

UFRJ – IM - DCC



# Sistemas Operacionais I

---

## Unidade I Fundamentos de Hardware e de Software



# ORGANIZAÇÃO DA UNIDADE

- Introdução
- Fundamentos de Hardware e Software
  - Organização e Componentes de um SC
  - Organização Física e Funcional do Processador
  - Estrutura de Armazenamento
  - Estrutura de E/S
  - Linguagens, Programas e Instruções
- Estruturas de Sistemas Operacionais



# Analógico x Digital

- Circuito Analógico
  - Trabalha com o sinal na forma contínua
  - Construído com base em componentes eletrônicos discretos: resistores, capacitores, indutores, válvulas e transistores
- Circuito Digital
  - Trabalha com o sinal na forma digital
  - Construído com base em portas lógicas, que implementam a lógica booleana



# Portas Lógicas Básicas

- Portas E, Ou, Não E, Não Ou, Ou Exclusivo

A	B	E	Ou	Não E	Não Ou	XOR
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0



# Tipos de Circuitos Digitais

- Combinacionais
  - Implementam a lógica digital sem memória – a saída no instante “t” depende apenas das entradas em “t”
- Sequenciais
  - Implementam a lógica digital com memória – a saída no instante “t” depende da entrada e do estado em “t-1”
  - Contadores, buffers, registradores, memória, ...



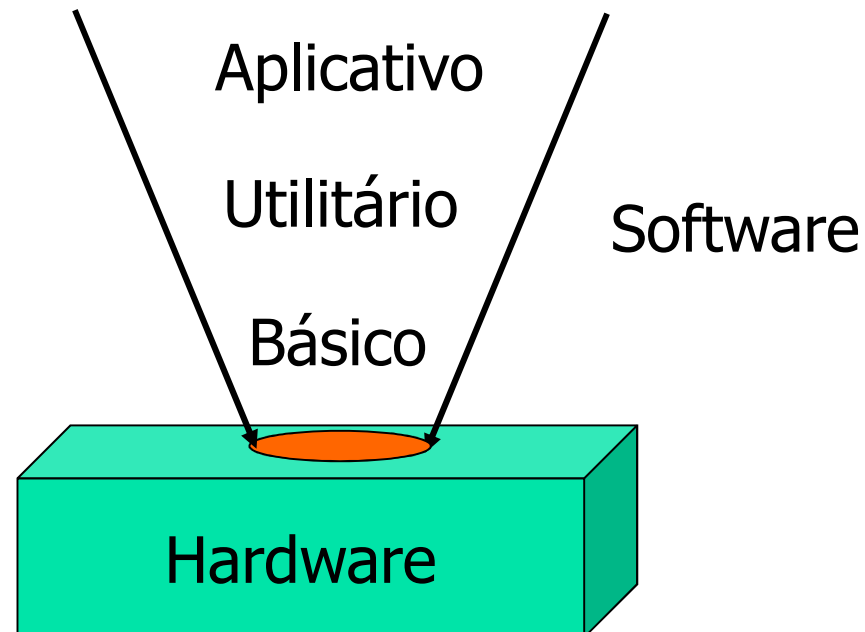
# Hardware x Software

- Com os circuitos sequenciais surgiu o conceito de memória
- Com a memória surgiu a possibilidade de armazenar seqüências de valores binários
- Com os valores binários armazenados aplicados sobre circuitos combinacionais e sequenciais, surge o conceito de programa internamente armazenado
- A adição de lógica (desvios no fluxo de execução) no programa armazenado, dá uma sintaxe e uma semântica às unidades de operação (instruções), e surge o conceito de software.



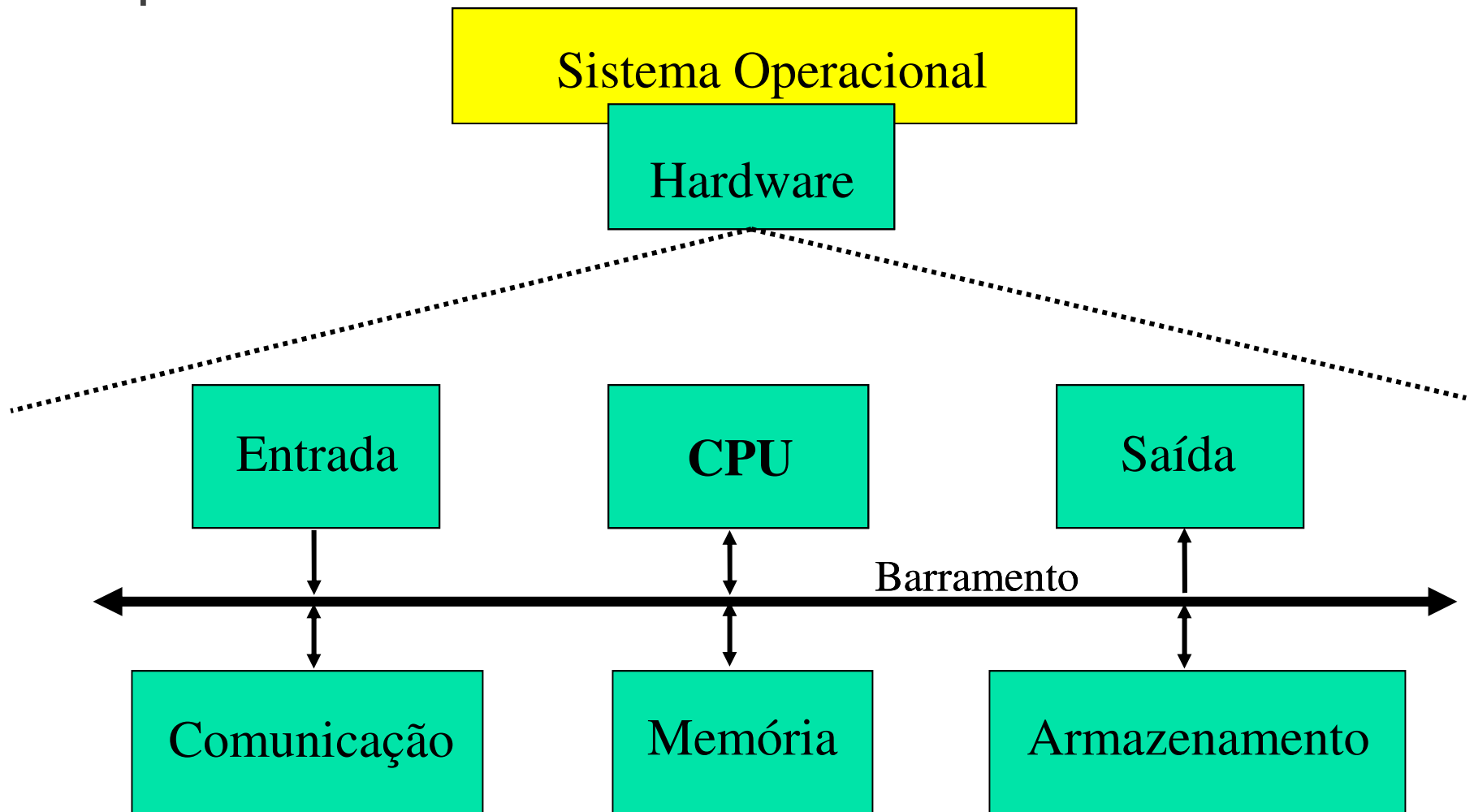
# Computador

- Surge da união de Hardware e Software para uso genérico





# Organização e Componentes







# Barramento

- É o meio de comunicação entre os diferentes componentes de um Sistema de Computação
- É o caminho por onde transitam:
  - dados,
  - endereços e
  - sinais de controle.



# Barramentos

## Barramento Síncrono

- Tem por base um relógio temporizador que define a frequência de operação do mesmo (ciclo de barramento).
- Toda operação de transferência é sincronizada com a frequência do barramento.
- Toda operação consome um número inteiro e conhecido de ciclos.

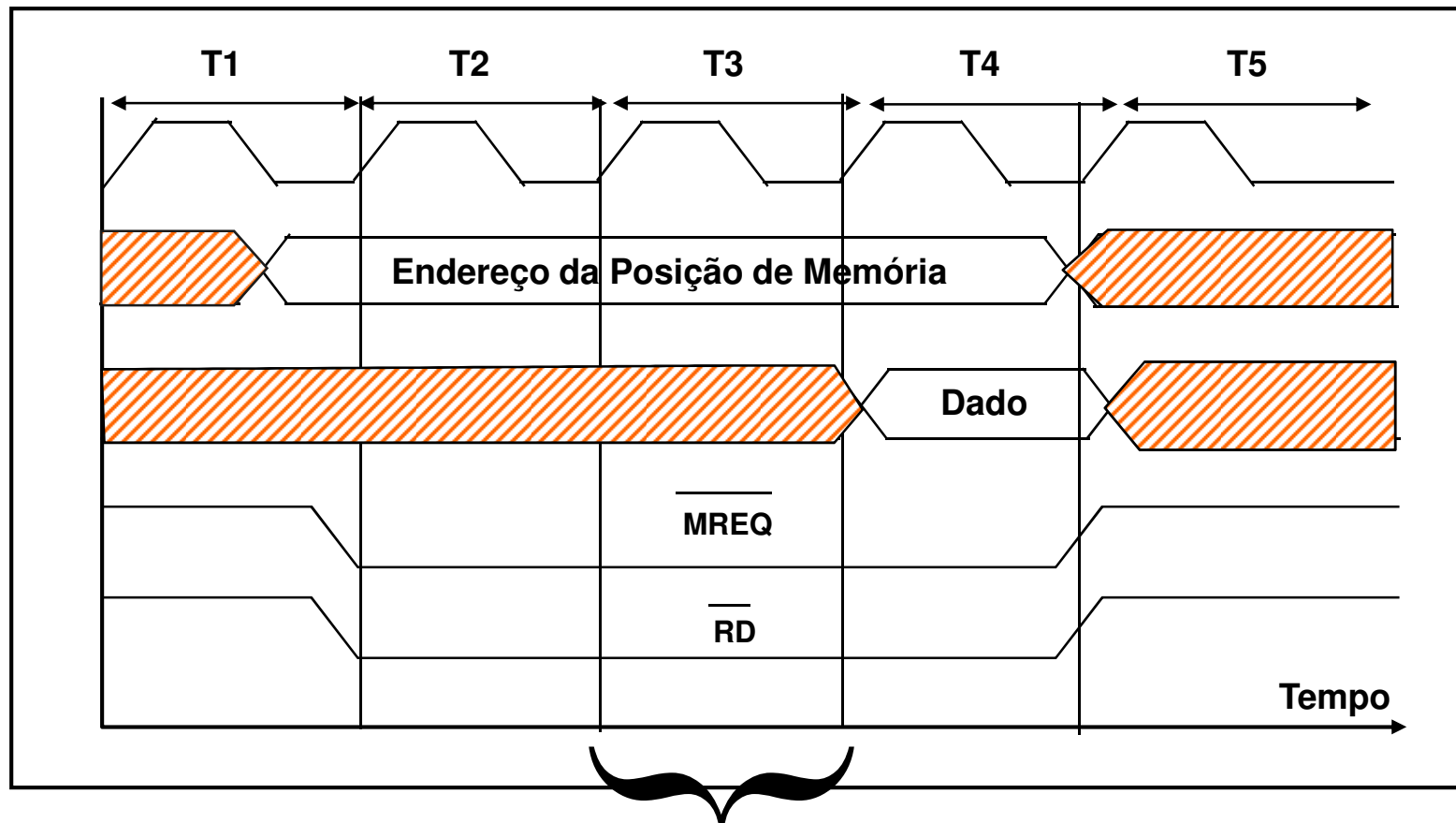
## Barramento Assíncrono

- Não existe temporizador
- Os ciclos duram o tempo que for requerido pela operação
- Uma mesma operação pode ter ciclos variáveis
- Faz uso de um processo de sinalização (handshake)



# Barramento

## Barramento Síncrono

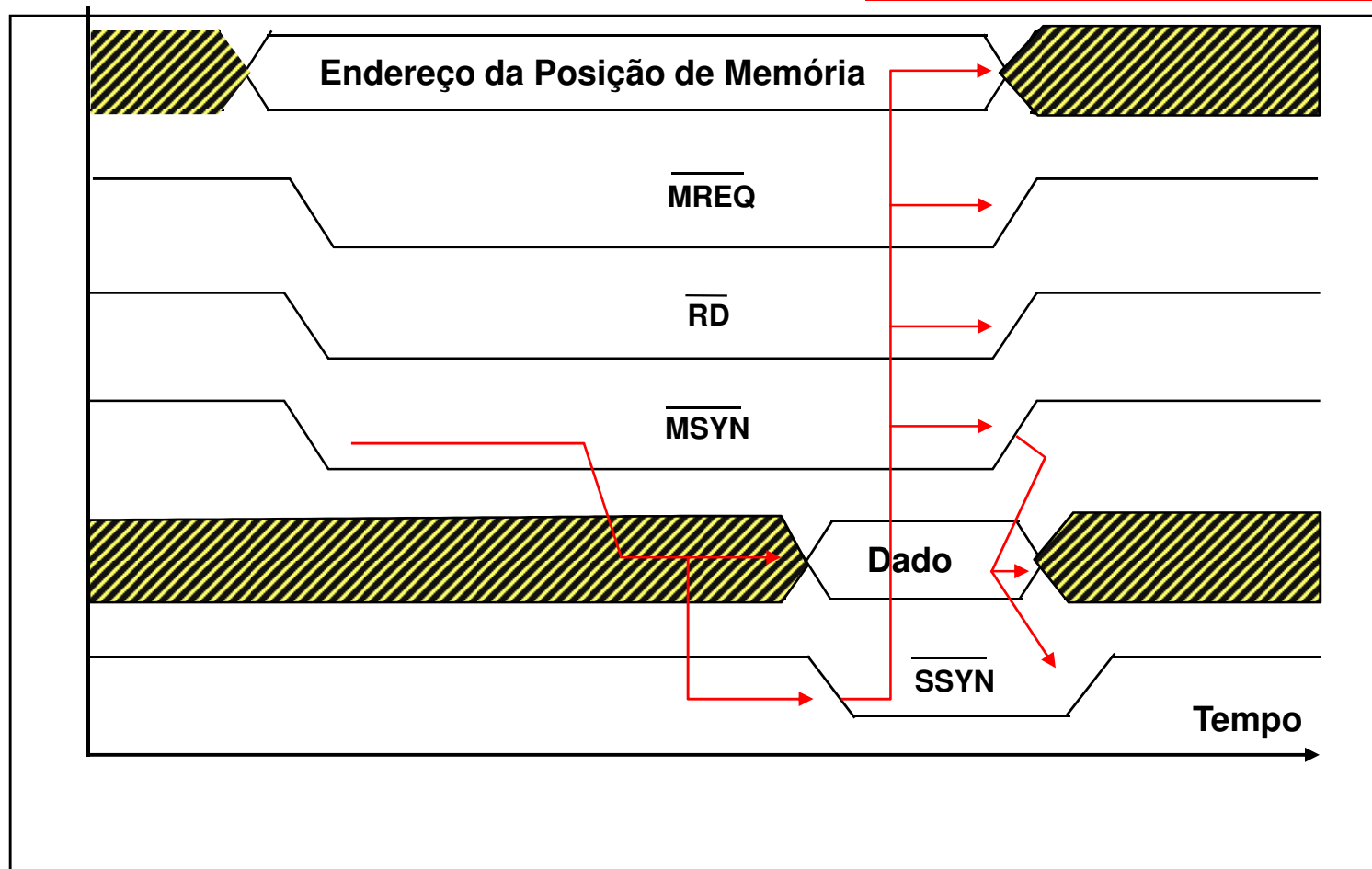


Ciclo de barramento



# Barramentos

## Barramento Assíncrono





# Barramentos

## Síncrono

- Mais fácil de ser construído
- Maior velocidade de operação

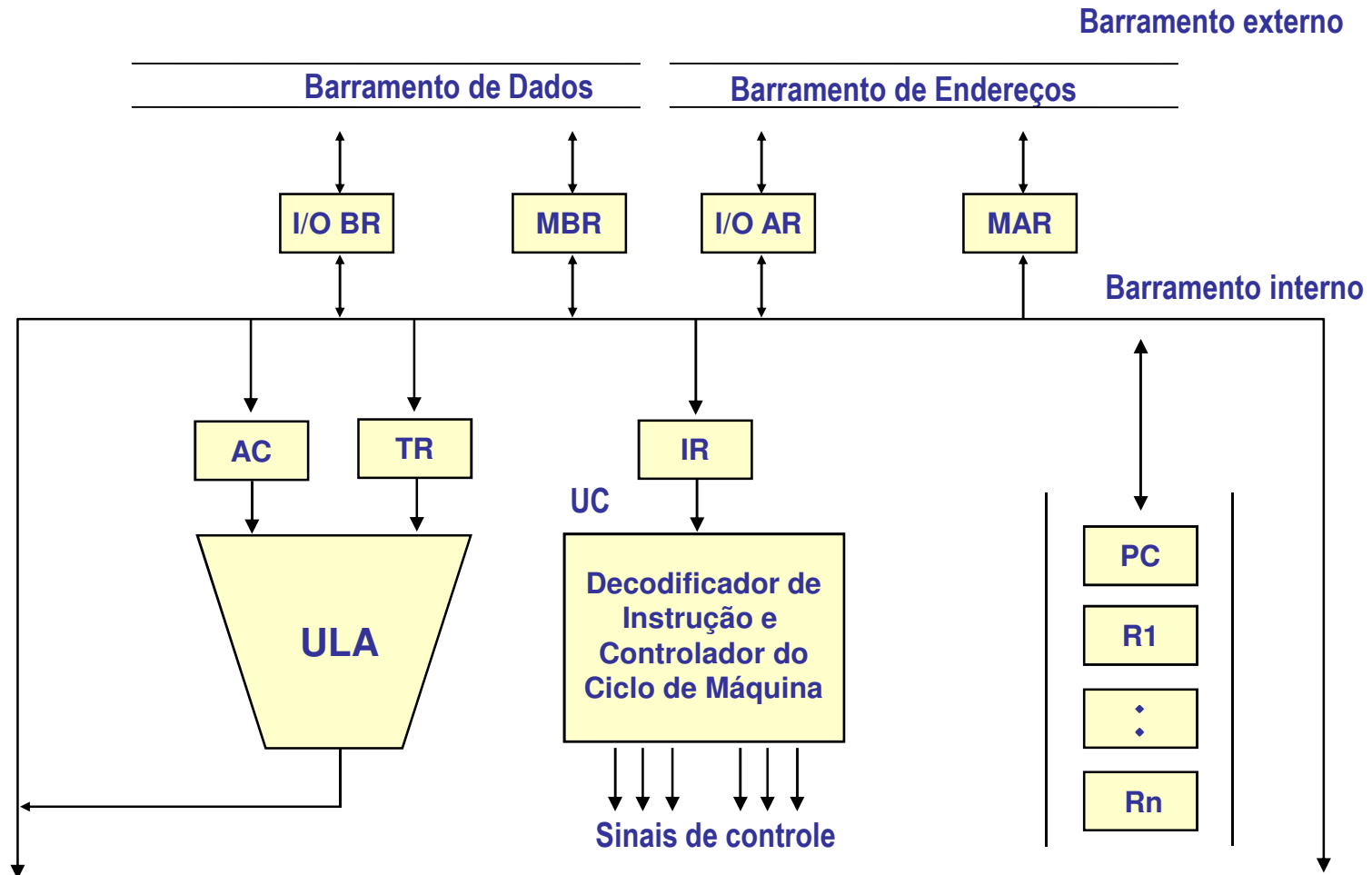
## Assíncrono

- Maior flexibilidade de uso
- Oferecem vantagens para atendimento a um conjunto heterogêneo de dispositivos (lentos e rápidos)

**A maioria dos barramentos é Síncrono**

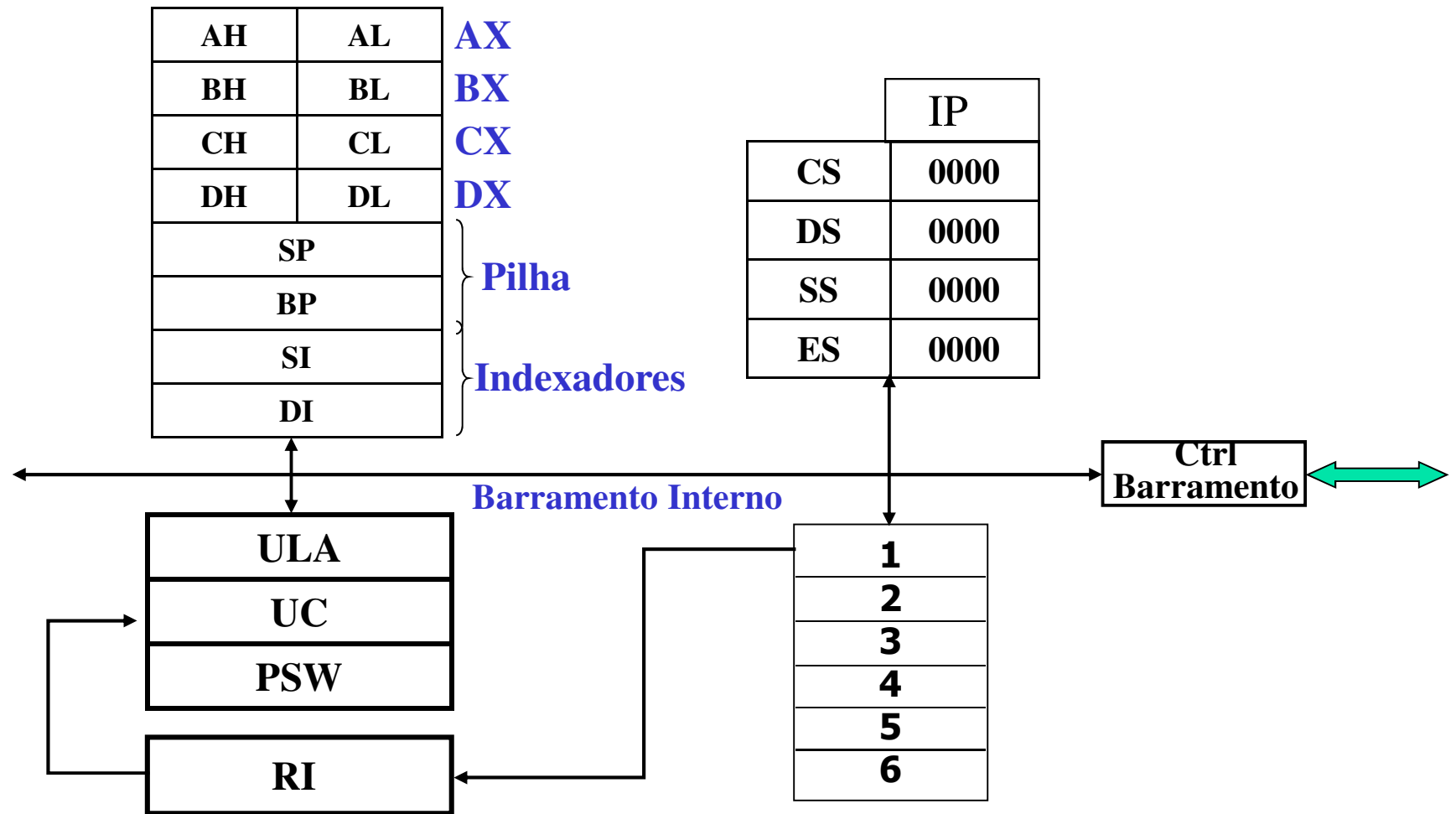


# Organização do processador



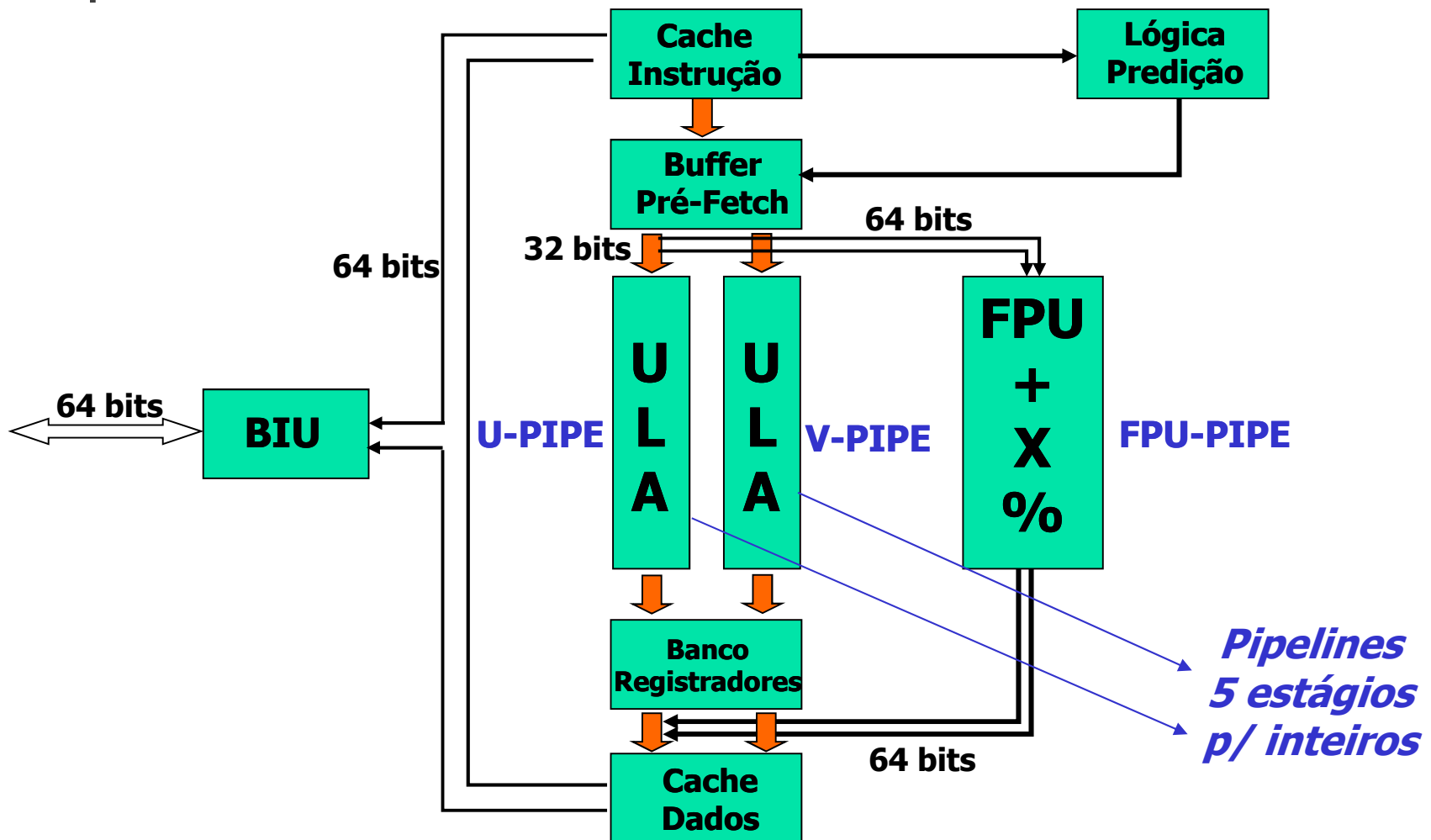


# CPU 8086





# CPU Pentium Superscalar







# Registradores da CPU

Componentes internos de trabalho da CPU, constituem uma área de armazenamento de acesso mais rápido porém de menor capacidade que a memória principal.

**Tipos:** {  
    ☞ Registradores Visíveis : {  
        ✓ Registradores de dados  
        ✓ Registradores de endereço  
        ✓ Códigos de condição  
    ☞ Registradores de Controle e Status



# Registradores da CPU

## Registradores Visíveis:

- Armazenamento temporário de dados durante o processamento
- Acessíveis via linguagem de máquina.

## Registradores de Controle e Status:

- Controlam o funcionamento do processador e a execução dos programas
- Quando acessíveis por software, o são somente por rotinas privilegiadas.



# Registradores da CPU

## Registradores Visíveis : Dados

- Podem ser usados pelo usuário para manipular dados e executar funções
- O programador/compilador determina a sua função

### Exemplo:

- Acumulador (AC)
  - Registrador que armazena uma das entradas da ULA
- Registrador Temporário (TR)
- Registradores de uso genérico



# Registradores da CPU

## Registradores Visíveis : Endereço

- Contêm endereços de memória dos dados e das instruções
- Podem conter uma parte de um endereço que será usado para calcular o endereço completo.

- Exemplo:**
- Stack Pointer (SP)
    - Endereço do topo da pilha
  - Segment Pointer (SX)
    - Endereço inicial de um segmento
      - Quando a memória é dividida em segmentos, ela é sempre referenciada pelo segmento e pelo deslocamento dentro do segmento (offset)
  - Index Register / Offset
    - Índice a ser adicionado a uma base para se obter o endereço do dado.



# Registradores da CPU

## Registradores Visíveis : Condições

- Contem bits e flags de condição do resultado de uma operação
- Os bits são setados pelo hardware em função do resultado de cada operação realizada
- Podem ser acessados por um programa, apenas no modo leitura.

### Exemplo:

- **Flag de sinal** → Resultado positivo / negativo
- **Flag de zero** → Resultado zero
- **Flag de overflow** → Resultado com estouro



# Registradores da CPU

## Registradores Invisíveis : Controle e Status

- Program Counter (PC)
  - Contém o endereço da próxima instrução a ser executada
- Instruction Register (IR)
  - Contém a última instrução carregada da memória (corrente)
- Program Status Word (PSW)
  - É um registrador (ou um grupo de registradores) que contém:
    - códigos de condição e os bits de informação do status
    - bit de interrupção habilitado/desabilitado
    - bit de modo de operação supervisor/usuário



# Grupos de Registradores 8086

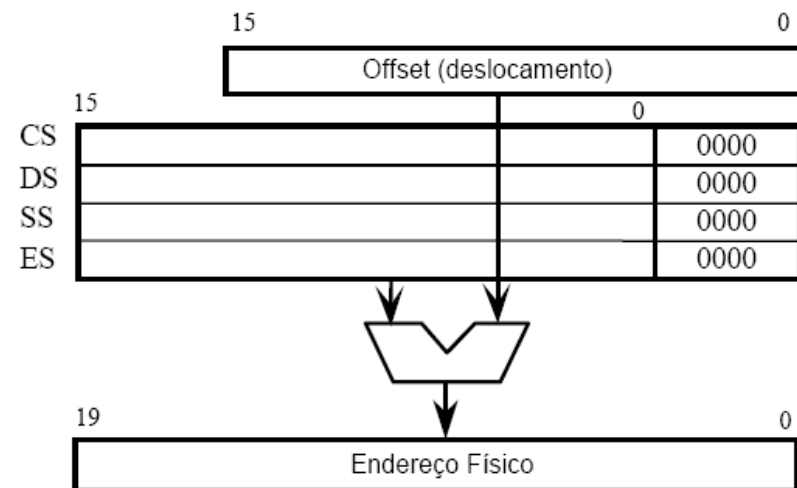
Grupo	Descrição
Registradores de uso geral (RG)	Podem ser utilizados como um registrador de 16 bits ou em dois de 8 bits. São utilizados em operações lógicas e aritméticas
Registradores de pilha (RP)	São utilizados para acessar dados no segmento de pilha, mas também podem ser utilizados em operações lógicas e aritméticas de 16 bits.
Registradores de Indexação (RI)	São utilizados para acesso aos dados, principalmente em operações com cadeias de caracteres ( <i>string</i> ).
Registradores de segmento (RS)	São utilizados como seletores de faixas de endereços. O endereço de um segmento é dado pelo conteúdo do registrador de segmento deslocado de 4 bits à esquerda.
Registrador de estado (PSW – Processor Status Word)	Retrata o estado do programa em execução.. No 8086, somente os 16 primeiros bits existem e os bits 12, 13, 14 e 15 são reservados.



# Registradores 8086

	15	8	7	0	
AX	AH		AL		acumulador
BX	BH		BL		base
CX	CH		CL		contador
DX	DH		DL		dado
SP					ponteiro para pilha
BP					ponteiro base
SI					índice fonte
DI					índice destino
IP					apontador de instruções
FLAGS					flags
CS					segmento de código
DS					segmento de dados
SS					segmento de pilha
ES					segmento extra

## Formação de endereço







# PSW do 8086

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
				O	D	I	T	S	Z		A		P		C

***C – Vai Um***

***P – Paridade***

***A – Vai Um Aux***

***Z – Zero***

***S – Sinal***

***T – Trap***

***I – Interrupção***

***D – Direção***

***O – Overflow***



# Registradores de segmento 8086

## Registradores de Segmento

**CS**

Designa o endereço base do segmento de código do programa. O registrador de deslocamento associado é o registrador IP.

**DS**

Utilizado como referência para acesso ao segmento de dados, exceto para operações com a pilha e operações utilizando cadeia de caracteres

**SS**

Referência para acesso à pilha, utilizando SP e BP como registradores de deslocamento.

**ES**

Em conjunto com o registrador DI é utilizado para operações com cadeias de caracteres



# Registradores do 80386

## Registradores de uso geral

EAX		<b>AH</b>	<b>AL</b>	Acumulador
EBX		<b>BH</b>	<b>BL</b>	Base
ECX		<b>CX</b>	<b>CL</b>	Contador
EDX		<b>DX</b>	<b>DL</b>	Dados

## Registradores de ponteiros e Pilha

Source Index	ESI		<b>SI</b>
Destination Index	EDI		<b>DI</b>
Instruction Pointer	IP		<b>IP</b>
Stack pointer	ESP		<b>SP</b>
Base pointer	EBP		<b>BP</b>



# Registradores adicionais do 80386

## Registradores de Controle

Os registradores de 32 bits CR0, CR1, CR2 e CR3 contêm informações importantes para utilização dos novos recursos. No registrador CR0 cada bit possui uma determinada função, como por exemplo o bit 0, PE (*protection enable*), responsável pela seleção entre modo real e modo protegido. O registrador CR1 é reservado enquanto CR2 armazena o endereço linear que provocou uma falha de página e CR3 armazena informações de controle de paginação como os bits 12 a 31 que contém o endereço base do diretório de páginas.

## Registradores de depuração e testes

Os registradores de depuração, também de 32 bits, DR0...DR7, fornecem recursos para rastreamento dos programas através do armazenamento de pontos onde o programa deverá ser executado passo a passo, enquanto os de teste, TR6 e TR7, são utilizados basicamente na inicialização do sistema operacional.

## Registradores de segmento de 32 bits

Os registradores de segmento foram mantidos com 16 bits, porém foram adicionados os registradores FS e GS, que em conjunto com DS e ES são utilizados para apontar os quatro segmentos de dados permitidos. O endereçamento através do conteúdo do registrador de segmento adicionado ao deslocamento dentro do segmento fica então restrito ao modo real. Em modo protegido é utilizado o endereçamento via descritores



# Registradores apontadores do 80386

Registradores apontadores de tabela para endereçamento em modo protegido

GDTR		End da GDT	Tam da GDT
IDTR		End da IDT	Tam da IDT
LDTR	descriptor da LDT na GDT	End base da LDT da tarefa em execução	Tam da LDT
TR	descriptor da TSS na GDT	End base da TSS da tarefa em execução	Tam da TSS
	63	47	15 0

**GDTR** → registrador que aponta para a base da tabela de descritores globais (GDT)

**LDTR** → registrador que aponta para a base da tabela de descritores locais (LDT)

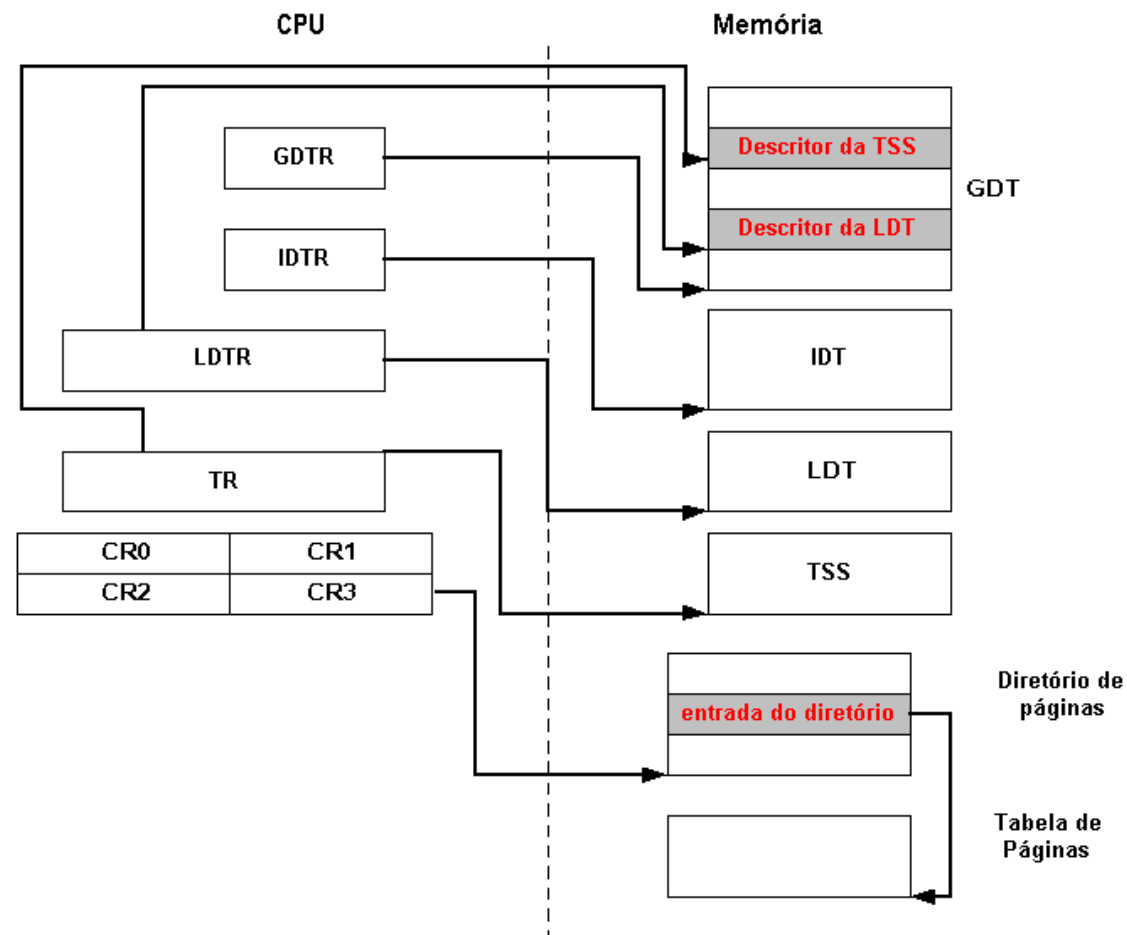
**IDTR** → registrador que aponta para a base da tabela de interrupções (IDT)

**TR** → registrador que aponta para a base do segmento de estado de uma tarefa (TSS)



# Modo protegido do 80386

## Acesso a memória em modo protegido





# Modo protegido do 80386

## Descrição das tabelas

- **Tabela de descritores globais (GDT)**  
Criada e gerenciada pelo sistema operacional, armazena descritores de segmento para os segmentos que podem ser acessados por todas as tarefas.
- **Tabela de descritores locais (LDT)**  
Armazena descritores utilizados para cada tarefa ativa.
- **Tabela de descritores de interrupção (IDT):**  
Armazena as informações para desviar o controle para a rotina de tratamento adequada.
- **Segmento de estado da tarefa (TSS)**  
É criado pelo sistema operacional para cada tarefa em execução concorrente e armazena o conteúdo dos registradores que serão carregados nos registradores do processador quando uma tarefa for selecionada para execução.
- **Diretório de páginas e tabela de páginas**  
Utilizados na implementação de memória virtual.



## Modo protegido do 80386

### Obtenção do endereço linear

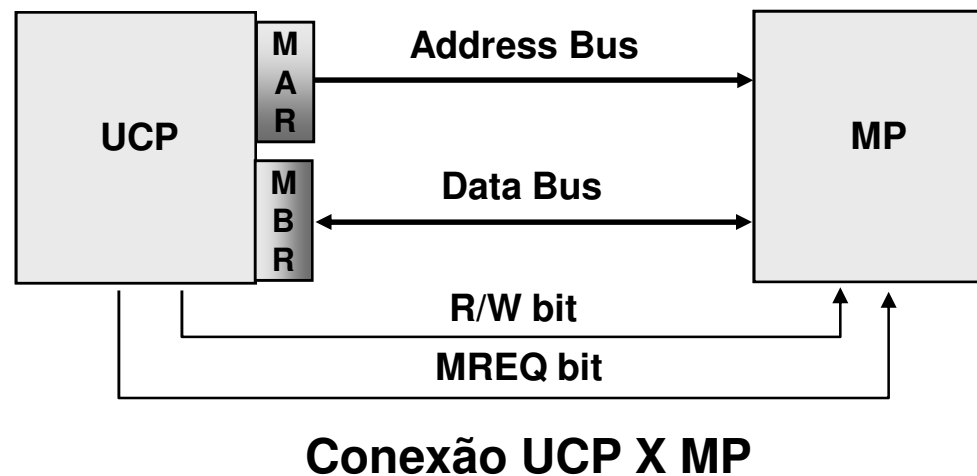
1. Obtém-se dos últimos 13 bits do seletor de segmento ( CS, DS, ES, FS, GS ou SS) o índice para acesso a tabela de descritores GDT ou LTD, conforme o terceiro bit do mesmo seletor:
  - 0 para acesso a GDT
  - 1 para acesso a LTD.
2. Cada uma das tabelas pode possuir 8192 descritores, uma vez que  $2^{13} = 8192$ .
3. Os primeiros 32 bits da tabela na posição indicada pelo seletor corresponderá ao endereço do segmento.
4. Este endereço base é adicionado ao conteúdo do registrador de deslocamento resultando no endereço linear.







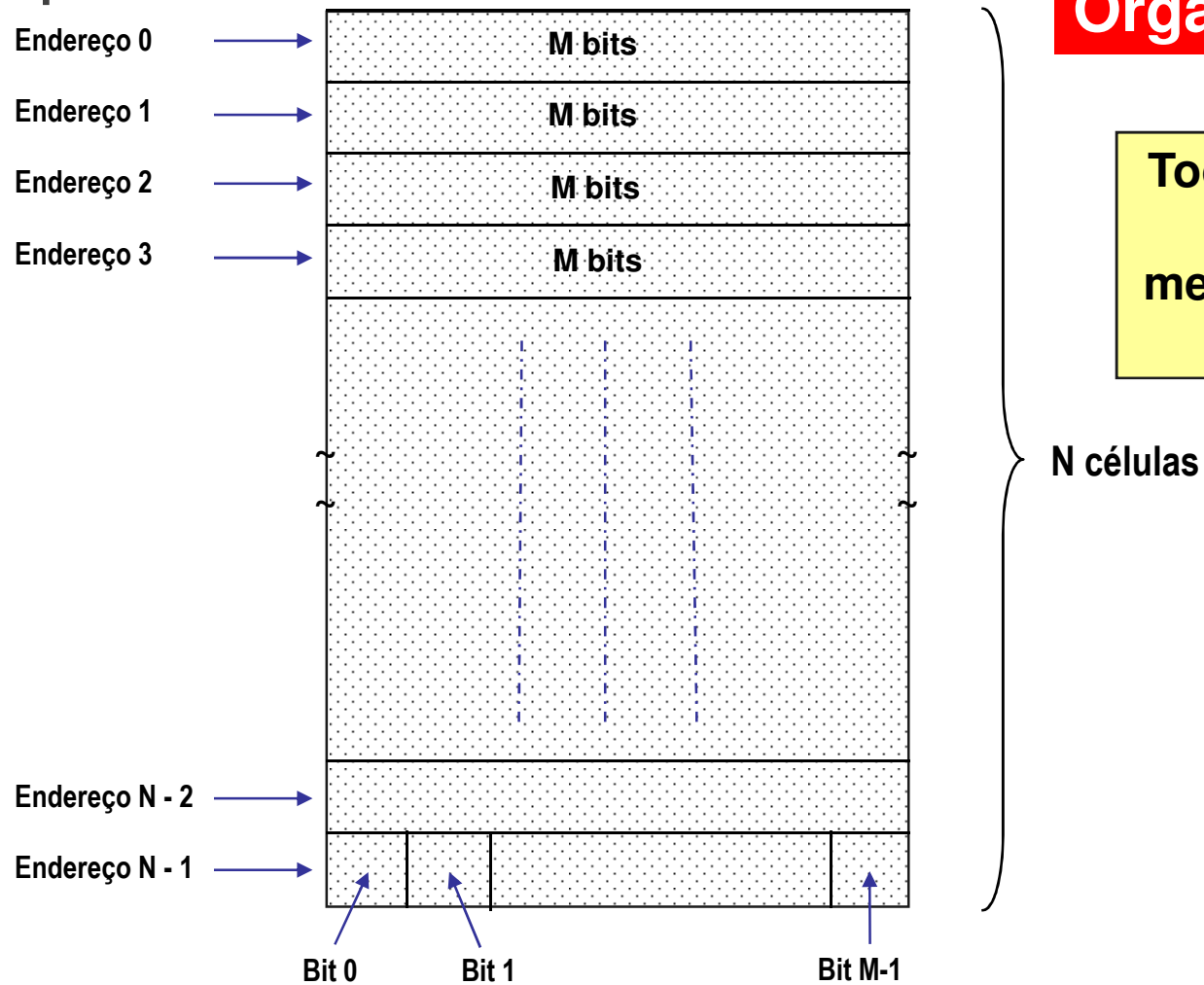
- armazena dados e programas
- também conhecida como memória real ou primária
- volátil
- endereçável por célula





# Memória

## Organização Básica

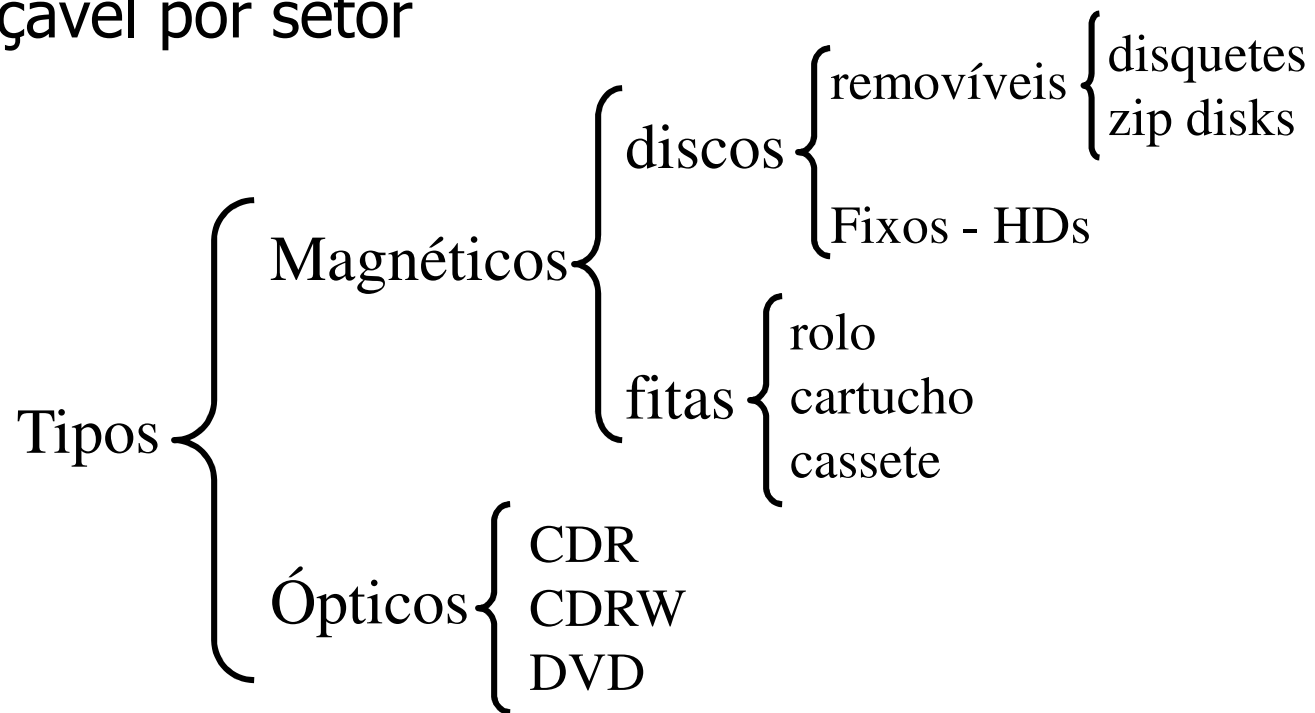


**Todas as células têm  
a  
mesma quantidade  $M$   
de bits**



# Armazenamento

- armazena de forma permanente dados e programas
- também conhecido como memória secundária
- não volátil
- endereçável por setor



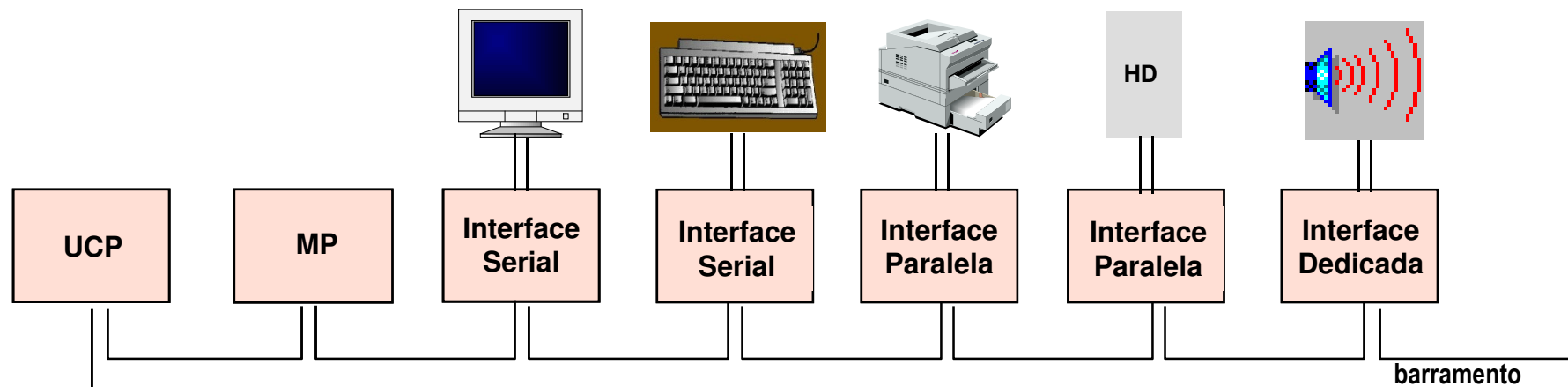


# Entrada e Saída

- transportam dados entre o computador (CPU/memória) e seu ambiente externo (periféricos) como:
  - memória secundária (ex. disco rígido)
  - teclado, monitor, ...
  - dispositivos de comunicação

## Requisitos:

- controladora
- driver



**Estrutura de um Sistema de I/O**



# Linguagens, programas e instruções

## Linguagem

Vocabulário e conjunto de regras de sintaxe e semântica usados para a construção de programas.

## Programa

Sequência de instruções organizadas de forma lógica para ao ser executado pelo computador (hardware) executa alguma função específica.

## Instrução

Sequência de bits que são interpretados pela UC e que disparam operações lógicas ou aritméticas a serem executadas pelos circuitos do hardware. (dependente do hardware)



# Programas e Instruções



*Sou o primeiro vírus excêntrico!*

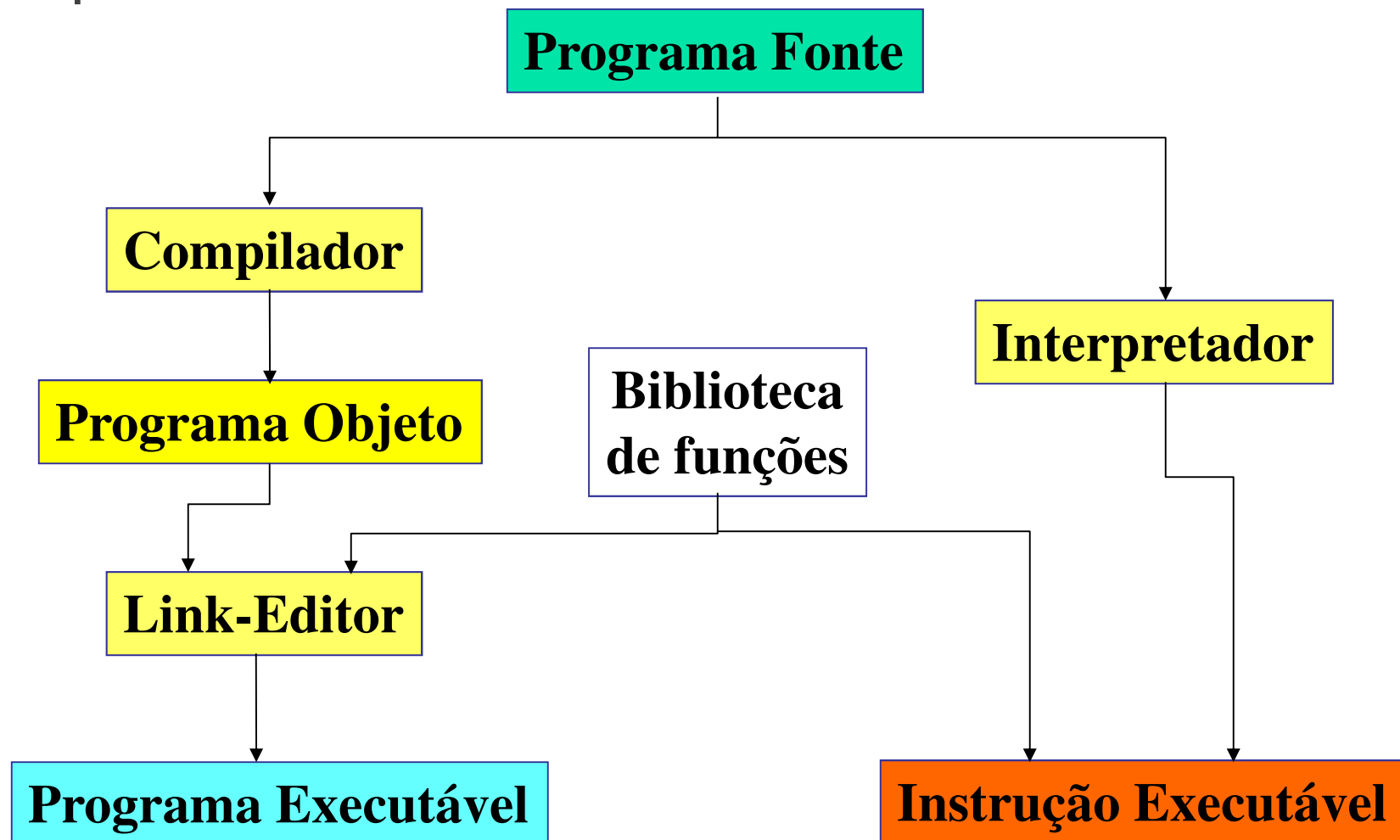
*Como nós, os excêntricos, não temos  
experiência em programação em  
computadores, este vírus só funciona  
a base da confiança.*

*Por favor, apague TODOS os arquivos do  
seu computador manualmente e envie esta  
mensagem a todos os membros da sua lista  
de endereços de correio eletrônico.*

*Obrigado pela colaboração. Manuel*



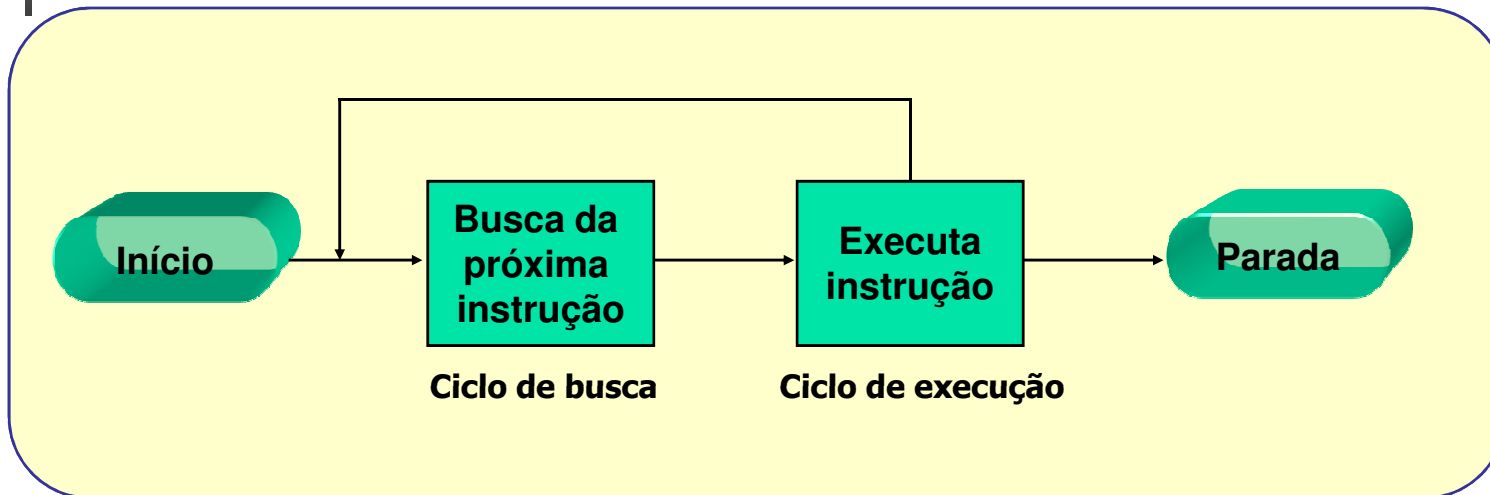
# Tradução de programas







# Ciclo básico de instruções

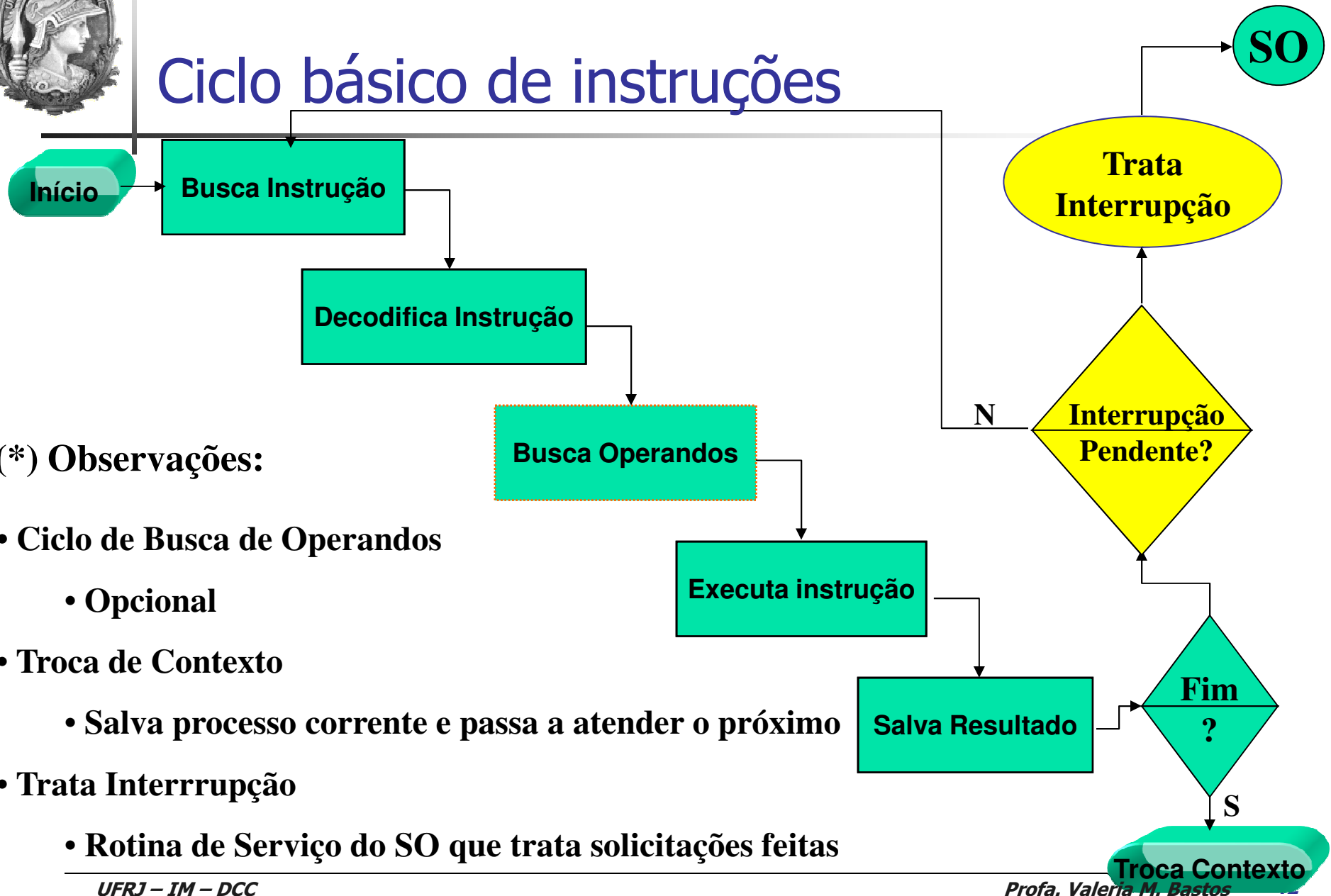


**O ciclo de instrução é uma máquina de estados em hardware:**

- Estado 1: Busca da próxima instrução
  - $IR \leftarrow M[PC]; PC \leftarrow PC + 1;$
- Estado 2: UC decodifica e executa IR
- Se fim  $\rightarrow$  parada, senão volta estado 1



# Ciclo básico de instruções



## (\*) Observações:

- Ciclo de Busca de Operandos
  - Opcional
- Troca de Contexto
  - Salva processo corrente e passa a atender o próximo
- Trata Interrupção
  - Rotina de Serviço do SO que trata solicitações feitas



# Tipos de instrução

- **Acesso à memória**
  - **Transferência de dados entre o processador e a memória**
- **Entrada / saída**
  - **Transferência de dados entre o processador e o dispositivo**
- **Tratamento de dados**
  - **Operações aritméticas ou lógicas**
- **Controle (desvios)**
  - **Alteração da seqüência de execução de instruções**

**Formatos de Instrução : 0 / 1 / 2 / 3 endereços**



# Instruções do 80386

## Modos de Endereçamento

■ Imediato	<i>ADD CH,5F</i>
■ Registrador	<i>ADD BX,DX</i>
■ Direto	<i>ADD VAR,BX</i>
■ Registrador indireto	<i>ADD CX,[BX]</i>
■ Indexado (ou base)	<i>ADD [SI+6].AL</i>
■ Base indexado com deslocamento	<i>ADD [BX+DI+5].DX</i>

## Tipos de instrução

■ Acesso à memória	<i>MOV AX, [TOTAL]</i>	<i>MOV [TOTAL] ,AX</i>
■ Entrada / saída	<i>IN AX, 72h</i>	<i>OUT DX, AX</i>
■ Tratamento de dados	<i>XOR AX, BX</i>	<i>ADD AX,BX</i>
■ Controle (desvios)	<i>JNE</i>	<i>JMP Label</i>



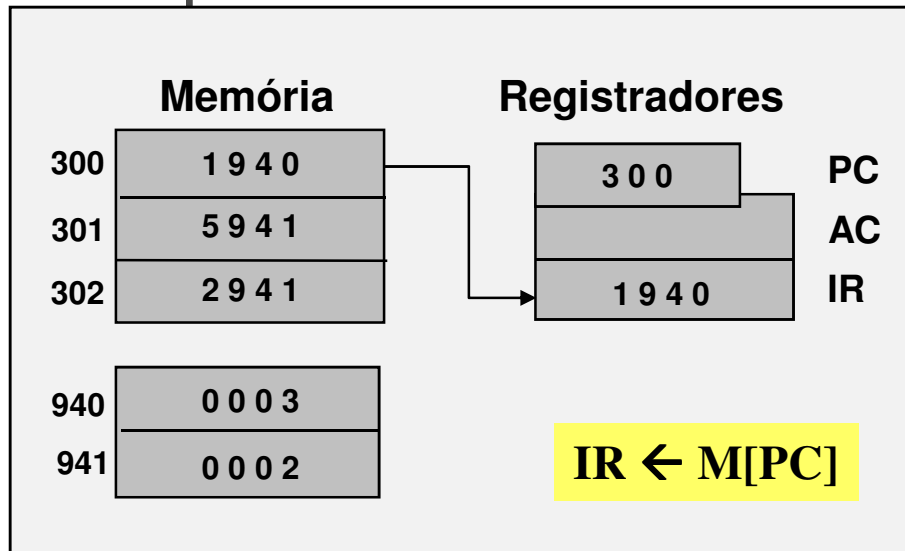
# Fluxo de Execução

- **Passo 1** – Busca da Instrução a ser executada  
 $IR \leftarrow M[PC]$
- **Passo 2** – Instrução corrente é decodificada  
UC decodifica IR
- **Passo 3** – Operandos são buscados  
 $R_x \leftarrow M[IR(end)]$
- **Passo 4** – O conteúdo de PC é atualizado  
 $PC \leftarrow PC + D$
- **Passo 5** – A instrução corrente é executada  
UC gera sinais de controle de acordo com o campo IR(Opcode)
- **Passo 6** – O resultado da operação é salvo  
 $M[?] \leftarrow \text{resultado ou pode permanecer em registrador}$
- **Passo 7** – Verifica término do programa  
Se positivo troca contexto e volta para passo 1
- **Passo 8** – Verifica existência de interrupção pendente  
Se positivo trata interrupção, senão volta para passo 1



# Fluxo de execução - Exemplo

## Passo 1



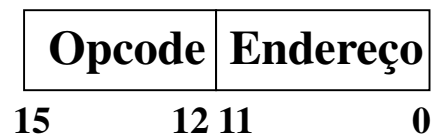
### Lista parcial dos códigos de operação

0001 = Carrega AC da memória

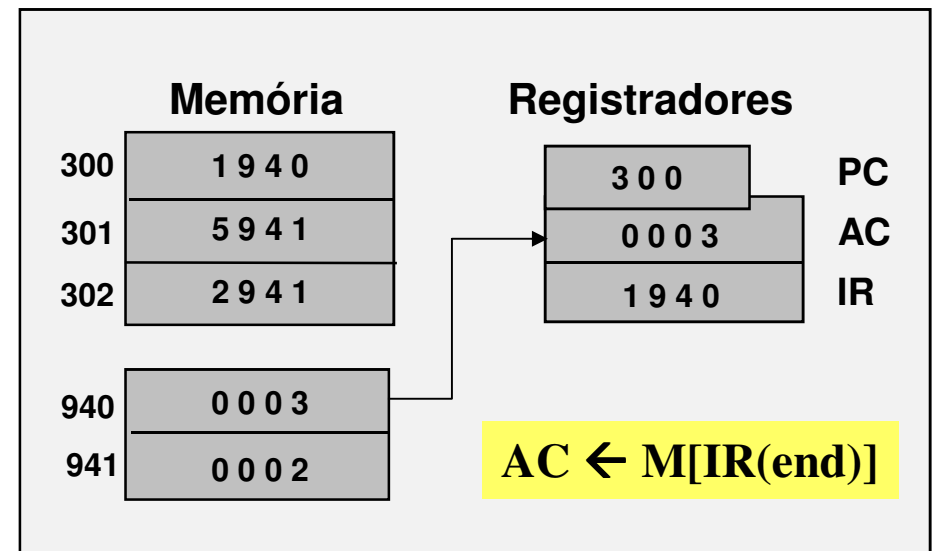
0010 = Armazena AC na memória

0101 = Adiciona em AC o valor da memória

### Formato da Instrução



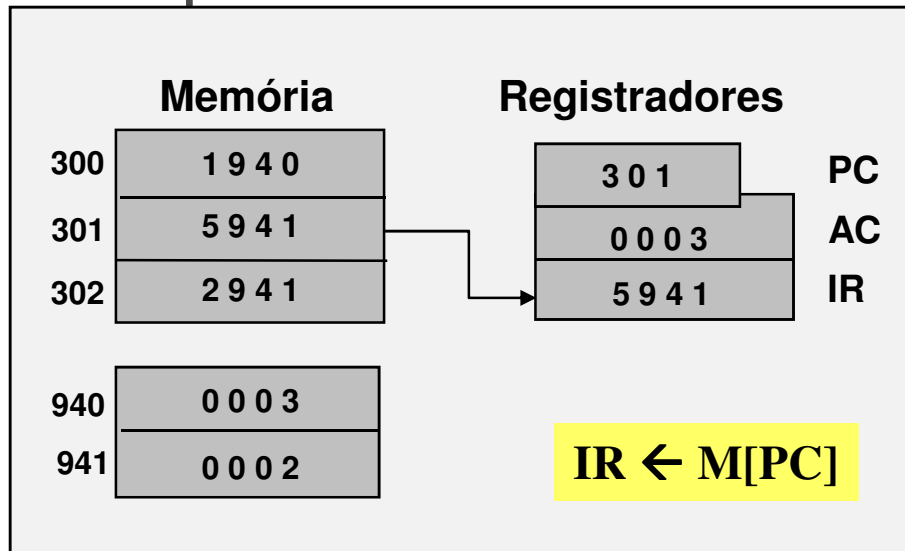
## Passo 2





# Fluxo de execução - Exemplo

## Passo 3



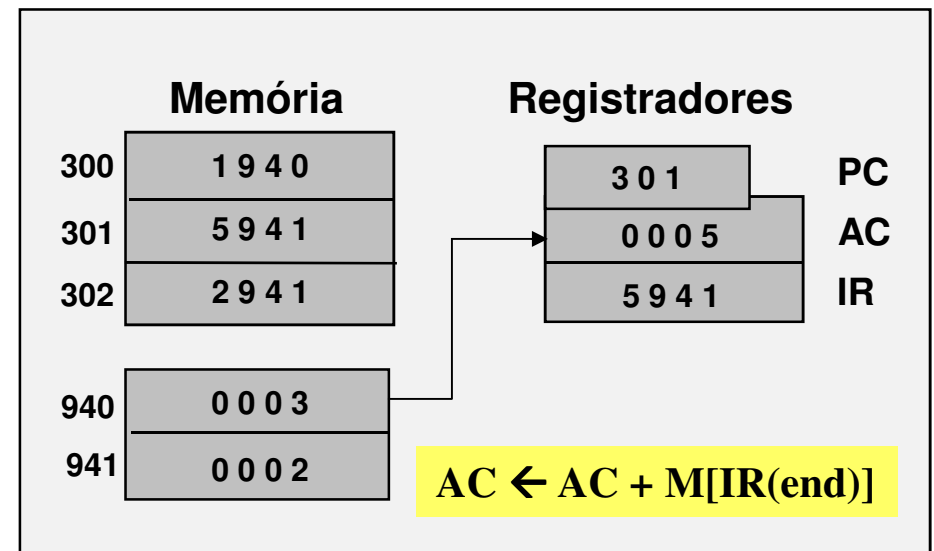
### Lista parcial dos códigos de operação

0001 = Carrega AC da memória

0010 = Armazena AC na memória

0101 = Adiciona em AC o valor da memória

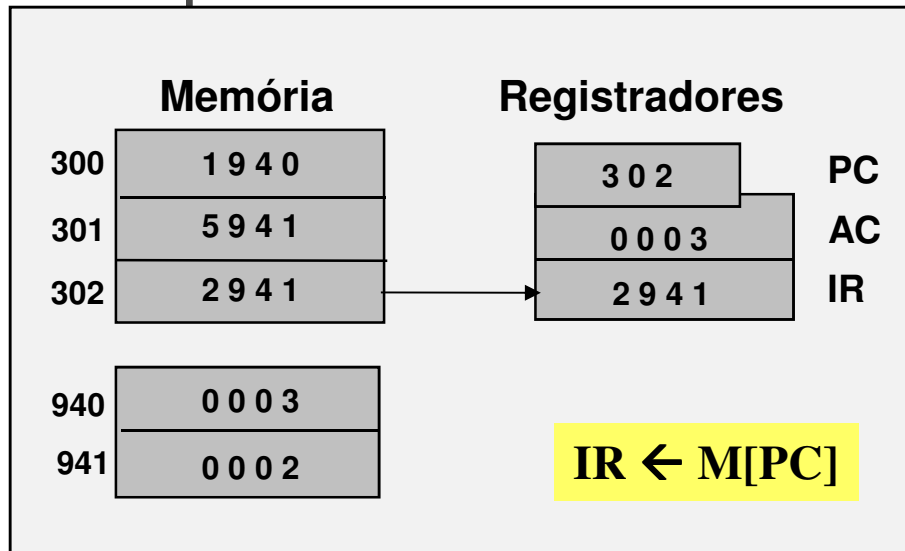
## Passo 4





# Fluxo de execução - Exemplo

## Passo 5



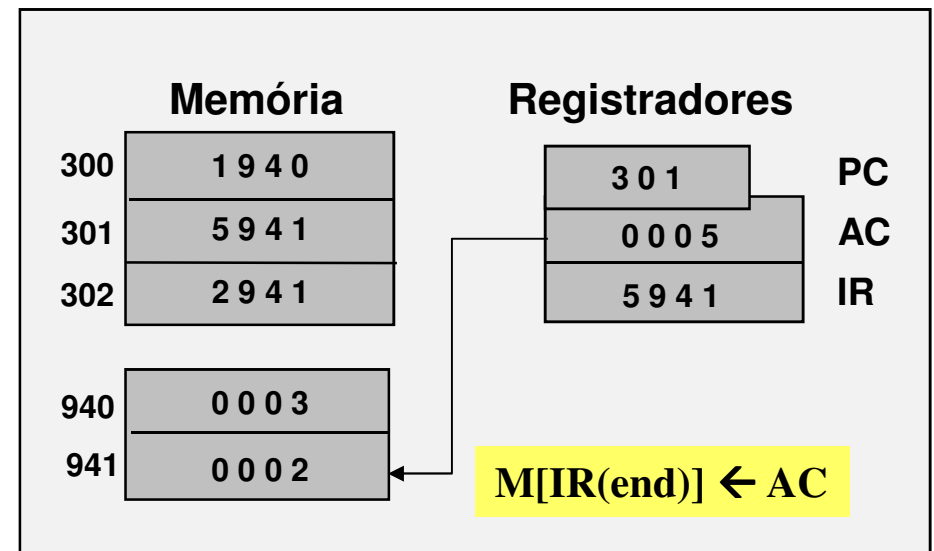
### Lista parcial dos códigos de operação

0001 = Carrega AC da memória

0010 = Armazena AC na memória

0101 = Adiciona em AC o valor da memória

## Passo 6







# Fundamentos de Hardware





# Tendências de Hardware

- Processadores com
  - múltiplos núcleos – família Core 2, Quad e I7
  - múltiplos pipelines
  - múltiplos níveis de cache
- Armazenamento
  - múltiplos discos em organização RAID
- Diversos
  - tela touchscreen
  - periféricos USB