64 | 32 | 4 G |
| Pentium 4 | 64 | 36 | 64 G |

Tabela 1 – Especificação de barramento para cada processador

Na maioria dos casos, o barramento do processador é o mais veloz existente em uma placa de CPU, mas isto nem sempre ocorre. Por exemplo, uma placa de CPU pode ter o processador operando com barramento de 100 MHz e as memórias operando a 133 MHz. Ter a memória mais rápida é vantajoso no caso de placas com vídeo onboard, já que estaria sendo acessada, ora pelo processador, ora pelos circuitos de vídeo. Neste caso o barramento da memória seria o de tráfego mais intenso do computador.

Duas características principais determinam a performance da CPU:

- número de pinos de endereço;
- número de pinos de dados.

O número de pinos de endereço determina a quantidade máxima de endereçamento. Um chip com m pinos de endereço pode endereçar até 2^m locais de memória. Os valores comuns de m são 16, 20, 32 e 64.

O número de pinos de dados determina a largura da leitura/escrita. Um chip com n pinos de dados pode ler ou escrever uma palavra de n bits em apenas uma operação. Os valores comuns de n : 8, 16, 32, 64. Exemplo: Uma CPU com 8 pinos de dados precisa de 4 operações para ler palavra de 32 bits. E uma CPU com 32 pinos de dados faz em apenas 1 operação a leitura de uma palavra de 32 bits.

A figura 8 mostra o diagrama em blocos de uma placa de CPU. Nele podemos ver o processador, as memórias, o barramento de I/O e suas interfaces.

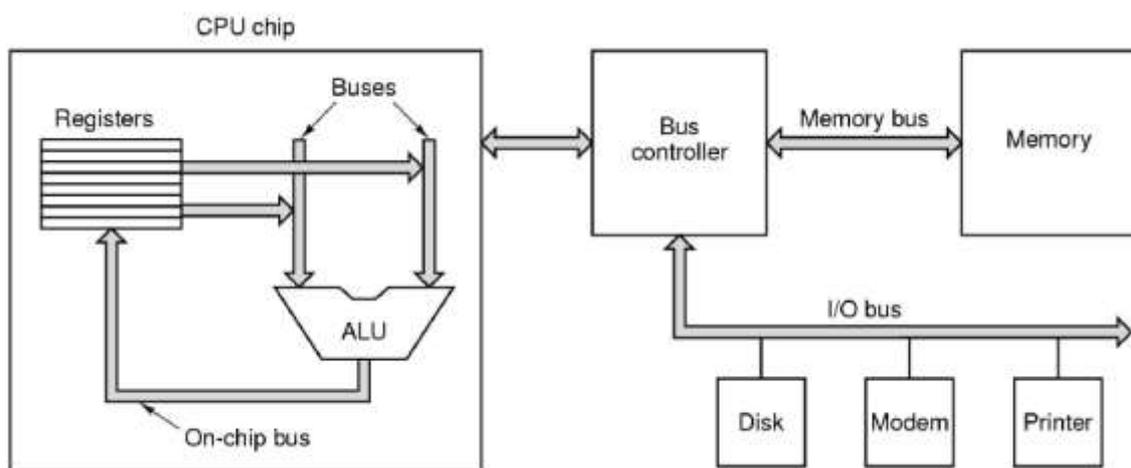


Figura 32 – Esquema simplificado de conexão

5.7.2 Barramento LPC - substituto do ISA em placas modernas

As placas de CPU modernas não utilizam mais slots ISA. Ainda assim possuem circuitos internos que precisam estar disponíveis, apesar de serem originalmente ligados ao barramento ISA. São as interfaces seriais, a interface paralela, a interface para drives de disquetes, o CMOS, a interface de teclado, a interface para mouse, o PC Speaker e o BIOS. Note que nas placas modernas, com exceção do BIOS, todos esses circuitos fazem parte do chip conhecido como Super I/O. Seria preciso manter um barramento ISA interno apenas para a ligação desses dispositivos. Outro problema é que o barramento ISA possui muitos pinos. Mesmo descontando pinos que normalmente não são necessários para o funcionamento do Super I/O e do BIOS, o número total continua sendo muito grande. Para resolver esses problemas, a Intel criou o barramento LPC (Low Pin Count).

Este barramento opera com 33 MHz, mas utiliza apenas 13 pinos. Como é destinado à conexão de dispositivos lentos, não precisa utilizar barramentos independentes de dados e endereços. Na verdade nem precisa fornecer todos esses bits ao mesmo tempo. O LPC transfere as informações no formato serial, usando um barramento de apenas 4 bits. Utilizado essas 4 linhas são fornecidos de forma serial,

dados, endereços, comandos, wait states e todas as demais informações necessárias à sua operação.

A figura 33 mostra o diagrama de uma moderna placa de CPU na qual não existe barramento ISA. Observe que o chip South Bridge (no exemplo, o AMD-766) tem como uma de suas funções, gerar os sinais do barramento LPC, no qual estão conectados os chips Super I/O e o Flash BIOS. Da mesma forma existem vários chips Super I/O e várias Flash ROMs que também adotaram o padrão LPC.

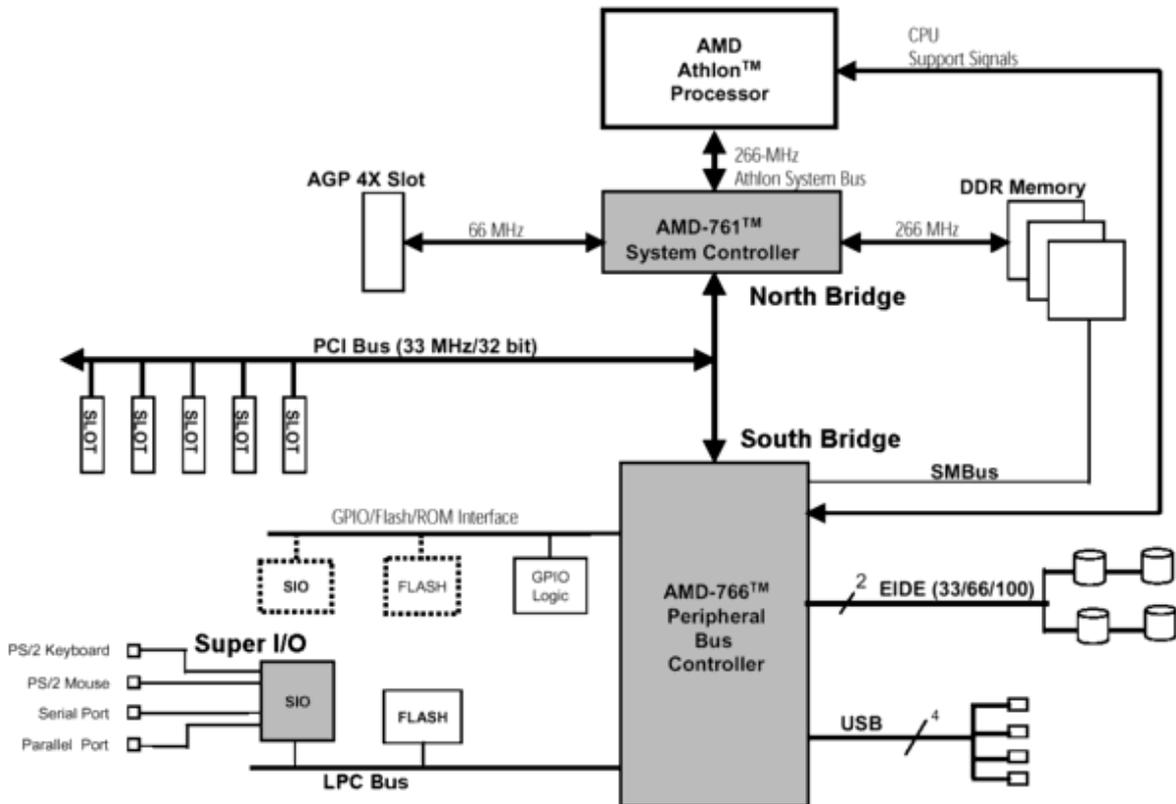


Figura 33 - Diagrama de uma placa de CPU moderna

5.8 Memória

Pode-se dizer que depois da CPU, é a parte mais importante de um computador é a memória. Tanto é assim que ouve-se muitos usuários de micros falarem: "... possuo um Pentium 4 com 1 GB de memória...".

A memória principal é aquela que é acessada diretamente pelo processador. É formada por diversos tipos de chips. Pode-se ver na figura 34 alguns tipos de chips de memória, usados para formar a memória principal de PC's.

Cada posição da memória principal tem um endereço único e a maioria das instruções de máquina refere-se a um ou mais endereços da memória principal. A memória principal geralmente é combinada com uma memória cachê menor e de maior velocidade de acesso. A memória cachê não é visível para o programador, na maioria dos casos, e nem para o processador.

Além da memória principal, que é diretamente acessada pela CPU, existe

também a memória secundária. A memória secundária não é acessada diretamente pela CPU. Seu acesso é feito através de interfaces ou controladoras especiais. Pode-se citar como exemplo de memória secundária, o disco rígido. A memória secundária não é formada por chips, e sim, por dispositivos que utilizam outras tecnologias de armazenamento. O disco rígido, assim como os disquetes e as unidades de fita, usam a tecnologia magnética para armazenar dados. Os discos CD-ROM usam tecnologia ótica.

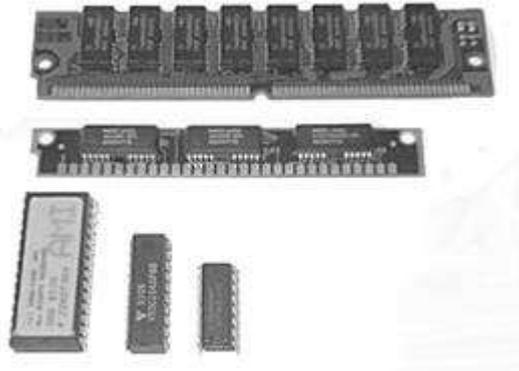


Figura 34 – modelos de memórias

No caso dos microcomputadores, quase toda a memória principal fica localizada na placa de CPU. Entretanto, algumas outras placas, chamadas de placas de expansão, também podem conter mais memória. E um bom exemplo é a placa de vídeo. Os chips de memória podem ser divididos em duas categorias:

RAM - São chips de memória (figura 34) que podem ser lidos e gravados pela CPU a qualquer instante. A CPU usa a RAM para armazenar e executar programas vindos do disco, para ler e escrever os dados que estão sendo processados. Uma outra característica da RAM, é que se trata de uma memória volátil. Isso significa que quando o computador é desligado, todos os seus dados são apagados. Por essa razão, é necessário que os programas e dados fiquem gravados no disco, que é uma memória permanente.

As memórias SDRAM (Synchronous Dynamic RAM), são capazes de trabalhar sincronizadas com os ciclos da placa-mãe, sem tempos de espera. Isso significa que a temporização das memórias SDRAM é sempre de uma leitura por ciclo. Independentemente da frequência de barramento utilizada, os tempos de acesso serão sempre de 6-1-1-1, ou mesmo 5-1-1-1. As memórias passaram a ser rotuladas de acordo com a frequência em que são capazes de operar. No caso das memórias SDRAM tem-se as memórias PC-66, PC-100 e PC-133, no caso das DDR tem PC-200, PC-266, PC-333, PC-400 (e assim por diante), enquanto nas DDR2 tem PC-533, PC-666, PC-800, PC-933, PC-1066 e PC-1200.

Por exemplo, um módulo de memória PC-133 deve ser capaz de operar a 133 MHz, fornecendo 133 milhões de leituras (teóricas) por segundo. Entretanto, essa velocidade é atingida apenas quando o módulo realiza um burst de várias leituras. O primeiro acesso continua levando 5, 6 ou mesmo 7 ciclos da placa-mãe, como nas memórias antigas.

Como sugere a lógica, as memórias DDR3 realizam 8 acessos por ciclo, contra os 4 acessos por ciclo das memórias DDR2. Assim como na tecnologia anterior, os acessos são realizados a endereços subjacentes, de forma que não existe necessidade

de aumentar a frequência "real" das células de memória. Inicialmente, os módulos DDR3 foram lançados em versão DDR3-1066 (133 MHz x 8) e DDR3-1333 (166 MHz x 8), seguidos pelo padrão DDR3-1600 (200 MHz x 8). Os três padrões são também chamados respectivamente de PC3-8500, PC3-10667 e PC3-12800, nesse caso dando ênfase à taxa de transferência teórica.

ROM - É a abreviatura de read only memory, ou seja, "memória para leitura apenas". São chips de memória que podem ser lidos pela CPU a qualquer instante, mas não podem ser gravados. Sua gravação é feita apenas pelo fabricante do computador, ou pelo fabricante de memórias. A outra característica importante de ROM é que trata-se de uma memória permanente. Seu conteúdo nunca é perdido, mesmo com o computador desligado. Nos microcomputadores, existe um programa muito importante chamado de BIOS (Basic Input-Output System - Sistema Básico de Entrada e Saída). O BIOS tem várias funções, entre as quais, a de realizar a "partida" do computador. Quando ligamos o computador, o BIOS realiza a contagem de memória, faz uma rápida checagem do funcionamento do computador e realiza a carga do Sistema Operacional que deve estar armazenado no disco. O BIOS está gravado em uma memória ROM localizada na placa de CPU. Existem tipos de ROM que podem ser gravados através de programas especiais. Por exemplo as do tipo Flash ROM. Também ficam disponíveis apenas para leitura, mas utilizando um programa especial fornecido pelo fabricante da placa de CPU, o BIOS pode ser regravado.

Como já mostrado anteriormente, a placa de CPU contém quase toda a memória de um PC, mas outras placas também podem conter memórias, do tipo RAM e do tipo ROM. Por exemplo, as placas de vídeo contém uma ROM com o seu próprio BIOS, e contém uma RAM chamada de memória de vídeo, que armazena os caracteres e gráficos que são mostrados na tela.

A memória secundária também é chamada de memória de massa. É uma memória do tipo permanente (não se apaga quando o computador está desligado), que tem uma alta capacidade de armazenamento, e um custo muito mais baixo que o da memória principal. Considere por exemplo, um computador com 1 GB de RAM (que faz parte da memória principal) e um disco rígido de 160 GB (que faz parte da memória secundária). Os 160 GB do disco rígido servem para armazenar diversos programas e diversos dados. O 1 GB de RAM servem para manter apenas os programas e os dados que estão sendo processados em um dado instante. Pode-se também mostrar como o custo da memória secundária é muito mais baixo que o da memória principal. Enquanto 1 GB de RAM custam cerca de \$80, um disco rígido de 160 GB custa cerca de \$180 (preços de 2008).

| Meio | Tempo (em segundo) |
|--------------|--------------------|
| RAM | 0,000.000.010 s |
| ROM | 0,000.000.200 s |
| Disco rígido | 0,010 s |
| Disquete | 0,100 s |
| CD-ROM | 0,200 s |
| Fita | 60 s |

Tabela 2 – Tempo de acesso dos meios de armazenamento

Os meios de armazenamento secundário são mais lentos porque envolvem movimento de suas partes mecânicas. O disco rígido (figura 35a), o drive de disquete (figura 35b) e o drive de CD-ROM (figura 35c), citados na tabela 2, precisam mover as cabeças de leitura até o ponto onde será feita a leitura. A unidade de fita (figura 35d) precisa girar a fita até o ponto a ser acessado.



Figura 35 – Disco rígido (a), Disquete (b), CD-Rom (c) e Fita Magnética (d)

Memória Flash - Existem dois tipos de memória Flash. A primeira tecnologia de memória Flash a se popularizar foi o tipo **NOR**, que chegou ao mercado em 1988. Os chips de memória Flash NOR possuem uma interface de endereços similar à da memória RAM. Assim, eles rapidamente passaram a ser usados para armazenar o BIOS da placa-mãe e firmwares em dispositivos diversos, que antes eram armazenados em chips de memória ROM ou EEPROM. Nos primeiros PCs, por exemplo, o BIOS da placa-mãe era gravado em um chip de memória ROM e por isso não era atualizável, a menos que o chip fosse fisicamente substituído. A memória Flash do tipo NOR é bastante usada até hoje em palmtops, celulares e diversos tipos de dispositivos, para armazenar o sistema operacional, nesse caso chamado de firmware, que é carregado durante o boot, sem poder ser alterado. A vantagem nesse caso é o processo XiP (execute in place), onde o sistema pode rodar diretamente a partir do chip de memória, sem precisar ser primeiro copiado para a memória RAM. O chip de memória NOR é complementado por uma pequena quantidade de memória SRAM ou DRAM, que é utilizada como memória de trabalho. E também pode ser usada para armazenar dados e configurações que podem ser perdidos quando a carga da bateria se esgotar completamente.

Além de serem mais baratas que as NOR, as memórias **NAND** também são muito mais rápidas na hora de gravar dados. A principal limitação é que elas são endereçadas usando páginas de 2 KB e acessadas através de um barramento serial. Ou seja, do ponto de vista do sistema, um cartão de memória Flash NAND está mais para um HD do que para uma unidade de memória. Você pode usá-lo para guardar dados, mas na hora que o sistema precisa rodar um programa, precisa primeiro copiá-lo para a memória RAM, da mesma forma que faria ao usar um HD.

5.9 Dispositivos de entrada e saída

Já vimos que um computador passa o tempo todo realizando três tarefas:

| | |
|----------------------|--|
| Entrada | Realizado por chips e dispositivos especializados em leitura de dados, sob a coordenação da CPU. |
| Processamento | Realizado pela CPU |
| Saída | Realizado por chips e dispositivos especializados em transmissão ou gravação de dados, sob a coordenação da CPU. |

Existem portanto no computador, os chamados dispositivos de entrada e saída, também chamados de periféricos. Através desses dispositivos, o computador pode armazenar, ler, transmitir e receber dados. A memória secundária, já estudada na seção anterior, é formada por diversos dispositivos de entrada e saída.



Figura 36 – Dispositivos de Entrada

O termo “Entrada e Saída” é abreviado por E/S, ou para quem preferir a língua inglesa, I/O (Input/Output). Dentre os diversos dispositivos de E/S, existem alguns que são especializados apenas em entrada, outros especializados apenas em saída e outros em ambos os processos. Pode-se citar os seguintes exemplos:

| | |
|--|---|
| Entrada: (figura 36) | Teclado - Lê os caracteres digitados pelo usuário Mouse - Lê os movimentos e toques de botões Drive de CD-ROM/DVD-ROM - Lê dados de CD's ou DVD's Microfone - Transmite sons para o computador Scanner - Capta figuras, imagens, textos e fotos |
| Saída: (Figura 37) | Vídeo - Mostra na tela caracteres e gráficos Impressora - Imprime caracteres e gráficos Alto-falante - Emite som |
| Entrada e saída: (figura 38) | Disco rígido - Grava e lê dados Drive de disquete - Grava e lê dados em disquetes Unidade de fita - Grava e lê em fitas magnéticas Modem - Transmite e recebe dados pela linha telefônica |



Figura 37 – Dispositivos de Saída

A CPU não pode comunicar-se diretamente com os periféricos. Esta comunicação é feita com a ajuda de circuitos chamados de interfaces. Por exemplo, os dados que são recebidos do mouse chegam até a CPU através da interface serial. Os dados a serem impressos são transmitidos até a impressora através de um circuito chamado de interface paralela. Os dados provenientes de um microfone chegam até a CPU através de uma interface, chamada de conversor analógico-digital, localizado na placa de som.



Figura 38 – Dispositivos de Entrada/Saída

7 CISC x RISC

7.1 Apresentação

RISC é abreviação de Reduce Instruction Set Computer, computador com conjunto reduzido de instruções. Já CISC significa Complex Instruction Set Computer, computador com conjunto complexo de instruções. RISC é a identificação de um tipo de arquitetura de CPU e de todo um sistema de computação que se contrapõe à arquitetura até então predominante, denominada CISC.

Durante grande parte da história dos computadores pessoais, o modelo predominante dos microprocessadores tem sido da Intel Corporation. O primeiro processador do IBM PC foi o Intel 8088. As gerações seguintes de processadores Intel foram da família 80X86 que eram versões aperfeiçoadas do 8088 original, mas com desempenho melhorado de duas maneiras – operando mais rapidamente ou manejando mais dados simultaneamente.

Mas apesar das mudanças, os processadores Intel até 80486 eram baseados em uma filosofia de projeto denominada CISC, isto é, uma grande quantidade de instruções com variedade de modos de endereçamento, poucos registradores de dados na CPU e processamento controlado por microprograma.

Desde o aparecimento das estações de trabalho SPARC, lançadas no final da década de 80 pela Sun Microsystems, dos EUA, o conceito de arquitetura RISC vem se tornando de interesse crítico para todo grande fabricante de computadores.

Os chips RISC podem ser fisicamente menores que os chips CISC. E pelo fato de usarem menos transistores, geralmente sua fabricação é mais barata e produzem menos calor.

Os primeiros computadores tinham relativamente poucas instruções e somente um ou dois modos de endereçamento. Tudo mudou com a introdução da série IBM 360, em 1964. Todos os modelos da série 360 eram microprogramados. Embora as microarquitecturas dos vários modelos 360 fossem suficientemente diretas, os microprogramas que neles eram executados apresentaram aos usuários um conjunto altamente complexo de instruções no nível de máquina convencional ou melhor dizendo, "linguagem de máquina".

Com o tempo os microprocessadores que começaram com arquitetura mínimas atingiram uma alta complexidade. Esta tendência foi encorajada pelo largo uso de linguagens de alto nível. Porém esta linguagem de alto nível precisava ser compilada para a linguagem de máquina e isto tornou difícil a construção de compiladores. Já que baixar o nível das linguagens estava fora de questão, chegou-se a conclusão que o caminho mais viável seria elevar o nível da linguagem da máquina.

Outro fator que encorajou o aumento destas máquinas CISC foi à velocidade relativamente baixa da memória principal, quando comparada à CPU. A Segunda maneira era colocar estas rotinas no microprograma e adicionar novas instruções à arquitetura, esta buscava a rotina a partir de uma ROM rápida dentro da CPU. Sob estas condições a tentação de colocar mais e mais complexidade no microprograma era irresistível.

A medida que a linguagem da máquina torna-se maior e mais complicada, seu interpretador, o microprograma, se torna maior e mais lento. Mais instruções significam mais tempo gasto na decodificação dos opcodes (código da instrução). Ainda mais importante, um grande número de modos de endereçamento significa que a análise de endereços não pode mais ser feita in-line porque o mesmo (extenso) microcódigo teria que ser repetido centenas de vezes no microprograma. Assim, são necessários microprocedimentos, e a maioria das instruções devem chamar um microprocedimento para analisar os modos de endereçamento.

Com a invenção da memória RAM semicondutora o cenário passou a mudar. Percebeu-se que computadores poderiam ser executados muito mais rapidamente jogando fora todo o interpretador, e tendo cada programa compilado diretamente para o microcódigo e executado a partir de uma rápida memória RAM semicondutora.

Em resumo, uma máquina RISC é essencialmente apenas um computador com um pequeno número de microinstruções verticais. Programas de usuários são compilados para seqüências destas microinstruções, e então executadas diretamente pelo hardware. Sem nenhum interpretador interveniente.

Antes de inventarem a microprogramação, todos os computadores eram máquinas RISC, com instruções simples executadas diretamente pelo hardware. Depois que a microprogramação tomou conta, os computadores se tornaram mais complexos e menos eficientes. Agora a indústria está voltando às suas raízes, e construindo máquinas rápidas e simples novamente.

7.2 A arquitetura RISC

Uma característica de um sistema RISC, é a sua tendência a possuir um conjunto menor de instruções que o das máquinas CISC de mesma capacidade.

Com menor quantidade de instruções e com cada uma delas tendo sua execução otimizada, o sistema deve produzir seus resultados com melhor desempenho, mesmo considerando-se que uma menor quantidade de instruções vai conduzir a programas um pouco mais longos.

Outra característica importante da arquitetura RISC, que a distingue da arquitetura CISC, refere-se ao modo de realizar chamadas de rotinas e passagem de parâmetros. Os estudos sobre comportamento dos programas revelaram que chamadas de funções (que consomem tempo de processamento) requerem usualmente poucos dados, mas consomem, na transferência, demorados acessos à memória em leituras e escritas. Nas máquinas CISC a chamada de funções conduz a operação de leitura/escrita com a memória para passagem de parâmetro e recuperação de dados, nas máquinas com a arquitetura RISC isto ocorre basicamente no processador, utilizando-se para isso mais registradores que nas máquinas CISC; os parâmetros e variáveis são manuseados na própria CPU. A possibilidade de colocação de mais registradores na CPU é possível devido à redução dos circuitos necessários à decodificação e execução de instruções.

Assim, o desempenho total do processador melhora, executando de forma otimizada as chamadas de funções que ocorrem em quantidade razoável na maioria dos programas.

Para facilitar o trabalho dos compiladores, o conjunto de instruções de máquinas

CISC tende a possuir vários modos de endereçamento. Por exemplo, uma simples instrução de soma pode ser realizada com os operandos localizados de diversos modos: podem-se somar valores que estão armazenados em registradores; outra instrução pode realizar a mesma soma, com um operando na memória e outro em um registrador, ou ainda uma outra instrução pode realizar a operação de soma com os dois operando armazenados na memória.

No caso das máquinas RISC, a busca por soluções mais simples conduziu à criação, de um modo geral, de dois tipos de instruções: LOAD/STORE para acesso à memória utilizando somente o modo direto, e demais operações matemáticas do processador. Esta técnica simplifica consideravelmente o projeto e a implementação as instruções, reduzindo ainda mais os ciclos do relógio necessários à sua realização.

Projetar processadores que executam várias instruções quase que totalmente em paralelo é uma técnica bastante eficaz para acelerar o desempenho dos processadores, reduzindo o tempo de execução das instruções para poucos ciclos. Pipelining é utilizado em larga escala em arquiteturas RISC.

Máquinas RISC tentem a executar instruções bem mais rápido porque as instruções possuem Código de Operação com menor quantidade de bits e, portanto, o tempo de decodificação é menor que o das máquinas CISC e porque as instruções são executadas diretamente pelo hardware e não por um microprograma. Conquanto um processador microprogramado traga mais flexibilidade ao projeto das máquinas, ele também acarreta uma sobrecarga adicional de interpretação de cada instrução. Máquinas RISC não são microprogramadas.

Compiladores para máquinas RISC possuem claramente alguns problemas que os compiladores convencionais não possuem. Os principais problemas são o tratamento de cargas, os armazenamentos e os saltos atrasados. O compilador RISC tem seu maior trabalho em tentar produzir um código que é tanto correto quanto eficiente. Máquinas CISC têm uma geração de código mais fácil.

7.3 Pipeline

Uma instrução simples pode ser executada em apenas um ciclo de relógio (clock), enquanto instruções mais complexas demoravam vários ciclos de relógio para serem concluídas. Para melhorar o desempenho do processador foi usado uma técnica denominada pipeline, que consiste em dividir o processador em vários estágios distintos. Por exemplo, o 486, possui um pipeline de 5 níveis, ou seja, é dividido em 5 estágios.

Quando é carregada uma nova instrução, ela primeiramente passa pelo primeiro estágio, que trabalha nela durante apenas um ciclo de relógio, passando-a adiante para o segundo estágio. A instrução continua então sendo processada sucessivamente pelo segundo, terceiro, quarto e quinto estágios do processador. A vantagem desta técnica, é que o primeiro estágio não precisa ficar esperando a instrução passar por todos os demais para carregar a próxima, e sim carregar uma nova instrução assim que se livra da primeira, ou seja, depois do primeiro pulso de relógio.

As instruções trafegam dentro do processador na ordem em que são processadas. Mesmo que a instrução já tenha sido processada ao passar pelo primeiro ou segundo estágio, terá que continuar seu caminho e passar por todos. Se por acaso a instrução não tenha sido completada mesmo após passar pelos 5, voltará para o

primeiro e será novamente processada, até que tenha sido concluída.

Os processadores atuais utilizam um número muito maior de estágios de pipeline. O Pentium II possui 10, o Athlon Thunderbird possui 12 e o Pentium 4 possui 20. No Pentium 4 por exemplo, cada estágio processa uma pequena parte da instrução, que só são concluídas pelo último estágio, a cada ciclo, cada um dos estágios passa a instrução para a frente e recebe uma nova. Cada instrução demora 20 ciclos para ser processada, mas em compensação são processadas 20 instruções ao mesmo tempo, em fila. Mais estágios permitem que o processador seja capaz de atingir frequências mais altas, já que cada estágio fará menos trabalho por ciclo, suportando mais ciclos por segundo, mas, por outro lado, o uso de muitos estágios pode prejudicar o desempenho do processador nas operações de tomada de decisão, já que cada instrução demorará mais ciclos para ser concluída.

A técnica “pipeline” realiza este paralelismo com, por exemplo, as funções de:

- Busca das instruções
- Decodificação
- Busca de dados na memória e
- Execução

Uma execução normal, sem pipeline segue a seqüência apresentada na figura 17 para a execução de instruções:

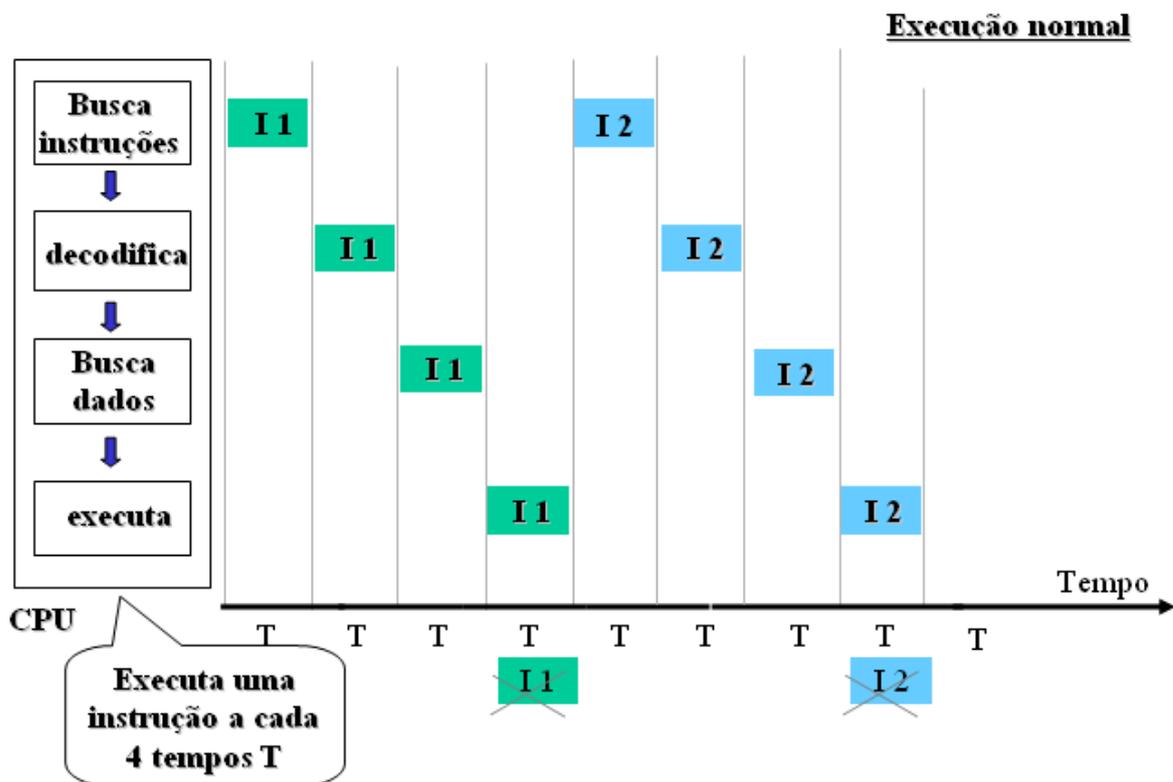


Figura 39 – Seqüência de execução sem pipeline

A execução com pipeline de seqüências de instruções podem ser observadas na figura 18, observe que o processador inicia a outra instrução antes de terminar a primeira. Este é um exemplo em quatro estágios, mas existem outras.

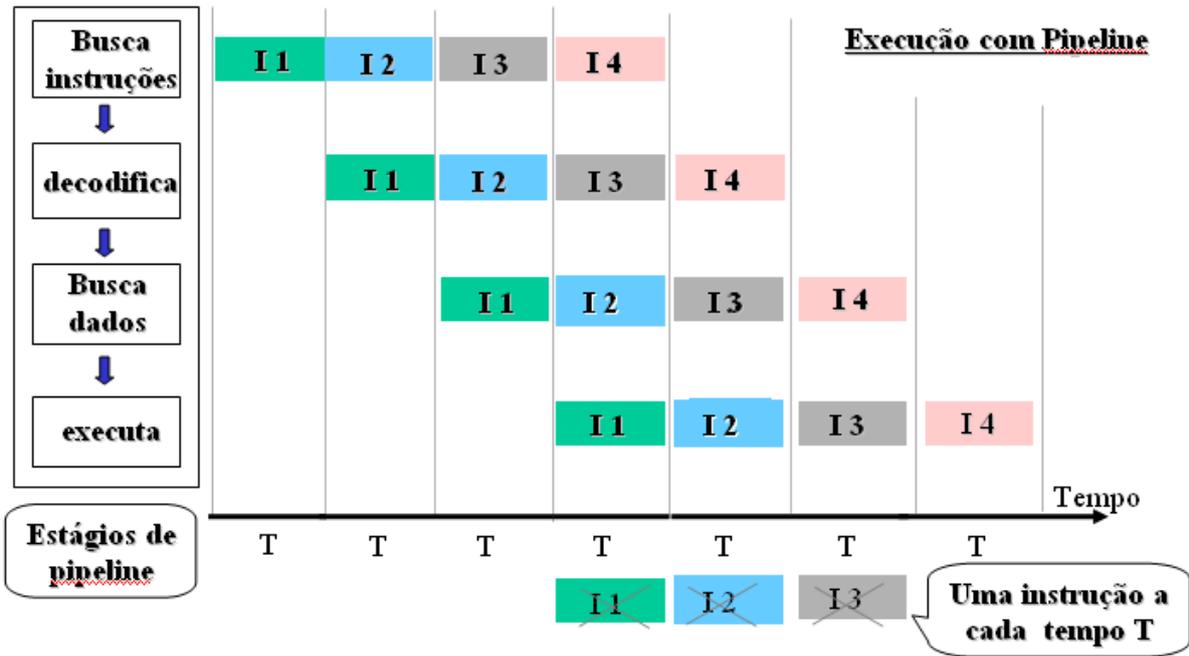


Figura 40 – Sequência de execução de instruções com pipeline

8 Instruções de Máquina

Instruções muito simples que as CPUs são capazes de decodificar, ou seja, entender o que é para ser feito e com que operandos realizar (se necessário). Cada processador tem o seu conjunto de instruções (ISA – Instruction Set Architecture). Este conjunto contém todas as instruções, em linguagem de máquina, que o processador pode executar. Uma CPU entende uma ordem de grandeza de cerca de 100 a 400 instruções de máquina.

Exemplo:

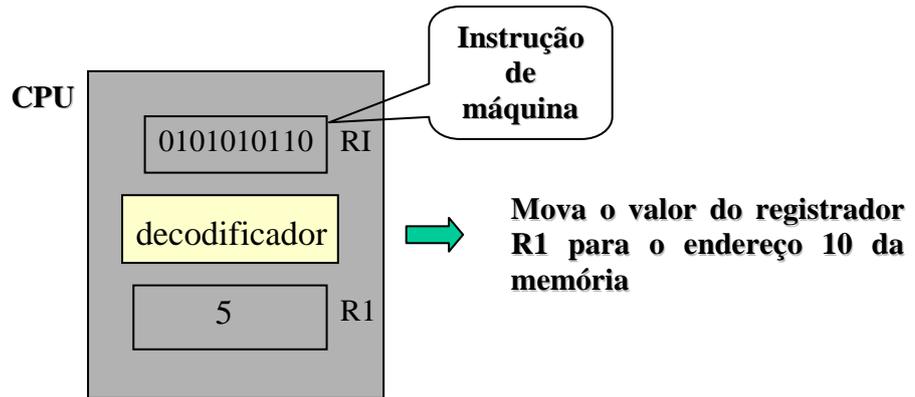


Figura 41 – Exemplo de Instrução de máquina

7.1 Tipos de Instruções

- Lógicas e Aritméticas - Ex: some o conteúdo do endereço 20 da memória com o valor do registrador R1 e coloque resultado em R1
- Movimentação de dados - Ex: mova o valor do registrador R2 para o endereço da memória 100.
- Desvios condicionais e incondicionais – Ex: a) Execute instrução do endereço 200 ($CI \leftarrow 200$) \Rightarrow Incondicional ;b) Execute instrução do endereço 200 se ACC = 0 \Rightarrow Condicional .
- Manipulação de subrotinas - Ex: Execute procedimento que está no endereço 100 e retorne.

7.2 Formato das Instruções

Toda instrução possui um código de operação. Pode possuir 0,1, 2 ou mais campos operandos. Não é comum mais de 2 operandos. Uma mesma CPU pode ter instruções de 0,1 ou 2 operandos o que significa que as instruções de uma CPU não são do mesmo tamanho.



Figura 42 – Formato genérico de uma instrução

O Código da Operação diz qual é a operação que deve ser realizada e qual é o Modo de Endereçamento dos operandos.

a) Instruções de 2 operandos:

ADD AX,BX -> Soma o conteúdo do registrador AX com o do registrador BX, o resultado permanecendo em AX.

b) instruções de 1 operando:

INC CX -> O conteúdo do registrador CX é incrementado de 1 unidade.

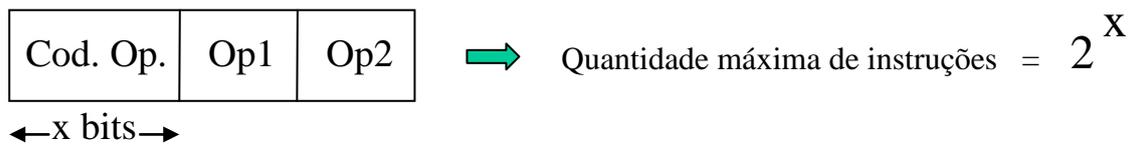
c) Instruções de 0 operandos:

HLT -> (halt) Pára completamente o processador.

O campo Código de Operação é do mesmo tamanho para todas as instruções de uma mesma CPU. Seu tamanho define a quantidade máxima de instruções que uma CPU pode ter.

Os campos de Operandos variam de tamanho dependendo do modo de endereçamento utilizado.

Exemplos:

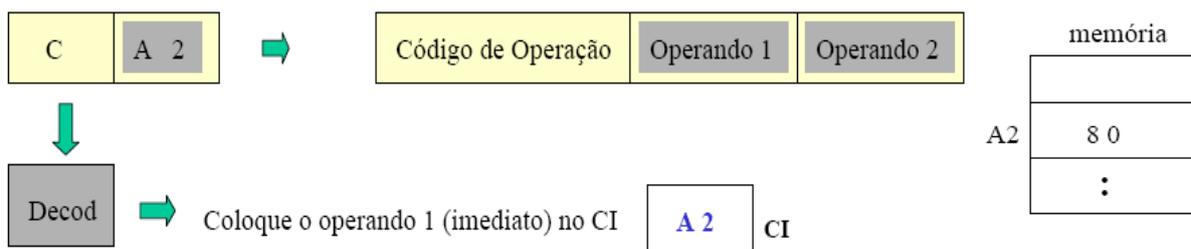


7.3 Modos de Endereçamento

É a forma em que um operando é especificado numa instrução.

7.3.1 Imediato

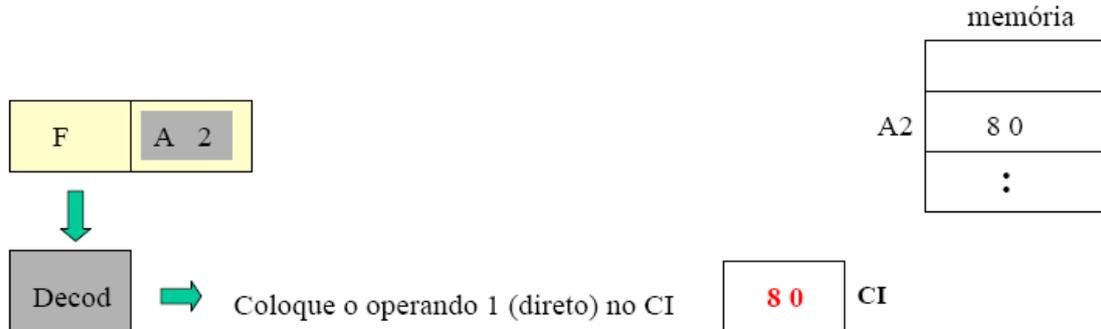
O campo operando é o próprio operando. O tamanho do campo é compatível com maior valor a ser manipulado pela instrução. Exemplo:



7.3.2 Direto

O campo operando é o endereço da memória onde está o operando. O tamanho do campo é equivalente ao maior endereço da memória que pode ser endereçado.

Exemplo:



7.3.3 Modo Registrador

O operando é um registrador da CPU.

a) Modo Registrador Indireto

O offset do endereço do operando está contido num registrador e o registrador especificado atua como ponteiro para a posição de memória onde está a informação.

Exemplo: Supondo que SI = 0100h e que a palavra contida na posição de memória de offset 0100h seja 1234h:

MOV AX,SI ;AX recebe 0100h

MOV AX,[SI] ;AX recebe 1234h

b) Modo por Base (Based Mode)

O offset do endereço do operando é obtido adicionando um deslocamento ao conteúdo de um registrador base. O deslocamento pode ser:

- o offset de endereço de uma variável uma constante (positiva ou negativa)
- o offset de endereço de uma variável mais ou menos uma constante

Exemplo: Supondo que BX contendo o valor 4d:

LISTA DW 10,20,30,40,50,60,70,80,90,100

MOV AX, [LISTA + BX] ;resulta que AX = 30

c) Modo Indexado (Indexed Mode)

O offset do endereço do operando é obtido adicionando um deslocamento ao conteúdo de um registrador indexador. As opções de deslocamento são as mesmas do Modo por Base.

Exemplo: Supondo que SI contenha o offset de endereço de LISTA:

LISTA DW 10,20,30,40,50,60,70,80,90,100

LEA SI,LISTA ;SI recebe o offset de LISTA

MOV AX, [SI + 12] ;resulta que AX = 70