

# **Arquitetura e Organização de Computadores**

***Governador do Estado do Amazonas***

José Melo

***Universidade do Estado do Amazonas - UEA***

Cleinaldo de Almeida Costa

Mario Augusto Bessa de Figueiredo

Wladimir Leite Correia Filho

André Luiz Tannus Dutra

Luciano Balbino dos Santos

Samara Barbosa de Meneses

Glaucia Maria de Araújo Ribeiro

Valteir Martins

***Reitor***  
***Vice-Reitor***  
***Pró-Reitor de Administração***  
***Pró-Reitor de Extensão e Assuntos***  
***Comunitários***  
***Pró-Reitor de Graduação***  
***Pró-Reitora de Interiorização***  
***Pró-Reitora de Planejamento***  
***Pró-Reitor de Pesquisa e Pós-***  
***Graduação***

***Curso***  
***Coordenador***  
***Disciplina***  
***Professora***

Licenciatura em Computação – Humaitá

Marcelo Carvalho Tavares

Arquitetura e Organização de Computadores

Luiz Sérgio de Oliveira Barbosa

***Coordenador***  
***Design Instrucional***  
***Designer***  
***Bolsista***  
***Bolsista***  
***Bolsista***  
***Bolsista***

***Pró-Inovalab – Itacoatiara***

Marcelo Carvalho Tavares

João da Mata Libório Filho

Raimundo Nonato Nunes do Nascimento

Luis Paulo da Silva Lima

Gustavo Simão Rabelo

Lennon Simões Azevedo

Alessandro da Silva Silveira

## Sumário

APRESENTAÇÃO .....	6
CAPÍTULO 1 .....	7
INTRODUÇÃO A ORGANIZAÇÃO DE COMPUTADORES.....	7
1.1 Breve histórico dos computadores.....	8
1.1.2 computadores Mecânicos .....	8
1.1.3 Computadores Eletrônicos Analógicos .....	11
1.1.4 Computadores Eletrônicos Digitais.....	12
1.1.5 Computadores com Transistores.....	13
1.1.6 Computadores com Circuito Integrado .....	14
1.1.7 Computadores com Micro Chips.....	15
1.1.8 Computadores atuais.....	16
1.1.9 Classificação dos Computadores: .....	16
1.2 Capacidade de armazenamento de dados.....	18
1.3 O computador e seus componentes (Parte Física e Parte Lógica) .....	20
1.3.1 Hardware .....	20
1.3.2 Processador .....	20
1.3.3 Memória .....	21
1.3.4 A Placa-Mãe.....	21
1.3.5 Software .....	22
CAPÍTULO 2 .....	24
COMPONENTES DE FLUXO DE DADOS.....	24
2.1 Processamento e Armazenamento de Dados .....	24
2.2 Fases do Processamento de Dados: .....	25
CAPÍTULO 3 .....	25
BARRAMENTOS.....	25
3.1 Sistemas de Barramento .....	26
3.1.1 Funções do Barramento .....	26
3.2 Hierarquias de Barramentos .....	27
3.3 Tipos de Barramentos.....	27
Barramento do Processador.....	27
Barramento de Cache.....	28

Barramento de Memória.....	28
Barramento de Entrada e Saída .....	28
Barramento de Dados.....	29
3.3.1 Taxas de Transferência.....	29
CAPÍTULO 4 .....	31
MEMÓRIA.....	31
4.1 Sistemas de Memória.....	32
4.2 Tipos de Memória .....	33
RAM e DRAM.....	33
DIP e SIMM .....	34
FPM e EDO .....	34
DIMM e SDRAM .....	35
RIMM e PC100 .....	35
DDR, DDR2 e DDR3.....	36
4.3 Hierarquia de Memória.....	37
4.4 Memória Cache e Memória Virtual.....	39
4.5 Organização das Caches .....	39
Mapeamento Direto.....	39
Totalmente associativa .....	40
Associatividade por Conjunto (N-Way) .....	41
4.6 Mapeamento de memória cache.....	41
CAPÍTULO 5 .....	42
5.1 Dispositivos e Controladoras de Entrada/Saída .....	43
5.2 Métodos de Entrada e Saída .....	47
CAPÍTULO 6 .....	52
CPU .....	52
6.1 Unidade aritmética e lógica.....	53
6.2 Unidade de controle e microprogramação.....	54
6.3 Execução de Instruções .....	58
6.4 Ciclo de execução de uma Instrução.....	60
6.5 Interrupções.....	60
6.6 Pipeline.....	61
6.7 Operandos .....	63
6.8 Modo de Endereçamento .....	64

CAPÍTULO 7 .....	65
PROCESSADORES .....	65
7.1 CISC.....	65
7.2 RISC.....	66
<b>Comparação</b> .....	69
7.3 Máquinas Superescalares .....	69

## APRESENTAÇÃO

Prezado Aluno!

Você está iniciando mais uma etapa do Curso Licenciatura em Computação da Universidade do Estado do Amazonas (UEA). O componente curricular “Arquitetura e Organização de Computadores” é importante para sua formação, pois aborda temas relevantes sobre o estudo da estrutura funcional e organizacional de um sistema computacional. Todas as áreas da Ciência da Computação baseiam-se no processamento de informações que por sua vez são realizados por meio de operações básicas, mas que são utilizadas na implementação de poderosos e complexos sistemas computacionais.

O conteúdo deste componente curricular contempla todo o material necessário ao suporte para o seu aprendizado. Neste formato, a disciplina Arquitetura e Organização de computadores foi elaborada pensando numa leitura rápida e dinâmica, abordando o centro de cada conteúdo, explanado em aulas bem objetivas.

Para uma melhor compreensão, o conteúdo deste componente curricular está estruturado em tópicos e sub-tópicos. Este material disponibilizará o conhecimento básico necessário ao entendimento da operação do hardware dos computadores.

Esta disciplina é composta por sete tópicos de acordo com a ementa e estão assim distribuídos: Tópico 1: Introdução a Organização de Computadores, Tópico 2: Componentes de fluxo de dados, Tópico 3: Barramentos, Tópico 4: Memória, Tópico 5: Dispositivos de Entrada e Saída, Tópico 6: CPU, Tópico 7: Processadores. Portanto, será necessário que você se empenhe em ler o conteúdo e durante este processo seja capaz de identificar os pontos principais de cada um destes tópicos. Vale ressaltar que os temas aqui mencionados e trabalhados em sala de aula, necessitam para uma melhor compreensão e assimilação, separa um tempo para as leituras complementares do material que ficará disponível no Ambiente Virtual de Aprendizagem (AVA UEA), bem como a dedicação na resolução dos exercícios propostos.

**O Autor**

LUIZ SÉRGIO DE OLIVEIRA BARBOSA

## CAPÍTULO 1

### INTRODUÇÃO A ORGANIZAÇÃO DE COMPUTADORES

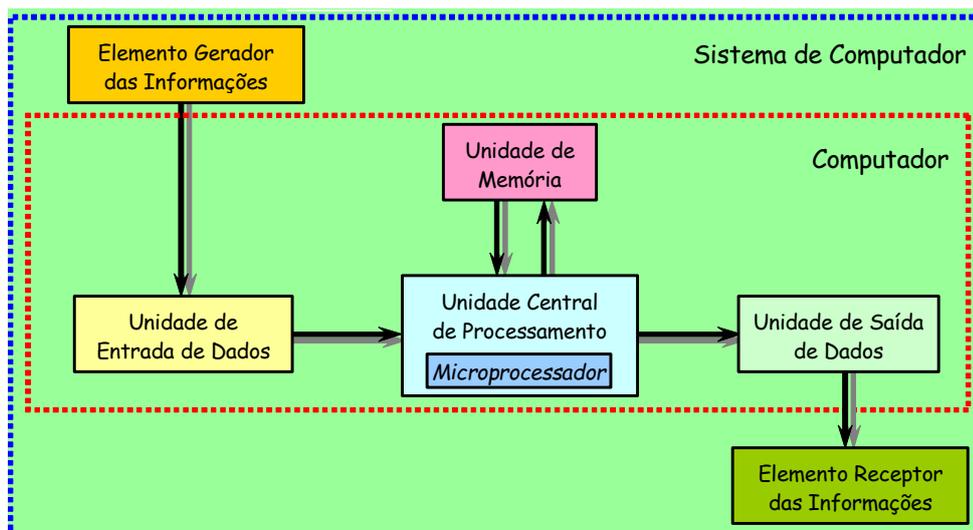
#### Visão geral

**Arquitetura de Computadores:** refere-se aos atributos de um sistema que são visíveis para o programador, ou seja, que têm impacto direto sobre a execução lógica de um programa:

- conjunto de instruções
- Número de bits usados para representar dados
- Mecanismos de E/S
- Técnicas de endereçamento de memória

**Organização de Computadores:** refere-se às unidades operacionais e suas interconexões que implementam as especificações da sua arquitetura:

- Sinais de controle
- Interfaces com periféricos
- Tecnologia de memória



Por exemplo, vamos imaginar uma determinada instrução. Implementar ou não a instrução é uma decisão de projeto da arquitetura do computador. Por outro lado, a

forma de implementação constitui uma decisão do projeto de organização do computador. Ou seja, a organização é projetada para implementar uma especificação particular de arquitetura.

Embora tenham ocorrido revolucionárias transformações na área de Eletrônica, o microcomputador de hoje ainda mantém a mesma concepção funcional dos primeiros computadores eletrônicos. Tal concepção, conhecida como Arquitetura de Von Neumann, é definida da seguinte forma:

Uma unidade central de processamento recebe informações através de uma unidade de entrada de dados, processa estas informações segundo as especificações de um programa armazenado em uma unidade de memória, e devolve os resultados através de uma unidade de saída de dados.

## 1.1 Breve histórico dos computadores

A história da informática confunde-se com a própria história humana, concebendo-a como sendo a ciência da informação.



Os primeiros instrumentos que o ser humano utilizou para facilitar os cálculos foram os dedos das mãos, essa ferramenta era suficiente para a época, pois as operações aritméticas a serem efetuadas eram muito simples.

### 1.1.2 computadores Mecânicos

**2600 a.c** → alguns pesquisadores consideram Stonehenge o 1º computador feito pelo homem, trata-se de um monumento paleolítico constituído de menires de 3 a metros de altura situado na Grã-bretanha;

**2000 a.c** → O ábaco é o nome genérico atribuído aos contadores em geral utilizado no oriente;

→ Era feito inicialmente de conchas e seixos e evoluiu para contas móveis que se movimentam em hastes.

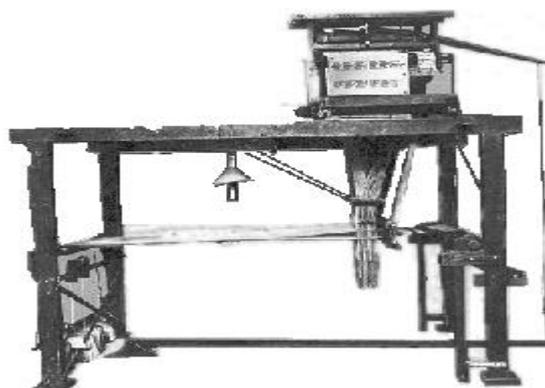


**1621** → O matemático inglês William Outgred inventa a régua de cálculo;

**1623** → Wilhem Schilkard começa a construção da 1ª máquina de calcular;

**1642/1647** → O francês Blaise Pascal, utilizando sua máquina conhecida como pascalina, conseguia somar e subtrair por meio de engrenagens mecânicas;

**1801** → O matemático francês Joseph Marie consegue armazenar informações em placas perfuradas para controle de máquinas de tecelagem, o fato gerou grande temor de desemprego, por ser considerada uma máquina pré-automação.



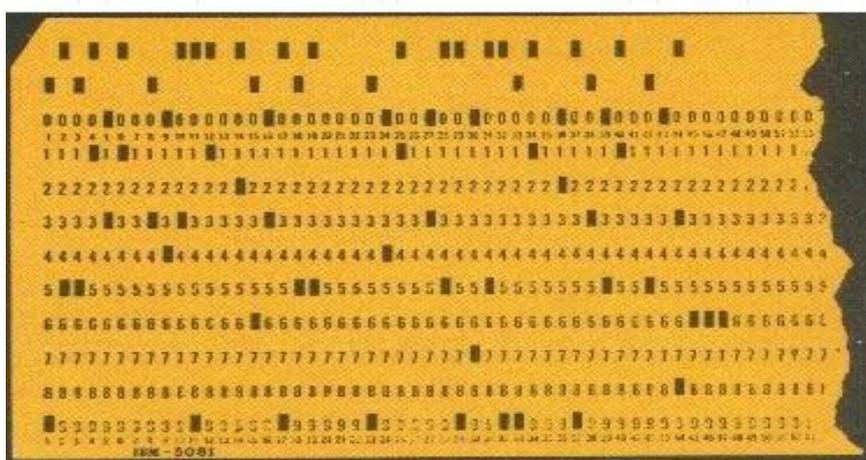
**1820** → Uma máquina de calcular idealizada pelo francês Charles Thomas vende mais de 1500 unidades, foi o 1º sucesso comercial nesse setor.

**1822** → O inglês Charles Babbage anuncia sua máquina diferencial, mas o motor analítico do computador era composto por engrenagens que não possuíam a precisão adequada, não funcionou de forma satisfatória;

Babbage, sabendo que precisava de um software para o motor, contrata a 1ª programadora do mundo, Ada Lovelace;

**1854** → George Boole estabelece os princípios binários que seriam utilizados posteriormente como base pra o estudo de lógica matemática;

**1880** → Hermann Hollerith, baseado nas ideias de Babbage e Joseph, constrói uma máquina de tabulação chamada “tabuladora” a qual foi usada no censo norte-americano;



**1896** → Hermann funda a Tabulating Machine Company;

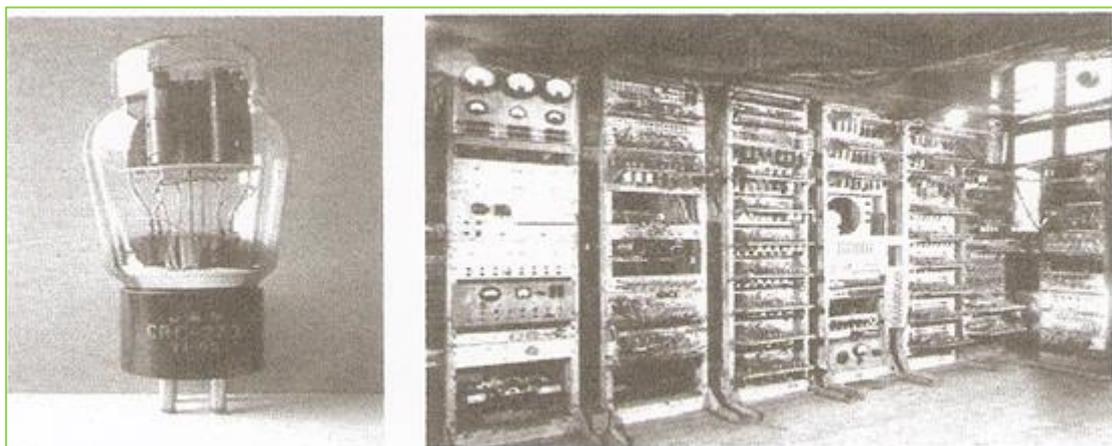


**1911** → Hermann associa-se a outras empresas e a Tabulating Machine Company passa a ser dirigida por Tomas Watson;

**1924** → Nasce a IBM ( International Business Machine), resultado da associação de Hermann e Watson.

### 1.1.3 Computadores Eletrônicos Analógicos

Esses computadores eletrônicos utilizavam válvulas que eram ligadas por kms de fios, com a invenção da válvula eletrônica foi possível realizar operações aritméticas por meio de circuitos eletrônicos;



**1931** → O 1º computador analógico é construído pelo MIT (Massachusetts Institute of Technology);

**1937** → IBM fabricou o 1º computador eletromecânico, o MARK I;

#### 1.1.4 Computadores Eletrônicos Digitais

**1939** → O 1º computador eletrônico digital, é apresentado pelo professor de matemática John Atanasoft, o 1º a usar válvula para os circuitos lógicos, mas sua construção foi abandonada em 1942;

**1943** → A Inglaterra constrói dez computadores COLOSSUS I, esse equipamento eletrônico digital a válvulas, foi utilizado para decifrar códigos militares dos alemães;

**1946** → O ENIAC (Eletronic Numeric Integrator Analyser and Calculator) foi apresentado como o 1º grande computador eletrônico, ocupava quase 200 metros quadrados, pesava 30 toneladas e utilizava 18 mil válvulas, 10 mil capacitores e milhares de relés e resistores, conseguia 5 mil adições por segundo;



**ENIAC**

**1950** → John Von Neuman, Arthur Burks e Hermn Goldstine desenvolvem a lógica dos circuitos, conceitos de programas e operações por números binários utilizados até hoje;

**1951** → O UNIVAC I, 1º computador a utilizar os conceitos de Von Neuman, é produzido em escala comercial, pesava 5 toneladas e ocupava 20 metros quadrados;



**UNIVAC I**

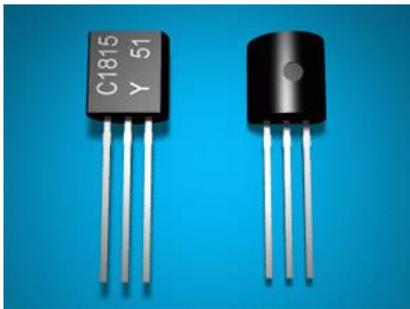
**1959** → Fim dos computadores pioneiros ou de 1ª geração, baseados em válvulas;

#### **Características da 1ª Geração:**

- Circuitos eletrônicos e válvulas;
- Uso restrito;
- Precisava ser reprogramado a cada tarefa;
- Grande consumo de energia;
- Problemas devido a muito aquecimento.

#### **1.1.5 Computadores com Transistores**

A 2ª geração dos computadores se deu entre as décadas de 50 e 60, onde o grande avanço foi a substituição das válvulas pelos transistores, seu tamanho era 100 vezes menor que o da válvula.



**Transistores**



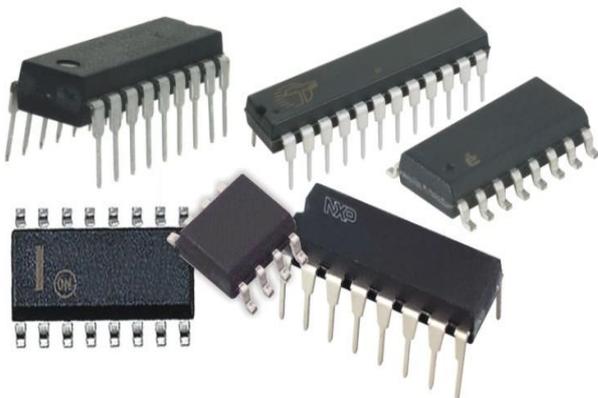
**IBM 1401**

### Características da 2ª Geração:

- Início do uso comercial;
- Grande ganho em velocidade, tamanho e custo;
- As linguagens utilizadas Fortran, COBOL ou ALGOL;
- Consumia menos energia, era mais rápido e confiável;
- Processamento em microssegundos.

### 1.1.6 Computadores com Circuito Integrado

A terceira geração se deu nas décadas de 60 e 70, os transistores foram substituídos pela tecnologia de circuitos integrados (associação de transistores em pequena placa de silício).



**Circuitos Integrados**



**IBM 360**

### Características da 3ª geração:

- Diminuição do tamanho;
- Maior capacidade de processamento em nanossegundos;
- Início dos computadores pessoais;
- Baixo consumo de energia;
- Mais confiáveis, compactos e menor custo.

#### 1.1.7 Computadores com Micro Chips

A 4ª geração se estende desde a década de 70 até os dias atuais, onde se teve uma otimização da tecnologia já existente, surgem os micros chips ou processadores.



**Micro Chips**



**Computador Pessoal**

### Características da 4ª Geração:

- Surgem os softwares integrados;
- Processadores de texto;
- Planilhas eletrônicas;
- Supercomputadores;
- Automação comercial e industrial;
- Robótica;
- Era on line.

### 1.1.8 Computadores atuais

Os computadores da quinta geração usam processadores com milhões de transistores. Nesta geração surgiram as arquiteturas de 64 bits, os processadores que utilizam tecnologias RISC e CISC, discos rígidos com capacidade superior a 600GB, pen-drives com mais de 1GB de memória e utilização de disco ótico com mais de 50GB de armazenamento.

A quinta geração está sendo marcada pela **inteligência artificial** e por sua **conectividade**. A inteligência artificial pode ser verificada em jogos e robôes ao conseguir desafiar a inteligência humana. A conectividade é cada vez mais um requisito das indústrias de computadores. Hoje em dia, queremos que nossos computadores se conectem ao celular, a televisão e a muitos outros dispositivos como geladeira e câmeras de segurança.

#### Características da 5ª Geração:

- O nascimento da Inteligência Artificial;
- Reconhecimento de voz;
- Sistemas inteligentes;
- Computação Distribuída;
- Computação nas Nuvens (Cloud Computing);
- Computação Móvel;
- Computação Ubíqua (presença direta das tecnologias na vida das pessoas, em casa ou em convívio social);
- Realidade Aumentada.

#### 1.1.9 Classificação dos Computadores:

- Quanto à característica de operação;
- Quanto ao porte (tamanho);
- Quanto à característica de construção.

## Quanto a Característica de Operação

**Analógicos** → São computadores que processam sinais elétricos variáveis, aplicados a problemas de controle de processos, sua precisão e velocidade são menores que a dos computadores digitais.

**Digitais** → Representam tanto a programação como os dados por meio de dígitos, através de ( 0 e 1), sua velocidade é medida em megahertz.

## Quanto ao Porte - Tamanho

- **Mainframes** (Computadores de Grande Porte)

Manipulam grande quantidade de informações atendendo vários usuários ao mesmo tempo, especialmente voltados a aplicações comerciais,  $v=10$  mips.

- **Supercomputadores**

Utilização em laboratórios de pesquisa, centros militares de inteligência artificial, muito rápido, avalia-se o desempenho através de instruções executadas por segundo,  $v=100$  mips.

- **Minicomputadores**

Possuem alguns dos recursos de um mainframe, tem um bom processamento e equipam laboratórios de empresas de desenvolvimento e centros de estudos.

- **Microcomputadores**

São máquinas voltadas para o uso de pequenas empresas, escolas e uso doméstico, são divididos em duas categorias: mesa (desktops e portáteis (notebooks)).

## Quanto a Característica de Construção

Os computadores são agrupados em geração, de acordo com a alteração da tecnologia:

Gerações	Primeira	Segunda	Terceira	Quarta	Quinta
Época	1957-1959	1959-1965	1965-1975	1975	199?
Exemplos Típicos	ENLAC UNIVAC I EDVAC WHIRLWIND IBM 650	IBM 1401 IBM 7094 CDC6600	IBM 360 IBM370 DECPDP-8	IBM 3090 CRAY Micros (evolução do computador digital)	Pentium
Tecnologia Básica (Componentes)	Válvula	Transistor	Circuito Integrado (CI)	CI - VLSI	CI - ULSI
Memória	2K	32K	128K	>1 M	M e G
MIPS	0.01	0.1	1K	> 10	> 100 ou gigaflops (1 bilhão de operações por segundo)

### 1.2 Capacidade de armazenamento de dados

Em Computação é muito importante considerar a capacidade de armazenamento, já que quando se faz algo no computador, trabalha-se com arquivos que podem ser guardados para uso posterior. Evidentemente, quando se armazena algo, isto ocupa um certo *espaço de armazenamento*.

Assim como a água é medida em litros ou o açúcar é medido em quilos, os dados de um computador são medidos em bits e bytes. Cada valor do código binário foi denominado "bit" (*binary digit*), que é a menor unidade de informação.

Cada conjunto de 8 bits forma o *byte*, o qual corresponde a um caracter, seguindo o código binário.

1        0        0        1        0        1        1        0

Por que 1 Kb equivale a 1024 bytes?

No caso do quilo e de outras medidas de nosso dia a dia, a estrutura numérica é construída sobre a base 10. O termo quilo representa a milhar constituída de alguma coisa. Nossa base de trabalho numérica, sendo 10, faz com que, quando a base é elevada à terceira potência, atinja a milhar exatamente com 1000 unidades.

Mas, quando falamos em bytes, grupos de bits, não estamos falando em base 10, mas sim em uma estrutura fundamentada no código binário, ou seja, na base 2, nos 2 modos que o computador detecta, geralmente chamados de 0 e 1.

Assim, quando queremos um quilo de bytes, temos que elevar essa base a algum número inteiro, até conseguir atingir a milhar.

Mas não há número inteiro possível que atinja exatamente o valor 1.000. Então, ao elevarmos a base 2 à décima potência, teremos 1024.

Com esse raciocínio agora podemos entender a seguinte tabela:

Medida	Sigla		Caracteres	
Byte		$2^0$	1 (8 bits)	1 byte
Kilobyte	<b>KB</b>	$2^{10}$	1.024	1.024 bytes
Megabyte	<b>MB</b>	$2^{20}$	1.048.576	1.024 KBytes
Gigabyte	<b>GB</b>	$2^{30}$	1.073.741.824	1.024 MBytes
Terabyte	<b>TB</b>	$2^{40}$	1.099.511.627.776	1.024 GBytes
Pentabyte	<b>PB</b>	$2^{50}$	1.125.899.906.842.624	1.024 TBytes
Hexabyte	<b>HB</b>	$2^{60}$	1.152.921.504.606.846.976	1.024 PBytes
Yotabyte	<b>YB</b>	$2^{80}$	1.208.925.819.614.630.000.000.000	1.024 Hexabyte

Neste caso, concluímos que, por exemplo, quando se diz que um disco rígido tem capacidade de armazenamento de 4,3 Gb, são armazenados aproximadamente 4 milhões e 500 mil caracteres.

## 1.3 O computador e seus componentes (Parte Física e Parte Lógica)

### 1.3.1 Hardware

**Hardware** é um termo em inglês que não tem uma tradução específica, deriva da palavra *hard* em inglês, que significa duro. É usado para definir todos os componentes físicos (mecânicos, magnéticos e eletrônicos) de um computador, ou seja, a máquina propriamente dita. O hardware de um computador pode ser dividido basicamente em quatro categorias: processador, placa-mãe, memória e dispositivos de entrada e saída.



### 1.3.2 Processador

O **processador**, também chamado de Unidade Central de Processamento (UCP), do inglês *Central Processing Unit* (CPU), é o principal componente eletrônico de um computador. Ele fica acoplado na placa-mãe e é o responsável pela execução, com auxílio da memória e dos dispositivos de entrada e saída, de todas as operações no computador. Pode ser considerado o “cérebro” do computador.



Processador da Intel



Processador da AMD

### 1.3.3 Memória

A **memória** do computador é formada por todos os dispositivos que são utilizados para armazenar dados e instruções, seja de forma temporária ou permanente. Pode-se dizer que depois do processador, a parte mais importante de um computador é a sua memória. Há diversos dispositivos submetidos a uma hierarquia formando a memória do computador, mas para se ter uma visão inicial simplificada, é possível dividi-los em dois grandes grupos principais: a memória principal e a memória secundária.

A **memória principal** é aquela que é acessada diretamente pelo processador e armazena os dados de forma eletrônica. Apresenta alta velocidade e baixa capacidade de armazenamento. É formada pela **memória RAM** (do inglês Random Access Memory, que significa “memória de acesso aleatório”), que armazena as informações que são utilizadas no trabalho do processador; e pela **memória ROM** (Read Only Memory, “memória somente para leitura”), que armazena as instruções de inicialização do computador.



**Memória principal**



**Memória secundária**

### 1.3.4 A Placa-Mãe

A **placa-mãe**, do inglês *motherboard*, também chamada *mainboard* (inglês para “placa principal”), é a principal placa do computador, responsável pela interconexão

de todos os dispositivos que o compõem. Se o processador é o “cérebro” do computador, pode-se comparar a placa-mãe a um “sistema nervoso”.



A Placa-Mãe

### 1.3.5 Software

Para que serviriam todos os circuitos eletrônicos de um computador se não fosse possível fazer nada com eles? Se não receber ordem para realizar alguma atividade, o computador não passa de algumas luzes acesas e circuitos pelos quais passam corrente elétrica. A princípio, os componentes eletrônicos do computador eram operados manualmente, o que era extremamente trabalhoso. Depois, criou-se uma forma de instruí-los a realizar determinada tarefa quando esta lhes fosse solicitada, facilitando o manuseio do computador. Eis que surgiu então o software.

Em contraposição ao hardware, que são os equipamentos, o **software**, ou programa, é a parte lógica do computador, ou seja, o conjunto de dados e instruções passadas para os componentes físicos de um computador para que ele possa executar determinada tarefa, para a qual o software foi projetado.

Software, assim como hardware, é um termo em inglês que não tem tradução, derivando da palavra *soft* em inglês, que significa “mole”. Esse termo surgiu na verdade como uma gíria da informática, sendo usado para distinguir os

componentes físicos do computador, as “ferragens” nas quais podemos tocar, dos componentes lógicos do computador, os programas, os quais não podem ser tocados.

A função do software é facilitar a interação dos usuários com os componentes físicos do computador, transformando-o em algo realmente útil. Existe um programa para cada tarefa que se deseja executar utilizando o computador. Para que um computador seja capaz de editar textos, por exemplo, deve possuir um programa editor de textos. Para que ele seja capaz de reproduzir um DVD, deve possuir um programa reprodutor de mídias. Para acessar sites na Internet, é utilizado um navegador, e assim por diante.

Nem todos os programas de computador, no entanto, realizam tarefas específicas. Estes são os chamados aplicativos. Há programas com funções mais gerais.

Há o software operacional (também chamado de **sistema operacional**) que é o software responsável por operacionalizar o computador, ou seja, torná-lo operacional. Ele gerencia a comunicação entre todos os seus componentes (entre hardware e software, entre um aplicativo e outro), e entre estes e o usuário, de forma a tornar o computador uma máquina funcional, capaz de “atender ordens”. Ele também dispõe de recursos para tornar mais fácil a comunicação entre o usuário e o computador, como a interface gráfica.

Há também o **BIOS** (do inglês *Basic Input/Output System*, sistema básico de entrada e saída), que é o primeiro programa a ser executado quando o computador é ligado. Ele reconhece os componentes de hardware instalados, verifica se todos estão funcionando e passa o comando do computador ao sistema operacional, provendo a estas informações sobre o hardware da máquina.

Podemos então classificar o software de um computador em dois grupos, diferentes entre si quanto à importância que eles possuem para o funcionamento da máquina e a sua utilidade: básicos e aplicativos.

Os **softwares básicos** de um computador são aqueles essenciais ao seu funcionamento, como o sistema operacional, o BIOS e os programas fornecidos pelo fabricante junto com a máquina. Já os **softwares aplicativos**, ou simplesmente

aplicativos, são desenvolvidos para auxiliar o usuário na execução de alguma tarefa ou na obtenção da solução para algum problema, ou seja, são softwares desenvolvidos para uma aplicação específica. Quase todos os programas que não são softwares básicos são softwares aplicativos.

## CAPÍTULO 2

### COMPONENTES DE FLUXO DE DADOS

O computador transforma dados em informação, através do hardware (parte física) e do software (parte lógica) que são os componentes responsáveis pelo processamento destes dados, transformando-o saída de informações.

#### 2.1 Processamento e Armazenamento de Dados

**Computador** → É uma máquina capaz de realizar os mais difíceis cálculos, bem como processar as mais difíceis informações a partir de dados nele introduzidos, para se obter um resultado rápido e eficaz.

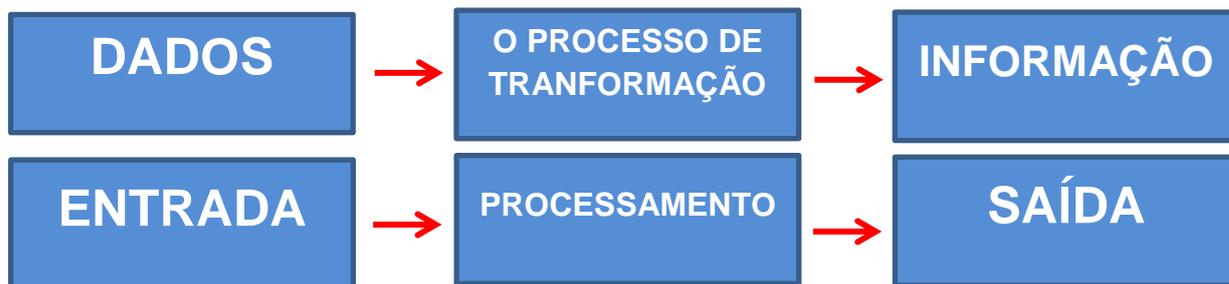
**Informação** → É um conjunto de dados que transforma uma condição, situação, fatos, ou seja, é matéria-prima com que o computador trabalha.

**Dados** → É o elemento básico da informação, é um registro da informação, qualquer elemento que sirva de ponto de partida para uma decisão, cálculo ou medida, os dados ao serem processados dão lugar à informação que é o resultado de processamento de dados. Exemplo: o nome de um funcionário, a quantidade de horas trabalhadas, quantidade de peças em estoque.

DADOS	REPRESENTAÇÃO
ALFANUMÉRICOS	LETRAS, NÚMEROS
IMAGENS	IMAGENS GRÁFICAS, FOTOS
ÁUDIO	SOM, RUÍDOS OU TONS
VÍDEO	IMAGENS EM MOVIMENTOS OU FOTOS

Exemplo de Tipos de Dados

## 2.2 Fases do Processamento de Dados:



**Entrada** → É atividade de reunir e coletar dados brutos (dados não trabalhados).

**Processamento** → Envolve a conversão e a transformação de dados brutos em dados úteis.

**Saída** → Envolve a produção da informação útil, geralmente em forma de documentos ou relatórios.

## CAPÍTULO 3

### BARRAMENTOS

**Barramento** é um conjunto de linhas de comunicação que permitem a interligação entre dispositivos, como a CPU, a memória e outros periféricos. **São** as linhas de transmissão que transmitem as informações entre o processador, memória e todos os demais periféricos do computador.

Praticamente todos os componentes de um computador, como processadores, memórias, placas de vídeo e diversos outros, são conectados à placa-mãe a partir do que chamamos de barramento. Sem entrar em termos técnicos, ele é o encaixe de que cada peça precisa para funcionar corretamente.

Há barramentos específicos para praticamente todos os componentes conectados ao sistema, geralmente em siglas muito conhecidas pelos usuários, mas que não são atreladas diretamente à função que realizam. Confira abaixo alguns tópicos interessantes sobre barramentos.

### 3.1 Sistemas de Barramento

Em Arquitetura de Computadores, um **barramento** é um conjunto de linhas de comunicação (condutor elétrico ou fibra ótica) que permitem a interligação entre dispositivos de um sistema de computação (CPU; Memória Principal; HD e outros periféricos), ou entre vários sistemas de computação.

O desempenho do barramento é medido pela sua largura de banda (quantidade de bits que podem ser transmitidos ao mesmo tempo), geralmente potências de dois:

- 8 bits, 16 bits, 32 bits, 64 bits, etc.

Também pela velocidade da transmissão medida em bps (bits por segundo) por exemplo:

- 10 bps, 160 Kbps, 100 Mbps, 1 Gbps etc.

#### 3.1.1 Funções do Barramento

Existem 3 funções distintas nos barramentos:

- Comunicação de Dados: função de transporte dos dados. Tipo bidirecional;

- Comunicação de Endereços: função de indicar endereço de memória dos dados que o processador deve retirar ou enviar. Tipo unidirecional, e;
- Comunicação de Controle: função que controla as ações dos barramentos anteriores. Controla solicitações e confirmações. Tipo bidirecional.

### 3.2 Hierarquias de Barramentos

- Quanto maior o número de dispositivos conectados maior o comprimento do barramento
- Assim maior o atraso na propagação dos sinais
- Esse atraso define o tempo para que um dispositivo obtenha o controle do barramento
- O atraso pode comprometer o desempenho
- O barramento pode se tornar um gargalo quando a demanda de dados se aproxima da sua capacidade de transmissão
- Aumentar a largura do barramento soluciona o problema mas amplia o espaço ocupado pelos dispositivos
- Outra alternativa é ampliar a velocidade de transferência, contudo nem todos dispositivos podem trabalhar e altas velocidades
- A solução é criar uma hierarquia de barramentos
- Num sistema hierárquico de barramentos existem vários níveis de barramento divididos pela prioridade e velocidade
- Estes se níveis se comunicam através de interfaces

### 3.3 Tipos de Barramentos

Na arquitetura de computadores são categorizados em:

#### **Barramento do Processador**

É utilizado pelo processador internamente e para envio de sinais para outros componentes do sistema computacional.

Atualmente, os barramentos dos processadores (os de transferência de dados) têm sido bastante aprimorados com o objetivo de maior velocidade de processamentos de dados.

### **Barramento de Cache**

É o barramento dedicado para acesso à memória cache do computador, memória estática de alto desempenho localizada próximo ao processador.

### **Barramento de Memória**

É o barramento responsável pela conexão da memória principal ao processador. É um barramento de alta velocidade que varia de micro para micro e atualmente gira em torno de 133 MHz a 1600 MHz, como nas memórias do tipo DDR3.

### **Barramento de Entrada e Saída**

É o barramento I/O (ou E/S), responsável pela comunicação das diversas interfaces e periféricos ligados à placa-mãe, possibilitando a instalação de novas placas, os mais conhecidos são: PCI, AGP e USB.

Os periféricos não se conectam diretamente ao barramento de sistema devido:

- Há uma larga variedade de periféricos com vários métodos de operação sendo inviável incorporar diversas lógicas de controle dentro do processador...
- A taxa de transferência de dados dos periféricos é normalmente mais baixa do que a da memória do processador sendo inviável usar o barramento de alta-velocidade para comunicação com periférico.
- Os periféricos usam normalmente formatos de dados e tamanhos de palavras diferentes dos do computador a que estão agarrados.

Permitem a conexão de dispositivos como:

- Placa gráfica; Rede; Placa de Som; Mouse; Teclado; Modem; etc.

São exemplos de Barramentos de Entrada e Saída:

- AGP; AMR; EISA; IrDA; ISA; MCA; PCI; PCI-e; Pipeline; SCSI; VESA; USB, e; PS/2.

### **Barramento de Dados**

É o barramento Data Bus, responsável por transportar informação da instrução (através do código de operação), variável do processamento (cálculo intermediário de uma fórmula por exemplo) ou informação de um periférico de E/S (dado digitado em um teclado). O tamanho da via de dados determina respectivamente o máximo número de instruções (e portanto o potencial de processamento) e a precisão do processamento aritmético (através do cálculo de ponto flutuante) ou o número de símbolos possíveis a ser representado (por exemplo, pontos de uma foto).

Exemplo:

- 1) O processador 8085 possui 8 bits no barramento → máximo de 256 instruções e variáveis numéricas entre -128 a +127 (ou 0 a 255).
- 2) O processador Pentium IV possui 32 bits no barramento → variáveis numéricas entre - 2147483648 e + 2147483647.
- O tamanho do barramento de dados está ligado à capacidade de processamento do sistema. Se o processamento é simples (como o controle booleano de um CLP) 8 bits são suficientes. Por outro lado, se há a necessidade de um processamento complexo (como os sistemas de multimídia onde há a necessidade de processarmos milhões de pontos de imagem) processamento de até 128 bits já estão disponíveis. Obviamente existe a necessidade de aumentar igualmente a velocidade do sistema pois a "paciência" do usuário é a mesma ao ligar uma lâmpada ou processar uma imagem fotográfica.

#### **3.3.1 Taxas de Transferência**

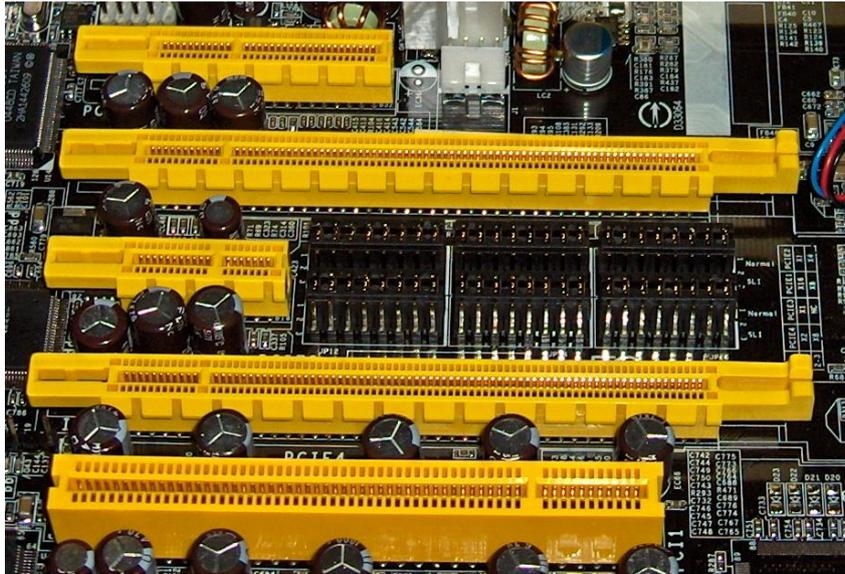
O principal fator que determina o desempenho de um barramento é a sua taxa de transferência. A mesma é obtida através de cálculos com base nas características do barramento, sendo elas:

- Clock (relógio)

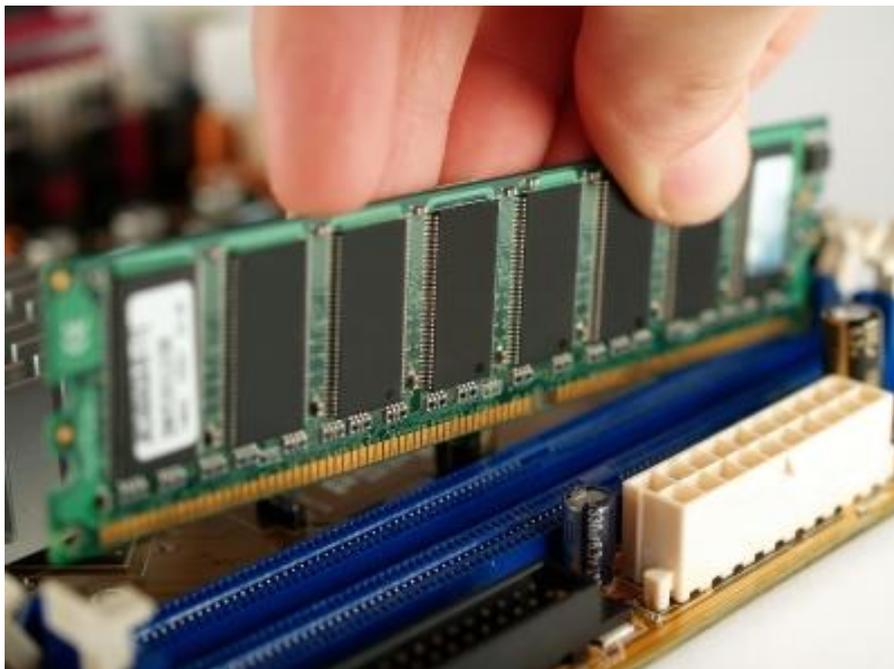
- Largura
- Transferências por ciclo de clock

A fórmula teórica para calcular a frequência máxima de transmissão é:

- taxa de transferência  $\rightarrow n^{\circ}$  de bits  $\times$  clock total  $\times n^{\circ}$  de dados transmitidos por clock / 8



**Barramentos**



**Barramento de memória**

## CAPÍTULO 4

### MEMÓRIA

Existem em um microcomputador dois tipos de memória:

#### Memória Principal ou Primária

É um conjunto de chips que acumulam as informações enquanto estão sendo processadas, cuja ação se restringe ao tempo em que o micro estiver ligado. A memória RAM é vendida na forma de pequenas placas chamadas “pentes”

#### Memória RAM

RAM significa memória de acesso aleatório, ou seja, o computador altera seu conteúdo constantemente, como a memória RAM não grava os dados de forma permanente, seu conteúdo é esvaziado quando desligamos o computador, sem chance de recuperação.

MEMÓRIA PRINCIPAL	IMAGEM
MEMÓRIA RAM DO TIPO DIMM DE 168 PINOS (DUAL IN-LINE MEMORY MODULE)	
MEMÓRIA RAM DO TIPO SIMM DE 30 PINOS (SINGLE IN-LINE MEMORY MODULE)	
MEMÓRIA RAM DO TIPO RIMM DE 168 PINOS (RAMBUS IN-LINE MEMORY MODULE)	

Exemplos de Memória RAM

#### Memória Auxiliar ou Secundária

É aquela usada para gravar os dados física e permanentemente, uma vez gravados os dados em qualquer tipo de memória auxiliar, os mesmos não serão perdidos caso o micro seja desligado.

## Memória ROM

A memória ROM (read only memory) mantém informações básicas sobre o hardware do computador, gravada fisicamente em um chip pelo fabricante do micro.

MEMÓRIA SECUNDÁRIA	IMAGEM
HARD DISK (HD)	
DVD	
CD	
MEMÓRIA USB	

**Exemplos de Memória Secundária**

### 4.1 Sistemas de Memória

A memória principal, também chamada de memória real, são memórias que o processador pode endereçar diretamente, sem as quais o computador não pode funcionar. Estas fornecem geralmente uma ponte para as secundárias, mas a sua função principal é a de conter a informação necessária para o processador num determinado momento; esta informação pode ser, por exemplo, os programas em execução.

Nas memórias externas, basicamente se referem aos dispositivos de armazenamento periféricos, como discos ópticos e fitas magnéticas, que são acessados através dos controladores de E/S.

Existem outros tipos de memórias internas, assim como os registradores do processador, memória principal cache. Quando falamos deste tipo de memória, estamos falando uma capacidade pequena em relação a externas, já que tem uma capacidade em bytes (1 byte = 8 bits) ou palavras, normalmente atingindo 8, 16, 32 bits. Quanto a unidade de transferência, geralmente é igual ao número de linhas

elétricas para dentro e para fora do módulo de memória, podendo ser igual ao tamanho da palavra, mas normalmente é maior, como 64, 128 e 256 bytes.

## 4.2 Tipos de Memória

**ROM** → Read Only Memory (gravada de fábrica - permanente);

**PROM** → Programmable ROM (pode ser gravada uma única vez pelo usuário, através de equipamentos especiais);

**EPROM** → Erasable PROM (pode ser apagada através de raios ultravioletas);

**EEPROM** → Electrical EPROM (pode ser apagada através de sinais elétricos);

**Memória Flash** → é uma memória que pode ser utilizada como apenas de leitura ou como de leitura e gravação, dependendo do driver utilizado. Geralmente é utilizada como um disco em pastilhas (chips), SSD - Solid State Disk.

## RAM e DRAM

Foi em algum ponto na década de 50 que surgiram as primeiras ideias de criar uma Memória de Acesso Aleatório (RAM). Apesar disso, nosso papo começa em 1966, ano que foi marcado pela criação da memória DRAM (invenção do Dr. Robert Dennard) e pelo lançamento de uma calculadora Toshiba que já armazenava dados temporariamente.

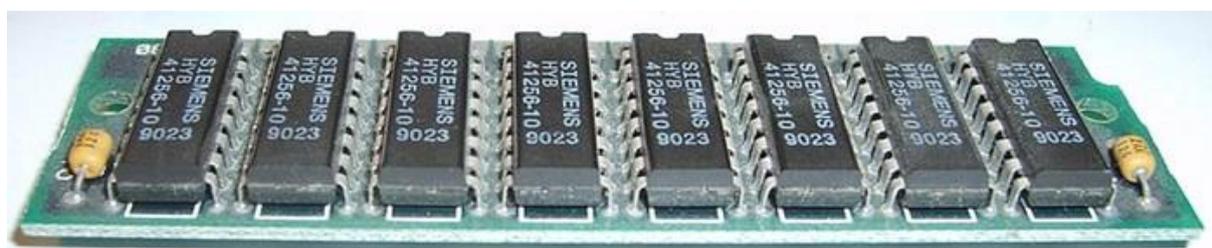
A DRAM (Memória de Acesso Aleatório Dinâmico) é o padrão de memória que perdura até hoje, mas para chegar aos atuais módulos, a história teve grandes reviravoltas. Em 1970, a Intel lançou sua primeira memória DRAM, porém, o projeto não era de autoria da fabricante e apresentou diversos problemas. No mesmo ano, a Intel lançou a memória DRAM 1103, que foi disponibilizada para o comércio “geral” (que na época era composto por grandes empresas).

A partir da metade da década de 70, a memória DRAM foi definida como padrão mundial, dominando mais de 70% do mercado. Nesse ponto da história, a DRAM já havia evoluído consideravelmente e tinha os conceitos básicos que são usados nas memórias atuais.

## DIP e SIMM

Antes da chegada dos antiquíssimos 286, os computadores usam chips DIP. Esse tipo de memória vinha embutido na placa-mãe e servia para auxiliar o processador e armazenar uma quantidade muito pequena de dados.

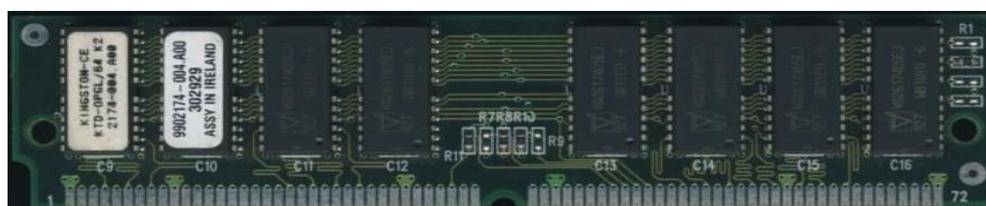
Foi com a popularização dos computadores e o surgimento da onda de PCs (Computadores Pessoais) que houve um salto no tipo de memória. Num primeiro instante, as fabricantes adotaram o padrão SIMM, que era muito parecido com os produtos atuais, mas que trazia chips de memória em apenas um dos lados do módulo.



**Memória SIMM de 256 KB**

## FPM e EDO

A tecnologia FPM (Fast Page Mode) foi utilizada para desenvolver algumas memórias do padrão SIMM. Módulos com essa tecnologia podiam armazenar incríveis 256 kbytes. Basicamente, o diferencial dessa memória era a possibilidade de escrever ou ler múltiplos dados de uma linha sucessivamente.



**Memória EDO**

As memórias com tecnologia EDO apareceram em 1995, trazendo um aumento de desempenho de 5% se comparadas às que utilizavam a tecnologia FPM. A tecnologia EDO (Extended Data Out) era quase idêntica à FPM, exceto que possibilitava iniciar um novo ciclo de dados antes que os dados de saída do anterior fossem enviados para outros componentes.

## DIMM e SDRAM

Quando as fabricantes notaram que o padrão SIMM já não era o suficiente para comportar a quantidade de dados requisitados pelos processadores, foi necessário migrar para um novo padrão: o DIMM. A diferença básica é que com os módulos DIMM havia chips de memórias instalados dos dois lados (ou a possibilidade de instalar tais chips), o que poderia aumentar a quantidade de memória total de um único módulo.



### Memória EDO

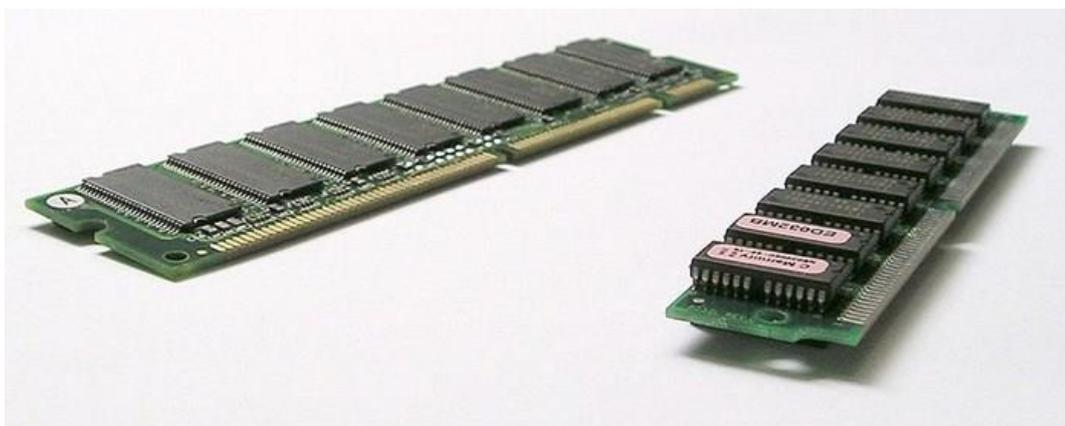
Outra mudança que chegou com as DIMMs e causou impacto no desempenho dos computadores foi a alteração na transmissão de dados, que aumentou de 32 para 64 bits. O padrão DIMM foi o mais apropriado para o desenvolvimento de diversos outros padrões, assim surgiram diversos tipos de memórias baseados no DIMM, mas com ordenação (e número) de pinos e características diferentes.

Com a evolução das DIMMs, as memórias SDRAM foram adotadas por padrão, deixando para trás o padrão DRAM. As SDRAMs são diferentes, pois têm os dados sincronizados com o barramento do sistema. Isso quer dizer que a memória aguarda por um pulso de sinal antes de responder. Com isso, ela pode operar em conjunto com os demais dispositivos e, em consequência, ter velocidade consideravelmente superior.

## RIMM e PC100

Pouco depois do padrão DIMM, apareceram as memórias RIMM. Muito semelhantes, as RIMM se diferenciavam basicamente pela ordenação e formato dos pinos. Houve certo incentivo por parte da Intel para a utilização de memórias RIMM, no entanto, o padrão não tinha grandes chances de prospectiva e foi abandonado ainda em 2001.

As memórias RIMM ainda apareceram no Nintendo 64 e no Playstation 2 – o que comprova que elas tinham grande capacidade para determinadas atividades. Ocorre que, no entanto, o padrão não conseguiu acompanhar a evolução que ocorreu com as memórias DIMM.



**Memória PC133 e EDO**

O padrão PC100 (que era uma memória SDR SDRAM) surgiu na mesma época em que as memórias RIMM estavam no auge. Esse padrão foi criado pela JEDEC, empresa que posteriormente definiu como seria o DDR. A partir do PC100, as fabricantes começaram a dar atenção ao quesito frequência. Posteriormente, o sufixo PC serviu para indicar a largura de banda das memórias (como no caso de memórias PC3200 que tinham largura de 3200 MB/s).

### **DDR, DDR2 e DDR3**

Depois de mais de 30 anos de história, muitos padrões e tecnologias, finalmente chegamos aos tipos de memórias presentes nos computadores atuais. No começo, eram as memórias DDR, que operavam com frequências de até 200 MHz. Apesar de esse ser o clock efetivo nos chips, o valor usado pelo barramento do sistema é de apenas metade, ou seja, 100 MHz.

Assim, fica claro que a frequência do BUS não duplica, o que ocorre é que o dobro de dados transita simultaneamente. Aliás, a sigla DDR significa Double Data Rate, que significa Dupla Taxa de Transferência. Para entender como a taxa de transferência aumenta em duas vezes, basta realizar o cálculo:

Do padrão DDR para o DDR2 foi um pulo fácil. Bastou adicionar alguns circuitos para que a taxa de dados dobrasse novamente. Além do aumento na largura de banda, o padrão DDR2 veio para economizar energia e reduzir as temperaturas. As memórias DDR2 mais avançadas alcançam clocks de até 1.300 MHz (frequência DDR), ou seja, 650 MHz real.



**Memórias DDR1**

### 4.3 Hierarquia de Memória

O termo Hierarquia de memória se refere a uma classificação de tipos de **memória** em função de desempenho. Essa classificação geralmente segue duas dimensões: tamanho e velocidade de acesso, sendo tradicionalmente representada por uma pirâmide.

Para o correto e eficaz funcionamento da manipulação das informações (instruções de um programa e dados) de e para a memória de um computador, verifica-se a necessidade de se ter, em um mesmo computador, diferentes tipos de memória. Para certas atividades, por exemplo, é fundamental que a transferência de informações seja a mais rápida possível. É o caso das atividades realizadas internamente no processador central, onde a velocidade é primordial, porém a quantidade de bits a ser manipulada é muito pequena (em geral, corresponde à quantidade de bits necessária para representar um único valor - um único dado).

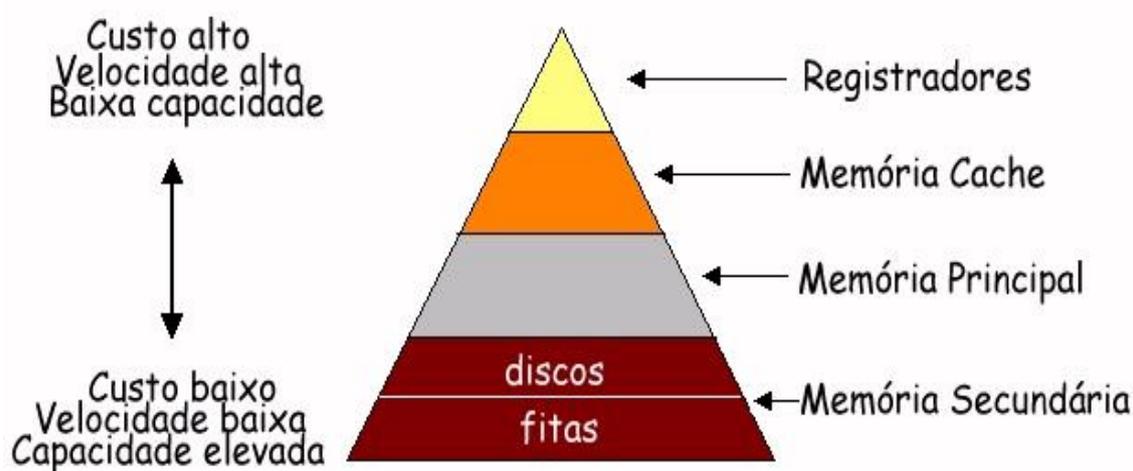
Isso caracteriza um tipo de memória diferente, por exemplo, daquele em que a capacidade da memória (disponibilidade de espaço para guardar informações) é mais importante que a sua velocidade de transferência.

Ainda em relação ao tipo de alta velocidade e pequena quantidade de bits armazenáveis, que se usa na CPU, existem variações decorrentes do tipo de tecnologia utilizada na fabricação da memória.

Devido a essa grande variedade de tipos de memória, não é possível implementar um sistema de computação com uma única memória. Na realidade, há muitas memórias no computador, as quais se interligam de forma bem estruturada, constituindo um sistema em si, parte do sistema global de computação, podendo ser denominado subsistema de memória.

Esse subsistema é projetado de modo que seus componentes sejam organizados hierarquicamente, conforme mostrado na estrutura em forma de pirâmide da figura abaixo.

A pirâmide em questão é projetada com uma base larga, que simboliza a elevada capacidade, o tempo de uso e o custo do componente que a representa.



### Hierarquia de Memória

É comum representar-se a hierarquia de memória de um computador por uma pirâmide. A variação crescente dos valores de certos parâmetros que caracterizam um tipo de memória pode ser mostrado no formato inclinado de uma pirâmide.

#### 4.4 Memória Cache e Memória Virtual

**Memória cache ou simplesmente cache** é usado para armazenar dados acessados com frequência, a fim de acessar rapidamente os dados sempre que for necessário. Ambos são conceitualmente a mesma coisa, porém diferem principalmente em matéria de execução, que resulta em diferentes aspectos, como velocidade e controle de mecanismo.

**Memória virtual é usada para ocultar a informação da memória física real do sistema.** Estende-se a memória disponível do computador, armazenando as partes inativas do conteúdo RAM em um disco. Memória virtual cria uma ilusão de que um usuário tem um ou mais espaços de endereços contíguos que começam com endereço zero.

**Memória Cache armazena os dados desses endereços da memória principal que são usados várias vezes.** Ele pode estar disponível como uma parte reservada da memória principal ou existir como um dispositivo de armazenamento de alta velocidade independente. Memória cache pode ser acessada em um ritmo muito rápido do que em comparação com a memória principal normal.

**Diferença entre memória virtual e cache.** Memória virtual é uma abstração da memória principal. Estende-se a memória disponível do computador, armazenando as partes inativas do conteúdo em um **disco RAM**.

#### 4.5 Organização das Caches

Memórias cache são fisicamente organizadas através de linhas ou blocos de dados que podem organizados em conjuntos (*sets*). Essas formas de organização são descritas abaixo:

##### Mapeamento Direto

No mapeamento direto, cada bloco da memória principal é mapeado para uma linha do cache, cada linha da cache possui 3 campos, *índice(ou linha)*, *tag* e o *endereço da palavra*. O *tag* é usado para validar se a linha procurada é a mesma que está na cache. O *índice* serve como um endereço da cache, apontado aonde pode estar a

linha procurada. O *endereço da palavra* são os bits menos significativos que identificam uma palavra dentro de um bloco da memória Principal. No início do mapeamento, cada bloco da memória principal tem sua linha exclusiva mapeada na cache. Este mapeamento é facilmente implementado. Tudo começa com o endereço da Memória Principal. Através deste endereço, para o acesso a cache, deve se identificar os três campos citados acima. Para achar o End da palavra, é necessário fazer uma simples conta de logaritmo, tendo em base a quantidade de bytes por linha da cache.  **$\text{LOG}_2 \text{Nr bytes}$** . Para achar o índice, deve levar em conta o número de linhas que possui a cache, fazendo o mesmo cálculo de logaritmo:  **$\text{LOG}_2 \text{Na Linhas}$**  O restante será designado a tal.

Suponhamos, em um exemplo que o endereço da linha seja de 32 bits, e a cache possui 1024 linhas com 64 Bytes cada linha.  $\text{Log}_2 64 = 6 \text{ bits para End}$ .  $\text{Log}_2 1024 = 10 \text{ bits para índice}$ .  $32 - (10 + 6) = 16 \text{ bits para tal}$ .

Na operação de leitura, a CPU envia um endereço de 32 bits para o circuito de controle da cache que já identifica os campos para começar a pesquisa nas linhas. Inicialmente começando pelo índice, onde a busca ocorre para tentar achar a linha desejada. Após, o próximo passo é comparar os valores da tal do endereço com o valor da tal da linha. Se forem iguais, em seguida, a palavra que está na linha (6 últimos bits) é transferida pra CPU. Caso os valores das estas não forem iguais, isso significa que a linha desejada pela CPU não está na cache, portanto é buscada na Memória Principal e transferida para a mesma linha, substituindo-a.

### **Totalmente associativa**

No mapeamento direto, cada linha da memória principal tinha um lugar exclusivo na memória cache. Já no mapeamento associativo não, aqui, cada linha da MP pode ser carregada em qualquer lugar da cache. Neste modo haverá apenas dois campos. A tal e o end da palavra. Estes dois campos têm as mesmas funções dos dois campos do mapeamento direto. Porém o campo tal terá uma quantidade de bits maior. No mapeamento associativo, para verificar se a linha está na cache, é comparado a tag de cada linha na cache com a tag do endereço apresentado pela

CPU. Se achou, então os bytes da palavra são transferidos para a CPU, caso contrário, busca-se o endereço na MP e guarda na cache.

### **Associatividade por Conjunto (N-Way)**

Este tipo de organização intercala as vantagens dos outros dois mapeamentos citados acima, acabando com a comparação exaustiva dos campos tag (causado no mapeamento totalmente associativo) e o problema de conflito de endereços por uma mesma linha na cache (causado no mapeamento direto). Neste caso, a cache nada mais é do que uma serie de conjuntos, constituídos por uma série de linhas em cada conjunto. A sigla N-Way significa quantos conjuntos tem a cache, onde N é a quantidade de conjuntos.

Cada linha da memória principal pode ser armazenada em qualquer linha de um conjunto específico. O conjunto é determinado pelo endereço, que é dividido em 3 campos: *TAG*, *Nr do conjunto* e o *End da palavra*. Dado um endereço a cache para leitura, ela separa os endereço nesses três campos. O primeiro campo a ser analisado é o Nr do conjunto, que define em qual conjunto a linha vai ser pesquisada. Após, o campo TAG é comparado com todas as linhas deste conjunto para achar a linha desejada.

### **4.6 Mapeamento de memória cache**

**Mapeamento direto:** Do Inglês: "Direct Mapped". Cada palavra deve ser armazenada em um lugar específico na cache, o qual depende do seu endereço na memória principal. Normalmente utiliza-se a seguinte fórmula para mapeamento da cache:

Endereço na cache = (endereço na MP) MOD (tamanho da cache)  
O endereço é dividido em duas partes: Tag e Índice.

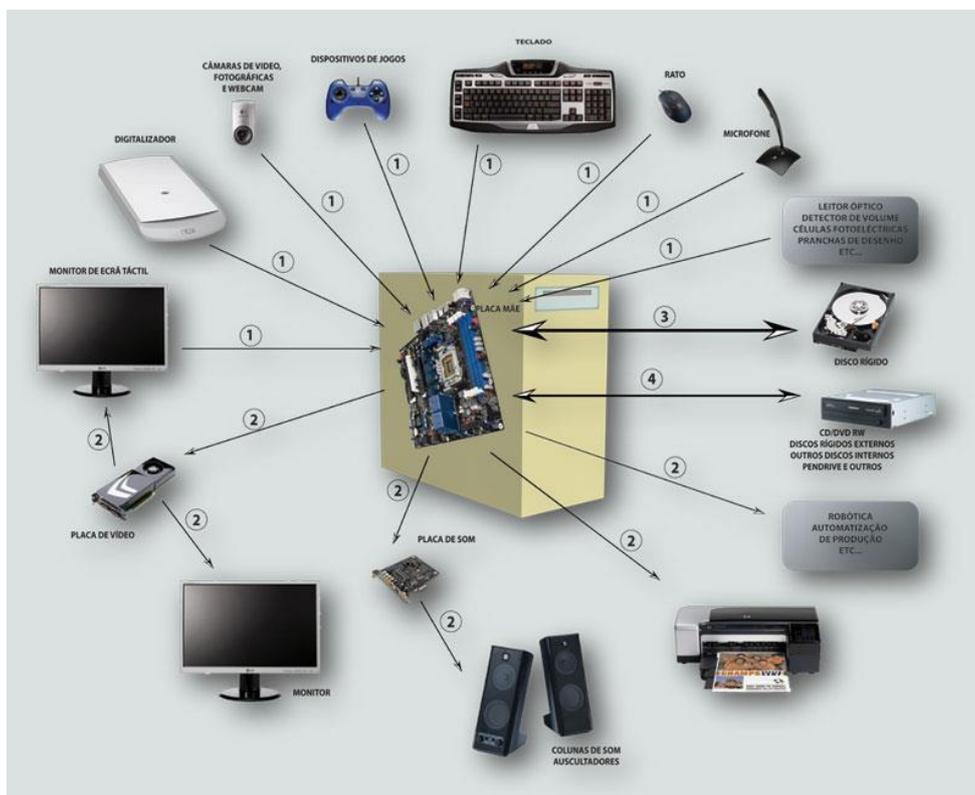
- O **Índice** é usado como endereço na cache. Indica a posição da cache onde pode estar armazenada a palavra.
- O **Tag** é usado para conferir se a palavra que está na cache é a que está sendo procurada, uma vez que endereços diferentes, com o mesmo índice serão mapeados sempre para a mesma posição da cache.

- Vantagens:
  - Não há necessidade de algoritmo de substituição.
  - Hardware simples e de baixo custo.
  - Alta velocidade de operação.
- Desvantagens:
  - Desempenho cai se acessos consecutivos são feitos a palavras de mesmo índice.
  - Hit ratio inferior ao de caches com mapeamento associativo.

## CAPÍTULO 5

### DISPOSITIVOS DE ENTRADA E SAÍDA

Chamamos de dispositivos de entrada e saída aos dispositivos encarregados de incorporar e extrair informação de um computador. Eles se enquadram dentro da denominada Arquitetura de Von Neumann, que explica as principais partes de um computador. Estes dispositivos evoluíram com o tempo, existindo na atualidade muitas variantes que no início da informática pareciam impossíveis.



## 5.1 Dispositivos e Controladoras de Entrada/Saída

As unidades de e/s constituem de um componente eletrônico e um mecânico. O elemento eletrônico é chamado de controlador de dispositivo ou adaptador. Nos computadores pessoais, o controlador de dispositivo aparece em forma de uma placa de circuito impresso. Nessa placa, tem um conector que pode ser plugado outros dispositivos. (Se for uma interface padrão, entre o dispositivo e o controlador), ou seja, deve ter uma interface baixa entre o controlador e um dispositivo.

- Cada controlador de dispositivo tem seus registradores. Esses registradores são usados para comunicar com a **CPU**. Por meio da escrita nesses registradores do controlador de dispositivo, o S.O pode comandar o dispositivo para aceitar, executar, desligar.
- A partir da escrita nesses registradores, o S.O pode saber o estado de um dispositivo, se ele está apto a receber um novo comando, etc. Além dos registradores, os dispositivos têm buffers, no qual o S.O lê e escreve.

### Hardware:

- Dispositivos de Entrada;
- Dispositivos de Saída;
- Dispositivos de Entrada/Saída (Mistos);
- Dispositivos de Armazenamento.

### Dispositivos de Entrada

- Os dispositivos de entrada permitem a comunicação no sentido do utilizador para o computador onde o utilizador pode controlar ou mesmo interagir com este.

### Exemplos:

- Teclados - Os teclados são dispositivos que permitem digitar dados ou instruções para o computador.



Mouses - Permitem deslocar no ecrã o ponteiro e realizar a introdução de ordens para o computador, através da emissão de sinais eléctricos.



JoyStick - Os joysticks são dispositivos utilizados essencialmente para jogar, no entanto podem assumir uma grande variedade de formas e funções, podendo controlar uma aplicação multimídia.



TouchPads - Os Touchpads são dispositivos que substituem os ratos nos portáteis. A maioria dos atuais têm quase todas as funções que podem ser desempenhadas pelos mouses.



### Dispositivos de Saída

- Os dispositivos de saída permitem a comunicação no sentido do computador para o utilizador.

### Exemplos:

- Monitores - Os monitores são dispositivos que constituem o principal meio de comunicação entre o computador e o utilizador. Apresentando várias características diferentes, como, as dimensões, resolução e frequência.



Placas Gráficas - Estes dispositivos são responsáveis pela interligação do monitor com o processador. Podemos distinguir as placas gráficas, de acordo com as suas características, tais como, o número de cores, a resolução e a capacidade de memória.



Impressoras - São dispositivos que permitem imprimir os resultados das operações de processamento do computador. Cada impressora tem as suas próprias características, associadas ao número de página que imprimem por minuto (ppm), a tecnologia da impressão (Laser, Jacto, etc...) e a resolução ou pontos por polegada (dpi). Podendo estas também ter scanners incorporadas.



### Dispositivos de Entrada/Saída (Mistos)

- Os dispositivos de entrada/saída permitem a comunicação em ambos os sentidos do computador para o utilizador e vice-versa.

### Exemplos:

- Placas de Som - As placas de som são dispositivos que suportam áudio digital e MIDI, aumentando a capacidade de gravar e reproduzir sons a partir do computador. Com estas podemos ligar ao computador, microfones, altifalantes, unidade de leitura de CD, entre outros.



Dispositivos de Ligação a Redes - Estes dispositivos permitem ligar um computador a uma rede de computadores. Temos, como exemplo, as placas de rede, os modems e dispositivos bluetooth.



Touch Screens - São ecrãs sensíveis ao toque onde nos é apresentada a informação. São de fácil utilização sendo muito utilizados em postos de venda, vindo assim substituir a utilização do mouse.



Placas de captura de TV - São dispositivos que permitem a sintonia de sinais TV e rádio. Tem como função converter sinais analógicos em digitais, para poderem ser processados pelo computador.



## 5.2 Métodos de Entrada e Saída

## E/S por Programa

Inicialmente a comunicação entre UCP e periféricos era controlada por um conjunto especial de instruções de E/S executadas pela própria UCP. Com as controladoras (ou interfaces), a UCP podia solicitar o início da transferência de dados à controladora e depois a UCP ficava testando o estado do dispositivo para verificar se a operação de E/S terminara. Assim, a UCP ficava num loop de espera e teste ("busy wait").

Outra forma foi liberar a UCP para outras tarefas e, a intervalos regulares, testar o dispositivo ("polling"). Este esquema viabilizou os primeiros sistemas multiprogramáveis, pois no intervalo de tempo entre duas varreduras consecutivas do dispositivo (entre dois pollings) a UCP ficava livre para executar outras tarefas. O polling ainda hoje é usado, por exemplo, em linhas de comunicação serial.

São 2 tempos de espera:

- **Polling** - é o questionamento ou pedido de acesso que a UCP faz à *interface* de E/S para saber da disponibilidade do dispositivo
- **Transferência** dos dados

Podemos imaginar a situação, fazendo uma analogia com um telefone que não tivesse campainha. Seria necessário tirar o fone do gancho periodicamente, para ver se alguma comunicação está em curso. No entanto, tal como o telefone, há outro meio de resolver esse problema: usar um "sinal de chamada", o que equivale a solicitar ao dispositivo que execute uma tarefa e que avise quando tiver terminado; enquanto isto, o processador poderia fazer outras coisas.

## E/S com Interrupção

Neste esquema, a UCP emite a instrução de E/S para a *interface* e desvia sua atenção para executar outra tarefa. A *interface* verifica o estado do dispositivo e, quando este estiver pronto para iniciar a transferência de dados com a UCP, a *interface* envia um "aviso" (*interrupção*) para a UCP, informando que o periférico está "pronto para transmitir".

Neste caso, a UCP só ficará ociosa durante o tempo de transferência dos dados. No entanto se o programa depender dessa transferência para, por exemplo, executar a próxima instrução, a UCP terá que esperar tal como no item 1.

### **Observações:**

- Interrupções são modificações no fluxo de controle causadas por uma ação externa, geralmente relacionada à Entrada ou Saída. Uma interrupção é um sinal de controle enviado por um agente externo (um dispositivo) ao microprocessador, quando um determinado evento é detectado. A interrupção é um sinal de hardware.

Este mecanismo força o processador a tratar o evento externo.

A detecção de uma interrupção faz com que o processador transfira o controle para uma rotina de tratamento de interrupção ("interrupt handler").

A rotina de tratamento de interrupções faz o processador executar as seguintes ações:

- Detectar a fonte da interrupção (o dispositivo que interrompeu),
- Executar as ações apropriadas (que dependem do dispositivo) e finalmente
- Retornar ao ponto do programa em que estava quando iniciou o atendimento à interrupção.

### **Interrupções de Software ("Traps" Ou "Exceptions")**

Interrupções de software são devidas a:

- Algum evento gerado pela execução de uma instrução, como por exemplo uma divisão por zero, overflow, código de operação inválido, tentativa de acesso a uma área de memória protegida ou inexistente, etc ou
- A um evento programado.

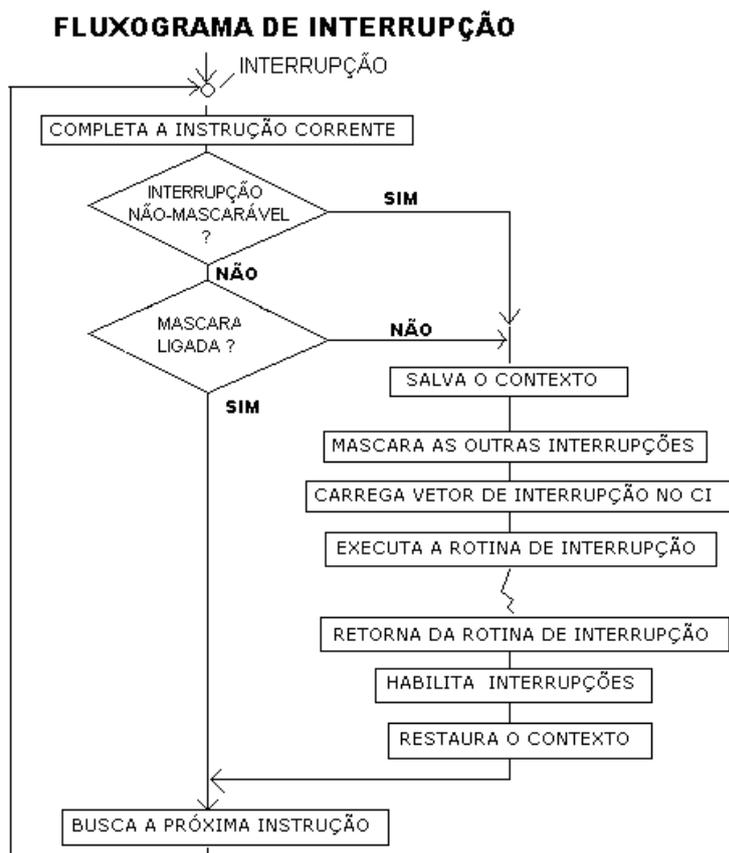
### **Prioridades e Mascaramento das Interrupções**

Pode ocorrer que um ou mais dispositivos enviem sinal de interrupção

simultaneamente ao processador. Neste caso, o processador atenderá primeiramente ao dispositivo de mais alta PRIORIDADE.

São prioritárias para atendimento as interrupções devidas a:

- Emergências de *hardware*, tais como atendimento a *reset* (reinicialização) e erros de *hardware* (erro de paridade de memória, etc.);
- Eventos de alta prioridade
- E/S de dispositivos de alta velocidade
- Durante o atendimento a uma interrupção, o processador não poderá simultaneamente atender a outra interrupção. Para evitar essa situação, existe o recurso de habilitar / desabilitar as interrupções, de forma que o processador, ao iniciar o atendimento a uma interrupção, **DESABILITA** as demais interrupções e, ao terminar o atendimento, volta a habilitar todas as interrupções. Algumas interrupções não são MASCARÁVEIS (isto é, não podem ser desabilitadas), tais como os erros de *hardware* e *reset*.



## **DMA ou Acesso Direto à Memória (implementada na controladora)**

Nesse estágio da tecnologia, um problema permanecia insolúvel: toda transferência de dados entre memória e periféricos exigia a intervenção da UCP. A técnica de DMA (Acesso Direto à Memória ou, em inglês, *Direct Memory Access*) permite que dados sejam transferidos entre a memória e periféricos, ocorrendo intervenção da UCP apenas no início e ao final da transferência.

### **Considerações**

Quando a UCP deseja ler ou gravar algum bloco de dados em um dispositivo de E/S, a UCP envia um comando específico para o controlador DMA. Este comando contém:

- Tipo de tarefa a ser efetuada (E/S)
- Endereço do dispositivo desejado
- Endereço inicial da MP onde o bloco de dados será lido ou gravado
- Quantidade de *bytes* a serem lidos ou gravados (tamanho do bloco de dados)

Durante uma transferência DMA, a UCP fica liberada para efetuar outras tarefas. Ao completar a operação de transferência, o controlador DMA envia um sinal para a UCP (uma interrupção) avisando que a transferência foi completada e os dados estão disponíveis. A área de memória utilizada pelo controlador na técnica de DMA é chamada '*buffer*'.

O controlador DMA pode assumir temporariamente o controle do barramento do sistema. Para isso, será necessário que a UCP suspenda o acesso ao barramento. O barramento de dados do sistema é usado para transferir instruções e dados, tanto pela UCP quanto pelos controladores de E/S. Desta forma, quando em um determinado instante existir concorrência entre um dispositivo DMA e a UCP pela utilização do barramento, será necessário que um controlador (chamado arbitrador ou árbitro de barramento) controle o acesso ao barramento, determinando qual dispositivo terá prioridade para acessar o barramento.

Ao contrário do que poderia parecer óbvio, a prioridade geralmente é para dispositivos como a unidade de disco magnético, porque sendo um dispositivo com partes móveis que não podem ser paradas, submetê-los a espera poderia em alguns casos causar perda de dados. Se a UCP em um dado instante não tem concorrência pelo acesso ao barramento, todos os ciclos do barramento podem ser usados pela UCP para acesso à memória.

No entanto, quando algum dispositivo estiver requerendo acesso ao barramento, ocorrerá roubo de ciclos do barramento para possibilitar a transferência dos dados do dispositivo, o que ocasiona perda de desempenho do computador.

## CAPÍTULO 6

### CPU

A unidade central de processamento ou CPU (*Central Processing Unit*), também conhecido como processador, é a parte de um sistema computacional, que realiza as instruções de um programa de computador, para executar a aritmética básica, lógica, e a entrada e saída de dados. A CPU tem papel parecido ao cérebro no computador. O termo vem sendo usado desde o início de 1960. A forma, desenho e implementação mudaram drasticamente desde os primeiros exemplos, porém o seu funcionamento fundamental permanece o mesmo.

As primeiras CPUs personalizadas foram concebidas como parte de um computador maior. No entanto, este método caro de fazer CPUs personalizadas para uma determinada aplicação rumou para o desenvolvimento de processadores produzidos em massa que são feitas para um ou vários propósitos. Esta tendência de padronização em geral começou na época de discretos minicomputadores e mainframes transístores e acelerou rapidamente com a popularização dos circuitos integrados (CI).

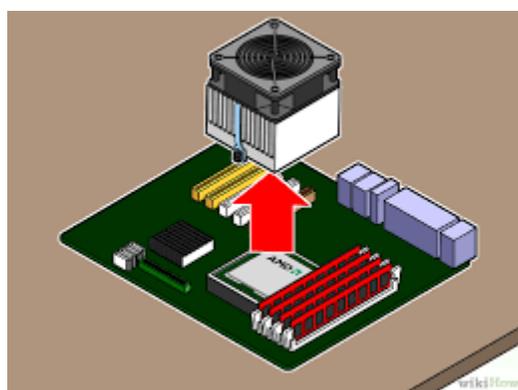
Os CI têm permitido processadores cada vez mais complexos para serem concebidos e fabricados em tamanhos da ordem de nanômetros. Tanto a miniaturização como a padronização dos processadores têm aumentado a presença destes dispositivos digitais na vida moderna, muito além da aplicação limitada

dedicada a computadores. Os microprocessadores modernos aparecem em tudo, desde automóveis até celulares e brinquedos para crianças.

Em máquinas grandes, CPUs podem exigir uma ou mais placas de circuito impresso. Em computadores pessoais e estações de trabalho de pequeno porte, a CPU fica em um único chip de silício chamado de microprocessador. Desde 1970 a classe de microprocessadores de CPUs quase completamente ultrapassado todas as implementações de outra CPU. CPUs modernos são circuitos integrados de grande escala em pequenos pacotes retangulares, com vários pinos de conexão.

Um CPU é composto basicamente, pela maioria dos autores, pelos três seguintes componentes:

- Unidade lógica e aritmética (ULA ou ALU): executa operações lógicas e aritméticas;
- Unidade de controle: decodifica instruções, busca operandos, controla o ponto de execução e desvios;
- Registradores: armazenar dados para o processamento.



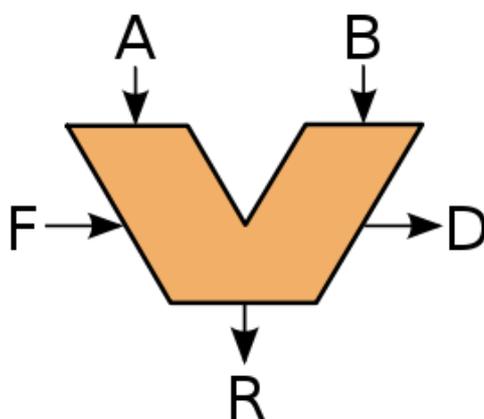
## 6.1 Unidade aritmética e lógica

A unidade lógica e aritmética (ULA) ou em inglês *Arithmetic Logic Unit (ALU)* é um circuito digital que realiza operações lógicas e aritméticas. A ULA é uma peça fundamental da unidade central de processamento (CPU), e até dos mais simples microprocessadores. É na verdade, uma "grande calculadora eletrônica" do tipo desenvolvido durante a II Guerra Mundial, e sua tecnologia já estava disponível quando os primeiros computadores modernos foram construídos.

O matemático John von Neumann propôs o conceito de ULA em 1945, quando escreveu um relatório sobre os fundamentos para um novo computador chamado EDVAC. Pesquisas sobre ULAs ainda são uma parte importante da ciência da computação.

A tecnologia utilizada foi inicialmente relés, herança da telefonia, e posteriormente válvulas, herança da radiofonia. Com o aparecimento dos transistores, e depois dos circuitos integrados, os circuitos da unidade aritmética e lógica passaram a ser implementados com a tecnologia de semi-condutores.

A ULA executa as principais operações lógicas e aritméticas do computador. Ela soma, subtrai, divide, determina se um número é positivo ou negativo ou se é zero. Além de executar funções aritméticas, uma ULA deve ser capaz de determinar se uma quantidade é menor ou maior que outra e quando quantidades são iguais. A ULA pode executar funções lógicas com letras e com números.



Um símbolo esquemático típico para uma ULA, onde "A" e "B" são operandos, "R" é a saída, "F" é a entrada da unidade de controle e "D" é a saída de status.

## 6.2 Unidade de controle e microprogramação

A unidade de controle micro-programada é hoje em dia uma miniatura de computador que pode ser programada com sequência de bits.

A unidade de controle micro-programada é utilizada para se desenvolver a implementação de complexas instruções que não podem ser implementadas e forma de hardware.

O inventor da micro-programação foi Maurice Wilkes em 1953, na Inglaterra. Sua ideia era de criar uma unidade de controle mais fácil de se projetar e mais flexível. A utilização da unidade de controle micro-programada tem algumas vantagens:

- Um computador base, que use a micro-programação pode emular outro computador.
- Uma instrução pode ser desenvolvida e ser utilizada em diferentes modelos de hardware

A IBM reconheceu a importância da unidade de controle micro-programada e implementou no computador IBM 360 em abril de 1964. Todos os modelos desta família de computadores contavam com uma memória de controle rápida e barata, a qual estava micro-programada. Isto se tornou um grande sucesso a família IBM e desde então a micro-programação ficou popular em todo o mundo.

### **Micro-arquitetura**

Estrutura lógica e capacidades funcionais de hardware tal como é visto por um microprogramador.

### **Micro-programação**

É uma estratégia para implementar a unidade de controle de um processador que consiste em programar ao nível das operações de transferência de registros.

### **Unidade de Controle micro-programada**

É um circuito lógico que permite realizar a execução em sequência de um conjunto de instruções muito elementares chamadas de micro-instruções. A Unidade de Controle é formada por:

Memória de Controle (contém as micro-instruções)

Sequenciador - É o encarregado de manter a ordem correta da execução das micro-instruções. Informa o local da próxima micro-instrução a ser executada e guarda no contador de micro-programa. Contador de micro-programa: É o registro que

armazena a localização da próxima instrução a ser executada. Este loca se decodificara em DEC e selecionara uma posição da memória de controle

### Formato de uma Micro-instrução

O micro-programa é uma representação simbólica do controle, que deverá ser traduzida por um programa para uma lógica de controle. O formato da micro-instrução deve ser escolhido de modo a simplificar a representação, tornando o micro-programa mais fácil de entender e de se escrever. Podemos distinguir que uma micro-instrução é inconsistente sempre que um certo sinal de controle tenha que receber dois valores diferentes. Para evitar um formato que permita micro-instruções inconsistentes, podemos fazer com que cada campo da micro-instrução seja responsável por especificar um conjunto de sinais de controle que não se sobreponham. Um exemplo seria: Os primeiros seis campos da micro-instrução controlam o caminho de dados, enquanto o campo de “sequenciamento” especifica como selecionar a próxima micro-instrução.

1º campo	Controle da UAL	Especifica a operação a ser realizada pela UAL neste ciclo de klok. O resultado é sempre escrito em UALSaida
2º campo	SRC1	Especifica a fonte do primeiro operando da UAL
3º campo	SRC2	Especifica a fonte do segundo operando da UAL
4º campo	Controle de registrador	Especifica um leitura ou escrita no banco de registradores, e a fonte do valor a ser escrito.
5º campo	Memória	Especifica uma leitura ou uma escrita na memória, e a fonte. No caso de uma leitura, especifica o registrador de destino.
6º campo	Controle de PC	Especifica a escrita do PC
7º campo	Sequenciamento	Especifica como escolher a próxima micro-instrução a ser executada.

As micro-instruções são geralmente colocadas em uma ROM ou em uma PLA e são utilizadas para implementar o controle.

Podemos atribuir os endereços às micro-instruções, os endereços costumam ser dados em sequência.

Existem disponíveis três métodos para a escolha da próxima micro-instrução: Incrementar o endereço da micro-instrução corrente, para assim obtermos o endereço da próxima.

Desviar para a micro-instrução que começa a execução da próxima instrução. Colocamos um indicador no campo de sequenciamento para indicar qual a próxima instrução.

Escolher a próxima micro-instrução com base na entrada da unidade de controle. Esta escolha também é chamada de “despacho”. Na maioria das vezes as operações de despacho são implementadas a partir da criação de uma tabela que contenha os endereços das micro-instruções alvo. Podemos implementar estas tabelas em uma Rom ou em uma PLA.

**Micro-instrução** - É uma palavra da ROM que contém micro-operações de controle.

**Micro-operação** -É o sinal digital que atua no controle.

**Memória de Controle** - Memória ROM que armazena o micro-programa

**Micro-programa** - É o conjunto de micro-instruções que executa o controle

**Micro-rotina** - É o conjunto de micro-instruções que executa uma função específica no micro-programa.

## **Sequenciador**

É um circuito responsável por informar a localização da próxima micro-instrução que será executada a partir da micro-instrução atual, o indicador de condição e o código da operação do registro de instruções.

Ele é encarregado de direcionar a memória do micro-programa baseando-se no contador de micro-programa. O Contador de micro-programa é um registro que se incrementa com um relógio interno, mas que depende dos registros de estado e

instrução e de alguns bits do registro da micro palavra. Em algumas situações determinadas toma valores preestabelecidos (RESET, INTERRUPT, HOLD, etc).

### Unidade de Controle por Microprograma

Exemplo: microprograma para o computador simplificado

Op-code	Endereço												
Fase: busca	0000			1									
	0001	1			1						1	1	
01	0010		1										
	0011									1	1	1	
	0100					1							
10	0101		1										
	0110									1	1	1	
	0111					1							
	1000						1						
	1001						1						
11	1010		1										
	1011					1					0	1	
		TR	TIR	TPC	IPC	R <sub>A</sub>	W <sub>A</sub>	C	I	R	W	RAV	CS

### 6.3 Execução de Instruções

Cada instrução é executada como uma sequência de três fases:

- 1. acesso à memória (fetch)
- 2. decodificação e
- 3. execução.

A instrução é inicialmente acessada na memória e transferida para o interior da CPU, mais especificamente num registrador especial da unidade de controle chamado de RI ("Registrador de Instrução"). Uma vez no RI, a instrução é interpretada por um circuito decodificador. Finalmente, ela é executada. A sequência apropriada de sinais é gerada pela unidade de controle, resultando nas transferências de dados e operações apropriadas.

A primeira fase é o acesso à instrução ("fetch"): o conteúdo do IP ("Instruction Pointer") é posto no barramento de endereço e a posição de memória é selecionada. Um comando de leitura é então fornecido à memória. Após alguns nanosegundos - correspondentes ao tempo de leitura da memória (ou tempo de acesso) - os oito bits do dado selecionado por esse endereço se tornam disponíveis nos pinos de saída

da memória e são propagados no barramento de dados do sistema (no caso, da direita para a esquerda). O dado é encaminhado para o interior do chip CPU até o registrador de instrução (RI) da unidade de controle, completando assim o ciclo de fetch.

Os dois ciclos seguintes são a decodificação e a execução. O RI é decodificado pela unidade de controle e os sinais apropriados são gerados, resultando na execução da instrução.

Uma instrução não necessariamente tem oito bits de comprimento. De fato, geralmente se usa um, dois, três ou quatro bytes, isto é, 8, 16, 24 ou 32 bits. Quando uma instrução tem mais de oito bits, a unidade de controle deve voltar à memória e ler os bytes seguintes.

O primeiro byte da instrução sempre contém a operação, isto é, o código binário que especifica a operação a ser feita (adição, subtração, desvio, transferência etc.). Após a decodificação do código de operação, a unidade de controle "sabe" se deve ou não voltar à memória para acessar bytes adicionais.

Tendo descrito a sequência de eventos na execução de uma instrução, vamos solucionar o próximo problema, que é o de sequenciamento automático, isto é, a execução sequencial de instruções. Um programa é armazenado como uma sequência de instruções em binário residindo em endereços sucessivos da memória.

Após uma instrução ser acessada e executada, o registrador IP (Instruction Pointer) deve ter sido atualizado. Esse problema é resolvido incrementando-se o IP. Assim, cada vez que o IP é usado (isto é, seu conteúdo é utilizado) seu valor é automaticamente incrementado e carregado no IP. A Próxima vez que o IP for utilizado ele estará apontando para o próximo byte na memória, isto é, para a próxima instrução ou para o próximo byte de uma instrução, se essa instrução usa dois ou mais bytes. O incremento é mostrado como +1 na Figura 11.

As instruções de um programa são armazenadas sequencialmente (uma após outra) na memória. Entretanto, as instruções nem sempre são executadas em sequência. Para tomar diferentes ações baseadas nos resultados de testes ou decisões, partes diferentes do programa podem ser executados. Neste caso, a próxima instrução a

ser executada é especificada pela instrução corrente. Uma tal instrução, que inibe o mecanismo de sequenciamento automático, é chamada de "instrução de desvio". O programador pode fazer um desvio para um endereço de memória especificado. Uma instrução como essa forçará um novo valor no Instruction Pointer.

#### 6.4 Ciclo de execução de uma Instrução

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

A expressão "ciclo de busca e execução" também é muito usada, pois descreve em essência o modo como um computador funciona: a instrução deve ser **buscada** na memória principal e depois executada pela CPU. Deste ciclo emergem todas as funções do computador que são familiares para o usuário final.

Cada CPU de computador pode ter vários ciclos diferentes, baseados em diferentes conjuntos de instruções.

O ciclo de execução de uma instrução é dividido nos seguintes estados:

1. Cálculo do endereço de memória que contém a instrução
2. Busca da instrução
3. Decodificação da instrução
4. Cálculo do endereço dos operandos
5. Busca do operando (Operand Fetch)
6. Execução da operação
7. Armazenamento do resultado em um endereço de memória

#### 6.5 Interrupções

Em Ciência da Computação, uma interrupção é um sinal de um dispositivo que tipicamente resulta em uma troca de contextos, isto é, o processador para de fazer o que está fazendo para atender o dispositivo que pediu a interrupção.

Computadores digitais geralmente oferecem uma maneira de iniciar rotinas de software em resposta a eventos eletrônicos assíncronos. Esses eventos são sinalizados para o processador através de *pedidos de interrupção* (IRQs). O processamento da interrupção compõe uma troca de contexto para uma rotina de software especificamente escrita para *tratar* a interrupção. Essa rotina é chamada rotina de serviço de interrupção, ou tratador de interrupção (*interrupt handler*). Os endereços dessas rotinas são chamados vetores de interrupção e são armazenados geralmente em uma tabela na memória RAM, permitindo sua modificação caso seja necessário. As Interrupções foram concebidas para evitar o desperdício de tempo computacional em *loops* de software (chamados *polling loops*) esperando eventos que serão disparados por dispositivos. Ao invés de ficarem parados esperando o evento acontecer, os processadores tornaram-se capazes de realizar outras tarefas enquanto os eventos estão pendentes. A interrupção avisa ao processador quando o evento ocorreu, permitindo dessa forma uma acomodação eficiente para dispositivos mais lentos.

Interrupções permitem aos processadores modernos responder a eventos gerados por dispositivos enquanto outro trabalho está sendo realizado. Os processadores também oferecem instruções para permitir os processos dispararem *interrupções de software* (*traps*). Isso pode ser usado para implementar uma multitarefa cooperativa.

## 6.6 Pipeline

Pipeline é uma técnica de *hardware* que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada. Estas instruções são colocadas em uma fila de memória dentro do processador (CPU) onde aguardam o momento de serem executadas: assim que uma instrução termina o primeiro estágio e parte para o segundo, a próxima instrução já ocupa o primeiro estágio.

Em resumo, é o processo pelo qual uma instrução de processamento é subdividida em etapas, uma vez que cada uma destas etapas é executada por uma porção especializada da CPU, podendo colocar mais de uma instrução em execução

simultânea. Isto traz um uso mais racional da capacidade computacional com ganho substancial de velocidade. Entre os problemas enfrentados estão a dependência de instruções anteriores e desvios que dificultam o processo, bem como a diferença de complexidade de instruções que fazem com que as mesmas possam levar um tempo variável para execução.

A técnica de pipeline é utilizada para acelerar a velocidade de operação da CPU, uma vez que a próxima instrução a ser executada está normalmente armazenada nos registradores da CPU e não precisa ser buscada da memória principal que é muito mais lenta.

A técnica de pipeline é semelhante a uma linha de produção de fábrica. Cada instrução de um microprocessador passa por diversas fases até sua execução. Estas fases podem ser:

- Decodificação
- Acesso à memória ou aos registradores
- Processamento aritmético

Se conseguirmos separar todas estas fases de forma independente, e separar cada fase por ciclo de relógio teríamos (neste exemplo) 3 ciclos por instrução. Se usarmos uma técnica de pipeline poderíamos colocar 3 instruções ao mesmo tempo no microprocessador (cada uma numa fase distinta) e termos 3 instruções em 3 ciclos (1 instrução por ciclo idealmente). Outros problemas advêm desta técnica, como desvios (como saber as próximas instruções), e dependência de instruções (a próxima depende da anterior). Na prática todos os microprocessadores modernos utilizam-se de várias (dezenas) fases no processamento para usufruir de *clocks* maiores (quanto menor a fase, mais rápido pode ser o ciclo).

Algumas CPUs incluem conceitos muito mais avançados de pipeline:

- Pré-decodificação: a CPU pode iniciar a decodificação de diversas instruções (paralelamente) e antes do momento das mesmas serem executadas.
- Execução fora-de-sequência: algumas CPUs podem além de pré-decodificar, executar previamente um determinado número de instruções. Numa etapa

posterior, a ordem de execução é verificada e os resultados das operações são consolidados na sua ordem correta.

- Previsão de desvio: caso exista uma instrução de desvio dentro do pipeline e a sua execução for consolidada, todas as instruções posteriores a mesma e que se encontram na fila devem ser abortadas.

## 6.7 Operandos

Operando (s) - é o campo da instrução cujo valor binário sinaliza a localização do dado (ou é o próprio dado) que será manipulado (processado) pela instrução durante a operação

Em geral, um operando identifica o endereço de memória onde está contido o dado que será manipulado, ou pode conter o endereço onde o resultado da operação será armazenado. Finalmente, um operando pode também indicar um Registrador (que conterà o dado propriamente dito ou um endereço de memória onde está armazenado o dado). Os operandos na verdade fornecem os dados da instrução.

Obs: Existem instruções que não tem operando. Ex.: Instrução HALT (PARE).

A função do operando é identificar / localizar o dado que será processado. Quanto ao número de operandos, podemos considerar:

- Número maior: permite maior completeza, por possibilitar ter todos os operandos necessários à indicação dos dados para a realização de operação. Em consequência, embora mais operandos ocupem mais memória, será necessária uma menor quantidade de instruções, utilizando menos memória.
- Número menor: as referências ficam menos completas, e serão necessárias mais instruções no programa. No entanto, tendo menos operandos, o espaço ocupado na memória pelos operandos será menor.

Como geralmente o operando contém um endereço da MP, o número de bits ocupado por um operando depende do número de células endereçáveis de memória, ou seja, é preciso saber quantas células podem ser endereçadas na memória principal para saber quantos bits serão necessários para o operando.

## 6.8 Modo de Endereçamento

**Modo Implícito** – Operandos são especificados implicitamente na definição da instrução (instruções de acumulador ou de zero endereços).

**Modo Imediato** – Operandos especificados na instrução propriamente dita. O campo do operando contém o que será de fato usado em conjunção com a operação especificada na instrução.

**Modo Registrador** – O campo endereço especifica um registrador do processador. Os operandos são os valores nos registradores.

**Modo Registrador Indireto** – A instrução especifica um registrador cujo conteúdo fornece o endereço do operando na memória (isto é, o registrador contém um endereço, e não um valor).

**Modo de Auto-incremento e Auto-decremento** – O mesmo que registrador indireto, exceto que o endereço no registrador é incrementado ou decrementado dependendo da operação.

**Modo de Endereçamento Direto** – O endereço efetivo é igual ao endereço contido na instrução. O operando está na memória, e é especificado no campo do endereço.

**Modo de Endereçamento Indireto** – O campo endereço fornece outro endereço na memória, onde o dado está efetivamente armazenado. A unidade de controle coleta a instrução da memória e usa o seu campo de endereço para novamente ler o endereço efetivo.

**Modo de Endereçamento Relativo** – O conteúdo do registrador PC é adicionado ao campo endereço contido na instrução. Este campo pode ser um número positivo ou negativo. O resultado produz um endereço efetivo, relativo à atual (ou próxima) instrução (usada para branching).

**Modo de Endereçamento Indexado** – O conteúdo de um registrador de índice é adicionado ao endereço parte da instrução, para se obter o endereço efetivo. O campo endereço especifica o endereço inicial de um arranjo de dados na memória. Cada operando no arranjo é armazenado na memória relativa ao endereço inicial. A

distância entre o endereço inicial e o endereço do operando é armazenada no registrador de índice.

**Modo de Endereçamento do Registrador Base** – O conteúdo de um registrador base é adicionado ao campo endereço da instrução para se obter o endereço efetivo. Funciona de forma similar ao modo de endereçamento indexado, exceto que o registrador agora é chamado de registrador base ao invés de registrador de índice.

## CAPÍTULO 7

### PROCESSADORES

As arquiteturas de computadores, muitas vezes são pouco exploradas e conhecidas, mas as suas diferenças e peculiaridades são bem interessantes de se estudar e garantem uma melhor compreensão sobre o funcionamento dos processadores desde os primórdios da era da computação. A arquitetura CISC, ou Complex Instruction Set Computer, é uma arquitetura bem complexa sendo capaz de executar várias centenas de instruções complexas diferentes, se tornando extremamente versátil.

Exemplos famosos de processadores CISC são o 386 e o 486. Já a arquitetura RISC, ou Reduced Instruction Set Computer, é um tipo de arquitetura de processador que surgiu começo da década de 80 quando a tendência era construir chips com conjuntos de instruções cada vez mais complexos, os famosos processadores CISC. Foi então que alguns fabricantes tiveram a ideia de seguir o caminho oposto, criando o padrão RISC (Reduced Instruction Set Computer, ou “computador com um conjunto reduzido de instruções”).

#### 7.1 CISC

CISC ou Complex Instruction Set Computer é uma arquitetura de processadores capaz de executar centenas de instruções complexas diferentes o que a torna extremamente versátil. Exemplos de processadores que utilizam essa arquitetura são os 386 e os 486 da Intel. Os processadores baseados na computação de conjunto de instruções complexas contêm uma micro-programação, ou seja, um

conjunto de códigos de instruções que são gravados no processador, permitindo-lhe receber as instruções dos programas e executá-las, utilizando as instruções contidas na sua micro-programação. Seria como quebrar estas instruções, já em baixo nível, em diversas instruções mais próximas do hardware (as instruções contidas no microcódigo do processador).



Tecnologia CISC

Como característica marcante esta arquitetura contém um conjunto grande de instruções, a maioria deles em um elevado grau de complexidade. Algumas características dessa arquitetura são:

- Controle microprogramado;
- Modos registrador-registrador, registrador-memória, e memória registrador;
- Múltiplos modos de endereçamento à memória, incluindo indexação (vetores);
- Instruções de largura (tamanho) variável, conforme modo de endereçamento utilizado;
- Instruções requerem múltiplos ciclos de máquina para execução, variando também com o modo de endereçamento;
- Poucos registradores;
- Registradores especializados.

## 7.2 RISC

RISC (acrônimo de Reduced Instruction Set Computer; em português, "Computador com um conjunto reduzido de instruções") A arquitetura RISC é constituída por um pequeno conjunto de instruções simples que são executadas diretamente pelo

hardware, onde não há a intervenção de um interpretador (microcódigo), o que significa que as instruções são executadas em apenas uma micro-instrução (de uma única forma e seguindo um mesmo padrão). As máquinas RISC só se tornaram viáveis devido aos avanços de software otimizado para essa arquitetura, através da utilização de compiladores otimizados e que compensem a simplicidade dessa arquitetura. Existe um conjunto de características que permite uma definição de arquitetura básica RISC, são elas:

- Utilização de apenas uma instrução por ciclo do datapath (ULA, registradores e os barramentos que fazem sua conexão);
- O processo de carregar/armazenar, ou seja, as referências à memória são feitas por instruções especiais de load/store;
- Inexistência de microcódigo, fazendo com que a complexidade esteja no compilador;
- Instruções de formato fixo;
- Conjunto reduzido de instruções, facilitando a organização da UC de modo que esta tenha uma interpretação simples e rápida;
- Utilização de pipeline ( é uma técnica de dividir a execução de uma instrução em fases ou estágios, abrindo espaço para execução simultânea de múltiplas instruções);
- Utilização de múltiplos conjuntos de registradores.

Ao contrário dos complexos CISC, os processadores RISC são capazes de executar apenas poucas instruções simples, e justamente por isso que os chips baseados nesta arquitetura são mais simples e muito mais baratos.

Uma outra vantagem dos processadores que utilizam essa arquitetura é o fato de terem um menor número de circuitos internos, permite que se trabalhe com clocks mais altos. Um bom exemplo são os processadores Alpha, que em 97 já operavam a 600 MHz.

## Exemplos



Famílias bem conhecidas incluem RISC DEC Alpha, a AMD 29k, ARC, ARM, Atmel AVR, MIPS, PA-RISC, Power (incluindo PowerPC), SuperH e SPARC.

Processadores RISC

## Comparação

Ao contrário da complexa arquitetura CISC, os processadores RISC são capazes de executar apenas algumas poucas instruções simples, o que justifica o fato de os chips baseados nesta arquitetura serem mais simples e muito mais baratos. Outra vantagem dos processadores RISC, é que por terem um menor número de circuitos internos, podem trabalhar com clocks mais altos. Um processador RISC é capaz de executar instruções muito mais rapidamente. A seguir um quadro comparando os recursos de ambas arquiteturas:

RISC	CISC
Múltiplos conjuntos de registradores, muitas vezes superando 256	Único conjunto de registradores, tipicamente entre 6 e 16 registradores
Três operandos de registradores permitidos por instrução (por ex., add R1, R2, R3)	Um ou dois operandos de registradores permitidos por instrução (por ex., add R1, R2)
Passagem eficiente de parâmetros por registradores no chip (processador)	Passagem de parâmetros ineficiente através da memória
Instruções de um único ciclo (ex. load e store)	Instruções de múltiplos ciclos
Controle hardwired (embutido no hardware)	Controle microprogramado
Altamente paralelizado (pipelined)	Fracamente paralelizado
Instruções simples e em número reduzido	Muitas instruções complexas
Instruções de tamanho fixo	Instruções de tamanho variável
Complexidade no compilador	Complexidade no código
Apenas instruções load e store podem acessar a memória	Muitas instruções podem acessar a memória
Poucos modos de endereçamento	Muitos modos de endereçamento

RISC x CISC

### 7.3 Máquinas Superescalares

O termo escalar é utilizado para distinguir os modelos de execução:

- De uma única instrução manipulando operandos discretos;
- Das instruções vetoriais, que desencadeiam a ativação em paralelo de múltiplos elementos de processamento, todos executando uma única operação.

Além destes dois modelos, existe o que executa várias instruções concorrentemente, sem que elas precisem ter o mesmo código de operação (não

vetoriais). Este modelo é utilizado pelos computadores superescalares, assim chamados, por processarem instruções escalares em grande grau de paralelismo.

Computadores superescalares são máquinas projetadas para melhorar a performance de execução de instruções escalares, em outras palavras, eles exploram o paralelismo em baixo nível e, assim, podem executar diversas instruções de máquina simultaneamente, contanto que, todas sejam provenientes do mesmo programa objeto.

O nome contrasta com o propósito dos processadores vetoriais e matriciais, pois, utilizam a técnica de processar várias instruções de um mesmo tipo em paralelo, enquanto que um superescalar pode processar simultaneamente instruções diferentes.

Este debate é feito porque cada máquina é especializada em executar determinados tipos de programas específicos e, como na maioria das aplicações, a massa de operações são quantidades escalares, a "filosofia" superescalar representa o próximo passo na condução dos processadores de propósito geral para a alta performance, que podem então, executar eficientemente uma larga gama de programas de aplicação não se limitando a aplicações vetoriais ou matriciais.

Segundo (Stallings, 2002), numa implementação da arquitetura superescalar de um processador, instruções comuns como: aritmética inteira e de ponto flutuante, leitura e escrita na memória e desvios condicionais, podem ser inicializadas simultaneamente e executadas independentemente e tal implementação aumenta o grau de complexidade relacionada às instruções pipelined.

As implementações da arquitetura superescalar são feitas baseadas, principalmente, na arquitetura RISC, porém, elas podem ser feitas tanto na RISC como na CISC. A primeira afirmação é devida ao fato de que o pequeno conjunto de instruções da arquitetura RISC, já o revela em parte como sendo a técnica superescalar.

Apesar do período de gestação da arquitetura RISC para a decolagem comercial desde o verdadeiro início da pesquisa com o IBM801 e o RISC de Berkeley ter sido de sete a oito anos, a disponibilização da primeira máquina superescalar comercialmente ocorreu em um ou dois anos.

## 7.4 Atualidades

Atualmente não se pode afirmar com 100% de certeza que um processador utiliza apenas a arquitetura CISC ou RISC, pois os modelos atuais de processadores abrigam as características de ambas as arquiteturas. Processadores ARM usados em celulares são um exemplo de uso da arquitetura RISC, outro exemplo de uso dessa arquitetura é em consoles como o Nintendo 64 e o Playstation.



Com o passar dos anos, tanto a Intel quanto a AMD perceberam que usar alguns conceitos da arquitetura RISC em seus processadores poderia ajudá-las a criar processadores mais rápidos. Porém, ao mesmo tempo, existia a necessidade de continuar criando processadores compatíveis com os antigos. A ideia então passou a ser construir chips híbridos, que fossem capazes de executar as instruções x86, sendo compatíveis com todos os programas, mas ao mesmo tempo comportando-se internamente como chips RISC, quebrando estas instruções complexas em instruções simples, que podem ser processadas por seu núcleo RISC.

## REFERÊNCIAS

PATTERSON, David A.; HENNESSY, John L. Organização e Projeto de Computadores. Tradução de 3ª edição. Editora Campus, 2005.

STALLINGS, William. Arquitetura e Organização de Computadores. Tradução da 8ª edição. Editora Prentice Hall Brasil, 2002.

TANENBAUM, Andrew S. Organização Estruturada de Computadores. Tradução da 5ª edição. Editora Prentice Hall Brasil, 2007.

MONTEIRO, Mário A. Introdução à Organização de Computadores, 5ª edição. Editora LTC, 2007.

MORIMOTO, Carlos E. *Manual de Hardware Completo*. 3ª Edição. Junho de 2002. Disponível na Internet em: <http://www.hardware.com.br/livros/hardware-manual/>. Acessado em: 09/04/2012.

VASCONCELOS, Laércio. *Hardware Total*. 1ª Edição. Editora Makron Books, 2002.

Sites consultados:

[Processadores: clock, bits, memória cache e múltiplos núcleos \(Parte 1\)](#)

[Microinformática – Hardware – Processador](#)

[Microinformática – Hardware – Memória](#)

[Microinformática – Hardware – Dispositivos de entrada e/ou saída](#)

[Iniciação à Informática \(Disciplina de Informática Médica – FMUSP\)](#)

<http://www.cultura.ufpa.br/dicas/progra/arq-uni.htm>

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

<http://www.juliomoraes.com/pt/2008/07/idades-de-medida-do-sistema-binario-dos-computadores/>

<http://www.sistemasembarcados.org/2015/11/15/processadores-arquitetura-risc-e-cisc/>

<http://www.hardware.com.br/termos/risc>

<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1259>

<http://www.hardware.com.br/artigos/risc-cisc/>

<http://riscv.org/>

<https://pt.wikipedia.org/wiki/CISC>