GBC046– Arq. e Org. de Computadores II

# Organização Estruturada de Computadores

## Arquitetura e Organização de Computadores

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

## Nesta Aula

- O Computador como uma pilha de abstrações;
- Os cinco elementos básicos de um computador moderno;
- Abstração do nível de Linguagem;
- Abstração do nível Organizacional;
- Abstração do nível de Subsistemas;
- Abstração do nível Lógico;
- Visão Geral de Sistemas Computacionais;
- Interconexão de Componentes ;
  - Barramentos;
- Comunicação com o Mundo Exterior;
- Hierarquia de Memória;
- Entrada e Saída de Dados;
- Origem e Motivação da arquitetura CISC ;
- Razões para a manutenção da CISC;
- Origem e Motivação da arquitetura RISC;

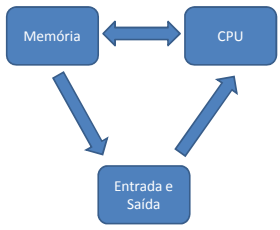
2

## 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

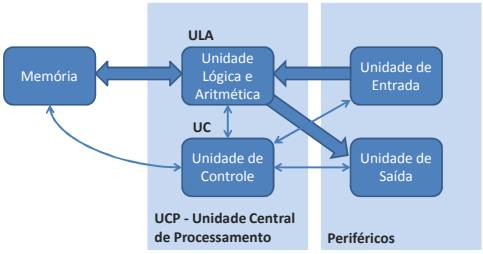
3

## O Modelo von Neumann




4

## O Modelo von Neumann




5

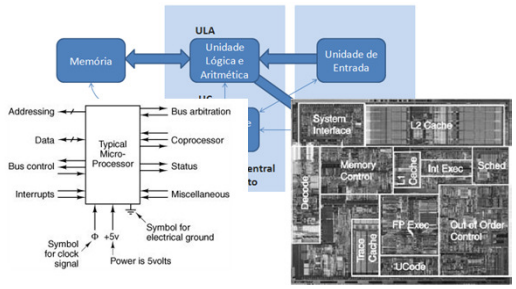
## Uma Pilha de Abstrações

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

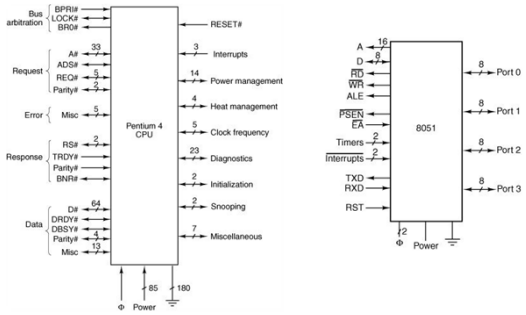


6

## Uma visão geral de um processador A Arquitetura “von Neumann”

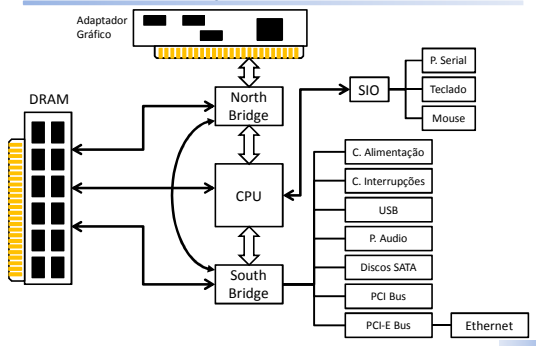



7




8

## Computador vs Sistema Computacional




9

## North Bridge

- Graphics Memory Controller Hub – GMCH;
- Cuida da transferência de dados entre CPU, Memória e Adaptador Gráfico;
- Localizado fisicamente próx. à memória e ao Processador.

10

## South Bridge

- I/O Controller Hub – ICH;
- Cuida de todo o resto do tráfico no sistema computacional:
  - Barramentos
    - PCI
    - PCI-Express
    - SATA
  - Audio (Build-in Audio)
  - Controlador de USB
  - Controlador de Interrupção
  - Controlador de Alimentação

11

## SIO – Super Input and Output

- Legado de antigos PCs;
- Cuida do mouse, teclado e comunicação serial.

12

## Barramentos

- Conjunto de “linhas” de comunicação que interligam os diversos módulos de um sistema computacional;
- Comunicação compartilhada;
- Normalmente barramentos são divididos em três tipos:
  - Dados
  - Endereços
  - Controle
- Alguns sistemas reutilizam linhas de barramento para múltiplas funções;

13

## Barramentos

- Como o barramento conecta diversos dispositivos, deve haver um conjunto de regras que rejam a comunicação (protocolo);
- Um barramento requer um “controlador de barramento” que é um circuito digital que implementa o protocolo de comunicação no barramento;
- Para entendermos como um barramento funciona, primeiro precisamos entender que sinais devem ser considerados.

14

## Barramentos – Sinais de Controle

- Escrita de Memória
- Leitura de Memória
- Escrita de E/S
- Leitura de E/S
- ACK de Transferência
- Solicitação de Barramento
- Concessão de Barramento
- Requisição de Interrupção
- ACK de Interrupção
- Clock
- Reset

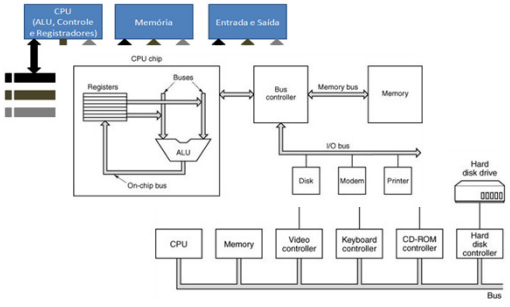
15

## Hierarquia de Barramentos

- Muitos dispositivos → barramento se torna o “gargalo” do sistema computacional;
  - Barramento longo → atraso de propagação
  - Muitos dispositivos → concorrência → atraso

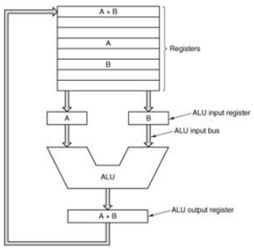
16

## Arquitetura de Barramentos



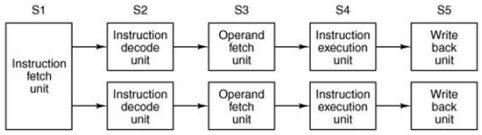

17

## Data path



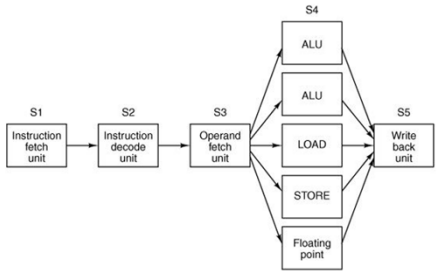

18

## Processadores – Pipelining



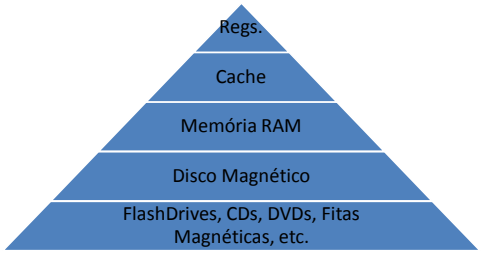

19

## Processadores Superescalares




20

## Hierarquia de Memórias




21

## Abstração Assembly

```
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
```

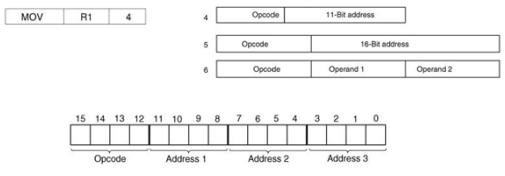
```
while (save[i] == k)
    i = i +j;
```

```
# 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:
```

22

## Abstração ISA

- Instruções;
- Tipos de dados;
- Interrupções;



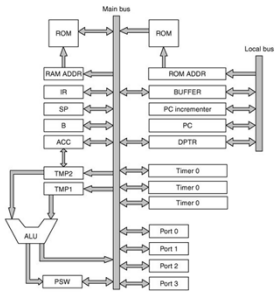




23

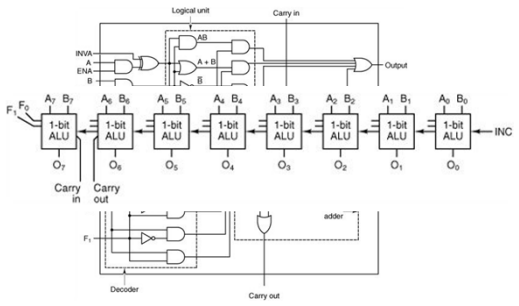
## Abstração de Microarquitetura

- Implementa a ISA – Instruction Set Architecture



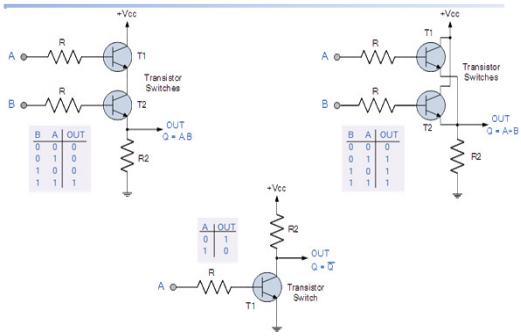

24

## Abstração do Nível Lógico Digital



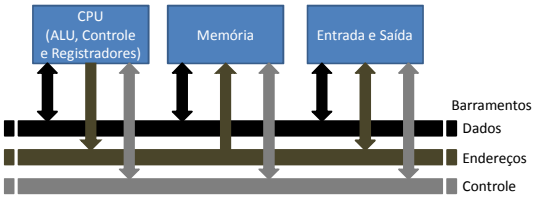

25

## Abstração do Nível Elétrico



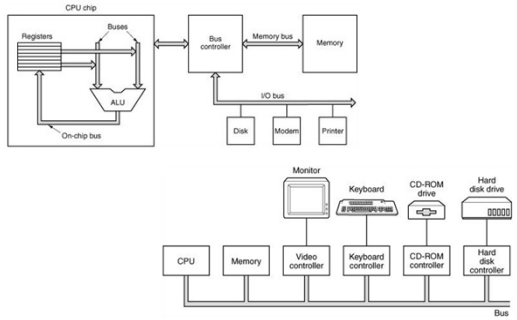

26

## Modelo de Barramento



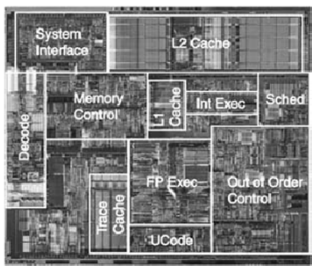

27

## Modelo de Barramento




28

## Abstração do Processador




29

## Barramentos

- Conjunto de “linhas” de comunicação que interligam os diversos módulos de um sistema computacional;
- Comunicação compartilhada;
- Normalmente barramentos são divididos em três tipos:
  - Dados
  - Endereços
  - Controle
- Alguns sistemas reutilizam linhas de barramento para múltiplas funções;

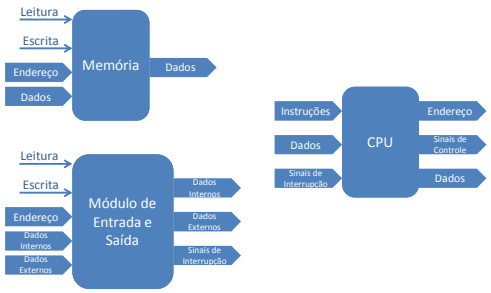
30

## Barramentos

- Como o barramento conecta diversos dispositivos, deve haver um conjunto de regras que rejam a comunicação (protocolo);
- Um barramento requer um "controlador de barramento" que é um circuito digital que implementa o protocolo de comunicação no barramento;
- Para entendermos como um barramento funciona, primeiro precisamos entender que sinais devem ser considerados.

31

## Sinais dos Módulos de um Sistema Computacional




32

## Barramentos – Sinais de Controle

- Escrita de Memória
- Leitura de Memória
- Escrita de E/S
- Leitura de E/S
- ACK de Transferência
- Solicitação de Barramento
- Concessão de Barramento
- Requisição de Interrupção
- ACK de Interrupção
- Clock
- Reset

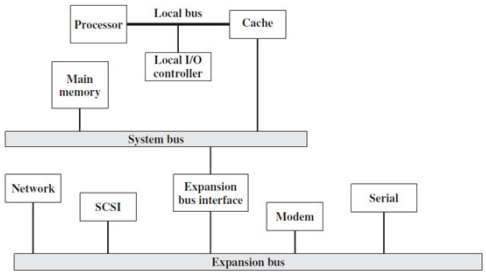
33

## Hierarquia de Barramentos

- Muitos dispositivos → barramento se torna o "gargalo" do sistema computacional
  - Barramento longo → atraso de propagação
  - Muitos dispositivos → concorrência → atraso

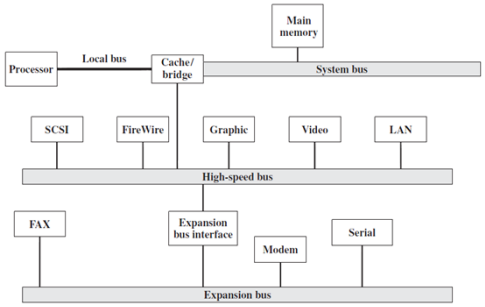
34

## Barramento Comum




35

## Barramento de Alta Velocidade




36