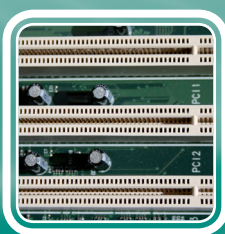
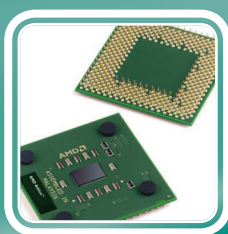


Organização e Arquitetura de Computadores

Eliane Maria de Bortoli Fávero

Curso Técnico em Informática





e-Tec Brasil
Escola Técnica Aberta do Brasil

Organização e Arquitetura de Computadores

Eliane Maria de Bortoli Fávero

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Pato Branco-PR
2011

Presidência da República Federativa do Brasil

Ministério da Educação

Secretaria de Educação a Distância

© Universidade Tecnológica Federal do Paraná

Este Caderno foi elaborado em parceria entre a Universidade Tecnológica Federal do Paraná e a Universidade Federal de Santa Catarina para o Sistema Escola Técnica Aberta do Brasil – e-Tec Brasil.

Equipe de Elaboração

Universidade Tecnológica Federal do Paraná – UTFPR

Coordenação do Curso

Eliane Maria de Bortoli Fávero/UTFPR

Professora-autora

Eliane Maria de Bortoli Fávero/UTFPR

Comissão de Acompanhamento e Validação

Universidade Federal de Santa Catarina – UFSC

Coordenação Institucional

Araci Hack Catapan/UFSC

Coordenação do Projeto

Sílvia Modesto Nassar/UFSC

Coordenação de Design Instrucional

Beatriz Helena Dal Molin/UNIOESTE e UFSC

Coordenação de Design Gráfico

Carlos Antonio Ramirez Righi/UFSC

Design Instrucional

Mariano Castro Neto/UFSC

Web Master

Rafaela Lunardi Comarella/UFSC

Web Design

Beatriz Wilges/UFSC

Gustavo Mateus/UFSC

Mônica Nassar Machuca/UFSC

Diagramação

Andréia Takeuchi/UFSC

Caroline Ferreira da Silva/UFSC

Guilherme Ataíde Costa/UFSC

Juliana Tonietto/UFSC

Revisão

Júlio César Ramos/UFSC

Projeto Gráfico

e-Tec/MEC

Catálogo na fonte pela Biblioteca Universitária da UFSC

F273o

Fávero, Eliane Maria de Bortoli

Organização e arquitetura de computadores / Eliane de Bortoli Fávero. – Pato Branco : Universidade Tecnológica Federal do Paraná, 2011.

114p. : il. ; tabs.

Inclui bibliografia

ISBN: 978-85-7014-082-1

1. Computação – Estudo e ensino. 2. Arquitetura de computadores. 4. Ensino a distância. I. Título. II.

CDU: 681.31:519.687.4

Apresentação e-Tec Brasil

Prezado estudante,

Bem-vindo ao e-Tec Brasil!

Você faz parte de uma rede nacional pública de ensino, a Escola Técnica Aberta do Brasil, instituída pelo Decreto nº 6.301, de 12 de dezembro 2007, com o objetivo de democratizar o acesso ao ensino técnico público, na modalidade a distância. O programa é resultado de uma parceria entre o Ministério da Educação, por meio das Secretarias de Educação a Distância (SEED) e de Educação Profissional e Tecnológica (SETEC), as universidades e escolas técnicas estaduais e federais.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade, e promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

O e-Tec Brasil leva os cursos técnicos a locais distantes das instituições de ensino e para a periferia das grandes cidades, incentivando os jovens a concluir o ensino médio. Os cursos são ofertados pelas instituições públicas de ensino e o atendimento ao estudante é realizado em escolas-polo integrantes das redes públicas municipais e estaduais.

O Ministério da Educação, as instituições públicas de ensino técnico, seus servidores técnicos e professores acreditam que uma educação profissional qualificada – integradora do ensino médio e educação técnica, – é capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação
Janeiro de 2010

Nosso contato
etecbrasil@mec.gov.br

Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



Atenção: indica pontos de maior relevância no texto.



Saiba mais: oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais, ambiente AVEA e outras.



Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.

Sumário

Palavra da professora-autora	9
Apresentação da disciplina	11
Projeto instrucional	13
Aula 1 - Evolução da arquitetura de computadores	15
1.1 Elementos do sistema computacional	15
1.2 Computadores analógicos x digitais	16
1.3 Evolução tecnológica	17
1.4 Componentes básicos de um sistema computacional	25
Aula 2 – Sistemas de numeração	29
2.1 Bases e sistemas de numeração	29
Aula 3 – Portas lógicas e suas funções	35
3.1 Funções e portas lógicas	35
Aula 4 - Subsistema de memória	45
4.1 Sistema de memória e suas características	45
4.2 Registradores	47
4.3 Memória <i>cache</i>	48
4.4 Memória principal	49
4.5 Memória secundária	55
Aula 5 – O Processador: organização e arquitetura	57
5.1 Organização do processador	57
5.2 Unidade funcional de processamento	59
5.3 Unidade funcional de controle	62
5.4 Instruções de máquina	73
5.5 Arquiteturas RISC e CISC	78
Aula 6 – Representação de dados	85
6.1 Introdução	85
6.2 Formas de representação	86
6.3 Tipos de dados	86

Aula 7 - Dispositivos de entrada e saída	95
7.1 Introdução a dispositivos de entrada e saída.....	95

Palavra da professora-autora

Caro estudante!

Este caderno foi elaborado com o cuidado para ajudá-lo a realizar seus estudos de forma autônoma, objetivando que você desenvolva conhecimento adequado, apresentando plenas condições de dar continuidade à realização do curso do qual esta disciplina faz parte.

Destacamos que o seu aprendizado não será construído apenas a partir do estudo dos textos apresentados neste caderno. Essa prática deve ser combinada com sua participação efetiva nas aulas, realização das atividades de aprendizagem, leituras complementares e, principalmente, com a sua interação com o professor e os demais estudantes desta disciplina. Para que possamos alcançar os objetivos propostos, a dinâmica da disciplina propõe sua efetiva participação e comprometimento nas atividades de aprendizagem e na utilização das ferramentas de informação e comunicação (síncronas e assíncronas), tais como: fóruns de discussão, *chat* e outras ferramentas disponibilizadas pelo Ambiente Virtual de Ensino e Aprendizagem (AVEA) Moodle. Alertamos que para um aproveitamento considerado ideal nesta disciplina, é imprescindível a leitura dos textos das aulas e uma efetiva utilização dos vários recursos disponíveis no Moodle. Pois além das ferramentas de informação e comunicação, o Moodle também disponibiliza materiais complementares, como por exemplo vídeos, artigos, textos, páginas WEB, repositórios de objetos de aprendizagem, entre outros materiais relacionados aos conteúdos discutidos nas nossas aulas.

Desejamos um bom início nesta disciplina e um excelente aproveitamento deste caderno e de todos os recursos disponibilizados.

Um abraço e sucesso nos seus estudos!

Eliane de Bortoli Fávero

Apresentação da disciplina

A disciplina Organização e Arquitetura de Computadores visa apresentar e discutir alguns conceitos e princípios básicos que envolvem a organização interna de um sistema computacional (computador), seus componentes e interconexões, a partir de uma visão crítica quanto à sua estrutura e desempenho. Propõe o reconhecimento e análise das arquiteturas dos processadores, memórias e dispositivos de entrada e saída, bem como o entendimento do funcionamento da arquitetura quanto à execução de programas.

Nesse sentido o foco desta disciplina é o modo como ocorre a organização interna dos componentes de um computador (ex.: processador, memória, dispositivos de E/S), no que tange à tecnologia utilizada, suas características e como ocorre a comunicação entre esses componentes.

Para melhor compreensão dos textos, este caderno apresenta sete aulas organizadas da seguinte forma:

Aula 1: Evolução da arquitetura de computadores – Nesta aula serão apresentados e discutidos alguns componentes do sistema computacional, diferenças entre o computador analógico e o digital, através das várias gerações de computadores, com ênfase nas tecnologias empregadas no modelo de Von Neumann e nos componentes básicos de um computador.

Aula 2: Sistemas de numeração – Esta aula visa apresentar o sistema de numeração utilizado pelos sistemas computacionais, dando ênfase nos sistemas: binário, octal e hexadecimal.

Aula 3: Portas lógicas e suas funções – O objetivo desta aula é apresentar os elementos básicos de organização de computadores: portas lógicas e circuitos combinacionais, visando à compreensão do que há no interior de um circuito integrado (CI).

Aula 4: Subsistema de memória – Esta aula objetiva apresentar e discutir a organização básica de memória de um computador, com ênfase nas características que diferenciam os diferentes tipos de memória.

Aula 5: Organização e arquitetura do processador – Nesta aula vamos compreender como ocorre a organização básica do processador e a organização de microprocessadores atuais, seu funcionamento interno, e como ocorre a execução de instruções em baixo nível.

Aula 6: Representação de dados – Nesta aula vamos aprender como ocorre a representação de dados no formato interno dos sistemas computacionais, considerando os diversos tipos de dados.

Aula 7: Dispositivos de entrada e saída – Esta aula apresenta e discute algumas características do funcionamento básico dos dispositivos de entrada e saída disponíveis em um computador.

Bons estudos!

Projeto instrucional

Disciplina: Organização e Arquitetura de Computadores (carga horária: 60h).

Ementa: Conceitos de arquitetura e organização dos computadores. Componentes internos: processadores, memórias e dispositivos auxiliares. Interconexão dos componentes. Circuitos básicos. Elementos funcionais e dispositivos. Instruções de baixo nível. Arquiteturas avançadas de computadores.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Evolução da arquitetura de computadores	Entender a evolução da arquitetura de computadores e seu modo de funcionamento. Identificar os componentes do computador por meio da introdução do modelo de Von Neumann.	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Texto: Nanotecnologia. Disponível em: http://pt.wikipedia.org/wiki/Nanotecnologia. Acesso em: 19 jul. 2010; Texto: Computador óptico – futuro dos computadores. Disponível em: http://www.lucalm.hpg.ig.com.br/mat_esp/comp_optico/computador_optico.htm. Acesso em: 19 jul. 2010; Texto: Computador quântico já funciona. Disponível em: http://info.abril.com.br/aberto/info-news/022007/15022007-3.shl. Acesso em: 19 jul. 2010; Texto: Simulador de computador quântico. Disponível em: http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=010150040809. Acesso em: 19 jul. 2010; Vídeo: Como os chips são fabricados, disponível em: http://www.clubedohardware.com.br/artigos/1131. Acesso em: 19 jul. 2010; Vídeo: Como os computadores funcionam, disponível em: http://www.youtube.com/watch?v=Wf1jnh8TCXA. Acesso em: 19 jul. 2010; Artigo disponível em: http://www.guiadohardware.net/artigos/evolucao-processadores. Acesso em: 19 jul. 2010.	8
2. Sistemas de numeração	Compreender o sistema de numeração utilizado pelos sistemas computacionais.	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Vídeo: Números binários. Disponível em: http://www.youtube.com/watch?v=TJRYW-ISocU. Acesso em: 19 jul. 2010. Vídeo: Sistemas de representação e conversão entre bases. Disponível em: http://www.youtube.com/watch?v=RQfit_s7Afg&feature=related. Acesso em: 19 jul. 2010.	8
continua			

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
3. Portas lógicas e suas funções	Identificar os elementos básicos de organização: portas lógicas e circuitos combinacionais. Entender o funcionamento de um circuito integrado (CI).	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Artigo: Introdução às portas lógicas. Disponível em: http://www.clubedohardware.com.br/artigos/1139 . Acesso em: 19 jul. 2010; Vídeo: "Portas lógicas AND, OR, XOR, NOT". Disponível em: http://www.youtube.com/watch?v=4ENGYy68JqM&feature=related . Acesso em: 19 jul. 2010.	8
4. Subsistema de memória	Analisar e compreender a organização básica de memória de um computador.	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Artigo: Memórias, disponível em: http://www.gdhpress.com.br/hardware/leia/index.php?p=cap4-1 . Acesso em: 19 jul. 2010.	10
5. O Processador – organização e arquitetura	Analisar a organização básica do processador. Analisar e compreender a organização de microprocessadores atuais. Compreender o funcionamento interno e a execução de instruções em baixo nível.	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Texto: Barramentos: ISA, AGP, PCI, PCI Express, AMR e outros. Disponível em: http://www.infowester.com/barramentos.php . Acesso em: 19 jul. 2010; Texto: Tecnologia USB (Universal Serial Bus). Disponível em: http://www.infowester.com/usb.php . Acesso em: 19 jul. 2010; Texto: Tecnologia PCI Express. Disponível em: http://www.infowester.com/pciexpress.php . Acesso em: 19 jul. 2010; Texto: Esquema geral de funcionamento do processador. Disponível em: http://sca.unioeste-foz.br/~grupob2/00/esq-geral.htm . Acesso em: 19 jul. 2010; Texto: Processadores RISC x processadores CISC. Disponível em: http://www.guiadohardware.net/artigos/risc-cisc . Acesso em: 19 jun. 2010.	12
6. Representação de dados	Conhecer a representação de dados no formato interno dos sistemas computacionais.	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Texto: Representação de dados. Disponível em: http://www.cristiancechinell.pro.br/my_files/algorithms/bookhtml/node23.html . Acesso em: 19 jul. 2010.	8
7. Dispositivos de entrada e saída	Conhecer as características e compreender o funcionamento básico dos dispositivos de entrada e saída acoplados ao sistema computacional.	Ambiente Virtual de Ensino e Aprendizagem (AVEA); Texto: Como funciona o LCD. Disponível em: http://www.gdhpress.com.br/hmc/leia/index.php?p=cap7-3 . Acesso em: 19 jul. 2010; Texto: Monitores de vídeo. Disponível em: http://www.clubedohardware.com.br/artigos/Monitores-de-Video/920/1 . Acesso em: 19 jul. 2010; Texto: Introdução às impressoras matriciais, jato de tinta e Laser. Disponível em: http://www.infowester.com/impressoras.php . Acesso em: 19 jul. 2010; Texto: Anatomia de um disco rígido. Disponível em: http://www.clubedohardware.com.br/artigos/Anatomia-de-um-Disco-Rigido/1056/5 . Acesso em: 19 jul. 2010; Texto: Conhecendo o disco rígido (HD). Disponível em: http://www.infowester.com/hds1.php . Acesso em: 19 jul. 2010; Texto: Mouses: funcionamento, tipos e principais características. Disponível em: http://www.infowester.com/mouse.php . Acesso em: 19 jul. 2010.	6

Aula 1 - Evolução da arquitetura de computadores

Objetivos

Entender a evolução da arquitetura de computadores e seu modo de funcionamento.

Identificar os componentes do computador por meio da introdução do modelo de Von – Neumann.

1.1 Elementos do sistema computacional

O computador é uma máquina ou **dispositivo** capaz de executar uma sequência de **instruções** definidas pelo homem para gerar um determinado resultado, o qual atenda a uma necessidade específica (ex.: realizar cálculos, gerar relatórios). Essa sequência de instruções é denominada algoritmo, o qual pode ser definido como um conjunto de regras expressas por uma sequência lógica finita de instruções, que ao serem executadas pelo computador, resolvem um problema específico. Assim, podemos dizer que um ou mais algoritmos compõem o que conhecemos como programa de computador, que no âmbito profissional da área de informática é conhecido como software.

As partes físicas de um computador, tais como: dispositivos de entrada e saída (ex.: monitor, teclado, impressora, *webcam*), dispositivos de armazenamento (ex. memória volátil e permanente), processador, assim como todo o conjunto de elementos que compõem um computador são chamados de *hardware*. A Figura 1.1 apresenta os elementos que compõem o *hardware*.



Figura. 1.1: Elementos de *hardware*.

Fonte: Elaborada pela autora.

A-Z

Dispositivo

mecanismo usado para obter um resultado

Instruções

ordens ou comandos para que um computador execute uma solicitação do homem (usuário), a fim de atender a uma determinada necessidade

Dessa forma, pode-se dizer que a combinação do *hardware* e do *software* forma o *sistema computacional*. A disciplina Organização e Arquitetura de Computadores enfatiza o estudo dos componentes de *hardware* de um computador.

1.2 Computadores analógicos x digitais

Os computadores podem ser classificados em dois tipos principais: analógicos e digitais.

Os computadores analógicos não trabalham com números nem com símbolos que representam os números; eles procuram fazer analogia entre quantidades (ex. pesos, quantidade de elementos, níveis de tensão, pressões hidráulicas). Alguns exemplos desse tipo de computador são o Ábaco – que se utilizava de pequenos carretéis embutidos em um pequeno filete de metal para realizar cálculos – ou a régua de cálculo – que utiliza comprimentos de escalas especialmente calibradas para facilitar a multiplicação, a divisão e outras funções.

Pode-se dizer que o computador analógico é uma categoria de computadores que se utiliza de eventos elétricos, mecânicos ou hidráulicos para resolver problemas do homem. Ou seja, tais computadores representam o comportamento de um sistema real utilizando-se para isso de grandezas físicas.

Segundo Computador... (2010), computadores analógicos são normalmente criados para uma finalidade específica, assim como ocorre com a construção de circuitos eletrônicos que implementam sistemas de controle (ex.: sistemas de segurança, sistemas de controle de nível). Nesses sistemas, os resultados da computação analógica são utilizados dentro do próprio sistema. Assim, uma pessoa era responsável pela programação e funcionamento desses computadores analógicos, realizando a programação diretamente no *hardware* (ex.: engrenagens, roldanas). No início do século XX as primeiras calculadoras mecânicas, caixas registradoras e máquinas de cálculo em geral foram redesenhadas para utilizar motores elétricos, com a posição das engrenagens representando o estado de uma variável. Exemplos de variáveis utilizadas em computadores analógicos são: a intensidade de uma corrente elétrica em um resistor, o ângulo de giro de uma engrenagem, o nível de água em um recipiente.

Diferentemente dos computadores analógicos, que representam números por meio da analogia direta entre quantidades, os computadores digitais

resolvem problemas realizando operações diretamente com números, enquanto os analógicos medem. Os computadores digitais resolvem os problemas realizando cálculos e tratando cada número, dígito por dígito. De acordo com Monteiro (2007), um computador digital é uma máquina projetada para armazenar e manipular informações representadas apenas por algarismos ou **dígitos**, que só podem assumir dois valores distintos, 0 e 1, razão pela qual é denominado de computador digital.

Outra grande diferença dessa categoria de computadores é que eles podem resolver problemas por meio de uma sequência programada de instruções com o mínimo de intervenção humana.

Assim, podemos dizer que o computador digital surgiu como uma solução rápida e com um nível de automação bem mais elevado de realizar grandes computações numéricas. Muitas são as necessidades do homem em termos de computação, especialmente nas áreas de engenharia, além de demonstrações e aplicações teóricas (ex.: cálculo de um fatorial, progressões aritméticas). Sem o uso da tecnologia, muitos cálculos manuais se tornavam inviáveis, tanto pelo custo em termos de esforço quanto pelo risco de gerar resultados incorretos.

Dessa forma, os computadores digitais foram um passo determinante para o progresso que é possível perceber atualmente em termos de computação. O sonho do homem em realizar cálculos de forma automática, fazendo do computador um dispositivo semelhante ao cérebro humano, mas com capacidades infinitamente maiores do que o ser humano poderia suportar, virou realidade e permite hoje automatizar grande parte das tarefas do ser humano, facilitando sua vida pessoal e profissional.

1.3 Evolução tecnológica

Como foi possível perceber na seção 1.2, houve uma grande evolução desde o surgimento do computador analógico até o desenvolvimento do computador digital. O que impactou nessa evolução foram as **tecnologias** utilizadas na construção de tais computadores, pois, no decorrer dos anos, foram sendo descobertos novos conhecimentos, materiais e dispositivos os quais permitiram a substituição de tecnologias antigas de processamento de informações por novas tecnologias mais eficientes em termos de computação.

A-Z

Dígitos

Dígitos são símbolos usados na representação numérica inteira ou fracionária



Realize uma pesquisa, e elabore em uma mídia digital (arquivo de texto ou apresentação), sobre alguns dos computadores analógicos que você encontrar. Procure ilustrar e comentar sua apresentação. Esse arquivo deverá ser postado no AVEA como atividade.

A-Z

Tecnologia

Tecnologia: é tudo o que o homem inventou para facilitar a resolução de seus problemas, incluindo a realização de tarefas. Portanto, exemplos de tecnologia podem variar de acordo com o contexto em que se aplica (ex.: a descoberta da fotografia, os óculos, que resolveram o problema dos deficientes visuais, o computador). Assim, tecnologia pode ser definida com técnica, conhecimento, método, materiais, ferramentas e processos usados para resolver problemas ou facilitar sua solução.

Com o surgimento dos primeiros computadores, foi possível classificá-los em gerações, de acordo com as tecnologias utilizadas para sua fabricação. A seguir apresentamos as tecnologias utilizadas em cada geração.

1.3.1 Geração zero: computadores mecânicos (1642-1945)

Essa geração foi caracterizada pelos computadores essencialmente analógicos, conforme descritos na seção 1.2, os quais eram construídos a partir de engrenagens mecânicas e eletromecânicas, operavam em baixa velocidade e eram destinados a resolver problemas específicos. São exemplos dessa geração, além dos já citados anteriormente, o mecanismo de Antikythera, a máquina de Pascal e a máquina das diferenças de Babbage.

1.3.2 Primeira geração: válvulas (1945-1955)

Podemos dizer que a Segunda Guerra Mundial foi o marco do surgimento da computação moderna. Foi nesse contexto que começaram a surgir novas tecnologias mais modernas capazes de substituir os componentes mecânicos utilizados até então nos computadores analógicos, possibilitando o surgimento dos computadores digitais. Esse foi um dos motivos pelos quais os computadores da época ficaram conhecidos como computadores de “primeira geração”.

A-Z

Relé

Relé: é um interruptor acionado eletricamente.

Capacitor

Capacitor: é um dispositivo que permite armazenar cargas elétricas na forma de um campo eletrostático e mantê-las durante certo período, mesmo que a alimentação elétrica seja cortada.

Alguns dos componentes utilizados na fabricação desses computadores eram os **relés**, os **capacitores** e as válvulas, sendo as últimas o mais importante deles. As válvulas possibilitaram cálculos milhares de vezes mais rápidos do que os efetuados com os relés eletromecânicos utilizados inicialmente. A Figura 1.2 apresenta uma válvula típica.

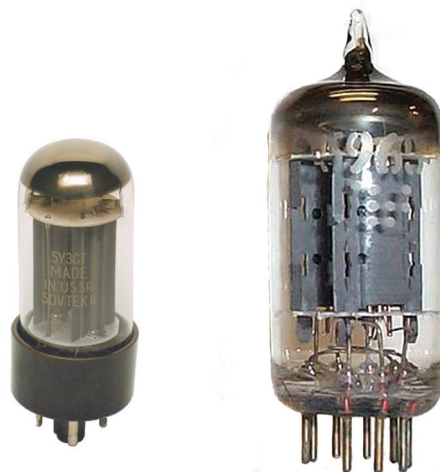


Figura. 1.2: Válvula eletrônica

Fonte: <http://valvestate.fateback.com> e http://www.pt.wikipedia.org/wiki/História_do_hardware

A entrada de dados e instruções nesses computadores, bem como a sua memória temporária, ocorria frequentemente pela utilização de cartões perfurados. Como os computadores tinham seu funcionamento baseado em válvulas (cuja função básica era controlar o fluxo da corrente, amplificando a tensão que recebe de entrada), normalmente quebravam após algum tempo de uso contínuo em função da queima delas, o que resultava em uma falta de confiabilidade, principalmente nos resultados finais. Além de ocupar muito espaço, seu processamento era lento e o consumo de energia elevado.

Dentre as primeiras calculadoras e os primeiros computadores (eletrônicos) a utilizarem válvulas, destacamos:

- a)** ENIAC, na Universidade da Pennsylvania;
- b)** IBM 603, 604, 701 e SSEC;
- c)** EDSAC, na Universidade de Cambridge;
- d)** UNIVAC I, de Eckert e Mauchly.

Listamos algumas características do *Electronic Numerical Integrator and Computer* (ENIAC), destacando o efeito do uso de válvulas na construção de computadores (Figura 1.3):

- a)** levou tres anos para ser construído;
- b)** funcionava com aproximadamente 19.000 válvulas;
- c)** consumia 200 quilowatts;
- d)** pesava 30 toneladas;
- e)** tinha altura de 5,5m;
- f)** seu comprimento era de 25 m;
- g)** tinha o tamanho de 150 m².

É possível imaginar a quantidade de energia consumida e o calor produzido por quase 19.000 válvulas?! A finalidade do ENIAC era o cálculo de tabelas de balística para o exército americano. Tratava-se de uma máquina decimal, ou seja, não binária (baseada em 0's e 1's) e sua programação envolvia a configuração de diversos cabos e chaves (como é possível observar na Figura 1.3), podendo levar vários dias.



Curiosidade: Como não se tinha confiança nos resultados, devido à constante queima de válvulas, cada cálculo era efetuado por três circuitos diferentes e os resultados comparados; se dois deles coincidisse, aquele era considerado o resultado certo.

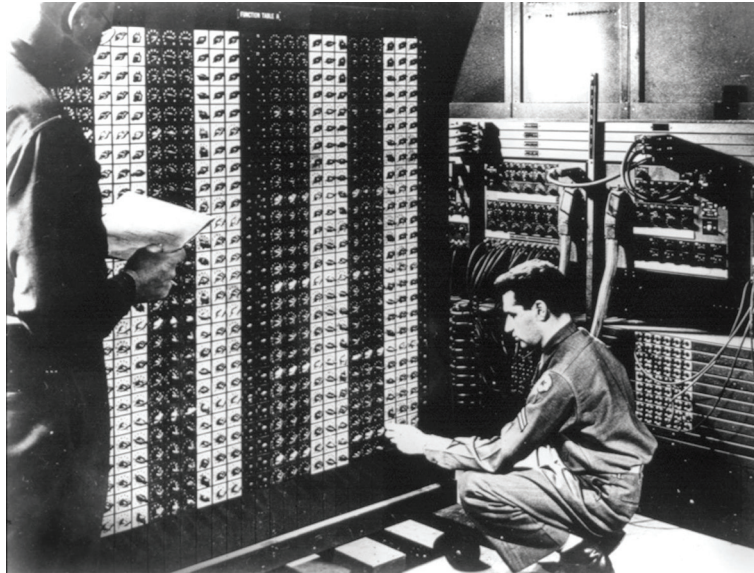


Figura 1.3: ENIAC

Fonte: www.techclube.com.br/blog/?p=218

1.3.2.1 Modelo de Von Neumann

John Von Neumann foi um matemático natural da Hungria que viveu a maior parte de sua vida nos Estados Unidos. Contribuiu de forma significativa para a evolução dos computadores. Suas contribuições perduram até os dias atuais, sendo que a principal delas foi a construção de um computador sequencial binário de programa armazenado. Podemos dizer que ele propôs os elementos críticos de um sistema computacional, denominado de Modelo de Von Neumann. A arquitetura de computador proposta por esse modelo é composta basicamente por (TANENBAUM, 2007):

- a) uma memória física (para armazenar programas e dados – representados por 0's e 1's);
- b) uma Unidade Aritmética e Lógica (ULA), cuja função é executar operações indicadas pelas instruções de um programa. Seu trabalho é apoiado por diversos registradores (ex.: acumulador);
- c) uma Unidade de Controle (UC), cuja função é buscar um programa na memória, instrução por instrução, e executá-lo sobre os dados de entrada (que também se encontram na memória); e
- d) equipamento de entrada e saída.

É importante esclarecer que a ULA e a UC, juntamente com diversos registradores específicos, formam a Unidade Central de Processamento (CPU) do computador.

A Figura 1.4 apresenta os componentes da arquitetura de Von Neumann descritos acima:

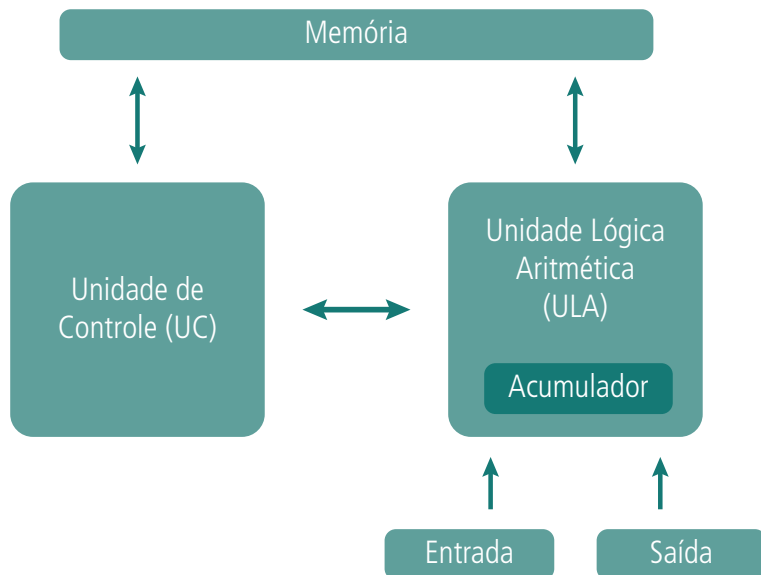


Figura. 1.4: Arquitetura de Von Neumann

Fonte: Adaptada de Tanenbaum (2007)

Destacamos que a proposta inicial de Von Neumann ainda vem sendo utilizada, mas não em seu formato original, pois muitas melhorias foram realizadas visando obter uma máquina com desempenho cada vez mais elevado, como é o caso das arquiteturas paralelas, que replicam alguns elementos da arquitetura básica de Neumann. Atualmente, muitos pesquisadores têm desenvolvido estudos visando obter uma alternativa a esse padrão, mas ainda não obtiveram sucesso.

1.3.3 Segunda geração: transistores (1955-1965)

Nessa geração, a válvula foi substituída pelo transistor, o qual passou a ser um componente básico na construção de computadores. O transistor foi desenvolvido pelo *Bell Telephones Laboratories* em 1948.

Esse dispositivo reduziu de forma significativa o volume dos computadores e aumentou a sua capacidade de armazenamento. Além disso, o transistor apresentava aquecimento mínimo, baixo consumo de energia e era mais confiável que as válvulas (que queimavam com facilidade). Para você ter uma ideia, um transistor apresentava apenas 1/200 (0,005) do tamanho de uma das primeiras válvulas e consumia menos de 1/100 (0,01) da sua energia.

A-Z

Interruptor

Interruptor: também chamado de chave, é um mecanismo que serve para interromper ou iniciar um circuito elétrico.

Operações lógicas

Operações lógicas: são funções cujos resultados consideram as condições de "verdadeiro" ou "falso", o que em termos binários podem ser representados por "1" e "0" (zero) respectivamente.

A função básica do transistor em circuitos componentes de um computador é a de um **interruptor** eletrônico para executar **operações lógicas**. Existem diversos modelos de transistores, os quais podem possuir características diferenciadas de acordo com a sua aplicação. A Figura 1.5 apresenta as características físicas de um transistor convencional.

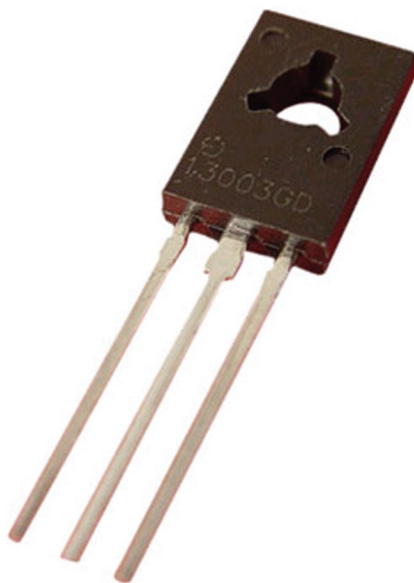


Figura. 1.5: Transistor

Fonte: www.germes-online.com/catalog/87/128/page2/



Curiosidade: Em 1954 a Texas Instruments iniciou a produção comercial. Em 1955, a Bell lançou o primeiro computador transistorizado, o TRADIC, que possuía 800 transistores.

Os materiais utilizados na fabricação do transistor são principalmente: o silício (Si), o germânio (Ge), o gálio (Ga) e alguns óxidos.

1.3.4 Terceira geração: circuitos integrados (1965-1980)

É a partir dessa geração que surgem os primeiros circuitos integrados (CI): dispositivos que incorporam inúmeros transistores e outros componentes eletrônicos em formato de miniaturas em um único **encapsulamento**. Portanto, cada chip é equivalente a inúmeros transistores. Essa tecnologia substituiu os transistores, os quais apresentam as seguintes vantagens: maior confiabilidade (não possui partes móveis); muito menores (equipamento mais compacto e mais rápido pela proximidade dos circuitos); baixo consumo de energia (miniaturização dos componentes) e custo de fabricação muito menor. Dessa forma, os computadores passaram a tornar-se mais acessíveis.

A-Z

Encapsulamento

Encapsulamento: é a inclusão de um objeto dentro de outro, de forma que o objeto incluído não fique visível. Também chamado de chip ou pastilha.

A Figura 1.6 apresenta um circuito integrado.

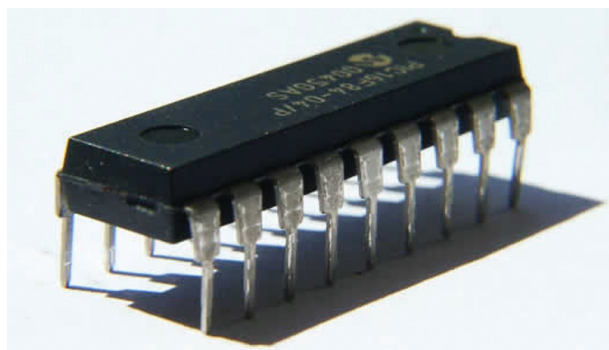


Figura 1.6: Circuito integrado

Fonte: www.bpiropo.com.br/fpc20051107.htm

Diferentemente dos computadores das gerações anteriores, a entrada de dados e instruções passaram a ser realizadas por dispositivos de entrada e saída, tais como teclados e monitores. A velocidade do processamento era da ordem de microssegundos.

Um dos computadores considerados precursor dessa geração foi o IBM 360, o qual era capaz de realizar 2 milhões de adições por segundo e cerca de 500 mil multiplicações, tornando seus antecessores totalmente obsoletos.

1.3.5 Quarta geração: microprocessadores (1970 - atual)

Há circuitos integrados de diversos tamanhos, tipos e funções, desde os que contêm algumas dezenas de milhares de transistores até circuitos integrados extraordinariamente mais complexos e “inteligentes” – ou seja, capazes de cumprir múltiplas funções de acordo com comandos ou “instruções” a eles fornecidos.

A partir de 1970, as evoluções tecnológicas ocorreram principalmente na miniaturização dos componentes internos dos computadores; entretanto, os avanços ficaram relacionados à escala de integração dos circuitos integrados, ou seja, na quantidade de dispositivos era possível incluir em um único *chip*. A Tabela 1.1 apresenta as características de cada escala.



Curiosidade: Em novembro de 1971, a INTEL introduziu o primeiro microprocessador comercial, o 4004. Continha 2.300 transistores e executava cerca de 60.000 cálculos por segundo. Processava apenas quatro bits por vez e operava a apenas 1 MHz. A partir dessas evoluções surgem os computadores denominados microcomputadores. Como o próprio nome diz, computadores extremamente menores que seus antecessores, mais rápidos e cada vez mais acessíveis às pessoas. O primeiro kit de microcomputador foi o ALTAIR 8800 em 1974.

Curiosidade: O cérebro humano, de um ponto de vista estritamente funcional, pode ser definido como um sistema complexo de 100 bilhões de neurônios. Para conter o mesmo número de elementos do cérebro, um computador dos anos 1940, a fase das válvulas, teria o tamanho de São Paulo. Em fins dos anos 1950, com os transistores, bastava um computador com as dimensões do Cristo Redentor. Anos 1960: o computador seria como um ônibus. Atualmente, com a aglomeração de alguns milhões de componentes num único chip, cérebro e computador entraram num acordo de dimensões



Alguns historiadores afirmam que a quarta geração de computadores se estende até os dias de hoje, com os circuitos VLSI. Outros criaram a quinta geração de computadores. Para saber um pouco mais sobre essa última geração, consulte: http://www.cesarkallas.net/arquivos/faculdade/inteligencia_artificial/outs/artdp2a.pdf e em seguida monte um pequeno texto, contendo possíveis ilustrações. Esse texto e as ilustrações devem ser postados em formato de blog criado no AVEA.

Tabela 1.1: Escalas de integração

Denominação		Complexidade (números de transistores)		
		Interpretação comum	Tanenbaum ^[2]	Texas Instruments ^[8]
SSI	<i>Small Scale Integration</i>	10	1–10	abaixo de 12
MSI	<i>Medium Scale Integration</i>	100	10–100	12–99
LSI	<i>Large Scale Integration</i>	1.000	100–100.000	100–999
VLSI	<i>Very Large Scale Integration</i>	10.000–100.000	a partir de 100.000	acima de 1.000
ULSI	<i>Ultra Large Scale Integration</i>	100.000–1.000.000	—	—
SLSI	<i>Super Large Scale Integration</i>	1.000.000–10.000.000	—	—

Fonte: Computador... (2010)

Nessa geração os circuitos passaram a uma larga escala de integração – *Large Scale Integration* (LSI), a partir do desenvolvimento de várias técnicas, e aumentou significativamente o número de componentes em um mesmo *chip*. Como podemos observar na Figura 1.7, as características físicas de um microprocessador (frente e verso).

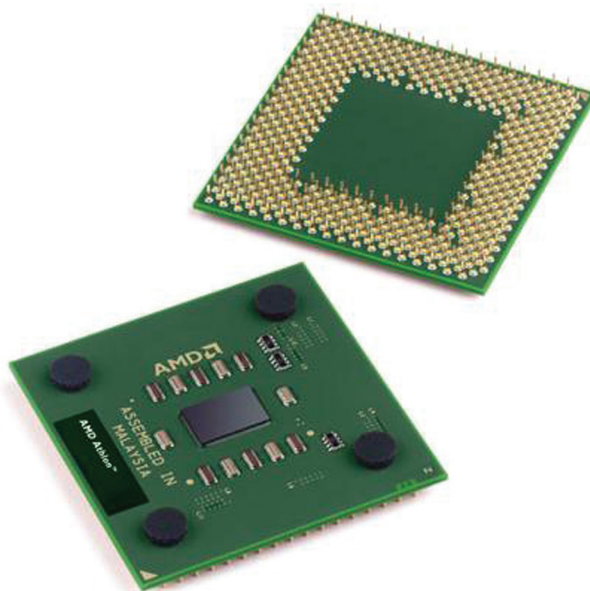


Figura 1.7: Microprocessador

Fonte: www.sharkyextreme.com/.../xp2200_dual_slant.jpg

Em 1970, a INTEL Corporation lançou no mercado um tipo novo de circuito integrado: o microprocessador. Os microprocessadores concentravam todos os componentes principais de um computador: a *Central Processing Unit* (CPU) ou Unidade Central de Processamento; controladores de memória e de entrada e saída. Assim, os primeiros computadores ao utilizarem o microprocessador eram denominados “computadores de quarta geração”.

1.3.6 O futuro

Atualmente dispomos de computadores extremamente velozes, que apresentam tamanhos cada vez menores; esses avanços são possíveis graças às pesquisas que não cessam, por parte de inúmeras universidades e instituições de pesquisa dispostas a descobrir o novo. Assim surgiu o que conhecemos por nanotecnologia, ou seja, a capacidade potencial de criar a partir da miniaturização, permitindo, assim, o desenvolvimento de dispositivos miniaturizados para compor um computador, por exemplo.

Sempre em busca da descoberta de novas tecnologias, nesse caso que possibilitem que o computador se torne mais rápido que seu antecessor (mesmo que o seu antecessor já seja extremamente rápido e possua desempenho além das expectativas dos usuários), surgiram novas tecnologias para a construção de computadores não mais baseadas em conceitos digitais (0 e 1) e energia elétrica. Assim, surgiram os computadores ópticos: em que feixes de luz poderão se cruzar em um cubo óptico, transportando informação digital.

Os computadores quânticos também estão sendo largamente pesquisados ao redor do mundo, havendo iniciativas inclusive no Brasil. Nesse tipo de computador, são os átomos que desempenham o papel dos transistores. Ao contrário dos clássicos *bits* digitais (0 e 1), as menores unidades de informação de um computador quântico podem assumir qualquer valor entre zero e um. Dessa forma, existem previsões bem otimistas de que essa nova tecnologia substitua o silício (matéria-prima dos transistores) em pouco tempo.

Participe do fórum “O Futuro dos Computadores” disponível no AVEA e coloque as suas opiniões e questionamentos sobre o assunto, com base nos textos propostos acima e em outras informações que julgar relevantes.



Nanotecnologia. Disponível em: <http://pt.wikipedia.org/wiki/Nanotecnologia>. Acesso em: 19 jul. 2010.

Computador Óptico – Futuro dos Computadores. Disponível em: http://www.lucalm.hpg.ig.com.br/mat_esp/comp_optico/computador_optico.htm

Computador quântico já funciona. Disponível em: <http://info.abril.com.br/aberto/infonews/022007/15022007-3.shl>. Acesso em: 19 jul. 2010.

Simulador de computador quântico. Disponível em: <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=010150040809>. Acesso em: 19 jul. 2010.

1.4 Componentes básicos de um sistema computacional

Segundo a arquitetura de Von Neumann, os computadores possuem quatro componentes principais: Unidade Central de Processamento (UCP) – composta pela Unidade Lógica e Aritmética (ULA) e a Unidade de Controle (UC), a memória e os dispositivos de entrada e saída. Tais componentes são interconectadas por barramentos. E todos esses itens constituem o *hardware* de um computador (seu conjunto de componentes físicos), os quais são agrupados em módulos específicos, constituindo a estrutura básica de um computador. A Figura 1.8 mostra, de forma genérica, essa estrutura.





Para complementar o conteúdo apresentado nesta unidade, recomenda-se:

- Assistir ao vídeo: "Como os chips são fabricados", disponível em: <http://www.clubedohardware.com.br/artigos/1131>
Acesso em: 19 jul. 2010.

- Assistir ao vídeo: "Como os Computadores Funcionam", disponível em: <http://www.youtube.com/watch?v=Wf1jnh8TCXA>.
Acesso em: 19 jul. 2010.

- Ler o artigo disponível em: <http://www.guiadohardware.net/artigos/evolucao-processadores>.
Acesso em: 19 jul. 2010.

Feito isso, realize a postagem de seus comentários, dúvidas ou informações relevantes sobre a evolução dos processadores e o seu processo de fabricação no blog "Evolução dos processadores e suas tecnologias", disponível no AVEA.

Cabe esclarecer que quando se fala em processador está se falando genericamente da UCP. Muitas pessoas usam a sigla UCP ou CPU para indicar o gabinete do computador, o que é errado.

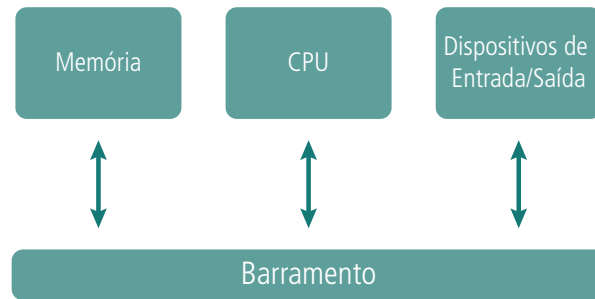


Figura. 1.8: Microprocessador

Fonte: Elaborada pela autora

A seguir serão descritos cada um dos principais componentes de um computador:

- a) UCP: sigla representativa de Unidade Central de Processamento. Podemos dizer que se trata do componente principal do computador. Algumas pessoas chamam de processador ou microprocessador. É responsável pela execução de dados e instruções armazenadas em memória (código de programas e dados);
- b) Memória: existem diversos tipos de memória em um computador (ex.: RAM (principal), ROM, *cache*, registradores), mas existe uma delas denominada memória principal, a qual é indispensável. A memória principal é tão importante quanto a UCP, pois sem ela não seria possível disponibilizar os programas e seus dados para o processamento pela CPU. Portanto, a memória é responsável por armazenar todos os programas que executam no computador e os dados que utilizam;
- c) Dispositivos de Entrada e Saída (E/S): são dispositivos responsáveis pelas entradas e saídas de dados, ou seja, pelas interações entre o computador e o mundo externo (usuários). São exemplos de dispositivos de E/S: monitor de vídeo, teclado, *mouse*, *webcam*, impressora, entre outros;
- d) Barramento: é responsável por interligar todos os componentes listados acima. Trata-se de uma via de comunicação composta por diversos fios ou condutores elétricos por onde circulam os dados manipulados pelo computador.

As aulas subsequentes objetivam apresentar e explicar cada um dos elementos que compõem um computador, para que você entenda o seu funcionamento e suas especificações técnicas. Inicialmente vamos estudar os sistemas de numeração, para que você compreenda a linguagem que o computador entende (binária), seguido do estudo das portas lógicas e suas funções, de forma a entender os elementos internos de cada um dos componentes apresentados acima.

Resumo

Esta aula apresentou e discutiu os componentes do sistema computacional, identificando características e diferenças entre o computador analógico e o computador digital. Este último é o modelo de computador utilizado atualmente, o qual se baseia na representação de dados por meio dos dígitos 0 e 1. Foi apresentada a evolução dos computadores, sempre focando nas tecnologias utilizadas na construção dos computadores em cada uma das gerações. Sendo assim, foram apresentadas:

- Geração 0 (1642-1945): que possuía essencialmente computadores mecânicos;
- Primeira geração (1945-1955): utilizava-se essencialmente de válvulas eletrônicas para a construção de computadores;
- Segunda geração (1955-1965): baseava-se no uso de transistores. Foi proposto por John Von Neumann o modelo de computador utilizado até os dias de hoje, composto de: memória, unidade de processamento e dispositivos de entrada e saída;
- Terceira geração: (1965-1980): baseava-se no uso de circuitos integrados;
- Quarta geração (1970-atual): surgiu o microprocessador.

Outras iniciativas de pesquisa resultaram em novas tecnologias que devem ser lançadas oficialmente em breve, sempre visando aumentar o desempenho e reduzir o volume das máquinas computacionais. Dentre essas novas tecnologias, destacam-se os computadores quânticos e os computadores ópticos.

Foram discutidos os componentes básicos de um computador, visando esclarecer todos os tópicos a serem tratados neste caderno. São eles: Unidade Central de Processamento (UCP) – composta pela Unidade Lógica e Aritmética (ULA) e a Unidade de Controle (UC), a memória e os dispositivos de entrada e saída e o barramento.

Atividades de aprendizagem

1. Em relação aos aspectos relacionados à evolução do sistema computacional, responda aos seguintes questionamentos:
 - a) Explique com suas palavras no que consiste um sistema computacional.
 - b) Qual a diferença entre um computador analógico e um computador digital?
 - c) Os computadores atuais são analógicos ou digitais? Explique.
 - d) O que John Von Neumann significou para a computação?
 - e) Qual a composição do modelo de Von Neumann e qual a relação desse modelo com os computadores atuais?
 - f) No que consiste um transistor e qual a sua contribuição para a evolução dos computadores?
 - g) Qual a composição de um circuito integrado?
 - h) Qual a função dos circuitos integrados?
 - i) Em qual das escalas de integração se classificam os microprocessadores?
2. Atividade de pesquisa - realizar uma pesquisa sobre computadores quânticos:
 - Deve conter no mínimo uma e no máximo três páginas (fonte tamanho 12, tipo Arial ou Times New Roman);
 - Deve apresentar as referências, ou seja, a(s) fonte(s) de onde foram retiradas as informações da pesquisa;
 - Deve conter um último tópico chamado Conclusão, com pelo menos um e no máximo três parágrafos apresentando o seu entendimento sobre o assunto;
 - As fontes da pesquisa deverão ser os sites da internet;
 - Essa atividade deverá ser postada como atividade no AVEA, no formato de um arquivo texto.
3. Procure por equipamentos que contenham transistores, capacitores ou circuitos integrados em sua construção e faça fotos desses circuitos. Insira essas fotos num blog a ser criado no AVEA. Cada foto deve estar acompanhada de uma descrição contendo características como: equipamento a que pertence (nome, função, especificação), a idade aproximada do equipamento ou ano de fabricação.

Aula 2 – Sistemas de numeração

Objetivo

Compreender o sistema de numeração utilizado pelos sistemas computacionais.

2.1 Bases e sistemas de numeração

Desde o início de sua existência, o homem sentiu a necessidade de contar objetos, fazer divisões, diminuir, somar, entre outras operações aritméticas de que hoje se tem conhecimento. Diversas formas de contagem e representação de valores foram propostas. Podemos dizer que a forma mais utilizada para a representação numérica é a notação posicional.

Segundo Monteiro (2007), na notação posicional, os algarismos componentes de um número assumem valores diferentes, dependendo de sua posição relativa nele. O valor total do número é a soma dos valores relativos de cada algarismo. Dessa forma, dependendo do sistema de numeração adotado, é dito que a quantidade de algarismos que o compõem é denominada base. Assim, a partir do conceito de notação posicional, tornou-se possível a conversão entre diferentes bases.

Considerando os aspectos apresentados em relação à representação numérica, as seções a seguir irão tratar com mais detalhes sobre cada um deles. Destacamos que o texto que segue foi baseado na obra de Monteiro (2007).

2.1.1 Notação posicional

A notação posicional é uma consequência da utilização dos numerais hindu-árabicos. Os números romanos, por exemplo, não utilizam a notação posicional. Desejando efetuar uma operação de soma ou subtração, basta colocar um número acima do outro e efetuar a operação desejada entre os numerais, obedecendo a sua ordem. A civilização ocidental adotou um sistema de numeração que possui dez algarismos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9), denominado de sistema decimal.

A quantidade de algarismos de um dado sistema é chamada de *base*; portanto, no sistema decimal a base é 10. O sistema binário possui apenas dois algarismos (0 e 1), sendo que sua base é 2.



Exemplos:

$$4325_{10} = 5 \times 10^0 + 2 \times 10^1 + 3 \times 10^2 + 4 \times 10^3$$

$$1011_2 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 1 + 2 + 0 + 8 = 11_{10}$$

$$3621_8 = 1 \times 8^0 + 2 \times 8^1 + 6 \times 8^2 + 3 \times 8^3 = 1937_{10}$$

Generalizando, num sistema de numeração posicional qualquer, um número

N é expresso da seguinte forma:

$$N = d_{n-1} \times b^{n-1} + d_{n-2} \times b^{n-2} + \dots + d_1 \times b^1 + d_0 \times b^0$$

Onde:



Observações importantes:

- O número de algarismos diferentes em uma base é igual à própria base.
- Em uma base “**b**” e utilizando “**n**” ordens temos **bn** números diferentes.

2.1.2 Conversão de bases

O Quadro 2.1 mostra a equivalência entre as bases decimal, binária, octal e hexadecimal.

Quadro 2.1: Exemplo de conversão de bases envolvendo as bases 2, 8 e 16			
Decimal	Binário	Hexadecimal	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
(continua)			

Decimal	Binário	Hexadecimal	Octal
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
(conclusão)			

Fonte: Adaptado de Monteiro (2007)

2.1.2.1 Base binária para base octal ou hexadecimal

Observe que os dígitos octais e hexadecimais correspondem às combinações de 3 (para octais) e 4 (para hexadecimais) bits (ou seja, da representação binária – disponível na tabela de equivalências apresentada anteriormente), permitindo a fácil conversão entre estes sistemas.

$$\underbrace{101}_5 \underbrace{111}_7 \underbrace{011}_3 \underbrace{101}_5_2 = 5735_8$$

$$\underbrace{1011}_B \underbrace{1101}_D \underbrace{1101}_D_2 = BDD_{16}$$

2.1.2.2 Base octal ou hexadecimal para base binária

A conversão inversa de octal ou hexadecimal para binário deve ser feita a partir da representação binária de cada algarismo do número, seja octal ou hexadecimal.

2.1.2.3 Base octal para base hexadecimal (e vice-versa)

A representação binária de um número octal é idêntica à representação binária de um número hexadecimal, a conversão de um número octal para hexadecimal consiste simplesmente em agrupar os *bits* não mais de três em três (octal), mas sim de quatro em quatro *bits* (hexadecimal), e vice-versa.

2.1.2.4 Base B (qualquer) para base decimal

Atenção, nos exemplos de casos citados a seguir, sempre utilizamos a definição de Notação Posicional:

$$101101_2 = ?_{10}$$

$$b = 2, n = 6$$

Portanto: $1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 8 + 4 + 1 = 45_{10}$

Logo: $101101_2 = 45_{10}$

$27_8 = ?_{10}$

$b = 8, n = 2$

Portanto: $2 \times 8^1 + 7 \times 8^0 = 23_{10}$

Logo: $27_8 = 23_{10}$

$2A5_{16} = ?_{10}$

$b = 16, n = 3$

Portanto: $2 \times 16^2 + 10 \times 16^1 + 5 \times 16^0 = 512 + 160 + 5 = 677_{10}$

Logo: $2A5_{16} = 677_{10}$

2.1.2.5 Base decimal para base B (qualquer)

Consiste no processo inverso, ou seja, efetuamos divisões sucessivas do número decimal pela base desejada, até que o quociente seja menor que a referida base. Utilizamos os restos e o último quociente (a começar dele) para formação do número desejado, conforme Quadro 2.2.



Para auxiliar no entendimento do conteúdo apresentado nesta unidade, recomenda-se: Assistir ao vídeo: "Números Binários". Disponível em: <http://www.youtube.com/watch?v=TJRYW-lSocU>. Acesso em: 19 jul. 2010. Assistir ao vídeo: "Sistemas de Representação e Conversão entre Bases". Disponível em: http://www.youtube.com/watch?v=RQfit_s7Afg&feature=related. Acesso em: 19 jul. 2010. Teça um pequeno comentário sobre o que assistiu e poste-o em forma de apresentação digital no fórum de nosso curso.

Quadro 2.2: Exemplo de conversão de bases envolvendo as bases 2, 8 e 16

$25_{10} = ?_2$	$3964_{10} = ?_8$	$2754_{10} = ?_{16}$
$25 : 2 = 12$ (resto 1)	$3964 : 8 = 495$ (resto 4)	$2754 : 16 = 172$ (resto 2)
$12 : 2 = 6$ (resto 0)	$495 : 8 = 61$ (resto 7)	$172 : 16 = 10$ (resto 12)
$6 : 2 = 3$ (resto 0)	$61 : 8 = 7$ (resto 5)	Logo: $2754_{10} = AC2_{16}$
$3 : 2 = 1$ (resto 1)	Logo: 7574_8	
Logo: 11001_2		

Fonte: Elaborado pela autora

Resumo

Esta aula nos mostrou que a conversão de bases é um conteúdo importante e bastante utilizado quando se discutem ou se projetam ambientes computacionais.

Podemos dizer que a forma mais utilizada para a representação numérica é a **notação posicional**. Na notação posicional, os algarismos componentes de um número assumem valores diferentes, dependendo de sua posição relativa no número. O valor total do número é a soma dos valores relativos de cada algarismo. Dessa forma, dependendo do sistema de numeração adotado, é dito que a quantidade de algarismos que o compõem é denominada **base**. Assim, a partir do conceito de notação posicional, tornou-se possível a conversão entre diferentes bases (MONTEIRO, 2007).

A quantidade de algarismos de um dado sistema é chamada de base; portanto, no sistema decimal a base é 10. O sistema binário possui apenas dois algarismos (0 e 1), sendo que sua base é 2. Da mesma forma para a base 8 (octal) e base 16 (hexadecimal). Uma tabela de conversão é utilizada (Quadro 2.1).

Os dígitos octais e hexadecimais correspondem a combinações de 3 (para octais) e 4 (para hexadecimais) *bits* – de acordo com a tabela de conversão, o que permite a fácil conversão entre estes sistemas. Uma vez que a representação binária de um número octal é idêntica à representação binária de um número hexadecimal, a conversão de um número octal para hexadecimal consiste simplesmente em agrupar os *bits* não mais de três em três (octal), mas sim de quatro em quatro *bits* (hexadecimal), e vice-versa.

Para converter um número de uma base B qualquer para decimal, basta usar a notação posicional. O processo contrário é obtido pela realização de divisões sucessivas do número decimal pela base desejada, até que o quociente seja menor que a referida base. Utilizam-se os **restos** e o último **quociente** (a começar dele) para formação do número desejado.

Atividades de aprendizagem

As atividades abaixo devem ser realizadas em um documento de texto e postadas como atividade no AVEA. Quem resolver as atividades fazendo uso de lápis, ou caneta, poderá postar um arquivo escaneado.

1. Converter os seguintes valores decimais em valores binários e hexadecimais:

a) 329_{10}

b) 284_{10}

c) 473_{10}

d) 69_{10}

e) 135_{10}

2. Converter os seguintes valores binários em valores decimais, octais e hexadecimais:

a) 11011101010_2

b) 11001101101_2

c) 1000001111_2

d) 11101100010_2

e) 111001101001_2

3. Converter os seguintes valores hexadecimais em valores decimais e binários:

a) $3A2_{16}$

b) $33B_{16}$

c) 621_{16}

d) 99_{16}

e) $1ED4_{16}$

Aula 3 – Portas lógicas e suas funções

Objetivos

Identificar os elementos básicos de organização: portas lógicas e circuitos combinacionais.

Entender o funcionamento de um circuito integrado (CI).

3.1 Funções e portas lógicas

De acordo com Monteiro (2007), um computador é constituído de elementos eletrônicos, como resistores, capacitores e principalmente transistores. Nesses computadores, os transistores são, em geral, componentes de determinados circuitos eletrônicos que precisam armazenar os sinais binários e realizar certos tipos de operações com eles. Esses circuitos, chamados de circuitos digitais, são formados por pequenos elementos capazes de manipular grandezas apenas binárias. Os pequenos elementos referidos são conhecidos como portas (*gates*) lógicas, por permitirem ou não a passagem desses sinais, e os circuitos que contêm as portas lógicas são conhecidos como circuitos lógicos.

Assim, circuitos lógicos são montados a partir da combinação de uma unidade básica construtiva denominada porta lógica, a qual é obtida mediante a combinação de transistores e dispositivos **semicondutores** auxiliares. Portanto, a porta lógica é a base para a construção de qualquer sistema digital (ex.: o microprocessador).

Em geral, os circuitos lógicos são agrupados e embutidos em um Circuito Integrado (CI). Esses dispositivos implementam uma determinada função com o objetivo de cumprir uma tarefa específica.

Portas lógicas são encontradas desde o nível de integração em Ultra Larga Escala (ULSI) ou Super Larga Escala (SLSI) – conforme apresentado ou discutido na Aula 1 (utilizadas em microprocessadores) – até o nível de integração existente em circuitos digitais mais simples, desempenhando funções mais básicas (ex.: comparações, somas, multiplicações).

A-Z

Semicondutores

Semicondutores: são materiais que se encontram em uma posição intermediária, não sendo bons condutores e nem bons isolantes, mas, ao serem tratados por processos químicos, permitem controlar a passagem de uma corrente elétrica.

Álgebra de Chaveamentos

George Boole (1815-1864) foi um matemático e filósofo britânico, criador da Álgebra Booleana, base da atual aritmética computacional.

Dessa forma, segundo Monteiro (2007), o projeto de circuitos digitais e a análise de seu comportamento em um computador podem ser realizados por meio da aplicação de conceitos e regras estabelecidas por uma disciplina conhecida como **Álgebra de Chaveamentos**, a qual é um ramo da Álgebra de Boole ou Álgebra Booleana.

Semelhante à álgebra tradicional (estudada no ensino médio), torna-se necessário definir símbolos matemáticos e gráficos para representar as operações lógicas e seus operadores.

Uma operação lógica qualquer (ex.: soma ou multiplicação de dígitos binários) sempre irá resultar em dois valores possíveis: 0 (falso) ou 1 (verdadeiro). Assim, pode-se pre-definir todos os possíveis resultados de uma operação lógica, de acordo com os possíveis valores de entrada. Para representar tais possibilidades, utiliza-se de uma forma de organizá-las chamada Tabela Verdade. Dessa forma, cada operação lógica possui sua própria tabela verdade (MONTEIRO, 2007). A seguir será apresentado o conjunto básico de portas lógicas e suas respectivas tabelas verdade.

3.1.1 Porta AND

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura 3.1. Os operandos são binários simples (0 e 1).

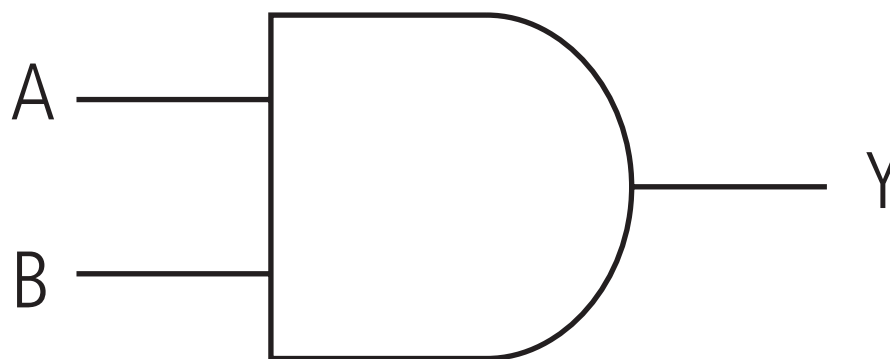


Figura. 3.1: Porta lógica AND (E)

Fonte: Adaptada de Monteiro (2007)

Pode-se dizer que a operação AND simula uma multiplicação binária, permitindo os possíveis resultados conforme mostra a Tabela 3.1.

Tabela 3.1: Tabela verdade da porta lógica AND		
Entrada		Saída
A	B	$X = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

Fonte: Adaptada de Monteiro (2007)

Conforme é possível observar, a regra é: “se o primeiro operando é 1 e o segundo operando é 1, o resultado é 1 (Verdadeiro), senão o resultado é 0 (Falso)”.

Um exemplo de aplicação de uma porta AND seria na composição de um circuito para a transferência de *bits* de dados de um local para outro (ex.: da memória para a CPU). Nesse caso, a finalidade seria a de garantir que um *bit* de origem seja o mesmo *bit* de destino (MONTEIRO, 2007).

3.1.2 Porta OR

Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura 3.2. Os operandos são binários simples (0 e 1).

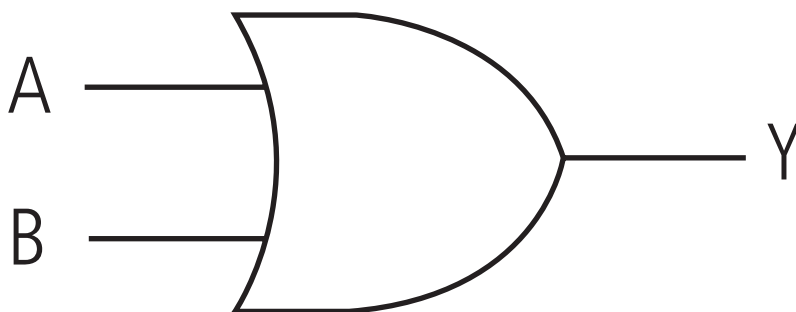


Figura. 3.2: Porta lógica OR (OU)

Fonte: Adaptada de Monteiro (2007)

Podemos dizer que a operação OR simula uma soma de binários, permitindo os possíveis resultados conforme mostra a Tabela 3.2.

Tabela 3.2: Tabela verdade da porta lógica OR		
Entrada		Saída
A	B	$X = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Fonte: Adaptada de Monteiro (2007)

Conforme podemos observar, a regra é: “se o primeiro operando é 1 ou o segundo operando é 1, ou se os dois operandos forem 1, o resultado é 1, senão o resultado é 0”.

As operações lógicas OR são muito utilizadas em lógica digital ou mesmo em comandos de decisão de algumas linguagens de programação (ex.: **Se** (X=1 **OU** Y=1) **Então** Executa uma ação) (MONTEIRO, 2007).

3.1.3 Porta NOT

A porta NOT representa um inversor. Essa operação aceita apenas um operando ou uma entradas (A), conforme mostra a Figura 3.3. O operando pode ser um dígito binário (0 ou 1).

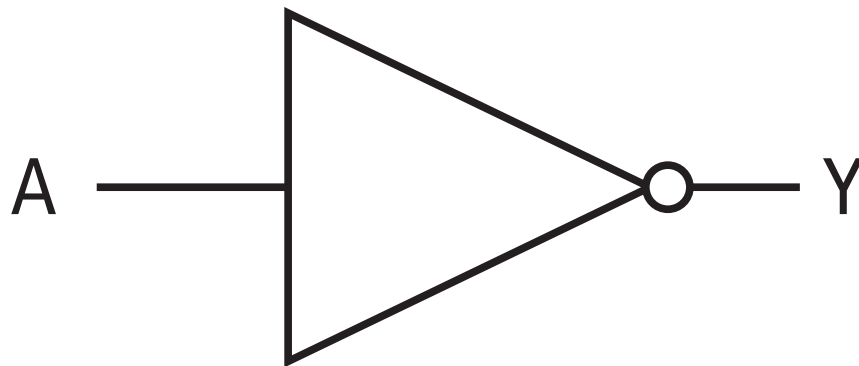


Figura. 3.3: Porta lógica NOT (NÃO)

Fonte: Adaptada de Monteiro (2007)

Pode-se dizer que a operação NOT realiza a inversão de um dígito binário, permitindo os possíveis resultados conforme mostra a Tabela 3.3.

Tabela 3.3: Tabela verdade da porta lógica NOT

Entrada	Saída
A	$X = \bar{A}$
0	1
1	0

Fonte: Adaptada de Monteiro (2007)

Conforme é possível observar, a regra é: “se o operando for 1, o resultado é 0, senão o resultado é 1”.

Entre as principais aplicações dos circuitos inversores está a representação de números negativos no formato binário, quando se usa o que é chamado de complemento a 1 ou complemento a 2, fazendo-se necessária a inversão de um grupo de *bits* representativos de um número negativo (MONTEIRO, 2007).

3.1.4 Porta XOR

A denominação XOR é a abreviação do termo EXCLUSIVE OR. Trata-se de uma operação que aceita dois operandos ou duas entradas (A e B), conforme mostra a Figura 3.4. Os operandos são binários simples (0 e 1).

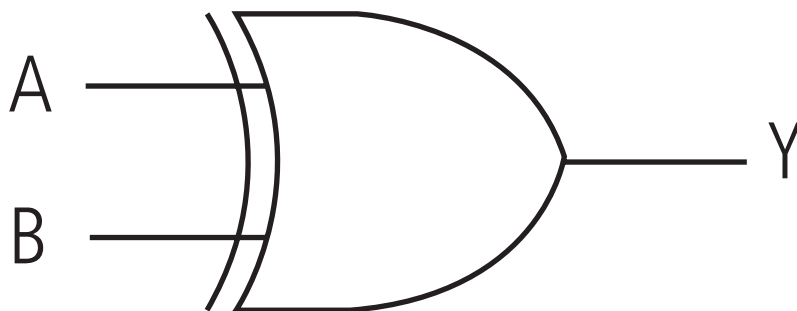


Figura. 3.4: Porta lógica NOT (NÃO)

Fonte: Adaptada de Monteiro (2007)

Pode-se dizer que a operação XOR possui como principal função a verificação de igualdade, permitindo os possíveis resultados conforme mostra a Tabela 3.4.

Tabela 3.4: Tabela verdade da porta lógica XOR		
Entrada		Saída
A	B	$X = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Fonte: Adaptada de Monteiro (2007)

Conforme é possível observar, a regra é: “se o primeiro operando ou o segundo operando for igual a 1, o resultado é 1; senão, o resultado é 0”. Ou seja, para entradas iguais a saída será 0 e para entradas diferentes a saída será 1.

A porta XOR possui inúmeras aplicações, sendo um elemento lógico bastante versátil, permitindo, por exemplo, a fabricação de um testador de igualdade entre valores, para testar, de modo rápido, se duas palavras de dados são iguais (MONTEIRO, 2007).

3.1.5 Outras portas lógicas e circuitos combinatórios

Existem outras portas lógicas derivadas das portas lógicas apresentadas acima, tais como as portas NAND (porta AND invertida) e a porta NOR (porta OR invertida). Ambas são apresentadas nas Figuras 3.5 e 3.6, respectivamente.

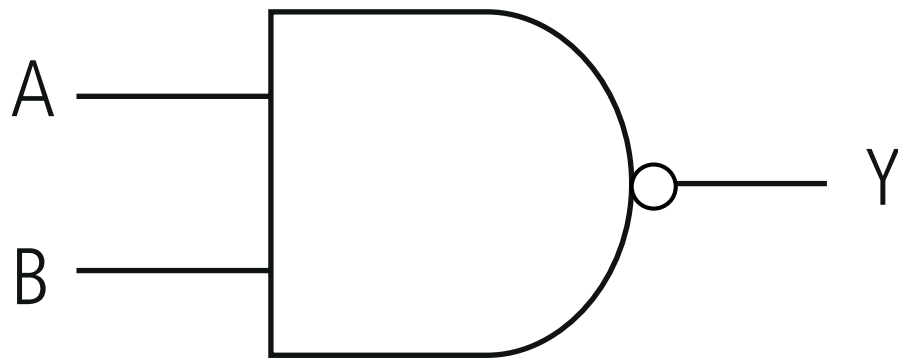


Figura. 3.5: Porta lógica NAND (NÃO E)

Fonte: Adaptadas de Monteiro (2007).

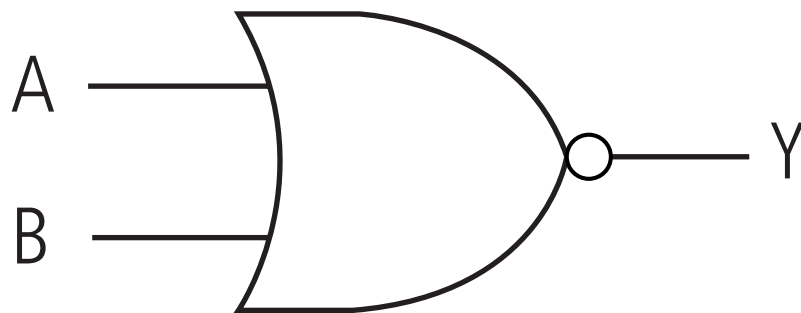


Figura. 3.6: Porta lógica NOR (NÃO OU)

Fonte: Adaptadas de Monteiro (2007).

As operações lógicas são realizadas em dois passos: primeiro a operação AND ou OR e, em seguida, o seu resultado é invertido. Esse tipo de portas lógicas também possui diversas aplicações, sendo utilizado para reduzir a complexidade e a quantidade de portas lógicas necessárias a um determinado circuito lógico.

Não esqueça de que um circuito lógico pode possuir diversas portas lógicas e, portanto, suas tabelas verdade poderão ter inúmeras entradas e inúmeras saídas (as quais poderão ser representadas por suas respectivas equações booleanas). A Figura 3.7 mostra o resumo dos símbolos gráficos e matemáticos (equação booleana) de portas lógicas.

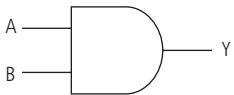
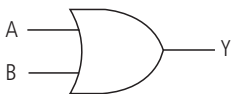
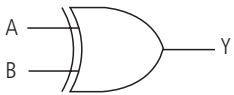
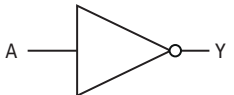
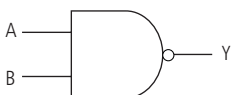
Função Lógica Básica	Símbolo Gráfico da Porta	Equação Booleana
AND		$Y = A.B$
OR		$Y = A+B$
XOR		$Y = A \oplus B$
NOT		$Y = \bar{A}$
NAND		$Y = \overline{A.B}$
NOR		$Y = \overline{A+B}$

Figura. 3.7: Símbolos gráficos e equações booleanas de portas lógicas

Fonte: Adaptada de Monteiro (2007)

A partir das portas lógicas básicas, é possível interligar diversas de suas unidades, de modo a construir redes lógicas, também chamadas de circuitos combinatórios. Monteiro (2007) explica que um circuito combinatório é definido como um conjunto de portas lógicas cuja saída em qualquer instante de tempo é função somente das entradas. O autor afirma ainda que existe outra categoria de circuitos que combina portas lógicas, denominada circuitos sequenciais, os quais, além de possuir portas, contêm elementos de armazenamento (uma espécie de memória). A Figura 3.8 exemplifica um circuito combinatório implementado para uma determinada função.

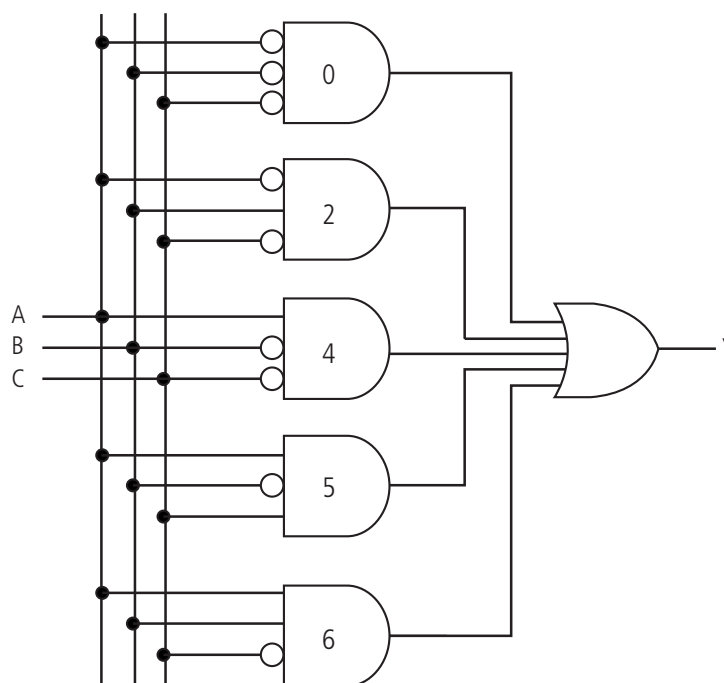


Figura. 3.8: Circuito combinatório

Fonte: <http://www.mspc.eng.br/eledig/eldg0410.shtml>



Para auxiliar na compreensão do conteúdo desta aula, recomenda-se:

Ler o artigo: "Introdução às Portas Lógicas". Disponível em: <http://www.clubedohardware.com.br/artigos/1139>. Acesso em: 19 jul. 2010.

Assistir ao vídeo: "Portas Lógicas AND, OR, XOR, NOT". Disponível em: <http://www.youtube.com/watch?v=4ENGYy68JqM&feature=related>. Acesso em: 19 jul. 2010.

Dentre os exemplos de circuitos combinatórios, destacamos o das funções essenciais da maioria dos computadores e sistemas digitais, ou seja, a realização de operações aritméticas tais como: adição, subtração, multiplicação e divisão. Essas operações são realizadas na Unidade Lógica Aritmética (ULA) destes sistemas digitais, onde uma série de portas lógicas são combinadas para adicionar, subtrair, multiplicar ou dividir números binários. No caso das operações de multiplicação e divisão, além das portas lógicas, há a necessidade de circuitos sequenciais.

Resumo

Nesta aula apresentamos e discutimos os elementos básicos de construção de circuitos digitais (ex.: microprocessador). Esses elementos são denominados portas lógicas.

As portas lógicas executam operações sobre operadores binários (0 e 1), podendo ser aplicadas para diversas finalidades. As principais portas são: AND, OR, NOT e XOR.

Quando agrupadas formam os chamados circuitos combinatórios que, quando possuem uma memória acoplada, são chamados de sequenciais. Esses circuitos digitais podem representar, por exemplo, a Unidade Lógica Aritmética (ULA).

Atividades de aprendizagem

As atividades abaixo devem ser realizadas em um documento de texto e postadas como atividade no AVEA. Quem resolver as atividades fazendo uso de lápis, ou caneta, poderá postar um arquivo escaneado.

1. Desenhe os circuitos que implementam as seguintes expressões booleanas.

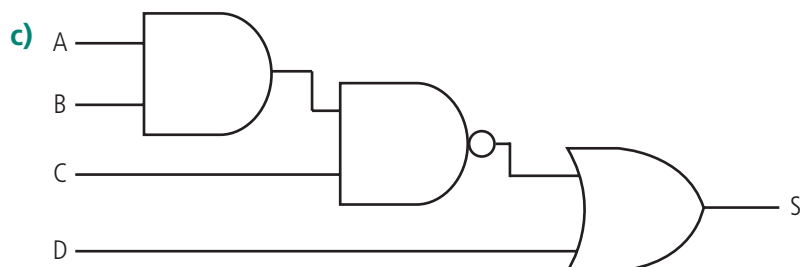
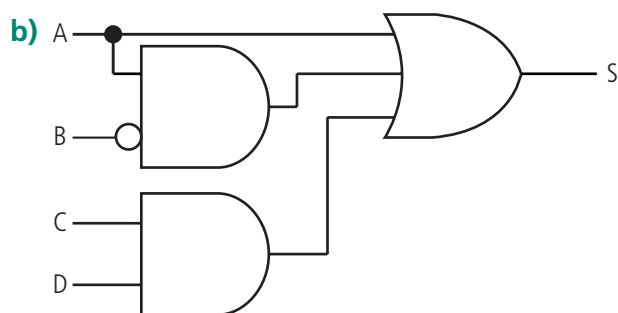
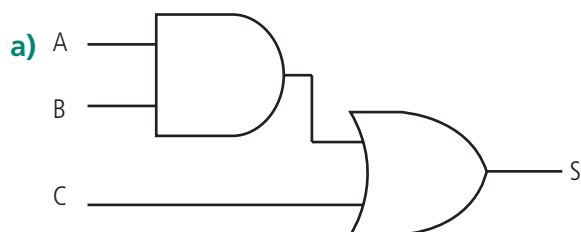
a) $S = \overline{A\overline{B}} + \overline{C\overline{D}}$

b) $S = \overline{(A\overline{B}\overline{C} + \overline{C}\overline{D}) \oplus D}$

c) $S = A\overline{B} + (C\overline{D}E)$

d) $S = A + (B + C\overline{D})(B + A)$

2. Escreva a expressão booleana executada pelos circuitos abaixo:



3. Desenvolva a tabela da verdade para as seguintes expressões:

a) $ABC + \overline{ABC}$

b) $A(\overline{C} + B + \overline{D})$

4. Justifique o estudo do conteúdo utilizado nos exercícios acima na disciplina Organização e Arquitetura de computadores. Para isso, faça seus comentários no fórum sobre “Portas Lógicas e a Construção dos Computadores”, disponível no AVEA.

Aula 4 – Subsistema de memória

Objetivos

Analisar e compreender a organização básica de memória de um computador.

4.1 Sistema de memória e suas características

De acordo com o modelo de Von Neumann (Aula 1), a função da Unidade Central de Processamento (UCP) ou processador é, essencialmente, capturar dados e instruções que compõem um programa e processá-los, não importando sua origem ou destino. Mas para que o processador possa executar os programas, seus dados e instruções devem estar armazenados na memória.

Portanto, a memória dos computadores é um elemento indispensável e tão importante quanto a Unidade Central de Processamento (CPU) ou processador. A memória é um dispositivo que permite ao computador armazenar dados de forma temporária ou permanente. Segundo Tanenbaum (2007), a memória é a parte do computador onde os programas e os dados são armazenados. Sem uma memória na qual os processadores (CPU) possam ler ou escrever informações, o conceito de computador digital com programa armazenado não pode ser implementado.

Para o funcionamento adequado de um computador, é necessário dispor, nele mesmo, de diferentes tipos de memória. Em algumas tarefas, pode ser fundamental que a transferência de dados seja feita da forma mais rápida possível – é o caso das tarefas realizadas pela CPU, onde a velocidade é fator preponderante, ao passo que a quantidade de *bits* a ser manipulada é muito pequena. Esse tipo de memória deve possuir características diferentes daquele em que a capacidade de armazenamento é mais importante que a sua velocidade de transferência de e para outros dispositivos. Destacamos que a necessidade da existência de vários tipos de memória ocorre em virtude de vários fatores concorrentes, mas principalmente em função do aumento da velocidade das CPUs (a qual é muito maior do que o tempo de acesso da memória) e da capacidade de armazenamento.

Se existisse apenas um tipo de memória, sua velocidade deveria ser compatível com a da CPU, de modo que esta não ficasse esperando muito tempo por um dado que estivesse sendo transferido. Segundo Patterson e Hannessy (2000), a CPU manipula um dado em 5 ns, ao passo que a memória transfere um dado em 60 ns.

Considerando os diversos tipos de memórias existentes, as quais variam em função de sua tecnologia de fabricação, capacidade de armazenamento, velocidade e custo, pode-se dizer que fica muito difícil projetar um computador utilizando-se apenas um único tipo de memória. Dessa forma, segundo Monteiro (2007), o computador possui muitas memórias, as quais se encontram interligadas de forma bem estruturada, constituindo o que é chamado de subsistema de memória, o qual é parte do sistema computacional.

O subsistema de memória é projetado de modo que seus componentes sejam organizados hierarquicamente. A Figura 4.1 apresenta uma pirâmide contendo a hierarquia das memórias existentes em um computador. Observa-se que a base da pirâmide é larga, simbolizando a elevada capacidade de armazenamento, o tempo de uso, a velocidade e o custo de sua tecnologia de fabricação. Assim, a base da pirâmide representa dispositivos de armazenamento de massa (memória secundária), de baixo custo por *byte* armazenado, mas ao mesmo tempo com baixa velocidade de acesso. A flecha na direção do topo indica que quanto mais rápidas forem as memórias, mais elevado será o seu custo em relação à tecnologia e menor a sua capacidade de armazenamento em um computador.



Figura 4.1: Hierarquia de memórias

Fonte: Adaptado de Monteiro (2007)

O Quadro 4.1 apresenta as características básicas de cada tipo de memória, a seguir.

Quadro 4.1: Características básicas dos tipos de memória					
	Localização	É Volátil?	Velocidade	Capacidade de armazenamento	Custo por bit
Registrador	Processador	Sim	Muito alta (opera na velocidade do processador)	Muito baixa (Bytes)	Muito alto
Cache	Processador	Sim	Alta (opera na velocidade do processador)	Baixa (KB)	Alto
Principal	Placa-mãe	RAM – sim ROM – não	Depende do tipo de memória instalada	Média (MB)	Médio (tem caído muito)
Secundária	HD, CDs, etc.	Não	Baixa (lenta)	Alta (GB)	Baixo (tem caído muito)

Fonte: Adaptada de Murdocca (2000)

A seguir apresentamos de forma detalhada cada um dos tipos de memória apresentados na Figura 4.1.

4.2 Registradores

A função da memória é a de armazenar dados destinados a serem, em algum momento, utilizados pelo processador (MONTEIRO, 2007). O processador busca dados e instruções de onde estiverem armazenadas e os deposita temporariamente em seu interior para que possa realizar as operações solicitadas utilizando seus demais componentes (seria análogo à função memória de uma calculadora). Os dispositivos denominados registradores são os locais onde esse conteúdo fica armazenado.

Assim, o conceito de registrador surgiu da necessidade do processador de armazenar temporariamente dados intermediários durante um processamento. Por exemplo, quando um dado resultado de operação precisa ser armazenado até que o resultado de uma busca da memória esteja disponível para com ele realizar uma nova operação.

Os registradores são dispositivos de armazenamento temporário (volátil), localizados no interior do processador (CPU). Por causa da tecnologia utilizada, os registradores são um tipo de memória extremamente rápida e bastante cara. Por esse motivo, sua disponibilidade em um computador é muito

A-Z

Memórias voláteis

Memórias voláteis: são as que requerem energia para manter a informação armazenada, ao contrário das memórias permanentes.

Placa-mãe

Placa-mãe: trata-se de uma placa de circuito impresso, que serve como base para a instalação dos demais componentes de um computador, como o processador, a memória RAM, os circuitos de apoio, as placas controladoras, os slots, entre outros.

A-Z

palavra

De forma genérica o termo palavra é um grupo de bits de tamanho fixo que é processado em conjunto em um computador.

O número de bits em uma palavra (comprimento da palavra) é uma característica importante de uma arquitetura de computador. A maioria dos registradores em um computador possui o tamanho da palavra.

limitada. Cada registrador possui capacidade para manter apenas um dado (**uma palavra**).

Em resumo, os registradores, conforme mostra a Figura 4.1, ficam no topo da pirâmide, o que representa que sua velocidade de transferência de dados dentro do processador é bastante elevada; em consequência disso, sua capacidade e armazenamento é baixa e seu custo é alto.

4.3 Memória cache

Considerando a premissa de que o processador precisa buscar dados e instruções em uma memória externa – denominada memória principal, a qual será apresentada na seção a seguir – para processá-los e, considerando que a tecnologia desenvolvida para os processadores fez com que se esses dispositivos sejam bem mais rápidos que a memória principal, surgiu a necessidade de diminuir esse atraso gerado pela transferência de dados entre a memória e o processador.

Na busca de soluções para a limitação imposta pela comunicação entre processador e memória, foi desenvolvida uma técnica que consiste na inclusão de um dispositivo de memória entre a memória principal e o processador. Esse dispositivo é denominado memória *cache*. Sua função principal é acelerar a velocidade de transferência das informações entre processador e memória principal e, com isso, aumentar o desempenho dos sistemas de computação. As memórias *cache* são voláteis, assim como os registradores, pois dependem de energia para manter o seu conteúdo armazenado. A Figura 4.2 apresenta um diagrama de blocos de um processador Pentium original, a distribuição da memória cache e sua relação com a memória principal (MONTEIRO, 2007).

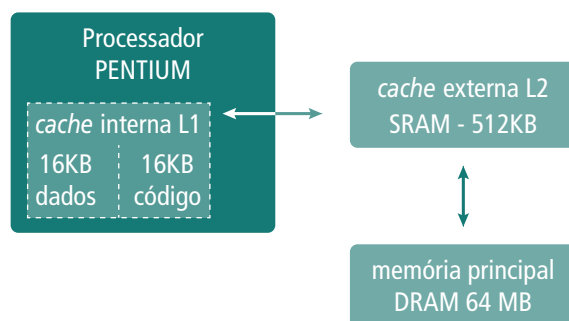


Figura 4.2: Memória cache no Pentium

Fonte: <http://www.letronet.com.br/psist/ppesq/ppesqlivcap/ppesqpc/ppesqpcd5/tppc5/pc5-11.jpg>

Assim, a memória *cache* é um tipo de memória construída com tecnologias semelhantes às do processador, isso eleva os custos de produção. Para amenizar o fator custo e dispor das vantagens de um sistema computacional com desempenho mais rápido, foram incorporadas ao computador pequenas porções de memória *cache*, localizadas internamente ao processador e entre ele e a memória principal, as quais funcionam como um espelho de parte da memória principal.

Nesse caso, quando o processador solicita um determinado dado e o encontra na *cache*, não há necessidade de requisitá-lo à memória principal, reduzindo significativamente o tempo de processamento. Ou seja, quanto mais memória *cache* um processador possuir, melhor será o seu desempenho.

A tecnologia de fabricação da memória *cache* é SRAM (*Static Random Access Memory*), a qual é bastante diferente das memórias DRAM (*Dynamic Random Access Memory*) – tecnologia da memória principal. A diferença é que nas memórias SRAM não há necessidade de *refresh* ou realimentação constante para que os dados armazenados não sejam perdidos. Isso é possível porque as memórias SRAM se utilizam de seis transistores (ou quatro transistores e dois resistores) para formar uma célula de memória. Assim, o *refresh* passa a não ser necessário, o que faz com que esse tipo de memória seja mais rápida e consuma menos energia.

De acordo com Alecrim (2010), os processadores trabalham, basicamente, com dois tipos de *cache*: *cache* L1 (*Level 1* ou Nível 1) e *cache* L2 (*Level 2* ou Nível 2). Normalmente a *cache* L2 é um pouco maior que a L1 e foi implantada quando a *cache* L1 se mostrou insuficiente.

Nas gerações anteriores, a *cache* L1 ficava localizada no interior do processador e a *cache* L2 era externa a ele. Nas gerações de computadores atuais, ambos os tipos ficam localizados dentro do *chip* do processador, sendo que, em muitos casos, a *cache* L1 é dividida em duas partes: “L1 para dados” e “L1 para instruções”. Alecrim (2010) destaca, ainda, que dependendo da arquitetura do processador, é possível o surgimento de modelos que tenham um terceiro nível de *cache* (L3).

4.4 Memória principal

É um tipo de memória indispensável para o funcionamento do computador, à qual o processador pode fazer acesso direto. Além de alocar os dados e

instruções de programas a serem manipulados pelo processador, esse tipo de memória dá acesso às memórias secundárias, de forma a disponibilizar dados ao processador.

A memória principal é denominada memória RAM (*Random Access Memory*), corresponde a um tipo de memória volátil, ou seja, seu conteúdo fica armazenado enquanto o computador estiver ligado (energizado); ao desligar a corrente elétrica, o conteúdo da memória RAM é apagado. Esse é o motivo pelo qual muitas pessoas perdem arquivos que estão utilizando quando ocorrem fatos como, por exemplo, alguém esbarrar no cabo ligado à tomada de energia elétrica ou mesmo cessar o fornecimento de energia. Isso acontece porque ao ocorrerem tais fatos, o arquivo ainda não havia sido salvo em algum tipo de memória permanente (ex.: o disco do computador). A Figura 4.3 apresenta um pente (módulo) de memória RAM típico.

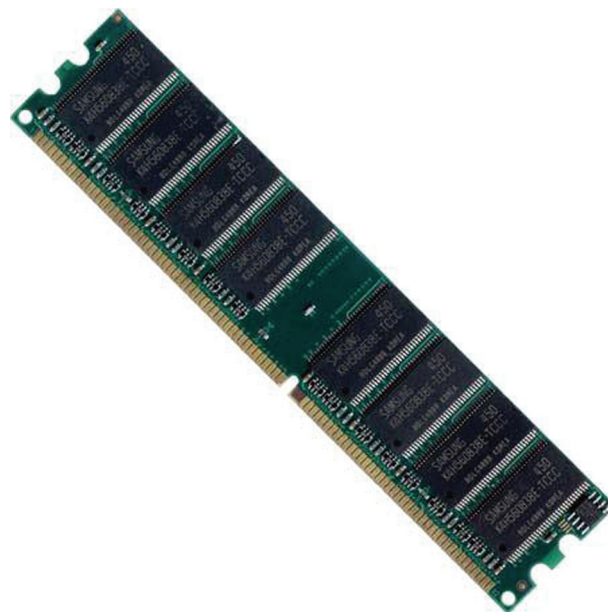


Figura 4.3: Pente de memória RAM

Fonte: <http://barreiras.olx.com.br/memoria-ram-ddr-400-1gb-iid-39138709>

A memória RAM é denominada genericamente de DRAM (*Dynamic RAM*), ou RAM dinâmica, pelo fato de possuir uma característica chamada **refrescamento de memória**, que tem a finalidade de manter os dados armazenados enquanto o computador estiver ligado. Essa denominação está ligada à tecnologia utilizada na fabricação desse tipo de memória, a qual se baseia na utilização de dispositivos semicondutores, mais especificamente capacitores associados a transistores para representar *bits* de dados armazenados. Podemos dizer que são necessários um transistor e um capacitor para representar uma célula de memória.

Pelo fato de precisarem ser “refrescadas” ou realimentadas constantemente, as memórias DRAM consomem muitos ciclos do processador para a realimentação, além de consumirem mais energia que outros tipos de memória. Por isso, são mais lentas e possuem custo muito menor e capacidade de armazenamento de dados consideravelmente maior que as memórias estáticas (ex.: *cache*).

Atualmente, podemos contar com muitas opções de padrões de memória RAM, devido à busca constante por uma memória de maior capacidade, maior velocidade de acesso, menor consumo de energia e de tempo de realimentação.

Apresentamos alguns padrões de memória RAM disponíveis mais utilizados atualmente:

- a) **DDR (*Double Data Rate*)**: duplicam o desempenho da memória, possibilitando a transferência de dois lotes de dados – entre processador e memória – por ciclo de *clock*. **DDR-2**: possibilitam a transferência de quatro lotes de dados por **ciclo de clock** e apresentam menor consumo de energia que a DDR original.
- b) **DDR-3**: transferem oito lotes de dados por ciclo de *clock* e consomem ainda menos energia que sua versão anterior.

A-Z

Ciclo de clock

Ciclo de clock: é o tempo que o processador leva para executar uma operação elementar (ex.: buscar ou transferir um dado da ou para a memória principal).

A memória RAM é comercializada para uso nos computadores no formato de pentes ou módulos de memória, contendo uma determinada quantidade desses recursos. Os pentes de memória podem variar de acordo com as características apresentadas pela memória, especialmente ligadas ao desempenho ou velocidade de transferência. Sendo assim, existem diferentes modelos de módulos de memória disponíveis no mercado. Dentre os que são utilizados atualmente, podem-se citar:

- a) **Módulos SIMM (*Single In Line Memory Module*)**: apresentam um pequeno orifício nas linhas de contato. Foram utilizados em memórias FPM e EDO RAM. Não se encontram disponíveis no mercado atualmente;
- b) **Módulos DIMM (*Double In Line Memory Module*)**: não apresentam orifícios nas linhas de contato e apresentam contatos em ambos os lados do módulo. São utilizados atualmente em memórias DDR, DDR2 e DDR3. A Figura 4.4 apresenta exemplos de módulos DIMM.

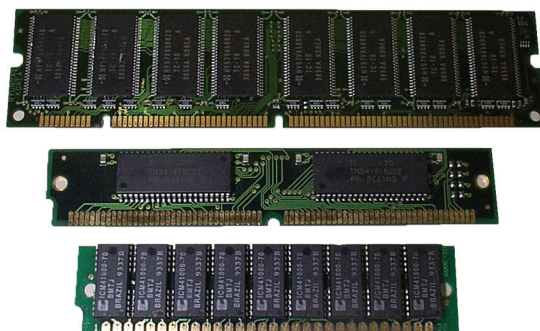


Figura 4.4: Módulos DIMM

Fonte http://www.gdhpress.com.br/hmc/leia/cap3-4_html_64f84747.jpg

4.4.1 Organização da memória principal

Como a memória principal é o local onde os dados e as instruções de um programa ficam armazenados para serem utilizados pelo processador durante a sua execução, é preciso ficar claro que esse conteúdo precisa estar organizado em uma estrutura padrão que permita a identificação do local onde estão armazenados cada um dos seus itens (ex.: uma instrução ou um dado).

Assim, a memória principal encontra-se organizada em um conjunto de células, sendo que cada uma delas representa o agrupamento de uma quantidade de *bits*. Cada célula caracteriza uma unidade de armazenamento na memória e possui um endereço único, o qual é utilizado pelo processador para acessar seu conteúdo. Portanto, a célula é a menor unidade endereçável em um computador. Monteiro (2007) complementa essa explicação afirmando que cada célula é constituída de um conjunto de circuitos eletrônicos, baseados em semicondutores, que permitem o armazenamento de valores 0 ou 1, os quais representam um dado ou uma instrução.

A quantidade de *bits* que pode ser armazenada em uma célula é definida pelo fabricante. Uma célula contendo N *bits* permite o armazenamento de 2^N combinações de valores, o que representará a quantidade de células possíveis na memória. Um tamanho comum de célula adotado pelos fabricantes é 8 *bits* (1 *byte*).

Se for possível armazenar em uma memória de 2^N combinações possíveis de células (cada uma delas contendo dados armazenados), então será possível calcular a capacidade de armazenamento da memória principal, da seguinte forma:

- a) se $N = 9$ *bits*, tem-se que $2^9 = 512$ (células de memória);
- b) se cada célula pode armazenar 8 *bits*, tem-se que: $512 \times 8 = 4\text{KB}$ (4 quilo *byte*) de espaço em memória.

O acesso a cada posição (célula) de memória pode ser feito de modo aleatório, proporcionando grande flexibilidade, graças à sua tecnologia de fabricação. São duas as operações que podem ser realizadas em uma memória: escrita (*write*) – para o armazenamento de dados na memória – e leitura (*read*) – para a recuperação de dados e instruções armazenados na memória.

Pesquisem em *sites* da internet por especificações de computadores que contenham cada um dos tipos de memória apresentados acima. Se possível, copie a figura que representa o equipamento e, logo em seguida, apresente suas características (nome, modelo, ano de fabricação, tipo de memória que possui e quantidade). Esse conteúdo deve ser postado em um *blog* a ser criado no AVEA.



O termo palavra também é utilizado ao se tratar de memória de um computador, mas não deve ser confundido com célula, pois palavra é utilizada para definir a unidade de transferência e processamento e o número de *bits* que podem ser transferidos entre a memória principal e a CPU para processamento. Acerca desse tema, Monteiro (2007) afirma que a memória principal deve ser organizada em um conjunto sequencial de palavras, cada uma diretamente acessível pelo processador.

4.4.2 Memória ROM

A memória ROM (*Read Only Memory*) também é considerada uma memória principal, mas apresenta algumas diferenças em relação à memória RAM. A primeira delas é o fato de ser uma memória somente de leitura, ou seja, seu conteúdo é escrito uma vez e não é mais alterado, apenas consultado. Outra característica das memórias ROM é que elas são do tipo não voláteis, isto é, os dados gravados não são perdidos na ausência de energia elétrica ao dispositivo.

É dito que um *software* que é armazenado em uma memória ROM passa a ser chamado de *firmware*. Em um computador existem diversos *software* desse tipo disponíveis em memórias ROM, pois não podem ser apagados ao desligar o computador e devem ficar disponíveis sempre que for necessário.

Dessa forma, as memórias ROM são aplicadas em um computador para armazenar três programas principais (TORRES, 2010):

- a) BIOS (*Basic Input Output System*): ou Sistema Básico de Entrada e Saída, é responsável por ensinar o processador da máquina a operar com os dispositivos básicos de entrada e saída;

A-Z

CMOS

CMOS: Uma abreviação de Complementary Metal Oxide Semiconductor. É uma tecnologia de construção de circuitos integrados. CMOS é uma pequena área de memória volátil, alimentada por uma bateria, que é usada para gravar as configurações do setup na placa-mãe.

- b) POST (*Power On Self Test*): Autoteste – programa de verificação e teste que se executa após a ligação do computador, realizando diversas ações sobre o *hardware* (ex.: contagem de memória);
- c) SETUP: Programa que altera os parâmetros armazenados na memória de configuração (**CMOS**).

De acordo com Alecrim (2010), as memórias ROM podem ser classificadas em:

- a) PROM (*Programmable Read-Only Memory*): este é um dos primeiros tipos de memória ROM. A gravação de dados neste tipo é realizada por meio de aparelhos que trabalham através de uma reação física com elementos elétricos. Uma vez que isso ocorre, os dados gravados na memória PROM não podem ser apagados ou alterados;
- b) EPROM (*Erasable Programmable Read-Only Memory*): as memórias EPROM têm como principal característica a capacidade de permitir que dados sejam regravados no dispositivo. Isso é feito com o auxílio de um componente que emite luz ultravioleta. Nesse processo, os dados gravados precisam ser apagados por completo. Somente após esse procedimento uma nova gravação pode ser realizada;
- c) EEPROM (*Electrically-Erasable Programmable Read-Only Memory*): este tipo de memória ROM também permite a regravação de dados; no entanto, ao contrário do que acontece com as memórias EPROM, os processos para apagar e gravar dados são feitos eletricamente, fazendo com que não seja necessário mover o dispositivo de seu lugar para um aparelho especial para que a regravação ocorra;
- d) EARAM (*Electrically-Alterable Programmable Read-Only Memory*): as memórias EARAM podem ser vistas como um tipo de EEPROM. Sua principal característica é o fato de que os dados gravados podem ser alterados aos poucos, razão pela qual esse tipo é geralmente utilizado em aplicações que exigem apenas reescrita parcial de informações;
- e) *Flash*: as memórias *Flash* também podem ser vistas como um tipo de EEPROM; no entanto, o processo de gravação (e regravação) é muito mais rápido. Além disso, memórias *Flash* são mais duráveis e podem guardar um volume elevado de dados. Trata-se do tipo de memória utilizada em *pen-drive*;
- f) CD-ROM, DVD-ROM e afins: essa é uma categoria de discos ópticos onde os dados são gravados apenas uma vez, seja de fábrica, como os CDs de músicas, ou com dados próprios do usuário, quando este efetua a gravação. Há também uma categoria que pode ser comparada ao tipo EEPROM, pois permite a regravação de dados: CD-RW e DVD-RW e afins.

4.5 Memória secundária

A memória secundária também é denominada de memória de massa, por possuir uma capacidade de armazenamento muito superior à das outras memórias conforme discutido neste tópico. Outra característica que difere a memória secundária das outras memórias é o fato de ser permanente (não volátil), ou seja, não perde o conteúdo armazenado caso o computador seja desligado. Por estar na base da pirâmide (Figura 4.1), apresenta o menor custo por *byte* armazenado.

Este tipo de memória não possui acesso direto pelo processador, sempre havendo a necessidade de carregamento de dados dos dispositivos de memória secundária para a memória principal, para que então sejam enviados ao processador.

Monte um arquivo com imagens de memórias secundárias, seguidas da descrição de características de cada uma delas (marca, modelo, capacidade de armazenamento, valor aproximado, aspectos de velocidade (ex.: rpm, taxa de transferência). Esse arquivo deverá ser postado como atividade no AVEA.



De acordo com Monteiro (2007), a memória secundária pode ser constituída por diferentes tipos dispositivos, alguns diretamente ligados ao sistema para acesso imediato (ex.: discos rígidos) e outros que podem ser conectados quando desejado (ex.: *pen-drive*, CD/DVD).

Em relação à tecnologia de fabricação, existe uma variedade muito grande de tipos, assim como a variedade de dispositivos que se enquadra nessa categoria de memória. Para cada dispositivo, existem diferentes tecnologias de fabricação, não sendo possível abordar todas nessa seção.



Leia o artigo: "Memórias", disponível em:
<http://www.gdhpress.com.br/hardware/leia/index.php?p=cap4-1>.
Acesso em: 19 jul. 2010

Resumo

Nesta aula discutimos o subsistema de memória de um sistema computacional, caracterizando os diferentes tipos de memórias disponíveis.

Nesse sentido, foram abordadas as memórias permanentes ou não voláteis e as voláteis.

As do primeiro tipo mantêm o conteúdo armazenado mesmo na ausência de energia, sendo elas: memória secundária (ex.: disco rígido, CD/DVD, *pen-drive*) e a memória primária do tipo ROM (*Read Only Memory*).

As do segundo tipo apagam o conteúdo armazenado caso o computador seja desligado, ou seja, quando cessar o fornecimento de energia. Enquadram-se nessa categoria: os registradores, as memórias *cache* e a memória principal do tipo RAM.

Atividades de aprendizagem

Responda às questões a seguir sobre o conteúdo desta aula. As questões a, b, c e d devem ser respondidas em um documento de texto e postadas como atividade no AVEA. Já as questões e e f deverão ser respondidas e comentadas no fórum sobre “Memórias de um computador”, também disponível no AVEA.

- a) Quais são os dois tipos principais de memória? Explique.
- b) O que são memórias voláteis e não voláteis? Exemplifique.
- c) Qual a função da memória RAM (memória principal) no computador? É possível que um computador funcione utilizando-se apenas dos demais tipos de memória? Explique.
- d) O que diferencia uma memória que se utiliza da tecnologia DRAM de outra que se utiliza da SRAM?
- e) Faça uma análise dos diversos tipos de memória em relação aos fatores velocidade e custo.
- f) Conceitue memória *cache* e fale da atuação dela no desempenho do computador.

Aula 5 – O Processador: organização e arquitetura

Objetivos

Analisar a organização básica do processador.

Analisar e compreender a organização de microprocessadores atuais.

Compreender o funcionamento interno e a execução de instruções em baixo nível.

5.1 Organização do processador

Na Aula 1 estudamos os componentes básicos da arquitetura de um computador, segundo o modelo de Von Neumann, cuja proposta vale a pena relembrar no início desta seção, a qual irá tratar especificamente de um dos elementos dessa arquitetura: o processador. A proposta de Von Neumann para a construção de um computador previa que:

- a) Codificasse instruções que pudessem ser armazenadas na memória e sugeriu que se usassem cadeias de uns e zeros (binário) para codificá-lo;
- b) Armazenasse na memória as instruções e todas as informações que fossem necessárias para a execução da tarefa desejada;
- c) Ao processar o programa, as instruções fossem buscadas diretamente na memória.

A Unidade Central de Processamento (UCP) é responsável pelo processamento e execução de programas armazenados na memória principal, buscando suas instruções, examinando-as e, então, executando uma após a outra. Cabe lembrar que sempre que houver referências à sigla UCP, está se fazendo referência ao processador do computador.

A UCP é composta por várias partes distintas, entre elas: registradores, Unidade de Controle (UC) e Unidade Lógica Aritmética (ULA).

A UCP pode ser dividida em duas categorias funcionais, as quais podem ser chamadas de unidade, conforme segue: **Unidade Funcional de Controle** e **Unidade Funcional de Processamento**. A Figura 5.1 apresenta um diagrama funcional básico da UCP, o qual mostra os elementos essenciais para o seu funcionamento – em azul os componentes da Unidade Funcional de Controle e em amarelo claro os elementos da Unidade Funcional de Processamento.

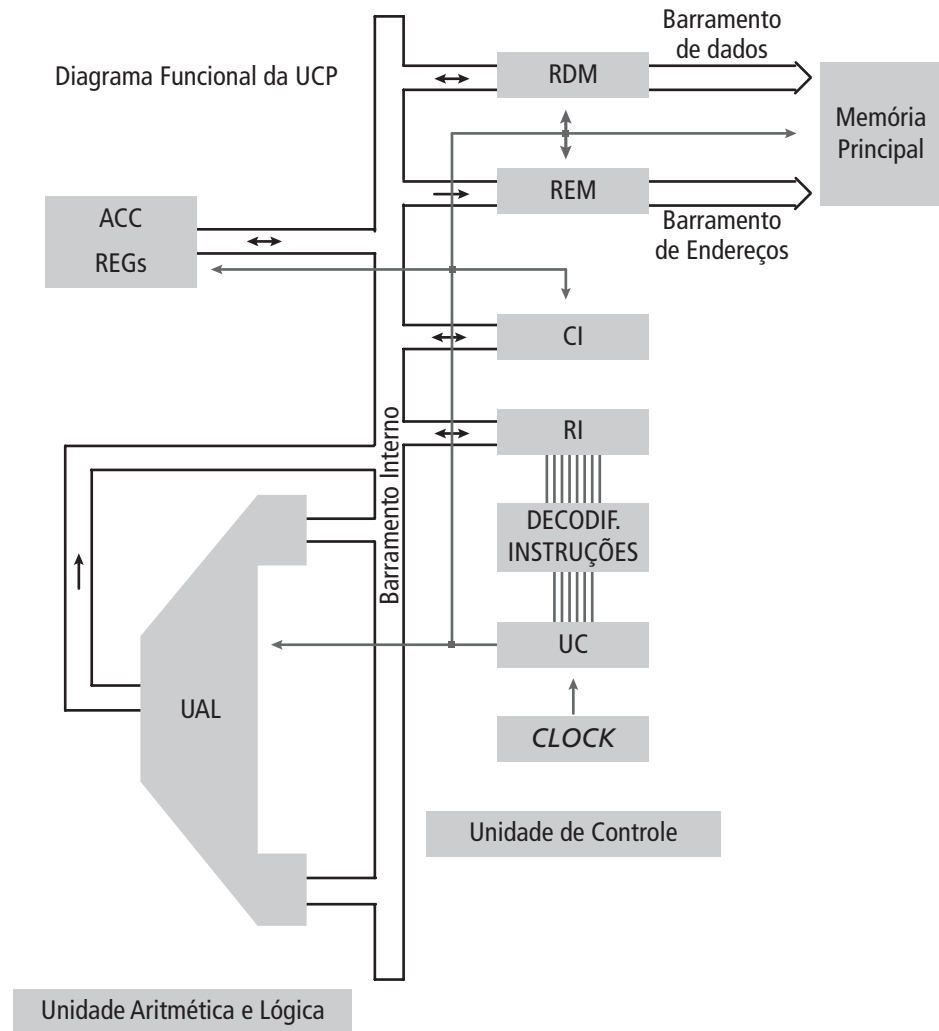


Figura 5.1: Diagrama funcional da CPU

Fonte: <http://www.users.rdc.puc-rio.br/rmano/comp8ucp.html>

A Unidade Funcional de Processamento é composta pelos seguintes elementos: Registradores, ACC, UAL. A Unidade Funcional de Controle é composta pelos seguintes elementos: RDM, REM, CI, RI, Decodificador de Instruções, UC, Clock (relógio).

Os componentes do processador são interligados por meio de um barramento, que consiste em um conjunto de fios paralelos que permitem a transmissão de dados, endereços e sinais de controle entre a UCP, memória e dispositivos de entrada e saída. Existem barramentos externos ao processador, cuja função é conectá-lo à memória e aos dispositivos de entrada/saída, além dos barramentos internos à UCP.

A seguir apresentamos de forma detalhada cada um dos elementos que compõem as unidades funcionais da UCP, conforme citados acima.

5.2 Unidade funcional de processamento

O processamento de dados é a ação de manipular um ou mais valores (dados) em certa sequência de ações, de modo a produzir um resultado útil. Para o autor, processar dados é a finalidade do sistema computacional e consiste em executar uma ação, com os dados, que produza algum tipo de resultado. Algumas das tarefas mais comuns da função processamento são: operações aritméticas (somar, subtrair, multiplicar, dividir); operações lógicas (AND, OR, XOR, entre outras) e movimentação de dados entre a UCP e a memória e vice-versa, entre outras (MONTEIRO, 2007).

A seguir serão apresentados os dispositivos que compõem a Unidade Funcional de Processamento (ULA) e registradores, os quais estão em destaque na Figura 5.2.

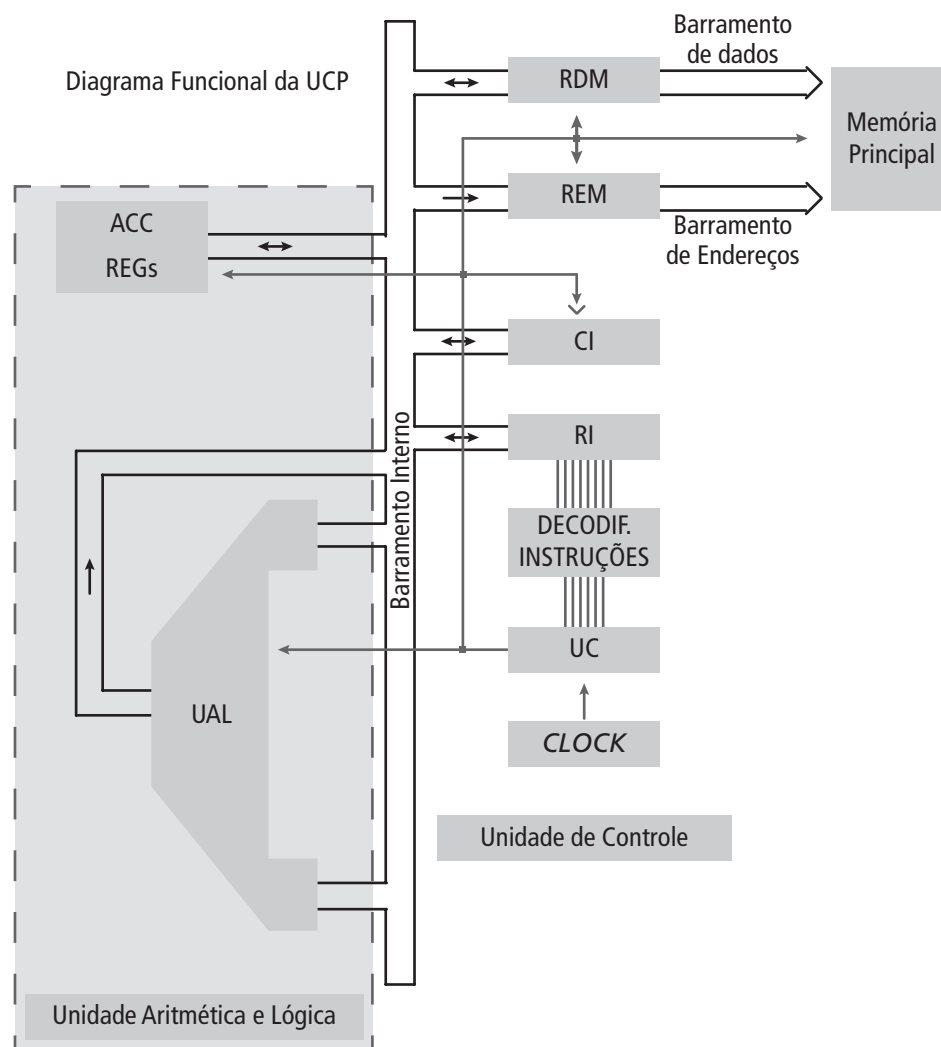


Figura 5.2: Diagrama funcional da CPU – componentes de processamento

Fonte: <http://www.users.rdc.puc-rio.br/rmano/comp8ucp.html>

5.2.1 Unidade lógica e aritmética (ULA)

A função efetiva deste dispositivo é a execução das instruções dos programas que se encontram armazenadas na memória. Ao chegarem à UCP, essas instruções são interpretadas e traduzidas em operações matemáticas a serem executadas pela ULA.

Podemos dizer que a ULA é um aglomerado de circuitos lógicos e componentes eletrônicos simples que, integrados, realizam as operações aritméticas e lógicas. São exemplos de operações executadas pela ULA: soma, multiplicação, operações lógicas (AND, OR, NOT, XOR, entre outras), incremento, decremento e operação de complemento.

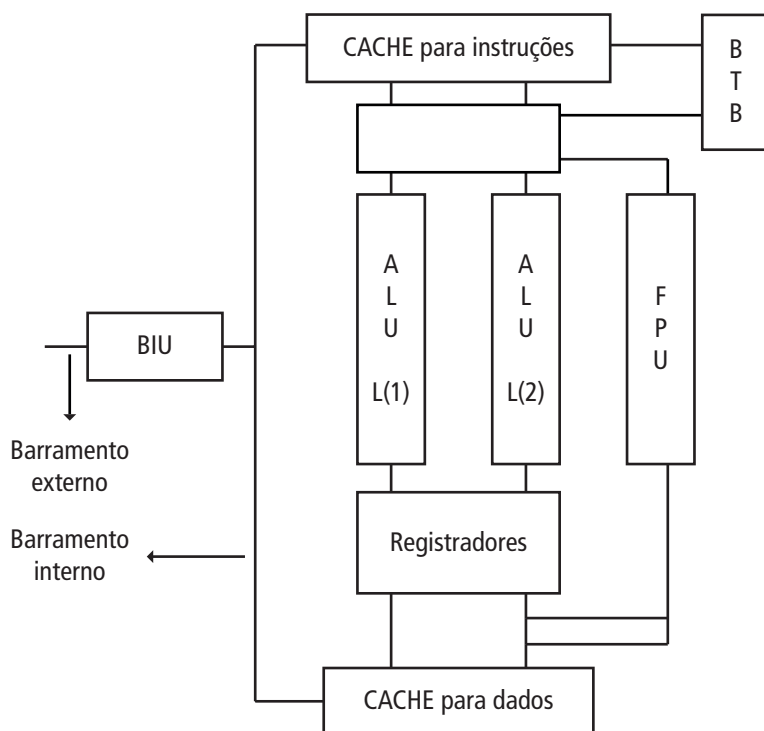


Figura 5.3: Arquitetura básica da UCP do *Pentium* original

Fonte: Monteiro (2007)

A ULA é uma pequena parte do circuito integrado da CPU, utilizada em pequenos sistemas, ou pode compreender um considerável conjunto de componentes lógicos de alta velocidade. De acordo com Monteiro (2007), isso pode ser constatado em processadores mais modernos, os quais utilizam em sua arquitetura mais de uma ULA, de modo a tornar a execução das instruções mais rápida. Por exemplo: os processadores *Pentium* possuem três ULAs, duas delas para processamento de números inteiros e a terceira para números fracionários, conforme apresenta a Figura 5.3.

5.2.2 Registradores

São elementos de armazenamento temporário, localizados na UCP, os quais são extremamente rápidos por causa da sua tecnologia de fabricação (conforme apresentado na Aula 4). Assim, as UCPs são fabricadas com certa quantidade de registradores destinados ao armazenamento de dados que estão sendo utilizados durante o processamento e, portanto, servem de memória auxiliar básica da ULA.

A quantidade e o emprego dos registradores variam bastante de modelo para modelo de processador. Devido à sua tecnologia de construção e por estarem localizados no interior da UCP, são muito caros e, por isso, disponíveis em quantidade limitada.

Os sistemas mais antigos possuíam um registrador especial chamado acumulador ou ACC (de *accumulator*), o qual, além de armazenar dados, servia de elemento de ligação entre a ULA e os demais dispositivos da UCP (MONTEIRO, 2007). Nos computadores mais simples é encontrado apenas um acumulador, conforme apresentado na Figura 5.2. Em arquiteturas mais complexas, vários registradores podem desempenhar as funções de um acumulador, além de haver diversos registradores de dados de uso geral.

Outro fator importante relacionado aos registradores é o tamanho da palavra, a qual está vinculada ao projeto de fabricação da UCP, correspondendo ao tamanho dos elementos ligados à área de processamento, a ULA e os registradores de dados. A capacidade de processamento de uma UCP, ou seja, sua velocidade, é bastante influenciada pelo tamanho da palavra. Atualmente há computadores referenciados como tendo uma arquitetura de 32 *bits* ou uma arquitetura de 64 *bits*, o que corresponde ao tamanho de sua palavra.



Pesquise na internet sobre o número de ULAs utilizadas nos processadores atuais bem como sobre o tamanho da palavra de um computador (UCP). Procure anotar dados relativos a esses aspectos e faça a postagem deles no fórum sobre “Projeto de UCP – ULA e registradores” disponível no AVEA. Observe as postagens dos colegas e faça comentários a respeito.

5.3 Unidade funcional de controle

Conforme apresentado no início desta aula, a Unidade Central de Processamento (UCP) é responsável pelo processamento e execução de programas armazenados na memória principal, sendo que a ULA é o elemento da UCP responsável pela execução das operações solicitadas. Dessa forma, a Unidade Funcional de Controle é responsável pela realização das seguintes atividades (MONTEIRO, 2007):

- a) busca da instrução que será executada, armazenando-a em um registrador da UCP;
- b) interpretação das instruções a fim de saber quais operações deverão ser executadas pela ULA (ex.: soma, subtração, comparação) e como realizá-las;
- c) geração de sinais de controle apropriados para a ativação das atividades necessárias à execução propriamente dita da instrução identificada. Esses sinais de controle são enviados aos diversos componentes do sistema,

sejam eles internos à UCP (ex.: a ULA) ou externos (ex.: memória e dispositivos de entrada e saída).

A seguir apresentamos os dispositivos que compõem a Unidade Funcional de Controle – RDM, REM, CI, RI, Decodificador de Instruções, UC, *Clock*, os quais estão em destaque na Figura 5.4.

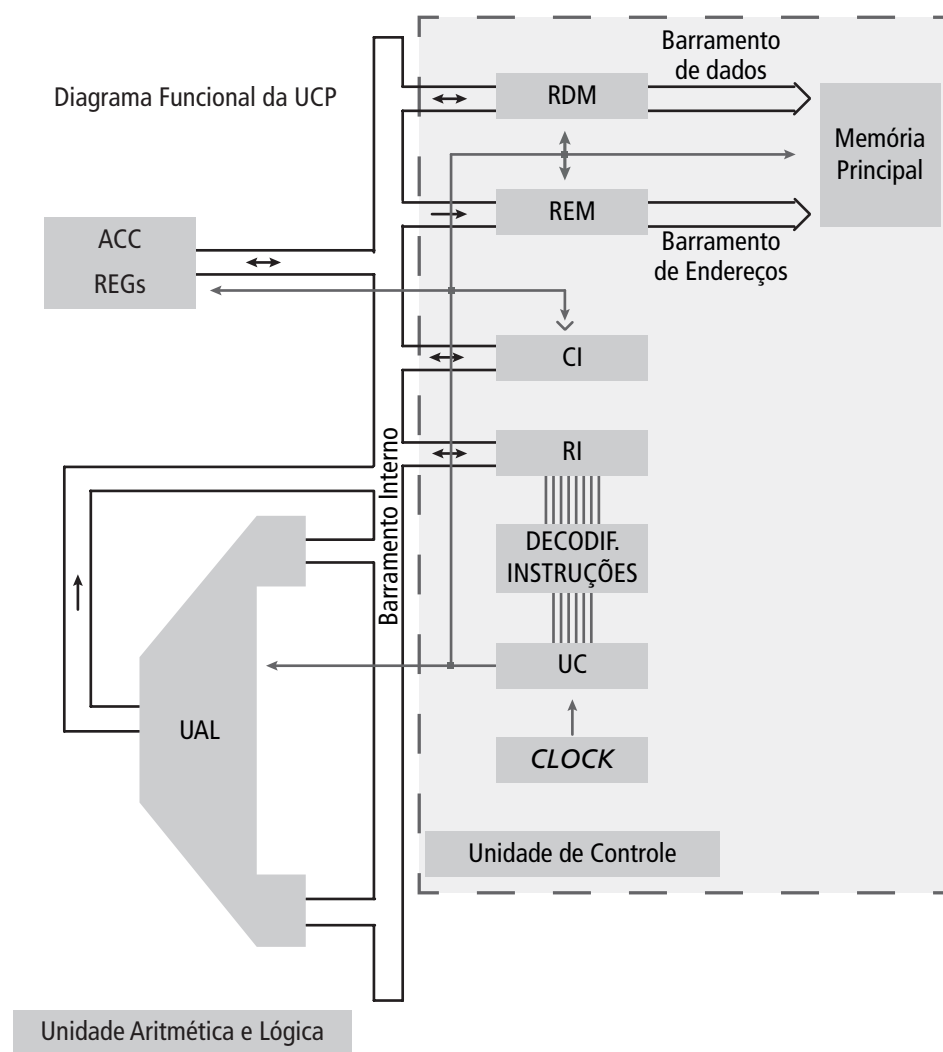


Figura 5.4: Diagrama funcional da CPU – componentes de controle

Fonte: <http://www.users.rdc.puc-rio.br/rmano/comp8ucp.html>

5.3.1 Registrador de dados de memória (RDM) e de endereços de memória (REM)

O RDM, também chamado de MBR (*Memory Buffer Register*), é um registrador que armazena temporariamente dados (conteúdo de uma ou mais células) que estão sendo transferidos da memória principal para a UCP (em uma operação de leitura) ou da UCP para a memória principal (em uma

operação de escrita). Em seguida, o referido dado é reencaminhado para outro elemento da UCP para processamento ou para uma célula da memória principal, se for um resultado de um processamento. A quantidade de *bits* que pode ser armazenada no RDM é a mesma quantidade suportada pelo barramento de dados.

O REM, também chamado de MAR (*Memory Address Register*), é um registrador que armazena temporariamente o endereço de acesso a uma posição de memória, necessário ao se iniciar uma operação de leitura ou de escrita. Em seguida, o referido endereço é encaminhado à controladora da memória, principal identificação e localização da célula desejada. Permite armazenar a mesma quantidade de bits do barramento de endereço.

5.3.2 Contador de instruções (CI)

Este registrador é também denominado de *Program Counter* (PC) ou contador de programa. De acordo com Tanenbaum (2007), é o CI cujo valor aponta para a próxima instrução a ser buscada da memória a ser executada no processador. De acordo com Monteiro (2007), tão logo a instrução que vai ser executada seja buscada da memória principal para a CPU, o sistema automaticamente efetiva a modificação do conteúdo do CI de modo que ele passe a armazenar o endereço da próxima instrução na sequência. Assim, o CI é um registrador crucial para o processo de controle e de sequenciamento da execução dos programas.

5.3.3 Registrador de instruções (RI)

Este registrador tem a função de armazenar a instrução a ser executada pela UCP. Ao se iniciar um ciclo de instrução (MONTEIRO, 2007), a UC emite sinais de controle em sequência no tempo, de modo que se processe a realização de um ciclo de leitura para buscar a instrução na memória. Ao final do ciclo de leitura a instrução desejada será armazenada no RI, via barramento de dados e RDM. A Figura 5.5 mostra o RI ligado diretamente ao decodificador de instruções, o qual irá interpretar a instrução e avisar à Unidade de Controle (UC).

5.3.4 Decodificador de instruções

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, e é função do decodificador de instrução identificar que operação será realizada, correlacionada à instrução cujo código de operação foi decodificado. Assim, o RI irá passar ao decodificador uma sequência de *bits* representando uma instrução a ser executada.

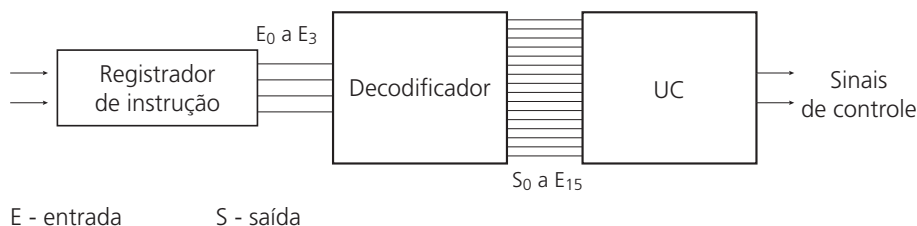


Figura 5.5: Diagrama em bloco da decodificação em uma UCP

Fonte: Adaptado de Monteiro (2007)

Um decodificador possui 2^n saídas, sendo n a quantidade de algarismos binários do valor de entrada. A Figura 5.5 mostra um diagrama em blocos do processo de decodificação na UCP, no qual o RI passa um código de instrução ao decodificador de tamanho de 4 bits, que é decodificado (interpretado) e encaminhado à UC para que ela emita os sinais de controle para os demais elementos da UCP.

O componente decodificador foi incorporado à UCP com o advento das máquinas CISC (Complex Instruction Set Computer) e trata-se de uma categoria de arquitetura de processadores que favorece um conjunto simples e pequeno de instruções de máquinas. Uma instrução de máquina é uma operação básica que o hardware realiza diretamente. Este assunto será discutido em mais detalhes nas seções subsequentes.

5.3.5 Registrador de instruções (RI)

Esses registradores têm como função controlar a execução das instruções e os demais componentes da UCP. Dispositivo que possui a lógica necessária para realizar a movimentação de dados e de instruções da/para a CPU, através de sinais de controle que emite em instantes de tempo programados.

Os sinais de controle 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 denominado relógio (*clock*).

5.3.6 Relógio (*clock*)

Monteiro (2007) define este dispositivo como um gerador de pulsos, cuja duração é chamada de *ciclo*, e a quantidade de vezes que esse pulso básico se repete em um segundo define a unidade de medida do relógio, denominada **frequência**, a qual também é usada para definir a velocidade na CPU.

A unidade de medida utilizada para a frequência do relógio da UCP é o *hertz* (Hz), que significa um ciclo por segundo.

A-Z

Ciclo de instrução:

É o período de tempo no qual a UCP lê e processa uma instrução armazenada na memória, ou ainda, pode ser definido pela sequência de ações que a UCP realiza para executar cada instrução de um programa.

A cada pulso é realizada uma operação elementar, durante o ciclo de uma instrução (ex.: busca de dados, envio da instrução para o RI, sinal de controle).

Como os computadores atuais apresentam frequências bastante elevadas, utiliza-se a medida de milhões de ciclos por segundo (*mega-hertz* – MHz) ou bilhões de ciclos por segundo (*giga-hertz* – GHz).

5.3.7 Barramentos

Os diversos componentes de um computador se comunicam através de barramentos, os quais se caracterizam como um conjunto de condutores elétricos que interligam os diversos componentes do computador e de circuitos eletrônicos que controlam o fluxo dos *bits*. O barramento conduz de modo sincronizado o fluxo de informações (dados e instruções, endereços e controles) de um componente para outro ao longo da placa-mãe. O barramento organiza o tráfego de informações observando as necessidades de recursos e as limitações de tempo de cada componente, de forma que não ocorram colisões, ou mesmo, algum componente deixe de ser atendido (MONTEIRO, 2007).

O barramento de um sistema computacional, denominado barramento do sistema, é o caminho por onde trafegam todas as informações dentro do computador. Esse barramento é formado basicamente por três vias específicas: barramento de dados, barramento de endereços e barramento de controle, conforme mostra a Figura 5.6.

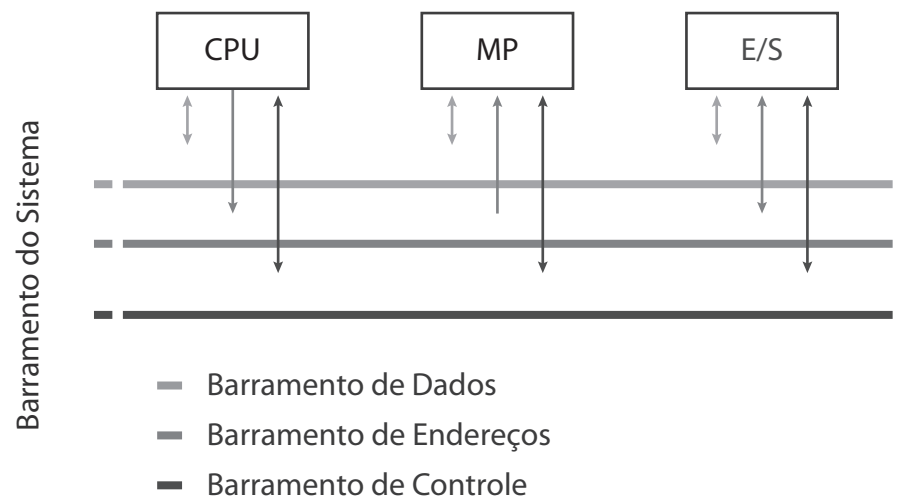


Figura 5.6: Barramento do sistema

Fonte: Adaptada de Murdocca (2000)

a) Barramento de dados

Este barramento interliga o RDM (localizado na UCP) à memória principal, para transferência de instruções ou dados a serem executados. É bidirecional, isto é, ora os sinais percorrem o barramento vindo da UCP para a memória principal (operação de escrita), ora percorrem o caminho inverso (operação de leitura). Possui influência direta no **desempenho** do **sistema**, pois, quanto maior a sua largura, maior o número de *bits* (dados) transferidos por vez e consequentemente mais rapidamente esses dados chegarão ao seu destino (UCP ou memória).

Os primeiros computadores pessoais (ex.: PC-XT) possuíam barramento de dados de oito vias, ou seja, capaz de transferir oito *bits* por vez. Atualmente, conforme a arquitetura do processador, podem existir barramento de dados de 32, 64 ou 128 *bits*.

b) Barramento de endereços

Interliga o REM (localizado na UCP) à memória principal, para transferência dos *bits* que representam um determinado endereço de memória onde se localiza uma instrução ou dado a ser executado. É unidirecional, visto que somente a UCP aciona a memória principal para a realização de operações de leitura ou escrita. Possui tantas vias de transmissão quantos são os *bits* que representam o valor de um endereço.

Seguindo o exemplo anterior, no antigo PC-XT, este barramento possuía 20 linhas: com isso era possível utilizar endereços de no máximo 20 *bits*. Logo, o maior endereço possível, será:

$$2^{20} = 1.048.576 \text{ Bytes} = 1 \text{ MB}$$

Dessa forma, a capacidade de armazenamento da memória RAM poderá ser de no máximo 1MB. É possível afirmar que o tamanho do barramento de endereços determina a quantidade máxima de armazenamento de dados que a memória principal pode dispor. Atualmente, os barramentos dispõem de significativa capacidade de armazenamento (ex.: 32,64, 128 *bits*), possibilitando grandes espaços para armazenamento na memória.

c) Barramento de controle

Interliga a UCP, mais especificamente a Unidade de Controle (UC), aos demais componentes do sistema computacional (memória principal, componentes de entrada e de saída) para passagem de sinais de controle gerados pelo sistema. São exemplos de sinais de controle: leitura e escrita de dados na memória principal, leitura e escrita de componentes de entrada e saída, certificação de transferência de dados – o dispositivo acusa o término da transferência para a UCP, pedido de interrupção, relógio (*clock*) – por onde passam os pulsos de sincronização dos eventos durante o funcionamento do sistema (MONTEIRO, 2007).

A-Z

Interrupção:

É um aviso gerado à UCP comunicando que um dispositivo deseja realizar uma operação (ex.: leitura de dados de um pen-drive, transferência de dados do disco rígido para a memória, chegada de um pacote de dados da rede).

É bidirecional, porque a UCP, por exemplo, pode enviar sinais de controle para a memória principal, como um sinal indicador de que deseja uma operação de leitura ou de escrita, e a memória principal pode enviar sinais do tipo *wait* (espere), para a UCP aguardar o término de uma operação.

Os barramentos compartilham suas vias de comunicação (normalmente fios de cobre) entre diversos componentes neles conectados como mostra a Figura 5.6. Nesse caso, somente é permitida a passagem de um conjunto de *bits* de cada vez e, por esse motivo, o controle do barramento torna-se um processo essencial para o funcionamento adequado do sistema.

De acordo com Monteiro (2007), no modelo apresentado na Figura 5.6, há um único barramento de dados, endereços e controle interconectando todos os componentes do computador. Isso se justifica pela grande diferença de características dos diversos componentes existentes, principalmente periféricos (ex.: a velocidade de uma transferência de dados de um teclado é muitas vezes menor que a velocidade de transferência de dados de um disco magnético).

Considerando esse fato, os projetistas de sistemas de computação criaram diversos tipos de barramento, apresentando taxas de transferência de bits diferentes e apropriadas às velocidades dos componentes interconectados (ex.: UCP, memória, disco rígido, teclado). Nesse caso há uma hierarquia em que os barramentos são organizados e, de acordo com Monteiro (2007), atualmente os modelos de organização de sistemas de computação adotados pelos fabricantes possuem diferentes tipos de barramentos:

- a) Barramento local: possui maior velocidade de transferência de dados, funcionando normalmente na mesma frequência do relógio do proces-

sador. Este barramento costuma interligar o processador aos dispositivos de maior velocidade (visando não atrasar as operações do processador): memória *cache* e memória principal;

- b)** Barramento do sistema: podemos dizer que se trata de um barramento opcional, adotado por alguns fabricantes, fazendo com que o barramento local faça a ligação entre o processador e a memória *cache* e esta se interligue com os módulos de memória principal (RAM) através do chamado barramento do sistema, de modo a não permitir acesso direto do processador à memória principal. Um circuito integrado denominado ponte (*chipset*) sincroniza o acesso entre as memórias;
- c)** Barramento de expansão: também chamado de barramento de entrada e de saída (E/S), é responsável por interligar os diversos dispositivos de E/S aos demais componentes do computador, tais como: monitor de vídeo, impressoras, CD/DVD, etc. Também se utiliza de uma ponte para se conectar ao barramento do sistema; as pontes sincronizam as diferentes velocidades dos barramentos.

Considerando a acentuada diferença de velocidade apresentadas pelos diversos dispositivos de E/S existentes atualmente, a indústria de computadores tem desenvolvido alternativas visando maximizar o desempenho nas transferências de dados entre dispositivos (ex.: entre disco e memória principal). A alternativa encontrada foi separar o barramento de expansão (ou de E/S) em dois, sendo um de alta velocidade para dispositivos mais rápidos (ex.: redes, placas gráficas) e outro de menor velocidade para dispositivos mais lentos (ex.: teclado, *modems*, *mouse*). A Figura 5.7 apresenta este conceito (MONTEIRO, 2007).

A-Z

Chipset:

É um conjunto de circuitos integrados ou chips, que são projetados para trabalhar em conjunto, sendo considerados circuitos de apoio na placa-mãe para controlar a interconexão entre os seus componentes.

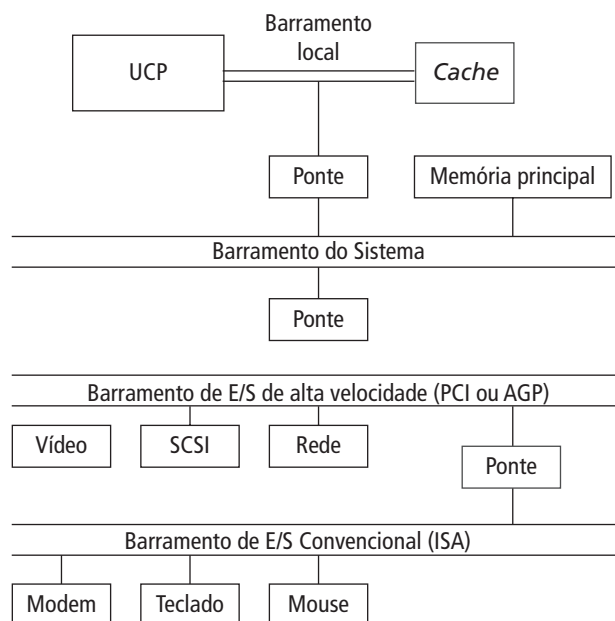


Figura 5.7: Exemplo de barramento de maior desempenho usado atualmente

Fonte: Monteiro (2007)

Não importa o tipo de barramento, um fator importante que influencia no desempenho do sistema computacional é a **largura** (tamanho) do barramento, que diz respeito à quantidade de informações (no formato de *bits*) que poderão ser transmitidas simultaneamente por ele. Podemos dizer que é a quantidade de fios ou de vias que um barramento apresenta que vai caracterizar sua largura.

Quando se fala em quantidade de bits que podem trafegar em um barramento, fala-se em taxa de transferência, que revela a medida dessa quantidade, a qual é especificada em bits por segundo (normalmente K *bits*, M *bits*, etc.).

Conforme discutido anteriormente, cada um dos barramentos permite o seu compartilhamento com os demais componentes do sistema, especialmente o barramento de expansão, que é compartilhado com diversos dispositivos de entrada e saída. Para que isso ocorra, é indispensável um mecanismo de controle de acesso baseado em regras, que garanta que quando um dos dispositivos estiver utilizando o barramento, os demais componentes deverão aguardar a sua liberação.

Esse mecanismo de controle de acesso é denominado protocolo. Isso faz com que um barramento não seja composto somente de fios condutores, mas também de um protocolo (MONTEIRO, 2007).

Os fabricantes de computadores têm procurado uma padronização na definição de protocolos, de forma a evitar que cada um crie o seu próprio, com características diferentes dos demais, o que tornaria muito inflexível o uso de certos componentes (ex.: vídeo, impressoras, discos rígidos) disponibilizados no mercado. Nesse cenário, não importa o fabricante do componente nem suas características físicas, desde que ele esteja de acordo com os padrões de protocolos definidos pela indústria de computadores.

Sendo assim, muitos padrões de barramento de expansão foram desenvolvidos ao longo do tempo, alguns deles já não mais utilizados. Os mais populares são apresentados a seguir, conforme Monteiro (2007):

- a) ISA (*Industry Standard Adapter*): desenvolvido pela IBM. Apresenta uma taxa de transferência baixa, mas apesar disso, foi adotado por toda a indústria. Os sistemas atuais não mais o empregaram;
- b) PCI (*Peripheral Component Interconnect*): desenvolvido pela Intel, tornando-se quase um padrão para todo o mercado, como barramento de alta velocidade. Permite transferência de dados em 32 ou 64 *bits* a velocidades de 33 MHz e de 66 MHz. Cada controlador permite cerca de quatro dispositivos;
- c) USB (*Universal Serial Bus*): tem a característica particular de permitir a conexão de muitos periféricos simultaneamente (pode conectar até 127 dispositivos em um barramento – por meio de uma espécie de centralizador) ao barramento; este, por uma única porta (conector), conecta-se à placa-mãe. Grande parte dos dispositivos USB é desenvolvida com a característica de eles serem conectados ao computador e utilizados logo em seguida, o que é chamado de *plug-and-play*;
- d) AGP (*Accelerated Graphics Port*): barramento desenvolvido por vários fabricantes, porém liderados pela Intel, com o objetivo de acelerar as transferências de dados do vídeo para a memória principal, especialmente dados em 3D (terceira dimensão), muito utilizados em aplicativos gráficos (ex.: jogos);
- e) PCI Express (*Peripheral Component Interconnect Express*): esse barramento foi construído por um grupo de empresas denominado PCI-SIG (*Peripheral Component Interconnect Special Interest Group*), composto por empresas como a *Intel*, *AMD*, *IBM*, *HP* e *Microsoft*. Este barramento



"Barramentos: ISA, AGP, PCI, PCI Express, AMR e outros". Disponível em: <http://www.infowester.com/barramentos.php>. Acesso em: 19 jul. 2010.



"Tecnologia USB (Universal Serial Bus)". Disponível em: <http://www.infowester.com/usb.php>. Acesso em: 19 jul. 2010.



"Tecnologia PCI Express". Disponível em: <http://www.infowester.com/pciexpress.php>. Acesso em: 19 jul. 2010

veio para atender às demandas por mais velocidade gerada por novos chips gráficos e tecnologias de rede apresentando altas taxas de transferência. Assim, o PCI e o AGP foram substituídos pelo *PCI Express*. Nesse barramento, a conexão entre dois dispositivos ocorre de modo ponto a ponto (exclusivo) – comunicação serial. Por esse motivo, o *PCI Express* não é considerado um barramento propriamente dito (considerando que barramento é um caminho de dados onde você pode ligar vários dispositivos ao mesmo tempo, compartilhando-o). Essa característica é que o faz ser o meio de comunicação entre dois dispositivos de um computador mais rápido atualmente. Até o momento existiram três versões desse barramento (1.0 – lançado em 2004; 2.0 – lançado em 2007; e o 3.0 – lançado em 2010).

Cada barramento possui um protocolo padrão que é utilizado pela indústria de computadores para a fabricação de todos os dispositivos de entrada e saída a serem conectados nos diferentes tipos de barramento. Dessa forma, foram desenvolvidos os chamados *slots*. Um *slot* nada mais é do que um orifício ou um encaixe padronizado inserido na placa-mãe dos computadores de maneira que os diversos dispositivos possam ser encaixados, desde que atendam a um dos padrões disponíveis nela. Assim, nas Figuras 5.8, 5.9 e 5.10 são apresentados alguns slots correspondentes aos padrões de barramentos de expansão listados acima.

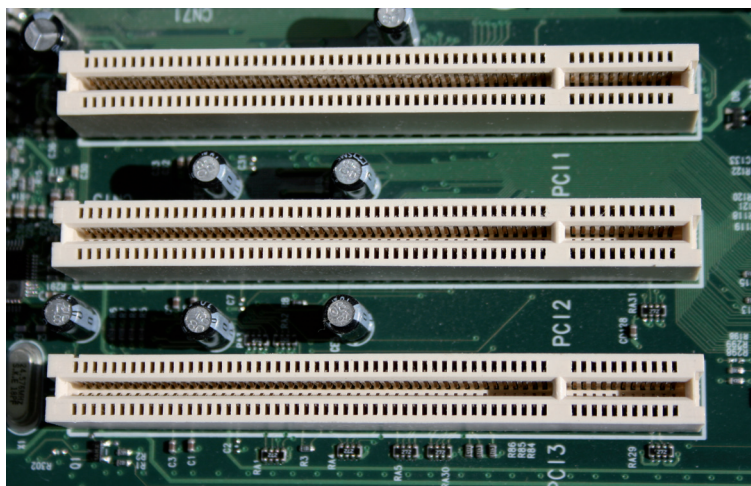


Figura 5.8: Slot de barramento PCI

Fonte: http://www.gdhpess.com.br-hmc-leia-cap4-6_html_m1d0275e6.jpg

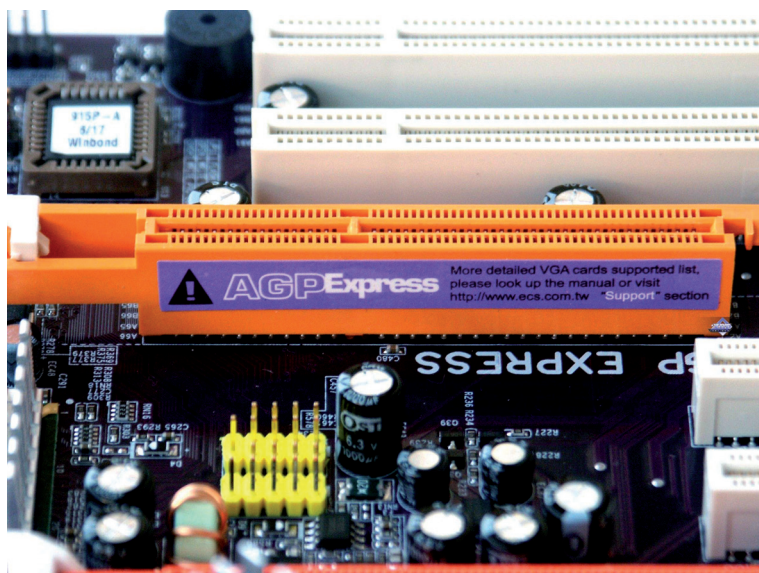


Figura 5.9: Slot AGP 8x

Fonte: <http://www.ixbt.com-video2-images-agp-express-ecs2.jpg>

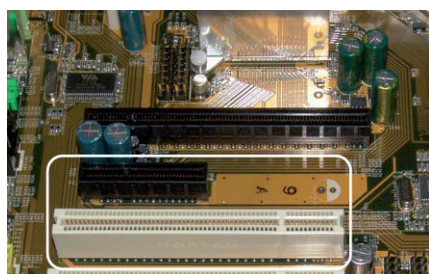


Figura 5.10: Slots PCI Express 16x (branco) e 4x (preto)

Fonte: http://www.hwupgrade.it/articoli/skmadri/1087/via-k8t890-preview-pci-express-per-athlon-64_4.html

Nesse caso, a principal diferença entre os diversos tipos de barramentos está na quantidade de bits que podem ser transmitidos por vez e na frequência de operação utilizada. Os barramentos AGP e PCI Express são considerados os mais rápidos, seguidos pelo barramento PCI original, sendo esses os barramentos mais utilizados em computadores atualmente.

5.4 Instruções de máquina

O termo instrução de máquina já foi mencionado em itens anteriores, mas sem uma explicação efetiva do que realmente significa. Foi demonstrado que a UCP é responsável pela execução de instruções e dados de programas, os quais que se encontram armazenados na memória. Além disso, foi colocado que uma instrução é uma ordem para que a UCP realize determinada operação (ex.: somar, subtrair, mover um dado de um local para outro, transferir um dado para um dispositivo de saída).



Realize uma pesquisa em sites da internet acerca de computadores e os respectivos barramentos disponíveis em sua arquitetura, destacando qual o modelo de computador, quais barramentos ele contém e quais as características desses barramentos (taxa de transmissão, periféricos que são acoplados a eles, entre outros). Essa pesquisa deverá ser postada como atividade no AVEA, no formato de um documento de texto.

Segundo Monteiro (2007), uma máquina pode executar tarefas complicadas e sucessivas se for “instruída” sobre o que fazer e em que sequência isso deve ser feito. Os seres humanos, ao receberem uma instrução (ex.: “trazer a pasta da funcionária Ana”), precisaram realizar uma série de ações intermediárias, até que a tarefa seja realizada por completo. Então, considerando esse exemplo, seria necessário: localizar o arquivo em que as pastas de todos os funcionários estão arquivadas; localizar a pasta da funcionária Ana, trazê-la a quem solicitou.

Da mesma forma, para a máquina (computador) é necessário que cada instrução seja detalhada em pequenas etapas. Isso ocorre porque os computadores são projetados para entender e executar pequenas operações, ou seja, as operações mais básicas (ex.: soma, subtração). Essas pequenas etapas de uma instrução dependem do conjunto de instruções do computador.

Assim, uma instrução de máquina pode ser definida pela formalização de uma operação básica que o *hardware* é capaz de realizar diretamente (MONTEIRO, 2007). Em outras palavras, consiste em transformar instruções mais complexas em uma sequência de instruções básicas e compreensíveis pelo processador.

Para exemplificar, vamos considerar um processador com uma ULA capaz de executar a soma ou a multiplicação de dois números (operações básicas), mas não as duas coisas ao mesmo tempo. Agora imaginem que esse processador precisa executar: $X = A + B * C$ de uma só vez. Isso irá gerar a necessidade de transformar essa instrução, considerada complexa para esse processador, em uma sequência de instruções mais básicas, da seguinte forma:

- Executar primeiro: $T = B * C$ (sendo que T é um registrador ou memória temporária)
- Em seguida realizar a operação: $X = A + T$

O projeto de um processador é centrado no conjunto de instruções de máquina que se deseja que ele execute, ou seja, do conjunto de operações primitivas que ele poderá executar. Quanto menor e mais simples for o conjunto de instruções, mais rápido é o ciclo de tempo do processador (MONTEIRO, 2007).

Um processador precisa dispor de instruções para: movimentação de dados; aritméticas; lógicas; edição; deslocamento; manipulação de registros de índice; desvio; modificação de memória; formais de ligação à sub-rotina; manipulação de pilha; entrada e saída e de controle.

Atualmente, há duas tecnologias de projeto de processadores empregadas pelos fabricantes de computadores (MONTEIRO, 2007):

- Sistemas com conjunto de instruções complexo (*Complex Instruction Set Computers* – CISC), e
- Sistemas com conjunto de instruções reduzido (*Reduced Instruction Set Computers* – RISC).

Ambas as tecnologias serão abordadas em seções subsequentes.

5.4.1 Formato das instruções

De acordo com Monteiro (2007), uma instrução é formada basicamente por dois campos:

- a) Código de operação (*Opcode*): um subgrupo de *bits* que identifica a operação a ser realizada pelo processador. É o campo da instrução cujo valor binário identifica a operação a ser realizada, conforme exemplo da Figura 5.11. Esse valor é a entrada no Decodificador de Instruções na Unidade de Controle. Cada instrução deverá ter um código único que a identifique.

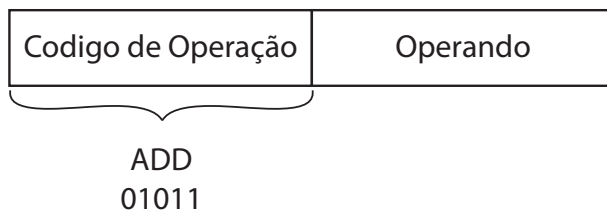


Figura 5.11: Código de operação de uma instrução

Fonte: Adaptado de Monteiro (2007)

- b) Operando: um subgrupo de *bits* que identifica o endereço de memória onde está contido o dado que será manipulado, ou pode conter o endereço onde o resultado da operação será armazenado (Figura 5.12).

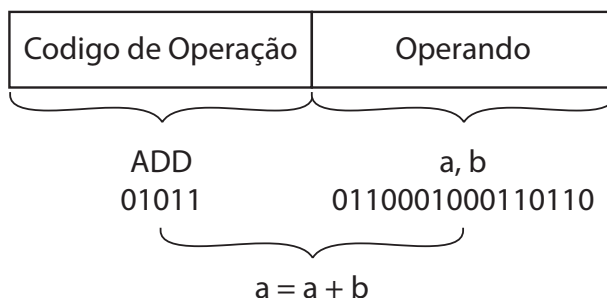


Figura 5.12: Operando de uma instrução

Fonte: Adaptado de Monteiro (2007)

6.4.2 Ciclo de instrução

A partir da proposta da arquitetura de Von Neumann, da qual os conceitos básicos ainda são válidos, propunha-se que as instruções fossem executadas sequencialmente (a não ser pela ocorrência de um desvio), uma a uma. O contador de instruções indica a sequência de execução, isto é, o CI controla o fluxo de execução das instruções. A seguir (Figura 5.13) é ilustrado o ciclo de execução de uma instrução. Em seguida, é descrita em mais detalhes cada uma das fases.

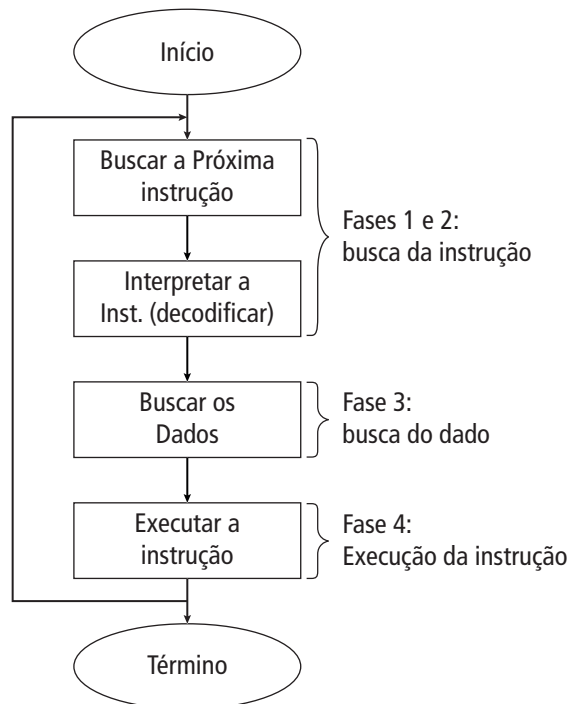


Figura 5.13: Ciclo de instrução

Fonte: <http://www.users.rdc.puc-rio.br/rmano/comp8ucp.html>

6.4.2.1 Fase 1

A UCP busca o código de operação (*Opcode*) na memória principal, o qual está localizado no endereço contido no CI (endereço da próxima instrução a ser executada) e armazena-o no Registrador de Instrução (RI): **RI ← (CI)**

Micro-operações da fase 1:

- a) a UC lê o conteúdo do CI (endereço da próxima instrução) e coloca o endereço no REM;
- b) a UC envia um sinal – via barramento de controle – à controladora da memória principal para que realize uma operação de leitura;

- c) a memória principal lê o endereço que está no REM – via barramento de endereço – e busca o conteúdo da célula referenciada;
- d) a memória principal coloca no RDM – via barramento de dados – o conteúdo da célula;
- e) a controladora da memória principal envia à UC – via barramento de controle – o sinal de leitura concluída;
- f) a UC transfere o código de operação (conteúdo que está no RDM) ao RI.

6.4.2.2 Fase 2

O Decodificador de Instrução decodifica (interpreta) o Código de Operação (*Opcode*) contido no RI.

Micro-operações da fase 2:

- a) o RI envia para o decodificador de instrução os *bits* correspondentes ao *Opcode*;
- b) o Decodificador de Instruções determina quantas células a instrução ocupa e identifica a operação a ser realizada;
- c) a UC envia um sinal de controle à ULA informando a operação a ser realizada e incrementa o CI para apontar para a próxima instrução:
 $CI \rightarrow (CI+N)$, onde, $N = n^{\circ}$ de células que a próxima instrução ocupa.

6.4.2.3 Fase 3

A UC busca (se houver) o(s) dado(s) (Operandos): $RI \leftarrow (Op)$

Micro-operações da fase 3:

- a) a UC envia um sinal – via barramento de controle – à controladora da memória principal para que realize uma operação de leitura;
- b) a memória principal lê o endereço que está no REM – via barramento de endereços – e busca o conteúdo da célula referenciada;
- c) a memória coloca no RDM – via barramento de dados – o conteúdo da célula lida;

- d) a memória principal envia à UC – via barramento de controle – um sinal de leitura concluída;
- e) a UC transfere o operando (conteúdo do RDM) ao RI (se for um código de operação) ou a um dos registradores internos da UCP (se for um dado).

Obs.: Esta fase se repete até que sejam trazidos para dentro da UCP todos os operandos necessários à execução da instrução.

6.4.2.4 Fase 4

A UC comanda a execução da instrução (a operação é executada sobre o(s) dado(s));

Micro-operações da fase 4:

- a) a ULA executa a instrução sobre os dados disponíveis nos registradores;
- b) ao concluir a operação, a ULA envia um sinal para a UC informando que a execução terminou;
- c) a UC identifica o endereço de memória para onde deve ser enviado o resultado da operação e o armazena no REM;
- d) a UC autoriza o envio do resultado da operação para o RDM;
- e) a UC autoriza a controladora de memória a realizar uma operação de leitura no REM para obter o endereço de memória onde deverá ser escrito o resultado e uma leitura no RDM para obter o resultado a ser escrito na memória.



Esquema Geral de Funcionamento do Processador.
Disponível em: <http://sca.unioeste-foz.br/~grupob2/00/esq-geral.htm>



Processadores RISC x Processadores CISC.
Disponível em: <http://www.guiadohardware.net/artigos/risc-cisc>. Acesso em 19 jun. 2010.

6.4.2.5 Fase 5

Se o programa tiver terminado, pára; senão, volta à Fase 1.

5.5 Arquiteturas RISC e CISC

Até o momento foram apresentadas algumas características das instruções de máquina, destacando que arquitetura de um processador está fortemente relacionada com o seu conjunto de instruções e que um processador sempre estará preparado para processar instruções básicas (ex.: soma, subtração, comparação).

A presença de outros tipos de instruções depende da orientação da arquitetura.

tura (*Complex Instruction Set Computer* – CISC ou *Reduced Instruction Set Computer* – RISC) e se está voltada para aplicações gerais ou específicas.

Resumo

Nesta aula discutimos a organização e arquitetura do processador (UCP), com destaque para o modelo de Von Neumann, o qual previa que o computador deveria: codificar instruções que pudessem ser armazenadas na memória, e que utilizassem cadeias de uns e zeros (binário) para codificá-las, armazenar na memória as instruções e todas as informações que fossem necessárias para a execução da tarefa desejada e, ao processarem o programa, as instruções fossem buscadas diretamente na memória.

Vimos também que a Unidade Central de Processamento (UCP) é composta por várias partes distintas, entre elas: registradores, Unidade de Controle (UC) e Unidade Lógica Aritmética (ULA). A UCP pode ser dividida em duas categorias funcionais, as quais podem ser chamadas de unidade, conforme segue: **Unidade Funcional de Controle** e a **Unidade Funcional de Processamento**.

A Unidade Funcional de Processamento é composta pelos seguintes elementos: Registradores, ACC, ULA. A Unidade Funcional de Controle é composta pelos seguintes elementos: RDM, REM, CI, RI, Decodificador de Instruções, UC, *Clock* (relógio).

Foi apresentado um tópico específico acerca de barramentos, em que foi mostrado o conceito de Monteiro (2007), que entende ser um conjunto de condutores elétricos que interligam os diversos componentes do computador e de circuitos eletrônicos que controlam o fluxo dos *bits*.

O barramento de um sistema computacional, denominado barramento do sistema, é o caminho por onde trafegam todas as informações dentro do computador. Esse barramento é formado basicamente por três vias específicas/ funções básicas: barramento de dados – interliga o RDM (localizado na UCP) à memória principal, para transferência de instruções ou dados a serem executados; barramento de endereços – interliga o REM (localizado na UCP) à memória principal, para transferência dos *bits* que representam um determinado endereço de memória onde se localiza uma instrução ou dado a ser executado e; barramento de controle – interliga a UCP, mais especificamente a Unidade de Controle (UC) aos demais componentes do sistema computacional (memória principal, componentes de entrada e de saída) para passagem de sinais de controle gerados pelo sistema.

De acordo com Monteiro (2007), existe uma grande diferença de características dos diversos componentes existentes em um computador, principalmente periféricos (ex.: a velocidade de uma transferência de dados de um teclado é muitas vezes menor que a velocidade de transferência de dados de um disco magnético). Considerando esse fato, os projetistas de sistemas de computação criaram diversos tipos de barramento, apresentando taxas de transferência de *bits* diferentes e apropriadas às velocidades dos componentes interconectados (ex.: UCP, memória, disco rígido, teclado). Assim, atualmente os modelos de organização de sistemas de computação adotados pelos fabricantes possuem diferentes tipos de barramento: barramento local – barramento de maior velocidade de transferência de dados, interligando basicamente a UCP à memória principal; barramento de sistema – trata-se de um barramento opcional, adotado por alguns fabricantes, fazendo com que o barramento local faça a ligação entre o processador e a memória cache e esta se interligue com os módulos de memória principal (RAM) através dele; e barramento de expansão (ou de entrada e saída) – é responsável por interligar os diversos dispositivos de entrada e saída aos demais componentes do computador.

Os barramentos permitem o seu compartilhamento com os demais componentes do sistema, especialmente o barramento de expansão, que é compartilhado com diversos dispositivos de entrada e saída. Para que isso ocorra, é indispensável um mecanismo de controle de acesso baseado em regras, que garanta que quando um dos dispositivos estiver utilizando o barramento, os demais componentes deverão aguardar a sua liberação. Esse mecanismo de controle de acesso é denominado protocolo, fazendo com que um barramento não seja composto somente de fios condutores, mas também de um **protocolo**. Sendo assim, os fabricantes de computadores têm procurado criar uma padronização na definição de protocolos, de forma a evitar que cada um crie o seu próprio, com características diferentes dos demais, o que tornaria muito inflexível o uso de certos componentes (ex.: vídeo, impressoras, discos rígidos) disponibilizados no mercado. Dessa forma, muitos padrões de barramento de expansão foram desenvolvidos ao longo do tempo, alguns deles já não mais utilizados. Os mais populares são apresentados a seguir, conforme Monteiro (2007): PCI, USB, AGP, PCI Express. Cada um desses barramentos apresenta um slot ou encaixe específico na placa-mãe do computador, permitindo maior flexibilidade no uso de periféricos.

Foi mostrado que a UCP é responsável pela execução de instruções e dados de programas, os quais que se encontram armazenados na memória e que

uma instrução é uma ordem para que a UCP realize determinada operação (ex.: somar, subtrair, mover um dado de um local para outro, transferir um dado para um dispositivo de saída).

Monteiro (2007) afirma ainda que o projeto de um processador é centrado no conjunto de instruções de máquina que se deseja que ele execute, ou seja, do conjunto de operações primitivas que ele poderá executar. Quanto menor e mais simples for o conjunto de instruções, mais rápido é o ciclo de tempo do processador.

Basicamente podemos concluir que um processador precisa dispor de instruções para: movimentação de dados; aritméticas; lógicas; edição; deslocamento; manipulação de registros de índice; desvio; modificação de memória; formais de ligação à sub-rotina; manipulação de pilha; entrada e saída e de controle.

Atualmente, há duas tecnologias de projeto de processadores empregadas pelos fabricantes de computadores (MONTEIRO, 2007): processadores com um conjunto de instruções complexo (*Complex Instruction Set Computers* – CISC) e processadores com um conjunto de instruções reduzido (*Reduced Instruction Set Computers* – RISC).

Ao considerar o formato de uma instrução, pode-se dizer que ela é composta por: um código de operação (é o campo da instrução cujo valor binário identifica a operação a ser realizada) e um operando (identifica o endereço de memória onde está contido o dado que será manipulado, ou pode conter o endereço onde o resultado da operação será armazenado).

O ciclo de uma instrução no processador é composto basicamente das seguintes etapas: busca da próxima instrução, interpretação (decodificação da instrução), busca de dados e execução da instrução.

Atividades de aprendizagem

Os itens “b” e “e” da Questão 1 devem ser respondidos e comentados no fórum “Arquitetura da UCP”, disponível no AVEA. Os demais itens da Questão 1 e as demais questões devem ser respondidos em um arquivo texto e postados como atividade no AVEA.

1. Responda os questionamentos abaixo em relação à arquitetura dos processadores (UCP):

- a) Qual era a proposta de Von Neumann para construção de computadores? Qual a influência do seu modelo na arquitetura dos computadores atuais?
 - b) Qual seria o gargalo (limitação) do modelo de Von Neumann em relação ao desempenho do computador? De que forma essa limitação poderia ser resolvida ou pelo menos minimizada?
 - c) A UCP é dividida em duas unidades funcionais principais. Como são chamadas tais unidades? Comente-as.
 - d) Qual é e em qual unidade funcional se localiza o registrador cujo conteúdo controla a sequência de processamento de instruções de um programa?
 - e) O que você entende por dados e instruções utilizados pela UCP para realizar suas operações?
 - f) Desenhe o fluxograma do ciclo básico de instruções e explique-o.
2. Em relação à arquitetura dos processadores (UCP), assinale as opções abaixo com V (verdadeiro) ou F (falso):
- () O componente responsável por coordenar e gerenciar toda a atividade de um processador é a Unidade Lógica e Aritmética (ULA).
 - () O registrador cujo conteúdo controla a sequência de processamento de instruções de um programa é denominado Registrador de Instruções (RI).
 - () Uma instrução representa é uma ordem para que o processador realize uma determinada operação.
 - () As fases que compõem o ciclo básico de uma instrução são: buscar a próxima instrução, interpretar a instrução (decodificar), buscar os dados, executar a instrução.
 - () É função da Unidade de Controle (UC) identificar que operação será realizada, correlacionada à instrução cujo código de operação foi interpretado.
 - () *Clock* ou relógio é um gerador de pulsos cuja duração é chamada de frequência e a quantidade de vezes que esse pulso básico se repete em um segundo é denominada velocidade.

() A Unidade Lógica e Aritmética (ULA) é um aglomerado de circuitos lógicos e componentes eletrônicos simples que, integrados, realizam as operações aritméticas e lógicas.

3. Em relação aos barramentos do sistema computacional, analise as opções abaixo:

I. O barramento consiste em um conjunto de condutores elétricos em um computador que permite a comunicação entre vários componentes do computador, tais como: CPU, memória, dispositivos de I/O.

II. O barramento local conecta o processador à memória RAM. Esse barramento é constituído de barramento de dados e de endereços.

III. O barramento de Entrada e Saída conecta periféricos (ex.: vídeo, impressoras, som).

IV. Entre as limitações para a velocidade do barramento, está a necessidade de suportar uma faixa de dispositivos com velocidades muito semelhantes e taxas de transferência de dados muito diferentes.

Em relação às opções listadas acima, assinale a alternativa que corresponde a(s) opção(ões) correta(s):

() Somente a II está correta.

() I e II estão corretas.

() I e III estão corretas.

() II e IV estão corretas.

Aula 6 – Representação de dados

Objetivos

Conhecer a representação de dados no formato interno dos sistemas computacionais.

6.1 Introdução

Um computador funciona por meio da execução sistemática de instruções que o orientam a realizar algum tipo de operação sobre valores (numéricos, alfanuméricos ou lógicos). Esses valores são genericamente conhecidos como **dados** (MONTEIRO, 2007).

Os dados são convertidos internamente em um código de armazenamento no formato binário. Para compreender melhor, pode-se considerar o exemplo a seguir:

Para o valor decimal **143**

00110001 (algarismo 1)

00110111 (algarismo 4)

00110011 (algarismo 3)

Qualquer que tenha sido a linguagem de programação utilizada para escrever o programa, ela deverá ser convertida para código-objeto (código binário) e, em seguida, para o código executável (conjunto de **códigos de máquina**), o qual é gerado pelo **compilador** da linguagem), conforme a Figura 6.1. Essa conversão também inclui dados, que deverão ser alterados de modo a estarem em uma forma apropriada para utilização pela ULA (ex.: números inteiros ou fracionários). Por exemplo, para efetivar uma soma, a ULA executa, passo a passo, uma série de micro-operações (um algoritmo): verificar o sinal dos números, verificar o tipo do número, etc.

A-Z

Códigos de máquina

conjunto de instruções que seu processador é capaz de executar (conforme descrito na seção 5.4)

Compilador

é um programa de sistema que traduz um programa descrito em uma linguagem de alto nível para um programa equivalente em código de máquina para um processador.



Pesquise sobre os compiladores existentes para as mais diversas linguagens de programação. Faça uma listagem desses compiladores acompanhados de suas características, coloque o conteúdo pesquisado em um arquivo texto e faça a postagem como atividade no AVEA.

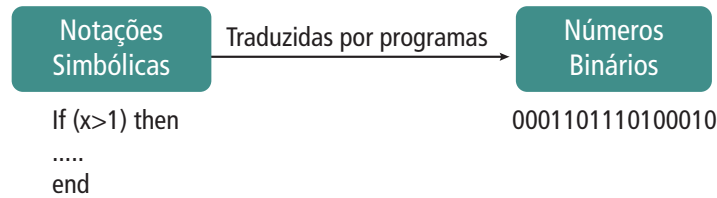


Figura 6.1: Processo de conversão
Fonte: Elaborada pela autora

6.2 Formas de representação

As diferentes formas de representação e respectivos algoritmos de realização das operações matemáticas são muito úteis, pois cada uma tem uma aplicação mais vantajosa que a outra. Cabe ao programador a escolha da forma a ser utilizada pelo sistema, podendo ser explícita ou implícita. Explícita, quando o programador define as variáveis e constantes em seu programa. Implícita, quando é deixado para que o compilador faça a escolha.

6.3 Tipos de dados

Definem para o sistema como cada dado deverá ser manipulado, pois conforme citado anteriormente, cada tipo de dado recebe um tratamento diferenciado pelo processador.

Exemplo:

```
VAR X:=INTEGER;  
VAR X:=REAL;
```

Os termos INTEGER e REAL são interpretados de modo diferente, acarretando alterações significativas tanto no modo de organizar os *bits* que representam um número quanto na sequência de etapas do algoritmo de execução de uma operação aritmética com o número (MONTEIRO, 2007).

De modo geral, as seguintes formas de dados são mais utilizadas nos programas atuais de computadores (formas primitivas, entendidas pelo *hardware*) (MONTEIRO, 2007):

- Tipo Caractere: dados sob forma de caractere;
- Tipo Lógico: dados sob forma lógica;
- Tipo Numérico: dados sob forma numérica.

Outras formas mais complexas são permitidas em certas linguagens modernas (como tipo REGISTRO, tipo ARRAY, tipo INDEX, tipo POINTER etc.). No entanto, durante o processo de compilação, os dados acabam sendo convertidos finalmente nas formas primitivas já mencionadas, para que o *hardware* possa executá-las.

6.3.1 Tipo caractere

A representação interna de informações em um computador é realizada através de uma correspondência entre o símbolo da informação e o grupo de algarismos binários (*bits*). Cada símbolo (caractere, número ou símbolo) possui uma identificação específica.

Exemplo: Símbolo “A” $\Rightarrow$ Algarismos binários “10101101”

Você pode se perguntar como é possível **representar, com apenas dois símbolos** (0 e 1), todos os caracteres alfabéticos, algarismos decimais, sinais de pontuação, de operações matemáticas, entre outros, necessários à elaboração de um programa de computador. Monteiro (2007) afirma que a resposta para essa pergunta seria: pela utilização do método chamado de codificação, pelo qual cada símbolo da nossa linguagem tem um correspondente grupo de *bits* que identifica univocamente o referido símbolo (caractere).

Existem alguns padrões de codificação previamente definidos, conforme apresentados no Quadro 6.1.

Quadro 6.1: Padrões de codificação de caracteres	
Códigos Caracteres	
BCD – <i>Binary Code Decimal</i>	Utiliza 6 <i>bits</i> /caracteres, codificando 64 caracteres.
EBCDIC – <i>Extended Binary Coded Decimal Interchange Code</i>	Exclusivo da IBM, utilizando 8 <i>bits</i> para codificar 256 caracteres.
ASCII – <i>American Standart Code for Information Interchange</i>	ASCII – usado pelos demais fabricantes. Utiliza oito <i>bits</i> /caractere em sua versão estendida, codificando 256 caracteres.
UNICODE	Código que utiliza 16 <i>bits</i> /símbolo, podendo representar 65.536 símbolos diferentes. Pretende codificar em um único código os símbolos de todas as linguagens conhecidas no mundo. Está sendo desenvolvido por um consórcio desde 1991 (www.unicode.org).

Fonte: Adaptada de Monteiro (2007)

A utilização de padrões de codificação (ex.: ASCII, Unicode) é o método primário de introdução de informações no computador. As demais formas de representação de informação (tipos de dados) surgem no decorrer do processo de compilação ou interpretação do programa. O padrão de codificação mais utilizado pela indústria de computadores é o ASCII. A codificação correspondente a esse padrão já é parte do *hardware* (armazenado em uma memória do tipo ROM) e é definida pelo próprio fabricante. A Figura 6.2 apresenta parte da tabela ASCII como exemplo.

Binário	Decimal	Hexa	Glifo	Binário	Decimal	Hexa	Glifo	Binário	Decimal	Hexa	Glifo
0010 0000	32	20		0010 0000	64	40	@	0010 0000	96	60	`
0010 0001	33	21	!	0010 0001	65	41	A	0010 0001	97	61	a
0010 0010	34	22	"	0010 0010	66	42	B	0010 0010	98	62	b
0010 0011	35	23	#	0010 0011	67	43	C	0010 0011	99	63	c
0010 0100	36	24	\$	0010 0100	68	44	D	0010 0100	100	64	d
0010 0101	37	25	%	0010 0101	69	45	E	0010 0101	101	65	e
0010 0110	38	26	&	0010 0110	70	46	F	0010 0110	102	66	f
0010 0111	39	27	'	0010 0111	71	47	G	0010 0111	103	67	g
0010 1000	40	28	(	0010 1000	72	48	H	0010 1000	104	68	h
0010 1001	41	29	)	0010 1001	73	49	I	0010 1001	105	69	i
0010 1010	42	2A	•	0010 1010	74	4A	J	0010 1010	106	6A	j
0010 1011	43	2B	+	0010 1011	75	4B	K	0010 1011	107	6B	k
0010 1100	44	2C	•	0010 1100	76	4C	L	0010 1100	108	6C	l
0010 1101	45	2D	-	0010 1101	77	4D	M	0010 1101	109	6D	m
0010 1110	46	2E	•	0010 1110	78	4E	N	0010 1110	110	6E	n
0011 1111	47	2F	/	0011 1111	79	4F	O	0011 1111	111	6F	o
0011 0000	48	30	0	0011 0000	80	50	P	0011 0000	112	70	p

Figura 6.2: Exemplo parcial da Tabela ASCII

Fonte: http://weblogbrasil.files.wordpress.com/2008/09/tabela_impri1.png

6.3.2 Tipo lógico

Permite a utilização de variáveis que possuem apenas dois valores para representação: Falso (*bit* 0) e Verdadeiro (*bit* 1). As funções de cada operador lógico estão apresentadas no Quadro 6.2.

Quadro 6.2: Operadores lógicos e suas funções

Porta Lógica	Definição
AND	O operador lógico AND é definido de modo que o resultado da operação com ele será VERDADE se e somente se todas as variáveis de entrada forem VERDADE (=1). Caso contrário, o resultado será FALSO (=0).

OR	O resultado da operação será VERDADE (=1) se um operando (ou variável lógica) ou o outro for verdadeiro. Basta que apenas um dos operandos seja verdadeiro. Caso contrário, o resultado será FALSO (=0). Operadores lógicos OR também são largamente utilizados em lógica digital ou na definição de condições em comandos de decisão de certas linguagens de programação.
	Continua
NOT	É definido de modo a produzir na saída um resultado de valor oposto (ou inverso) ao da variável de entrada. É usado apenas com uma única variável. Desse modo, se a variável tem o valor 0 (FALSO), o resultado da operação NOT sobre essa variável será 1 (VERDADE), e se a variável for igual a 1 (VERDADE), então o resultado do NOT será 0 (FALSO).
XOR	O operador lógico XOR (EXCLUSIVE-OR) ou OU EXCLUSIVO é definido de modo a prover um resultado VERDADEIRO se apenas uma das variáveis ou operadores for VERDADEIRA. Sendo $X=A \text{ XOR } B$, o resultado X será VERDADE se exclusivamente (daí o nome OU EXCLUSIVO) A OU B for VERDADE. Caso ambos sejam "VERDADE" ou ambos "FALSO", então o resultado será FALSO.
NAND	Negação do operador lógico AND.
NOR	Negação do operador lógico OR.

Fonte: Adaptada de Monteiro (2007)

6.3.3 Tipo numérico

Como os computadores são elementos binários, a forma mais eficiente de representar números deve ser "binária", isto é, converter o número diretamente de decimal para seu correspondente valor binário. Deste modo a ULA poderá executar as operações mais rapidamente.

Conforme Monteiro (2007), existem três fatores que devem ser considerados, pois podem acarretar inconvenientes no projeto e na utilização da máquina:

- a) a representação do sinal do número;
- b) a representação da vírgula (ou ponto) que separa a parte inteira da fracionária de um número não inteiro;
- c) a quantidade limite de algarismos possível de ser processada pela ULA.

O problema que consiste do sinal do número pode ser resolvido com o acréscimo de mais um *bit* na representação do número (adicionado à esquerda), como *bit* mais significativo. Esse *bit* adicional indica o sinal do número. A conversão adotada (conforme o Quadro 6.3) é:

- Valor positivo: *bit* de sinal igual a 0;
- Valor negativo: *bit* de sinal igual a 1.

Quadro 6.3: Exemplo

Valor Decimal	Valor Binário
- 47	100101111 (com 9 <i>bits</i>)
+ 47	000101111 (com 9 <i>bits</i>)

Fonte: Adaptado de Monteiro (2007)

Outro problema reside na forma de representação de números fracionários, por causa da dificuldade de representar a vírgula/ponto internamente, entre a posição de dois *bits*. O que ocorre é que a vírgula não é efetivamente representada, mais sim assumida sua posição no número e este sendo representado apenas pelos seus algarismos significativos como se fosse inteiro.

Exemplo: 110111,110 $\Rightarrow$ 110111110

O sistema identifica que quantidade de algarismos é inteira e que a quantidade é fracionária através da escolha entre dois modos de representação e de realização de operações aritméticas (MONTEIRO, 2007):

- a) representação em ponto fixo (vírgula fixa);
- b) representação em ponto flutuante (vírgula flutuante).

A quantidade de dígitos disponíveis no sistema de computação (processador) para representar números é um problema relevante. Na matemática os números reais existentes são infinitos; no entanto, computadores são máquinas onde as células e registradores possuem tamanho finito mas que têm capacidade de representar uma quantidade finita de números. Desse modo surge o aspecto denominado *overflow* ou estouro da capacidade de representar números. Há também o *underflow*, que se caracteriza por ocorrer um resultado cujo valor é menor que o menor valor representável.



Representação de Dados.
Disponível em: http://www.cristiancechinel.pro.br/my_files/algorithms/bookhtml/node23.html. Acesso em: 19 jul. 2010.

Resumo

A representação de dados esclarece o formato de representação dos diversos tipos de dados que o computador pode receber como entrada - seja a partir da execução de programas ou a partir de dispositivos de entrada e saída.

Os principais tipos de dados compreendidos pelo computador, também chamados de tipos primitivos de dados, são: caráter, lógico e numérico. A partir desses tipos básicos foram desenvolvidos alguns tipos considerados mais complexos e que não são compreendidos diretamente pelo *hardware* (ex.: vetor, índice, registro, ponteiros), sendo necessário para isso que tais tipos sejam convertidos para formatos primitivos equivalentes.

A tabela ASCII é o padrão de codificação utilizado para a conversão de caracteres, números e símbolos inseridos no computador via dispositivos de entrada e saída (ex.: teclado). Assim, cada caractere possui um código binário único que o representa.

Alguns problemas devem ser considerados em relação à representação do tipo numérico, como: a representação do sinal do número, a representação do ponto ou vírgula do número e o tamanho dos registradores da UCP, os quais limitam o tamanho dos números que poderão ser representados pelo *hardware*.

Atividades de aprendizagem

1. Em relação à representação de dados no computador, responda às seguintes questões em um arquivo texto e poste-as como atividade no AVEA:
 - a) Como são representados os dados e instruções informados pelo usuário para o computador?
 - b) O que você entende por formas primitivas de dados?
 - c) Como os comandos de um programa em linguagem de alto nível (ex.: Pascal, C) são compreendidos pelo *hardware*?
 - d) Como os dados do tipo vetor, registro e apontador são compreendidos pelo *hardware*?
 - e) O que é a tabela ASCII e quando ela é utilizada?

Aula 7 - Dispositivos de entrada e saída

Objetivos

Identificar as características e compreender o funcionamento básico dos dispositivos de entrada e saída, acoplados ao sistema computacional.

7.1 Introdução a dispositivos de entrada e saída

De acordo com o modelo previsto por Von Neumann e apresentado ao longo das seções anteriores, um computador deve ser capaz de armazenar dados e instruções necessários para a execução de uma tarefa, na memória e no formato binário. Para que possam ser executados, esses dados e instruções são buscados pela UCP diretamente na memória. É na memória também que o resultado desse processamento será disponibilizado. Como esses dados e instruções chegaram à memória? Como o resultado do processamento desse conteúdo mantido na memória é retornado ao usuário?

A resposta para essas perguntas é que são necessários elementos que permitam a **interface** do usuário com o computador, tanto para dar a entrada de dados e instruções quanto para proporcionar a saída de resultados ao usuário, no formato adequado, tal qual foi solicitado.

Esses elementos de interface podem ser chamados de dispositivos (ou periféricos) de entrada e saída (E/S). São considerados um subsistema de memória, pois fazem parte do sistema maior que é o sistema computacional, onde os dispositivos de E/S compõem o chamado subsistema de E/S, o qual, segundo Monteiro (2007), deve ser capaz de realizar duas funções:

- a) receber ou enviar informações ao meio exterior;
- b) converter as informações (de entrada e saída) em uma forma inteligível para a máquina (se estiver recebendo) ou para o programador ou usuário (se estiver enviando).

A-Z

Interface

Interface: é a fronteira ou canal de comunicação que define a forma de comunicação entre duas entidades (ex.: usuário, computador).

Dentre os diversos dispositivos de E/S podemos citar: teclado, mouse, monitor de vídeo, impressora, *webcam*, modem, dispositivos de armazenamento (ex.: disco rígido, CD/DVD – ROM, *pen-drive*). Esses dispositivos se interligam à UCP e memória principal através do barramento de expansão (apresentado na Aula 5) e podem ser classificados em duas categorias: entrada (teclado, mouse, *webcam*, modem, disco rígido) e saída (impressoras, disco rígido, monitor de vídeo). Sublinhamos que o disco rígido, assim como outros meios de armazenamento, são dispositivos tanto de entrada quanto de saída.

Lembramos que todos os estímulos gerados pelos dispositivos de E/S, por exemplo - a pressão de uma tecla, são convertidos em vários sinais elétricos, com diferentes intensidades, podendo representar os valores 0 ou 1.

Monteiro (2007) destaca algumas observações relevantes que influenciam na comunicação dos dispositivos com a UCP e memória principal. São elas:

- Os dispositivos de E/S apresentam diferentes características, o que tornaria a comunicação entre UCP e periférico extremamente complicada, caso esta fosse realizada direta e individualmente (ex.: comunicação direta entre UCP e teclado, entre UCP e vídeo). Isso ocorre em função da grande diferença de velocidade entre UCP e os dispositivos de E/S, além de haver grandes diferenças de velocidade entre os próprios dispositivos, como por exemplo o disco rígido, que é mais rápido que o teclado. A Tabela 7.1 apresenta alguns exemplos de dispositivos de E/S e sua velocidade de transmissão de dados aproximada.

Tabela 7.1: Dispositivos de E/S e sua velocidade de transmissão de dados

Dispositivo	Taxa de transmissão (KB/s)
Teclado	0,01
Mouse	0,02
Impressora matricial	1
Modem	2 a 8
Disquete	100
Impressora laser	200
Scanner	400
CD-ROM	1000
Rede local	500 a 6000
Vídeo gráfico	60.000
Disco rígido (HD)	2000 a 10.000

Fonte: Adaptada de Monteiro (2007)

Além da velocidade, outro aspecto que diferencia os dispositivos de E/S é a sua forma de comunicação. A comunicação entre o núcleo do computador e os dispositivos de E/S poderia ser classificada em (MONTEIRO, 2007):

- a) Comunicação serial: a informação pode ser transmitida/recebida, *bit* a *bit*, um em seguida do outro, conforme a Figura 7.1.

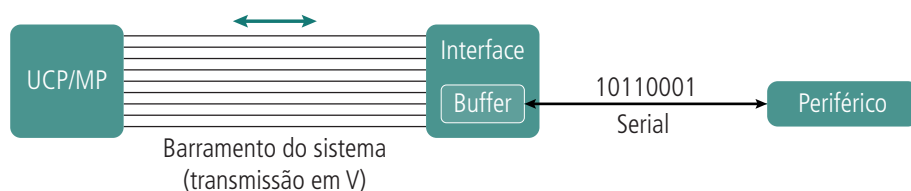


Figura 7.1: Transmissão serial entre interface e periférico

Fonte: Adaptada de Monteiro (2007)

- b) Comunicação paralela: a informação pode ser transmitida/recebida em grupos de *bits* de cada vez, isto é, um grupo de bits é transmitido simultaneamente de cada vez, conforme apresenta a Figura 7.2.

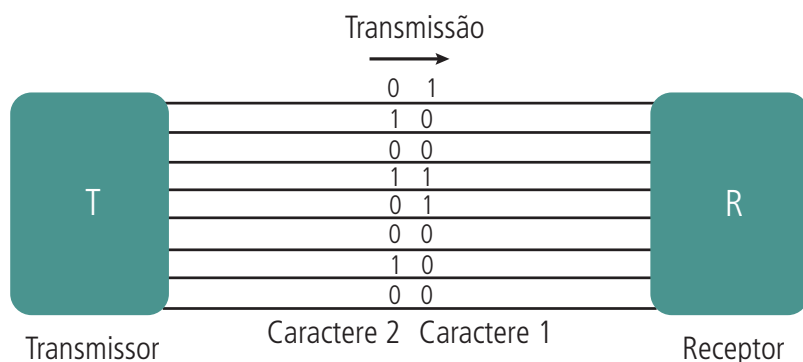


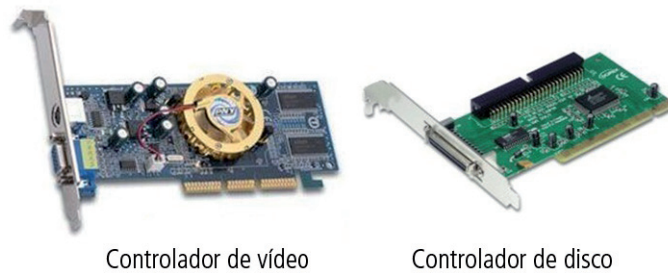
Figura 7.2: Transmissão paralela entre transmissor (interface) e receptor (periférico)

Fonte: Adaptada de Monteiro (2007)

Existem diferenças relativas à parte elétrica de geração e interpretação dos sinais de transmissão. O autor afirma que, devido a essas diferenças, na prática a UCP não se conecta diretamente com cada periférico, mas sim com dispositivos que realizam a “tradução” e a compatibilização das características de um (dispositivo de E/S) para o outro (UCP/MP), além de realizar outras tarefas de controle (MONTEIRO, 2007).

Os dispositivos possuem algumas denominações. A mais comum seria interface de E/S, mas comercialmente podem ser encontrados como controlador ou adaptador, adicionando-se o nome do dispositivo (ex.: controlador de vídeo, controlador de disco). A função de todos é sempre a mesma: compatibilizar as diferentes características de um periférico e da UCP/memória

principal, permitindo um fluxo correto de dados em uma velocidade adequada a ambos os elementos que estão sendo interconectados. A Figura 7.3 apresenta exemplos de interfaces.



Controlador de vídeo

Controlador de disco

Figura 7.3: Exemplos de controladores de E/S

Fonte: www.infowester.com.br

7.1.1 Metodologias de comunicação entre UCP e dispositivos de E/S

Em um sistema de computação há a necessidade de que a UCP se comunique com a memória principal (RAM) e com os dispositivos de E/S (ex.: teclado, *mouse*, monitor de vídeo, rede) para a transferência de dados. Semelhante ao que ocorre com a comunicação entre UCP e memória principal, na qual são definidos endereços para cada posição de memória, os quais são referenciados pela UCP, quando se trata de comunicação entre UCP e dispositivos, torna-se necessário que a UCP indique um endereço que corresponda ao periférico em questão.

Cada periférico acoplado ao sistema de computação possui um endereço, o qual é identificado como endereço de porta de E/S, ou seja, cada dispositivo de E/S possui um número de identificação única no sistema computacional onde se encontra. Dessa forma, se o endereço de porta ou endereço de E/S for um número de oito *bits*, significa que poderão ser conectados até 256 (2⁸) dispositivos ao sistema.

Diversas formas de comunicação entre UCP e memória principal foram propostas, as quais sofreram melhorias ao longo do tempo, buscando sempre alcançar uma melhor utilização da UCP e um melhor desempenho para o sistema como um todo. Murdocca (2000) destaca três métodos para gerenciar a entrada e saída:

a) Entrada e saída programada

Neste método, também chamado de *pooling*, a UCP precisa verificar conti-

nuamente se cada um dos dispositivos necessita de atendimento, ou seja, tudo depende da UCP. Por exemplo, se o disco quer transferir algum dado para a memória, a UCP deve ficar dedicada a esse processo de transferência até que esta seja concluída. A grande desvantagem é a subutilização da UCP, a qual, enquanto houver uma operação de transferência de dados, não realiza outras operações de processamento. Este método não é mais utilizado.

b) Entrada e saída controladas por interrupção

Este método possibilita que a UCP não fique presa em espera ocupada até que um dispositivo esteja pronto para realizar a transferência de dados propriamente dita. Assim, a UCP dá início à operação emitindo uma instrução de E/S para a interface ou controlador do dispositivo em questão e, quando o dispositivo estiver pronto para a operação de transferência, recebe uma **interrupção** avisando que ela poderá começar. Essa demora para iniciar a transferência, após as instruções da UCP, deve-se à lentidão dos dispositivos de E/S em relação à UCP, problema que é diminuído com o uso de interrupções. Este método sofreu melhorias e não é mais utilizado.

c) Acesso direto à memória (DMA)

A função do controlador (ou interface) é controlar seu dispositivo de E/S e manipular para ele o acesso ao barramento. Quando um programa quer dados do disco, por exemplo, ele envia um comando ao controlador de disco, que então emite comandos de busca e outras operações necessárias para que ocorra a transferência (TANENBAUM, 2007).

Para esse autor, quando o controlador lê ou escreve dados de/para a memória sem a intervenção da UCP, é dito que ele está executando um acesso direto à memória (*Direct Memory Access*), conhecido por DMA. Dessa forma, a UCP se limita a solicitar a transferência para um dispositivo denominado controlador de acesso direto à memória principal (*DMA Controller*), o qual se responsabiliza totalmente pela transferência. A UCP é avisada apenas no início e no final da operação de transferência entre dispositivo e memória principal. Este método é utilizado atualmente pelos computadores.

Destacamos que a UCP apresenta apenas uma linha para receber pedidos externos de interrupções de todos os dispositivos de E/S. Dessa forma, a fim de compartilhar essa linha com os diversos dispositivos existentes em um computador, foi desenvolvido o **controlador de interrupções**, o qual rece-

A-Z

Interrupção

Interrupção: pode ser definida como um evento ou um aviso à UCP de que algum dispositivo está solicitando a realização de uma operação (ex.: o recebimento de dados via rede, a pressão de uma tecla). Quando uma interrupção ocorre, a UCP deve suspender imediatamente a execução de seu programam corrente e começar a executar um procedimento de tratamento da interrupção.



Interrupção de Hardware.
Disponível em:

http://pt.wikipedia.org/wiki/Interrup%C3%A7%C3%A3o_de_hardware.

Acesso em: 19 jul. 2010



Arquitetura de PCs. Disponível em: <http://www.laercio.com.br/artigos/hardware/hard-010/hard-010.htm-010.htm>. Acesso em: 19 jul. 2010

be e encaminha, de acordo com a prioridade (definida para evitar conflitos de acesso à UCP), os pedidos de interrupção feitos pelos diferentes dispositivos de entrada/saída. Quando a UCP recebe o pedido de interrupção, verifica com o controlador de interrupções que dispositivo causou o pedido e então efetua o atendimento apropriado à interrupção.

A seguir, serão apresentadas as características de alguns dos principais dispositivos de E/S.

7.1.1.1 Teclado

O teclado é o dispositivo utilizado para dar a entrada a instruções vindas dos seres humanos ao computador. Nesse caso, é necessário que ele seja capaz de identificar a simbologia utilizada pelos seres humanos. Por isso, geralmente existe um teclado para cada língua (ex.: português, japonês) ou inclusive dentro do mesmo idioma podem existir adaptações de teclado para países diferentes. (ex.: para o Brasil (ABNT-2) e Portugal (Pt)).

Os teclados mais populares são os de uso geral, ou seja, aqueles compostos de todas as teclas alfabéticas, numéricas e caracteres especiais (sinais de pontuação e operações em geral), conforme mostra a Figura 7.4.



Figura 7.4: Exemplo de teclado de um PC

Fonte: <http://www.pontodamidia.com.br/loja/images/087.jpg>

A-Z

Capacitância

Capacitância (ou capacidade): é a grandeza elétrica de um capacitor, determinada pela quantidade de energia elétrica que pode ser armazenada em si por uma determinada tensão e pela quantidade de corrente alternada que o atravessa numa determinada frequência.

Existem três tecnologias de fabricação de teclas: teclas mecânicas (de contato direto), teclas capacitivas; e teclas de efeito Hall, sendo que a mais utilizada é a tecnologia capacitiva. Nesse último tipo, as teclas funcionam à base da variação de **capacitância** (uma propriedade elétrica) do acoplamento entre duas placas metálicas, variação essa que ocorre quando uma tecla é pressionada. Com isso, essas teclas apresentam baixo custo e menor tamanho, além de não apresentarem contatos mecânicos, que podem oxidar com o tempo (MONTEIRO, 2007).

O teclado é considerado um conjunto de chaves elétricas, sendo cada chave acionada por uma tecla. Cada uma das chaves corresponde a um código binário, o qual é enviado para a placa-mãe, via controlador de teclado, e é interpretado de acordo com o padrão de teclado que está sendo utilizado.

7.1.1.2 Monitor de vídeo

O monitor de vídeo é um dispositivo de saída indispensável ao uso dos computadores pessoais (PC). Por meio dele é possível obter – a partir da visão – a maior parte das saídas geradas pela UCP, ou seja, a resposta às solicitações que o usuário encaminha para dentro do computador (ex.: formatar um texto, realizar um cálculo, fazer uma pesquisa de um arquivo).

O resultado mostrado na tela é a imagem produzida e gerada no computador (pela UCP) e transmitida no formato binário para o controlador ou interface.

Diferentes tecnologias de fabricação de monitores de vídeos estão disponíveis atualmente. Segundo Monteiro (2007), os monitores de vídeo podem ser classificados de duas formas: de acordo com a tecnologia de criação e apresentação da imagem e quanto à forma com que os *bits* são passados do sistema para o vídeo. Sendo assim, as principais tecnologias são:

- a) CRT (Tubos de Raios Catódicos);
- b) LED (Diodos Emissores de Luz);
- c) LCD (Vídeos de Cristal Líquido);
- d) TDP (Vídeos com Painel Estreito).

Atualmente, os populares monitores de vídeo CRT (inclusive os utilizados nos televisores) estão gradativamente sendo substituídos pelos vídeos LCD. Nos vídeos CRT, a imagem se forma em uma tela, a qual é constituída por uma película de fósforo e, uma vez que um de seus pontos de imagem é atingido por um **elétron**, ele emite luz – e isso é o que o olho humano capta (MONTEIRO, 2007).

Os vídeos LCD são menos volumosos que os CRT, além de serem mais econômicos em relação ao consumo de energia e mais rápidos. São encontrados em *palmtops* e *notebooks*, por exemplo, passando a ser cada vez mais comuns em PCs.

A-Z

Elétron

Elétron: é uma partícula sub-atômica e de carga negativa, responsável pela criação de campos magnéticos e eléctricos

No monitor LCD é utilizada tecnologia de cristais líquidos para formar a imagem. Os cristais líquidos são substâncias que têm sua **estrutura molecular** alterada quando recebem corrente elétrica. Em seu estado normal, essas substâncias são transparentes, mas ao receberem uma carga elétrica tornam-se opacas, impedindo a passagem da luz. Mas nos vídeos LCD atuais, como os que são usados em *notebooks*, existem estados intermediários, possibilitando a formação das tonalidades de cinza ou em cores. Os tons intermediários são gerados pela aplicação de diferentes níveis de tensão (GDHPRESS, 2010).

O Guia GDH Press informa que para formar a tela de um monitor, uma fina camada de cristal líquido é colocada entre duas camadas de vidro. Essas finas placas possuem pequenas cavidades isoladas entre si, cada uma com um eletrodo ligado a um transistor. Cada uma dessas cavidades representa um dos pontos da imagem. Essa espécie de “sanduíche” é, por sua vez, colocada entre duas camadas de um elemento polarizador. Atrás dessa tela é instalada uma fonte de luz, geralmente composta de lâmpadas fluorescentes (usadas por gerarem pouco calor) ou então LEDs, responsáveis pela iluminação da tela.

A Figura 7.5 mostra um monitor de vídeo CRT, e a Figura 7.6 mostra outro monitor de vídeo LCD.



Figura 7.5: Exemplo de monitor CRT

Fonte: http://ocularis.es/blog/pics/monitor_CRT_17.jpg



Figura 7.6: Exemplo de monitor LCD

Fonte: <http://www.universotech.com/ut/wp-content/uploads/2009/04/monitorlcd.jpg>

Outra tecnologia de monitores de vídeo que vem ganhando espaço são os de plasma, os quais possuem uma estrutura que lembra os monitores LCD. Esse tipo de monitor também apresenta duas placas de vidro e eletrodos que aplicam tensões elétricas nas células que geram a imagem. O que realmente difere são as células, ou seja, ao invés de cristais líquidos, os monitores de plasma são formados por estruturas com o mesmo princípio de funcionamento das lâmpadas fluorescentes. De forma geral, é possível dizer que em um monitor de plasma a imagem é formada por alguns milhões de minúsculas lâmpadas fluorescentes verdes, azuis e vermelhas (GDHPRESS, 2010). Em se tratando de monitores de vídeo, um aspecto importante a ser considerado é a sua resolução, a qual é medida pela quantidade de *pixels* que pode ser apresentada em uma tela. Monteiro (2007), defende que essa quantidade seja descrita em termos de dois valores: a quantidade de *pixels* mostrados horizontalmente (em uma linha) e a quantidade de pixels mostrados verticalmente (em uma coluna). Por exemplo, para uma resolução de 800 x 600, significa que existirão 800 *pixels* em cada linha e 600 *pixels* em cada coluna. Quanto maior a resolução, maior a qualidade da imagem, ou seja, mais nítida e definida ela será.

Atualmente os padrões de resolução mais comuns disponíveis no mercado são apresentados na Tabela 7.2, a qual mostra o período aproximado de surgimento de cada padrão, sua descrição, resolução típica e número de cores.

A-Z

Pixel

Pixel: é o menor ponto que forma uma imagem digital, sendo que o conjunto de milhares de pixels forma uma imagem inteira

Tabela 7.2: Padrões de resolução mais comuns

Data	Standard Padrão	Description Descrição	Typical Resolution Resolução típica	Number of Colors Número de Cores
1981 1981	CGA CGA	Color Graphics Color Graphics Adapter Adaptador	640x200 640x200 160x200 160x200	2 2 16 16
1984 1984	EGA EGA	Enhanced Graphics Enhanced Graphics Adapter Adaptador	640x350 640x350	16 from 64 16 de 64
1987 1987	VGA VGA	Video Graphics Video Graphics Array Array	640x480 640x480 320x200 320x200	16 from 262,144 16 a partir de 262.144 256 256
	SVGA SVGA	Super Video Graphics Super Video Graphics Array Array	800x600 800x600	256 to 16.7 million 256-16700000
	8514/A 8514 / A	IBM interlaced standard IBM interlaced standard	1024x768 1024x768	16 from 262,144 16 a partir de 262.144
1990 1990	XGA XGA	Extended Graphics Array Extended Graphics Array	1024x768 1024x768	16.7 million 16700000
	SXGA SXGA	Super Extended Graphics Array Super Extended Graphics Array	1280x1024 1280x1024	16.7 million 16700000
	UXGA UXGA	Ultra XGA Ultra XGA	1600x1200 1600x1200	16.7 million 16700000
	WXGA WXGA	Wide XGA Wide XGA	1366x768 1366x768	16.7 million 16700000

Fonte: <http://www.infocellar.com/hardware/ga.htm>



Como Funciona o LCD.
Disponível em: <http://www.gdhpess.com.br/hmc/leia/index.php?p=cap7-3>.
Acesso em: 19 jul.2010.

Monitores de Vídeo.
Disponível em: <http://www.clubedohardware.com.br/artigos/Monitores-de-Video/920/1>.
Acesso em: 19 jul. 2010.

Assim, os controladores de vídeo disponíveis no mercado devem ser compatíveis com os padrões a que são indicados (ex.: XGA, WXGA), apresentando assim um nível menor ou maior de qualidade da imagem projetada no monitor de vídeo.

7.1.1.3 Impressoras

A impressora, considerando seus diversos tipos e formatos, é o um dispositivo de saída de extrema importância, pois é ela que é responsável por passar para o papel grande parte dos resultados de processamento da UCP, seja em formato de texto ou imagem.

Existem dois tipos básicos de impressoras, de acordo com as cores que são capazes de imprimir:

- a) Monocromáticas - imprimem com apenas uma cor e são mais rápidas que as coloridas;
- b) Coloridas - são capazes de criar imagens dividindo a página em milhares ou milhões de pequenos pontos, denominados *dots*, os quais podem

ser endereçados pelo computador. Dessa forma, durante a impressão, o **cabeçote** da impressora move-se no sentido horizontal e o papel movimenta-se no sentido vertical, podendo alternar a impressão de pontos coloridos com pontos em branco em cada linha.

A-Z

Cabeçote

Cabeçote: cabeça de impressão, responsável pela distribuição da cor nos pontos adequados para a formação do texto ou imagem, ponto a ponto.

Segundo Monteiro (2007), uma impressora deve ser analisada de acordo com algumas características básicas que podem determinar seu desempenho em relação a outros dispositivos.

A primeira característica está ligada ao volume de impressão que ela suporta em uma unidade de tempo. As impressoras podem indicar sua capacidade (ou velocidade) de impressão em caracteres por segundo (CPS), em linhas por minuto (LPM) e em páginas por minuto (PPM). Essa capacidade de uma impressora vai depender das tecnologias utilizadas na sua fabricação.

A segunda característica diz respeito às tecnologias de fabricação de impressoras, o que influencia diretamente na forma como os caracteres, a serem impressos, são gerados. Apresentamos, a seguir, algumas das principais tecnologias utilizadas no processo de fabricação de impressoras:

a) Matriciais (de impacto)

Enquadram-se na categoria de impressoras de impacto. Esse tipo de impressora se baseia normalmente na utilização de agulhas, que, quando pressionadas contra uma fita que contém um tipo de tinta, imprimem o símbolo no papel que se encontra atrás da fita. A Figura 7.7 mostra uma cabeça de leitura realizando a impressão por meio de uma fita, sendo possível observar as agulhas (pinos) responsáveis pela impressão.

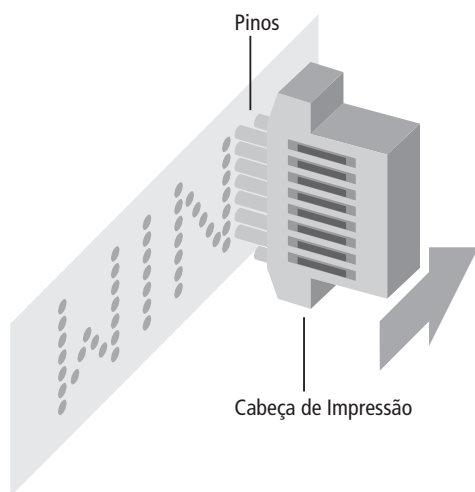


Figura 7.7: Mecanismo de impressão – impressora matricial

Fonte: <http://www.infowester.com/impressoras.php>

Segundo Monteiro (2007), o nome “matricial” se deve ao fato de que os caracteres impressos são formados por uma matriz de pontos.



Figura 7.8: Impressora matricial Okidata – ML 320

Fonte: http://www.intersolucao.com.br/produtosdetalhes_sv.asp?ProdutoID=46

Esse tipo de impressora é confiável, mas ela é lenta e barulhenta. É utilizada para impressão de formulários, relatórios, pedidos, notas fiscais, enfim, documentos de uso frequente das empresas. Também para impressão em espaços pequenos (ex.: de cupom fiscal) e com a utilização de vias carbonadas. Já foram impressoras baratas, mas, atualmente, em função da dificuldade de encontrá-las, tendo em vista o surgimento de novas tecnologias, têm se tornando bastante caras. A Figura 7.8 apresenta um exemplo de impressora matricial.

b) Jato de tinta (*ink-jet*)

Uma das principais diferenças das impressoras jato de tinta para as matriciais é a forma como os caracteres são projetados no papel e o fato de não serem impressoras de impacto. Ambas se utilizam de uma matriz de pontos, mas diferem na técnica utilizada para criar os pontos no papel. De acordo com Monteiro (2007), neste tipo de impressora, o ponto é o resultado de uma gota de tinta que é depositada no papel e secada por calor para não escorrer – essa tecnologia de aquecimento também é denominada *bubble-jet*. Diversas dessas gotas depositadas no papel moldam o formato do caractere, de modo semelhante aos pontos obtidos pela projeção das agulhas em impressoras matriciais. A Figura 7.9 mostra uma impressora jato de tinta atual.



Figura 7.9: Impressora jato de tinta *Epson Stylus Photo T50*

Fonte: <http://www.ms-service.com.br/produto.asp?id=00003582>

Esse tipo de impressora geralmente é de baixo custo, silenciosa, e com impressão de alta qualidade. Como a tinta é disponibilizada por cartuchos (somente preto ou colorido), apresenta um custo relativamente maior que as matriciais.

c) Impressora a *laser*

Trata-se de um tipo de impressora que tem se tornado bastante popular, não somente no ambiente empresarial, mas também no doméstico. Isso foi proporcionado pela baixa relação entre custo e desempenho. A Figura 7.10 apresenta uma impressora a laser típica.



Figura 7.10: Impressora a *laser* monocromática *Xerox Phaser 3124*

Fonte: <http://www.submarino.com.br>



Introdução às Impressoras Matriciais, Jato de Tinta e Laser.

Disponível em:

<http://www.infowester.com/impressoras.php>.

Acesso em: 19 jul. 2010



Faça uma pesquisa sobre os diversos tipos de monitores de vídeo e impressoras disponíveis atualmente no mercado, destacando para cada um deles suas principais características (ex.: marca, modelo, valor, qualidade), além das características específicas de cada equipamento (ex.: resolução de vídeo, número de páginas por minuto para cada impressora). Essa pesquisa deverá ser postada no AVEA no formato de um arquivo de texto.

Diferentemente das impressoras jato de tinta, em que a impressão ocorria pela aplicação de tinta ao papel para a formação dos caracteres, as impressoras a *laser* se utilizam de um produto chamado *toner*. Monteiro (2007) explica que o mecanismo de impressão funciona de modo semelhante ao das copiadoras de documentos. A ideia é formar, em um cilindro fotossensitivo, uma imagem da página que será impressa. Na sequência, o *toner*, composto de partículas minúsculas, é espalhado sobre a imagem criada no cilindro. Por fim, a imagem é transferida do cilindro para um papel e secada por intenso calor. Feito isso, o cilindro deve ter sua imagem apagada para que uma nova imagem possa ser nele criada. O autor destaca ainda que a imagem é formada no cilindro através de um feixe de *laser* que é aceso e apagado a cada ponto do cilindro (como *pixels* em um vídeo), conforme a configuração binária e a localização dos caracteres que se quer imprimir.

Além dos formatos de impressoras descritos acima, existem outras opções, as quais são desenvolvidas para trabalhos específicos e não são comumente encontradas. Um exemplo é o *plotter* (para grandes impressões, por exemplo, de um *banner*).

7.1.1.4 Discos magnéticos

Os discos magnéticos abrangem diversos dispositivos utilizados pelo sistema de computação, tais como: disco rígido, os antigos disquetes e CD-ROM. Esses dispositivos de E/S, considerados memórias secundárias (de armazenamento permanente de um volume considerável de dados), são largamente utilizados, com exceção dos disquetes ou discos flexíveis.

O disco magnético (disco rígido ou *hard disk*) é uma das principais unidades de E/S de um computador; é onde ficam localizados todos os dados (pastas e arquivos) e os programas a serem carregados para a memória principal para execução.

Um disco rígido, conforme mostra a Figura 7.11, é constituído de uma pilha de discos metálicos (normalmente de alumínio), inseridos sobre um mesmo eixo. Cada disco é uma superfície circular fina e coberta com uma camada de material magnetizável, o qual pode receber gravações. O material pode estar presente em uma ou em ambas as superfícies do disco, as quais são normalmente chamadas de faces (lados). Ainda segundo o autor, normalmente, um disco magnético é fabricado com dupla face de armazenamento, de modo que tenha maior capacidade de armazenar (MONTEIRO, 2007).



Figura 7.11: Modelo externo de disco rígido típico.

Fonte: <http://www.clubedohardware.com.br/artigos/1023>

Os discos são montados em um eixo, o qual sofre rotações provocadas por um motor de rotação, que é responsável por manter uma velocidade constante. Inicialmente os discos rígidos utilizavam motores de 3.600 rotações por minuto (RPM). Atualmente, os discos rígidos mais utilizados apresentam 5.600 ou 7.200 RPM. Existem discos com medidas de rotação maiores (ex.: 10.000 RPM), mas não são tão populares. Quanto maior a velocidade de rotação, maior é o preço, mas também maior o desempenho do disco.

De acordo com Monteiro (2007), sobre a superfície de um disco, há um elemento mecânico denominado **braço**, o qual transporta uma cabeça de leitura/gravação, efetuando um movimento transversal sobre as **trilhas** dispostas no disco, de modo a realizar as operações de leitura e gravação sobre a trilha.

Monteiro (2007) afirma, ainda, que as diversas trilhas de um disco são numeradas, ou seja, possuem um endereço específico, a partir da trilha mais externa (endereço 0), até a trilha mais interna (endereço N-1). Dessa forma, como cada trilha possui grande capacidade de armazenamento de *bytes*, elas são divididas em “pedaços” menores, chamados *setores*. A Figura 7.12 mostra os conceitos de trilha e **setor**.

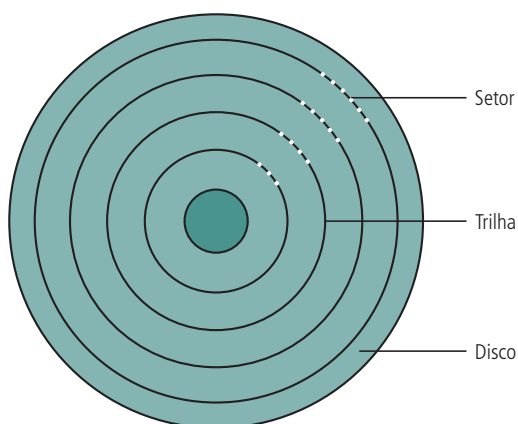


Figura 7.12: Trilhas e setores

Fonte: <http://www.infowester.com/hds1.php>

A-Z

Trilha

Trilha: é um caminho (linha) circular na superfície de um disco, no qual a informação é magneticamente gravada e do qual a informação gravada é lida. Pode-se dizer que uma trilha é um conjunto de setores consecutivos no disco que contêm blocos de dados. Cada trilha armazena a mesma quantidade de *bytes*

Setor

Setor: é uma subdivisão de uma trilha. Seria análogo a uma fatia de uma pizza. Cada setor armazena uma determinada quantidade de dados.

Conforme apresentado, o disco rígido pode conter vários discos. Existe uma cabeça de leitura/gravação para cada face (lado) do disco. As cabeças ficam montadas em um braço, movido por um elemento chamado **atuador**. Segundo Monteiro (2007), como os braços/cabeças se movimentam juntos, quando o atuador se desloca para acessar uma determinada trilha de certa superfície (ex.: trilha 12 da superfície 05), todas as cabeças estacionam sobre a trilha de mesmo endereço (no exemplo, trilha 12) em cada superfície (ou face de um dos discos).

Dessa forma o conjunto de trilhas de mesmo endereço (localizadas em cada superfície), acessado em um único movimento do atuador, denomina-se **cilindro**. Destaca-se que o acesso por cilindro aumenta a produtividade do sistema de disco, quando se movimentam grandes volumes de dados, pelo fato de economizar tempo de busca (*seek*), em relação ao acesso trilha por trilha (MONTEIRO, 2007).

Pelo fato de as cabeças de leitura/gravação estarem afixadas em uma mesma haste, todas as cabeças movimentam-se juntas. A Figura 7.13 apresenta o esquema da estrutura de um disco rígido com essas características. Já a Figura 7.14 mostra a imagem interna de um disco rígido típico, também apontando os principais elementos desse dispositivo.

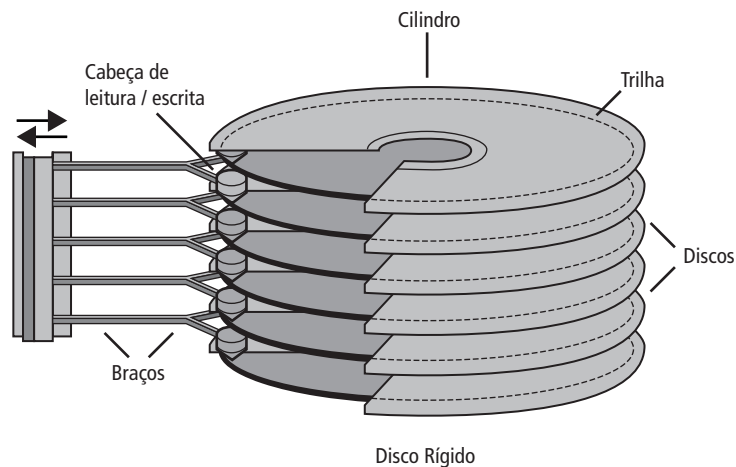


Figura 7.13: Esquema básico da estrutura de um disco rígido

Fonte: <http://www.portalsaofrancisco.com.br/alfa/historia-do-computador/unidades-de-memoria.php>

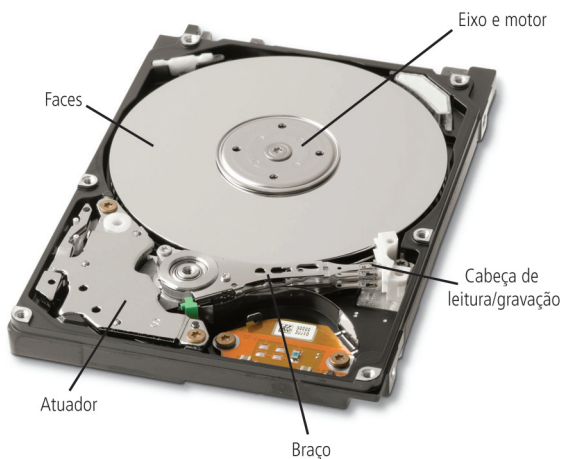


Figura 7.14: Imagem interna de um disco rígido típico

Fonte: <http://www.fabiomw.com/blog/index.php/Windows/Page-2.html>

Um aspecto importante dos discos rígidos e que está diretamente relacionado ao seu desempenho é o tempo de acesso, o qual, segundo Monteiro (2007), é o período gasto entre a ordem de acesso (de posse do respectivo endereço) e o final da transferência dos *bits*. Segundo o autor, esse tempo é formado pela soma de três tempos menores, conforme segue:

- a) Tempo de busca (seek) - gasto para interpretação do endereço pela unidade de controle e movimento mecânico do braço para cima da trilha desejada;
- b) Tempo de latência - período decorrido entre a chegada da cabeça de leitura e gravação sobre a trilha e a passagem do bloco (setor) desejada sobre a referida cabeça (o que depende da velocidade de rotação do disco);
- c) Tempo de transferência - gasto para a efetiva transmissão dos sinais elétricos (*bits*) para o dispositivo destinatário.

Resumo

Nesta aula discutimos alguns dos conceitos e características dos dispositivos de E/S. Classificamos tais dispositivos em duas categorias: entrada (teclado, *mouse*, *webcam*, *modem*, disco rígido) e saída (impressoras, disco rígido, monitor de vídeo). Destacamos que os discos rígidos, assim como outros meios de armazenamento, são dispositivos tanto de entrada quanto de saída.

Sublinhamos algumas observações relevantes que influenciam na comunicação dos dispositivos com a UCP e memória principal: grande diferença de velocidade entre UCP e os dispositivos de E/S, além de grandes diferenças de velocidade entre os próprios dispositivos, como por exemplo, o disco rí-



Anatomia de um Disco Rígido.
Disponível em:
<http://www.clubedohardware.com.br/artigos/Anatomia-de-um-Disco-Rigido/1056/5>.
Acesso em: 19 jul. 2010.

Conhecendo o Disco Rígido (HD).
Disponível em:
<http://www.infowester.com/hds1.php>.
Acesso em: 19 jul. 2010.

Mouses: funcionamento, tipos e principais características.
Disponível em:
<http://www.infowester.com/mouse.php>.
Acesso em: 19 jul. 2010.

gido que é mais rápido que o teclado. Além da velocidade, outro aspecto que diferencia os dispositivos de E/S é a sua forma de comunicação. Assim, segundo Monteiro (2007), pode-se dizer que a comunicação entre o núcleo do computador e os dispositivos de E/S poderia ser classificada em:

- Comunicação serial: a informação pode ser transmitida/recebida, *bit* a *bit*, um em seguida do outro;
- Comunicação paralela: a informação pode ser transmitida/recebida em grupos de *bits* de cada vez, isto é, um grupo de *bits* é transmitido simultaneamente de cada vez.

Na prática a UCP não se conecta diretamente com cada periférico, mas sim com dispositivos que realizam a “tradução” e a compatibilização das características de um (dispositivo de E/S) para o outro (UCP/MP), além de realizar outras tarefas de controle.

Tais dispositivos possuem algumas denominações. A mais comum seria interface de E/S, mas comercialmente podem ser encontrados como controlador ou adaptador, adicionando-se o nome do dispositivo (ex.: controlador de vídeo, controlador de disco).

Diversas formas de comunicação entre UCP e memória principal foram propostas, as quais sofreram melhorias ao longo do tempo, buscando sempre alcançar uma melhor utilização da UCP e um melhor desempenho para o sistema como um todo. Murdocca (2000) destaca três métodos para gerenciar a entrada e saída, são eles: Entrada e saída programada, entrada e saída controlada por interrupção, acesso direto à memória (DMA) – o que é utilizado atualmente.

O DMA consiste em fazer com que o controlador de um dispositivo leia e escreva dados de ou para a memória sem a intervenção da UCP. A UCP é avisada apenas no início e no final da operação de transferência entre dispositivo e memória principal. Esse método é utilizado atualmente pelos computadores.

O aviso à UCP é gerado por uma interrupção, ou seja, um evento ou um aviso à UCP de que algum dispositivo está solicitando a realização de uma operação (ex.: recebimento de dados via rede, a pressão de uma tecla).

Um dispositivo chamado controlador de interrupções é responsável por receber e encaminhar pedidos de interrupções à UCP, já que existe somente uma linha de interrupção para essa finalidade.

Alguns dos principais dispositivos de E/S foram apresentados, destacando-se suas principais características de construção e funcionamento: teclado, vídeo, impressoras e disco rígido.

Atividades de aprendizagem

As questões “e”, “f”, “h” e “i” devem ser respondidas no *blog* “Aspectos de Entrada e Saída” disponível no AVEA. As demais questões devem ser respondidas em arquivo texto e postadas como atividade no AVEA.

Responda aos seguintes questionamentos sobre os dispositivos de E/S:

- a) O que caracteriza os dispositivos de entrada e os de saída? Exemplifique cada um deles.
- b) Explique a diferença entre a comunicação serial e paralela para dispositivos de E/S.
- c) No que consiste um controlador ou interface de E/S? Qual a sua função no sistema computacional? Exemplifique.
- d) Quais são as metodologias de comunicação entre UCP e dispositivos de E/S? Qual a metodologia utilizada atualmente pelos computadores? Qual a vantagem dessa tecnologia em relação às suas antecessoras?
- e) O que é uma interrupção? Exemplifique.
- f) Qual a relação de uma interrupção com as metodologias de comunicação entre UCP e dispositivos de E/S?
- g) Quais as tecnologias de fabricação de teclas existentes? Qual é a mais utilizada? Explique o seu funcionamento.
- h) Explique as diferenças existentes entre um monitor de vídeo do tipo CRT e do tipo LCD.
- i) Explique o funcionamento de cada um dos tipos de impressoras.
- j) Explique o funcionamento básico dos discos magnéticos.
- k) Qual a relação entre a rotação que ocorre no eixo de um disco rígido com o seu desempenho de leitura/gravação? Explique.

Referências

ALECRIM, E. **Memórias RAM e ROM**. Disponível em: <http://www.infowester.com/memoria.php>. Acesso em: 20 maio 2010.

ALECRIM, E. **Processadores - Parte 1: clock, bits, memória cache e múltiplos núcleos**. Disponível em: <http://www.infowester.com/processadores1.php>. Acesso em: 20 maio 2010.

COMPUTADOR ANALÓGICO. In: WIKIPEDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2010. Disponível em: http://pt.wikipedia.org/w/index.php?title=Computador_anal%C3%B3gico&oldid=21384910>. Acesso em: 22 maio 2010.

CIRCUITO INTEGRADO. In: WIKIPEDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2010. Disponível em: http://pt.wikipedia.org/w/index.php?title=Circuito_integrado&oldid=22745851>. Acesso em: 22 maio 2010.

GDHPRESS. **Como funciona o LCD**. Disponível em: <http://www.gdhpress.com.br/hmc/leia/index.php?p=cap7-3>. Acesso em: 15 jun. 2010.

MONTEIRO, Mário A. **Introdução à organização de computadores**. Rio de Janeiro: LTC, 2007.

MURDOCCA, Miles J. **Introdução à arquitetura de computadores**. Rio de Janeiro: Campus, 2000.

PATTERSON, David A.; HENNESSY, John L. "Organização e projeto de computadores: a interface hardware/software." 2. ed. Rio de Janeiro: LTC, 2000. 551p.

TANENBAUM, Andrew S. **Organização estruturada de computadores**. São Paulo: Pearson Prentice Hall, 2007.

TORRES, G. **ROM**. Disponível em: <http://www.clubedohardware.com.br/dicionario/termo/239>. Acesso em: 21 maio 2010.

Currículo da professora-autora

Possui graduação em Tecnologia em Processamento de Dados pela Universidade Tecnológica Federal do Paraná (1999) e mestrado em Engenharia Elétrica e Informática Industrial, com ênfase em Informática Industrial pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI) da Universidade Tecnológica Federal do Paraná (2006).

Atua há mais de oito anos em cursos superiores da área de informática. Atualmente é professora de ensino superior da Universidade Tecnológica Federal do Paraná (UTFPR). Possui experiência na área de Ciência da Computação, com ênfase em Sistemas de Informação, atuando na realização de trabalhos nos seguintes temas: organização e arquitetura de computadores, engenharia de *software*, sistemas colaborativos e tecnologias educacionais.



