

## AULA Nº 04 ORGANIZAÇÃO DE COMPUTADORES

### Programa armazenado e organizações

### Conceito de programa armazenado

Desenvolver uma maneira de armazenar os programas na memória, de maneira similar aos dados (armazenados em acumuladores).

**Conceito:**

Atribuído a John von Neumann

Desenvolvida simultaneamente por Alan Turing

### ENIAC (*Electronic Numerical Integrator and Computer*)

1º computador digital eletrônico de grande escala. Foi desenvolvido na Universidade da Pensilvânia, com projeto liderado por John Eckert e John Mauchly e concluído em 1946.

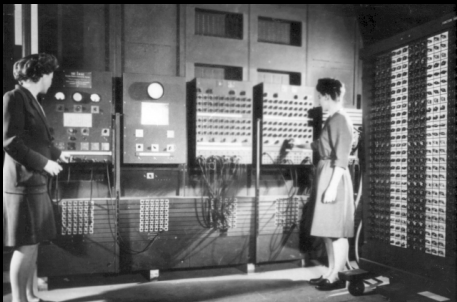
Programar significava **conectar fios, relês e sequências de chaves** para que se determinasse a tarefa a ser executada.

### Modelo de von Neumann

Define **conjunto de princípios e regras de projeto** que permitem projetar um computador digital capaz de executar programas armazenados em uma memória.

Define tanto os **aspectos organizacionais do computador** como também os **aspectos de sua programação**.

### ENIAC



Por United States Army - Domínio público, <https://commons.wikimedia.org/w/index.php?curid=978783>

### Princípios do modelo de von Neumann

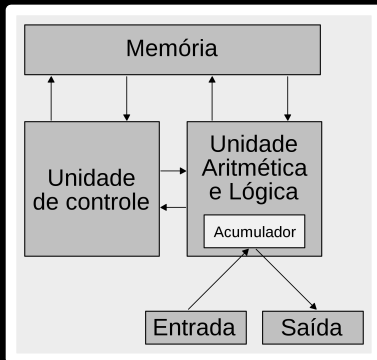
**Simplicidade:** operações elementares sobre operandos elementares

**Linearidade e Uniformidade:** memória linear e uniforme

**Sequencialidade e Centralidade:** processamento sequencial e controle centralizado

**Unicidade:** programa e dados armazenados numa mesma memória

## Modelo



[http://commons.wikimedia.org/wiki/File:Von\\_Neumann\\_architecture.svg](http://commons.wikimedia.org/wiki/File:Von_Neumann_architecture.svg)

## E hoje?

**Arquitetura Harvard** usada em:  
DSPs - *Digital signal processors*  
Microcontroladores

**Processadores modernos** (ARM, x86) :  
Cache nível 1 separado para dados e instruções.  
Memória principal contém dados e instruções, e é acessada através de barramento comum.

## Interpretador de instruções

Estruturado como um *loop* contínuo, ou máquina de estados.

3 estados:

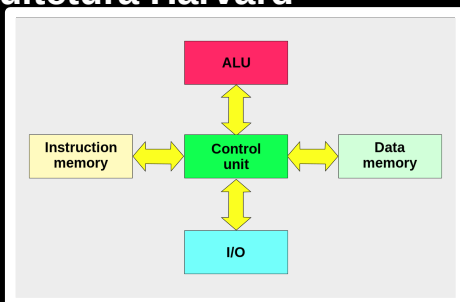
**Busca de instrução**

Cálculo de endereço e busca de **operandos**

**Execução** da instrução e armazenamento de resultados

## Organização do caminho de dados

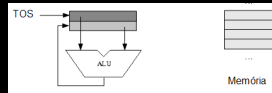
## Arquitetura Harvard



<https://commons.wikimedia.org/w/index.php?curid=10303637>

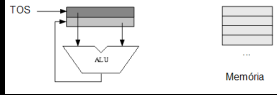
## Organização do caminho de dados

**Pilha**

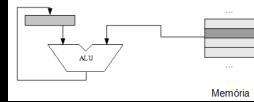


## Organização do caminho de dados

### Pilha



### Acumulador



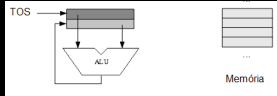
## Exemplo de instruções

Como fica o código  $C = A + B$  ?

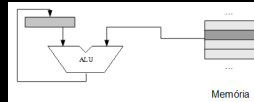
Pilha	Acumulador	Memória / Memória	Registrador / Memória	Load/store
Push A	Load Mem[A]	Add Mem[C], Mem[A], Mem[B]	Load R1, Mem[A]	Load R1, Mem[A]
Push B	Add Mem[B]		Add R1, Mem[B]	Load R2, Mem[B]
Add	Store Mem[C]		Store Mem[C], R1	Add R3, R1, R2
Pop C				Store Mem[C], R3

## Organização do caminho de dados

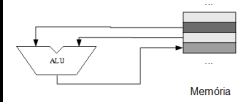
### Pilha



### Acumulador



### Memória-memória



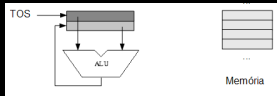
## Referências

Capítulos 1 e 2 - *Arquitetura e organização de computadores*, William Stallings, Editora Pearson, 8ª edição, 2010.

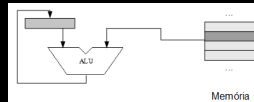
Seção 1.10 - “Organização e Projeto de Computadores – A Interface Hardware/Software”, David A. Patterson & John L. Hennessy, Campus, 4 edição, 2013.

## Organização do caminho de dados

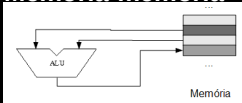
### Pilha



### Acumulador



### Memória-memória



### Registrador-registrador

