e-Tec Brasil
Escola Técnica Aberta do Brasil

Arquitetura de Computadores

Allan Francisco Forzza Amaral

Curso Técnico em Informática

INSTITUTO FEDERAL
ESPÍRITO SANTO

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Ministério da Educação

e-Tec Brasil
Escola Técnica Aberta do Brasil

Arquitetura de Computadores

Allan Francisco Forzza Amaral

Colatina - ES
2010

Presidência da República Federativa do Brasil

Ministério da Educação

Secretaria de Educação a Distância

© Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo

Este Caderno foi elaborado em parceria entre o Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo e a Universidade Federal de Santa Catarina para o Sistema Escola Técnica Aberta do Brasil – e-Tec Brasil.

Equipe de Elaboração
Instituto Federal do Espírito Santo – IFES

Coordenação do Curso
Joao Henrique Caminhas Ferreira/IFES

Professor-autor
Allan Francisco Forzza Amaral/IFES

Comissão de Acompanhamento e Validação
Universidade Federal de Santa Catarina – UFSC

Coordenação Institucional
Araci Hack Catapan/UFSC

Coordenação do Projeto
Silvia Modesto Nassar/UFSC

Coordenação de Design Instrucional
Beatriz Helena Dal Molin/UNIOESTE e UFSC

Coordenação de Design Gráfico
Carlos Antônio Ramirez Righi/UFSC

Design Instrucional
Alessandro Poleto Oliveira/IFES

Web Master
Rafaela Lunardi Comarella/UFSC

Web Design
CEAD/IFES

Diagramação
André Rodrigues da Silva/UFSC
Andréia Takeuchi/UFSC
Caroline Ferreira da Silva/UFSC
Juliana Tonietto/UFSC

Revisão
Luciane Ferreira Lacerda/ IFES

Projeto Gráfico
e-Tec/MEC

A485a **Amaral, Allan Francisco Forzza**

Arquitetura de computadores : Curso Técnico em Informática / Allan Francisco Forzza Amaral. – Colatina: CEAD / Ifes, 2010. 100 p. : il.

1. Arquitetura de computador. 2. Computadores. 3. Hardware. 4. Material didático. I. Instituto Federal do Espírito Santo. II. Título.

CDD: 004.22

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 Distancia (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 do professor-autor **9**

Apresentação da disciplina **11**

Projeto instrucional **13**

Aula 1 – Introdução à Arquitetura de Computadores **15**
1.1 Conceitos básicos de informática 15
1.2 Computadores: *hardware* e *software* 16

Aula 2 – Gerações e evoluções dos computadores **23**
2.1 Histórico 23
2.2 Gerações, evoluções e estágio atual 24

Aula 3 – Aritmética computacional e representação de dados **31**
3.1 Conceitos básicos: dado, informação e representação 31
3.2 Sistemas de numeração 31
3.3 Conversão de bases 34
3.4 Formatos de representação de dados 36

Aula 4 – Organização de Computadores **41**
4.1 Modelo de Von Neumann 41
4.2 Relação entre os componentes e suas funções 43
4.3 Funções dos computadores 45

Aula 5 – Barramento do sistema **49**
5.1 Estruturas de interconexão 49
5.2 Estruturas de barramentos 51
5.3 Múltiplos barramentos 53

Aula 6 - Unidade Central de Processamento **57**
6.1 Unidade Lógica e Aritmética (ULA) 57
6.2 Unidade de Controle (UC) 60
6.3 Organização de registradores 62

Aula 7 – Memória interna **67**
7.1 Memória principal 67
7.2 Memória *cache* 71
7.3 Comparação dos parâmetros das memórias 73

Aula 8 – Memória externa **77**
8.1 Organização e funcionamento dos discos magnéticos 77
8.2 Armazenamento com discos ópticos 81

Aula 9 – Entrada e Saída (E/S) **87**
9.1 Dispositivos externos 88
9.2 Módulos e funções de E/S 90
9.3 Principais técnicas de E/S 92

Referências **98**

Currículo do professor-autor **99**

Palavra do professor-autor

Olá,

Parabéns, caro estudante! Você está iniciando mais uma etapa do curso técnico em informática a distância. A equipe instrucional elaborou todo o material necessário ao suporte para o seu aprendizado. Neste formato, a disciplina de arquitetura de computadores foi elaborada pensando numa leitura rápida e dinâmica, abordando o centro de cada conteúdo, explanado em aulas bem objetivas. Como já é do seu conhecimento, estudar a distância é uma tarefa que envolve sua aplicação na resolução dos exercícios, mas contando com todo amparo da equipe que irá te apoiar no processo ensino-aprendizagem. Para que isso ocorra de forma efetiva, faz-se necessário separar um tempo para estudar o material e fazer as leituras complementares indicadas no caderno. Esperamos que você utilize todos os recursos do ambiente disponíveis para dar andamento aos estudos e avançar pelos módulos.

Um cordial abraço e sucesso!
Prof. Allan Francisco Forzza Amaral

Apresentação da disciplina

Nesta disciplina você obterá conhecimentos na área de arquitetura de computadores e como os componentes são organizados.

A arquitetura e os componentes de hardware e software evoluem rapidamente e fica impossível elaborar materiais atualizados. Pensando nisso, fo¬ram abordados itens básicos que quase nunca mudam, explicando de forma ilustrada o funcionamento interno dos principais componentes que formam um sistema computacional.

Esse material proporcionar-lhe-á muitos conhecimentos sobre os principais itens que formam o computador. A abordagem dada refere-se, principalmente, como os componentes internos dos computadores funcionam logicamente e como estão organizados. Foram preparadas nove aulas, ajustadas da seguinte forma:

Nas três primeiras aulas, abordamos os conceitos iniciais de informática (como *hardware* e *software*), o histórico e evolução dos computadores e por fim um estudo da aritmética computacional. Nas duas aulas seguintes apresentamos a organização dos computadores e como é sua estrutura de interligação. Em seguida, abordamos assuntos específicos sobre a organização de cada componente, como por exemplo, Unidade Central de Processamento e Memórias.

É fundamental dizer que algumas abordagens precisam ser aprofundadas com leituras extras. É necessário que você faça pesquisas e que busque novos conhecimentos sobre o assunto em revistas, livros e *sites* especializados.

Seu desempenho nesta disciplina será proveitoso à medida que for resolvendo os exercícios propostos e participando das atividades, em que poderá obter e compartilhar novos conhecimentos sobre temas atuais como evolução do funcionamento das arquiteturas dos computadores.

Siga em frente!

Prof. Allan Francisco Forzza Amaral

Projeto instrucional

Disciplina: Arquitetura de Computadores: (carga horária: 60 horas).

Ementa: História da computação. Fundamentos de sistemas de computação. Sistemas de Numeração. Representação dos Dados. Organização de Computadores. Memórias. Unidade Central de Processamento. Dispositivos de Entrada/Saída.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Introdução à Arquitetura de Computadores	Compreender os conceitos iniciais de informática. Diferenciar os principais elementos de *Hardware* e *Software*. Apresentar os principais itens de *Hardware* e *Software*. Compreender os termos utilizados no meio computacional	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	10
2. Gerações e evoluções dos computadores	Compreender a evolução do modelo computacional. Compreender a funcionalidade dos computadores na época vigente. Comparar modelos antigos a evolução dos computadores.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	05
3. Aritmética computacional e representação de dados	Compreender a linguagem numérica utilizada pelos computadores. Compreender as formas numéricas que se apresentam num sistema computacional. Compreender as formas de conversão de bases dos sistemas reconhecidos pelo homem e pela máquina. Compreender os principais tipos de dados e sua representação interna. Compreender o formato de apresentação dos dados no âmbito computacional.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	05
4. Organização de computadores	Compreender as operações básicas realizadas pelos computadores. Conhecer os principais modelos de organização de computadores. Compreender como estão relacionados os principais componentes dos computadores.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	05

continua

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
5. Barramento do sistema	Compreender o modelo de interconexão dos principais componentes. Conhecer as características básicas dos barramentos. Compreender as estruturas básicas dos barramentos.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	05
6. Unidade Central de Processamento	Examinar a estrutura e o funcionamento interno da CPU. Abordar questões relativas à arquitetura da CPU. Tratar aspectos de organização da CPU.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	10
7. Memória interna	Abordar as características fundamentais dos sistemas de memórias dos computadores. Examinar a hierarquia das memórias internas e seus subsistemas. Apresentar as variedades de tipos, tecnologias, organização, desempenho e custos.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	05
8. Memória externa	Abordar a organização e formatação de dados em discos magnéticos e ópticos. Examinar as características físicas dos discos. Apresentar os parâmetros de desempenhos dos discos.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	05
9. Entrada e Saída	Conhecer as interfaces de E/S de contato com o mundo externo. Abordar as principais técnicas de E/S. Apresentar as principais interfaces de E/S.	Caderno e Ambiente Virtual de Aprendizagem. www.cead.ifes.edu.br	10
			conclusão

Aula 1 – Introdução à Arquitetura de Computadores

Objetivos

Compreender os conceitos iniciais de informática.

Diferenciar os principais elementos de *Hardware* e *Software*.

Apresentar os principais itens de *Hardware* e *Software*.

Compreender os termos utilizados no meio computacional.

1.1 Conceitos básicos de informática

Um computador é uma máquina (conjunto de partes eletrônicas e eletromecânicas) capaz de sistematicamente coletar, manipular e fornecer os resultados da manipulação de dados para um ou mais objetivos. Por ser uma máquina composta de vários circuitos e componentes eletrônicos, também é chamado de equipamento de processamento eletrônico de dados.

A-Z

Informática
A informática pode ser considerada como "informação automática", ou seja, utilização de métodos e técnicas no tratamento automático da informação. Para tal, é preciso uma ferramenta adequada: o computador eletrônico.

O esquema básico de um processamento de **dados** (manual ou automático) resulta em um produto acabado: a informação. Assim, os dados precisam ser processados para que algum resultado tenha significado para alguém ou para o próprio computador.

A-Z

Dado
Dado é tudo aquilo que sozinho, pode não significar nada. Por exemplo: 5 m, 10 kg, 20 km/h. Mas se dissermos "uma passarela tem 5 m de altura", isso pode se constituir numa informação para um motorista de caminhão, que evitaria trafegar naquela via sabendo que seu caminhão tem uma altura maior. Já a informação subentende dados organizados segundo uma orientação específica, para o atendimento ou emprego de uma pessoa ou grupo que os recebe.

O computador processa os dados, orientado por um conjunto de instruções, para produzir resultados completos com um mínimo de intervenção humana. Entre seus vários benefícios, podemos citar:

a) grande velocidade no processamento e disponibilização de informações;

b) precisão no fornecimento das informações;

c) adequação para execução de tarefas repetitivas;

d) redução de custos operacionais;

e) compartilhamento de dados.

A busca de técnicas mais eficazes de processamento de dados, aliada ao natural avanço tecnológico em diversos outros ramos de atividade, como a eletrônica e a mecânica, por exemplo, conduziu ao desenvolvimento de equipamentos de processamento eletrônico de dados - os computadores - capazes de coletar, armazenar e processar dados muito mais rapidamente que os antigos meios manuais.

1.2 Computadores: *hardware* e *software*

A-Z

Hardware
O *hardware* é a parte física do computador. Já o *software* é o programa ou conjunto de programas que faz o computador funcionar.

Um sistema baseado em computador é, na verdade, composto por *hardware* e *software*. ***Hardware*** é o nome que se dá para a parte física do computador. É tudo que você pode tocar (*mouse*, teclado, caixas de som, placas, fios, componentes em geral). *Software* é o nome que se dá a toda parte lógica do computador, ou seja, os programas que você vê funcionar na tela e que dão "vida" ao computador. Sem um *software* adequado às suas necessidades, o computador, por mais bem equipado e avançado que seja, é completamente inútil.

1.2.1 *Hardware*: componentes dos computadores

1.2.1.1 Unidades de Entrada e Saída

> Os dispositivos de E/S (Entrada e Saída) são constituídos, geralmente, de duas partes: o controlador e o dispositivo propriamente dito. O controlador é um *chip* ou um conjunto de *chips* que controla fisicamente o dispositivo; ele recebe comandos do sistema operacional (*software*), por exemplo, para ler dados dos dispositivos e para enviá-los (TANEMBAUM, 2003).

Unidades de entrada são, portanto, dispositivos físicos que capturam os dados a serem processados. Os dados podem ser do tipo texto, vídeo ou áudio. Para cada tipo de dado temos um dispositivo de entrada especifico para sua leitura: teclado, *mouse, scanner*, etc.

Unidades de saída apresentam os resultados finais do processamento. Para o usuário, os dispositivos de saída são tão normais que quase não são percebidos: monitores de vídeo, impressoras, *plotter*, telas de terminais bancários, impressoras de extratos, painéis de senha, monitores de vídeo, quiosques de consultas de preços, etc.

Existem também alguns dispositivos que podem ser classificados com ambas as denominações, entrada e saída, por exemplo: unidades de disco (discos rígidos, disquetes, unidades leitoras e gravadoras de CD e DVD), unidades

de armazenamento USB (Universal Serial Bus – Barramento Serial Universal) – conhecidos como *pen drives* –, ***telas touch screen***, etc.

Figura 1.1: Exemplos de dispositivos de entrada e saída
Fonte: CEAD/ IFES

A-Z

Telas touch screen
As telas *Touch Screen* são aquelas em que a entrada dos dados é feita tocando ou escrevendo no próprio monitor, pois ele é sensível ao toque. Essa tecnologia está presente em alguns celulares e terminais de autoatendimento em bancos

1.2.1.2 Memória

O computador deve ser dotado de alguma forma de armazenamento (temporário ou permanente) para que os dados coletados ou processados possam ser armazenados. A essa estrutura damos o nome genérico de memória (não está contextualizado aqui o tipo da memória). A memória armazena, essencialmente, os *bits*. A menor unidade de informação em um computador é o *bit*, que pode assumir os valores 0 ou 1.

Como um único *bit* é insuficiente para representar informações mais complexas, eles são agrupados e combinados. Num primeiro agrupamento, são reunidos em conjuntos de oito bits, recebendo a denominação de *Byte*. Como a unidade *byte* (unidade de medida de armazenamento) também é, consideravelmente, pequena quando indicamos valores mais extensos, utilizamos múltiplos do *byte*: *quilobyte, megabyte, gigabyte, terabyte*, etc.

A-Z

Byte
As unidades de medida de informação indicam a quantidade de informação manipulada. No modelo digital utilizamos potências de base 2. Por exemplo: 1 *byte* é igual a 2^3 *bits*; 1 KB é igual a 2^{10} *bytes*. Já na engenharia, de um modo geral, cada unidade representa 1000 unidades da anterior. Por exemplo: 1 ton = 1000 kg ;1 kg = 1000 g.

1 *Quilobyte* = 1 KB = 1024 *Bytes*,

1 *Megabyte* = 1 MB = 1024 *Kbytes*,

1 *Gigabyte* = 1 GB = 1024 *Mbytes*, e

1 *Terabyte* = 1 TB = 1024 *Gbytes*.

1.2.1.3 Unidade Central de Processamento (UCP)

A CPU (*Central Processing Unit* ou Unidade Central de Processamento), microprocessador ou processador é a parte do computador que interpreta e executa as instruções contidas no *software*. Na maioria das CPUs essa tarefa é dividida entre uma unidade de controle que dirige o fluxo do programa e uma ou mais unidades que executam operações em dados.

> O ciclo básico de execução de qualquer CPU é buscar a primeira instrução da memória, decodificá-la para determinar seus operandos e qual operação executar com os mesmos, executá-la e então buscar, decodificar e executar a instrução subsequente (TANEMBAUM, 2003).

A-Z

ULA
A ULA é responsável por realizar as "contas" (cálculos matemáticos) e pelo uso de lógica de comparação. A UC representa um "coordenador" que controla as buscas das instruções e sincroniza sua execução. Os registradores basicamente armazenam temporariamente os resultados dos processamentos.

A CPU é constituída pelos seguintes componentes: a **ULA** (Unidade de Lógica e Aritmética), a UC (Unidade de Controle) e os registradores. Um tipo de registrador especial é o contador de programa, que contém o endereço de memória da próxima instrução que a CPU vai buscar. Assim, esse registrador vai ser atualizado para conter sempre o endereço da próxima instrução a ser processada.

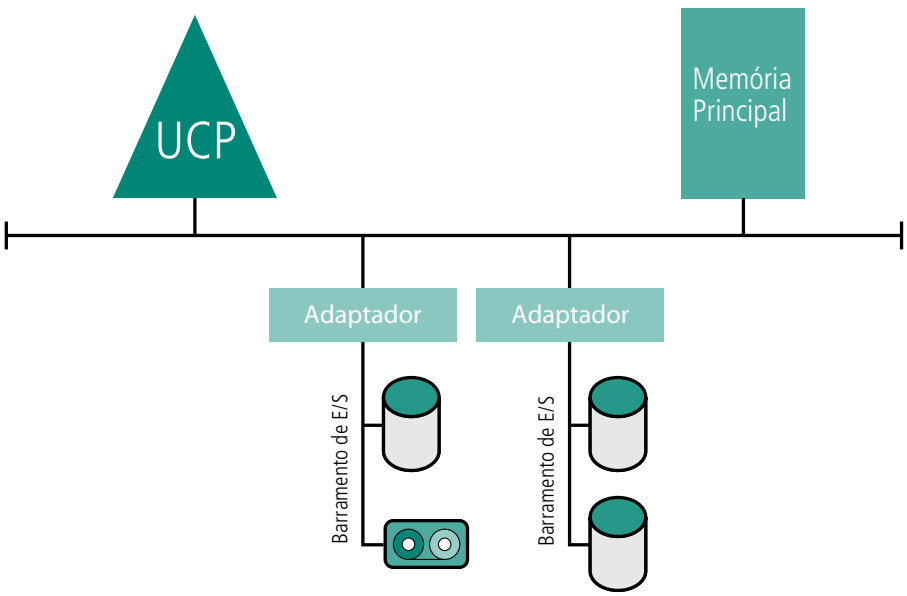


Figura 1.2: Organização e estrutura básica de funcionamento de um sistema computacional
Fonte: Arquitetura de Sistemas Operacionais. Francis B. Machado e Luiz Paulo Maia. 2007(Adaptação)

A Figura 1.2 ilustra um modelo de processamento de dados no qual existe uma entrada de dados, que pode ser originada de vários dispositivos que utilizam o barramento de entrada e saída (E/S), como um teclado ou um CD-ROM. O processamento (que utiliza a CPU e memória) e a saída (a devolução dos dados requeridos pelo usuário) que também utilizam o mesmo barramento

1.2.2 *Software*

Um programa de computador pode ser definido como uma série de instruções ou declarações, em forma inteligível pelo computador, preparada para obter certos resultados. Um programa pode ser chamado de *software*, porém esse termo pode ser utilizado também para um grupo de programas ou para todo o conjunto de programas de um computador.

Os *software* são escritos pelo homem em linguagem de alto nível, como, por exemplo, *Delphi*, C++, *Visual Basic*, dentre outras. Essas linguagens de alto nível possuem compiladores que transformam a linguagem escrita pelo homem em uma linguagem binária, que o computador entende.

Podemos classificar os *software* ou programas de várias formas. A seguir apresentamos uma classificação genérica.

1.2.2.1 *Software* básico

São *software* destinados à operação do computador. Têm como função principal controlar os diversos dispositivos do computador e servir de comunicação intermediária entre o computador e os outros programas normalmente utilizados, o que permite que esses possam ser executados. Sua classificação básica pode ser:

a) Sistemas ou Ambientes Operacionais

Chamado tradicionalmente de Sistema Operacional (ou simplesmente SO), esse *software* é o grande responsável pelo funcionamento do computador, a interação e sincronização de todos os elementos que fazem parte do conjunto computacional. Sem um sistema operacional o computador não passa de um monte de placas de circuito sem função definida. Os sistemas operacionais mais utilizados nos dias de hoje são o *Windows* e o *Linux*.

b) Linguagens

Podemos imaginar o computador como uma calculadora capaz de fazer cálculos muito mais rápido que nós, mas para isso devemos dizer-lhe o quê e como calcular. A função das linguagens de programação é exatamente esta: servir de um meio de comunicação entre computadores e humanos.

Existem basicamente dois tipos de linguagens de programação: as de baixo nível e as de alto nível. As linguagens de baixo nível são interpretadas diretamente pelo computador, tendo um resultado rápido, porém é muito difícil e incômodo se trabalhar com elas. Exemplos de linguagens de baixo nível são o *Basic* e o *Assembly*. As linguagens de alto nível são mais fáceis de trabalhar e de entender; as ações são representadas por palavras geralmente em inglês (por exemplo: *Do*, *Write*, etc.) e foram assim desenvolvidas para

facilitar a memorização e a lógica. Essas linguagens não são interpretadas diretamente pelo computador, sendo necessário traduzi-las para linguagem binária utilizando um programa chamado compilador.

1.2.2.2 *Software* aplicativo

São programas de computadores que têm alguma função específica. Podem ser, basicamente, de:

a) Uso geral

São os programas destinados a nos oferecer certos tipos de serviços. Podemos incluir nessa categoria os processadores de texto, as planilhas eletrônicas, os programas gráficos e os sistemas gerenciadores de banco de dados, exemplificados a seguir:

- Processadores de texto - dentre os vários disponíveis, destacamos os seguintes: *Word (Microsoft)* e *WordPerfect (Corel)*, *OpenEdit*;

- Planilhas eletrônicas ou planilhas de cálculo - dentre as mais comuns, destacamos o *MS-Excel*, o *Lotus* 1-2-3, o Quatro Pro e o *OpenCalc*;

- Programas Gráficos - permitem a criação de figuras e desenhos; alguns possuem recursos extras para animação. Podem ser conjugados com programas que adicionam sons às imagens. Dentre os mais sofisticados destacam-se o *Corel Draw* e o A*dobe PhotoShop*.

b) Uso específico

São *software* desenvolvidos sob encomenda para empresas que utilizam aplicações específicas, como, por exemplo, aplicações de controle financeiro, controle de produção, controle de clientes, faturamento, etc.

Normalmente esses *software* necessitam dos sistemas Gerenciadores de Bancos de Dados (programas ou coleção de programas voltados ao controle de grandes volumes de informações).

Resumo

Nesta aula falamos sobre os conceitos iniciais de informática, como *hardware* e *software*. Também abordamos o conceito de quantificação de informação (como *bits, bytes, Kbytes*, etc.) tão importante no mundo computacional. Citamos também como os computadores são organizados em termos de *hardware* e como os *softwares* são agrupados.

Atividades de aprendizagem

1. Conceitue *hardware* e *software*.

2. Dê exemplo de, no mínimo, três dispositivos de entrada e explique a função de cada um deles.

3. Um *pen drive* se classifica como um dispositivo de entrada, um dispositivo de saída ou dispositivo de entrada e saída?

4. Qual a função da memória em um computador? Como o seu tamanho pode interferir no funcionamento do computador?

5. Quantos MB (*Megabytes*) existem em 2,5 GB (*Gigabytes*)?

6. Cite quais são as funções da Unidade Lógica e Aritmética (ULA)?

7. O que é o *software* Sistema Operacional (SO)? Indique uma função que você mais utiliza desse tipo de *software*.

8. O que é o *software* Aplicativo? Dê alguns exemplos.

9. Pesquise e indique quais são as duas unidades de medida acima do *TeraByte (TB).*

10. Os dispositivos muitas vezes possuem um controlador que faz com que o dispositivo funcione de acordo com o comando do usuário no sistema operacional. Pensando assim, você entende que esse controlador é um *hardware* ou um *software*? Explique.

Aula 2 – Gerações e evoluções dos computadores

Objetivos

Compreender a evolução do modelo computacional.

Compreender a funcionalidade dos computadores na época vigente.

Comparar modelos antigos a evolução dos computadores.

2.1 Histórico

> O primeiro computador digital foi projetado pelo matemático inglês Charles Babbage (1792-1871). Embora, Babbage tenha empregado a maior parte de sua vida e de sua fortuna para construir sua "máquina analítica", ele nunca conseguiu vê-la funcionando de modo apropriado, pois era inteiramente mecânica e a tecnologia de sua época não poderia produzir as rodas, engrenagens e correias de alta precisão que eram necessárias (TANEMBAUM, 2003).

Ainda segundo Tanembaum (2003), foi em meados da década de 40 do século XX, que Howard Aiken, em Harvard; John Von Neumann, no Instituto para Estudos Avançados de Princeton; J. Presper Eckert e Willian Mauchley, na Universidade da Pensilvânia e Konrad Zuse, na Alemanha, entre outros, conseguiram construir máquinas de calcular.

Em 1943, durante a segunda guerra mundial, Mauchley e Eckert propuseram ao exército norte-americano a construção do primeiro computador, que poderia realizar cálculos balísticos com menor margem de erro. Sua conclusão foi anunciada em 1946 e batizado com o nome de ENIAC (*Electronic Numerical Integrator and Computer – Computador Eletrônico Numérico Integrado).*

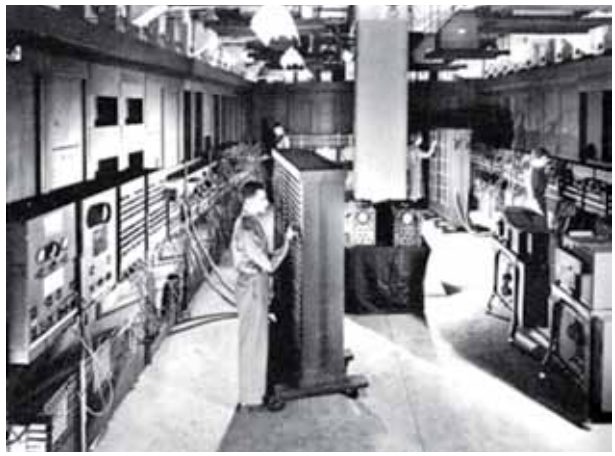


Figura 2.1: O ENIAC preenchia esta sala e não se compara aos miniaturizados e mais potentes computadores atuais
Fonte: Metamorfose Digital. http://www.mdig.com.br/index.php?itemid=692

O ENIAC pesava 30 toneladas, consumia 200.000 watts de potência e ocupava várias salas. Em 1955, um computador pesava 3 toneladas e consumia 50 kw de potência, tendo um custo de US$200.000,00. Com o tempo, os transistores passaram a ser a base da eletrônica, até chegarmos à VLSI (*Very Large Scale Integration* – Integração em Escala Muito Alta): construção de circuitos cada vez menores, mais leves e consumindo menos energia, por terem menor superfície para a dissipação (perda) de energia por calor. Vamos ver detalhes dessa história no item seguinte.

2.2 Gerações, evoluções e estágio atual

Ao longo dos anos os computadores apresentaram enorme evolução, embora alguns conceitos criados nos primórdios da era digital prevalecem até hoje. Alguns saltos evolutivos ou comportamentais caracterizaram o que chamamos de novas "gerações de computadores". Vamos dar um passeio pelas gerações que antecederam nossos poderosos computadores atuais. O mapeamento das datas de evoluções e gerações dos Sistemas Operacionais e Arquitetura de Computadores são, de certa forma, vagas e imprecisas, mas com certa estrutura.

a) Primeira geração (1945 a 1955): Válvulas e Painéis de Programação

Figura 2.2: Relés (E) e Válvulas (D)
Fonte: CEAD/ IFES

Os computadores da época eram máquinas de calcular e tinham seus ciclos medidos em segundos (ciclo: o tempo para buscar um dado, processar e retornar com o resultado). Esse tempo era alto devido ao uso de relés mecânicos muito lentos. Os relés foram mais tarde substituídos por válvulas.

Observe no texto seguinte a linha do tempo dessa geração, bem como os seus acontecimentos mais importantes:

- 1943 a 1945 - J. Presper Eckert, John V. Mauchly e Herman H. Goldstine, nos Estados Unidos, construíram o ENIAC (*Electronic Numerical Integrator Computer*), considerado o primeiro computador programável universal, que foi usado em uma simulação numérica para o projeto da bomba de hidrogênio, em projetos de túneis de vento, geradores de números randômicos e em previsões meteorológicas.

- 1946 - Max Newman e a equipe da Universidade de Manchester, na Inglaterra, com a participação de Alan Turing, construíram o *Manchester Mark I*, que apresentou a primeira visualização na tela de dados contidos na memória e teve o primeiro programa gravado, executado em 21 de junho de 1948.

- 1948 - A equipe da IBM utilizou pela primeira vez o SSEC (*Selective Sequence Electronic Calculator*), uma máquina híbrida de válvulas a vácuo e relés eletromecânicos, para calcular as tabelas de posições da Lua (utilizadas na primeira alunissagem).

- 1951 - J. Presper Eckert e John V. Mauchly construíram o UNIVAC (*Universal Automatic Computer*), o primeiro computador bem sucedido comercialmente, com aplicação na área de administração.

Os primeiros computadores eram usados em aplicações científicas e de engenharia.

b) Segunda geração (1956 a 1965): Computadores transistorizados

A introdução do transistor em meados da década de 50 mudou o quadro radicalmente. Os computadores tornaram-se suficientemente confiáveis para que pudessem ser fabricados e comercializados com a expectativa de que continuariam a funcionar por tempo suficiente para executar algum trabalho útil (TANEMBAUM, 2003).

Tecnologia: diodo discreto e transistor, formados por núcleos magnéticos de memória.

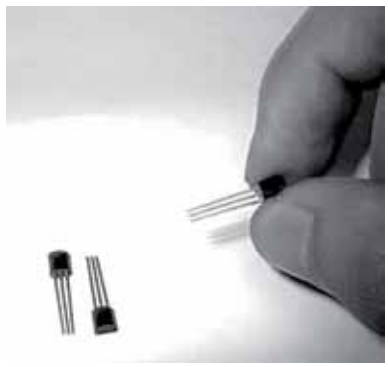


Figura 2.3: Transistores
Fonte: http://waytech.sites.uol.com.br/

Essas máquinas eram denominadas de *mainframes* ou computadores de grande porte. Eram acondicionadas em salas enormes, em ambientes de baixa temperatura, visando preservar seus inúmeros transistores. Pelo seu alto custo, apenas órgãos governamentais e universidades podiam pagar seu preço. Vejamos alguns exemplos:

- Supercomputadores: o LARC *(Livermore Atomic Research Computer* ou Computador de Pesquisas Atômicas) e o IBM 7030 (codinome *Stretch*): utilizavam formas primitivas de processamento paralelo, ou seja, aquele em que o processador pode estar em estágios diferentes da execução de sua instrução.

- Minicomputadores: PDP-8 (construídos com transistores e núcleos magnéticos, foram bem comercializados a partir de 1963).

Ainda com relação à tecnologia, essas máquinas apresentaram novidades na arquitetura dos processadores: registradores de índices para controle de

loops e unidades de ponto flutuante, específicos para cálculos com números com vírgula, propiciando cálculos mais rápidos e precisos.

A-Z

loops
Loop é um termo muito utilizado em informática, principalmente em redes e programação. Designa uma execução de uma tarefa que teoricamente pode nunca ter fim. Pode ser um aliado em processamentos que exigem volumes de dados muito grandes, como cálculos de previsão do tempo.

Figura 2.4: Computador IBM "transistorizado"
Fonte: IBM, Inc. http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_PP7090.html.

c) Terceira geração (1966 a 1980): Circuitos Integrados (CI)

A grande inovação é o uso de circuitos integrados (CI), **semicondutores** com muitos transistores construídos em um só componente. Os circuitos SSI (*Small-Scale Integration* ou integração em pequena escala), que tinham cerca de 10 transistores por circuito (ou *chip*) evoluíram para os MSI (*Medium-Scale Integration* ou integração em média escala) que tinham 100 transistores por circuito. Dessa geração é o microprocessador 4004, da Intel, que equipava os principais supercomputadores.

A-Z

semicondutores
Semicondutores são sólidos cujo material possui propriedades de condução ou isolamento da corrente, dependendo do tratamento químico que é dado a ele.

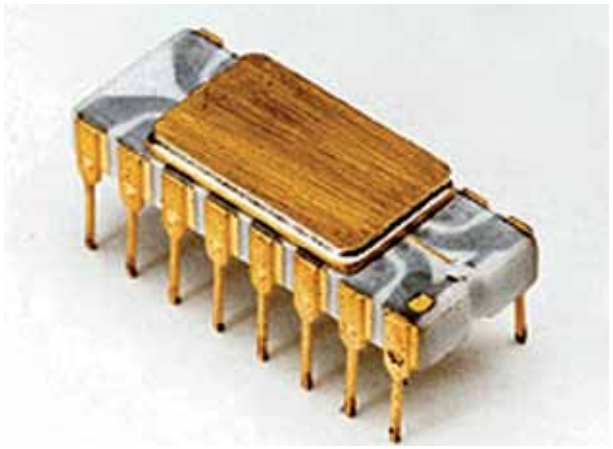


Figura 2.5: Primeiro Microprocessador Intel 4004, lançado em 1971
Fonte: Museu do computador. http://www.museudocomputador.com.br/encipro.php

Os computadores criados nessa época eram classificados como supercomputadores ou minicomputadores:

- Supercomputador, **processamento Vetorial**: IBM 360/91, *Solomon*, da *Westinghouse Corporation*, CDC 7600, STAR-100, TI-ASC.

A-Z

processamento Vetorial
Processamento vetorial é aquele em que o processador acessa uma sequência completa de instruções a serem executadas, como se estivessem empilhadas uma sobre a outra, diminuindo a quantidade de vezes que o processador precisa ir buscar uma instrução na memória.

- Supercomputador, processamento Paralelo: ILLIAC IV, desenvolvido pela Burroughs, para o Departamento de Defesa Americano e a Universidade de Illinois.

- Minicomputadores: série PDP-11, da DEC.

d) Quarta geração (1981 a 1990): A era dos computadores pessoais

Com o desenvolvimento de circuitos integrados em larga escala (*Large Scale* Integration – LSI), que são *chips* contendo milhares de transistores em um centímetro quadrado de silício, surgiu a era dos computadores pessoais. Em termos de arquitetura, os computadores pessoais (inicialmente denominados de microcomputadores) não eram muito diferentes dos minicomputadores da classe PDP-11, mas no preço eram claramente diferentes. Se o minicomputador tornou possível para um departamento, uma empresa ou uma universidade terem seu próprio computador, o *chip* microprocessador tornou possível a um indivíduo qualquer ter o seu computador pessoal, de acordo com Tanembaum (2003).

A tecnologia empregada nessa geração consistia principalmente em:

- uso de circuitos de larga escala (LSI - 1000 transistores por *chip*) e larguíssima escala (VLSI - 100.000 transistores por *chip*) na construção dos processadores,

- memórias com uso de circuitos semicondutores.

Para essa geração, podemos citar os seguintes computadores:

- Processamento Vetorial: CRAY 1, CRAY X-MP, CRAY 2 e CYBER 205, que chegava a executar 800 milhões de operações por segundo, com muita memória para a época e trabalhando com 64 *bits*.

- *Personal Minicomputer*: 1974: *Mark 8*, construído por Johnnatan Titus e vendido por correspondência.

- PC (*Personal Computer*): Microprocessadores 8008 e 8080:

  – 1975 (Janeiro) - *Altair 8800* (construído com o microprocessador Intel 8080);

- 1977 - *Apple II*;
- 1981 - PC IBM.

Nessa geração também é importante lembrar a invenção dos *floppy disks* (discos flexíveis ou disquetes).

e) Quinta geração (1991 a 2000)

Essa geração é caracterizada pelo início da disseminação dos microcomputadores nas empresas, principalmente pelo avanço das redes LAN (*Local Area Network* – Rede de Área Local) e WAN (*Wide Area Network* – Rede de Área Extensa). Com o avanço na criação de processadores, computadores com alta escala de integração passaram a ser fabricados. Alguns tópicos dessa geração que merecem ser citados:

- tecnologia: circuitos integrados com um milhão de transistores por *chip*;
- memórias semicondutoras tornam-se padrão;
- **processamento em paralelo** generalizado;
- uso de redes de computadores e estações de trabalho.

f) Computadores da Década de 2000

Foi nessa década que os computadores apresentaram uma evolução exponencial: microprocessadores cada vez mais rápidos e altamente integrados com memórias *cache* internas, discos rígidos de maior capacidade, memórias de alta velocidade e capacidade, dentre tantas outras evoluções de periféricos, como impressoras e monitores. Nessa geração, vale citar:

- tecnologia: alta velocidade e processamento paralelo combinado com processamento vetorial;
- crescimento e evolução das redes de computadores;
- aplicação: na área comercial, na ciência e tecnologia e, principalmente, na educação.

A-Z

processamento em paralelo
Processamento paralelo é aquele processamento em que uma tarefa demasiadamente grande, que consome muito tempo de processamento, pode ser dividida em várias pequenas tarefas e ser processada por dois ou mais processadores. Hoje em dia, com os computadores ligados em rede, é possível que esses processadores estejam em computadores separados, já que cada computador tem seu próprio processador.

Nesse momento da história, os computadores ficaram cada vez mais pessoais aqui no Brasil, tendo seu acesso facilitado devido à instalação de indústrias que importavam e montavam muitos desses equipamentos.

Resumo

O processo de evolução dos computadores iniciou-se na década de 40. O destaque é para o matemático John Von Neumann, que criou o conceito de programa armazenado, utilizado até hoje. Outras evoluções estão alinhadas principalmente com a eletrônica, pela substituição das válvulas (que aqueciam e queimavam frequentemente) pelos transistores. Daí por diante, os computadores foram reduzindo de tamanho e aumentado sua capacidade de processamento e armazenamento.

Atividades de aprendizagem

1. Para as alternativas abaixo, assinale V ou F:

a) ( ) Os computadores da primeira geração não consumiam muita energia, pois naquela época não existiam usinas hidrelétricas.

b) ( ) O primeiro microprocessador da Intel foi o 4004.

c) ( ) O uso de CI (Circuitos Integrados) fez os computadores executarem cálculos mais rápidos, porém, consumindo o dobro de energia.

d) ( ) Os computadores pessoais (PC – Personal Computer) surgiram com os microprocessadores 8080 da Intel.

e) ( ) Semicondutores são estruturas que, dependendo da química aplicada a ele, pode se tornar um condutor ou um isolante de corrente.

2. Faça uma pesquisa e defina o funcionamento de um relé e de uma válvula.

3. Os computadores das primeiras gerações eram usados apenas pelos projetistas que os construíram. Explique essa afirmação.

4. Para você, quais foram os principais benefícios observados com a introdução de transistores e circuitos integrados nos computadores? Faça um comparativo com a realidade atual.

Aula 3 – Aritmética computacional e representação de dados

Objetivos

Compreender a linguagem numérica utilizada pelos computadores.

Compreender as formas numéricas que se apresentam num sistema computacional.

Compreender as formas de conversão de bases dos sistemas reconhecidos pelo homem e pela máquina.

Compreender os principais tipos de dados e sua representação interna.

Compreender o formato de apresentação dos dados no âmbito computacional.

3.1 Conceitos básicos: dado, informação e representação

Na natureza, todo tipo de informação pode assumir qualquer valor compreendido em um intervalo de - ∞ a + ∞. Você consegue distinguir, por exemplo, uma cor vermelha que esteja um pouco mais clara de outro tom de vermelho. Esse tipo de informação é informação analógica.

Computadores lidam apenas com informações digitais, não abrindo espaço para **dados** estruturados de forma analógica, que podem assumir valores indefinidos.

A-Z

Dado
Dado é forma isolada de representação de um valor. Informação é agrupamento de dados que fornecem algum tipo de mecanismo de tomada de decisão. Representação é formato da gravação do dado (analógico ou digital).

A lógica digital utiliza apenas valores pré-estabelecidos (0) e (1). A lógica analógica admite a existência de valores infinitos.

3.2 Sistemas de numeração

O sistema de numeração decimal está tão presente no nosso cotidiano que nem notamos sua presença. O kg (quilograma), o M (metro), o Km/h (quilômetro por hora), tudo isso é medido numa forma que compreendemos no modelo decimal. Entretanto, este não é o modelo utilizado pelos computadores, pois as posições de memória basicamente conseguem representar apenas 0 ou 1 (como se estivesse ligado ou desligado). Por isso, existem outras notações, como a binária e a hexadecimal.

Essas notações são obstáculos para os iniciantes. Tornam-se, entretanto, simples depois de recordarmos o sistema de base decimal.

3.2.1 Decimal

Quando falamos do número 123, imaginamos certo número de itens que esse número representa e esquecemos o seu significado matemático. Na realidade 123 representa:

$(1 \times 10^2) + (2 \times 10^1) + (3 \times 10^0)$, ou seja: $100 + 20 + 3 = 123$

Figura 3.1: Representação do número decimal
Fonte: do autor

Observe que cada algarismo é multiplicado por uma potência de 10. Os expoentes de 10 são numerados da direita para a esquerda começando com 0 (zero).

Exemplos:

a) $17 = 1 \times 10^1 + 7 \times 10^0$

b) $4345 = 4 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$

Esses sistemas, decimal, binário e hexadecimal, são chamados de posicionais por isto o algarismo tem um valor diferente de acordo com a posição que ocupa dentro do número.

Veja o exemplo a seguir para o número 111 no sistema decimal:

a) o primeiro 1 vale $100 = 10^2$,

b) o segundo 1 vale $10 = 10^1$,

c) o terceiro 1 vale $1 = 10^0$.

3.2.2 Binário

O sistema binário funciona exatamente da mesma forma. O sistema decimal possui dez dígitos (de 0 a 9); o binário possui apenas dois (0 e 1). Fazemos potência de 10 para calcular o número no sistema decimal, então faremos potência de 2 para o sistema binário.

Por exemplo: o valor binário 11001010 representa o valor decimal 202, assim:

$(1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (0 \times 2^0)$, ou seja: $128 + 64 + 0 + 0 + 8 + 0 + 2 + 0 = 202$

Figura 3.2: Representação do número binário convertido para decimal
Fonte: do autor

3.2.3 Hexadecimal

O sistema hexadecimal representa os números em base 16. É usado na informática, pois os computadores costumam utilizar o *byte* como unidade básica da memória e com um *byte* podemos representar 256 valores possíveis, o que abrange todo alfabeto (maiúsculas e minúsculas), os números e vários caracteres especiais.

Como no sistema decimal dispomos de apenas dez dígitos, devemos incluir seis letras para representar o hexadecimal. O conjunto de algarismos hexadecimais fica, portanto, assim:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Nesse sistema, o **A** vale **10**, o **B** vale **11**, etc, até o **F**, que vale **15**.

Figura 3.3: Representação do número hexadecimal
Fonte: do autor

Cada algarismo é multiplicado por uma potência de 16. Os expoentes de 16 são numerados da direita para a esquerda começando com 0 (zero).

Exemplo: $3E0 = 3 \times 16^2 + 14 \times 16^1 + 0 \times 16^0 = 992$

Na relação entre sistema binário e decimal, cada algarismo hexadecimal corresponde a um número binário de 4 *bits*. Por exemplo, no hexadecimal acima temos:

3 = 0011 = 3 decimal

E = 1110 = 14 decimal

0 = 0000 = 0 decimal

Para as questões abaixo, marque um X nas alternativas corretas.

a) O sistema decimal é bem simples para nossa linguagem e por isso é um sistema numérico adequado aos computadores.

b) O sistema binário utiliza como artifício a simbologia do tipo "ligado" e "desligado", facilitando a lógica computacional.

c) O sistema decimal utiliza como potência-base o número 10.

d) O sistema hexadecimal utiliza como base 216, caracterizando 256 *bytes* possíveis.

e) O sistema hexadecimal vai de {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.

3.3 Conversão de bases

A conversão de bases é um processo que, à primeira instância, pode parecer complexo. No item anterior você já viu como se converte de binário para decimal e de hexa para decimal. Como você deve ter observado, entre hexadecimal e binário as conversões também são bastante simples: basta tratar os números como sequências de 4 *bits*. Nas conversões de decimal para binário e de decimal para hexadecimal usamos um processo diferente, porém os dois são semelhantes, como você verá a seguir.

3.3.1 Conversão de decimal para binário

Dividimos o número decimal seguidamente por 2 e utilizamos o resto de cada divisão para compor o número binário. Observe o exemplo a seguir.

Converter 45 (decimal) para binário:

45 / 2 = 22	sobra 1
22 / 2 = 11	sobra 0
11 / 2 = 5	sobra 1
5 / 2 = 2	sobra 1
2 / 2 = 1	sobra 0
1 / 2 = 0	sobra 1

Figura 3.4: Conversão de decimal para binário
Fonte: do autor

Como não dá mais para dividir, paramos. Pegamos todos os restos olhando de baixo para cima. Concluímos que o número 45 em decimal equivale 101101 em binário.

3.3.2 Conversão de decimal para hexadecimal

Por um processo parecido com o que já vimos, dividimos o número decimal seguidamente por 16 e utilizamos o resto de cada divisão para compor o hexadecimal. A única diferença é que o divisor é o 16. Veja o exemplo a seguir.

Converter 970 (decimal) para hexadecimal:

970 / 16 = 60,	resto 10 = A
60 / 16 = 3,	resto 12 = C
3 / 16 = 0,	resto 3 = 3

Figura 3.5: Conversão de decimal para hexadecimal
Fonte: do autor

Temos então que o número 970 (decimal) representa 3CA em hexadecimal (novamente pegamos os restos das divisões, de baixo para cima).

A simbologia hexadecimal não nos permite escrever o número como sendo 31210 ou 3.12.10. Então, substituímos 12 e 10 por C e A, respectivamente. Substituindo os valores de acordo com a Figura 3.5, temos o resultado 3CA.

3.3.3 Conversão de binário para decimal

Esse processo já foi apresentado na seção 3.2.2. Você pode montar uma tabela com as seguintes linhas:

- na primeira: escreva o número binário;
- na segunda: escreva os expoentes de 2, da direita para a esquerda, começando com 0;
- na terceira: calcule as potências de 2;
- na quarta: multiplique a primeira linha pela terceira.

Some a quarta linha e terá o número binário convertido para decimal.

Veja o exemplo de conversão do número binário 101101 observando a Tabela 3.1 a seguir.

Tabela 3.1: Exemplificação de conversão de números binários para decimais

número em binário	1	0	1	1	0	1
expoentes	2^5	2^4	2^3	2^2	2^1	2^0
potências	32	16	8	4	2	1
primeira x terceira	32	0	8	4	0	1

A soma dos resultados da quarta linha é 32 + 8 + 4 + 1 = 45.

3.3.4 Conversão de hexadecimal para decimal

Esse processo também foi apresentado na seção 3.2.3. Assim, você pode usar a mesma tabelinha, trocando 2 por 16. Vamos converter o hexadecimal 3F8 para decimal (Tabela 3.2) a seguir.

Tabela 3.2: Exemplificação de conversão de números hexadecimais para decimais			
número em hexadecimal	3	F	8
expoentes	16^2	16^1	16^0
potências	256	16	1
primeira x terceira	3 x 256	15 x 16	8

A soma dos resultados da quarta linha é o número decimal 1016. Você pode conferir os resultados de todos os itens anteriores utilizando a calculadora científica do *Windows.*

Converta os sistemas de bases de acordo com o que se pede:

a) 117 (decimal) para binário

b) 4E9 (hexadecimal) para decimal

c) 100111 (binário) para decimal

d) 986 (decimal) para hexadecimal

3.4 Formatos de representação de dados

A linguagem humana utiliza palavra que contem um número variável de caracteres. Por exemplo, numa conta com lápis e papel, o único limite para o término de uma conta (por exemplo, uma divisão que resulta uma dízima periódica) é o tamanho da folha (embora possamos emendar as folhas e continuar o cálculo). Em computação precisa-se ter atenção aos limites impostos pelas quantidades de *bits* representados num cálculo ou comparação (ações executadas pela ULA) e também os seus tipos.

Segundo Monteiro (2007) um dos aspectos mais importantes do processo de representação dos dados em um computador se refere à quantidade de algarismos que cada dado deve possuir na sua representação interna, que afetará a capacidade de inúmero componentes do sistema. Como exemplo, podemos citar um projetista de processador, que poderá definir o elemento de números inteiros de 32 algarismos (usualmente chamado de 32 *bits*). Na próxima seção apresentaremos os principais formatos de dados.

3.4.1 Tipo caractere

A representação interna de informação em um computador é realizada pela especificação de uma correspondência entre o símbolo da informação e um

grupo de algarismos binários (*bits*). Isso porque o computador, possuindo somente dois símbolos (0 ou 1) para representação, requer mais de um *bit* para identificar todos os possíveis símbolos que constituem as informações usadas pelo homem e que precisam ser armazenadas e processadas na máquina, conforme afirma Monteiro (2007).

Para podermos representar todos os caracteres alfabéticos e decimais é utilizado um método denominado de codificação. Nesse modelo, cada símbolo (caractere) da nossa linguagem é atribuído a um conjunto de *bits* que o identifica de forma única.

Quadro 3.1: Tipos de codificação e suas descrições

Codificação	Descrição
BCD – *Binary Coded Decimal*	É um grupo de 6 *bits*/caractere, o que permite a codificação de 64 caracteres (26). Praticamente não mais utilizado.
EBCDIC – *Extended Binary Coded Decimal Interchange Code*	Exclusivo da IBM, com agrupamentos de 8 *bits*, permitindo a codificação de 256 símbolos diferentes.
ASCII – *American Standard Code for Information Interchange*	Usado pelos demais fabricantes, há uma versão com 8 *bits*, desenvolvida para aplicações com os microcomputadores PC e compatíveis.
UNICODE	Um código de 16 *bits*, que pode representar 65.536 símbolos diferentes. Isso praticamente resolve os problemas de suficiência de códigos aos símbolos, já que podem atender de forma universal a todas as linguagens (grego, hebraico, chinês, japonês, francês, inglês, espanhol, símbolos da União Européia, etc.). Há um consórcio de empresas que escreve as revisões, que pode ser obtida no *site* da *Web* (www.unicode.org).

Faça uma busca nas referências e nos *sites* especializados e obtenha as seguintes informações:

Caractere "@" em binário (na Tabela ASCII);

Caractere ">" em binário (na Tabela ASCII);

* Dica: Procure nos apêndices dos principais livros da área de organização estruturada de computadores. Converta o valor em binário caso seja informado o valor em decimal ou hexadecimal.

3.4.2 Tipo lógico

O foco dessa seção é mostrar como os dados são representados e manipulados na CPU. O tipo de dados lógico permite utilizar variáveis que possuem apenas dois valores para representação, ou seja, FALSO (*bit* 0) ou VERDADEIRO (*bit* 1). Para a manipulação dos valores lógicos são utilizados os operadores, descritos a seguir.

a) Operador lógico AND (E): o resultado da operação será VERDADE se, e somente se, todas as variáveis de entrada forem VERDADE, ou seja, 1.

Tabela 3.3: Resultados da operação lógica AND

A	B	X = A * B
0	0	0
0	1	0
1	0	0
1	1	1

Na Tabela 3.3, o resultado (X) representado pela operação matemática da multiplicação (*) demonstra o funcionamento do operador AND.

b) Operador lógico OR (OU): é definido de modo que o resultado da operação será VERDADE se um operando for VERDADEIRO.

Tabela 3.4: Resultados da operação OR

A	B	X = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Utiliza-se a simbologia (+) da matemática para representar o resultado da operação lógica OR. Obviamente, neste caso, a lógica computacional da soma de 1 + 1 utilizando um operador OR é igual a 1, e não a 2, como na aritmética decimal.

c) Operador lógico NOT: também conhecido como inversor, de modo a produzir na saída um valor oposto ou inverso.

Tabela 3.5: Resultados da operação NOT

A	X = NOT A ou X = $\bar{A}$
0	1
1	0

Esses operadores lógicos são frequentemente vistos em linguagens de programação e em lógicas de programação. Usualmente, as tabelas verdades são utilizadas onde são definidos os operadores lógicos nas linhas de código do compilador utilizado (C++, Pascal, etc.).

Resumo

Nós seres humanos estamos habituados apenas com numeração decimal. Entretanto, esse simbolismo não é facilmente adaptável aos computadores. Dessa forma, a utilização de um sistema binário (0s e 1s) é a solução mais simples e prática para os computadores, já que os circuitos podem trabalhar com operadores lógicos desligado (*bit* 0) e ligado (*bit* 1). Como os computadores trabalham com agrupamentos desses *bits*, para nós, seres humanos, fica mais difícil interpretá-lo quando existem grandes cadeias destes números (ex: 00101001000011101001001001). Dessa forma, outras notações foram criadas (como por exemplo, hexadecimal), tornando mais curta e fácil sua compreensão. Outro aspecto importante refere-se à lógica dos operadores, que são usadas nas linguagens de programação e nos circuitos dos computadores.

Atividades de aprendizagem

Observe as tabelas e calcule os resultados observando seus operadores lógicos, a seguir.

a) Calcule: X = A * B * C

A	B	C	X = A * B * C
1	1	0	
0	1	0	
1	0	0	
1	1	0	

b) Calcule: X = (A + B) * C

A	B	C	X = (A + B) * C
0	0	1	
1	1	0	
0	0	0	
0	1	1	

c) Calcule: X = NOT (A + B + C)

A	B	C	X = NOT (A + B + C)
0	0	0	
1	1	0	
0	0	0	
0	1	1	

Aula 4 – Organização de Computadores

Objetivos

Compreender as operações básicas realizadas pelos computadores.

Conhecer os principais modelos de organização de computadores.

Compreender como estão relacionados os principais componentes dos computadores.

No nível mais alto, um computador é composto pela CPU, memória e dispositivos de E/S. Para executar programas (desempenhar a função básica de um computador) esses componentes precisam estar relacionados e interconectados de alguma maneira.

4.1 Modelo de Von Neumann

Segundo Stallings (2002), todos os projetos de computadores atuais são fundamentados nos conceitos desenvolvidos pelo matemático John Von Neumann, no Instituto de Estudos Avançados de Princeton. Esse projeto, conhecido como Arquitetura Von Neumann é baseado em três conceitos básicos:

a) os dados e as instruções são armazenados em uma única memória de leitura e escrita;

b) o conteúdo dessa memória é endereçado pela sua posição, independente do tipo de dados nela contido;

c) a execução de instruções ocorre de modo sequencial, exceto quando o programa altere a instrução para a seguinte.

De forma geral, esse conceito permite não precisar fazer modificações no *hardware* para cada aplicação que se deseja executar, já que ele usa o conceito de programa armazenado. Com a arquitetura de Von Neumann pode-se definir um código para cada propósito e utilizar o *hardware* de propósito

geral capaz de interpretar esses códigos. Assim, utiliza-se o método de programação, que é uma sequência de códigos e instruções, chamada, como você já sabe, de *software*.

A máquina de Von Neumann é uma máquina de utilização genérica, isto é, permite resolver qualquer problema que se possa exprimir sob a forma de um algoritmo. Permite-nos especificar programas que indicam as desejadas sequências de operações que se devem realizar sobre a informação representada em binário.

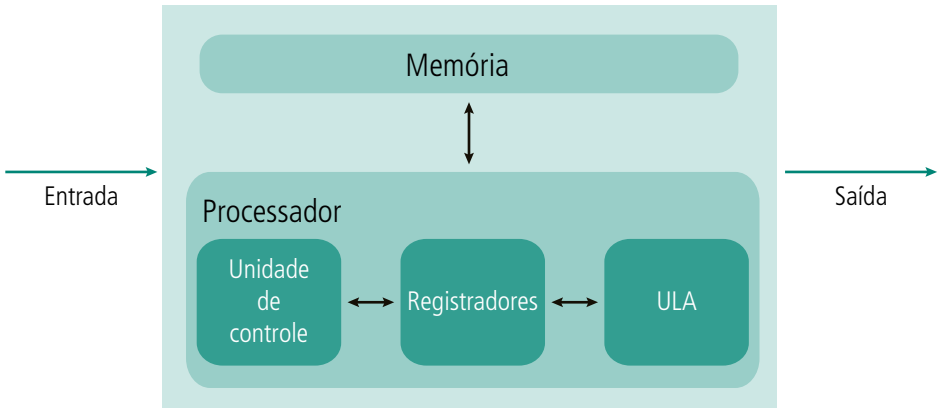


Figura 4.1: Modelo de Von Neumann
Fonte: do autor

No exemplo da Figura 4.1 temos um modelo expandido da visão de Von Neumann:

a) o processador desdobra-se em: Unidade de Controle (UC), necessária ao sincronismo da tarefa; ULA (Unidade Lógica e Aritmética), responsável pelos cálculos e os registradores, para armazenar temporariamente os resultados;

b) a memória, responsável por armazenar o resultado final da operação;

c) dispositivos de entrada e saída para a comunicação do sistema com o meio externo.

Os componentes do modelo são interligados por meio de uma arquitetura de barramentos.

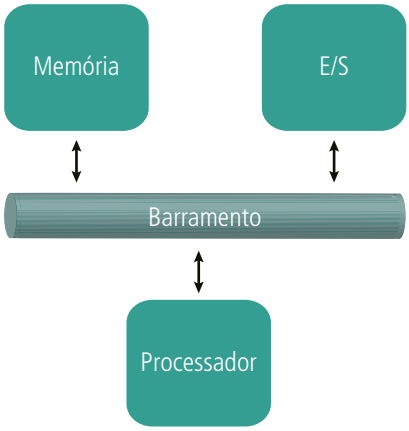


Figura 4.2: Ilustração simplificada do barramento do sistema
Fonte: do autor

O barramento da Figura 4.2 é uma estrutura interna que deve operar a velocidades altas e interligar os principais componentes do modelo de Von Neumann. Veremos mais sobre barramentos nas próximas seções.

4.2 Relação entre os componentes e suas funções

Os componentes que constituem a CPU possuem os módulos que interpretam os códigos (*software*) e os que executam as funções lógicas e aritméticas de propósito gerais. Vários outros componentes são necessários para que um computador possa funcionar. Os dados e instruções devem ser inseridos de alguma forma no sistema. Para isso, é necessário um módulo de entrada de dados. Da mesma forma, o sistema precisa ser capaz de mostrar os resultados. Esses módulos são chamados de componentes de E/S.

Esses módulos contêm componentes básicos que recebem dados e instruções em algum formato e os converte em uma representação interna, inteligível ao computador, ou externa, inteligível ao usuário.

Outro componente importante no modelo de Neumann é a memória. Observe:

a) os dispositivos de entrada fornecem instruções e dados de forma sequencial;

b) nem sempre um programa é executado de forma sequencial;

c) uma operação (aritmética ou lógica) pode precisar de mais de um dado de cada vez;

d) deve haver algum local para armazenar instruções e dados temporariamente. É o que chamamos de memória ou memória principal.

A interação entre os componentes podem ser exemplificado e simplificado da seguinte forma (observando a Figura 4.1):

a) a CPU troca dados com a memória;

b) usa tipicamente dois registradores: um registrador de endereçamen¬to à memória (que contém o endereço da memória a ser usada para a próxima instrução de leitura ou escrita) e outro registrador temporá¬rio de dados (que recebe um dado lido da memória ou armazena um valor a ser gravado na memória);

c) utilizando a mesma ideia, um registrador de endereçamento de E/S especifica um determinado dispositivo de E/S, e um registrador temporário de dados de E/S é usado para troca de dados entre um módulo de E/S e a CPU.

Faça o relacionamento a seguir:

1. Registrador de endereçamento de memória.

2. Processador.

3. Local onde se armazena dados de leitura e escrita.

4. Arquitetura que utiliza o conceito de programa armazenado.

5. É utilizado para interligar os componentes do modelo de Neumann.

6. Registrador de endereço de E/S.

7. Registrador temporário de dados de E/S.

( ) Modelo de Von Neumann.

( ) Utiliza, basicamente, dois registradores para trocar dados com a memória.

( ) Contém o endereço a ser usado para a próxima instrução de leitura e escrita.

( ) Memória principal.

( ) Utilizado para trocar dados entre o módulo de E/S e a CPU.

( ) Barramento.

( ) Especifica um determinado dispositivo de E/S.

4.3 Funções dos computadores

A função básica desempenhada por um computador é executar um programa, que é constituído por um conjunto de instruções armazenadas na memória. O processador realiza o trabalho efetivo de executar as instruções especificadas no programa, conforme Stallings (2002).

Na forma mais simples, existem dois passos para o processamento de instruções:

a) o processador lê instruções na memória, uma de cada vez (ciclo de busca);

b) o processador executa cada uma, podendo envolver diversas operações (ciclo de execução). Todo o processamento necessário para o ciclo de execução é denominado de ciclo de instrução.

4.3.1 Busca e execução

No início de cada ciclo de instrução, o processador busca uma instrução na memória. Existe um registrador chamado contador de programa (*program counter* – PC), que é usado para guardar o endereço da próxima instrução a ser buscada na memória.

O processador incrementa o contador de programa depois de cada busca de instrução, normalmente, mas nem sempre, de modo sequencial.

Conforme foi dito, após o processador buscar uma instrução na memória, ele incrementa o contador de programa, incluindo o endereço de memória seguinte. Entretanto, isso nem sempre acontece de forma sequencial. A instrução buscada na memória é carregada num registrador conhecido como registrador de instruções (*instruction register* – IR). Esse registrador armazena *bits* que especifica qual será a ação do processador. Segundo Stallings (2002), essas ações podem ser classificadas em quatro categorias:

a) Processador-memória: ação de transferência de dados do processador para a memória e vice-versa;

b) Processador-ES: ação de transferência de dados entre processador e um dispositivo de E/S, através de um módulo de E/S;

c) Processamento de dados: execução de operações lógicas ou aritméticas com os dados;

d) Controle: determinadas instruções podem dizer que a sequência de execução das instruções seja alterada, alterando a sequência do contador de programa. Por exemplo:

1. o processador pode buscar uma instrução na posição de memória de endereço 149;

2. a instrução contida nesse endereço 149 especifica que a próxima instrução a ser executada é aquela contida na posição de memória 182;

3. a instrução contida no endereço 182 consiste em armazenar o endereço 182 no contador de programa;

4. no próximo ciclo de busca, a instrução será obtida no endereço 182, e não no endereço 150.

4.3.2 Interrupções

Quase todos os computadores possuem algum mecanismo pelo qual componentes, como E/S e memória, podem interromper a sequência normal de execução de instruções do processador. O mecanismo de interrupções visa melhorar a eficiência de processamento. Segundo Monteiro (2007), uma interrupção consiste em uma série de procedimentos que suspendem o funcionamento corrente do processador, desviando sua atenção para outra atividade.

A interrupção é uma alternativa válida para evitar o desperdício de tempo do processador, atendendo a dispositivos bem mais lentos, como uma impressora por exemplo.

Observe a seguir o exemplo de uma interrupção com uma impressora:

a) o processador emite uma instrução de E/S para a interface da impressora (para checar seu *status*). Como não deverá ter uma resposta imediata, ele desvia sua atenção para executar outra atividade, suspendendo a execução daquele programa que requer a impressora (como por exemplo, a impressão de uma página no *Word*), e atendendo a execução de outro programa na fila (a execução de um áudio ou vídeo, por exemplo);

b) quando a interface de E/S (onde está conectada a impressora) está pronta para enviar os dados do periférico ao processador, ele o avisa por um sinal de interrupção, dizendo seu *status* (pronto, ocupado, com erro, sem papel, etc.);

c) o processador então inicia o programa de E/S (no caso, a impressão de uma página no *Word*), como se fosse o método anterior.

As interrupções podem ser:

- de *software* - gerada por alguma condição que ocorra como resultado da execução de uma instrução, como por exemplo, a divisão de um número por zero;

- de relógio - aquelas geradas pelo próprio processador e permite que o Sistema Operacional execute certas funções a intervalos regulares de tempo;

- de E/S - geradas por um controlador de E/S para sinalizar a conclusão de uma operação ou para sinalizar uma situação de erro. Dessa forma, a interface de E/S "avisa" ao processador que determinado periférico requer atenção para transferir dados (um *pen drive* ou impressora, por exemplo);

- de falha de *hardware* - gerada pela ocorrência de erro de *hardware*, como queda de energia, erros de memória ou mau funcionamento de um componente de *hardware*.

Resumo

Nesta aula, falamos sobre o conceito abordando por Von Neumann, que usa o conceito de programa armazenado. Nesse modelo, há uma memória que armazena um dado ou instrução e que é acessada pelo processador. Os componentes do modelo de Von Neumann (Memória, CPU e E/S) se comunicam por meio de uma estrutura de barramento. Os computadores, na sua essência, têm por função de busca e execução de instruções, ação essa realizada pelo processador. Os processadores precisam atender aos requisitos dos programas e dispositivos ligados a ele, dessa forma, as interrupções comandam o momento certo do uso do processador.

Atividades de aprendizagem

Faça o relacionamento a seguir:

1. Ciclo de instrução.

2. Ciclo de busca e Ciclo de execução.

3. Interrupção de E/S.

4. Interrupções.

5. Contador de programa.

6. Registrador de instruções.

7. Ações de controle.

( ) Gerada por um controlador de E/S para sinalizar a conclusão de uma operação.

( ) Guarda o endereço da próxima instrução a ser buscada na memória.

( ) Armazena uma instrução buscada na memória.

( ) Principais passos para o processamento de instruções.

( ) Podem informar, por exemplo, que a sequência de execução das instruções seja alterada.

( ) Todo processamento necessário para o ciclo de execução.

( ) Alternativa válida para evitar o desperdício de tempo do processador com dispositivos lentos.

Aula 5 – Barramento do sistema

Objetivos

Compreender o modelo de interconexão dos principais componentes.

Conhecer as características básicas dos barramentos.

Compreender as estruturas básicas dos barramentos.

Conforme já vimos no modelo de Von Neumann, um computador consiste num conjunto de módulo com três tipos básicos (processador, memória e E/S), que se comunica entre si. De fato, para que isso ocorra de maneira efetiva, devem existir caminhos de conexão entre esses módulos. Segundo Stallings (2002), a coleção de caminhos que conectam os vários módulos é chamada de estrutura de interconexão.

5.1 Estruturas de interconexão

O modelo da estrutura de interconexão depende das informações que são trocadas entre os vários módulos de processador, memória e E/S.

Observe na Figura 5.1 os principais tipos de troca de informações necessárias, indicando as principais formas de entrada e saída para cada tipo de módulo.

Figura 5.1: Troca de informações em memória
Fonte: Arquitetura e Organização de Computadores: Projeto para o Desempenho. Willian Stallings, 2002 (Adaptação).

Na Figura 5.1, a memória está tipicamente organizada em vários endereços numéricos, na qual cada endereço armazena o que chamamos de palavra, que pode ser lida ou escrita na memória. A posição de memória na qual deve ser efetuada a ação (leitura ou escrita) é especificada por um endereço.

A natureza da operação (operação de leitura ou escrita) é definida por meio de sinais de controle.

Figura 5.2: Troca de informações em módulo de E/S
Fonte: Arquitetura e Organização de Computadores: Projeto para o Desempenho. Willian Stallings, 2002 (Adaptação).

Na Figura 5.2, podemos verificar que, assim como na memória, dois tipos de operações podem ser feitas no módulo de E/S: leitura e escrita. Este módulo pode controlar vários dispositivos externos, e possuem caminhos externos para entrada (leitura) e saída (escrita).

Conforme já foi falado, o módulo de E/S também deve ser capaz de enviar sinais de interrupções ao processador. Isto é especialmente importante quando um dispositivo de E/S precisa enviar algum dado ou estabelecer comunicação com algum outro dispositivo.

Figura 5.3: Troca de informações envolvendo a CPU
Fonte: Arquitetura e Organização de Computadores: Projeto para o Desempenho. Willian Stallings, 2002 (Adaptação).

Na Figura 5.3, podemos observar a mesma estrutura de interconexão presente na CPU. Tipicamente, os barramentos de dados, endereços e controle aparecem com partes fundamentais.

Um sistema de computação típico contém diversos barramentos, fornecendo caminhos de comunicação entre os seus componentes. O barramento usado para conectar os principais componentes do computador (processador, memória, E/S) é chamado de barramento do sistema.

Um barramento consiste em vários caminhos ou linhas de comunicação capaz de transmitir sinais binários (0 ou 1). As diversas linhas de barramentos podem ser usadas para transmitir vários dígitos binários simultaneamente (em paralelo). Por exemplo, uma unidade de dados de 8 bits pode ser transmitida por oito linhas de barramento

5.2 Estruturas de barramentos

Um barramento do sistema contém várias linhas distintas (normalmente entre 50 e 100). De forma geral, as linhas de um barramento podem ser classificadas em três grupos: linhas de dados, linhas de endereço e linhas de controle.

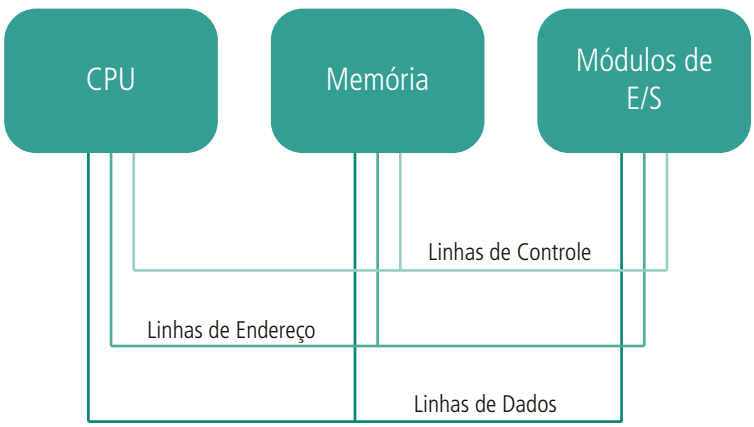


Figura 5.4: Esquema de interconexão de barramento
Fonte: do autor

Na Figura 5.4, as linhas de dados fornecem um caminho de transferência entre os módulos do sistema (CPU, Memória e E/S). Esse conjunto de linhas é chamado de barramento de dados. Esse barramento contém, usualmente, 8, 16 ou 32 linhas (o número de linhas é conhecido como a largura do barramento). Cada linha pode conduzir apenas 1 *bit* por vez.

A largura do barramento de dados constitui um parâmetro fundamental para o desempenho do sistema. Por exemplo, se o barramento de dados tem largura de 16 *bits* e cada instrução tem tamanho de 32 *bits,* o processador precisa acessar duas vezes o módulo de memória em cada ciclo de instrução.

As linhas de endereços são utilizadas para designar a fonte ou o destino dos dados transferidos pelo barramento de dados. Por exemplo, quando o processador precisa ler um dado na memória (palavra de 8, 16 ou 32 *bits)*, ele coloca o endereço da palavra desejada nas linhas de endereço. Ao conjunto dessas linhas damos o nome de barramento de endereço.

A largura do barramento de endereços determina a capacidade máxima da memória do sistema. De forma sintética, podemos dizer que um barramento de endereços de 16 bits consegue endereçar 2^{16} posições de memória.

Por fim, as linhas de controle são usadas para controlar o acesso e utilização das linhas de endereço e de dados. Isto é necessário, pois as linhas de dados e endereço são compartilhadas por todos os componentes. Ao conjunto de linhas de controle damos o nome de barramento de controle.

Os sinais de controle, que viajam pelas linhas de controle, controlam, dentre outras coisas, os sinais de comando, que especificam as operações a serem executadas (como por exemplo, escrita, leitura, requisições e concessões)

Stallings (2002) exemplifica a operação de um barramento:

a) quando um módulo do sistema deseja enviar dados para outro, ele deve:

- obter o controle do barramento;
- transferir os dados por meio do barramento.

b) quando um módulo deseja requisitar dados de outro módulo, ele deve:

- obter o controle do barramento;
- transferir uma requisição para outro módulo por meio das linhas de endereços e de controle apropriadas e esperar que o módulo envie os dados requisitados.

No computador (em forma de gabinetes, placas, etc.), o barramento é formado por um conjunto de fios interconectados. Na prática, esses fios são constituídos pelas trilhas que estão presentes na placa mãe. É muito comum que a estrutura de interligação dos componentes principais do sistema (CPU, Memória, E/S) seja feita de forma modular. Assim, podemos imaginar as estrutura de barramentos presentes na placa mãe como uma haste de metal onde podemos encaixar o processador, a memória e os dispositivos de E/S.

Atividades de aprendizagem:

Complete os espaços em branco nas frases a seguir:

1. Cada endereço de memória armazena o que chamamos de _____, que pode ser lida ou escrita.

2. O barramento de _____ leva sinais que vão determinar qual a posição de memória será utilizada na leitura ou escrita.

3. Os _____ vão determinar qual é a operação a ser realizada (como por exemplo, leitura ou escrita).

4. Para que um dispositivo possa se comunicar com o processador, o módulo de E/S deve gerar um sinal de _____ para avisá-lo do evento.

5. _____ acessos a memória são necessários para ler um dado de 32 *bits* por meio de um barramento de dados 8 *bits*.

6. O _____ é necessário, pois é necessário controlar o uso dos barramentos, já que são compartilhados por todos os outros módulos do sistema.

7. Quando um módulo deseja enviar um dado, ele deve primeiro obter o controle do barramento através de uma _____, por meio de um sinal de controle.

8. Os barramentos apresentados até agora são na prática _____ presentes na placa mãe do computador. Através dos vários soquetes modulares (para encaixes do processador e memória, por exemplo) eles são _____ e trocam dados entre si.

5.3 Múltiplos barramentos

Devido à grande quantidade de dispositivos conectados ao sistema (computador), o uso de um único barramento pode prejudicar o desempenho de dispositivos. Por exemplo, dispositivos lentos "atrasam" o funcionamento dos dispositivos rápidos. Assim, não seria interessante manter, no mesmo barramento, a memória e os dispositivos USB, por exemplo.

Desse modo, grande parte dos sistemas computacionais utiliza mais de um barramento, organizados de forma hierárquica.

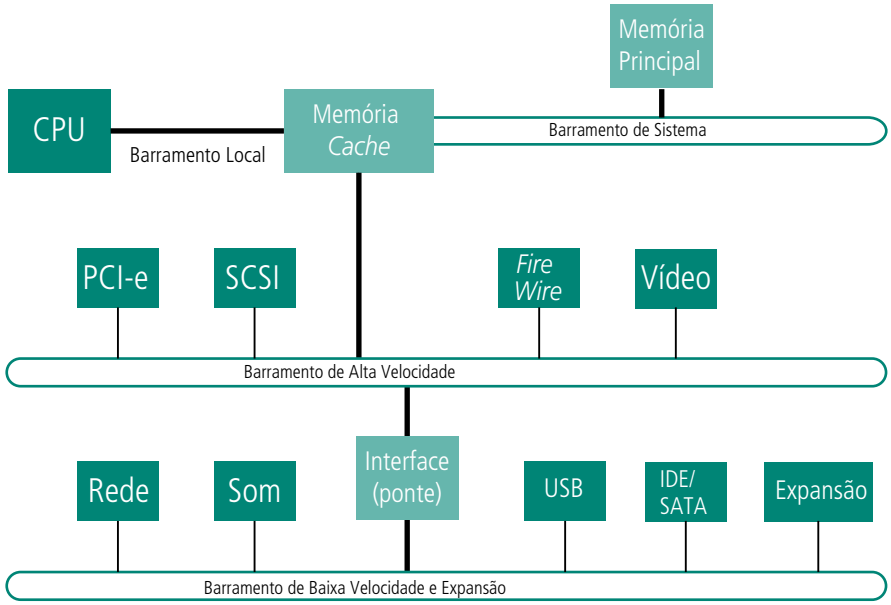


Figura 5.5: Arquitetura hierárquica de alto desempenho
Fonte: do autor

O uso da memória *cache* evita que o processador (CPU) tenha que acessar frequentemente a memória principal. Basicamente, quando o processador precisa fazer uma leitura na memória principal, ele busca primeiramente na memória *cache*. Isso torna o processo mais rápido, pois o acesso a memória *cache* é mais rápido do que o acesso a memória principal. Atualmente, as memórias cache estão embutidas na própria pastilha do processador.

Na Figura 5.5 temos uma arquitetura de barramentos visando o desempenho. Nela, podemos observar:

a) barramento local - de grande velocidade e conecta o processador (CPU) ao controle da memória *cache*, que por sua vez se conecta ao barramento do sistema;

b) barramento de alta velocidade - utilizados para interligar dispositivos de alta velocidade, como as interfaces (placas) de vídeo, os dispositivos SCSI e barramentos de expansão de alta velocidade, como o PCI-e (PCI-*Express*);

c) barramento de baixa velocidade - utilizados para interligar dispositivos de baixa velocidade, como as unidades de leitura e gravação (disquetes, discos rígidos, unidades de CD/DVD, placas de som, placas de rede e os dispositivos USB). Nesse barramento estão as possíveis expansões, em forma do *slot* PCI, conforme citaremos adiante.

5.3.1 Barramento PCI

O barramento PCI (*Peripheral Component Interconnect* – interconexão de componentes periféricos) tem largura de banda suficiente para agregar diversos dispositivos. O PCI entrou no lugar de antigos barramentos, como o MCA, ISA, EISA, VESA (que não discutiremos aqui). No momento em que este texto está sendo escrito, o barramento PCI está lentamente cedendo lugar aos barramentos PCI-e. O PCI pode contar com 32 ou 64 linhas de dados (32 ou 64 bits) funcionando a uma taxa bruta máxima de 528 MB/s (*Me-*

gaBytes por segundo). Esse barramento é tipicamente compartilhado com todos os outros dispositivos que o utiliza, causando uma queda generalizada de desempenho. Outro fator negativo é uso de transmissões paralelas, o que limita fisicamente as evoluções deste barramento, já que necessitariam de mais vias de dados para aumentar as taxas, o que nem sempre é possível.

5.3.2 Barramento PCI - *Express*

Também denominado PCI-e, esse é um barramento que utiliza transmissão serial em seus circuitos, diminuindo possíveis interferências ente vias de dados. Como é um barramento de alta velocidade, está substituindo os barramentos PCI e o barramento AGP.

Uma das características fundamentais do PCI *Express* é que ele é um barramento ponto a ponto, em que cada periférico possui um canal exclusivo de comunicação com o ***chipset***. No PCI tradicional, o barramento é compartilhado por todos os periféricos ligados a ele, o que pode criar gargalos, de acordo com Morimoto (2007).

Ainda segundo Morimoto (2007), existem quatro tipos de slots PCI-*Express*, que vão do 1x ao 16x. O número indica quantas linhas de dados são utilizadas pelo *slot* e, consequentemente, a banda disponível. Cada linha PCI-*Express* utiliza quatro pinos de dados (dois para enviar e dois para receber), que são capazes de transmitir a 250 MB/s em ambas as direções.

A-Z

chipset
O chipset é o componente integrado na placa mãe do computador responsável pela comunicação dos dispositivos até agora mencionados. Podemos dizer então que o chipset, indiretamente, é o barramento.

Temos então 250 MB/s de banda nos *slots* 1x, 1 GB/s nos *slots* 4x, 2 GB/s nos *slots* 8x e 4 GB/s nos *slots* 16x.

Resumo

As estruturas de interconexão dos módulos de Von Neumann chamam-se barramentos. Os barramentos são, basicamente, de três tipos: barramento de dados, de endereços e de controle. O barramento de dados tem grande influência no desempenho do processamento, pois ele pode carregar através das suas linhas, de 32 a 64 *bits* de dados, dependendo da arquitetura do processador. Para evitar gargalos, os barramentos são organizados segundo uma ordem hierárquica. Isso se deve ao fato de que dispositivos mais lentos e dispositivos mais rápidos necessitam se comunicar com o processador.

Atividades de aprendizagem

Responda as questões apresentadas a seguir.

1. Por que foi necessário criar múltiplos barramentos em hierarquia?

2. Na Figura 5.5, verificamos a existência do PCI-e conectado ao barramento de alta velocidade. Atualmente, são muito utilizados para as interfaces de vídeos, no lugar das antigas interfaces de vídeo com barramento AGP. Pesquise sobre estes dois barramentos (PCI-e e AGP) dando ênfase às comparações de suas principais taxas de operações.

3. No texto foi citado o conceito de *chipset*. Faça uma busca nas referências bibliográficas e dê uma definição mais completa sobre esse conceito. Dica: procure citar tipos e marcas.

4. Qual é a largura de banda de um barramento PCI-e 32x ?

5. Qual a largura de banda máxima de um barramento PCI?

6. Cite duas desvantagens do barramento PCI e duas vantagens do barramento PCI-e.

Aula 6 - Unidade Central de Processamento

Objetivos

Examinar a estrutura e o funcionamento interno da CPU.

Abordar questões relativas à arquitetura da CPU.

Tratar aspectos de organização da CPU.

Nas aulas anteriores foram apresentadas visões globais da estrutura sistêmica de um computador e a breve descrição de cada um, incluindo o barramento que é utilizado para interligar os componentes.

Nesta aula será analisado outro componente: o processador ou Unidade Central de Processamento. Como já mencionado, ele é o responsável pelas atividades de processamento do computador. Conforme cita Monteiro (2007), os processadores atuais são fabricados de modo que, num único invólucro (pastilha – *chip*) são inseridos todos os elementos necessários à realização de suas funções. Nosso estudo sobre processador começa com uma visão geral da unidade lógica e aritmética, passando pela unidade de controle e chegando por fim aos registradores.

6.1 Unidade Lógica e Aritmética (ULA)

Assim, como todos os demais elementos de um computador, a ULA é baseada em dispositivos lógicos digitais simples, capazes de armazenar números binários e efetuar operações aritméticas e lógicas.

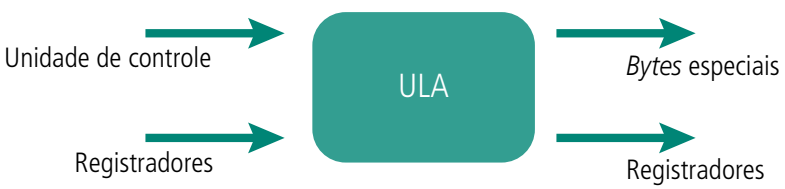


Figura 6.1: Estrutura geral de funcionamento da ULA
Fonte: do autor

Na Figura 6.1, podemos observar a interação e o funcionamento com o restante do sistema computacional. Os dados são fornecidos à ULA através dos registradores. Como já sabemos, os registradores são áreas de armazenamento temporário dentro da CPU, que se conectam a ULA através de caminhos de sinal. Segundo Stallings (2002), a ULA pode também ativar *bits* especiais (*flags*) para indicar o resultado de uma operação. Por exemplo, caso o resultado de uma operação exceda a capacidade de armazenamento de um registrador, isso é indicado atribuindo o valor 1 ao bit de ***overflow***. A unidade de controle (UC) controla a transferência de dados entre a ULA e os registradores.

overflow
O *overflow* pode ser comumente de chamado de estouro de pilha (de empilhamento), estouro de fluxo ou estouro de *buffer*. Isso quer dizer que o fluxo de dados enviados para o local de armazenamento é maior do que sua capacidade suporta.

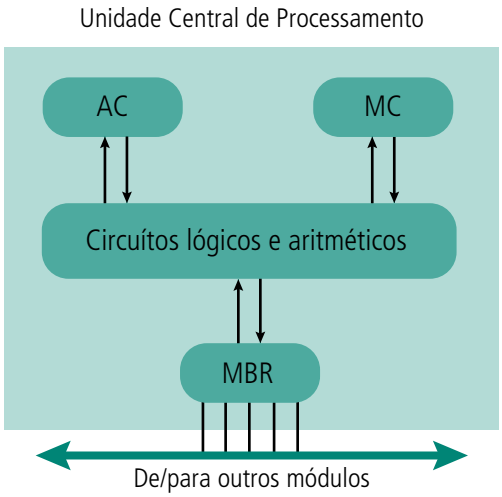


Figura 6.2: Interação dos registradores com a ULA
Fonte: do autor

O registrador AC armazena os bits mais significativos da operação e o MQ armazena os bits menos significativos

Na Figura 6.2 podemos observar que já conhecemos algumas estruturas, como o MBR (*Memory Buffer Register* – Registrador Temporário de Dados), que contém uma palavra com dados a ser armazenada na memória ou é utilizado para receber uma palavra da memória. Já o AC (*Accumulator* – Acumulador) e o MQ (*Multiplier Quotient* – Quociente de Multiplicação) são registradores utilizados para armazenar temporariamente os operandos e os resultados das operações efetuadas na ULA.

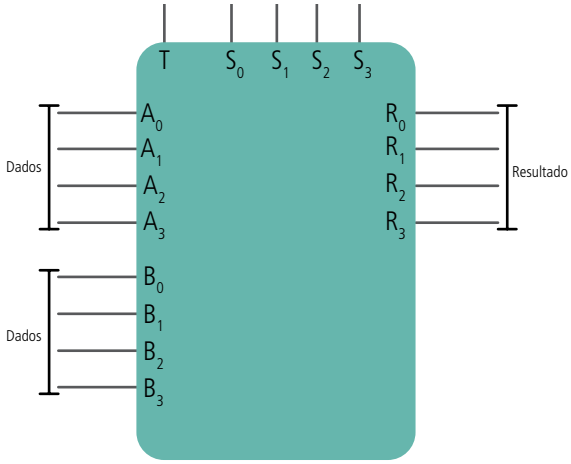


Figura 6.3: Diagrama simplificado dos circuitos lógicos e aritméticos de uma ULA de 4 bits
Fonte: do autor

Na Figura 6.3 temos um diagrama simplificado de uma ULA de 4 *bits* (observe as entrada de dados, que são 4 linhas para cada dado A e B).

Tabela 6.1: Descrição básica de uma ULA de 4 *bits*

PINOS	TIPO	DESCRIÇÃO
$A_0 – A_3$; $B_0 – B_3$	Entrada	Dados
T	Entrada	Pode assumir dois valores: 0 – para operações aritméticas. 1 – para operações lógicas.
$S_0 – S_3$	Entrada	Nesse caso, os sinais de controle poderão indicar 16 operações aritméticas (com T=0) ou 16 operações lógicas (com T=1).
$R_0 – R_3$	Saída	Resultado da operação

Com quatro *bits* (como no exemplo dos sinais de controle), podemos ter dezesseis códigos diferentes (em binário, é claro). Por exemplo: 0000, 0001, 0010, 0100, 1000, 0011... até 1111. É a combinação de dois números (0 e 1) em quatro posições, ou seja, $2^4 = 16$

A Figura 6.3 é uma descrição simplificada de uma ULA de quatro *bits*. Por exemplo, não estão indicados valores como "vai-um" ou "vem-um" das operações aritméticas, bem como as saídas indicando igualdade de comparações de valores. Podemos ainda associar a Figura 6.3 à Figura 6.1, em que o S indica os sinais de controle da unidade de controle e o R indica a saída para os registradores, bem como A e B indicam a entrada dos registradores.

Atividades de aprendizagem:

Associe (V) ou (F) para as seguintes proposições:

a) ( ) *Flag* é uma espécie de bandeira, que sinaliza uma ocorrência.

b) ( ) O registrador MQ é utilizado para armazenar valores de ponto flutuante.

c) ( ) A ULA busca no registrador MBR os dados a serem processados por ela.

d) ( ) Numa ULA típica de 8 *bits*, são possíveis 32 combinações de controle.

e) ( ) As saídas R_0 a R_3 do circuito exemplo da Figura 6.3 são armazenadas no registrador MBR.

f) ( ) O estouro de pilha é uma espécie de sinalização através dos *bits* especiais.

g) ( ) No circuito apresentado na Figura 6.3, quando T = 1, o seguinte cálculo é possível: R = $\bar{A}$ OR B. >

6.2 Unidade de Controle (UC)

A unidade de controle é a parte do processador que controla a execução de instruções. Ela gera sinais de controle externos ao processador para comandar a transferência de dados entre o processador e a memória ou os módulos de E/S. Ela também gera sinais de controle internos ao processador para mover dados entre registradores, para comandar a ULA na execução de uma determinada função e para controlar outras operações internas. As entradas para a unidade de controle consistem do registrador de instrução, bits de condição e sinais de controle gerados por fontes externas, por exemplo, sinais de interrupções, conforme Stallings (2002).

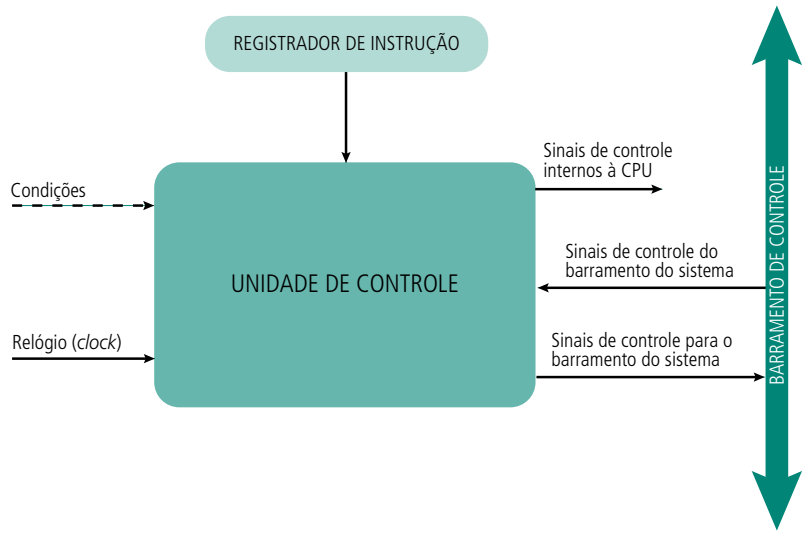


Figura 6.4: Modelo de uma unidade de controle.
Fonte: Arquitetura e Organização de Computadores: Projeto para o Desempenho. Willian Stallings, 2002 (Adaptação).

Observando a Figura 6.4, podemos tratar da interação da unidade de controle e os demais elementos do processador (Quadro 6.1), a seguir.

Quadro 6.1: Descrição das entradas e saídas da UC

SINAL	TIPO	DESCRIÇÃO
Relógio (*clock*)	**Entrada**	**É uma espécie de marcador de tempo. Ele cria a sincronia nas operações do processador (ciclos). Assim como um relógio mecânico faz todos seus movimentos baseados nos segundos, o processador faz seu trabalho baseado em ciclos.**
Registrador de instrução	**Entrada**	**Contém o código da instrução corrente usada para indicar quais as operações o processador deve executar.**
Condições	**Entrada**	**Determina o estado do processador e as saídas de operações executadas pela ULA.**
Sinais de controle do barramento de controle	**Entrada**	**Para fornecer sinais de reconhecimento e interrupção.**
Sinais de controle internos à CPU	**Saída**	**Sinais para causar movimentação de dados de um registrador ao outro e para ativar funções específicas da ULA.**
Sinais de controle para o barramento de controle	**Saída**	**Indica sinais de controle para a memória e sinais de controle para os módulos de E/S.**

Para exemplificar a Figura 6.4 e observar como é feito o controle pela UC, vamos considerar um ciclo de busca (é um bom momento para revisar a seção 4.3):

a) a UC mantém informação sobre o passo do ciclo de instrução que está sendo executado;

b) em um momento, ela saberá que o próximo ciclo a ser executado é um ciclo de busca;

c) o primeiro passo é transferir o conteúdo do registrador PC (*Program Counter*) para o registrador MAR (*Memory Address Register* – Registrador de Endereço de Memória). Isso é feito quando a UC ativa o sinal abrindo as portas lógicas entre PC e MAR;

d) o próximo passo é ler a palavra (dado) da memória para o registrador MBR (*Memory Buffer Register*) e depois incrementar o PC.

Para o passo (d), os seguintes sinais de controle são enviados:

1. um sinal para abrir as portas lógicas, que permite transferir o conteúdo do registrador MAR para o barramento de endereço;

2. um sinal de leitura na memória, colocado no barramento de controle;

3. um sinal para abrir as portas lógicas, que permite que o conteúdo do barramento de dados seja transferido para o registrador MBR;

4. sinal de controle para comandar a operação de somar 1 ao conteúdo do PC, armazenando o resultado no próprio PC (incremento).

Atividades de aprendizagem:

Responda às questões apresentadas a seguir:

1. O que são as portas lógicas mencionadas no texto?

2. Os sinais de interrupção, quando o módulo de E/S precisa ser atendido pelo processador para ler determinado dado de um dispositivo de E/S, é uma entrada ou saída na UC? Justifique sua resposta.

3. O *clock* está diretamente associado à UC. Podemos dizer que seu valor interfere na velocidade do sistema? Explique sua resposta.

4. De acordo com a questão (3), liste pelo menos dois fatores da arquitetura interna do processador que pode interferir na velocidade do sistema.

5. Para armazenar um dado na MBR, a unidade de controle coloca um sinal de leitura no barramento de dados. Qual o erro desta frase?

6.3 Organização de registradores

Programas privilegiados do sistema operacional são aqueles que são executados no modo *kernel*. Normalmente, são programas que fazem parte do núcleo do sistema operacional e precisa ser executado com privilégios.

Veremos mais adiante que um sistema de computação possui uma hierarquia de memória. Nos níveis mais altos dessa hierarquia, a memória é mais rápida, mais cara e menor. Os registradores estão no topo da hierarquia e como já sabemos, está localizado dentro da CPU. Os registradores podem ser:

a) Registradores visíveis para o usuário: permite ao programador fazer acesso.

b) Registradores de controle e de estado: usados pela UC para controlar a operação da CPU e por programas privilegiados do sistema operacional.

Um tipo importante de registradores visíveis ao usuário é o registrador de dados. Como o nome indica, ele não pode conter valores empregados no cálculo de endereços de operandos, apenas de dados.

Para que um dado possa ser transferido para ULA é necessário que ele permaneça armazenado em um registrador. Além disso, o resultado de uma operação feita na ULA precisa ser armazenado, de modo que possa ser reutilizado ou transferido para a memória.

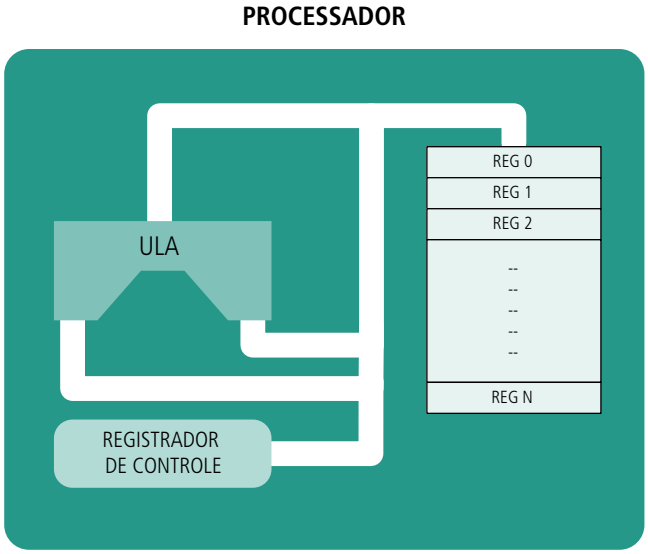


Figura 6.5: Registradores de dados na área funcional do processador
Fonte: Introdução à Organização de Computadores 5ª Ed. Mário A. Monteiro, 2007(Adaptação).

Na Figura 6.5, os registradores de dados estão representados por REG 0 até REG N.

Já os registradores de controle e estado são os vários registradores da CPU que são empregados para controlar a operação da CPU. Quatro destes registradores são essenciais para a execução de instruções:

a) contador de programa (PC) - contém o endereço da instrução a ser buscada;

b) registrador de instrução (IR) - contém a última instrução buscada;

c) registrador de endereçamento à memória (MAR) - contém o endereço de uma posição de memória;

d) registrador de armazenamento temporário de dados (MBR) - contém uma palavra de dados a ser escrita na memória ou a palavra lida mais recentemente.

Uma informação importante sobre os registradores de dados é a sua largura. Isso está diretamente relacionado ao tamanho estabelecido pelo fabricante para a palavra do referido processador. Assim, processadores que tem largura da palavra de 32 bits, devem ter seus registradores também com largura de 32 bits.

O contador de programa (PC) é atualizado pela CPU depois de cada busca de instrução, de modo que ele sempre indique a próxima instrução a ser executada. A instrução buscada é carregada no registrador de instrução (IR).

A troca de dados com a memória é feita usando o MAR e o MBR. O MAR está conectado diretamente ao barramento de endereços e o MBR, ao barramento de dados.

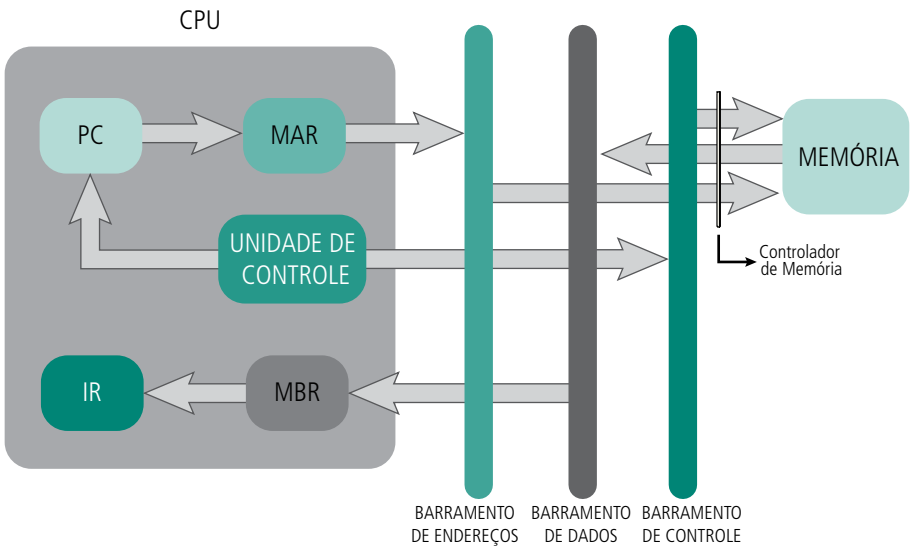


Figura 6.6: Fluxo dos dados usando os registradores
Fonte: Arquitetura e Organização de Computadores: Projeto para o Desempenho. Willian Stallings, 2002 (Adaptação).

Observe a Figura 6.6. O contador de programa (PC) contém o endereço da próxima instrução a ser buscada. Esse endereço é movido para o MAR e colocado no barramento de endereço. A UC requisita uma leitura na memória, e o resultado é colocado no barramento de dados e copiado no MBR e então movido para o IR. Enquanto esse processo acontece, o PC é incrementado em 1 para a próxima busca.

Resumo

O foco desta aula foi a lógica de funcionamento dos processadores. A essência dos processadores está em realizar operações lógicas e aritméticas (ULA), realizar controle dessas operações (UC) e armazenar dados ou resultados temporários (registradores). Para fazer isso, o processador precisa buscar os dados, e, mais importante do que isso, saber onde ele está. Aí entra o barramento de controle, em que a UC do processador controla a transferência do dado, pelo barramento de endereço, já sabendo onde ele está. Assim, finalmente, o dado é transferido para os registradores da CPU usando o barramento de dados. Por fim, a execução é realizada, e o processador parte para a próxima busca e execução.

Atividades de aprendizagem:

Complete os espaços em branco nas frases a seguir:

1. Os registradores_____ são aqueles utilizados internamente pela CPU e tem seu acesso controlado pela UC, que abre as portas lógicas para que seus valores tenham fluxos entre os registradores que são necessitados no ciclo de instrução.

2. Os registradores _____ são utilizados pela ULA para armazenar valores que são escritos na MBR ou serão reutilizados para realizar outros cálculos.

3. O registrador _____ é atualizado após cada busca no ciclo de instrução.

4. O registrador _____ está conectado a memória pelo barramento de _____.

Aula 7 – Memória interna

Objetivos

Abordar as características fundamentais dos sistemas de memórias dos computadores.

Examinar a hierarquia das memórias internas e seus subsistemas.

Apresentar as variedades de tipos, tecnologias, organização, desempenho e custos.

As memórias em um computador estão dispostas de forma hierárquica, assim como acontece nos barramentos. No topo dessa hierarquia estão os registradores do processador, que já vimos. Logo abaixo, estão os níveis das memórias *cache*, variavelmente, divididos em L1, L2, L3 ou mais, dependendo da arquitetura do processador. E por fim, estão as memórias de acesso aleatório, muitas vezes chamada de memória principal. Essas são as memórias internas de um sistema de computação. Nesta aula vamos explorar suas principais características.

Segundo Monteiro (2007), num sistema de computação não é possível construir ou utilizar apenas um tipo de memória, já que ela por si mesma já se constitui de um subsistema, sendo constituída por vários tipos de componentes.

7.1 Memória principal

Conforme vimos, uma das principais características do modelo de Von Neumann é o uso do conceito de programa armazenado. O processador pode acessar imediatamente uma instrução após a outra, pois elas estão armazenadas na memória. A memória especificada para armazenar o programa e os seus dados é chamada de memória principal.

A memória principal é comumente chamada de memória RAM (*Random-Access Memory* – Memória de Acesso Aleatório)

Uma característica importante das memórias RAM (memória de **acesso aleatório**) é a sua volatilidade. Isso quer dizer que ela requer o fornecimento de energia constante para que os dados fiquem armazenados na sua estrutura. Então, dizemos que essa memória é utilizada para armazenamento temporário de dados.

acesso aleatório
O acesso aleatório está diretamente relacionado à posição da memória endereçável. Dessa forma, dizemos que qualquer posição da memória pode ser selecionada, desde que endereçada corretamente.

capacitores
Um capacitor é um componente usado para armazenar cargas elétricas e mantê-las por certo período de tempo (muito curto, daí a necessidade de *refresh*), mesmo que a alimentação seja cortada. Já o *flip-flop* é um dispositivo biestável, ou seja, ele existe em um de dois estados estáveis e são controlados pelo uso de portas lógicas que ativam ou desativam seu estado, podendo funcionar como memória de 1 *bit*.

Existem duas tecnologias de memórias RAM que merecem ser citadas. São elas:

a) Memórias RAM dinâmicas ou DRAM - feitas de células que armazenam dados com carga de **capacitores**. A ausência de carga no capacitor indica um *bit* 0 e a presença de carga, *bit* 1. Um capacitor descarrega-se naturalmente, sendo necessário um circuito especial que faz a regeneração periódica de seu conteúdo (chamado de *refresh*);

b) Memórias RAM estáticas ou SRAM - os valores binários são armazenados usando *flip-flops*, mantendo seus dados enquanto houver fornecimento de energia.

Segundo Monteiro (2007) o controlador de memória tem por função gerar os sinais necessários para controlar o processo de leitura ou escrita, além de interligar a memória aos demais componentes do sistema de computação. Ele é o responsável, dentre outras tarefas, por decodificar o endereço colocando no barramento de endereços, localizando a célula desejada e liberando os bits para o barramento de dados.

A memória principal possui alguns parâmetros, com as seguintes características:

Quadro 7.1: Parâmetros da memória principal

Parâmetros	Características
Tempo de Acesso	É o tempo gasto no ciclo de memória, ou seja, o tempo gasto nas operações de escrita ou leitura, como por exemplo, 50 ns (nanossegundos).
Capacidade	Limitado pelo projeto do processador ou pelo controlador de memória (vide Figura 6.6) embutida no *chipset*. Os processadores de 32 *bits* podem endereçar até 4GB de memória e os processadores de 64 até 16EB (*exabytes*) de memória.
Volatilidade	Os semicondutores que constituem as pastilhas de memória são construídos com capacitores, que é volátil, ou seja, necessita de alimentação elétrica para manter seus valores armazenados.
Custo	As memórias dinâmicas usadas como memória principal têm custos mais baixos do que as memórias *cache*. Dessa forma, os computadores podem ser vendidos com quantidade apreciável de memória RAM.

A memória principal que está sendo apresentada é aquela que aparece em várias figuras mostradas no texto, dentre elas a Figura 6.6.

7.1.1 Memória ROM

Com relação à volatilidade apresentada no texto, há uma pequena porção da memória principal não volátil chamada de ROM, onde são armazenadas pequenas instruções executadas quando o computador é ligado.

Segundo Stallings (2002), em contraste com a memória RAM, existe a memória apenas para leitura (ROM – *Read Only Memory*). Esta memória contém um padrão permanente de dados que não pode ser alterado. Uma ROM é fabricada como qualquer outra pastilha, cujos dados são gravados durante

o processo de fabricação. Esse processo é relativamente ruim, pois qualquer erro pode fazer com que o lote gravado seja inutilizado. As memórias ROMs evoluíram no sentido de permitir alterações pelas gravações por processos especiais. Assim, surgiram os vários tipos de ROMs, que são:

a) PROM (ROM Programável) - é uma alternativa mais barata, uma vez que após o processo de fabricação, o fornecedor ou o cliente pode gravar seu conteúdo (apenas uma única vez, pois seu processo é irreversível) por um processo elétrico;

b) EPROM (ROM Programável e Apagável) - nela, os dados podem ser lidos e gravados eletricamente, mas é utilizada uma luz ultravioleta para apagar seu conteúdo e permitir uma nova escrita, contudo, para permitir a regravação do seu conteúdo, todos os dados devem ser apagados antes de iniciar uma nova gravação;

c) EEPROM (ROM Programável e Apagável Eletricamente) - essa é uma memória mais atraente por facilitar as regravações. Quaisquer dados podem ser gravados nessa memória sem que seja necessário apagar todo o seu conteúdo, já que apenas os bytes endereçados são atualizados;

d) Flash-ROM - é um tipo específico de memória EEPROM, em que são reprogramados grandes blocos de dados. Nesse tipo de memória, todo o conteúdo é apagado antes de haver uma nova reprogramação. O nome flash foi imaginado devido à elevada velocidade de apagamento dessas memórias em comparação com as antigas EPROM e EEPROM.

Faça o relacionamento a seguir:

(1) Memória Principal.

(2) Memória (do tipo registradores).

(3) Utiliza o conceito de programa armazenado.

(4) É a memória de acesso aleatório.

(5) É uma característica fundamental da memória RAM.

(6) São memórias que precisam de regeneração periódica, chamada de *refresh*.

(7) SRAM.

Um tipo de aplicação comum das memórias Flash-ROM é o seu uso em computadores, que possuem nessa memória três programas principais: a BIOS (Basic Input Output System – Sistema Básico de Entrada e Saída), o POST (Power On Self Test – Autoteste de ligação) e SETUP (programa de configuração do hardware instalado no computador). Os fabricantes de placas mãe desenvolvem versões de software (popularmente denominada de versões da BIOS) que podem ser atualizadas para corrigir problemas ou implantar novas características de funcionamento de suas placas. O procedimento para atualização da BIOS é descrita por cada fabricante, de modo que um técnico experiente possa realizá-lo. Atualmente, celulares e smartphones também implementam essa tecnologia que permite atualizações de software.

Uma aplicação típica de EEPROM consiste na programação de teclas de um teclado. A função de cada tecla é definida numa tabela e gravada numa EEPROM que forma o circuito dentro do teclado.

(8) Capacitor.

(9) Parâmetro da memória principal limitado pelo controlador de memória e pela arquitetura do processador.

(10) Memória ROM

(11) *Flash*-ROM.

(12) São memórias que podem ser apagadas e regravadas várias vezes por meio de um processo elétrico.

(13) BIOS, POST e SETUP

( ) Modelo de Von Neumann.

( ) É um tipo de memória não-volátil, apenas para leitura.

( ) Armazena cargas elétricas e as mantém por curto período de tempo.

( ) É o conjunto de subsistemas de memória de um computador.

( ) Estão no topo da hierarquia da memória principal.

( ) Capacidade.

( ) Memória RAM do tipo estática.

( ) Volatilidade.

( ) Um tipo de EEPROM que pode ser gravada rapidamente.

( ) DRAM.

( ) RAM.

( ) Programas armazenados numa típica *Flash*-ROM de uma placa mãe de computador.

( ) EEPROM.

7.2 Memória *cache*

Como já vimos, uma memória principal possui característica de possuir grande capacidade, porém seu acesso é relativamente lento. Dessa forma, essa memória é combinada com uma memória *cache*, menor é mais rápida. Assim, na pirâmide da memória, logo abaixo dos registradores, encontram-se as memórias *cache*.

A memória *cache* contém uma cópia de partes da memória principal

O elo entre a memória principal (RAM) e a CPU tem um ponto frágil - o ciclo de instrução é muito mais rápido do que o **ciclo de memória**.

Durante o ciclo de *refresh* nenhum dado pode ser lido ou gravado na memória principal (RAM).

Na busca pela solução do problema do gargalo de congestionamento na comunicação entre o processador e a memória principal, criou-se o mecanismo de memória *cache*, que tem por função acelerar a velocidade de transferência entre CPU e memória principal.

ciclo de memória
O ciclo de memória é o tempo decorrido entre dois ciclos sucessivos de acesso a memória. Assim, o ciclo de memória é o tempo que a memória consome para colocar o conteúdo de uma célula no barramento de dados mais o tempo gasto para fazer um *refresh* de suas células.

Essas memórias também possuem alguns parâmetros, conforme mostrado no Quadro 7.2 a seguir.

Quadro 7.2: Parâmetros da memória cache

Parâmetros	Características
Tempo de Acesso	É o tempo gasto no ciclo de memória, ou seja, o tempo gasto nas operações de escrita ou leitura. No caso de *cache*, varia entre 5 e 20 ns.
Capacidade	Capacidade limitada pelo seu alto custo de fabricação e por não aumentar a eficiên¬cia do sistema na proporção em que sua capacidade aumenta.
Volatilidade	A exemplo dos registradores, as memórias cache são circuitos eletrônicos que precisam de energia elétrica para manter seu funcionamento e armazenamento.
Custo	Alto custo de fabricação, sendo que o valor por *byte* está situado entre o valor dos registradores e o valor da memória principal (RAM), mais barata.
Tecnologia	Em geral, são memórias estáticas, do tipo SRAM.

Quando o processador deseja ler uma palavra da memória, é realizado um teste para verificar se a palavra está na memória cache. Se estiver, ela é fornecida ao processador. Caso contrário, um bloco de dados da memória principal (RAM) é lida para a memória *cache*.

Antigamente, as memórias *cache* estavam localizadas na placa mãe do computador, e era chamada de memória *cache* L2 (*Level* 2 ou Nível 2), com tamanhos variados (entre 128 KB e 512 KB). As memórias *cache* L1 ficam dentro da própria pastilha do processador, com tamanhos variados (entre 8 KB e 64 KB). Atualmente, as memórias *cache* L1 e L2 estão dentro do próprio processador, podendo chegar a 64KB e 12 MB, respectivamente, embora estabelecer em número seja um tanto impreciso.

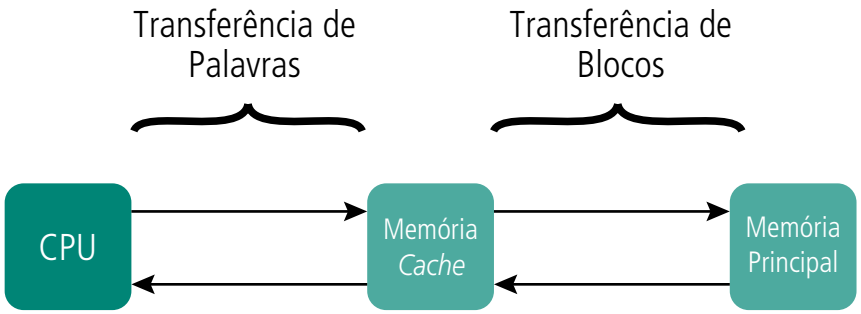


Figura 7.1: Fluxo de dados entre memória *cache* e principal
Fonte: do autor

Para que a memória *cache* funcione no sistema *CPU/CACHE/MP* é necessário que o controlador de *cache* (assim como o controlador de memória) considere a MP (Memória Principal) e X células ou X *bytes*.

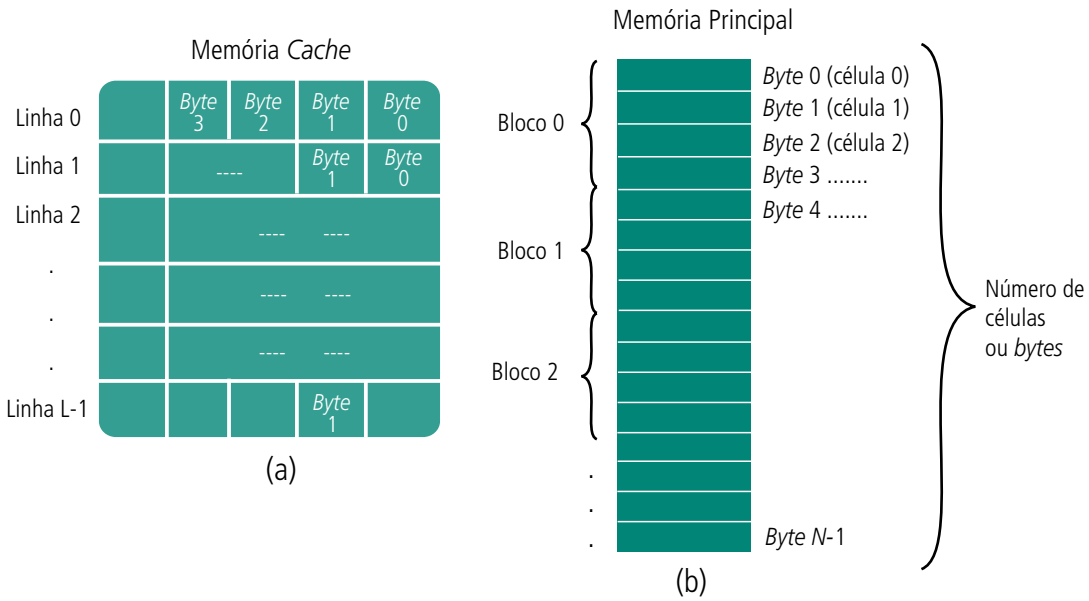


Figura 7.2: Organização da memória *cache* e da memória principal
Fonte: Introdução à Organização de Computadores 5ª Ed. Mário A. Monteiro, 2007(Adaptação).

Na Figura 7.1, quando há uma solicitação de transferência de dados pelo processador e o controlador de memória não o encontra na memória *cache*, ocorre a transferência do dado da memória principal para a *cache*. Entretanto, não apenas o byte requerido é transferido, mas todos os *bytes* subsequentes (o bloco inteiro, vide Figura 7.2 (b)). Dessa forma, pressupõe-se que o processador precisará de outro byte do mesmo bloco, que foi, portanto, carregado para a memória *cache*.

Na Figura 7.2(b), cada bloco possui quatro *bytes* de largura, que deve ser a mesma largura de uma linha de *bytes* da *cache*. A primeira coluna da *cache* (Figura 7.1(a)) é para identificar o número do bloco da memória principal em questão, e é chamada de *Tag*.

7.3 Comparação dos parâmetros das memórias

No texto foi apresentado o estudo sobre os diferentes tipos de memória interna que forma o sistema computacional. Façamos agora um breve comparativo entre suas tecnologias, no que se refere ao tamanho e desempenho.

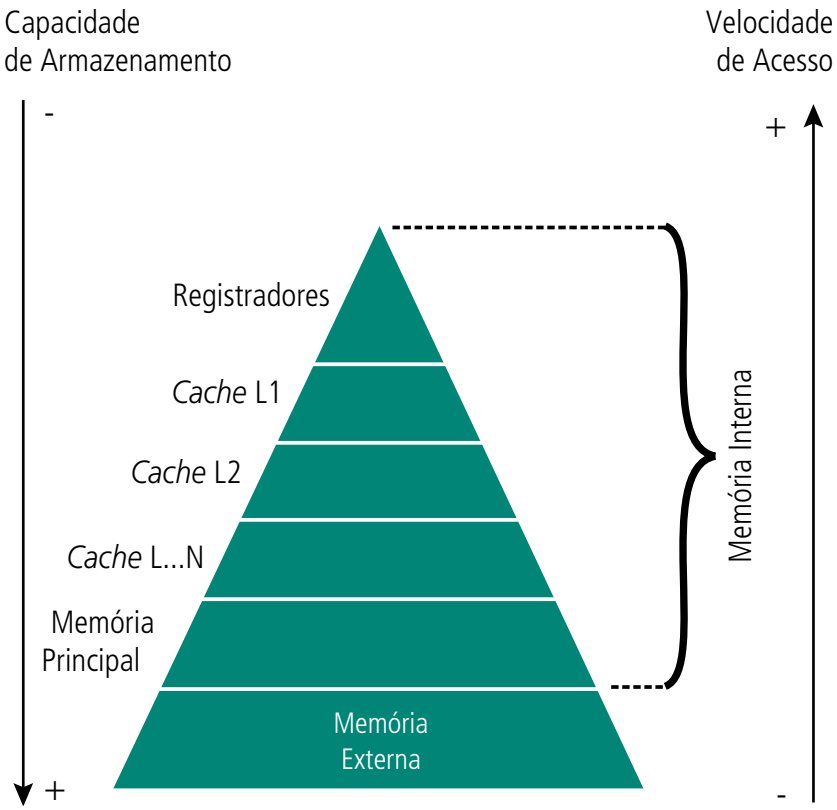


Figura 7.3: Pirâmide de organização dos parâmetros das memórias
Fonte: do autor

Observando a Figura 7.3, podemos concluir o seguinte:

a) os registradores são as memórias mais rápidas, porém de menor capacidade;

b) as memórias *cache* são mais rápidas do que a memória principal, porém menos rápidas do que os registradores. Possuem também alta velocidade de acesso e baixa capacidade de armazenamento nos níveis mais altos da hierarquia;

c) na organização da memória interna, a memória principal tem a maior capacidade de armazenamento e a menor velocidade de acesso;

d) a memória externa (representada pelos discos magnéticos e ópticos – que serão vistos na próxima aula) tem alta capacidade de armazenamento e baixa velocidade de acesso.

Resumo

Esta memória é usada como intermediária para armazenar dados e instruções a serem executadas. Normalmente ela é chamada de Memória Principal. Elas podem ser dividas em categorias. As RAMs são voláteis e perdem seu conteúdo quando o computador é desligado. Normalmente, são de grandes capacidades (DRAM), mas alguns tipos (como SRAM) são de baixa capacidade e constituem a memória *cache* do computador. Outro tipo de memória, as *ROMs* são apenas para leitura, e normalmente armazenam algum tipo de programa que precisa ser sempre executado ao inicializar um dispositivo. Apesar de sem apenas para leituras, novos tipos dessas memórias (EEPROM) podem ter seus programas regravados várias vezes.

Atividades de aprendizagem

Associe (V) ou (F) para as seguintes proposições:

a) ( ) O ciclo de *refresh* aumenta o tempo do ciclo de memória.

b) ( ) A memória principal está organizada em blocos e a *cache* em linhas.

c) ( ) O processador processa inteiramente os blocos da memória principal.

d) ( ) As memórias *cache* L2 são maiores e mais rápidas do que as cache L1.

e) ( ) A memória *cache* é um elo entre a memória principal e o processador.

f) ( ) Processadores que não possuíam memória *cache* eram mais rápidos, pois aproveitava seu espaço para adicionar mais registradores.

g) ( ) As memórias *cache* são, em geral, do tipo SRAM.

h) ( ) O aumento da capacidade de memória cache no processador aumenta seu desempenho e diminui o custo de fabricação.

i) ( ) Se a cada 200 pesquisas para verificar se o dado está na memória *cache* e encontrá-lo nela 190 vezes, a eficiência do uso da memória *cache* está em torno de 95%.

j) ( ) O *Tag* da memória *cache* armazena todos os bytes provenientes dos blocos da memória principal.

k) ( ) A memória *cache* é do tipo não volátil, ou seja, seu conteúdo se perde quando o computador é desligado.

Aula 8 – Memória externa

Objetivos

Abordar a organização e formatação de dados em discos magnéticos e ópticos.

Examinar as características físicas dos discos.

Apresentar os parâmetros de desempenho dos discos.

Os discos magnéticos são ainda os componentes mais importantes da memória externa. Esses tipos de discos são usados desde computadores pessoais simples até supercomputadores. O principal atrativo é a sua ampla capacidade de armazenamento.

Outros dispositivos largamente utilizados como memória de armazenamento secundário e como armazenamento de segurança (*backup*) são os discos ópticos CDs e DVDs.

Nas sessões seguintes serão explorados os conceitos de ambas as tecnologias.

Segundo Monteiro (2007), o primeiro disco rígido surgido foi lançado pela IBM em 1956, que tinha o tamanho de duas geladeiras residenciais e armazenava cerca de 5 milhões de caracteres (aproximadamente 5 MB). Em 1967 houve o lançamento dos discos flexíveis (os disquetes) pela IBM. Atualmente, como já sabemos, os disquetes caíram em desuso.

O armazenamento óptico foi criado pela Philips em 1980, com o objetivo de atender à indústria de áudio. Essa tecnologia teve sucesso, principalmente pela grande capacidade de armazenamento e pelo reduzido tempo de acesso, se comparado as tecnologias de fitas cassetes da época.

8.1 Organização e funcionamento dos discos magnéticos

Um disco rígido (*Hard Disk ou HD*) é formado por uma ou mais superfícies, chamadas de pratos. Essa superfície é circular, fina e coberta por uma camada de material magnetizável. Atualmente, as duas superfícies (as duas faces) do prato são cobertas por este material magnético. Cada face do prato de um disco é organizada em áreas circulares concêntricas, denominadas de trilhas. Em geral, como cada trilha possui grande capacidade de armazenamento em *bytes*, elas são divididas em partes menores, chamadas de setores.

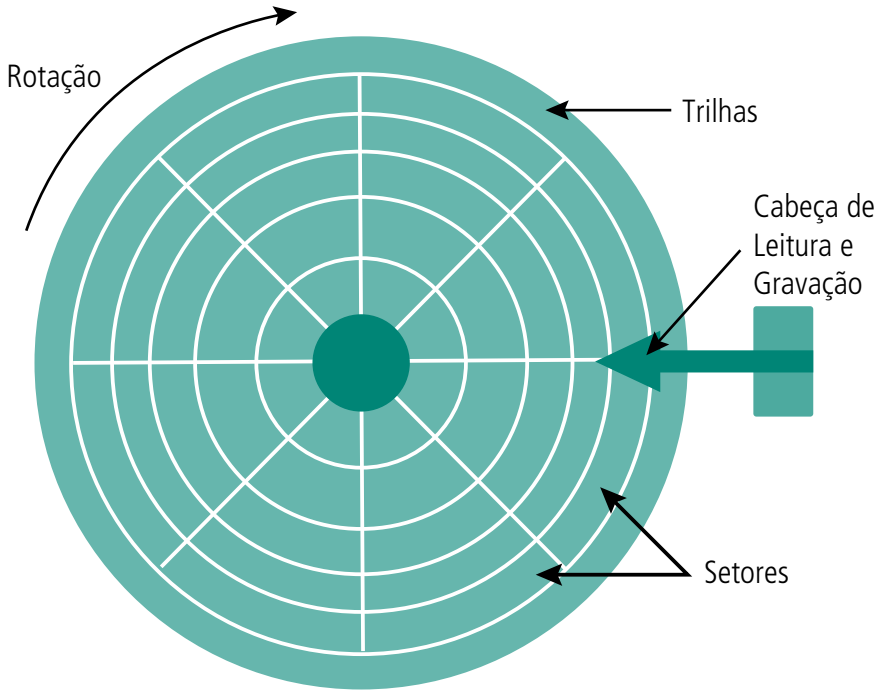


Figura 8.1: Organização de um disco rígido
Fonte: do autor

A junção da divisão tipo pizza com as trilhas formam o que chamados de setores, que possuem 512 bytes cada.

Mesmo que haja diferença no tamanho físico dos setores, eles armazenam sempre a mesma quantidade de informação, ou seja, 512 *bytes*. Isso se deve à densidade magnética de cada setor e pela não existência de trilhas próximas das bordas do prato e nem próximas ao centro.. O disco completo (ou seja, todos os pratos) gira numa velocidade constante em torno do seu eixo, podendo variar de 4.200 RPM a 7.200 RPM nas unidades utilizadas em computadores *desktop*. Em computadores servidores, essa rotação pode chegar a 15.000 RPM.

Quando adquirimos um disco rígido novo ou quando queremos apagar todas as informações de uma só vez, é realizado o processo de formatação, em que são redefinidos todas as trilhas e setores lógicos do disco.

Assim como nas trilhas, os setores do disco são identificados por um endereço, de S0 a S-1, que são definidas todas no momento da formatação do disco.

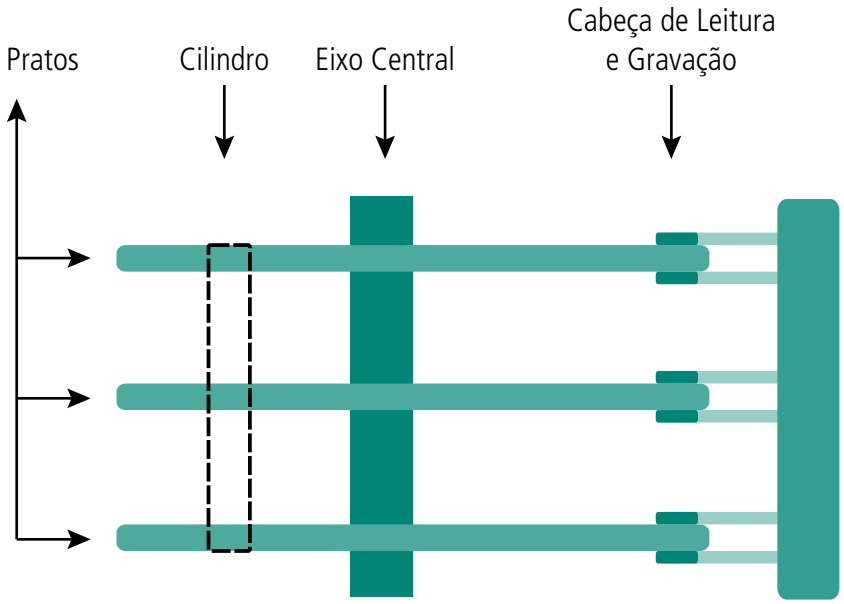


Figura 8.2: Organização de um disco rígido com vários pratos
Fonte: do autor

Observe a Figura 8.2. Sobre cada prato, que tem duas superfícies utilizadas para leitura e gravação, o braço transporta a cabeça de leitura e gravação, realizando um movimento transversal sobre as trilhas. O mecanismo do disco que executa o movimento do braço chama-se atuador.

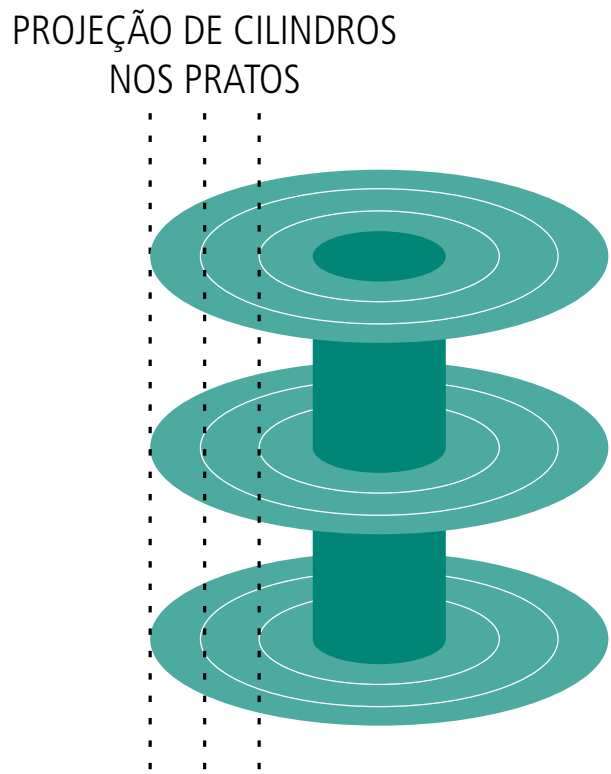


Figura 8.3: Projeção dos cilindros
Fonte: do autor

Observando as Figuras 8.2 e 8.3, podemos verificar a existência de um CILINDRO que é apenas uma projeção das trilhas sobre os pratos, ou seja, ele não existe fisicamente no disco. O que acontece é que, para que o disco faça a leitura ou gravação do dado, é necessário ter um endereço. O que acontece é que, para que o disco faça a leitura ou gravação do dado, é necessário ter um endereço que é composto pelas seguintes informações: o número do cilindro, o número da cabeça (no caso da Figura 8.2, são seis no total) e o número do setor. Esse modelo de endereçamento é, por muitas vezes, chamado de CHS (Cylinder – Head – Sector).

Uma das características do funcionamento dos discos rígidos relaciona-se ao tempo gasto nas etapas para leitura e escrita. O tempo gasto entre o instante de início da leitura ou escrita e seu término é chamado de tempo de acesso, e é formado pelos tempos mostrados no Quadro 8.1.

Quadro 8.1: Etapas do tempo de acesso

Etapa	Descrição
Tempo de interpretação do comando	Tempo gasto para o Sistema Operacional interpretar o comando e passar a ordem para controlador de disco.
Tempo de busca (seek)	Tempo gasto para o disco realizar o movimento do braço para a trilha desejada. É o maior componente do tempo de acesso, ficando em torno de 5 a 10 ms (milissegundos), que são valores típicos.
Tempo de latência	É o tempo gasto pela passagem do setor desejado sobre a cabeça de leitura e gravação. Esse valor depende da velocidade de rotação do disco.
Tempo de transferência	Tempo gasto para passagem dos bits (sinais elétricos) até chegar ao destino.

nostisi
Uma definição muito importante a ser dada refere-se a uma estrutura denominada de cluster. Um cluster (ou agrupamento) é uma unidade de referência de endereço que é utilizada pelo sistema operacional (como Windows, Linux, etc.).

Um dos módulos do sistema operacional chama-se Sistema de Arquivos, que faz o controle sobre as leituras e gravações no disco. Os sistemas de arquivos mais comuns são o FAT32 e NTFS (existem outros como Ext2, Ext3, *RaiserFS*, do *Linux*). O conjunto de vários setores do disco forma os *clusters*, que são referenciados por um único endereço pelo sistema operacional. A quantidade de setores usados para cada *cluster* depende do tipo de sistema de arquivos escolhido. Por exemplo, no sistema NTFS, os *clusters* variam de 512 *bytes* a 4 *Kbytes*. Já no FAT32, variam entre 2 e 32*Kbytes*, sempre dependendo do tamanho do disco.

A descrição dos tipos de sistemas de arquivos e suas características são abordadas normalmente nas disciplinas de sistemas operacionais, em que são fundamentados os conceitos desses tipos de sistemas e sua estrutura de funcionamento.

Atividades de aprendizagem:

Complete as lacunas em branco com o conceito correto:

1. _____ é uma memória externa que está na base da pirâmide, com alta capacidade de armazenamento.

2. O _____ é a superfície circular que forma o disco, contendo uma camada magnética para armazenar os dados.

3. As _____ são as áreas circulares concêntricas que formam os discos.

4. A projeção das trilhas sobre os vários pratos forma o que chamados de _____.

5. O modelo de endereçamento _____ é utilizado pelo disco para localizar os dados requeridos para leitura ou gravação.

6. O _____ é a menor unidade do disco, tendo tamanho de 512 *bytes*.

7. O _____ é o mecanismo do disco que realiza o movimento das cabeças de leitura e gravação.

8. O tempo de _____ é aquele gasto para realizar o movimento do braço para se posicionar no cilindro desejado.

9. _____ é o total de tempo gasto pelo disco rígido desde o momento da requisição do sistema operacional até a entrega do dado.

10. _____ é o agrupamento de vários setores, utilizados pelo sistema operacional.

11. _____ são partes do sistema operacional responsável por cuidar das tarefas pertinentes a organização de leitura e gravação de dados no disco.

12. O tamanho do *cluster* (ou agrupamento) depende da _____ do disco rígido e do _____ utilizado.

8.2 Armazenamento com discos ópticos

A memória óptica é um dos mais bens sucedidos produtos comerciais de todos os tempos. Criado inicialmente para um sistema de áudio digital de disco compacto, o CD é um disco que não pode ser apagado e tem capacidade para armazenar mais de 60 minutos de informação. Com a evolução dos

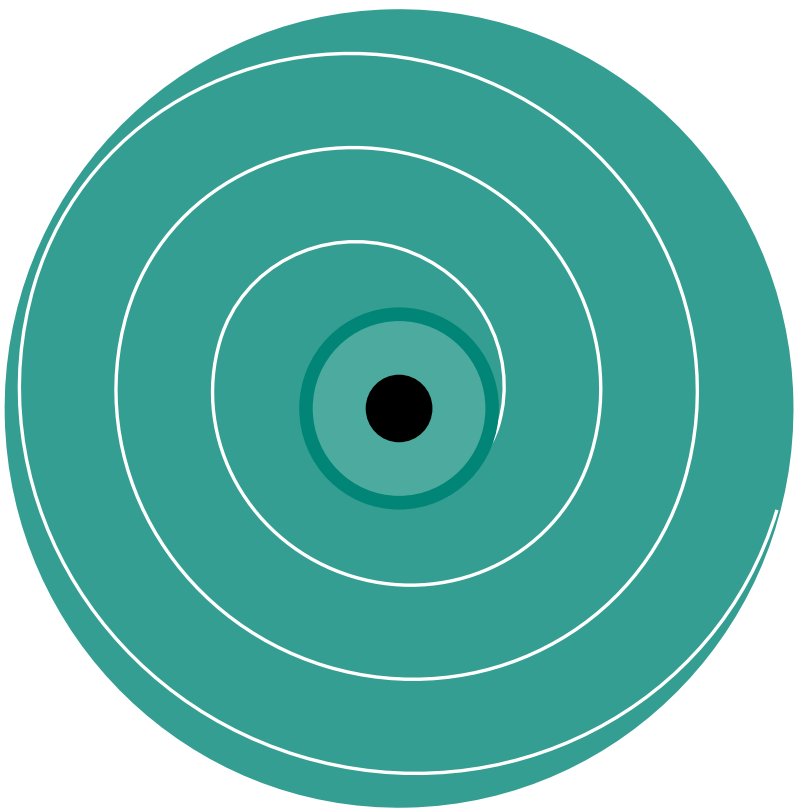
armazenamentos ópticos, como o CD-R e o DVD, uma variedade de novos sistemas foram introduzidos. Devido à alta capacidade de armazenamento (entre 650 e 750 MB) e à alta confiabilidade, surgiram três tipos de CDs no mercado: CD-ROM, CD-R e CD-RW.

8.2.1 CD-ROM *(Compact Disc – Read Only Memory)*

Também existem minidiscos óticos (ou mini CD), com diâmetro de 8 cm.

Foi o primeiro tipo utilizado pela indústria. Utiliza um sistema óptico para marcação dos *bits* (0s e 1s), diferente dos discos rígidos, que utilizam meios magnéticos. Esse disco é composto, em geral, por policarbonato, fabricado sob forma circular e têm espessura de 1,2 mm e diâmetro de 12 cm.

O processo de gravação é feita uma só vez na origem (como por exemplo, uma empresa fonográfica ou um fabricante de *software*), em que são marcados os elementos da informação e logo depois essa matriz é reproduzida em diversos outros discos, que é determinado pelo lote do fabricante. Empresas de *software* (como a *Microsoft*) ou empresas fonográficas (como a *Sony*) utilizam esses tipos de CDs para distribuir seus grandes volumes de programas



ou músicas, por serem mais econômicos.

Figura 8.4: Definição da trilha do CD
Fonte: do autor

Questões como velocidade linear e velocidade angular fazem com que discos rígidos (os pratos), diferentemente dos CDs, desperdicem espaços nas trilhas mais externas. Para informações complementares sobre velocidades lineares e angulares dos discos, consulte Monteiro (2007) e Stallings (2002).

Diferentemente dos discos rígidos, que possuem várias trilhas concêntricas, os CDs possuem apenas uma trilha, criada em espiral a partir de um ponto próximo ao centro do disco (observe a Figura 8.4). Com o propósito de aumentar a capacidade do disco, foi criada apenas uma trilha, onde contém todos os blocos com a mesma capacidade de armazenamento, do início ao final.

O processo usado para gravar dados no disco é, em linhas gerais, simples. Ele ocorre a partir do uso de um feixe de alta intensidade para criar no disco as valas (*pits*), separadas por regiões planas (*lands*).

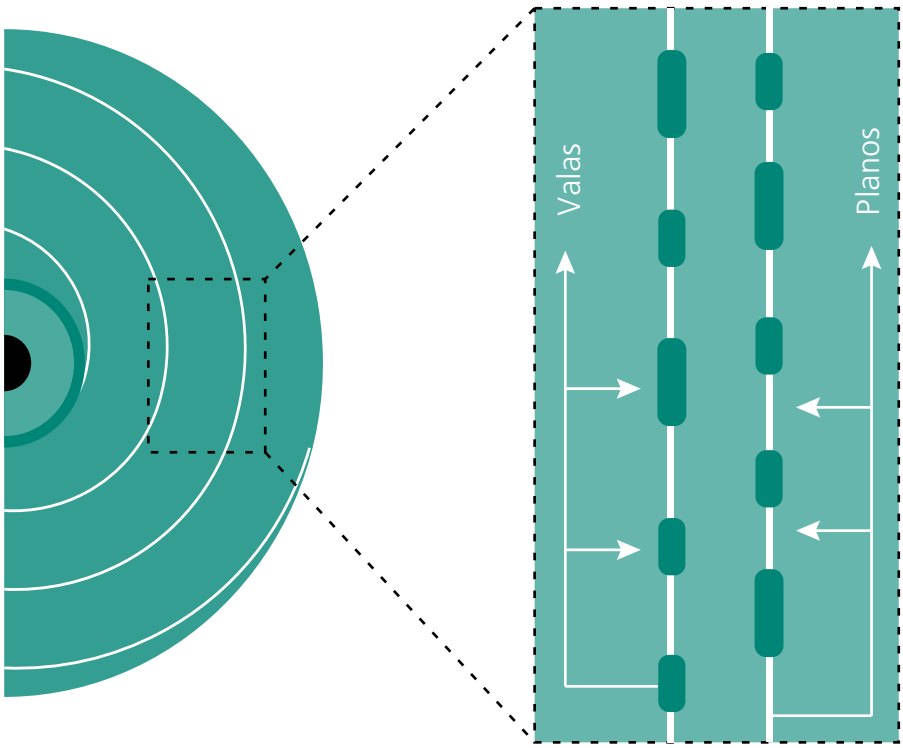


Figura 8.5: Os vales e planos de um CD
Fonte: do autor

O valor do *bit* 1 é definido pela passagem de uma vala para uma superfície plana, e dessa para uma vala. O valor do *bit* 0 é representado pelo espaço entre as valas (observe a Figura 8.5).

8.2.2 CD-R *(Compact Disc Recorded)*

No caso de obtenção de poucas cópias (baixos volumes) de CDs, o processo de gravação de CD-ROMs torna-se caro. A solução encontrada foi fabricar CDs virgens (bem mais baratos) e um gravador especial de CD para uso público. Dessa forma, o processo de gravação passou a ser individual (por CD) e não por matriz e prensagem, como é o caso dos CD-ROMs. Surgiu então um novo tipo de CD, chamado de CD-R.

A-Z

CD-Rs
Os CD-Rs utilizam uma técnica chamada WORM (Write Once Read Many – Grave uma vez, leia várias). Seus gravadores utilizam dois feixes de laser: um de alta intensidade, que é utilizado na gravação e outro de baixa intensidade, que é usado para leitura.

Segundo Monteiro (2007), um CD-R tem característica diferente do CD-ROM com relação à existência de uma camada refletora de alumínio, que é plana e possui uma camada adicional, ativa, constituída de um corante fotossensível.

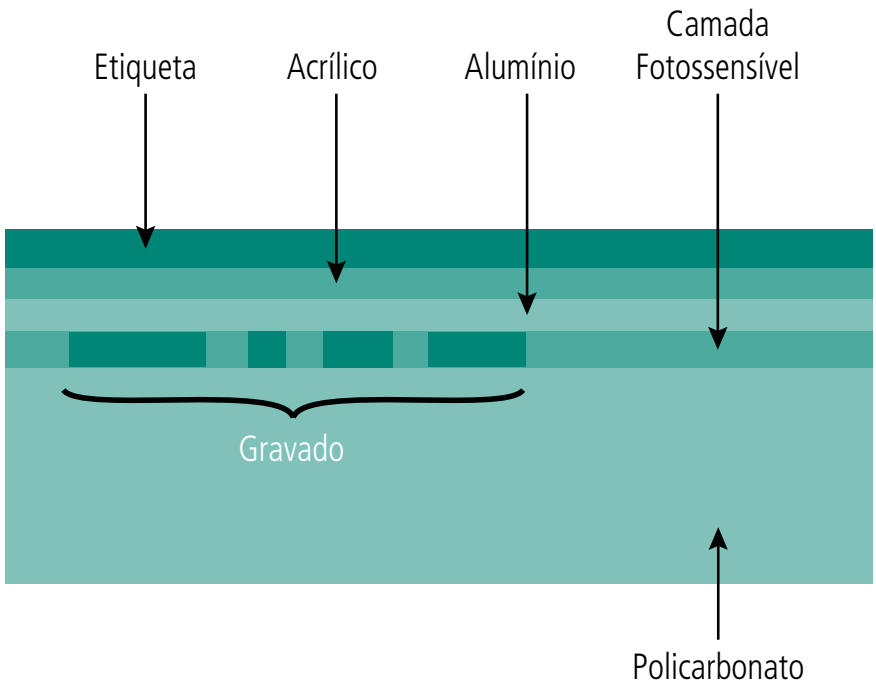


Figura 8.6: Estrutura de uma trilha de um CD-R
Fonte: Introdução à organização de computadores 5ª Ed. Mário A. Monteiro, 2007 (Adaptação)

Durante o processo de gravação (realizado uma única vez), o feixe de laser de alta intensidade percorre a trilha e atinge a camada ativa. Essa camada ativa é translúcida e se torna opaca quando atingida por um feixe de laser de alta intensidade. Dessa forma, podemos associar as regiões opacas aos *pits* (valas) e as regiões translúcidas aos *lands* (planos).

8.2.3 CD-RW *(Compact Disc ReWritable)*

Da mesma forma que as memórias ROMs evoluíram para PROMs e EEPROMs, os CDs também seguiram essa linha. A impossibilidade do CD-R em permitir regravações (como *backups* continuados e progressivos) o tornou pouco atrativo, principalmente para as empresas, a partir daí surgiram os CD-RWs.

Tecnicamente, a diferença básica do CD-RW para o CD-R reside na camada extra, ativa, que no caso do CD-RW, constitui-se por três camadas: duas para proteção, formadas por material dielétrico, e uma ativa, inseridas entre as duas primeiras. As camadas de proteção permanecem sempre num estado translúcido.

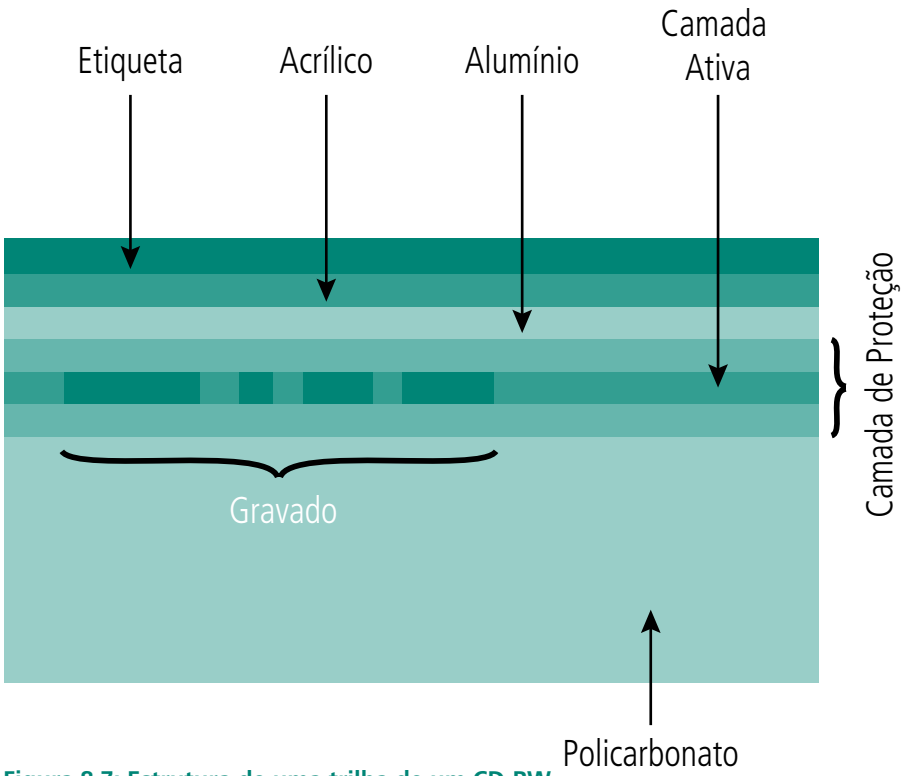


Figura 8.7: Estrutura de uma trilha de um CD-RW
Fonte: do autor

Para Monteiro (2007), o processo de apagamento e gravação de um CD-RW pode ser repetido milhares de vezes, mas aparentemente muito menos do que o apagamento e regravação nos discos rígidos.

O sistema de gravação do CD-RW dispõe de três intensidades de laser. O de baixa intensidade é usado para leitura, o de média intensidade utilizado para o apagamento dos dados na camada ativa, e o de alta intensidade é utilizado para uma nova gravação.

8.2.4 DVD *(Digital Video Disc)*

Os DVDs surgiram para substituir o CD utilizando uma mesma tecnologia óptica, porém com maior capacidade de armazenamento. Ele substituiu as antigas fitas de vídeos VHS analógicas e também está substituindo os CD-ROMs. Sua capacidade de armazenamento pode chegar a 17 GB.

Os DVDs podem ser do tipo comum, tendo uma matriz comum e reproduzidos sob demanda, como filmes, shows, programas de computador, etc. Com isso, ele se compara à tecnologia WORM do CD-ROM, ou seja, grave uma vez, leia várias.

Da mesma forma que existem os CD-R e os CD-RW, existem também os DVD-R e os DVD-RW, utilizando suas gravações de forma semelhante ao CD.

Resumo

As memórias de massa, como também é chamada a memória secundária (ou externa) se constituem em meios de armazenamento bem maiores do que a memória principal. Normalmente, utilizam meios magnéticos ou ópticos para armazenar informação. Os meios magnéticos mais conhecidos são os disquetes (que já caiu em desuso) e os discos rígidos (amplamente utilizados).

É importante mencionar o surgimento recente de novas tecnologias de armazenamento, como o *Blue-Ray,* com capacidade de armazenamento podendo chegar a 50 GB.

Os discos rígidos (ou HD – *Hard Drive*) são constituídos de pratos cobertos por uma camada magnética onde são armazenados os bits. Esses discos possuem cabeças de leituras e gravação com distância de alguns nanômetros do prato (não os tocam quando estão em rotação). Os discos ópticos (CD/DVD) utilizam laser de potências diferentes para leitura e gravação. Desta forma, a trilha que compõe a mídia (CD/DVD) altera seu estado em planos e valas para formar a combinação de 0s e 1s que serão lidos pelo laser da unidade.

Atividades de aprendizagem

Responda as questões a seguir.

1. O CD-ROM utiliza a técnica *WORM*. O que isso quer dizer?

2. Por que para quantidades menores, o CD-ROM é desvantajoso? Qual foi a alternativa usada?

3. Na estrutura da trilha do CD, o que são os pits e os lands? Como eles são formados?

4. Qual a diferença básica entre o CD-R e o CD-RW? Como funciona o *laser* do CD-RW?

5. Faça uma pesquisa e indique as principais características (positivas e negativas) comparando os discos de DVD e *Blue-Ray.*

Aula 9 – Entrada e Saída (E/S)

Objetivos

Conhecer as interfaces de E/S de contato com o mundo externo.

Abordar as principais técnicas de E/S.

Apresentar as principais interfaces de E/S.

Além dos componentes até agora apresentados (como processador e memória), um terceiro elemento é fundamental na composição do sistema: os módulos de E/S. Cada módulo se conecta ao barramento do sistema, podendo controlar um ou mais dispositivos periféricos. O módulo de E/S vai além de ser apenas um conjunto de conectores e cabos que ligam o dispositivo ao barramento, ele possui lógica associada para exercer o controle sobre os dispositivos periféricos.

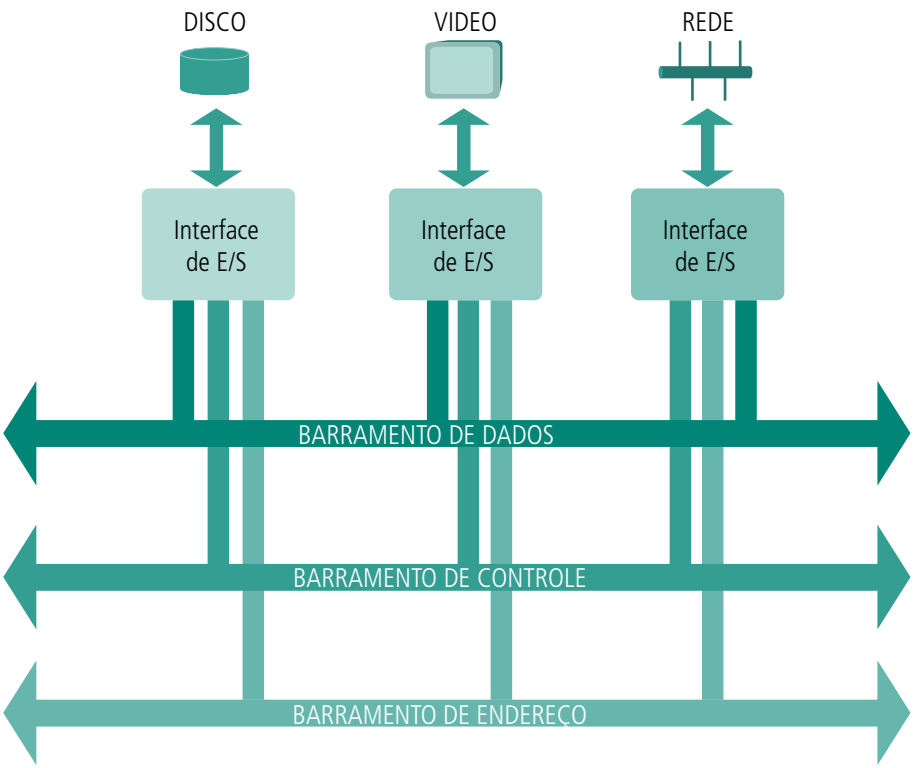


Figura 9.1: Exemplos de modelo de um módulo de E/S
Fonte: do autor

Segundo Stallings (2002), os periféricos não são ligados diretamente ao barramento do sistema pelos seguintes motivos:

a) a grande variedade de periféricos com diferentes mecanismos de operação torna impraticável incorporar ao processador toda a lógica necessária;

b) a taxa de transferência de dados dos periféricos é muito menor do que a taxa de transferência de dados da memória ou da CPU, tornando inviável o uso do barramento do sistema para comunicação direta com os dispositivos periféricos;

c) os formatos de dados e tamanhos das palavras dos periféricos são usualmente diferentes dos usados no computador ao qual estão conectados.

Para resolver os problemas de comunicação com o barramento, o módulo de E/S fornece uma interface com o processador e a memória e permite também interfaces com um ou mais dispositivos periféricos.

9.1 Dispositivos externos

Os dispositivos externos oferecem um meio de troca de dados entre o ambiente externo e o computador. A conexão, como já sabemos, é feita através de um módulo de E/S.

Os dispositivos externos podem ser classificados em três categorias:

a) voltados para a comunicação com o usuário - podemos citar aqui os monitores, terminais de vídeo, impressoras, teclados, *mouse*, dentre outros;

b) voltados para a comunicação com a máquina - discos magnéticos (discos rígidos) e discos ópticos são exemplos. É importante lembrar que, nesse momento, o ponto de vista das unidades é estrutural, ou seja, a forma como eles são controlados pelos módulos de E/S. Do ponto de vista de hierarquia, eles já foram apresentados como memória externa;

c) voltados para a comunicação com dispositivos remotos - redes de computadores usando interfaces de rede ou modem podem servir de exemplo.

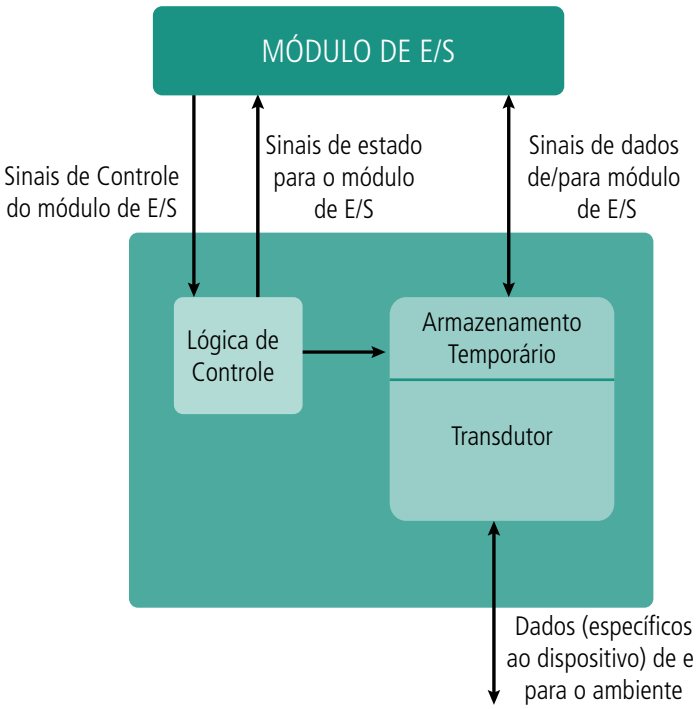


Figura 9.2: Modelo geral de um dispositivo externo
Fonte: do autor

Na Figura 9.2, podemos descrever detalhes da interação do dispositivo ex¬terno com o módulo de E/S. A interface com o módulo de E/S é formada de sinais de controle, estado e dados. Observe:

Um exemplo clássico que podemos observar com relação aos sinais de estado é o uso de uma impressora. Esse dispositivo, geralmente, informa ao usuário alguns status, como por exemplo, ociosa, pronta, sem papel, sem tinta, etc.

a) os sinais de controle determinam a função a ser executada pelo dispositivo (como enviar dados para o módulo de E/S – realizando um *INPUT* – ou receber dados do módulo de E/S – realizando um *OUTPUT*);

b) os sinais de estado indica o status do dispositivo, como por exemplo, se ele está pronto ou não (ligado, desligado, ocupado);

c) os sinais de dados formam o conjunto de *bits* a serem enviados para o módulo de E/S ou recebido dele.

A lógica de controle controla a operação do dispositivo, como o recebimento de um comando para movimentar o cabeçote de um disco para determinada posição. Essa função é dada pelo sinal de controle. O transdutor converte dados decodificados para sinais elétricos e vice-versa e geralmente usa um espaço temporário para armazenar os dados trabalhados.

9.2 Módulos e funções de E/S

As funções mais importantes de um módulo de E/S podem ser assim divididos:

a) controle e temporização;

b) comunicação com o processador;

c) comunicação com dispositivos;

d) área de armazenamento temporário de dados;

e) detecção de erros.

Levando em consideração que o processador pode se comunicar com um ou mais dispositivos para realizar E/S, um acesso a disco, por exemplo, o módulo de E/S inclui funções de controle e temporização. Observe o Quadro 9.1 a seguir.

Quadro 9.1: Etapas de controle e temporização de E/S

Etapa	Descrição
1	O processador interroga o módulo de E/S para verificar seu *status*.
2	O módulo de E/S retorna o *status* do dispositivo.
3	Se o dispositivo estiver em operação e pronto, o processador requisitará a transferência de dados, por um comando para o módulo de E/S.
4	O módulo de E/S recebe um bloco de dados do dispositivo externo.
5	Os dados são transferidos do módulo de E/S para o processador.

Conforme já citado, o armazenamento temporário de dados se constitui numa função extremamente necessária. Enquanto a taxa de transferência de dados entre a memória principal e o processador é bastante alta, as ordens de grandezas dos dispositivos periféricos são bem menores, compreendendo numa ampla faixa de valores. Este armazenamento serve como *buffer* (memória intermediária), agilizando os processos envolvendo o módulo de E/S e o processador..

O modelo apresentado no Quadro 9.1 indica que há necessidade de interação entre o módulo de E/S e o processador. Com isso, a comunicação com o processador envolve os seguintes aspectos:

a) decodificação de comando - o módulo de E/S recebe comandos do processador, enviados como sinais, pelo barramento de controle;

b) dados - são os dados transferidos entre o processador e o módulo de E/S pelo barramento de dados;

c) informação de estado - os periféricos, em geral, são mais lentos do que o conjunto processador-memória. Assim, é necessário saber seu estado antes de requisitar seus recursos novamente, como por exemplo, a escrita de um arquivo no disco, sendo que o módulo de E/S ainda estava processando uma solicitação anterior;

d) reconhecimento de endereço - da mesma forma que cada *byte* em memória tem seu endereço, cada dispositivo está associado a um endereço de E/S exclusivo.

Por último e não menos importante, o módulo de E/S precisa realizar detecção de erros. Tais erros podem variar desde um mau funcionamento mecânico ou elétrico até a alteração no padrão de *bits* transmitidos. Para se detectar erros de transmissão, utiliza-se algum tipo de código de transmissão. Um exemplo clássico é o **bit** de paridade.

Atividades de aprendizagem:

Complete as lacunas em branco com o conceito correto:

1. O _____ é o responsável por fazer interface entre os dispositivos externos e o processador.

2. Os dispositivos externos voltados à comunicação com a _____ podem ser os discos rígidos, unidades ópticas e unidades de fitas.

3. Os _____ emitem funções, como por exemplo, uma ação de *INPUT*, enviando dados para o módulo de E/S.

4. Os _____ indicam se um dispositivo, por exemplo, está ou não pronto para receber dados.

5. Ao receber um comando para gravar determinado setor do disco, a _____ controla a ação do dispositivo, enviando comandos primitivos para que ele movimente sua cabeça de gravação até determinado setor.

Assinale (V) ou (F) para as proposições:

a) ( ) Na primeira etapa de controle e temporização de E/S, o módulo de E/S retorna o status do dispositivo.

b) ( ) Para se comunicar com o processador, o módulo de E/S utiliza um endereço exclusivo para cada periférico.

c) ( ) Na decodificação de comando, o módulo de E/S recebe, pelo barramento de endereço, os sinais enviados pela unidade de controle do processador.

A-Z

Bit

O *bit* de paridade funciona da seguinte forma: na transmissão de um bloco de dados há uma associação a um tipo de código de paridade, que pode ser ímpar ou par, dependendo de como o circuito trabalha. Dessa forma, conta-se a quantidade de *bits* "1" transmitidos; por exemplo: na transmissão do bloco "0 1 1 0 1 1 0", qual é o *bit* de paridade de considerarmos o código como ímpar?
Solução: no bloco do exemplo, existem quatro *bits* "1". Como a paridade é ímpar, o último *bit* (chamado de *bit* de paridade) será 1, ficando assim: 0 1 1 0 1 1 0 1 . Se no mesmo exemplo fosse considerado o código como par, o *bit* de paridade seria "0", pois a quantidade de *bits* "1" é quatro (que é um número par).

d) ( ) A existência de um *buffer* no processo de troca de dados entre o módulo de E/S e o processador, faz com que a CPU trabalhe numa velocidade menor, para acompanhar os dispositivos externos lentos.

e) ( ) O *bit* de paridade da sequência 1 1 1 1 1 1 0 1 (nessa sequência não há o *bit* de paridade) é "1" se levarmos em consideração circuito utilizando paridade ímpar.

f) ( ) A sequência 0 0 1 0 1 1 0 0 1 (o último *bit* sublinhado já é o *bit* de paridade) está correto se consideramos circuito utilizando paridade par.

9.3 Principais técnicas de E/S

Existem, basicamente, três técnicas principais de E/S, que são:

a) E/S programada (PIO – *Programmed Input Output*), que é efetuada pelo programa que requisitou a operação de E/S;

b) E/S dirigida por interrupção, o programa requisita a E/S ao módulo de E/S e continua sua execução até que ocorra uma interrupção gerada por *hardware*;

c) Acesso direto a memória (DMA – *Direct Memory Access*), em que a E/S é controlada por um processador especial, encarregado por transferir blocos de dados.

A seguir serão abordados aspectos relevantes sobre todas as técnicas apresentadas.

9.3.1 E/S programada

Nessa técnica, o processador é utilizado intensamente para a realização de uma operação de E/S. Isso acontece, pois, durante o período da operação, a CPU fica a todo o momento questionando se o dispositivo está pronto ou não para uma nova operação.

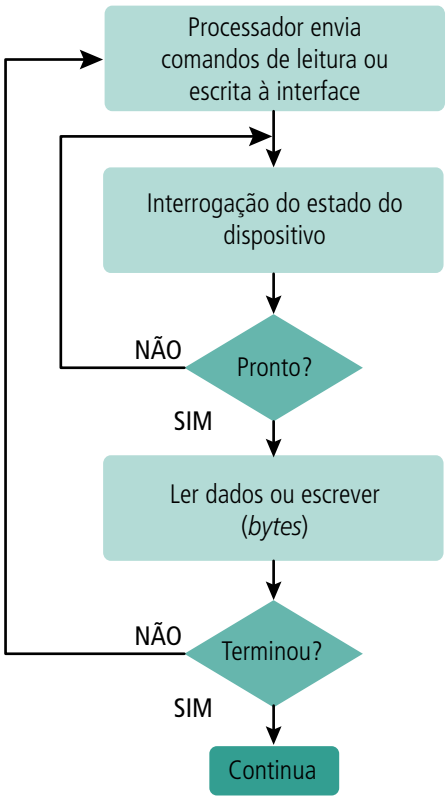


Figura 9.3: Fluxograma da técnica de E/S programada
Fonte: Introdução à Organização de Computadores 5ª Ed. Mário A. Monteiro,2007(Adaptação).

Conforme explica Monteiro (2007), na E/S programada, o processador testa o estado do dispositivo para verificar sua situação, podendo estar pronto, livre ou ocupado. Essa é a fase de interrogação, e, desse modo, se o dispositivo não estiver respondendo pronto, o processador realiza um *loop* de interrogação (observe a Figura 9.3). Ainda no mesmo fluxograma da Figura 9.3, quando o estado é reconhecido como pronto, o processador comanda a operação de escrita ou leitura, até completar todo o processo.

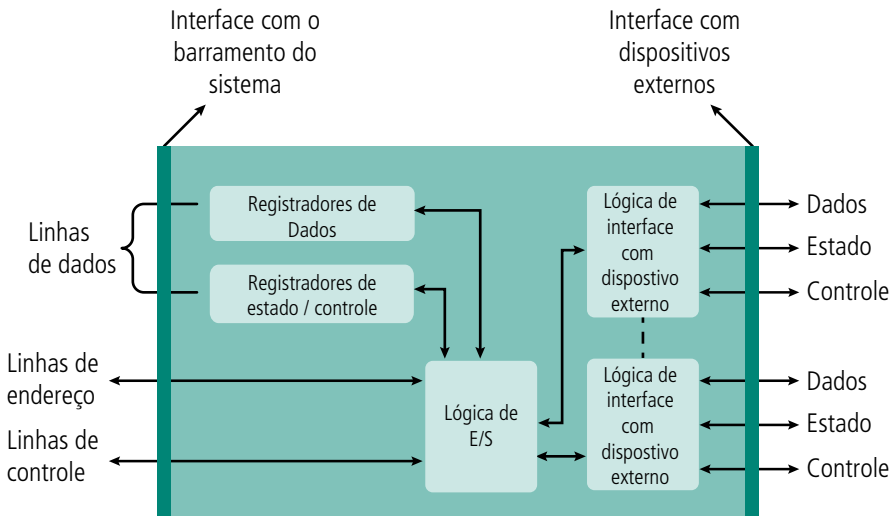


Figura 9.4: Diagrama de blocos de um módulo de E/S
Fonte: Arquitetura e Organização de Computadores: Projeto para o Desempenho. Willian Stallings, 2002 (Adaptação).

A Figura 9.4 consiste num diagrama funcional básico de um módulo de E/S. Na E/S programada, para executar uma operação de E/S, o processador deve executar os seguintes passos:

a) enviar o endereço do dispositivo através das linhas de endereço do barramento do sistema;

b) enviar o comando, que é colocado nas linhas de controle e pode ser, por exemplo, de leitura. Dessa maneira, a lógica de E/S interpreta o comando, busca o byte do dispositivo e o coloca em seu registrador de dados.

c) O processador precisa posteriormente receber o dado, solicitando que a interface o coloque nas linhas de dados do barramento do sistema.

Há uma grande desvantagem na E/S programada: o uso intenso do processador para comandar operações corriqueiras de E/S em vez de ser utilizado para o processamento de outras atividades mais importantes.

9.3.2 E/S dirigida por interrupção

Vimos que o método de E/S programada usa muito tempo do processador, principalmente nas interrogações do estado do dispositivo. Um método alternativo a esse modelo é o uso de uma técnica chamada interrupção. A seção 4.3.2 abordou os tipos de interrupções existentes. Revise-a antes de continuar.

As interfaces de E/S para discos rígidos antigos utilizam o PIO Mode (Modo E/S programado) nas ações de leitura e escrita. Isso era facilmente visto no *software* de monitoramento do uso da CPU quando havia leitura ou gravação de grandes blocos de dados.

No caso do uso de interrupções, a comunicação do processador com o módulo de E/S e o periférico funciona da seguinte forma:

a) o processador emite uma instrução para o módulo de E/S, e como não deverá ter logo uma resposta, ele não espera o periférico ficar pronto, desviando sua atenção para atender outra atividade (executar outro programa);

b) quando o periférico está pronto, seu módulo de E/S avisa o processador por um sinal de interrupção, o processador pára o que esta fazendo e atende ao dispositivo que o chamou;

c) finalmente, o processador inicia o programa de E/S, como no método anterior (parte de baixo da Figura 9.3).

9.3.3 Acesso Direto à Memória (DMA – *Direct Memory Access*)

Para Monteiro (2007), a melhor alternativa para se realizar operações de E/S com o máximo de rendimento do processador é utiliza a técnica de Acesso Direto à Memória -DMA.

De modo geral, essa técnica consiste na realização de transferência de dados entre uma determinada interface e a memória principal praticamente sem intervenção do processador.

Isso é possível porque a técnica de DMA envolve a adição de um módulo adicional ao barramento do sistema. Esse módulo é capaz de imitar o processador e controlar seu sistema, impondo interrupções a ele quando precisa transferir blocos de dados dos módulos de E/S para a memória e vice-versa.

Para que a técnica de DMA funcione corretamente, o módulo de DMA usa o barramento quando o mesmo não está sendo usado pelo processador ou quando esse módulo força o processador a suspender sua operação. Esta última ação é chamada de roubo de ciclo.

De forma prática, quando o processador deseja ler ou escrever um bloco de dados, um comando é enviado ao módulo DMA com as seguintes informações:

a) indicação de operação de leitura ou de escrita, enviada pela linha de controle de leitura ou de escrita entre o processador e o módulo de DMA;

b) o endereço do dispositivo de E/S envolvido;

c) o endereço de memória para início da leitura ou escrita de dados;

d) número de palavras a serem lidas ou escritas.

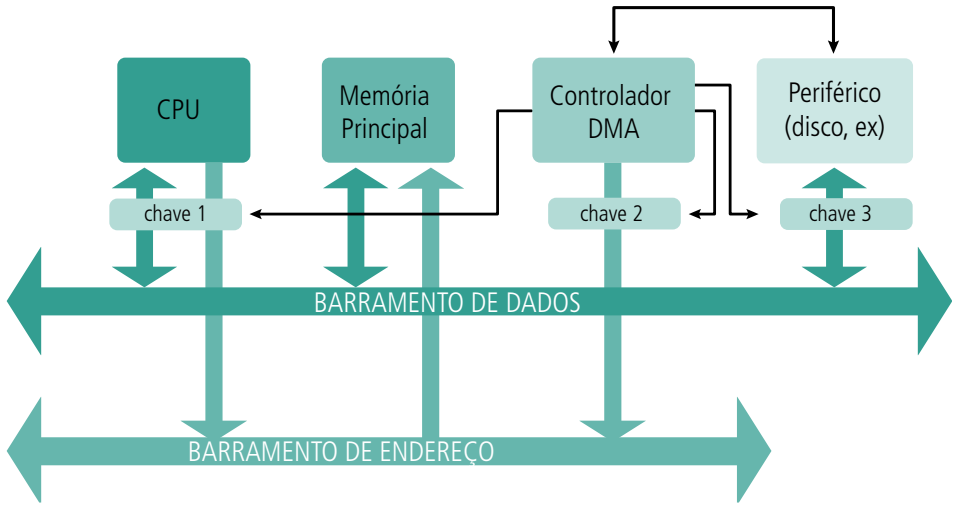


Figura 9.5: Operação de E/S com o uso da técnica de DMA
Fonte: do autor

Vamos exemplificar como funciona esse processo. Na Figura 9.4, durante a operação normal da CPU, sua chave (chave 1) esta ligada, permitindo a passagem de dados e endereços entre CPU e memória principal. Enquanto isso, as chaves do controlador de DMA e de Periférico estão desligadas (chaves 2 e 3). Quando o controlador de DMA entra em ação, as chaves 2 e 3 são ligadas. Dessa forma, o controlador de DMA coloca um endereço no barramento de endereço para a memória principal, e, ao mesmo tempo, autoriza (por sinal de controle) que o periférico envie ou receba dados diretamente da memória principal. Quando esta operação finaliza, o controlador de DMA devolve o controle do barramento ao processador.

Resumo

O módulo de E/S é responsável por coordenar o acesso aos recursos de processamento. Um dispositivo não pode estar conectado diretamente ao barramento de acesso ao processador, pois ele possui várias primitivas de acesso que o processador pode não entender, lembrando que existem milhares de dispositivos de E/S. Outro detalhe é que eles são mais lentos do que o processador, o que poderia tornar o sistema ineficiente. Assim, o módulo de E/S faz interface com o processador e o seu barramento. Com relação a E/S em memória, que constitui um processo que necessita de alto desempenho, a técnica de DMA é a mais eficiente, pois é adicionado um controlador de memória específico para controlar o uso da memória pelo dispositivo, liberando o processador para executar outras instruções.

Atividades de aprendizagem

Responda as seguintes questões:

1. Qual a ideia principal de E/S dirigida por interrupção?

2. Por que a E/S programada utiliza tanto tempo do processador?

3. Qual a importância do controlador de DMA para E/S? Por que ele é a melhor técnica de E/S existente?

Referências

MACHADO, Francis B. **Arquitetura de sistemas operacionais. 4º Ed.** Rio de Janeiro. LTC. 2007.

MONTEIRO, Mario A. **Introdução à Organização de Computadores. 5ª ed.** – Rio de Janeiro: LTC, 2007.

MORIMOTO, Carlos E. ***Hardware*, o Guia Definitivo.** Porto Alegre: GDH Press e Sul Editores, 2007.

STALLINGS, Willian. **Arquitetura e Organização de Computadores: Projeto para o Desempenho.** Tradução de Carlos Camarão de Figueiredo e Lucília de Figueiredo. São Paulo: Prentice Hall, 2002.

TANEMBAUM, Andrew S. **Sistemas Operacionais Modernos. 2ª ed.** Tradução de Ronaldo A.L. Gonçalves e Luis A. Consularo. São Paulo: Prentice Hall, 2003.

Currículo do professor-autor

Graduado em Processamento de Dados, com especialização em Informática Educativa. Professor do IFES/ Campus Colatina desde 2003, lecionando disciplinas para o curso Técnico em Informática e Superior de Tecnologia em Redes de Computadores. Em EaD, é o responsável pelas disciplinas de Arquitetura de Computadores, Fundamentos de *Hardware* e Montagem e Manutenção de Computadores, Introdução e Redes, Projeto de Redes e Laboratório de Redes. Atua na área de *hardware* e redes desde 1996, quando se formou Técnico em Processamento de Dados pela ETFES (Escola Técnica Federal do Espírito Santo, hoje, IFES). Participou da elaboração do projeto do Curso Superior de Tecnologia em Redes de Computadores e dos Cursos Técnicos em Informática (modalidade presencial e EaD). Também é consultor técnico de empresas que necessitam expandir seu parque tecnológico de comunicação e servidores.

e-Tec Brasil
Escola Técnica Aberta do Brasil

ISBN: