

**Universidade de Itaúna**  
**Arquitetura e Organização de Computadores I**

**Grupo:**

Davi Ventura Cardoso Perdigão - 82148

Edmilson Lino Cordeiro - 82920

Pedro Otávio Marques Silva - 82132

**Professor (a):**

Adriana Dornas Pereira

**Modelo de Von Neumann**

Itaúna - MG  
Setembro, 2020.

# História

---

Nos primórdios da Computação, não existia o conceito de programa armazenado, isto significa que os programas eram desenvolvidos, executados e apresentavam os resultados imediatamente.

O conceito de programa armazenado surgiu da necessidade de armazenar programas em um computador, pois, até então, ainda não havia formas de armazenar programas em um computador. John Von Neumann e outros pesquisadores descobriram que, utilizando dispositivos de memória em formas de linha de retardo de mercúrio, poderiam armazenar instruções de programas.

Atualmente esse conceito de programa armazenado já está muito bem edificado. Nós codificamos, desenvolvemos software em diversas linguagens de programação, armazenamos nosso código fonte em um HD Externo, ou no HD interno, ou na Nuvem se assim desejarmos. O código executável desses softwares compilado, que está armazenado em algum desses dispositivos, sempre passa pela memória principal, depois pela memória cache e então é executado na CPU.

O conceito de programa armazenado surgiu durante o projeto EDVAC e John Von Neumann acabou recebendo a maior parte dos créditos pela definição, mas a verdade é que várias pessoas participaram da construção. John Von Neumann ganhou mais destaque que os outros por ter escrito um relatório sobre o EDVAC, com 101 páginas, descrevendo formalmente o conceito, criando assim uma espécie de guia para a construção de computadores.

## A arquitetura de Von Neumann

---

A proposta inicial de um computador de programa armazenado, relatada pelo próprio Von Neumann, era dividida em cinco partes:

- central aritmética;
- central de controle;
- memória;
- gravação;
- entrada e saída.

**Primeira Parte Específica – Central Arithmetic (CA):** O dispositivo deve realizar as operações elementares da aritmética mais frequentemente, e por este motivo deve ter unidades especializadas apenas para essas operações.

**Segunda Parte Específica – Control Center (CC):** A sequenciação apropriada das operações pode ser executada, de forma mais eficiente, por um controle central.

### Terceira Parte Específica:

*Memória (M)*: Qualquer dispositivo que tiver que executar longas e complicadas seqüências de operações precisa ter uma memória considerável.

*Recording (R)*: Deve ser capaz de manter contato com a entrada e a saída, que são como os neurônios correspondentes sensoriais e motores do cérebro humano.

**Quarta Parte Específica – Input (I)**: É necessário ter unidades para transferência de informações de R para M.

**Quinta Parte Específica – Output (O)**: É necessário ter unidades para transferência de informações de M para R.

## Organização da CPU e seus componentes

---

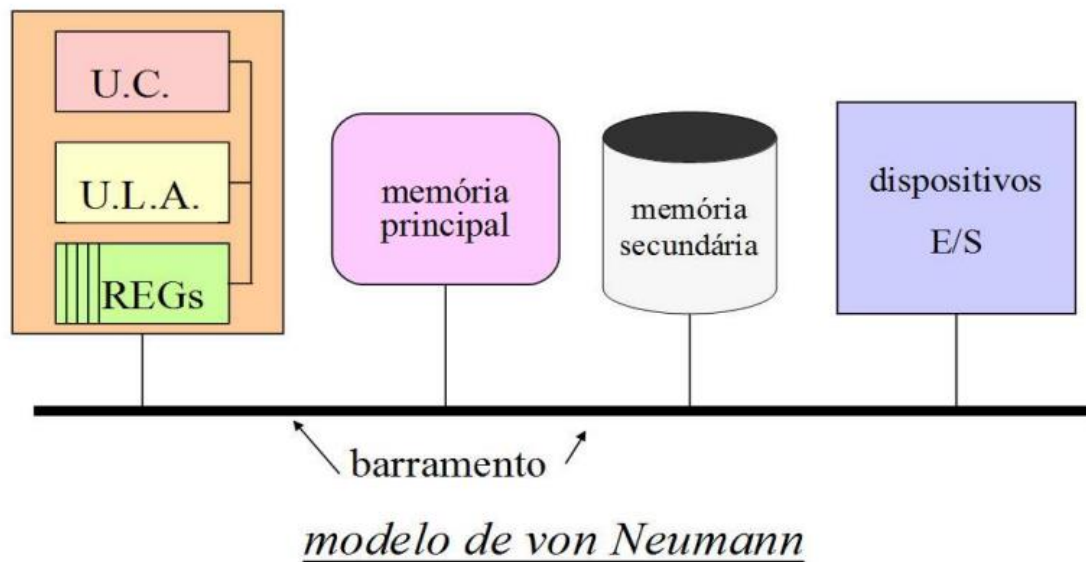


Figura 1 – Estrutura fundamental de um computador. Fonte: BESSA, 2008, p.8.



- **Registrador de instrução (RI):** guarda a instrução que deve ser interpretada e executada.
- **Contador de instruções (CI) - PC (Program Counter):** guarda o endereço da próxima instrução a ser executada.
- **Registradores de uso geral:** guardam diversos tipos de dados.
- **Acumulador:** armazena os resultados de um cálculo.
- **Unidade Lógica e Aritmética (ULA):** É um aglomerado de circuitos lógicos e componentes eletrônicos que, integrados, realizam as operações lógicas e aritméticas na CPU.

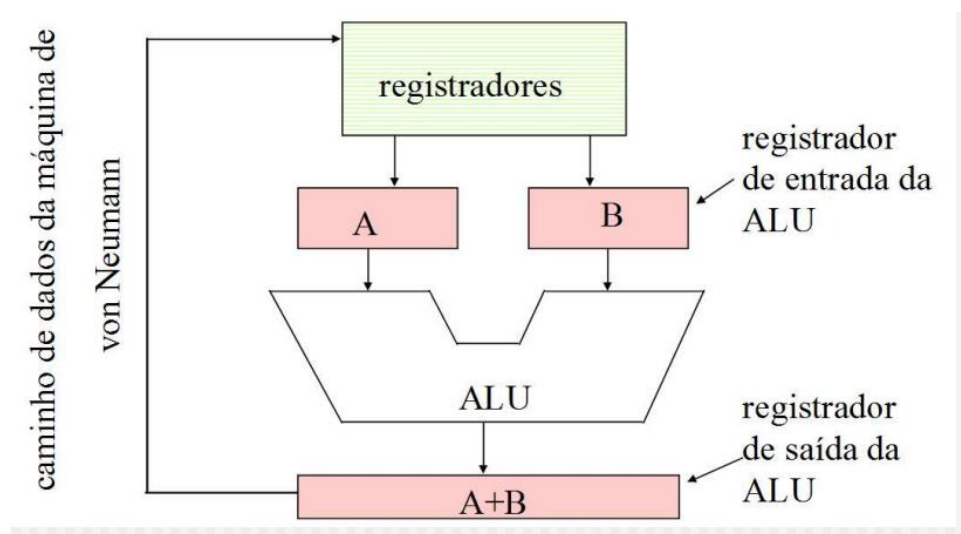


Figura 3 – Caminho de dados de máquina de Von Neumann. Fonte: BESSA, 2008, p.25.

## Ciclo de Instrução

Toda CPU trabalha em dois ciclos principais, o Ciclo de Busca e o Ciclo de Execução, como pode ser visto na figura abaixo. Assim que o computador é iniciado, a CPU entra no Ciclo de Busca, em seguida passa para o Ciclo de Execução e depois volta para o Ciclo de Busca. Ela continua nesse processo até que precise ser desligada, saindo do Ciclo de Execução para o estado final.

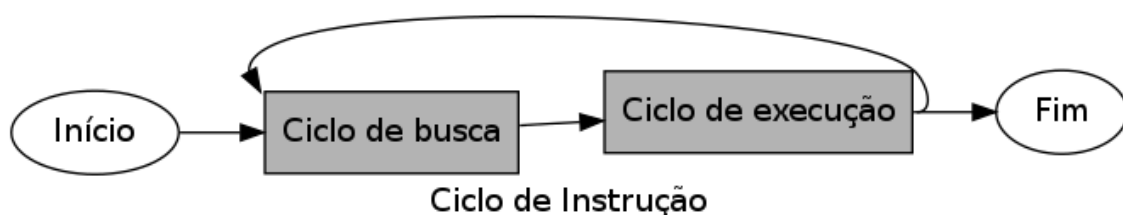


Figura 4 – Ciclo de instrução. Fonte: SANTANA, 2012, p.38.

Durante o Ciclo de Busca, é a Unidade de Controle que atua. Uma nova instrução é buscada da Memória para que possa ser decodificada. Nessa fase o Contador de Programas e o Registrador de Instrução são utilizados, como apresentados na seção anterior. O Contador de Programas é logo lido para se saber que instrução será executada, essa instrução é trazida para o Registrador de Instrução e, finalmente, é decodificada pela Unidade de Controle. Assim que esse processo termina, caso a instrução não diga respeito à um laço, ou à uma repetição, o conteúdo de Contador de Programas é incrementado. Ou seja, Contador de Programas recebe Contador de Programas + 1. Assim, no próximo Ciclo de Busca a instrução do endereço seguinte será carregada da memória e executada. Esse comportamento garante a característica de execução sequencial dos programas.

No passo seguinte a CPU entra em Ciclo de Execução. Nessa etapa atua a Unidade de Ciclo de Dados. Agora a Unidade de Controle já sabe exatamente que operação será executada, com quais dados e o que fazer com o resultado. Essa informação é passada para a Unidade Lógica e Aritmética e os registradores envolvidos. Durante o Ciclo de Execução há cinco possíveis tipos de operação que podem ser executadas:

**1 - Processador e memória:** trata simplesmente da transferência de dados entre CPU e memória principal;

**2 - Processador e Entrada e Saída:** diz respeito à transferência de dados entre a CPU e um dispositivo de Entrada e Saída, como teclado, mouse, monitor, rede, impressora etc.;

**3 - Processamento de Dados:** são operações simplesmente de processamento dos dados, como operação aritmética ou lógica sobre os registradores da CPU;

**4 - Controle:** são instruções que servem para controlar os dispositivos do computador, como para ligar um periférico, iniciar uma operação do disco rígido, ou transferir um dado que acabou de chegar pela Internet para a Memória Principal;

**5 - Operações compostas:** são operações que combinam uma ou mais instruções das outras em uma mesma operação.

## Busca de Dados

---

Em operações entre Processador e Memória, é necessário que dados sejam trazidos da memória para servirem de entrada para a Unidade Lógica e Aritmética, e/ou o resultado seja levado para armazenamento na memória no final da execução. Para isso acontecer, é executada uma Busca de Dados. Isso é determinado durante a decodificação da instrução, no ciclo de Busca de Instrução. Isso acontece quando um dos parâmetros de uma operação aritmética é um endereço de memória, e não um valor diretamente, nem um registrador. Para isso, parte do conteúdo de Registrador de Instrução é transferido para o Registrador de Endereço. Essa parte é justamente o

endereço do parâmetro da instrução. Em seguida a Unidade do Controle requisita à memória uma leitura. Assim, o endereço, agora no Registrador de Endereço, é passado para a memória e o conteúdo lido da memória é passado para o Registrador de Dados. Agora o conteúdo é transferido para a Unidade Lógica e Aritmética para que a operação seja executada (lógica ou aritmética).

Se a instrução tiver dois ou mais parâmetros de memória, serão necessárias outras Buscas de Dados. Como a memória é sempre mais lenta do que a CPU, instruções que necessitam Buscas de Dados são muito mais lentas do que instruções de Processamento de Dados.

Perceba que cada instrução pode exigir mais tempo de execução do que outras. Isso depende de quantos acessos à memória ela exigirá. Quanto mais acessos à memória, mais lenta a instrução. O ideal é sempre usar registradores. Mas nem sempre é possível utilizar registradores. Eles estão sempre em poucas quantidades e em menores tamanhos. Principalmente por serem caros. O que os computadores sempre tentam fazer é passar os dados da memória para os registradores assim que puderem, para que as próximas instruções sejam aceleradas.

## Conclusão

---

A Arquitetura de John Von Neumann é referência nos estudos sobre Microprocessadores e Arquitetura. Por mais que esse tipo de sistema computacional não seja mais utilizado, foi ele o precursor de tudo o que conhecemos hoje. Para podermos avançar e evoluir, é muito importante conhecer o passado, o que nos levou a estar no ponto onde estamos e, o que podemos fazer para melhorar ainda mais.

## Referências Bibliográficas

---

BESSA, GERALDO M. **Organização da CPU (arquitetura de Computadores)**. Universidade Federal do Acre (UFAC).

HENNESSY, JOHAN L. **Arquitetura de Computadores: uma abordagem quantitativa**. Rio de Janeiro: Campus, 2003.

NULL, Linda. **Princípios Básicos de Arquitetura e Organização de Computadores**. 2ª ed. Porto Alegre: Bookman, 2010.

PATTERSON, David A.; HENNESSY, John L. **Organização e Projeto de Computadores: A Interface Hardware e Software**. 2ª ed. Rio de Janeiro: LTC, 2000.

REBONATO, MARCELO T. **Organização de Computadores: notas de aula**. Universidade de Passo Fundo.