



Arquitetura von Neumann

Universidade Federal de Uberlândia
Faculdade de Computação
Prof. Dr. rer. nat. Daniel D. Abdala

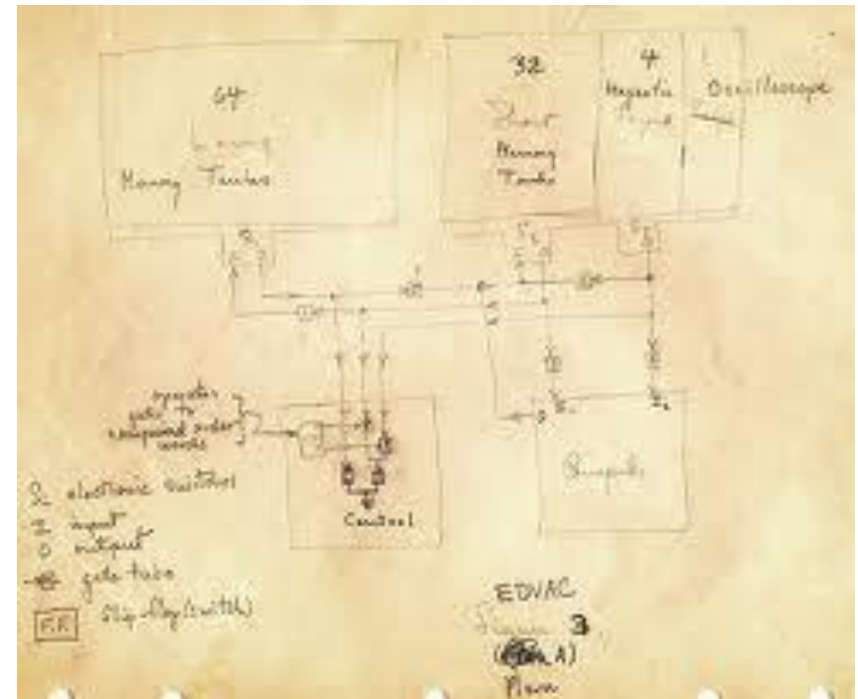
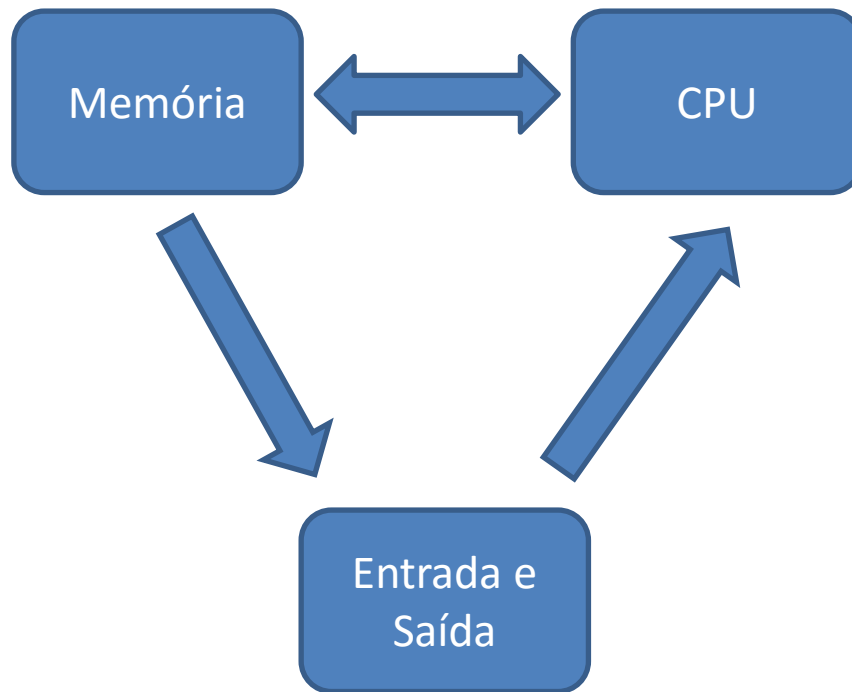
Na Aula Anterior ...

- Histórico da evolução dos computadores:
 - 1ª Geração
 - 2ª Geração
 - 3ª Geração
 - 4ª Geração
- Tendências atuais ...

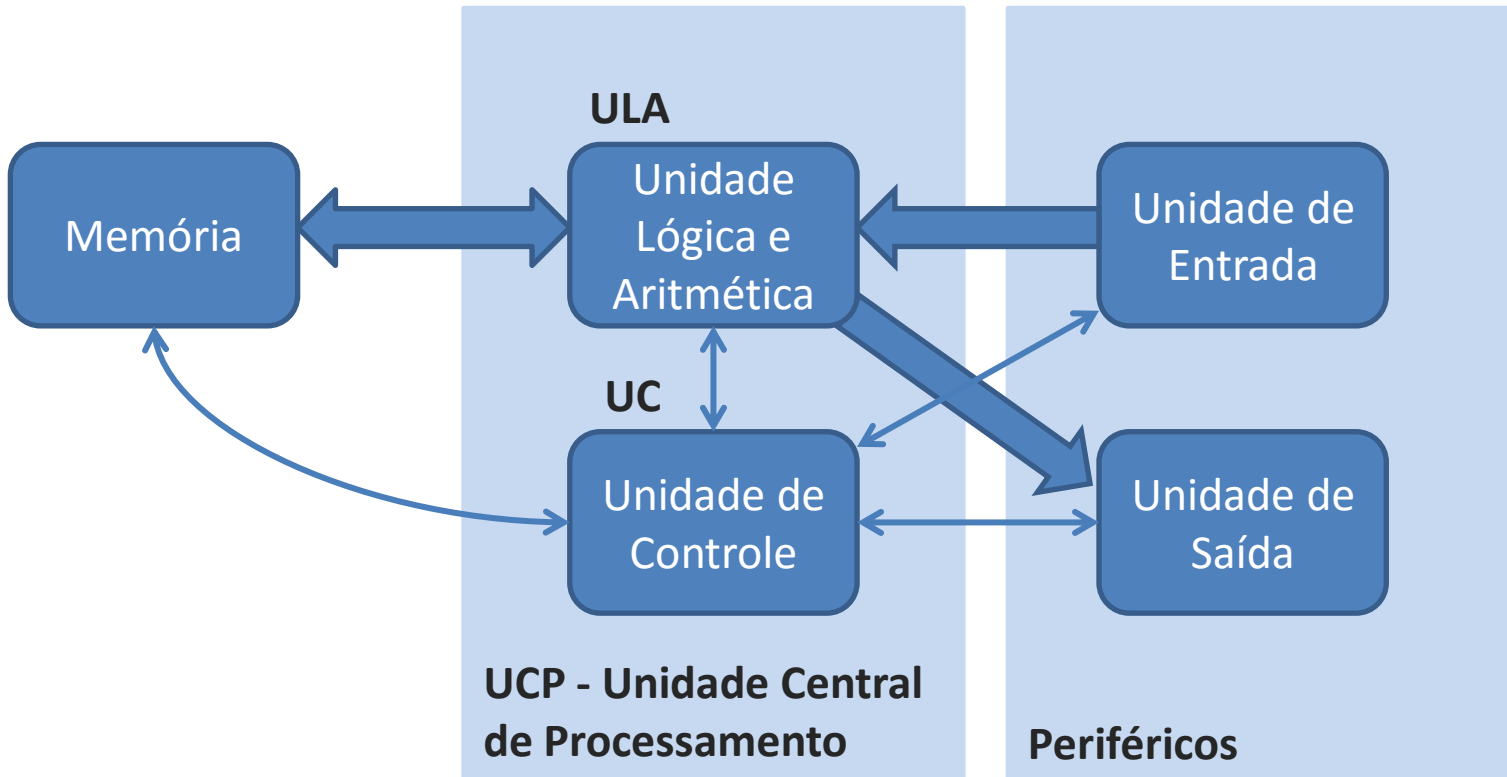
Nesta Aula

- O modelo básico da Arquitetura von Neumann;
- Programa armazenado em memória;
- O Gargalo de von Neumann;
- Abstrações;
- Introdução a Organização de Barramentos.

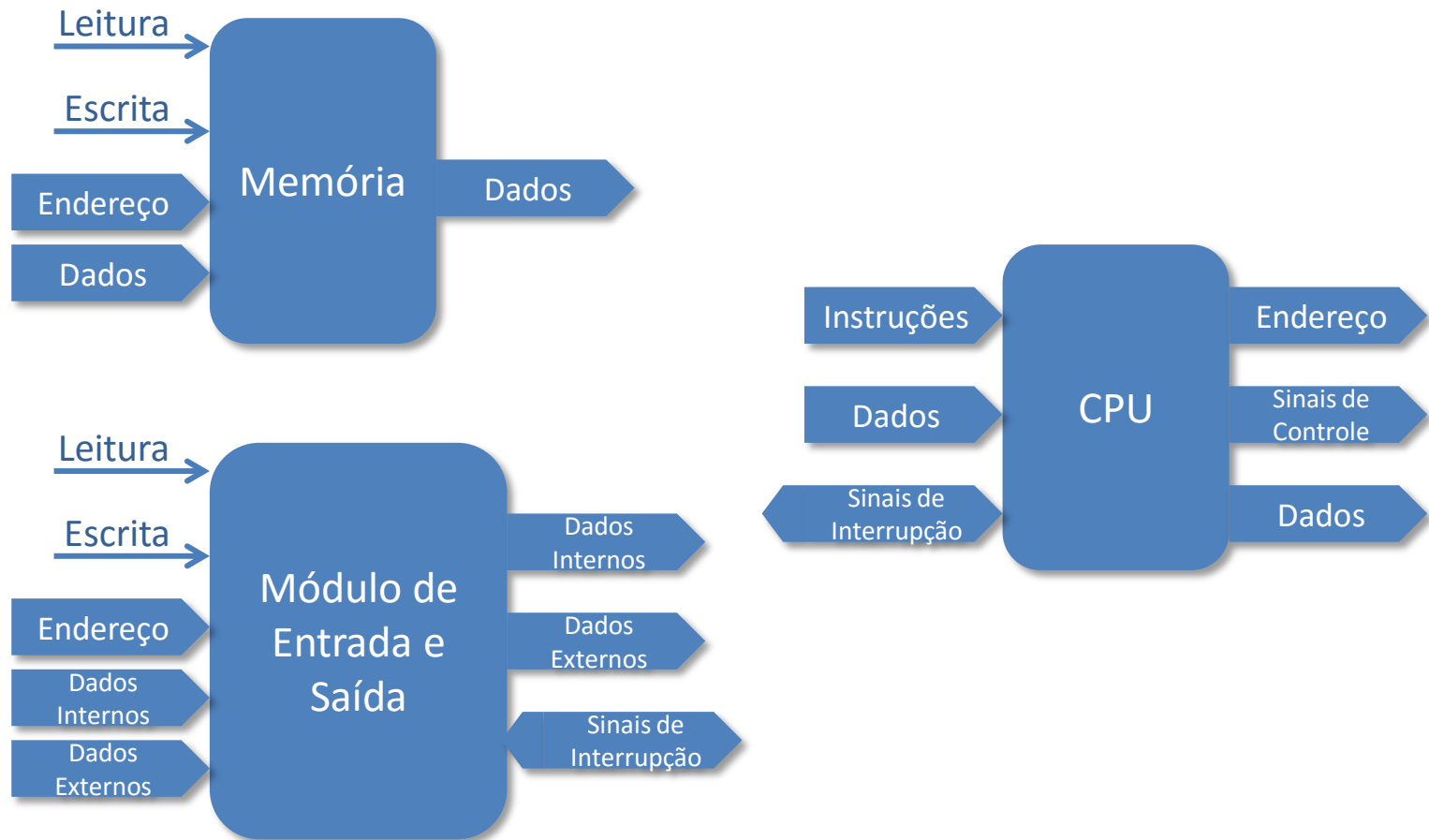
O Modelo von Neumann



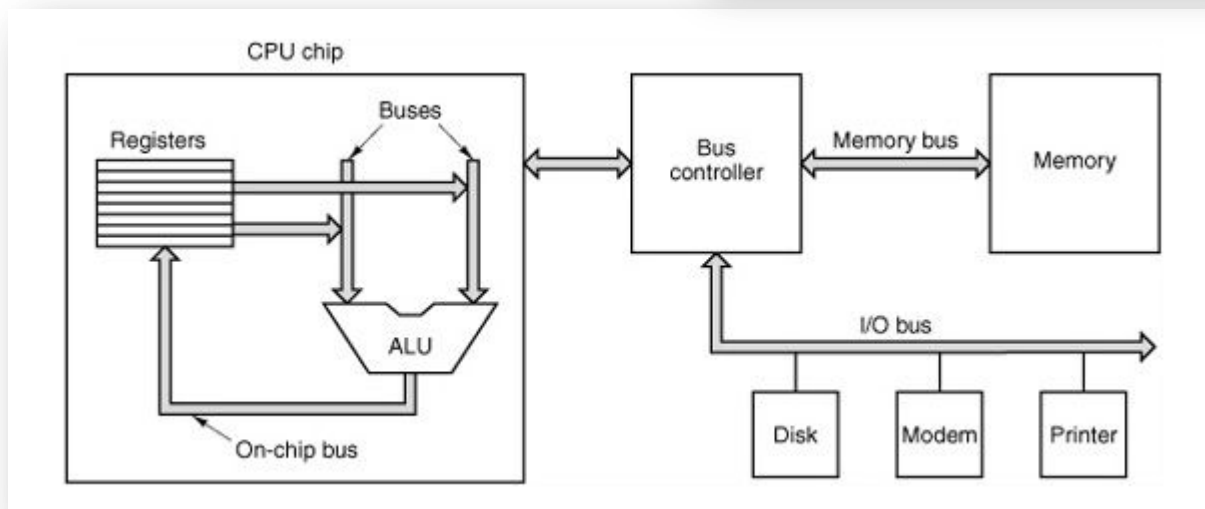
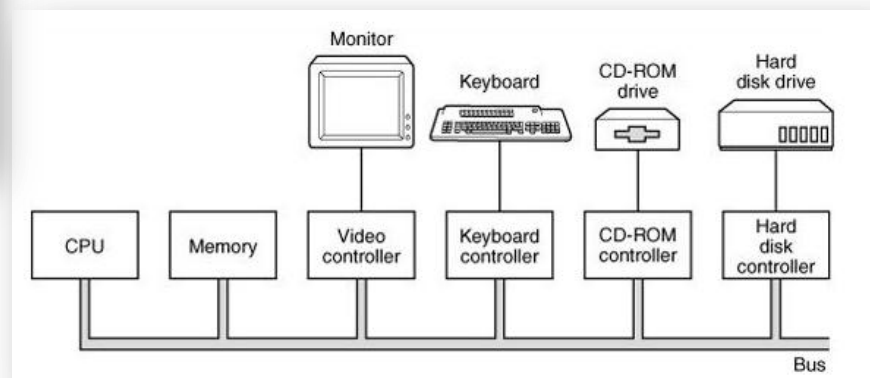
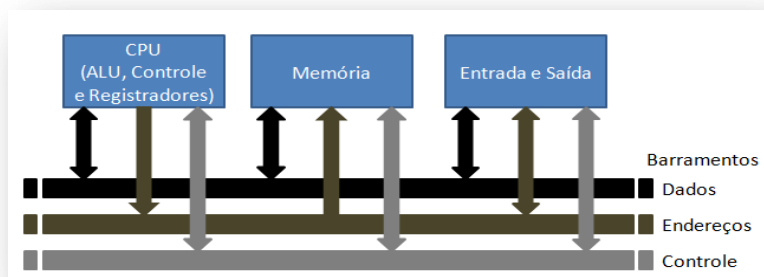
O Modelo von Neumann



Sinais dos Módulos de um Sistema Computacional



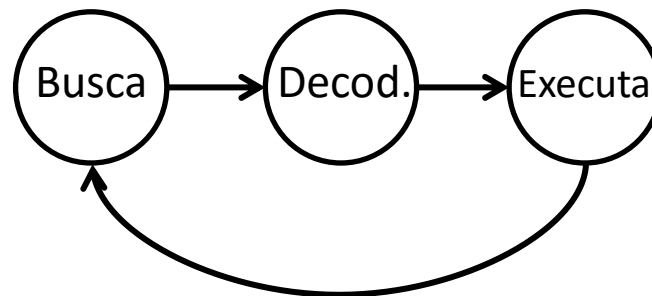
Realização em Sistemas Computacionais



Programa Armazenado em Memória

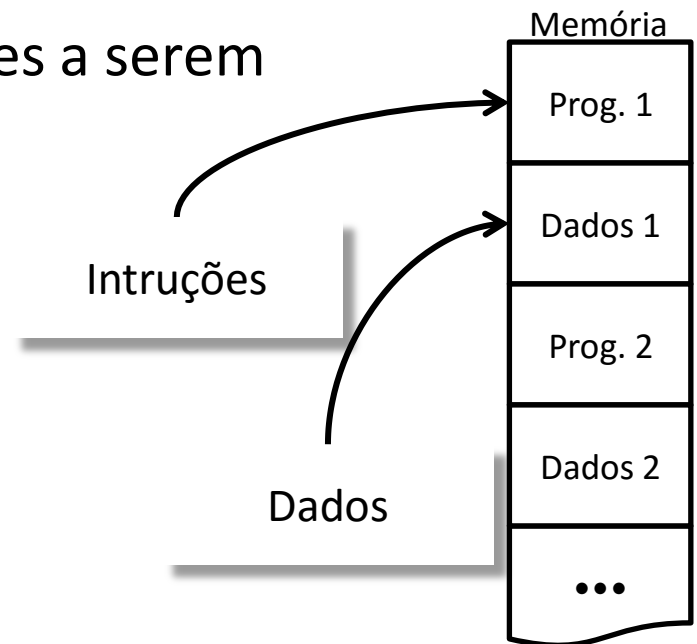
- Ideia fundamental introduzida por von Neumann;
- A memória contém tanto os dados a serem processados quanto as instruções que ditam ao processador como os dados devem ser processados;

Ciclo Básico de Execução de Instruções



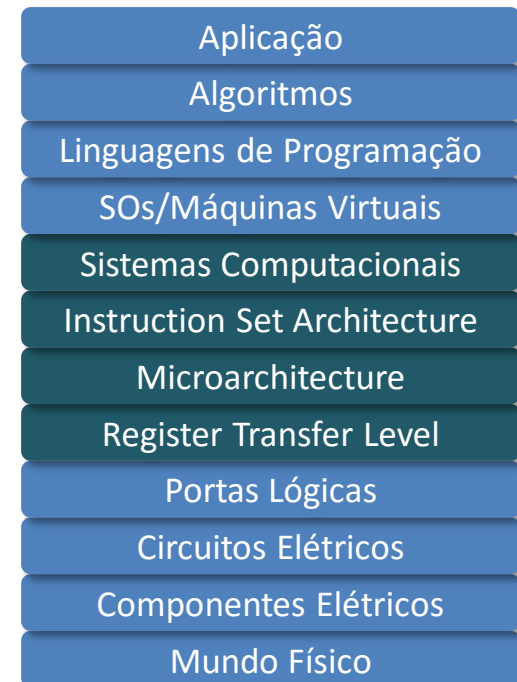
O “Gargalo” de von Neumann

- Também chamada “barreira da memória”;
- O ciclo básico de execução é diretamente dependente da velocidade de acesso da memória;
- Busca de dados atrasa ainda mais;
- CPU starvation → não há instruções a serem executadas;
- Temp. Acc. Mem. >> Ciclo Proc.;
- Como mitigar o problema?
 - Mem. Cache;
 - Arq. Harvard de Mem.;
 - ...



Abstração de Computadores

- O Computador é uma máquina complexa;
- Impossível de lidar com toda a complexidade de uma só vez. Muita informação;
- Solução: Abstrair níveis de complexidade.

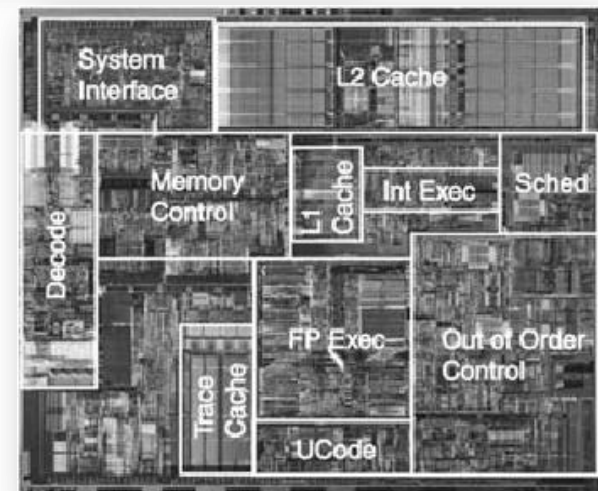
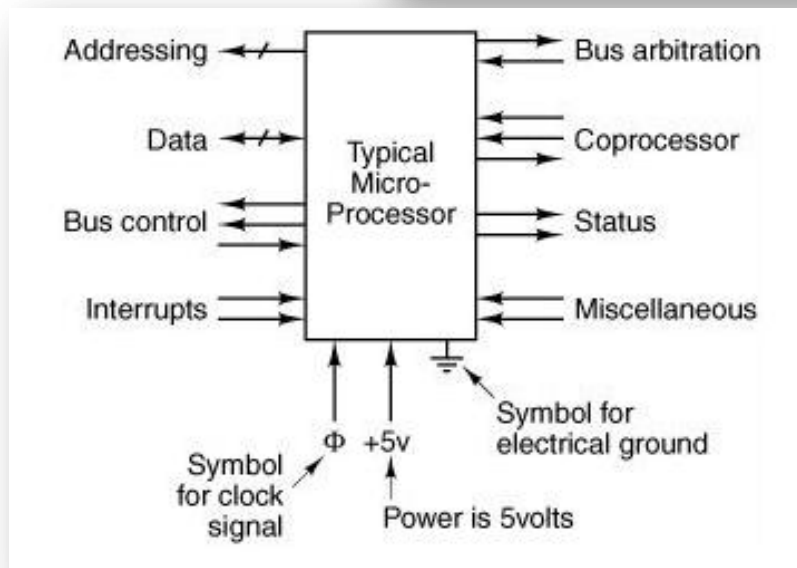
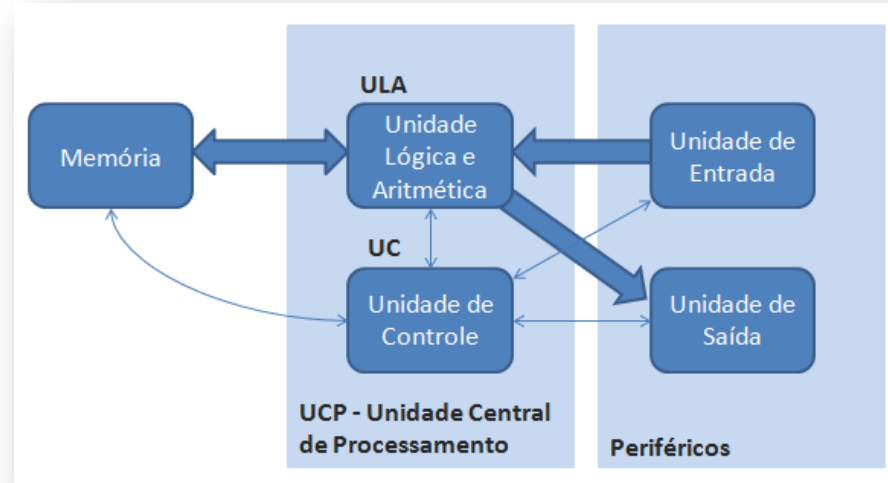


Uma Pilha de Abstrações

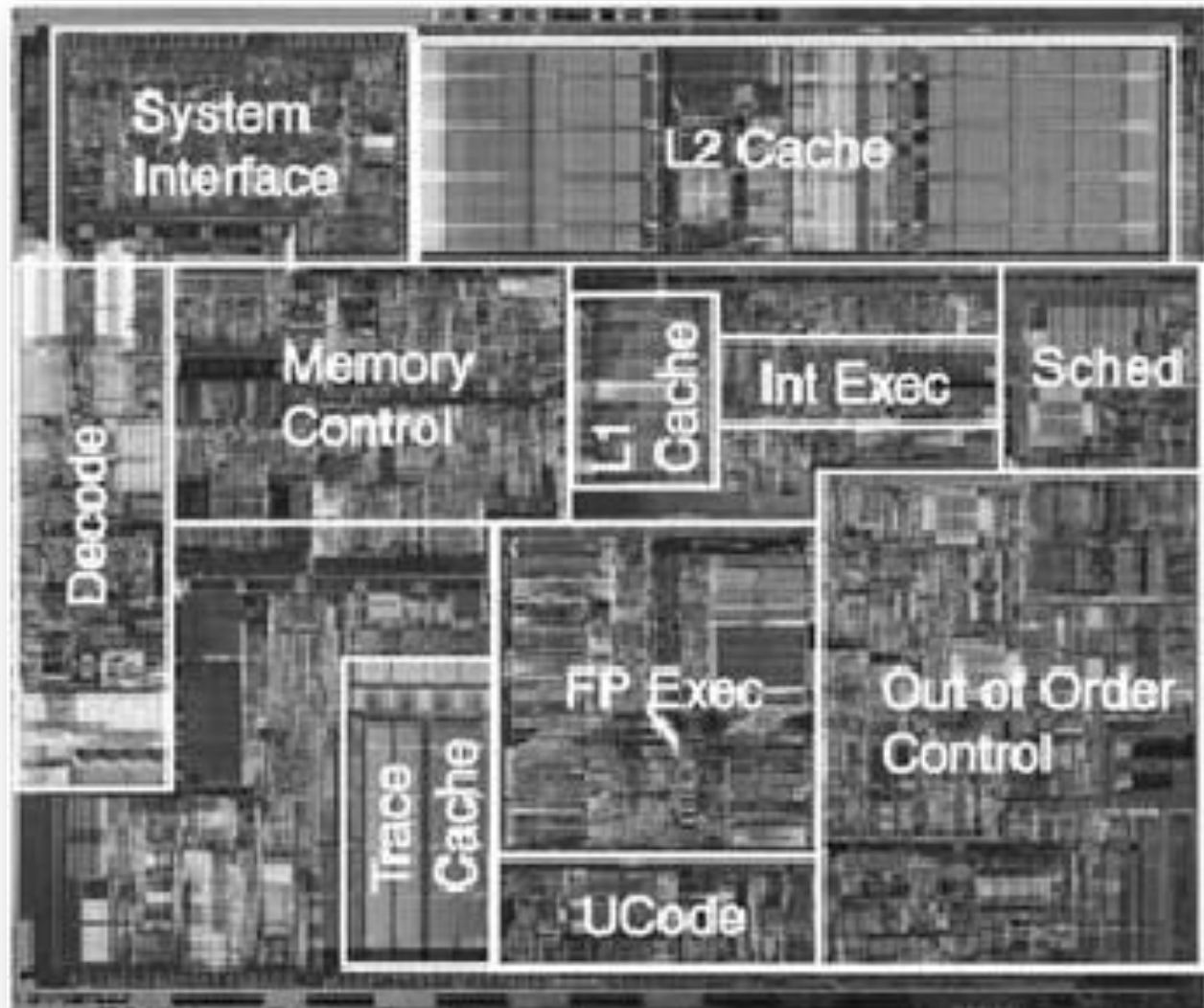
- O processador pode ser percebido de diversas formas;
- Em geral “abstraimos” detalhes e nos concentramos na parte funcional específica que estamos interessados;

Uma visão geral de um processador

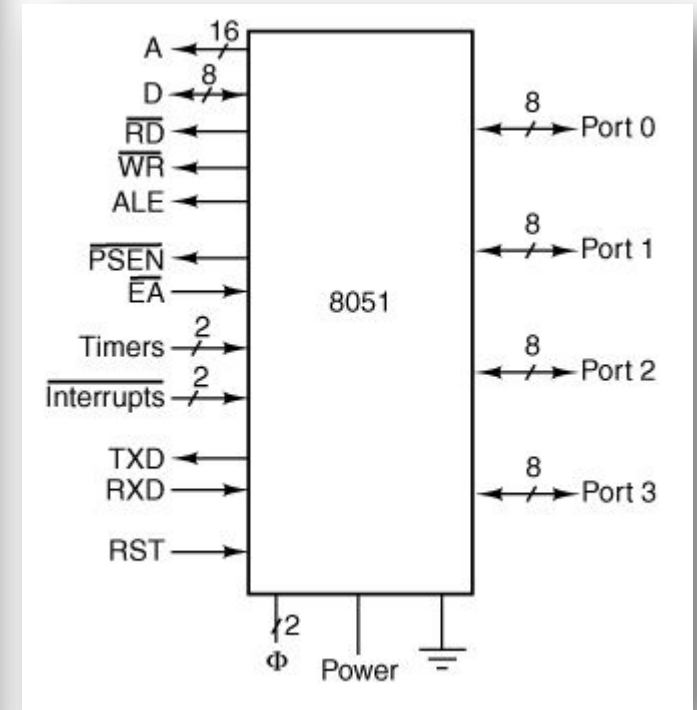
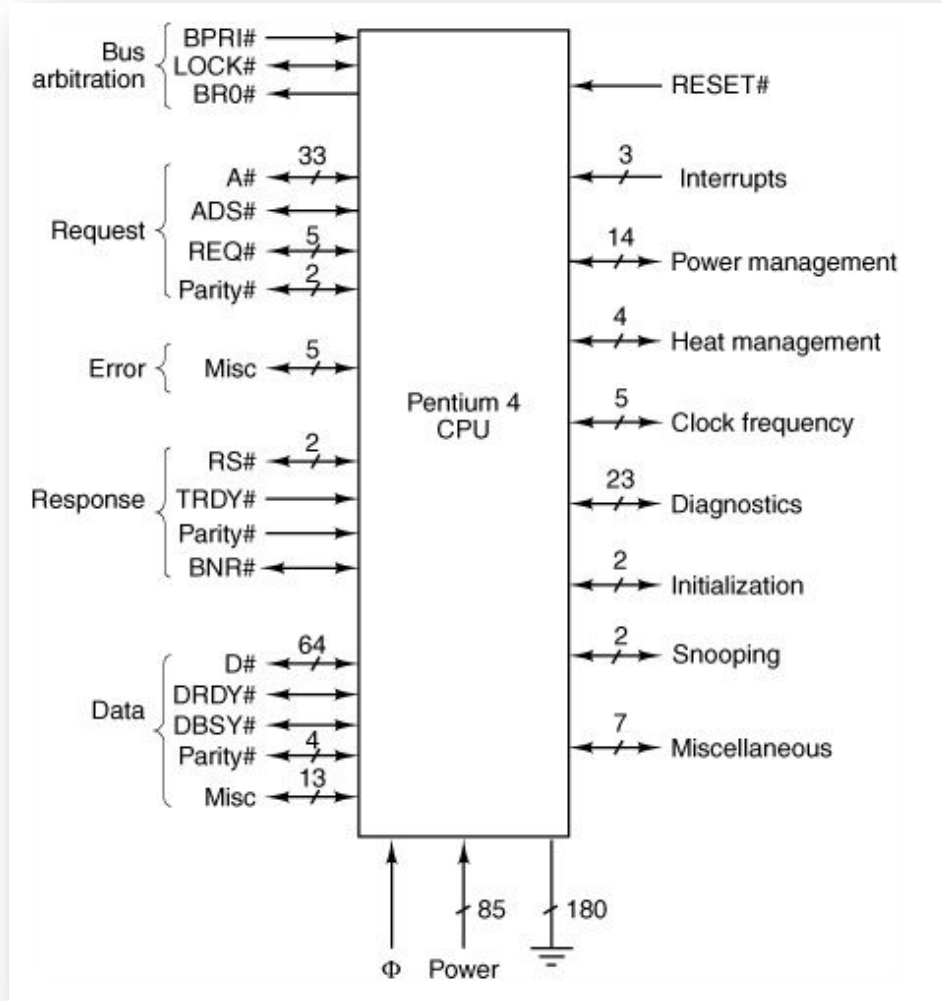
A Arquitetura “von Neumann”



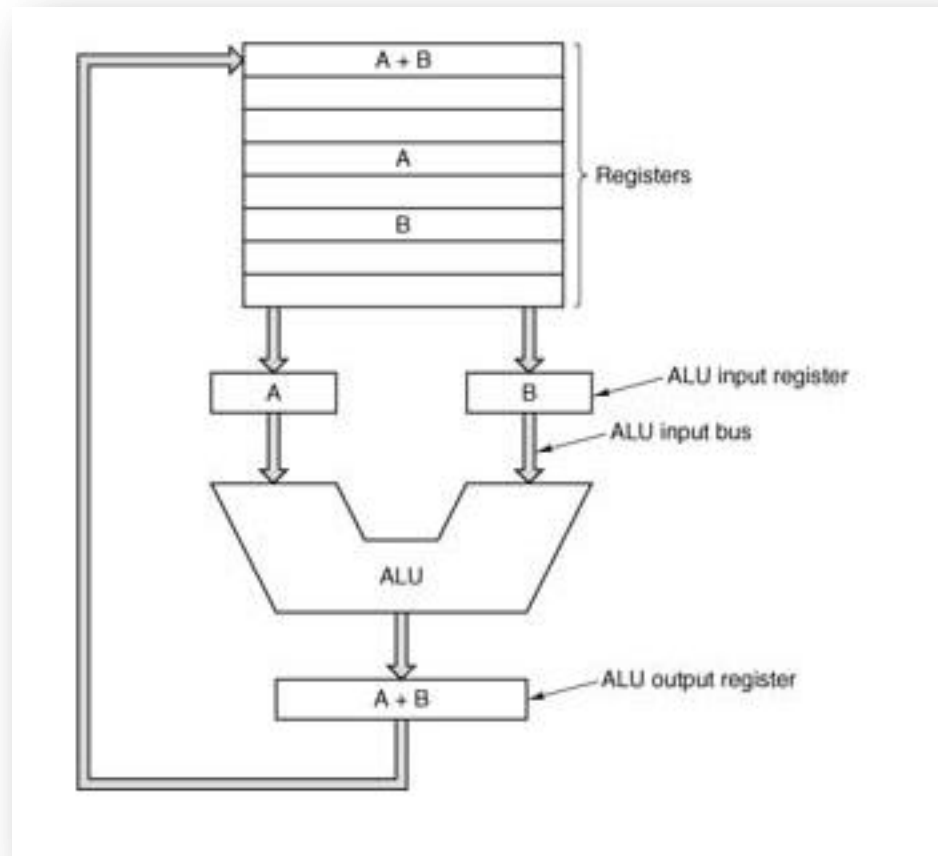
Abstração do Processador



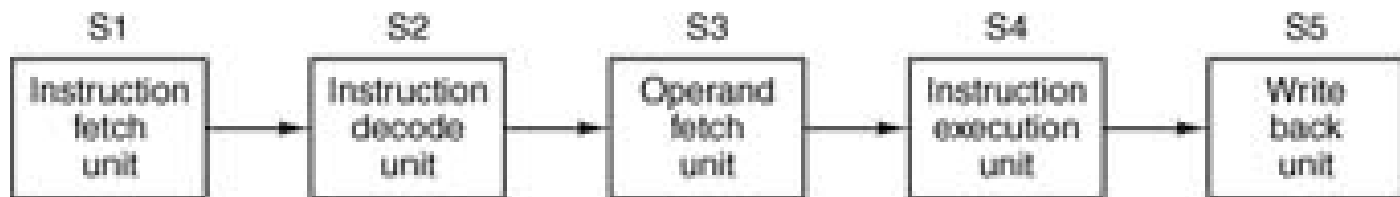
Exemplos



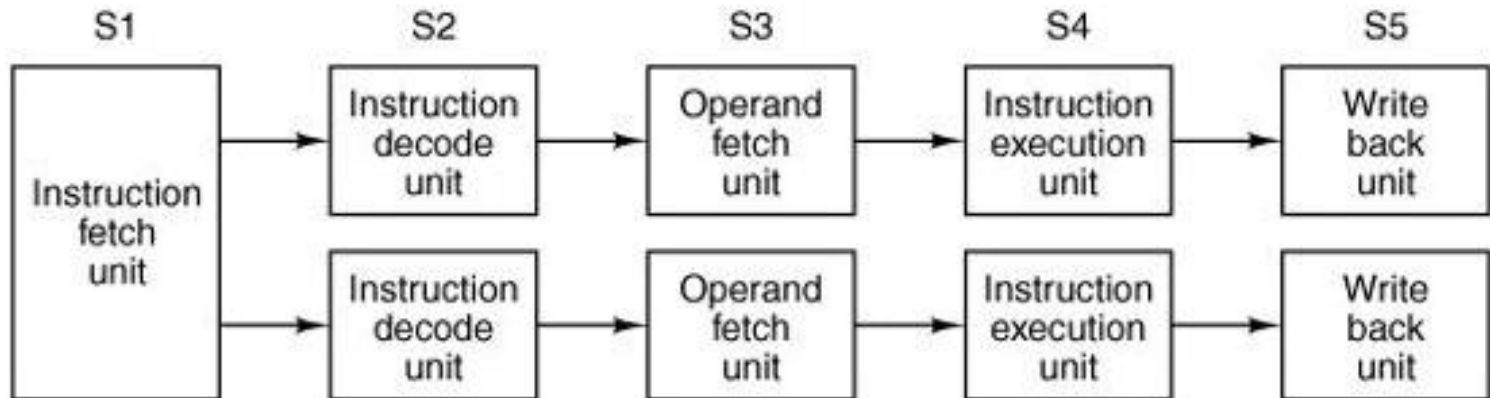
Data path



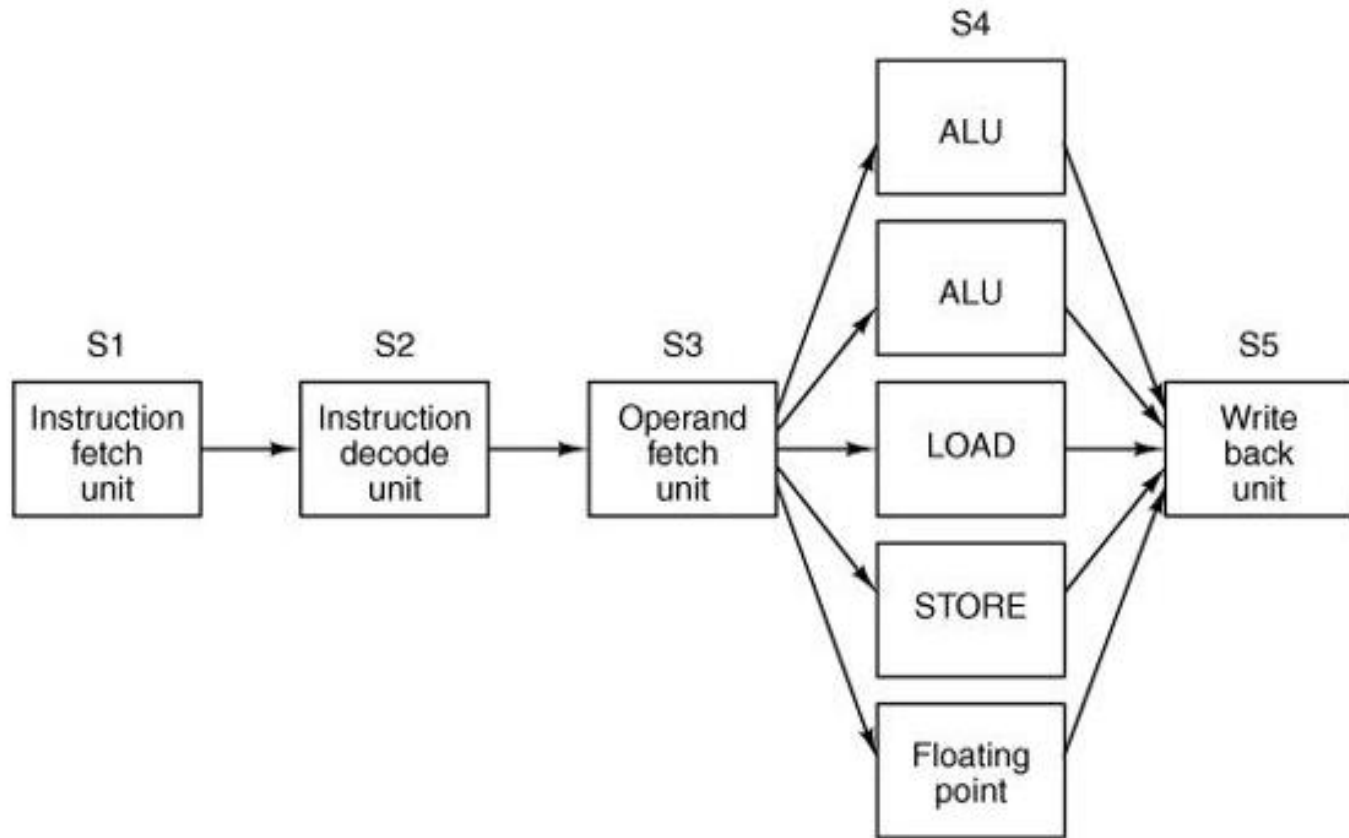
Processadores – Pipelining



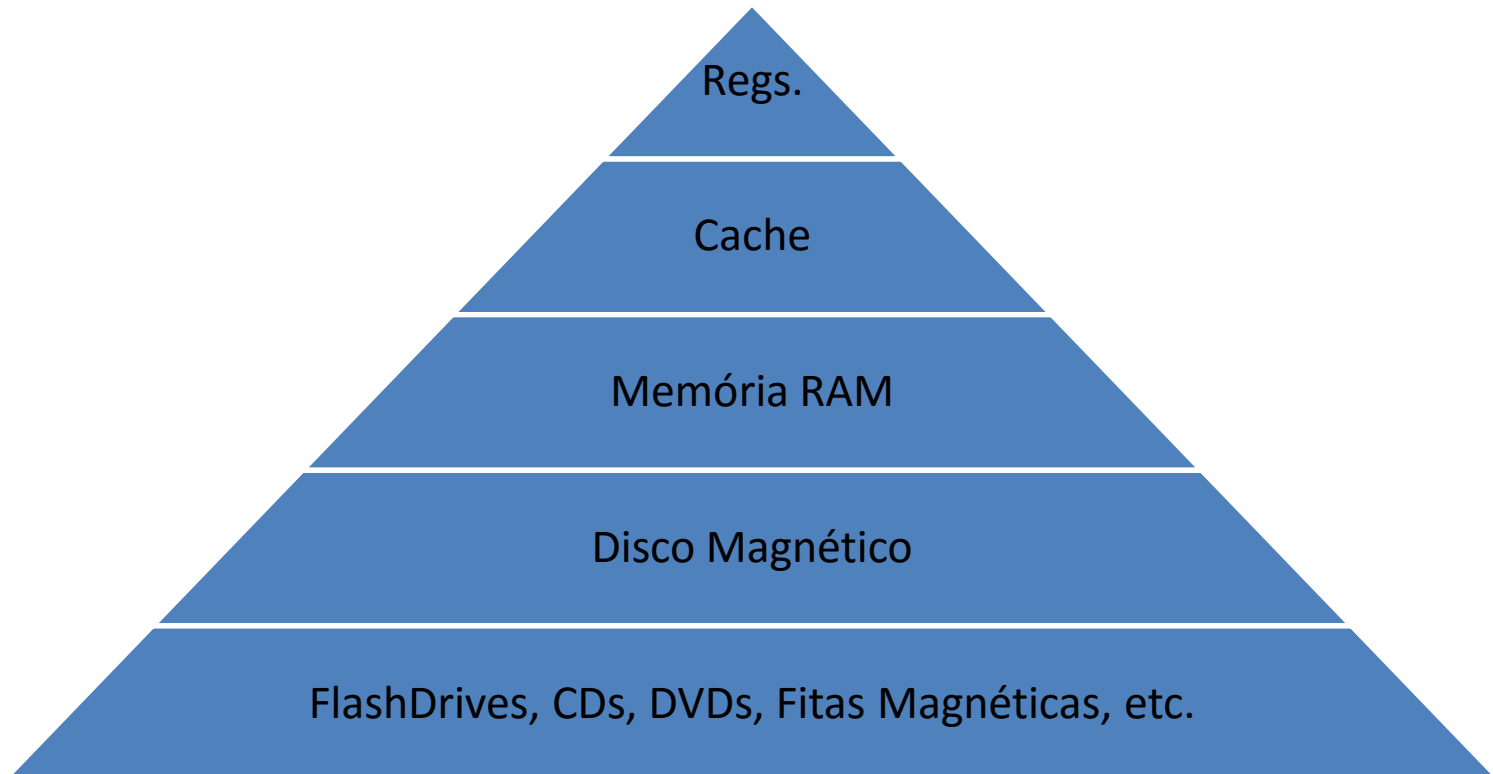
(a)



Processadores Superescalares



Hierarquia de Memórias



Abstração – Linguagem de Montagem

```
Loop: g = g + A[i];  
      i = i + j;  
      if (i != h) goto Loop;
```

```
g: $s1  
h: $s2  
i: $s3  
j: $s4  
Base of A: $s5
```

```
Loop: add $t1, $s3, $s3 # $t1 = 2 * i  
      add $t1, $t1, $t1 # $t1 = 4 * i  
      add $t1, $t1, $5  # $t1=address of A[i]  
      lw  $t0, 0($t1)  # $t0 = A[i]  
      add $s1, $s1, $t0 # g = g + A[i]  
      add $s3, $s3, $s4 # i = i + 1  
      bne $s3, $s2, Loop # go to Loop if i != h
```

```
while (save[i] == k)  
    i = i + j;
```

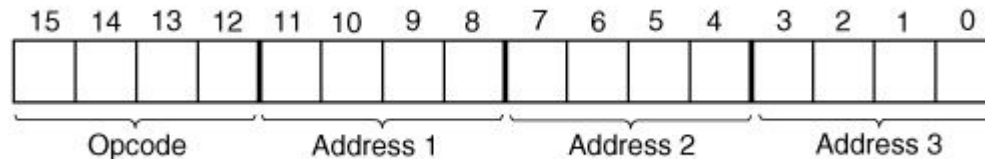
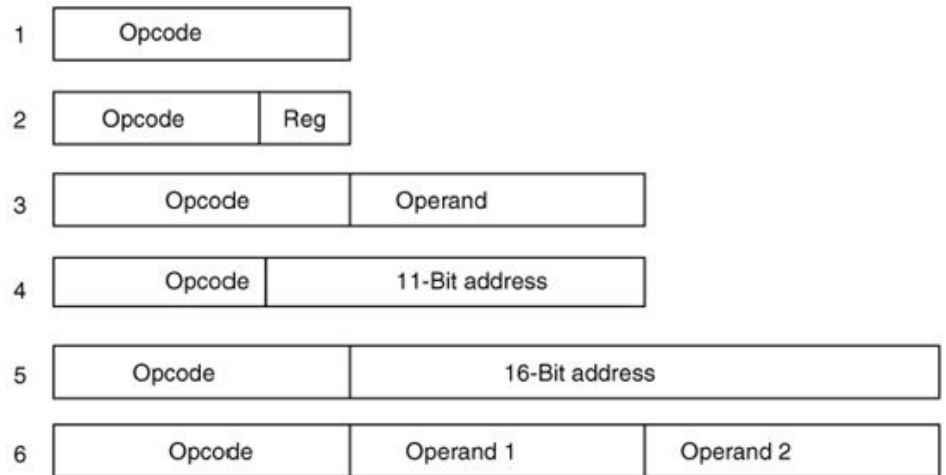
i: \$s3; j: \$s4; k: \$s5; base of save: \$s6

```
Loop: add $t1, $s3, $s3 # $t1 = 2 * i  
      add $t1, $t1, $t1 # $t1 = 4 * i  
      add $t1, $t1, $s6 # $t1 = address of save[i]  
      lw  $t0, 0($t1)  # $t0 = save[i]  
      bne $t0, $s5, Exit # go to Exit if save[i] != k  
      add $s3, $s3, $s4 # i = i + j  
      j    Loop        # go to Loop  
Exit:
```

Abstração ISA

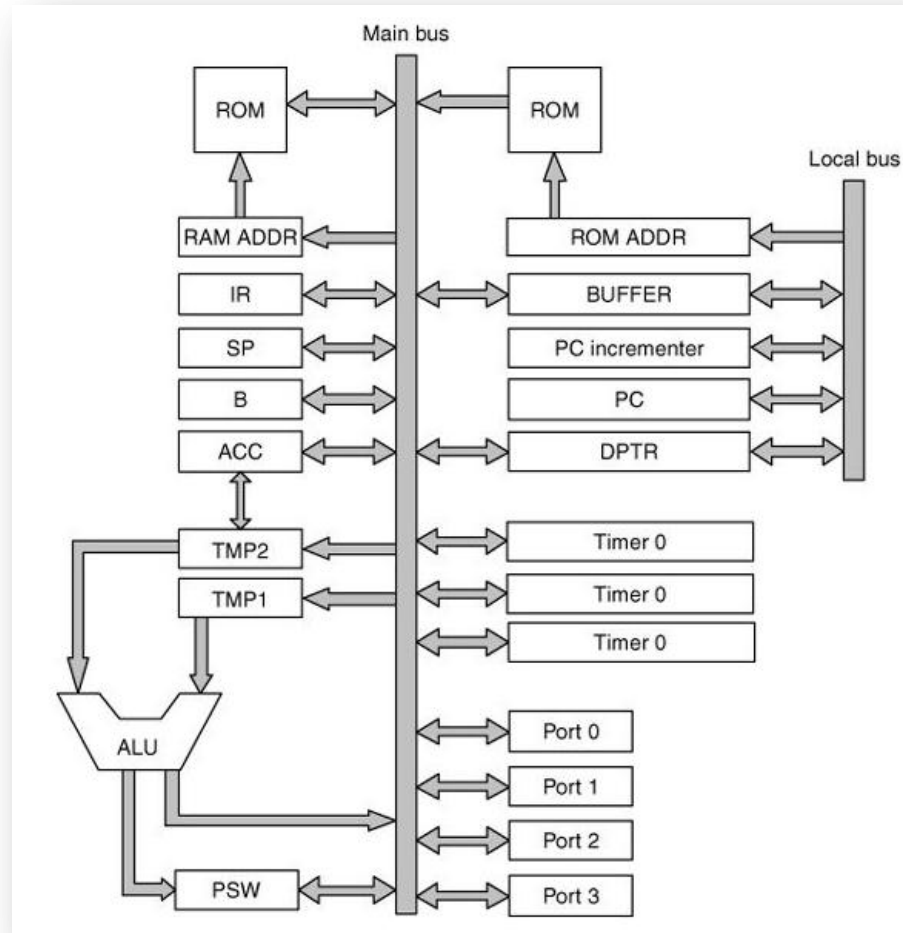
- Instruções;
- Tipos de dados;
- Interrupções;

MOV	R1	4
-----	----	---

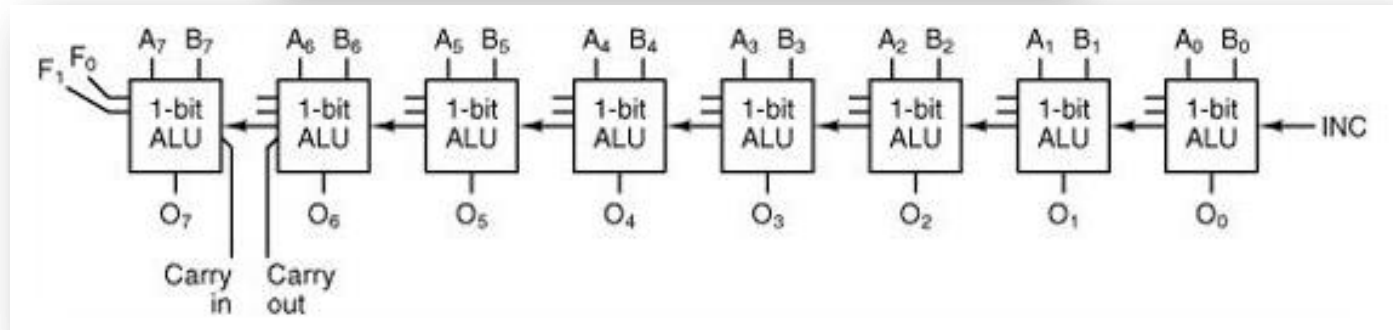
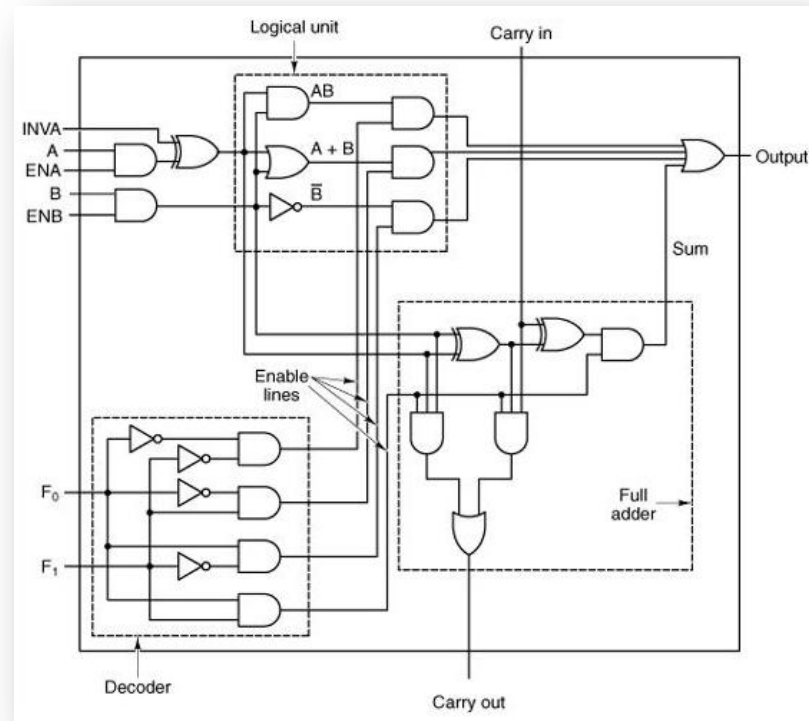


Abstração de Microarquitetura

- Implementa a ISA – Instruction Set Architecture



Abstração do Nível Lógico Digital



Abstração do Nível Elétrico

