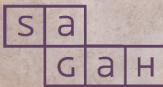


# ARQUITETURAS DE COMPUTADORES

Sandra Rovena Frigeri



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS

# **Os principais componentes de um computador**

## **Objetivos de aprendizagem**

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Explicar o funcionamento de um computador.
- Identificar os níveis hierárquicos de um sistema computacional.
- Reconhecer os principais componentes de um computador por meio do modelo de Von Neumann.

## **Introdução**

Um sistema é um conjunto de elementos interconectados que formam um todo organizado, que possui um objetivo e determinadas funções que deve realizar. O sistema computacional é formado por um conjunto de componentes, que são interconectados/inter-relacionados e cumprem a função de realizar o processamento de dados por meio de um computador. Deve-se ter em mente que um sistema computacional é formado pelo computador (*hardware*), pelos sistemas e programas (*software*) e pelos usuários (*peopleware*), todos trabalhando de forma integrada.

Os componentes do computador incluem dispositivos de entrada e saída de dados (teclado, tela, etc.), meios de armazenamento de informações voláteis ou não (memórias, discos rígidos, etc.) e estruturas responsáveis pelo processamento de dados. Assim, um computador é uma máquina que recebe dados a partir de dispositivos de entrada, armazena-os e os transforma por meio da execução de programas, gerando novas informações/novos resultados, que são enviados a dispositivos de saída ou transmitidos/armazenados. Dessa forma, os principais componentes de um computador são: processador, memória e dispositivos de entrada/saída. Todo o trabalho realizado pelo computador é controlado pelo processador, que é o responsável pela execução de instruções — comandos —, que realizam ações no computador.

Os profissionais da área de computação não devem considerar o computador como apenas uma caixa fechada, ignorando sua estrutura interna. É preciso conhecer seus componentes, suas características, seu desempenho e suas interações. Esse conhecimento é importante para entender como fazer o melhor uso dos recursos computacionais, compreender suas limitações, além de poder selecionar o melhor tipo de *hardware* para determinadas aplicações/necessidades.

Neste capítulo, você vai estudar o funcionamento dos computadores, identificando as estruturas que os compõem e conhecendo suas características e inter-relações. Você vai verificar que os computadores possuem uma estrutura hierárquica, o que permitirá a compreensão da sua organização. Além disso, você vai analisar o modelo de Von Neumann, de grande importância para a identificação dos principais elementos de um computador.

## Funcionamento do computador

O computador contém milhões de componentes eletrônicos e é considerado um sistema complexo. Para que se possa descrever seus componentes, é necessário conhecer sua natureza hierárquica. A partir dela, torna-se possível considerar os componentes do computador desde um nível mais externo, que é onde estão as partes mais visíveis — dispositivos de entrada/saída de dados, meios de armazenamento de dados, meios de comunicação de dados e elementos de processamento — até elementos que podem ser invisíveis a olho nu, como os componentes do processador. Dessa forma, pode-se considerar níveis particulares da hierarquia de um sistema, identificar seus elementos e suas inter-relações, além de conhecer suas funções individuais e como estas contribuem para as funções do componente superior na hierarquia. Normalmente, quando se trabalha em determinado nível, não é necessário conhecer a estrutura do nível mais baixo — é preciso conhecer apenas as funcionalidades e a estrutura do nível específico.

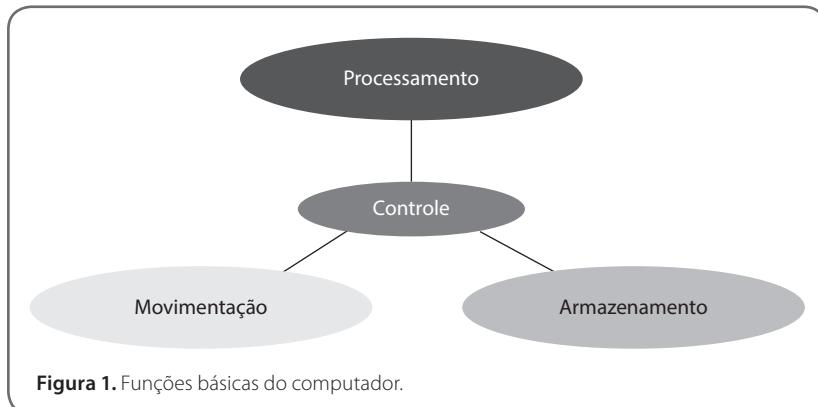
O nível mais próximo do usuário é considerado o **alto nível**, e aquele que representa os componentes eletrônicos é o **baixo nível**. Então, os níveis de um sistema computacional podem ser:

- **Nível do usuário:** formado pelas partes visíveis do computador e programas utilizados diretamente pelo usuário.

- **Nível da linguagem de alto nível:** é onde trabalham a maioria dos programadores que utilizam linguagens de alto nível para o desenvolvimento de aplicações.
- **Nível da linguagem de montagem (linguagem de máquina):** é onde o processador interpreta e executa as instruções, que estão em linguagem de montagem ou *Assembler*.
- **Nível de controle:** é de onde a unidade de controle (componente do processador) envia sinais de controle para realizar transferências de dados entre registradores, memória e outros componentes, além de selecionar operações na unidade lógica e aritmética (ULA).
- **Nível de unidades funcionais:** é onde estão os registradores, a ULA, a memória e os barramentos.
- **Nível das portas lógicas:** é onde fica a estrutura lógica dos componentes do computador, que são projetados usando portas lógicas.
- **Nível de transistores e fios:** é o nível mais baixo do computador, onde estão componentes eletrônicos e fios usados para implementar as portas lógicas.

Atualmente, os computadores estão cada vez mais potentes e menores, e seus usos são ilimitados. Os dados manipulados pelos computadores podem ter diversas formas e representações, e o espectro de possibilidades de usos e transformações desses dados é amplo. Eles são os elementos fundamentais de trabalho e determinam as funções básicas do computador: processamento de dados, armazenamento de dados, movimentação de dados e controle.

O **processamento de dados** deve permitir transformações de diferentes tipos de dados, gerando novos dados, realizando cálculos, ajustes, etc. Esse é o núcleo do trabalho do computador. Já o **armazenamento de dados** é uma função essencial, pois o computador precisa manter os dados que está utilizando no processamento, mesmo que de forma temporária, mas também precisa guardar dados de forma estável, para que possam ser utilizados posteriormente. Junto com essas funções, a **movimentação de dados** é uma necessidade para que o computador possa trabalhar, pois representa os caminhos e as funções utilizadas para receber dados de entrada, transmitir dados de saída, armazenar dados e recuperar dados armazenados. Finalmente, para garantir a execução correta dessas três funções, é exercido o **controle**, que coordena o trabalho de seus elementos funcionais. O esquema dessas funções pode ser visto na Figura 1.

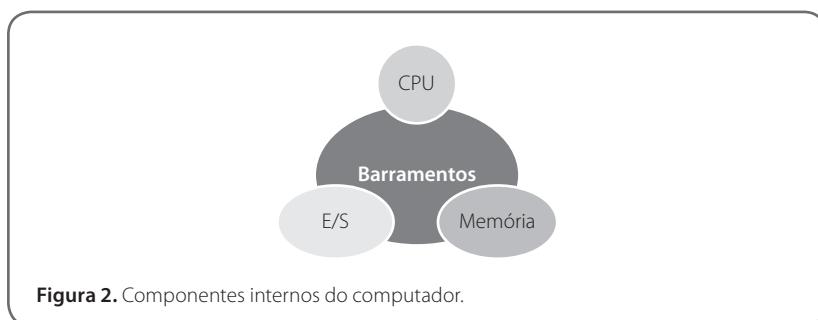


**Figura 1.** Funções básicas do computador.

Essas quatro funções básicas são percebidas tanto nos componentes mais externos do computador como nos internos, que você vai verificar na sequência. Internamente o computador possui os seguintes componentes:

- **Unidade central de processamento (CPU — central processing unit):** responsável pelo controle do computador e por realizar o processamento de dados; é comumente denominada apenas de processador.
- **Memória principal:** tem a função de armazenar dados.
- **Entrada/Saída (E/S):** controladores responsáveis pela movimentação de dados entre o computador e o meio externo.
- **Interconexão:** mecanismo de comunicação entre CPU, memória e E/S, normalmente um barramento, que consiste em uma série de fios condutores.

A estrutura dos componentes básicos do computador é apresentada na Figura 2.



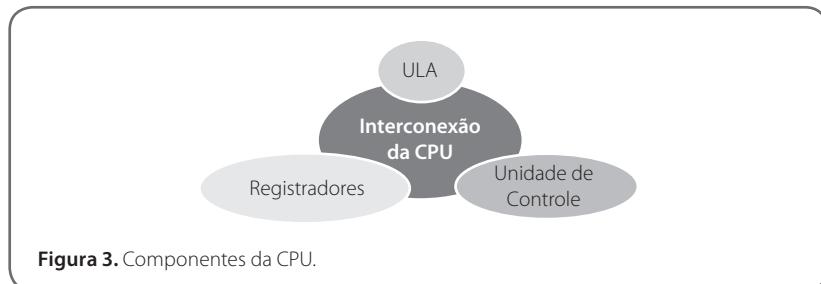
**Figura 2.** Componentes internos do computador.

Nos primeiros computadores, havia apenas um componente interno de cada tipo, mas, nos últimos anos, temos computadores com diversos processadores, além da replicação de outros componentes, ampliando cada vez mais suas capacidades.

A parte mais interessante do computador, com certeza, é a CPU, pois ela é o núcleo do processamento, e muitos investimentos têm sido realizados para ampliar sua capacidade e suas funcionalidades. Os principais componentes da CPU são:

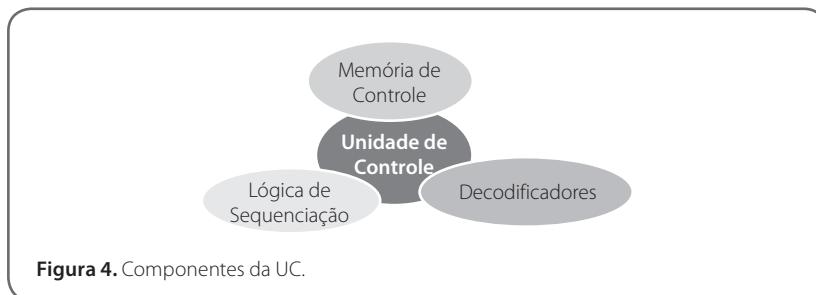
- **Unidade de controle (UC):** é responsável pelo controle da operação da CPU e, logo, pelo controle do computador.
- **Unidade lógica e aritmética (ULA, ou ALU, do inglês *arithmetic and logic unit*):** é responsável por realizar operações lógicas e aritméticas, que são necessárias para o processamento de dados no computador.
- **Registradores:** são responsáveis pelo armazenamento de dados utilizados pela CPU.
- **Interconexões da CPU:** são estruturas de comunicação entre unidade de controle, ALU e registradores, que formam os barramentos.

Veja esses componentes da CPU na Figura 3.



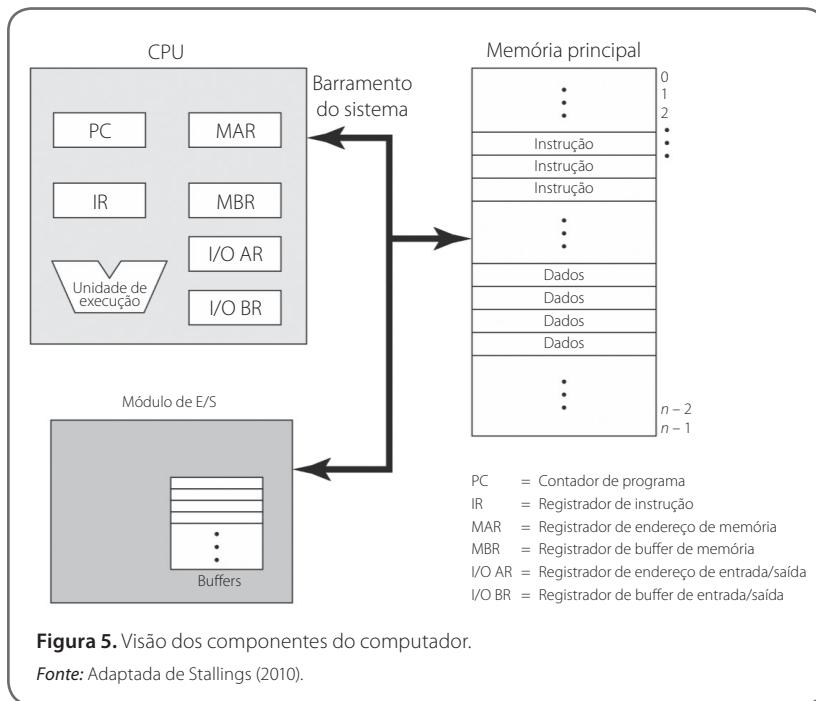
**Figura 3.** Componentes da CPU.

A UC é o componente central da CPU e pode ser implementada utilizando-se diversas técnicas. Uma delas é denominada unidade de controle microprogramada, que utiliza microinstruções. A UC possui uma **memória de controle**, que é formada por um conjunto de registradores que ela utiliza para guardar instruções, endereços de dados, etc. Ela também possui **decodificadores**, que são componentes projetados com circuitos lógicos que recebem o código de uma instrução e o decodificam, ou seja, verificam o que deve ser feito e, então, enviam sinais de controle para sua execução, que é controlada pela **lógica de sequenciação**. Veja a estrutura básica da UC na Figura 4.



**Figura 4.** Componentes da UC.

Pode-se compreender a estrutura hierárquica da organização dos computadores a partir de suas funções básicas (Figura 1), dos componentes básicos do computador (Figura 2), dos componentes da CPU (Figura 3) e dos componentes da UC (Figura 4). Já na Figura 5 você pode observar os principais componentes do computador envolvidos no processamento. No módulo de E/S passam os dados que vêm/vão dos/para os dispositivos de entrada/saída de dados. Essas informações são armazenadas na memória do computador, que também armazena os programas. Estes são conjuntos de instruções que serão executadas pela CPU, que contém um conjunto de registradores (PC, IR, etc.) e a ULA, tudo isso interligado pelos barramentos.



**Figura 5.** Visão dos componentes do computador.

*Fonte:* Adaptada de Stallings (2010).

## Processador



O processamento principal do computador é realizado pela execução de um programa, que é formado por um conjunto de instruções que são armazenadas na memória do computador. O processador é responsável por executar essas instruções, realizando o ciclo da instrução. Esquematicamente, esse processamento consiste em dois ciclos: busca e execução. Esses ciclos são repetidos sequencialmente até o computador ser desligado. Assim, no ciclo de busca, o processador lê instruções da memória, uma de cada vez. Já no ciclo de execução, o processador decodifica e realiza os procedimentos para sua execução, o que pode envolver a busca de dados na memória, a execução de operações na ULA, entre outras operações.

O processador busca uma instrução da memória no início de cada ciclo de instrução. Normalmente, o contador de programa (PC — *program counter*), que é um registrador, possui o endereço da próxima instrução a ser executada. Assim, o PC é incrementado a cada instrução, salvo quando há alguma instrução de desvio. Após a leitura da instrução, ela é carregada no IR (registrador de instrução — *instruction register*, em inglês). A instrução contém uma operação codificada em binário, que indica a ação a ser realizada, que pode ser de uma dessas categorias:

- **Processador-memória:** quando os dados são transferidos do processador para a memória ou vice-versa.
- **Processador-E/S:** quando os dados são transferidos entre um dispositivo periférico de E/S ou módulo de E/S e o processador, ou vice-versa.
- **Processamento de dados:** quando o processador realiza alguma operação aritmética ou lógica com os dados.
- **Controle:** quando uma instrução determina que a sequência de execução seja alterada, para a execução de um desvio, que altera o PC.

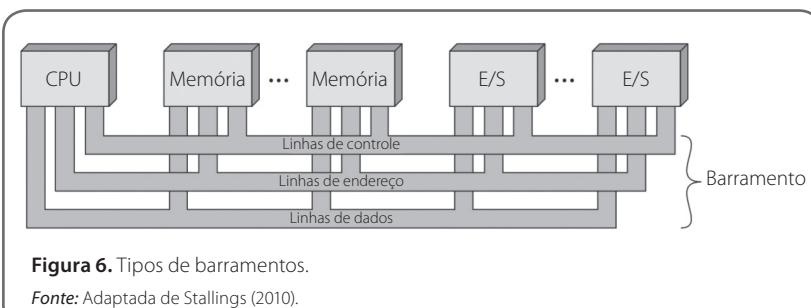
A maioria dos dispositivos externos ao processador são muito mais lentos do que ele; dessa forma, normalmente o processador solicita dados para esses dispositivos e espera seu retorno por meio de interrupções. De forma geral, todos os computadores possuem um mecanismo que permite que os dispositivos de E/S e memória possam interromper o processamento normal do processador. As classes de interrupções são:

- **Programa:** gerada por alguma situação que deriva do resultado da execução de alguma instrução, como o *overflow* aritmético, a divisão

por zero, a tentativa de executar uma instrução de máquina ilegal ou a referência fora do espaço de memória permitido para o usuário.

- **Timer:** gerada com base no *clock* (relógio) do processador; permite que ele realize determinadas funções periodicamente.
- **E/S:** gerada por um controlador de E/S para sinalizar o término normal de uma operação ou para sinalizar uma série de condições de erro.
- **Falha de hardware:** gerada por uma falha como falta de energia ou erro de paridade de memória.

Para a comunicação entre os componentes do processador, utiliza-se barramentos. Um barramento é uma estrutura física que define um caminho que permite a comunicação entre dois ou mais dispositivos; assim, ele é um meio de transmissão compartilhado. Múltiplos dispositivos podem ser conectados a um barramento e compartilham os sinais transmitidos; porém, somente um dispositivo por vez poderá utilizar o barramento para transmissão de dados. O barramento pode ser visto como um conjunto de linhas, em que cada linha vai representar uma informação binária (0 ou 1). O conjunto de linhas formará um código binário. Um sistema pode ter vários tipos de barramentos para realizar a sua interconexão de dados. Você pode vê-los na Figura 6.



**Figura 6.** Tipos de barramentos.

*Fonte:* Adaptada de Stallings (2010).

O **barramento de controle** é usado para controlar o acesso e o uso do **barramento de dados** e do **barramento de endereço**, pois estes são compartilhados por todos os componentes. Além disso, o processador emite sinais de controle pelo barramento de controle, que transmitem informações de comando e sincronização entre os componentes, que podem ser:

- **Escrita de memória:** indica o endereço e os dados a serem armazenados.
- **Leitura de memória:** indica o endereço em que os dados devem ser lidos e colocados no barramento.

- **Escrita de E/S:** faz com que os dados no barramento sejam enviados para a porta de E/S endereçada.
- **Leitura de E/S:** faz com que os dados da porta de E/S endereçada sejam colocados no barramento.
- **AcK de transferência:** indica que dados foram aceitos do barramento ou colocados nele (*AcK* vem do inglês *acknowledgement* — função confirma).
- **Solicitação de barramento (bus request):** indica que um módulo precisa obter controle do barramento.
- **Concessão de barramento (bus grant):** indica que um módulo solicitante recebeu controle do barramento.
- **Requisição de interrupção (interrupt request):** indica que a interrupção está pendente.
- **AcK de interrupção:** confirma que a interrupção pendente foi reconhecida.
- **Clock:** é usado para operações de sincronização.
- **Reset:** inicializa todos os módulos.



### Fique atento



A base da arquitetura dos computadores atuais é similar à proposta por Von Neumann em 1946.

## A máquina de Von Neumann

Matemático húngaro naturalizado americano e de origem judaica, John Von Neumann (1903-1957) foi professor na Universidade de Princeton e contribuiu em diversas áreas, dentre elas: mecânica quântica, teoria dos conjuntos, economia, ciências da computação e matemática. Participou da construção do ENIAC, o primeiro computador eletrônico. É conhecido por ter formalizado o projeto lógico de um computador que é utilizado pela maioria dos computadores utilizados atualmente. No modelo proposto por Von Neumann, foram definidos os seguintes aspectos:



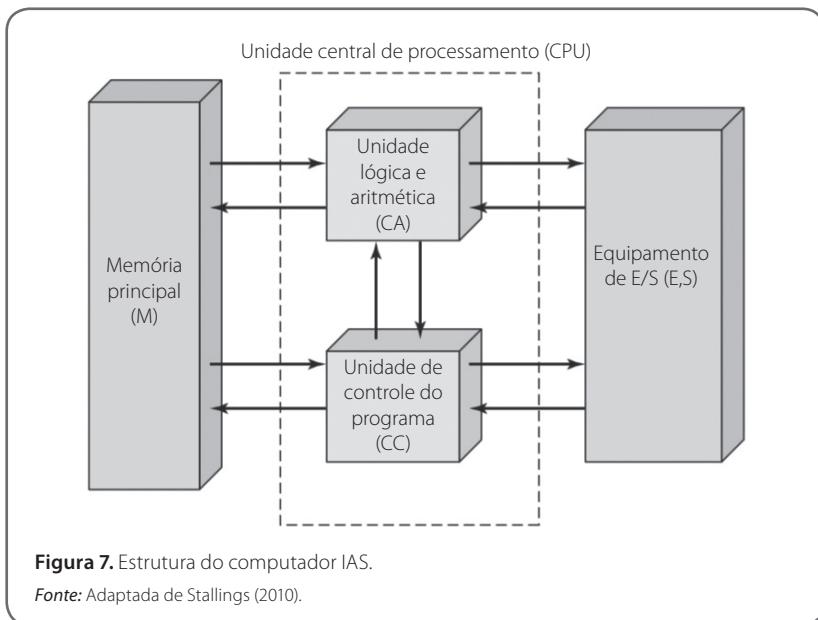
- Armazenamento das instruções na memória do computador. Antes elas não eram armazenadas, sendo lidas de cartões perfurados e executadas uma a uma. Com o armazenamento interno, foi possível obter mais rapidez. Esse conceito foi denominado **programa armazenado**, que também foi definido, na mesma época, por Allan Turing.
- Realização das operações aritméticas. Muito comuns para todo o processamento, elas deveriam ser realizadas por uma unidade especializada, que foi denominada **central aritmética**, que inicialmente trabalhava com dados binários.
- Controle do sequenciamento das instruções dos programas (**central de controle**), que, além de estarem armazenadas na memória do computador (programa armazenado), aquelas que fossem cálculos seriam realizadas pela central aritmética, devendo ser adequadamente sequenciadas, visando a uma maior eficiência de uso dos recursos.
- Memórias para todos os componentes, permitindo que os dados ficassem acessíveis rapidamente para a realização das operações.
- Unidades de transferência, definindo que os dispositivos precisam ter unidades que recebam dados e enviem dados para outros componentes (Entrada/Saída).

Von Neumann e seus colegas começaram o projeto em 1946, mas somente em 1952 concluíram a construção do computador IAS (sigla referente ao Institute for Advanced Studies, de Princeton), cuja estrutura é mostrada na Figura 7.



### Saiba mais

John Von Neumann foi indicado pelo presidente Eisenhower para a Comissão de Energia Atômica, onde supervisionou a construção do arsenal nuclear norte-americano no pós-guerra.



**Figura 7.** Estrutura do computador IAS.

*Fonte:* Adaptada de Stallings (2010).

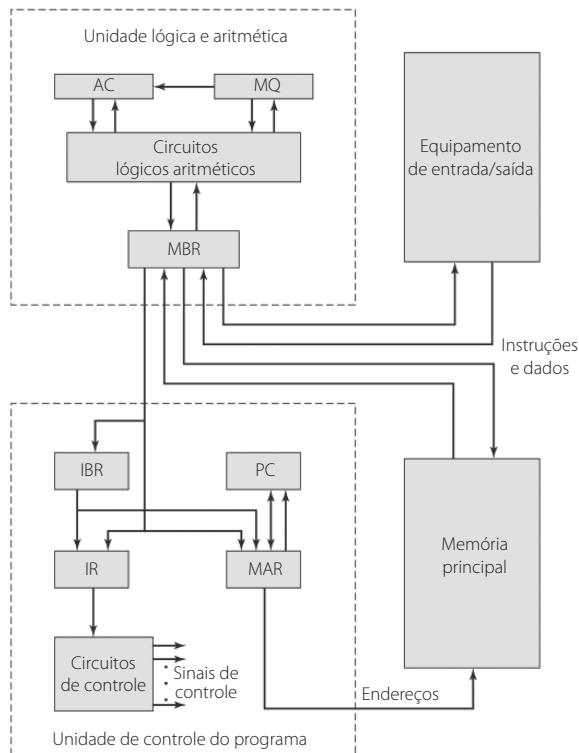
No IAS, a memória possui mil espaços para guardar dados — denominados palavras (*words*), com 40 dígitos binários (*bits*) cada uma —, onde podem ser armazenados dados e instruções, ambos codificados em binário. Os dados utilizam 1 *bit* para sinal e os outros para representar o valor. Uma palavra pode conter duas instruções de 20 *bits* cada, sendo 8 *bits* para a instrução (*OpCode* — código da operação) e 12 *bits* para o endereço de uma palavra de memória.

A UC e a ULA (central aritmética) possuem locais para guardar dados, denominados registradores, definidos da seguinte forma:

- **Registrador de buffer de memória** (MBR, do inglês *memory buffer register*): registra uma palavra que será armazenada na memória, transmitida para a unidade de E/S ou usada para receber uma palavra da memória ou de uma unidade de E/S.
- **Registrador de endereço de memória** (MAR, do inglês *memory address register*): contém o endereço na memória da palavra a ser escrita ou lida pelo MBR.
- **Registrador de instrução** (IR, do inglês *instruction register*): contém o código da operação de 8 *bits* da instrução que está sendo executada.
- **Registrador de buffer de instrução** (IBR, do inglês *instruction buffer register*): utilizado para conter a próxima instrução a ser executada.

- **Contador de programa** (PC, do inglês *program counter*): registra o endereço do próximo conjunto (par) de instruções a ser lido na memória.
- **Acumulador (Ac) e quociente multiplicador** (mq, do inglês *multiplier quotient*): registra os operandos e resultados da ULA.

No modelo de Von Neumann, o computador realiza repetidamente o ciclo de instrução, sendo que cada ciclo possui dois subciclos: **busca** (*fetch cycle*) e **execução** (*execution cycle*). No ciclo de busca, é carregado o código da próxima instrução a ser executada. O ciclo de execução interpreta o código da operação e executa a instrução, por meio do envio dos sinais de controle apropriados para que os dados sejam movidos ou que uma operação seja realizada pela ULA. Na Figura 8 você pode verificar a inter-relação entre os elementos que compõem a arquitetura do computador IAS.



**Figura 8.** Estrutura interna do computador IAS.

*Fonte:* Adaptada de Stallings (2010).



## Link

Acesse os *links* a seguir e conheça um pouco mais sobre a arquitetura de Von Neumann:

<https://goo.gl/B3Edes>

<https://goo.gl/Gy6WDT>

A quantidade de instruções realizada pelo computador IAS era pequena — apenas 21 instruções, que podem ser agrupadas em cinco tipos:

1. Transferência de dados: instruções para mover dados entre memória e registradores da central aritmética (atual ULA) ou entre registradores da central aritmética.
2. Desvio incondicional: instruções para direcionar a execução para outra linha do programa, alterando assim o PC.
3. Desvio condicional: instruções que realizam um teste e, conforme o seu resultado, alteram o PC.
4. Aritméticas: instruções que acionam as operações lógicas e aritméticas.
5. Modificação de endereço: instrução que gera endereços a partir de operações na central aritmética (ULA), que são inseridos em instruções armazenadas na memória.

Veja no Quadro 1 as instruções que podiam ser executadas no computador IAS:

**Quadro 1.** Instruções do computador IAS

<b>Tipo de instrução</b>	<b>Opcode</b>	<b>Representação simbólica</b>	<b>Descrição</b>
Transferência de dados	00001010	LOAD MQ	Transfere o conteúdo de MQ para AC
	00001001	LOAD MQ,M(X)	Transfere o conteúdo do local de memória X para MQ
	00100001	STOR M(X)	Transfere o conteúdo de AC para o local de memória X
	00000001	LOAD M(X)	Transfere M(X) para o AC
	00000010	LOAD - M(X)	Transfere - M(X) para o AC
	00000011	LOAD  M(X)	Transfere o valor absoluto de M(X) para o AC
	00000100	LOAD -  M(X)	Transfere - M(X)  para o acumulador
Desvio incondicional	00001101	JUMP M(X,0:19)	Apanha a próxima instrução da metade esquerda de M(X)
	00001110	JUMP M(X,20:39)	Apanha a próxima instrução da metade direita de M(X)
Desvio condicional	00001111	JUMP+ M(X,0:19)	Se o número no AC for não negativo, apanha a próxima instrução da metade esquerda de M(X)
	00010000	JUMP+ M(X,20:39)	Se o número no AC for não negativo, apanha a próxima instrução da metade direita de M(X)

(Continua)

(Continuação)

**Quadro 1.** Instruções do computador IAS

<b>Tipo de instrução</b>	<i>Opcode</i>	<b>Representação simbólica</b>	<b>Descrição</b>
Aritmética	00000101	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000111	ADD  M(X)	Soma  M(X)  a AC; coloca o resultado em AC
	00000110	SUB M(X)	Subtrai M(X) de AC; coloca o resultado em AC
	00001000	SUB  M(X)	Subtrai  M(X)  de AC; coloca o resto em AC
	00001011	MUL M(X)	Multiplica M(X) por MQ; coloca os <i>bits</i> mais significativos do resultado em AC; coloca <i>bits</i> menos significativos em MQ
	00001100	DIV M(X)	Divide AC por M(X); coloca o quociente em MQ e o resto em AC
	00010100	LSH	Multiplica o AC por 2; ou seja, desloca à esquerda uma posição de <i>bit</i>
	00010101	RSH	Divide o AC por 2; ou seja, desloca uma posição à direita
Modificação de endereço	00010010	STOR M(X,8:19)	Substitui campo de endereço da esquerda em M(X) por 12 <i>bits</i> mais à direita de AC
	00010011	STOR M(X,28:39)	Substitui campo de endereço da direita em M(X) por 12 <i>bits</i> mais à direita de AC

**Fonte:** Adaptado de Stallings (2010).

É fascinante perceber que, mesmo utilizando tecnologias eletrônicas mais avançadas e componentes mais sofisticados, com mais recursos e funções, os computadores atuais seguem a mesma estrutura do modelo de Von Neumann.



## Referência

STALLINGS, W. *Arquitetura e organização de computadores*. 8. ed. São Paulo: Pearson Pratice Hall, 2010.

## Leituras recomendadas

ALSINA. *Organização de computadores*. Disponível em: <[https://www.dca.ufrn.br/~pablo/FTP/arq\\_de\\_comp/apostilha/capitulo2.pdf](https://www.dca.ufrn.br/~pablo/FTP/arq_de_comp/apostilha/capitulo2.pdf)>. Acesso em: 7 dez. 2018.

NULL, L.; LOBUR, J. *Princípios básicos de arquitetura e organização de computadores*. 2. ed. Porto Alegre: Bookman, 2011.

TANENBAUM, A. S. *Organização estruturada de computadores*. 6. ed. São Paulo: Pearson, 2013.

VONNEUMAN, N. A. O legado filosófico de John von Neumann. *Estud. Av.*, São Paulo, v. 10, n. 26, p. 189–204, abr. 1996. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-40141996000100018&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40141996000100018&lng=en&nrm=iso)>. Acesso em: 7 dez. 2018.

WEBER, R. F. *Fundamentos de arquitetura de computadores*. 4. ed. Porto Alegre: Bookman: 2012.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS