Organização de Computadores

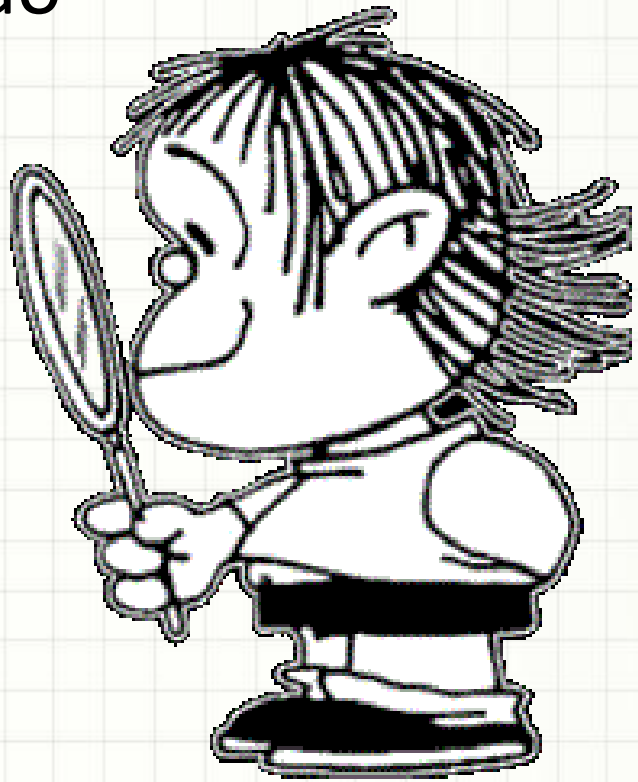
O Processador e Seus Componentes

Prof. Dr. Daniel Caetano

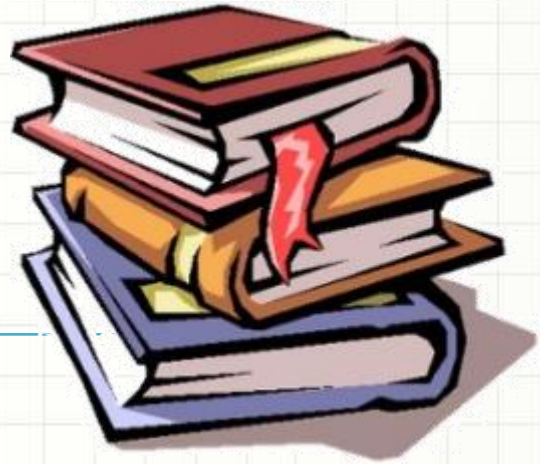
2014 - 1

Lembretes

- Recordar a organização interna da CPU
- Conhecer os registradores de controle
- Apresentar o ciclo de instrução
- Compreender o mecanismo de PipeLine



Material de Estudo



Material	Acesso ao Material
Notas de Aula	http://www.caetano.eng.br/ (Aula 8)
Apresentação	http://www.caetano.eng.br/ (Aula 8)
Material Didático	Fundamentos da Arquitetura de Computadores, páginas 27 a 48
Biblioteca Virtual	Introdução à Organização de Computadores, páginas 153 a 203 Arquitetura e Organização de Computadores, páginas 287 a 426

INTRODUÇÃO

Introdução

- ULA: Faz os Cálculos
- UC:
  - Controla a execução do programa (ordem de leitura das instruções)
  - Traz dados da memória e dispositivos para os registradores
  - Comanda a ULA

- Como isso tudo ocorre?

A Unidade de Controle

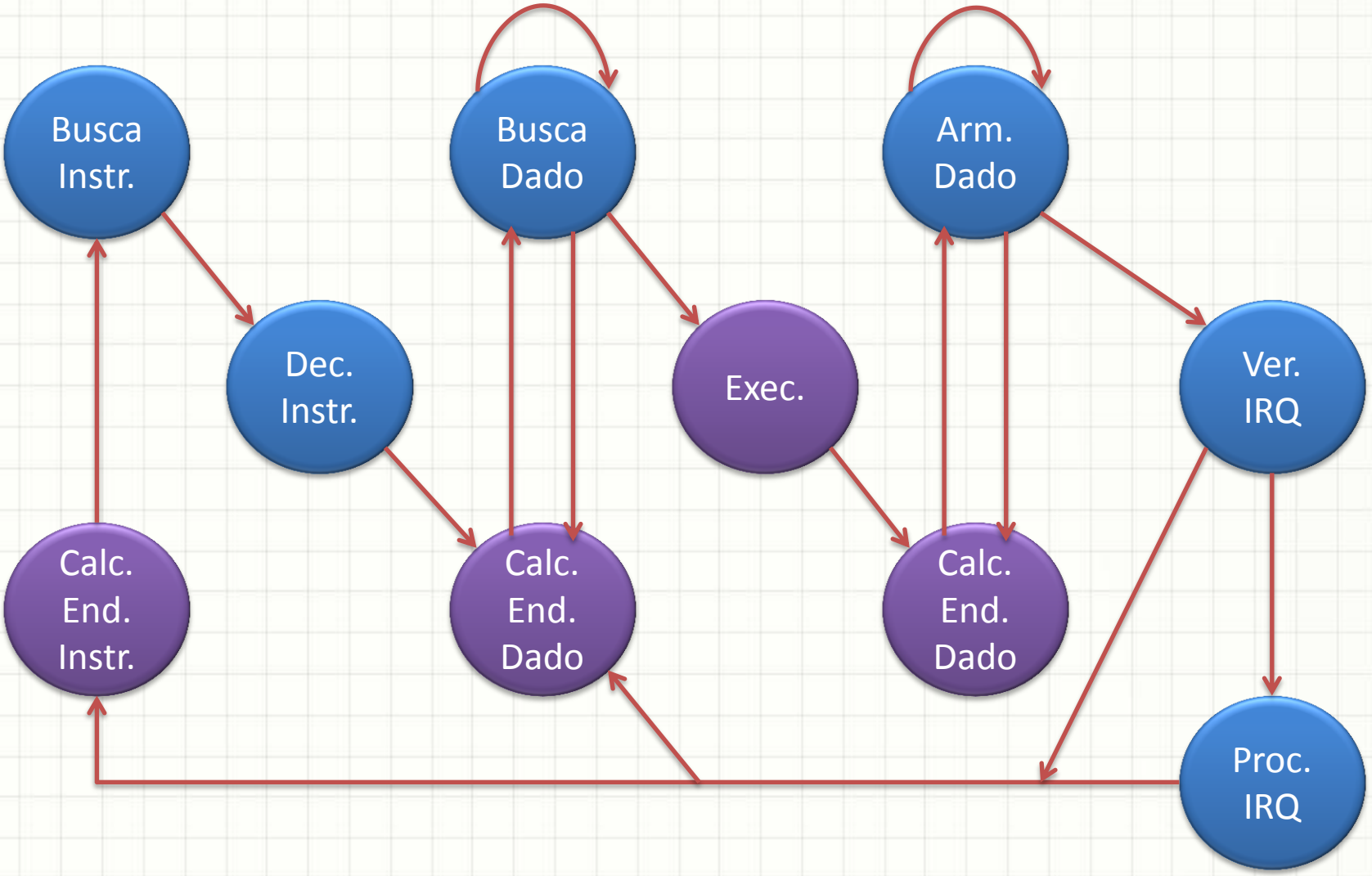
Responsabilidades da UC

- Analogia: Pessoa (UC) usando Calculadora (ULA)
- Responsabilidades
  - Controlar a execução de instruções na ordem certa
  - Leitura da memória principal e E/S
  - Escrita na memória principal e E/S
  - Controlar os ciclos de interrupção

Rotina de Operação da UC

a) Busca de Instrução
b) Interpretação da Instrução
c) Busca dados (se necessário)
d) Processa dados (se necessário) – **ULA**
e) Escrita de dados (se necessário)
f) Avaliação de Interrupções
g) Execução de Interrupção (se necessário)
h) Volta para (a)

Diagrama do Ciclo de Instrução



Registradores Especiais

Registradores Especiais da ULA

- Quais os registradores da ULA?
  - ULA faz “contas”...
- Acumulador: local onde os resultados são armazenados

- Mas... a ULA só faz essas “contas”?
  - Sim... mas a ULA calcula mais do que o resultado!

Registradores Especiais da ULA

- Ela também atualiza um registrador de “flags”

...	...	...	...	Neg	Carry	Zero	Parity

- Dão informações sobre os resultados
  - Se o resultado é zero, o bit “zero” vira 1
  - Se o resultado superou o limite, “carry” vira 1
  - Se o número de bits 1 é par, “parity” vira 1
  - Etc...

Registradores Especiais da UC

- E a UC?
  - Onde está a próxima instrução?
    - **PC** (Program Counter ou Contador de Programa)
  - Qual instrução está sendo processada?
    - **IR** (Intruction Register ou Registrador de Instrução)
  - Qual endereço sendo lido?
    - **MAR** (Memory Address Register ou Registrador de Endereço de Memória)
  - Qual é o dado sendo lido?
    - **MBR** (Memory Buffer Register ou Registrador de Buffer de Memória)
- MAR e MBR são ligados aos barramentos

Outros Registradores

- Registradores de Propósito Geral
  - B, C, D, E... (EBC, ECX, EDX...)
- Registradores de Pilha
  - Armazenamento de Dados (LIFO)
  - **SP** ou **BP** (Stack/Base Pointer: aponta para o topo da pilha)
- Registradores de Índices
  - **IX**, **SI**, **DI** (Index, Source Index, Destination Index)
- Registradores de Segmento (MMU)
  - **CS** (Code Segment ou Segmento de Código)
  - **DS** (Data Segment / Segmento de Dados)
  - **SS** (Stack Segment ou Segmento da Pilha)

A Pipeline

Pipeline

- Conceito de Linha de Produção
  - Quebrar tarefa complexa em tarefas menores
  - Ex.: Fazer carro
    - Fazer roda
    - Fazer motor
    - Fazer lataria
    - ...
- Por que aplicar isso para CPU?
  - Quando a memória é acessada, a ULA fica ociosa
  - Duas etapas: **busca (UC)** e **execução (ULA)**

Partes da CPU em Funcionamento

- Simplificadamente... sem pipeline

Tempo	Busca(UC)	Execução(ULA)
0	I1	-
1	-	I1
2	I2	-
3	-	I2
4	I3	-
5	-	I3

- E assim por diante...

Partes da CPU em Funcionamento

- Sim **2 ciclos por instrução!**

Tempo	Busca(UC)	Execução(ULA)
0	I1	-
1	-	I1
2	I2	-
3	-	I2
4	I3	-
5	-	I3

- E assim por diante...

Partes da CPU em Funcionamento

- Simplificadamente... **COM** pipeline

Tempo	Busca(UC)	Execução(ULA)
0	I1	-
1	I2	I1
2	I3	I2
3	I4	I3
4	I5	I4
5	I6	I5

- E assim por diante...

Partes da CPU em Funcionamento

- Sim **1 ciclo por instrução!**

Tempo	Busca(UC)	Execução(ULA)
0	I1	-
1	I2	I1
2	I3	I2
3	I4	I3
4	I5	I4
5	I6	I5

- E assim por diante...

Partes da CPU em Funcionamento

- Comparando lado a lado...

Sequência no Tempo	SEM pipeline		COM pipeline	
	Busca	Execução	Busca	Execução
0	I1	-	I1	-
1	-	I1	I2	I1
2	I2	-	I3	I2
3	-	I2	I4	I3
4	I3	-	I5	I4

- Maior eficiência: + instruções, - tempo

Pipeline de Múltiplos Níveis

- Quebrar processamento em 6 etapas
  - **BI**: Busca de Instruções
  - **DI**: Decodificação de Instruções
  - **CO**: Cálculo de Operandos
  - **BO**: Busca de Operandos
  - **EI**: Execução da Instrução
  - **EO**: Escrita de Operando
- Cada etapa dura 0,33 comparado com as anteriores
- Como é o processamento com tudo isso?

Pipeline de Múltiplos Níveis

- Sem pipeline

Tempo	BI	DI	CO	BO	EI	EO
0,00	I1	-	-	-	-	-
0,33	-	I1	-	-	-	-
0,66	-	-	I1	-	-	-
1,00	-	-	-	I1	-	-
1,33	-	-	-	-	I1	-
1,66	-	-	-	-	-	I1
2,00	I2	-	-	-	-	-

Pipeline de Múltiplos Níveis

- Sen

2 ciclos por instrução!

Tempo	BI	DI	CO	BO	EI	EO
0,00	I1	-	-	-	-	-
0,33	-	I1	-	-	-	-
0,66	-	-	I1	-	-	-
1,00	-	-	-	I1	-	-
1,33	-	-	-	-	I1	-
1,66	-	-	-	-	-	I1
2,00	I2	-	-	-	-	-

Pipeline de Múltiplos Níveis

- **Com** pipeline

Tempo	BI	DI	CO	BO	EI	EO
0,00	I1	-	-	-	-	-
0,33	I2	I1	-	-	-	-
0,66	I3	I2	I1	-	-	-
1,00	I4	I3	I2	I1	-	-
1,33	I5	I4	I3	I2	I1	-
1,66	I6	I5	I4	I3	I2	I1
2,00	I7	I6	I5	I4	I3	I2

Pipeline de Múltiplos Níveis

- **Co**

0,33 ciclos por instrução!

3 instruções por ciclo!

Temp						
0,00						
0,33	I2	I1	-	-	-	-
0,66	I3	I2	I1	-	-	-
1,00	I4	I3	I2	I1	-	-
1,33	I5	I4	I3	I2	I1	-
1,66	I6	I5	I4	I3	I2	I1
2,00	I7	I6	I5	I4	I3	I2

PERGUNTAS?

Conclusões

Resumo

- UC: função burocrática, mas importante!
- UC coordena:
  - sequência do programa
  - transferência de dados de e para a memória/disp.
- Pipeline: acelerar a execução das instruções

- Como é o acesso à memória?
- O que é cache?