

**INSTITUTO FEDERAL DO PIAUÍ – IFPI
CAMPUS ANGICAL
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMA**

José Henrique de Sousa Oliveira

Arquitetura de Computadores.

**Angical do Piauí
2025**

Arquitetura de Computadores.

José Henrique de Sousa Oliveira.

Resumo

Este artigo apresenta uma análise sobre a evolução dos computadores, abordando desde os primeiros sistemas sem sistemas operacionais até as gerações mais modernas. São discutidos os principais avanços tecnológicos que marcaram cada período, como a substituição das válvulas pelos transistores e o uso de circuitos integrados, além da introdução de linguagens de programação e sistemas operacionais que tornaram o uso dos computadores mais eficiente. Também são explorados conceitos fundamentais da arquitetura e organização de computadores, destacando a relação entre capacidade, velocidade e custo da memória, e a importância da hierarquia de memória para otimizar o desempenho do sistema. O estudo evidencia como as inovações tecnológicas transformaram o modo de operação, tornando os computadores mais rápidos, acessíveis e essenciais para o desenvolvimento científico e social.

Palavras-chave: Computadores; Arquitetura; Organização; Memória; Evolução tecnológica

Abstract

This article presents an analysis of the evolution of computers, covering early systems without operating systems to modern generations. The main technological advances of each period are discussed, such as the replacement of vacuum tubes with transistors, the use of integrated circuits, and the introduction of programming languages and operating systems that made computer usage more efficient. Fundamental concepts of computer architecture and organization are also explored, highlighting the relationship between memory capacity, speed, and cost, as well as the importance of memory hierarchy to optimize system performance. The study demonstrates how technological innovations have transformed computer operations, making them faster, more accessible, and essential for scientific and social development.

Keywords: Computers; Architecture; Organization; Memory; Technological Evolution

1 Introdução.

Arquitetura de computador refere-se aos atributos de um sistema visíveis a um programador ou, em outras palavras, aqueles atributos que possuem um impacto direto sobre a execução lógica de um programa. Um termo que é muitas vezes usado de maneira intercambiável com as arquiteturas de computadores é arquitetura de conjunto de instrução (ISA — do inglês, Instruction Set Architecture). O ISA define os formatos de instruções, códigos de operação da instrução (opcodes), registradores, memória de dados e instrução; o efeito das instruções executadas nos registradores e na memória; e um algoritmo para o controle da execução das instruções.

Organização de computador refere-se às unidades operacionais e suas interconexões que percebam as especificações de arquitetura. Os exemplos de atributos de arquitetura incluem o conjunto de instrução, o número de bits usados para representar vários tipos de dados (por exemplo, números, caracteres), mecanismos de E/S e técnicas para endereçamento de memória. Atributos organizacionais incluem os detalhes do hardware transparentes ao programador, como sinais de controle, interfaces entre o computador e periféricos e a tecnologia de memória utilizada.

A distinção entre Arquitetura e Organização é fundamental no projeto de computadores, sendo a Arquitetura o conjunto de atributos visíveis ao programador (como o conjunto de instruções, garantindo a compatibilidade do software), e a Organização a implementação física e as interconexões internas do hardware, que definem o desempenho e o custo. Historicamente, essa separação permitiu que fabricantes como a IBM oferecessem uma família de modelos (ex: System/370) com a mesma arquitetura para proteger o investimento em software do cliente, mas com organizações variadas para diferentes preços e velocidades, facilitando upgrades. Embora a Arquitetura tenha longevidade, a Organização muda com a tecnologia; contudo, em microcomputadores, a relação é mais próxima, e as mudanças tecnológicas frequentemente levam à criação de arquiteturas inteiramente novas (como o RISC).

2 Conceitos básicos de informática

Um computador é uma máquina (conjunto de partes eletrônicas e eletromecânicas) capaz de sistematicamente coletar, manipular e fornecer os resultados da manipulação de dados para um ou mais objetivos. Por ser uma máquina composta de vários circuitos e componentes eletrônicos, também é chamado de equipamento de processamento eletrônico de dados. O esquema básico de um processamento de dados (manual ou automático) resulta em um produto acabado: a informação. Assim, os dados precisam ser processados para que algum resultado tenha significado para alguém ou para o próprio computador.

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

- a) grande velocidade no processamento e disponibilização de informações;
- b) precisão no fornecimento das informações;
- c) adequação para execução de tarefas repetitivas; d) redução de custos operacionais;
- e) compartilhamento de dados.

2.1 Estrutura e função.

Um computador é um sistema complexo; computadores contemporâneos contêm milhões de componentes eletrônicos elementares. Como, então, alguém poderia descrevê-los com clareza? A chave é reconhecer a natureza hierárquica dos sistemas mais complexos, incluindo o computador (SIMON, 1996). Um sistema hierárquico é um conjunto de subsistemas interrelacionados, cada um deles, por sua vez, hierárquico em estrutura até alcançarmos algum nível mais baixo de subsistema elementar.

A natureza hierárquica dos sistemas complexos é essencial para seu projeto e sua descrição. O projetista só precisa lidar com um nível particular do sistema de cada vez. Em cada nível, o sistema consiste em um conjunto de componentes e suas inter-relações. O comportamento em cada nível depende somente de uma caracterização simplificada e resumida do sistema, no próximo nível mais baixo. Em cada nível, o projetista está interessado na estrutura e na função:

Estrutura: o modo como os componentes são inter-relacionados.

Função: a operação individual de cada componente como parte da estrutura.

2.2 Função:

Tanto a estrutura quanto o funcionamento de um computador são, essencialmente, simples. Em termos gerais, há somente quatro funções básicas que podem ser apresentadas pelo computador:

Processamento de Dados: Envolve a manipulação dos dados de entrada, independentemente de sua forma ou complexidade, utilizando métodos fundamentais de processamento.

Armazenamento de Dados: Refere-se à capacidade de guardar dados tanto a curto prazo (dados atualmente em uso) quanto a longo prazo (arquivos para recuperação e atualização futura).

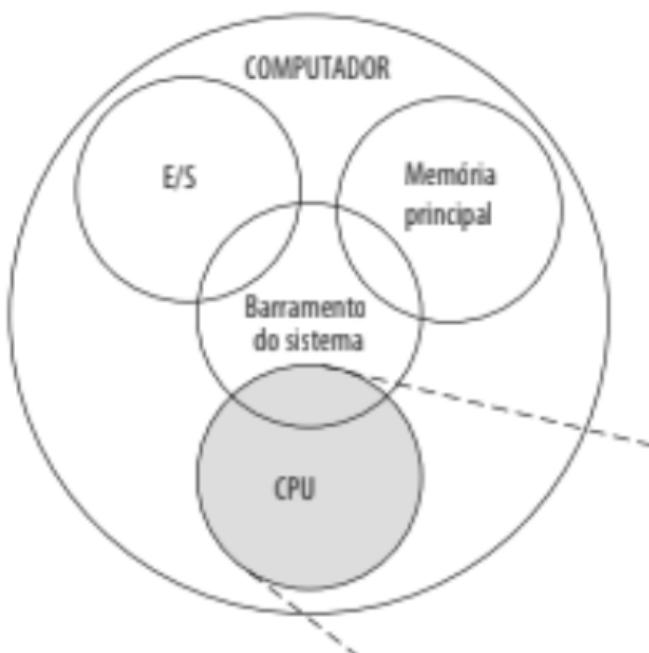
Movimentação de Dados: Abrange a transferência de informações entre o computador e seu ambiente operacional. Quando a transferência é feita para um dispositivo diretamente conectado, é chamada de Entrada-Saída (E/S ou I/O); quando é feita a longas distâncias (dispositivos remotos), é chamada de comunicações de dados.

Controle: É a função realizada pela unidade de controle (UC), que gerencia os recursos do computador e coordena o desempenho de todas as outras partes funcionais em resposta às instruções.

2.3 Estrutura:

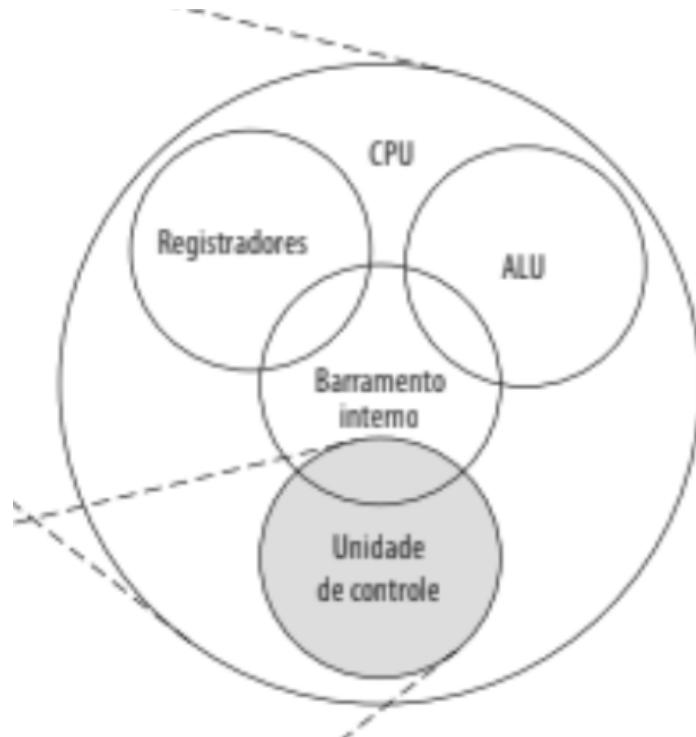
A estrutura de um computador é composta por quatro componentes principais de alto nível: a Unidade Central de Processamento (CPU), que controla as operações e processa os dados; a Memória Principal, que armazena dados; os dispositivos de E/S (Entrada/Saída), que movem dados entre o computador e o ambiente externo; e o Sistema de Interconexão (como o barramento do sistema), que permite a comunicação entre os demais. Por sua vez, o componente mais complexo, a CPU, é estruturado internamente pela Unidade de Controle (que gerencia a operação), a Unidade Lógica e Aritmética (ALU) (que executa as funções de processamento), os Registradores (para armazenamento interno) e a Interconexão da CPU, sendo que sua complexidade é intensificada por técnicas como pipeline e implementação microprogramada.

Figura 1: Um computador: estrutura de alto nível.



Fonte: STALLINGS, 2018, p. 4.

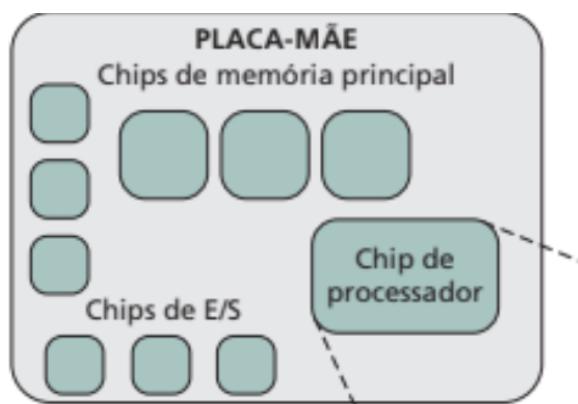
Figura 2: Um computador: estrutura de alto nível.



Fonte: STALLINGS, 2018, p. 4.

A placa-mãe contém um slot ou um soquete para o chip do processador, o que geralmente contém múltiplos cores individuais, que é conhecido como processador multicore. Há também slots para os chips da memória, chips de controlador E/S e outros componentes-chave do computador. Para computadores desktop, os slots de expansão permitem a inclusão de mais componentes na placa de expansão. Assim, uma placa-mãe moderna conecta somente poucos componentes individuais, com cada chip contendo de alguns milhares até centenas de milhões de transistores.

Figura 3: Um computador: estrutura de alto nível.



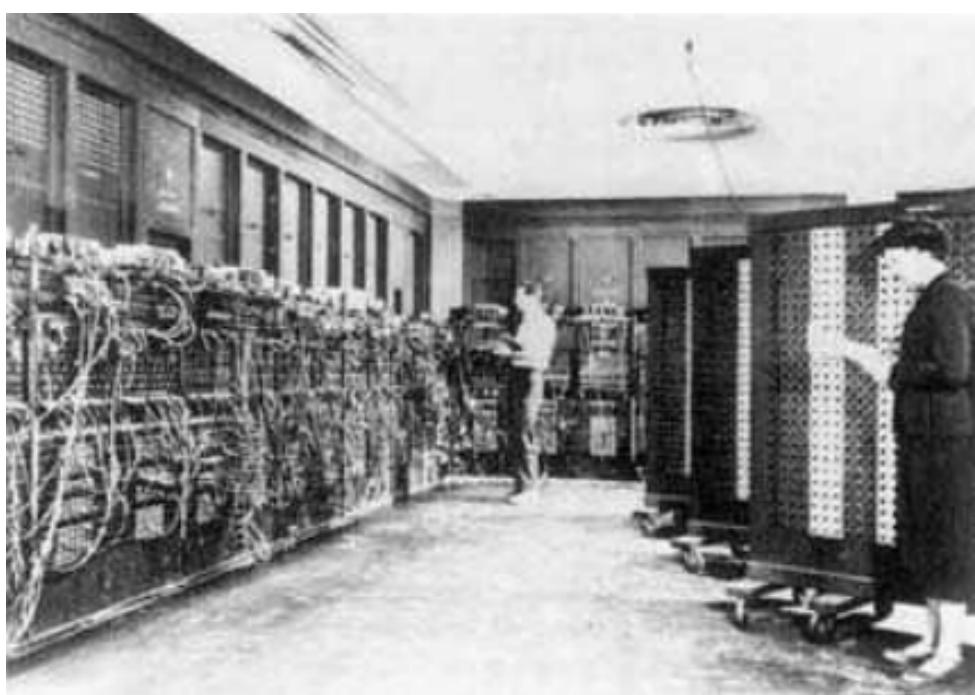
Fonte: STALLINGS, ed.10. 2018, p. 4.

3 Evolução e desempenho do computador.

A primeira geração: válvulas: os computadores usavam válvulas para elementos lógicos digitais e memória. Uma série de pesquisas e computadores comerciais foram construídos com válvulas. Para nosso propósito, será instrutivo examinar talvez o mais famoso computador de primeira geração, conhecido como computador IAS.

Uma técnica fundamental de projeto implementada em primeiro lugar no computador IAS é conhecida como conceito de programa armazenado. Essa ideia é geralmente atribuída ao matemático John von Neumann. Alan Turing desenvolveu a ideia ao mesmo tempo. A primeira publicação da ideia ocorreu em uma proposta em 1945, por von Neumann para um novo computador, o EDVAC (Electronic Discrete Variable Computer). Em 1946, von Neumann e seus colaboradores começaram a desenvolver um novo computador de programa armazenado, referido como computador IAS, no Princeton Institute for Advanced Studies.

Figura 4: O primeiro grande computador eletrônico [foto Exército EUA]

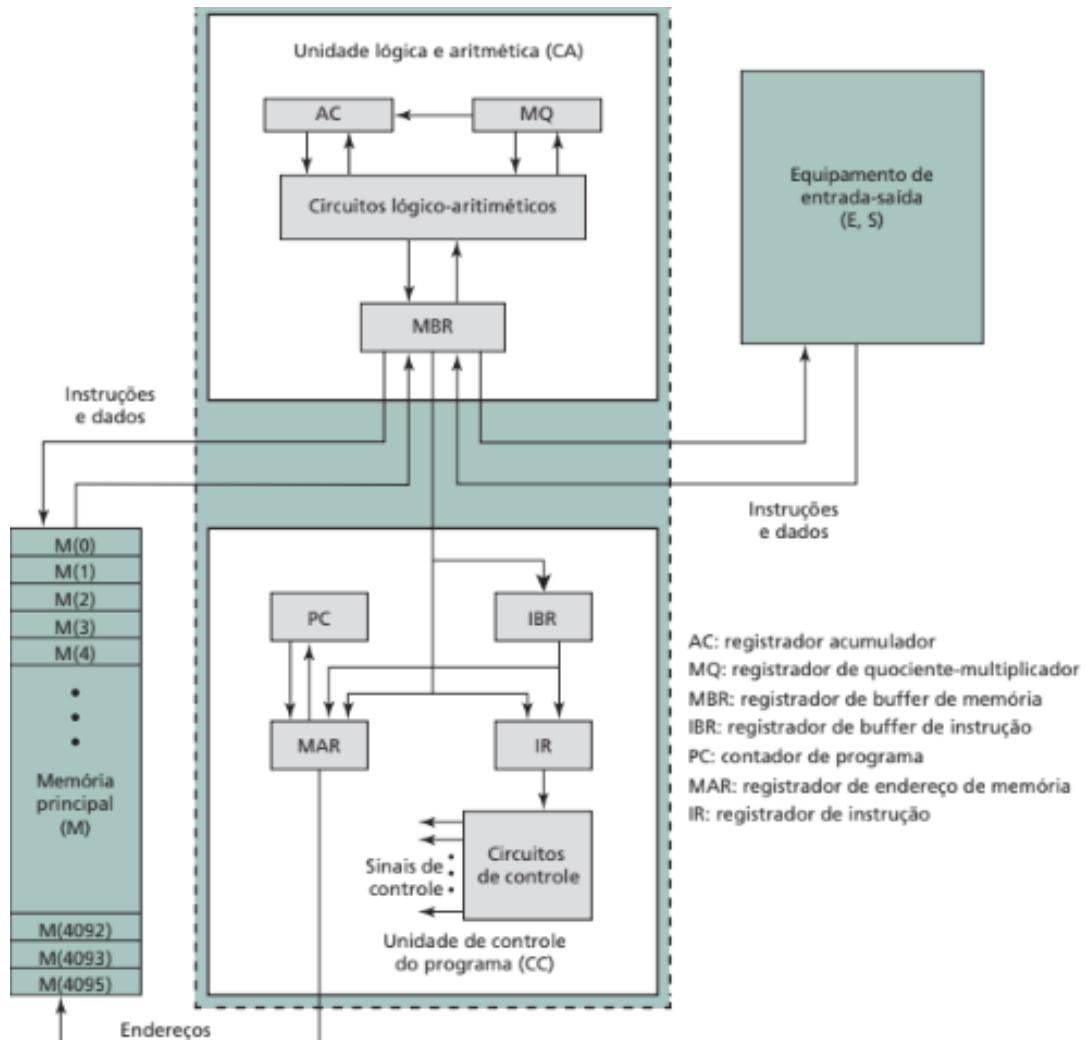


Fonte: Porto Fernandez, ed. 3, 2015, p. 14.

Eckert e Mauchly se juntaram ao matemático John Von Neumann em 1944 para projetar o EDVAC (Electronic Discrete Variable Automatic Computer), que abriu caminho para o programa armazenado. Ele foi um sucessor do ENIAC e usava numeração binária em vez de decimal e já utilizava o conceito de programa através de cartão perfurado. Isso permitia ao EDVAC mudanças de programas mais rápidas em comparação ao ENIAC. No entanto, as principais contribuições do EDVAC foi a arquitetura de processador, e memória e unidades de entrada e saída interligadas por um barra-

mento. Essa arquitetura, conhecida como Arquitetura Von Neumann, é utilizada até hoje pela maioria dos computadores. Mostramos a seguir a arquitetura do EDVAC.

Figura 5: - Unidade central de processamento (CPU).



Fonte: STALLINGS, ed.10. 2018, p. 10.

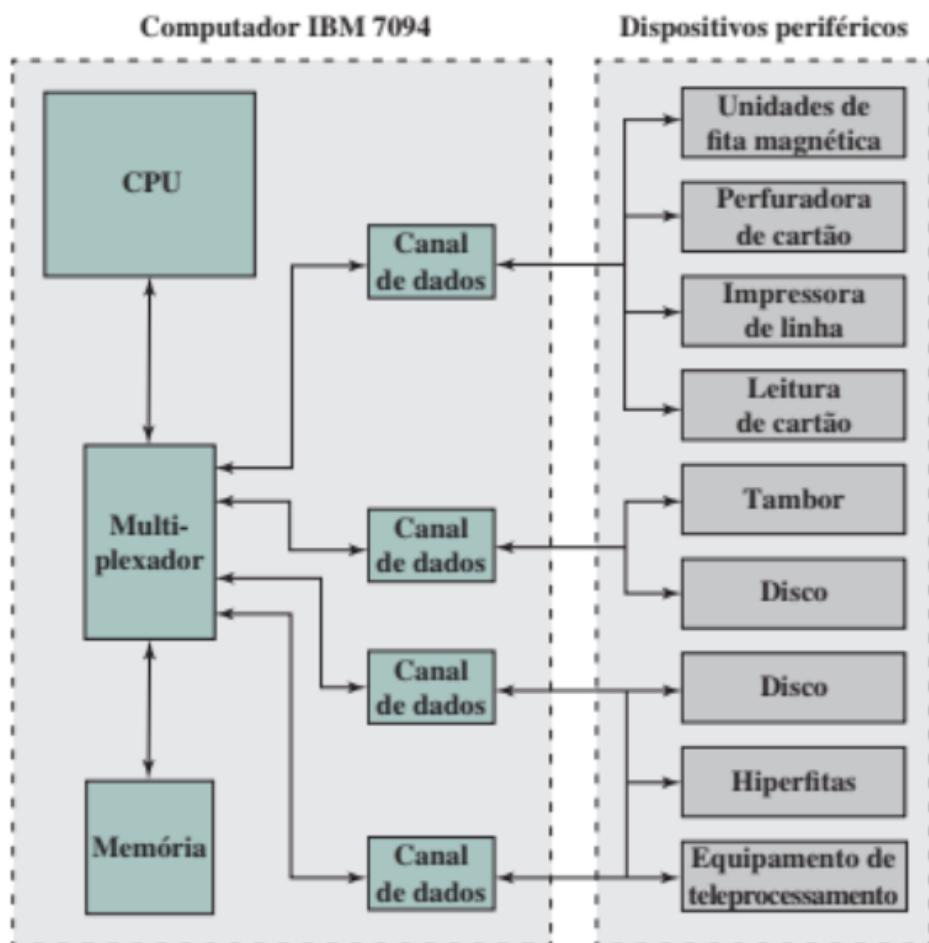
A segunda geração: transistores: o advento do transistor em 1947, um dispositivo de estado sólido baseado em silício, marcou a Segunda Geração de Computadores no final da década de 1950, superando as limitações das válvulas eletrônicas. Esta mudança tecnológica proporcionou ganhos exponenciais, resultando em máquinas com desempenho, capacidade de memória e confiabilidade significativamente superiores, além de um tamanho reduzido.

A evolução não se limitou ao hardware; a Segunda Geração consolidou o uso de linguagens de programação de alto nível e a introdução de software de sistema, que oferecia funcionalidades básicas de sistemas operacionais modernos, como carregamento de programas e movimentação de dados para periféricos. O IBM 7094 é um exemplo notável dessa era, demonstrando um aumento de 50 vezes na velocidade

da CPU e a queda do tempo de ciclo de memória de 30 ms para 1,4 ms. O aprimoramento do desempenho foi alcançado por meio de dispositivos eletrônicos mais rápidos e inovações em circuitos, como o Registrador de Backup de Instrução, que otimizava a busca de comandos.

Crucialmente, a introdução de canais de dados (módulos de E/S independentes com seu próprio processador) e do multiplexador permitiu que as tarefas de Entrada/Saída fossem executadas de forma autônoma em relação à CPU, liberando o processador principal de uma carga de trabalho considerável e permitindo a atuação independente dos diversos componentes do sistema.

Figura 6: - Configuração de um IBM 7094.



Fonte: STALLINGS, ed.10. 2018, p. 16.

A terceira geração: circuitos integrados: a terceira geração, que durou de 1964 a 1970, foi marcada pela substituição dos transistores pela tecnologia de circuitos integrados , transistores e componentes eletrônicos montados em um único substrato. O circuito integrado foi inventado por Jack St. Clair Kilby e Robert Noyce e permitiu a compactação dos computadores construídos com transistor. Nessa gera-

ção, o cálculo de operações matemáticas chegava à ordem de nanossegundos além do aumento da capacidade, redução das dimensões físicas e redução na dissipação de calor.

O principal computador comercial da terceira geração foi o IBM 360, lançado em 1964. Ele dispunha de muitos modelos e de várias opções de expansão que realizava mais de 2 milhões de adições por segundo e cerca de 500 mil multiplicações. O System/360 foi um dos maiores sucessos de venda da IBM.

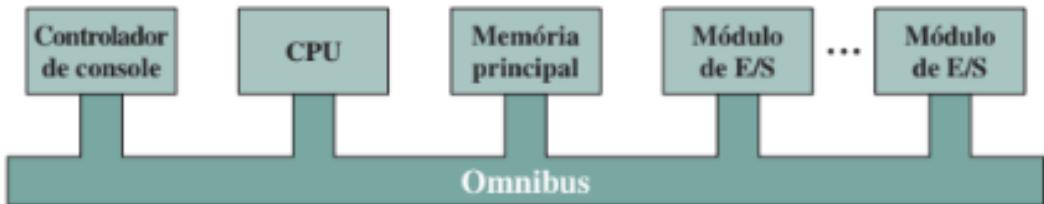
MICROELETRÔNICA: significa, literalmente, “pequena eletrônica”. Desde os primórdios da eletrônica digital e da indústria da computação, tem havido uma tendência persistente e consistente quanto à redução no tamanho dos circuitos eletrônicos digitais. Antes de examinarmos as implicações e os benefícios dessa tendência, precisamos dizer algo sobre a natureza da eletrônica digital.

IBM SYSTEM/360: o anúncio do IBM System/360 em 1964 marcou a introdução da Terceira Geração de Computadores e, apesar de ser inicialmente incompatível com os modelos 7000 anteriores (o que gerou resistência dos clientes), sua estratégia se mostrou um sucesso financeiro e técnico. A IBM visava resolver as limitações da arquitetura anterior e incorporar a nova tecnologia de circuito integrado.

O System/360 foi a primeira família de computadores planejada do setor, cobrindo uma vasta gama de desempenho e custo. O conceito chave era a compatibilidade de software: um programa escrito para um modelo mais fraco (como o Modelo 30) poderia ser executado em um modelo mais potente e caro (Modelo 75) sem modificações.

MEMÓRIA SEMICONDUTORA: a primeira aplicação da tecnologia de circuito integrado aos computadores foi a construção do processador (a unidade de controle e a unidade lógica e aritmética) em chips de circuito integrado. Mas também descobriu-se que essa mesma tecnologia poderia ser usada para construir memórias. Tradicionalmente, nas décadas de 1950 e 1960, a memória principal era construída a partir de pequenos anéis de material ferromagnético (memória core), que representavam bits por sua direção de magnetização. Embora fosse relativamente rápida (tempo de leitura de 1 microsssegundo), a memória core era cara, volumosa e destrutiva (a leitura apagava o dado, exigindo circuitos de restauração).

Figura 7: - Estrutura de barramento PDP-8.



Fonte: STALLINGS, ed.10. 2018, p. 20.

Em 1970, a Fairchild introduziu a primeira memória de semicondutor em um chip, capaz de armazenar 256 bits, que era não destrutiva e drasticamente mais rápida (70 nanossegundos para ler um bit), embora inicialmente mais cara por bit que a memória core.

O ponto de inflexão ocorreu em 1974, quando o preço por bit da memória semicondutora caiu e se tornou mais acessível que o preço da memória core. A partir desse momento, houve um declínio contínuo nos custos e um rápido aumento na densidade de armazenamento (evoluindo de 1K até 16 Gbits por chip). Essa evolução, combinada com o avanço dos processadores, resultou em máquinas menores, mais rápidas e com maior capacidade, permitindo a popularização dos computadores para o "usuário final", como máquinas de escritório e PCs.

MICROPROCESSADORES: assim como a densidade dos elementos nos chips de memória continuava a subir, a densidade dos elementos dos chips do processador subiram. Com o passar do tempo, mais e mais elementos eram colocados em cada chip, de modo que menos e menos chips eram necessários para construir um único processador do computador. Uma descoberta inovadora foi alcançada em 1971, quando a Intel desenvolveu seu 4004. Ele foi o primeiro chip a conter em si todos os componentes de uma CPU: nascia o microprocessador. O 4004 pode somar dois números de 4 bits e pode multiplicar apenas pela adição repetida. Pelos padrões de hoje, o 4004 é desesperadamente primitivo, mas ele marcou o início de uma evolução contínua da capacidade e do poder do microprocessador.

Quarta Geração: O marco da Quarta Geração de Computadores foi o lançamento do Intel 4004 em novembro de 1971, o primeiro microprocessador comercial, que, juntamente com a invenção da memória RAM semicondutora, viabilizou a construção de microcomputadores. Embora o 4004 fosse modesto em sua capacidade inicial (60.000 instruções por segundo), seus sucessores subsequentes (como o 8080 e a série 80x86) ofereceram aumentos exponenciais em velocidade e capacidade de processamento.

A principal vantagem do circuito integrado reside na capacidade de produção em massa simultânea dos componentes (transistores), o que reduz drasticamente os

custos e acelera o processo de fabricação (comparado ao impacto da invenção da prensa de Gutenberg nos livros). Essa tecnologia permitiu um salto colossal na densidade: enquanto um computador transistorizado de 1959 exigia 10 m² para 150.000 transistores individuais, nos anos 80, a mesma quantidade podia ser fabricada em um único circuito integrado. Atualmente, microprocessadores contêm dezenas de milhões de transistores fabricados simultaneamente em uma única pastilha de silício.

Figura 8: Microcomputador IBM PC [foto Computer History Museum]



Fonte: Porto Fernandez, ed. 3, 2015, p. 16.

Sistemas operacionais embarcados Há duas técnicas gerais para desenvolver o sistema operacional (SO) embarcado. A primeira técnica é pegar um SO existente e adaptar para a aplicação embarcada. Por exemplo, há versões embarcadas de Linux, Windows e Mac, bem como outros sistemas operacionais comerciais e particulares especializados para sistemas embarcados. A outra técnica é desenvolver e implementar um SO direcionado unicamente para o uso embarcado. Um exemplo é o TinyOS, amplamente usado em redes de sensor sem fio. Esse assunto foi explorado detalhadamente por este autor (STALLINGS, 2015).

4 Componentes de uma Arquitetura de Computador.

Um computador é construído com vários componentes individuais. As ligações entre os diversos componentes é a essência da Arquitetura de Computadores. A forma como os componente se interligam e se comunicam vai determinar a capacidade e velocidade de processamento de uma determinada arquitetura. A primeira seção mostra a arquitetura geral de um computador. Em seguida, apresentamos a Arquitetura Harvard e a Arquitetura Von Neumann, importantes para entender as Arquiteturas usadas

atualmente. Finalmente apresentamos a Lei de Moore, um prognóstico de evolução dos microprocessadores e alguns comentários sobre os limites dela.

4.1 Componentes de um computador.

O principal módulo de um computador é a Unidade Central de Processamento (CPU), o "cérebro" responsável por todo o processamento, geralmente construído em um único circuito integrado. A CPU é composta pela Unidade Lógica e Aritmética (ULA), que realiza operações aritméticas e lógicas; pelos Registradores, que são memórias internas muito rápidas para operações imediatas (auxiliadas pela memória Cache); e pela Unidade de Controle, que gerencia o funcionamento da CPU e interpreta o código do programa para nortear o processamento.

Externamente, a CPU se conecta a outros módulos por meio do Barramento, que interliga todos os componentes e é tipicamente dividido em barramentos de dados, endereço e controle. Os módulos externos incluem a Memória Principal, um conjunto rápido de memória semicondutora para armazenamento temporário de trabalho (volátil); o Disco, uma unidade eletromecânica de armazenamento permanente e de grande capacidade (porém mais lenta); e o módulo de Entrada e Saída (I/O), que conecta o computador a periféricos (como teclado, monitor e placas de comunicação) e é significativamente mais lento devido à natureza dos dispositivos externos.

4.2 Computadores: hardware e software.

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

Unidades de Entrada e Saída: O excerto explica que os dispositivos de Entrada e Saída (E/S) são estruturados em duas partes: o controlador (chips que controlam fisicamente o dispositivo e recebem comandos do software) e o dispositivo propriamente dito.

As Unidades de Entrada são dispositivos físicos (como teclado, mouse e scanner) que capturam os dados (texto, vídeo ou áudio) a serem processados. Já as Unidades de Saída são os dispositivos que apresentam os resultados finais do processamento ao usuário (como monitores, impressoras e painéis de senha).

Existem, ainda, dispositivos que atuam como ambos (E/S), permitindo tanto a entrada quanto a saída de dados. Exemplos incluem unidades de disco (discos

rígidos, CD/DVD), unidades USB (pen drives) e telas touch screen.

Figura 9: Exemplos de dispositivos de entrada e saída



Fonte: Forzza Amaral, 2010, p. 17.

Memória: O computador deve ser dotado de alguma forma de armazenamento (temporário ou permanente) para que os dados coletados ou processados possam ser armazenados. A essa estrutura damos o nome genérico de memória (não está contextualizado aqui o tipo da memória). A memória armazena, essencialmente, os bits. A menor unidade de informação em um computador é o bit, que pode assumir os valores 0 ou 1. Como um único bit é insuficiente para representar informações mais complexas, eles são agrupados e combinados. Num primeiro agrupamento, são reunidos em conjuntos de oito bits, recebendo a denominação de Byte. Como a unidade byte (unidade de medida de armazenamento) também é, consideravelmente, pequena quando indicamos valores mais extensos, utilizamos múltiplos do byte: quilobyte, megabyte, gigabyte, terabyte, etc.

1 Quilobyte = 1 KB = 1024 Bytes,

1 Megabyte = 1 MB = 1024 Kbytes,

1 Gigabyte = 1 GB = 1024 Mbytes,

1 Terabyte = 1 TB = 1024 Gbytes.

Unidade Central de Processamento (UCP): a CPU (Central Processing Unit ou Unidade Central de Processamento), microprocessador ou processador é a parte do computador que interpreta e executa as instruções contidas no software. Na maioria das CPUs essa tarefa é dividida entre uma unidade de controle que dirige o fluxo do programa e uma ou mais unidades que executam operações em dados. A CPU é constituída pelos seguintes componentes: a ULA (Unidade de Lógica e Aritmética), a UC (Unidade de Controle) e os registradores. Um tipo de registrador especial é o contador de programa, que contém o endereço de memória da próxima instrução que

a CPU vai buscar. Assim, esse registrador vai ser atualizado para conter sempre o endereço da próxima instrução a ser processada.

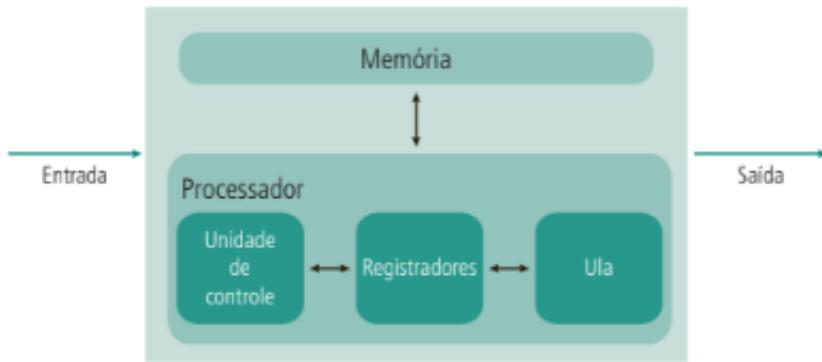
Software: um programa de computador pode ser definido como uma série de instruções ou declarações, em forma inteligível pelo computador, preparada para obter certos resultados. Um programa pode ser chamado de software, porém esse termo pode ser utilizado também para um grupo de programas ou para todo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

5 Modelo de Von Neumann.

A Arquitetura de von Neumann é um modelo de arquitetura para computador digital de programa armazenado onde existe uma única estrutura compartilhada para armazenamento de instruções e de dados. O nome é uma homenagem ao matemático e cientista de computação John Von Neumann, que primeiro propôs essa idéia. Um computador com Arquitetura von Neumann mantém suas instruções de programa e seus dados na memória de leitura/escrita (RAM). Essa arquitetura foi um avanço significativo sobre os computadores dos anos 40, tais como o Colossus e o ENIAC, que eram programados ajustando interruptores e introduzindo cabos para ligar os sinais de controle entre as várias unidades funcionais. Na maioria dos computadores modernos, usa-se a mesma memória para armazenar dados e instruções de programa.

Os termos “Arquitetura von Neumann” e “computador de Programa-Armazenado” são usados indistintamente neste texto. Ao contrário, a Arquitetura de Harvard armazena um programa em um local diferente do local utilizado para armazenar dados. Von Neumann foi envolvido no projeto de Manhattan. Lá juntou-se ao grupo que desenvolveu o ENIAC para desenvolver o projeto do computador de programa armazenado chamado EDVAC. O termo “Arquitetura von Neumann” foi mencionada em um relatório sobre o projeto EDVAC, datado de 30 de junho de 1945, que incluiu idéias de Eckert e de Mauchly. O relatório foi lido por vários colegas de von Neumann na América e na Europa, influenciando significativamente todos os futuros projetos de computador. Apesar de ser reconhecida a contribuição de J. Presper Eckert, John Mauchly e Alan Turing para o conceito de computador de programa armazenado, essa arquitetura é mundialmente denominada Arquitetura Von Neumann.

Figura 10: Modelo de Von Neumann,



Fonte: Forzza Amaral, 2010, p. 42.

6 Barramento do sistema.

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

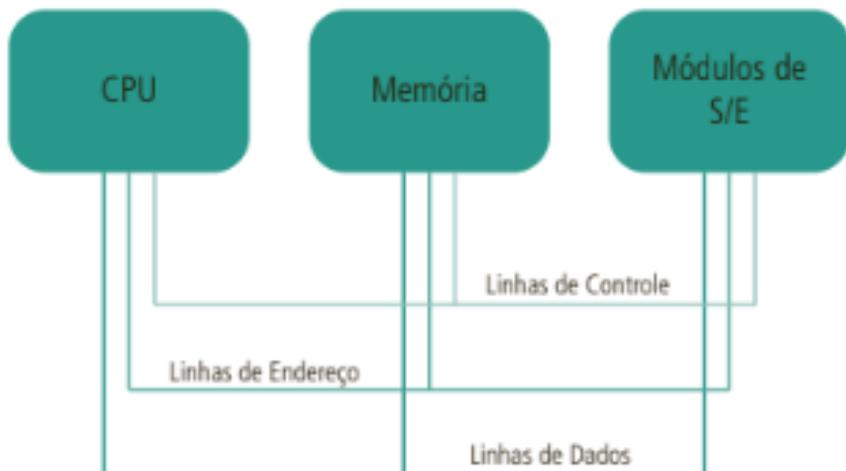
O modelo da estrutura de interconexão depende das informações que são trocadas entre os vários módulos de processador, memória e E/S. A natureza da operação (operação de leitura ou escrita) é definida por meio de sinais de controle. Conforme já foi falado, o módulo de E/S também deve ser capaz de enviar sinais de interrupções ao processador. Isto é especialmente importante quando um dispositivo de E/S precisa enviar algum dado ou estabelecer comunicação com algum outro dispositivo. Um sistema de computação típico contém diversos barramentos, fornecendo caminhos de comunicação entre os seus componentes. O barramento usado para conectar os principais componentes do computador (processador, memória, E/S) é chamado de barramento do sistema.

6.1 Estruturas de barramentos.

O barramento do sistema é o principal mecanismo de interconexão, composto por 50 a 100 linhas que se dividem em três grupos cruciais para a comunicação entre a CPU, a Memória e os módulos de E/S. O barramento de dados transfere os dados propriamente ditos e sua largura (8, 16 ou 32 bits) é um fator determinante do desempenho do sistema. O barramento de endereço é usado para designar a localização (fonte ou destino) dos dados transferidos, e sua largura máxima determina a capacidade máxima de memória do sistema. Por fim, o barramento de controle gerencia

o acesso e a utilização compartilhada das demais linhas, transmitindo sinais de comando (como leitura ou escrita). Para que qualquer módulo use o barramento (seja para enviar ou requisitar dados), ele deve primeiro obter o controle do barramento, sendo essa estrutura implementada fisicamente pelas trilhas da placa-mãe.

Figura 11: Esquema de interconexão de barramento.



Fonte: Forzza Amaral, 2010, p. 51.

Múltiplos barramentos: devido à grande quantidade de dispositivos conectados ao sistema (computador), o uso de um único barramento pode prejudicar o desempenho de dispositivos. Por exemplo, dispositivos lentos “atrasam” o funcionamento dos dispositivos rápidos. Assim, não seria interessante manter, no mesmo barramento, a memória e os dispositivos USB, por exemplo. Desse modo, grande parte dos sistemas computacionais utiliza mais de um barramento, organizados de forma hierárquica.

Barramento PCI: o barramento PCI (Peripheral Component Interconnect) é um mecanismo de interconexão que sucedeu barramentos mais antigos (como MCA, ISA, etc.) e que possui largura de banda suficiente para suportar diversos dispositivos. O PCI pode operar com 32 ou 64 linhas de dados, atingindo uma taxa bruta máxima de 528 MB/s. Contudo, essa tecnologia apresenta desvantagens: ela é compartilhada entre todos os dispositivos, o que causa uma queda generalizada no desempenho. Além disso, o uso de transmissões paralelas limita sua evolução, pois o aumento das taxas de transferência exigiria mais vias de dados, o que nem sempre é fisicamente viável. Atualmente, o PCI está sendo gradualmente substituído pelos barramentos PCI-e.

Barramento PCI - Express: Também denominado PCI-e, esse é um barramento que utiliza transmissão serial em seus circuitos, diminuindo possíveis interferências entre vias de dados. Como é um barramento de alta velocidade, está substituindo os barramentos PCI e o barramento AGP. Uma das características fundamentais do

PCI Express é que ele é um barramento ponto a ponto, em que cada periférico possui um canal exclusivo de comunicação com o chipset. No PCI tradicional, o barramento é compartilhado por todos os periféricos ligados a ele, o que pode criar gargalos, de acordo com Morimoto (2007). Ainda segundo Morimoto (2007), existem quatro tipos de slots PCI-Express, que vão do 1x ao 16x.

O número indica quantas linhas de dados são utilizadas pelo slot e, consequentemente, a banda disponível. Cada linha PCI-Express utiliza quatro pinos de dados (dois para enviar e dois para receber), que são capazes de transmitir a 250 MB/s em ambas as direções.

7 SISTEMA DE MEMÓRIA DO COMPUTADOR.

O complexo assunto da memória de computador pode ser mais bem compreendido se classificarmos os sistemas de memória de acordo com suas principais características. As mais importantes estão listadas na figura 12.

Figura 12: -Principais características dos sistemas de memória do computador.

Localização	Desempenho
Interna (por exemplo, registradores do processador, memória principal, cache)	Tempo de acesso
Externa (por exemplo, discos ópticos, discos magnéticos, fitas)	Tempo de ciclo
	Taxa de transferência
Método de acesso	Tipo físico
Sequencial	Semicondutor
Direto	Magnético
Aleatório	Óptico
Associativo	Magneto-óptico
Unidade de transferência	Características físicas
Palavra	Volátil/não volátil
Bloco	Apagável/não apagável
Capacidade	Organização
Número de palavras	Módulos de memória
Número de bytes	

Fonte: STALLINGS, ed.10. 2018, p. 100.

O termo localização na figura 12 indica se a memória é interna ou externa ao computador. A memória interna em geral significa a memória principal, mas existem outras formas de memória interna. O processador necessita de uma memória local própria, na forma de registradores. Além do mais, como será visto, a parte da unidade de controle do processador também pode exigir sua própria memória interna. Vamos deixar a discussão desses dois últimos tipos de memória interna para capítulos pos-

teriores. A cache é outra forma de memória interna. A memória externa consiste em dispositivos de armazenamento periféricos, como discos e fitas, que são acessíveis ao processador por meio de controladores de E/S.

Uma característica óbvia da memória é a sua capacidade. Para a memória interna, isso costuma ser expresso em termos de bytes (1 byte = 8 bits) ou palavras. Os tamanhos comuns de palavra são 8, 16 e 32 bits. A capacidade da memória externa normalmente é expressa em termos de bytes. Um conceito relacionado é a unidade de transferência. Para a memória interna, a unidade de transferência é igual ao número de linhas elétricas que chegam e que saem do módulo de memória. Isso pode ser igual ao tamanho da palavra, mas em geral é maior, como 64, 128 ou 256 bytes. Para esclarecer esse ponto, considere três conceitos relacionados à memória interna:

Palavra: é a unidade natural de organização da memória, geralmente correspondendo ao número de bits usados para representar um inteiro e ao tamanho da instrução. No entanto, há exceções. Por exemplo, o CRAY C90 possui palavras de 64 bits, mas inteiros de 46 bits, enquanto a arquitetura Intel x86 apresenta instruções de tamanhos variados, baseadas em múltiplos de bytes, com palavra de 32 bits.

Unidades endereçáveis: em alguns sistemas, a unidade endereçável é a palavra. Porém, muitos sistemas permitem o endereçamento no nível de byte. De qualquer forma, o relacionamento entre o tamanho em bits A de um endereço e o número N de unidades endereçáveis é $2^A = N$.

Unidade de transferência: para a memória principal, este é o número de bits lidos ou escritos na memória de uma só vez. A unidade de transferência não precisa ser igual a uma palavra ou uma unidade endereçável. Para a memória externa, os dados em geral são transferidos em unidades muito maiores que uma palavra, e estas são chamadas de blocos.

Método de acesso: acesso sequencial: os dados são organizados em registros e acessados em ordem linear, com tempo variável — típico das fitas magnéticas, acesso direto: combina acesso físico e busca sequencial até o dado desejado; o tempo também é variável — usado em discos rígidos, acesso aleatório: cada posição tem um endereço próprio e pode ser acessada diretamente em tempo constante — comum na memória principal e cache. acesso associativo: semelhante ao aleatório, mas localiza dados com base em parte do seu conteúdo, não pelo endereço — usado em memórias cache avançadas.

7.1 A hierarquia de memória.

As restrições de projeto da memória de um computador envolvem três questões principais: quantidade, velocidade e custo. A quantidade está relacionada à ca-

pacidade total disponível; quanto maior, mais aplicações podem ser executadas. A velocidade determina o desempenho do sistema, pois a memória deve acompanhar o processador para evitar atrasos na execução das instruções. Já o custo precisa ser equilibrado, mantendo o preço da memória compatível com o restante dos componentes do sistema.

Esses três fatores estão interligados: memórias mais rápidas são mais caras, memórias com maior capacidade tendem a ser mais lentas, e maior capacidade geralmente implica menor custo por bit. Esse equilíbrio cria um desafio para os projetistas, que precisam combinar desempenho, custo e tamanho de forma eficiente.

A solução encontrada é o uso de uma hierarquia de memória, que combina diferentes tecnologias. Nessa hierarquia, as memórias mais rápidas, menores e caras ficam próximas ao processador, enquanto as mais lentas, maiores e baratas ficam em níveis inferiores. À medida que se desce na hierarquia, o custo por bit diminui, a capacidade aumenta, o tempo de acesso cresce e a frequência de uso pelo processador diminui. Essa organização torna o sistema mais eficiente, equilibrando velocidade e custo, conceito que se aplica tanto à memória cache quanto à memória virtual.

Figura 13: -A hierarquia de memória.



Fonte: STALLINGS, ed.10. 2018, p. 103.

O uso de dois níveis de memória para reduzir o tempo médio de acesso funciona em princípio, mas somente se as condições (a) a (d) se aplicarem. Empregando diferentes tecnologias, existe um espectro de sistemas de memória que satisfaz às condições (a) a (c). Felizmente, a condição (d) também costuma ser válida. A base para a validade da condição (d) é um princípio conhecido como localidade de referência (DENNING, 1968).

Durante a execução de um programa, as referências de memória pelo processador, para instruções e para dados, tendem a se agrupar. Os programas em geral contêm uma série de loops iterativos e sub-rotinas. Quando um loop ou sub-rotina inicia sua execução, existem referências repetidas a um pequeno conjunto de instruções. De modo semelhante, operações sobre tabelas e arrays envolvem o acesso a um conjunto de palavras de dados agrupadas. Após um longo período os conjuntos mudam, mas para um pequeno período de tempo o processador trabalha com conjuntos fixos de referências à memória.

Memória Interna: as memórias semicondutoras podem ser do tipo RAM ou ROM. As memórias RAM podem ser lidas ou escritas e são voláteis. A memória ROM é apenas para leitura, mas é permanente. As memórias RAM podem ser estáticas ou dinâmicas. Suas características comuns são: acesso randômico, isto é, as posições de memória podem ser lidas/escritas em qualquer ordem, e são voláteis, isto é, o armazenamento é temporário; quando se desliga a energia os dados são perdidos. A figura abaixo mostra um esquema de uma memória RAM dinâmica e estática.

Memória RAM Estática: Os bits são armazenados em chaves ON/OFF (flip-flops). Os dados permanecem até o computador ser desligado, não precisando ser refrescadas. Sua construção é complexa e ocupa uma grande área, por isso é mais cara. A vantagem é que são muito rápidas, sendo apropriadas para construção de memória cache.

Memória RAM Dinâmica: os bits em uma memória dinâmica são armazenados em capacitores que se decarregam com o tempo, por isso é necessário refreshar o valor armazenado (refresh). Apesar da necessidade de um circuito especial para realizar o refresh, a construção de memória dinâmica é simples, ocupa menos espaço e é barata, permitindo a construção de módulos grandes de memória. No entanto, elas são mais lentas que a memória estáticas, sendo mais adequadas para construir a memória principal. O circuito de refresh é geralmente incluso no próprio chip e consiste em reescrever os dados para regenerar as informações. Por essa razão, há uma queda do desempenho do sistema.

Memória ROM: As memórias ROM tem armazenamento permanente e são usadas para guardar microprogramas e BIOS.

As memórias ROM podem ser:

- Gravadas na fabricação: cara para lotes pequenos.
- PROM: são programadas apenas uma vez.
- EPROM: são programadas eletricamente e apagadas por luz UV (ultravioleta).
- EEPROM: são programadas e apagadas eletricamente (apaga um byte por vez).
- FLASH: são programadas e apagadas eletricamente (apaga toda memória ou um bloco).

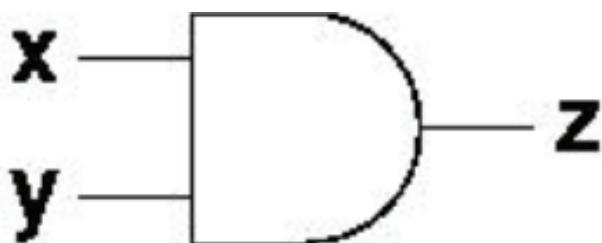
7.2 Portas Lógicas Básicas.

Os blocos básicos de uma arquitetura de computador são as portas lógicas ou apenas portas. As portas são os circuitos básicos que têm, pelo menos, uma (geralmente mais) entrada e exatamente uma saída. Os valores da entrada e da saída são os valores lógicos verdadeiro ou falso. Na convenção de computação é comum usar 0 para falso e 1 para verdadeiro. As portas não têm nenhuma memória. O valor da saída depende somente do valor atual das entradas. Esse fato torna possível usar uma tabela de verdade para descrever inteiramente o comportamento de uma porta. Para representar matematicamente a lógica digital, utilizamos a Álgebra de Boole.

Portas básicas: portas básicas são portas fundamentais que podem formar outras portas com funções especiais. Nós consideramos geralmente três tipos básicos das portas: as portas E, as portas OU e as portas NÃO (ou inversores). Também é usual chamar as portas pelo termo em inglês, respectivamente, AND, OR e NOT.

Porta E (AND): uma porta E pode ter um número arbitrário de entradas. O valor da saída é 1 se e somente se todas as entradas são 1. Se não, o valor da saída é 0. O nome foi escolhido porque a saída é 1 se e somente se a primeira entrada e a segunda entrada, e,..., e a n-ésima entrada são 1. É frequentemente útil desenhar diagramas das portas e das suas interconexões. Em tais diagramas, a porta E é desenhada como:

Figura 14: Porta E (AND).

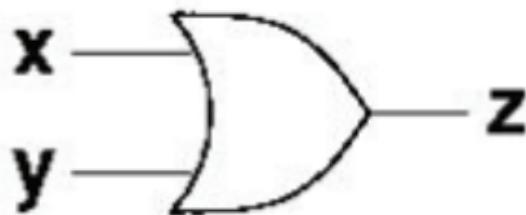


Fonte: Porto Fernandez, ed. 3, 2015, p. 30.

Porta OU (OR): como a porta E, a porta OU pode ter um número arbitrário de entradas. O valor da saída é 1 se e somente se pelo menos de um dos valores

da entrada é 1. Caso contrário, a saída é 0. Ou seja, o valor da saída é 0 somente se todas as entradas são 0. O nome foi escolhido porque a saída é 1 se e somente se a primeira entrada ou a segunda entrada, ou,..., ou a n-ésima entrada são 1. Em esquemas de circuito, desenha-se a porta OU como:

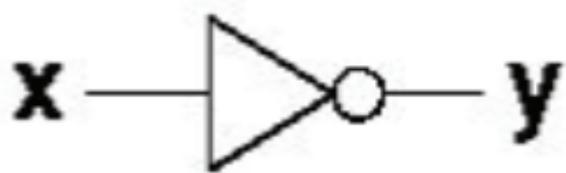
Figura 15: Porta OU (OR).



Fonte: Porto Fernandez, ed. 3, 2015, p. 30.

Porta NÃO (NOT): a porta NÃO também é conhecida como inversor e tem exatamente uma entrada e uma saída. O valor da saída é 1 se e somente se a entrada é 0. Se não, a saída é 0, ou seja, o valor da saída é exatamente o oposto do valor da entrada. Em esquemas de circuito, nós desenhamos a porta NÃO como:

Figura 16: Porta NÃO (NOT).



Fonte: Porto Fernandez, ed. 3, 2015, p. 31.

Portas compostas: às vezes, é prático combinar funções das portas básicas em portas mais complexas a fim reduzir o espaço em diagramas de circuito. Nesta seção, nós mostraremos algumas portas compostas junto com suas tabelas de verdade.

8 Unidade Central de Processamento (UCP).

A Unidade Central de Processamento (UCP), ou Central Processing Unit (CPU), é o módulo principal da arquitetura de um computador. Ela é responsável por realizar o controle do sistema de computador, além de executar funções de cálculo aritmético. O controle do sistema é realizado através da execução de um programa, isto é, uma sequência de comandos previamente definidos necessários para realizar um processamento. Desde os primeiros computadores, sempre houve uma UCP, implementada

em um gabinete ou em uma placa. Mas, atualmente, quando falamos em UCP, geralmente estamos nos referindo ao circuito integrado que realiza todas essas funções – o microprocessador.

A evolução tecnológica dos microprocessadores é surpreendente mente grande. A partir de microprocessadores que trabalhavam com relógio (clock) de alguns kHz e que podiam processar poucos milhares de instruções por segundo, atualmente atingiu-se relógios da ordem de 4GHz com poder de processamento de vários bilhões de instruções por segundo. A complexidade dos circuitos integrados também cresceu: de alguns milhares de transístores para centenas de milhões de transístores em um chip. A UCP tem como função principal integrar todo o sistema do computador, isto é, realiza o controle de funcionamento de todas as unidades funcionais e é responsável pelo controle da execução de todos os programas do sistema.

8.1 Arquitetura de uma Unidade Central de Processamento (UCP).

A função fundamental da Unidade Central de Processamento (UCP), não obstante o formato físico e a arquitetura, é executar uma sequência de instruções armazenadas chamada programa. O programa é representado por uma série de números que são mantidos em algum tipo de memória no computador. Há quatro etapas que quase todos os processadores centrais executam em sua operação: a busca de instruções, a decodificação, a execução e escrita de dados. A figura ao lado mostra o diagrama em blocos de uma Unidade Central de Processamento (UCP).

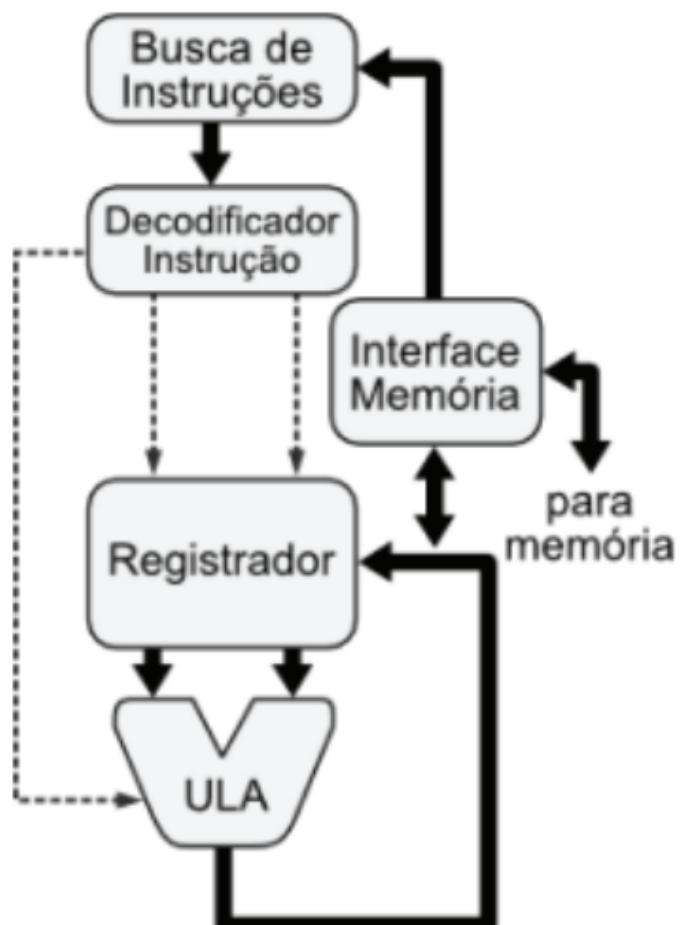
O ciclo de execução de uma instrução em um processador envolve diversas etapas interligadas. Primeiramente, ocorre a busca da instrução na memória, cujo endereço é controlado pelo Contador de Programa (PC), responsável por indicar a posição da próxima instrução a ser executada. Após a busca, o PC é incrementado para apontar para a próxima instrução. Como a memória costuma ser mais lenta que o processador, utiliza-se a memória cache, que armazena instruções em uma área mais rápida, acelerando a execução. Em seguida, na etapa de decodificação, a instrução é dividida em partes e interpretada de acordo com o Conjunto de Instruções (ISA). O opcode identifica a operação a ser realizada, enquanto os demais campos indicam os operandos e seus endereços. Processadores modernos podem usar microprogramas, que decompõem instruções complexas em operações mais simples, permitindo maior flexibilidade.

Na fase de execução, a Unidade Lógica e Aritmética (ULA) realiza as operações determinadas, manipulando dados de registradores e gerando resultados. Caso o resultado ultrapasse a capacidade do processador, um flag é ativado para indicar excesso aritmético. Depois, ocorre a escrita dos dados, quando o resultado é armazenado em um registrador interno ou na memória. Algumas instruções, chamadas

saltos, modificam o PC para alterar a sequência de execução, possibilitando laços, condições e chamadas de funções. O ciclo então se repete continuamente, podendo ser otimizado pelo uso de pipeline, técnica que permite a execução simultânea de várias instruções em diferentes estágios.

Os processadores são síncronos, operando de acordo com um sinal de relógio (clock), que define o ritmo das operações. Esse sinal deve ser ajustado para garantir que todos os componentes trabalhem de forma sincronizada, evitando erros de temporização. Contudo, taxas de clock mais altas aumentam o calor dissipado, exigindo sistemas de refrigeração eficientes. Assim, embora o aumento da frequência do clock melhore o desempenho, ele também traz desafios relacionados à sincronização, consumo de energia e aquecimento, o que levou ao desenvolvimento de técnicas de paralelismo e otimização arquitetural nos processadores modernos.

Figura 17: - Diagrama em blocos de uma Unidade Central de Processamento



Fonte: Porto Fernandez, ed. 3, 2015, p. 60.

9 Tipos de sistemas operacionais.

Certas características básicas servem para diferenciar diversos tipos de sistemas operacionais. As características espalham-se por duas dimensões independentes. A primeira dimensão especifica se o sistema é em lote ou interativo. Em um sistema interativo, o usuário/programador interage diretamente com o computador, normalmente por meio de um terminal com teclado/vídeo, para solicitar a execução de uma tarefa ou realizar uma transação. Além do mais, dependendo da natureza da operação, o usuário pode comunicar-se com o computador durante a execução do job. Um sistema em lote é o oposto do interativo. O programa de um usuário é mantido junto com programas de outros usuários e submetido por um operador de computador. Depois que o programa termina, os resultados são impressos para o usuário. Os sistemas puramente em lote são raros hoje em dia. No entanto, será útil para a descrição dos sistemas operacionais atuais examinar os sistemas em lote rapidamente. Uma dimensão independente especifica se o sistema emprega multiprogramação ou não. Com a multiprogramação, tenta-se manter o processador o mais ocupado possível, fazendo-o trabalhar em mais de um programa de cada vez. Vários programas são carregados na memória e o processador alterna rapidamente entre eles. A alternativa é um sistema de uniprogramação, que trabalha apenas com um programa de cada vez.

SISTEMAS ANTIGOS: Nos primeiros computadores, entre o final da década de 1940 e meados dos anos 1950, não existiam sistemas operacionais, e o programador interagia diretamente com o hardware por meio de consoles compostos por lâmpadas, chaves, dispositivos de entrada e impressoras. Os programas eram inseridos manualmente, geralmente por leitores de cartões perfurados, e qualquer erro era sinalizado pelas lâmpadas do painel. O programador, então, precisava analisar os registradores e a memória principal para identificar o problema. Caso o programa fosse executado com sucesso, o resultado era impresso diretamente.

Esses sistemas apresentavam duas grandes limitações. A primeira era o escalonamento, pois o uso do computador era agendado manualmente — os usuários reservavam períodos fixos, o que frequentemente resultava em desperdício de tempo ocioso ou interrupções prematuras. A segunda limitação era o tempo de preparação, já que a execução de um programa envolvia várias etapas demoradas, como carregar compiladores, montar fitas e preparar cartões. Qualquer erro obrigava o usuário a reiniciar todo o processo, tornando o trabalho lento e ineficiente.

Esse tipo de operação, conhecido como processamento serial, caracterizava-se pelo uso sequencial do computador por um único usuário por vez. Com o tempo, surgiram ferramentas de software que tornaram o processo mais eficiente, como bibliotecas de funções, editores de ligação (linkers), carregadores, depuradores e rotinas

de entrada e saída (E/S), que passaram a compor o software básico disponível para todos os usuários.

SISTEMAS EM LOTE SIMPLES: O monitor controlava todo o processo: permanecia parcialmente residente na memória principal e lia os jobs um por um, executando-os em sequência. Quando um job terminava, o controle voltava automaticamente ao monitor, que iniciava o próximo. Assim, o problema do escalonamento era resolvido, já que o processador permanecia ocupado continuamente, sem tempo ocioso. Além disso, o monitor utilizava a linguagem de controle de jobs (JCL), que permitia incluir instruções especiais junto ao programa do usuário, como comandos para compilar, carregar e executar o código. Essa automação reduzia significativamente o tempo de preparação e a necessidade de intervenção manual.

O sistema em lote também introduziu novos recursos de hardware essenciais para o funcionamento seguro e eficiente do monitor. A proteção de memória impedia que programas do usuário alterassem as áreas onde o monitor estava armazenado. O timer evitava que um único job monopolizasse o processador, interrompendo-o quando o tempo limite era atingido. Já as instruções privilegiadas restringiam certas operações — como as de entrada e saída (E/S) — apenas ao monitor, garantindo o controle sobre os dispositivos e evitando que programas interferissem uns nos outros. Por fim, o uso de interrupções permitia que o sistema operacional alternasse o controle entre o monitor e os programas de usuário de forma mais eficiente.

Embora o sistema em lote simples exigisse uma parte da memória e do tempo do processador para o funcionamento do monitor — representando uma certa sobrecarga —, ele aumentou significativamente a utilização do computador, tornando o processamento mais rápido, contínuo e organizado em comparação aos métodos manuais anteriores.

Figura 18: -Layout de memória para um monitor residente.



Fonte: STALLINGS, ed.10. 2018, p. 238.

10 Conclusão.

A evolução das cinco gerações de computadores evidencia o avanço contínuo da tecnologia, desde os sistemas baseados em válvulas eletrônicas até os atuais voltados à inteligência artificial e computação em nuvem. Cada geração trouxe inovações significativas: a primeira com válvulas e programas armazenados por Von Neumann, a segunda com transistores, linguagens de alto nível e sistemas de lotes, a terceira com circuitos integrados e miniaturização dos computadores, aumentando velocidade, confiabilidade e reduzindo custos. Esses progressos não apenas transformaram a forma como os computadores eram projetados e utilizados, mas também abriram caminho para o desenvolvimento de novas linguagens, sistemas de armazenamento e arquite-

turas mais eficientes. O estudo dessas gerações mostra como a ciência e a engenharia da computação evoluíram, moldando a sociedade contemporânea e preparando o terreno para as tecnologias futuras.

Referências

- 1 STALLINGS, William. *Arquitetura e organização de computadores*. 10. ed. Tradução de Sérgio Nascimento. São Paulo: Pearson Education do Brasil, 2018. ISBN 978-85-430-2053-2.
- 2 STALLINGS, William. *Arquitetura e organização de computadores*. 8. ed. São Paulo: Pearson Education do Brasil, 2010. ISBN 978-85-7605-564-8.
- 3 FERNANDEZ, Marcial Porto. *Arquitetura de computadores*. 3. ed. Fortaleza, CE: UECE, 2015.
- 4 AMARAL, Allan Francisco Forzza. *Arquitetura de computadores*. Colatina, ES: Instituto Federal Espírito Santo, 2010.