

Assembly Language for Intel-Based Computers, 5th Edition

Kip Irvine

Capítulo 2: Arquitetura de Processamento IA-32

Slides prepared by the author

Revision date: June 4, 2006

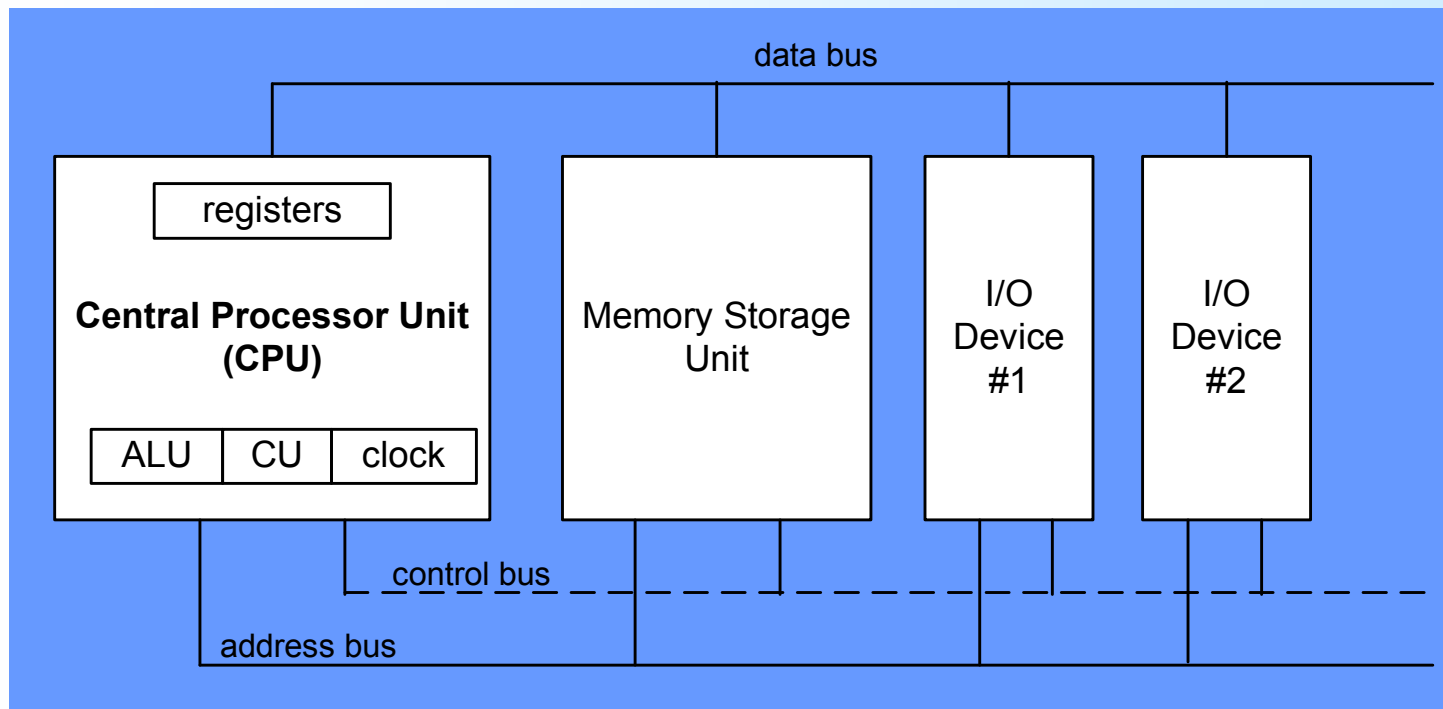
(c) Pearson Education, 2006-2007. All rights reserved. You may modify and copy this slide show for your personal use, or for use in the classroom, as long as this copyright statement, the author's name, and the title are not changed.

Resumo do Capítulo

- **Conceitos Gerais**
- Arquitetura IA-32
- Gerenciamento de memória IA-32
- Componentes de um Microcomputador IA-32
- Sistema de Entrada-Saída

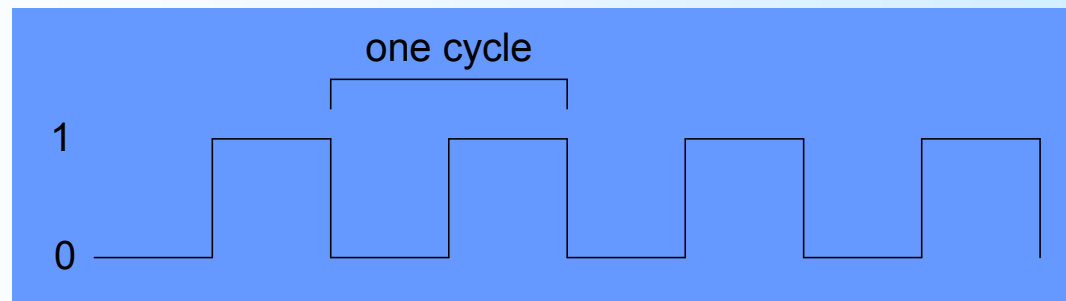
Projeto de Microcomputador Básico

- As operações de CPU são sincronizadas pelo clock
- A unidade de controle coordena os passos da sequência de execução
- ALU realiza o processamento aritmético e lógico



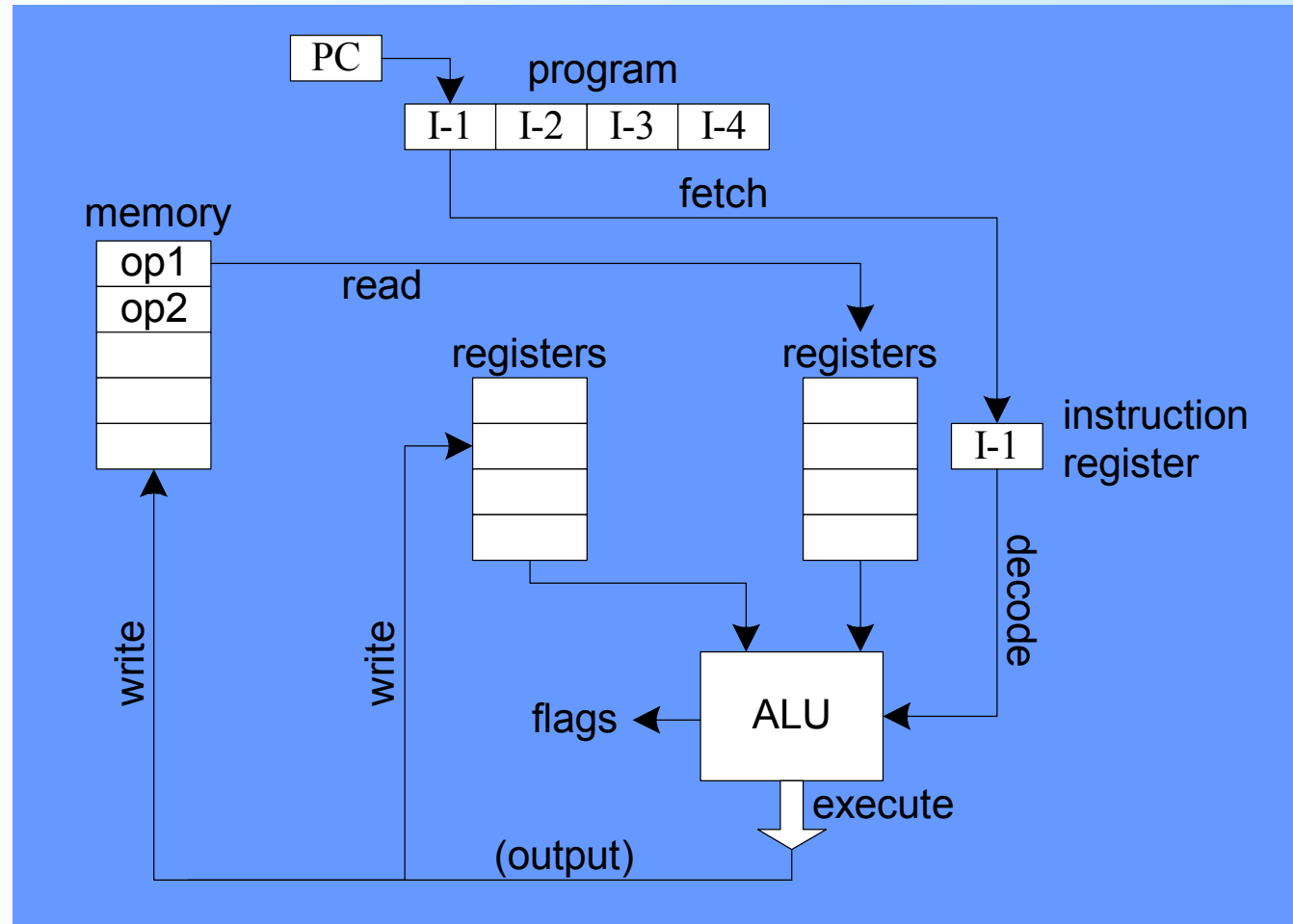
Clock

- Sincroniza todas as operações de CPU e barramento
- O tempo de ciclo mede a duração de uma única operação
- O clock é usado para disparar eventos



Ciclo de execução de instrução

- Fetch
- Decode
- Fetch operands
- Execute
- Store output



Pipeline Multi-Estágio

- O pipeline torna possível a execução de instruções em paralelo
- A execução de uma instrução é dividida em estágios discretos

Exemplo de um
processador **não-pipeline**.
Muitos ciclos perdidos !

		Stages					
Cycles		S1	S2	S3	S4	S5	S6
	1	I-1					
	2		I-1				
	3			I-1			
	4				I-1		
	5					I-1	
	6						I-1
	7	I-2					
	8		I-2				
	9			I-2			
	10				I-2		
	11					I-2	
	12						I-2

Execução em pipeline

- Uso eficiente dos ciclos, com alto **throughput** de instruções:

		Stages					
Cycles		S1	S2	S3	S4	S5	S6
	1	I-1					
	2	I-2	I-1				
	3		I-2	I-1			
	4			I-2	I-1		
	5				I-2	I-1	
	6					I-2	I-1
	7						I-2

Para k estágios e n instruções, o número de ciclos requeridos é:

$$k + (n - 1)$$

Ciclos perdidos (pipeline)

- Quando um dos estágios requer dois ou mais ciclos, novamente perdem-se ciclos.

		Stages					
		exe					
Cycles		S1	S2	S3	S4	S5	S6
	1	I-1					
	2	I-2	I-1				
	3	I-3	I-2	I-1			
	4		I-3	I-2	I-1		
	5			I-3	I-1		
	6				I-2	I-1	
	7				I-2		I-1
	8				I-3	I-2	
	9				I-3		I-2
	10					I-3	
	11						I-3

Para k estágios e n instruções, o número de ciclos requerido é:

$$k + (2n - 1)$$

Superescalar

Um processador superescalar faz uso de múltiplos pipelines de execução. No exemplo, note que o estágio S4 tem pipelines u e v.

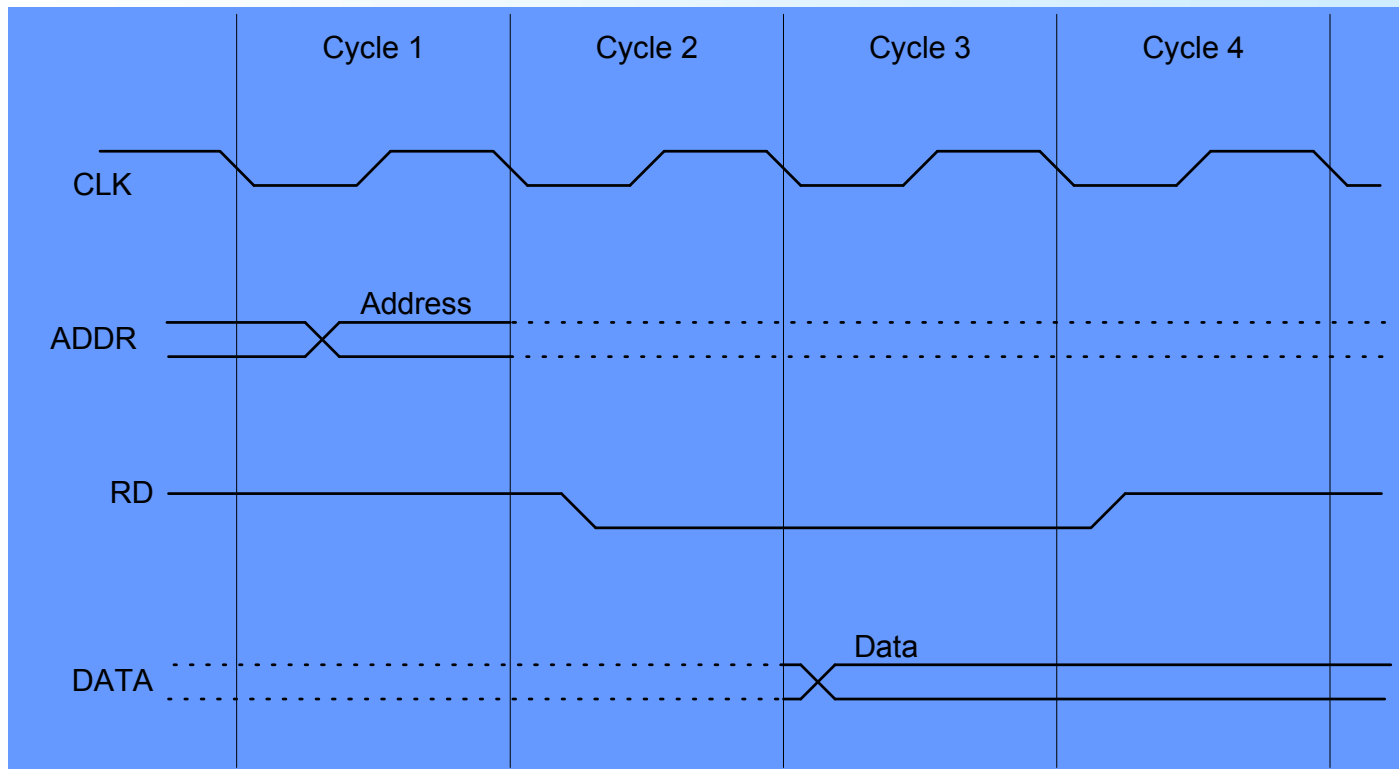
		Stages						
		S1	S2	S3	S4		S5	S6
Cycles	1	I-1						
	2	I-2	I-1					
	3	I-3	I-2	I-1				
	4	I-4	I-3	I-2	I-1			
	5		I-4	I-3	I-1	I-2		
	6			I-4	I-3	I-2	I-1	
	7				I-3	I-4	I-2	I-1
	8					I-4	I-3	I-2
	9						I-4	I-3
	10							I-4

Para k estágios e n instruções, o número de ciclos requeridos é:

$$k + n$$

Leitura da Memória

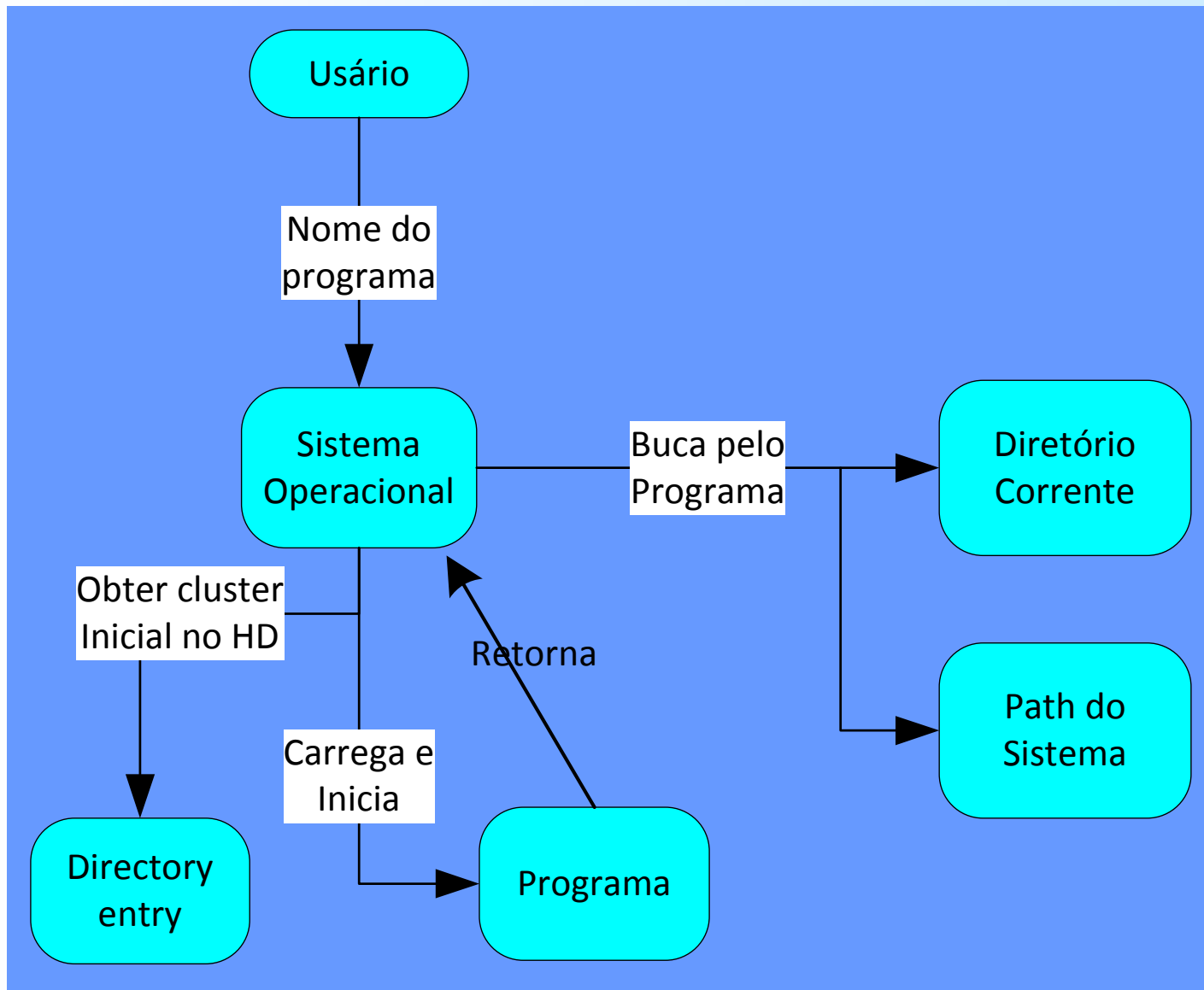
- Múltiplos ciclos são requeridos para a leitura de memória, pois a memória é muito mais lenta que a CPU. Os passos são:
 - Colocar o endereço no barramento de endereço
 - Acionar a linha Read (RD) para zero
 - CPU espera um ciclo pela resposta de memória
 - A linha Read (RD) vai para 1, indicando que o dado está presente no barramento de dados



Memória Cache

- RAM estática de alta velocidade e alto custo, que pode estar dentro ou fora do chip de CPU.
 - Cache nível-1 : dentro da CPU
 - Cache nível-2 : fora da CPU
- Acerto (hit): quando o dado a ser lido se encontra no cache
- Falta (miss): caso contrário.

Como um programa é executado



Multitarefa

- O Sistema Operacional (SO) pode rodar múltiplos programas ao mesmo tempo.
- Múltiplos threads de execução dentro de um mesmo programa.
- O Scheduler atribui uma dada quantidade de tempo de CPU para cada programa em execução.
- Comutação rápida de tarefas
 - ilusão de que todos os programas estão sendo executados simultaneamente
 - o processador deve suportar a comutação de tarefas.

IA-32-Arquitetura do processador

- Arquitetura IA32 (ou X86):
 - Arquitetura de 32 bits dos processadores Intel;
 - Compatibilidade de código binário;
 - Maioria dos processadores vendidos nos últimos 25 anos;
 - Arquitetura a ser usada p/ programação assembly neste curso;

IA-32- Modos de Operação

- **Modo protegido (Windows, Linux)**
- Modo de endereçamento real (MS-DOS)
- Modo de gerenciamento do sistema
(gerenciamento de potência, segurança do sistema, diagnóstico)

- **Modo Virtual-8086**
 - Híbrido de protegido
 - Cada programa tem seu próprio computador 8086

IA-32- Ambiente básico de execução

- Memória endereçável
- Registradores de propósito geral
- Registradores de índice e base
- Uso especializado de registradores
- Flags de Status
- Registradores de ponto-flutuante, MMX e XMM

IA-32- Memória endereçável

- Modo protegido
 - 4 GB
 - Endereçamento de 32-bits
- Modos de endereçamento real e Virtual-8086
 - Espaço de 1 MB
 - Endereçamento de 20-bits

IA-32- Registradores de propósito geral

Disponíveis na CPU, e visíveis ao usuário (programador e compilador)

32-bit General-Purpose Registers

EAX
EBX
ECX
EDX

EBP
ESP
ESI
EDI

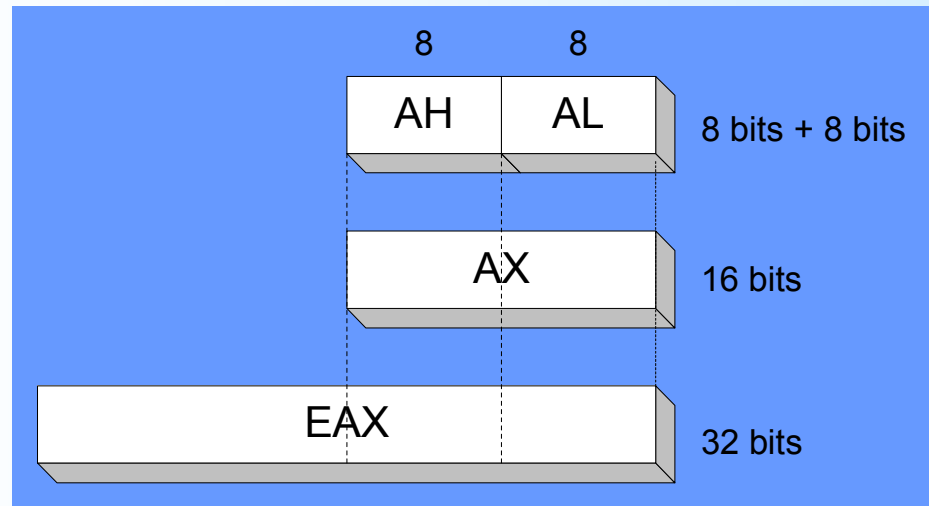
16-bit Segment Registers

EFLAGS
EIP

CS	ES
SS	FS
DS	GS

IA-32- Acessando parte dos registradores

- Usa nomes diferentes para 8-bits, 16-bits ou 32-bits
- Aplicados para EAX, EBX, ECX e EDX



32-bit	16-bit	8-bit (high)	8-bit (low)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL

IA-32- Registradores de Índice e Base

- Referenciados apenas p/ 16 e 32 bits:

32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP

IA-32- Uso de registradores

- Uso geral
 - EAX,EBX,EDX: geral
 - ECX: geral e contador em instrução loop
 - ESI, EDI – geral e índice de vetores
 - ESP – ponteiro de pilha
 - EBP – ponteiro de estrutura (pilha)
- Uso p/ referenciar Segmentos de Memória
 - CS – segmento de código
 - DS – segmento de dados
 - SS – segmento de pilha
 - ES, FS, GS – segmentos adicionais

IA-32- Uso de registradores

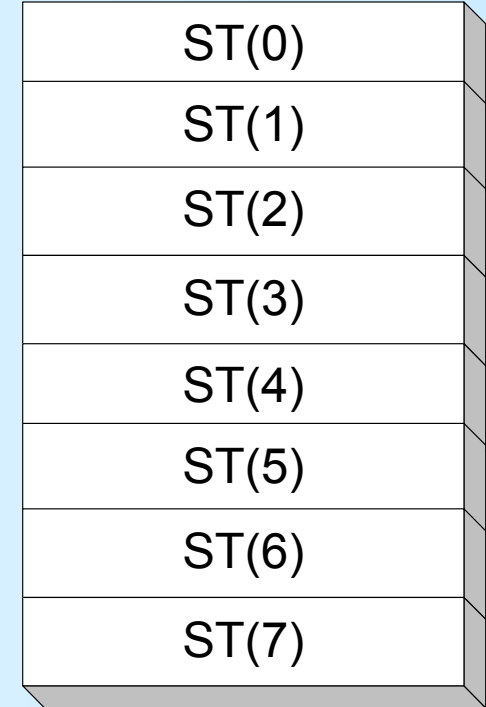
- Uso Especializado:
 - EIP – ponteiro da instrução a ser executada (PC)
 - EFLAGS
 - Flags de status e controle
 - cada flag é um único bit

IA-32- Flags de Status

- Carry
 - Estouro em aritmética sem sinal
- Overflow
 - Estouro em aritmética com sinal
- Sign
 - Resultado negativo
- Zero
 - Resultado é zero
- Auxiliary Carry
 - carry do bit3 para o bit 4
- Parity
 - Soma de bits 1 é um número par

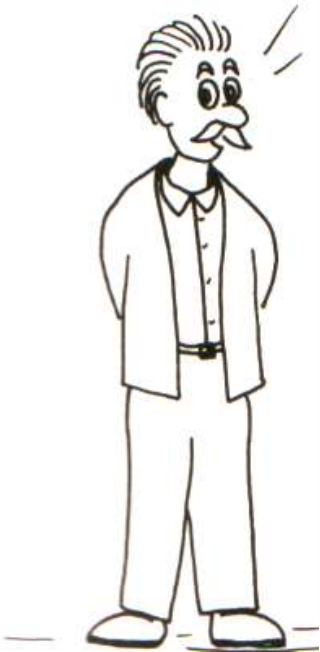
IA-32- Registradores de ponto-flutuante, MMX, e XMM

- Oito registradores de ponto-flutuante de 80-bits
 - ST(0), ST(1), . . . , ST(7)
 - em estrutura de pilha
 - usados para aritmética de ponto-flutuante
- Oito registradores MMX de 64-bits
- Oito registradores XMM de 128-bits para operações SIMD(single-instruction-multiple-dada)



História dos microprocessadores Intel

- Intel 8086, 80286
- Família de processadores IA-32
- Família de processadores P6
- CISC e RISC

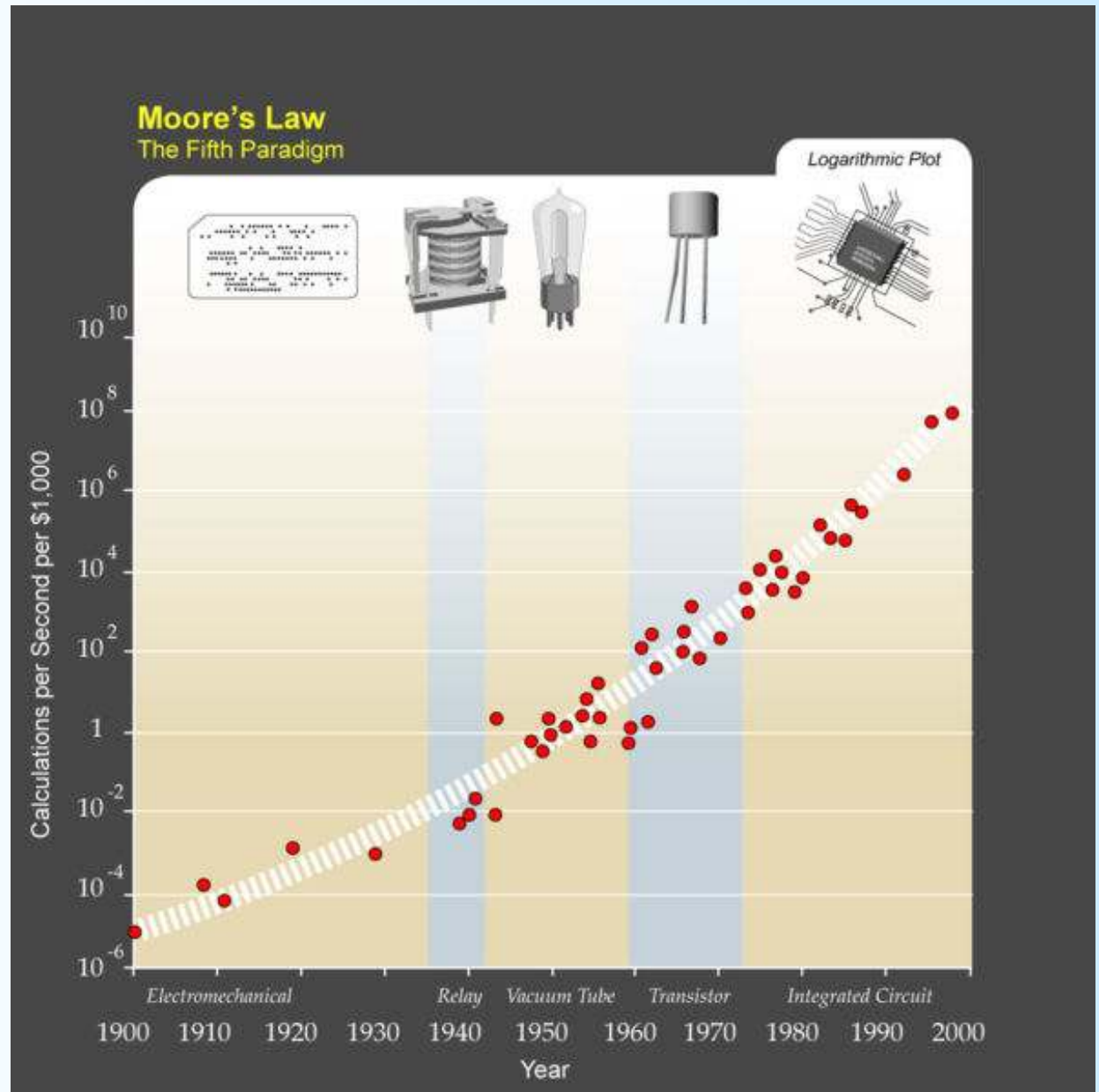


IA-16 – arquitetura Intel de 16 bits (8086, 8088, 80286)

IA-32 – arquitetura Intel de 32 bits (80386, 80486, Pentium,...)

IA-64 – arquitetura Intel de 64 bits (Itanium)

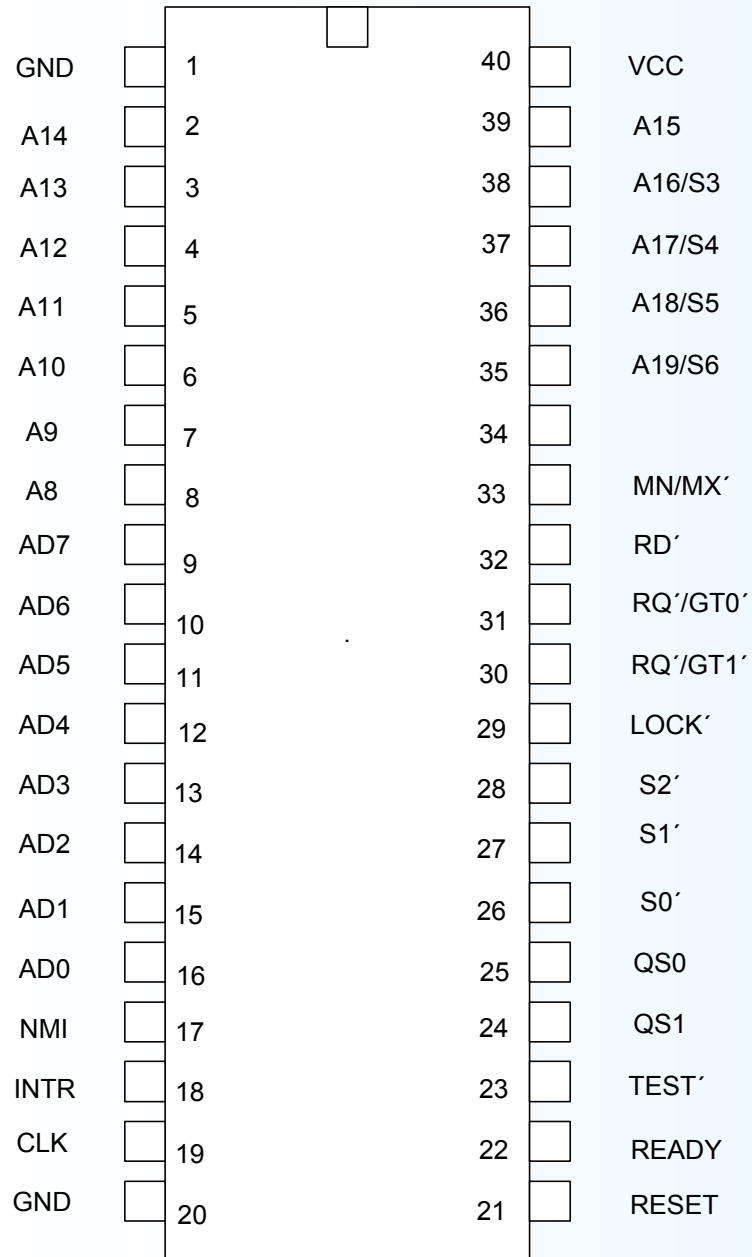
Lei de Moore



Primeiros microprocessadores Intel

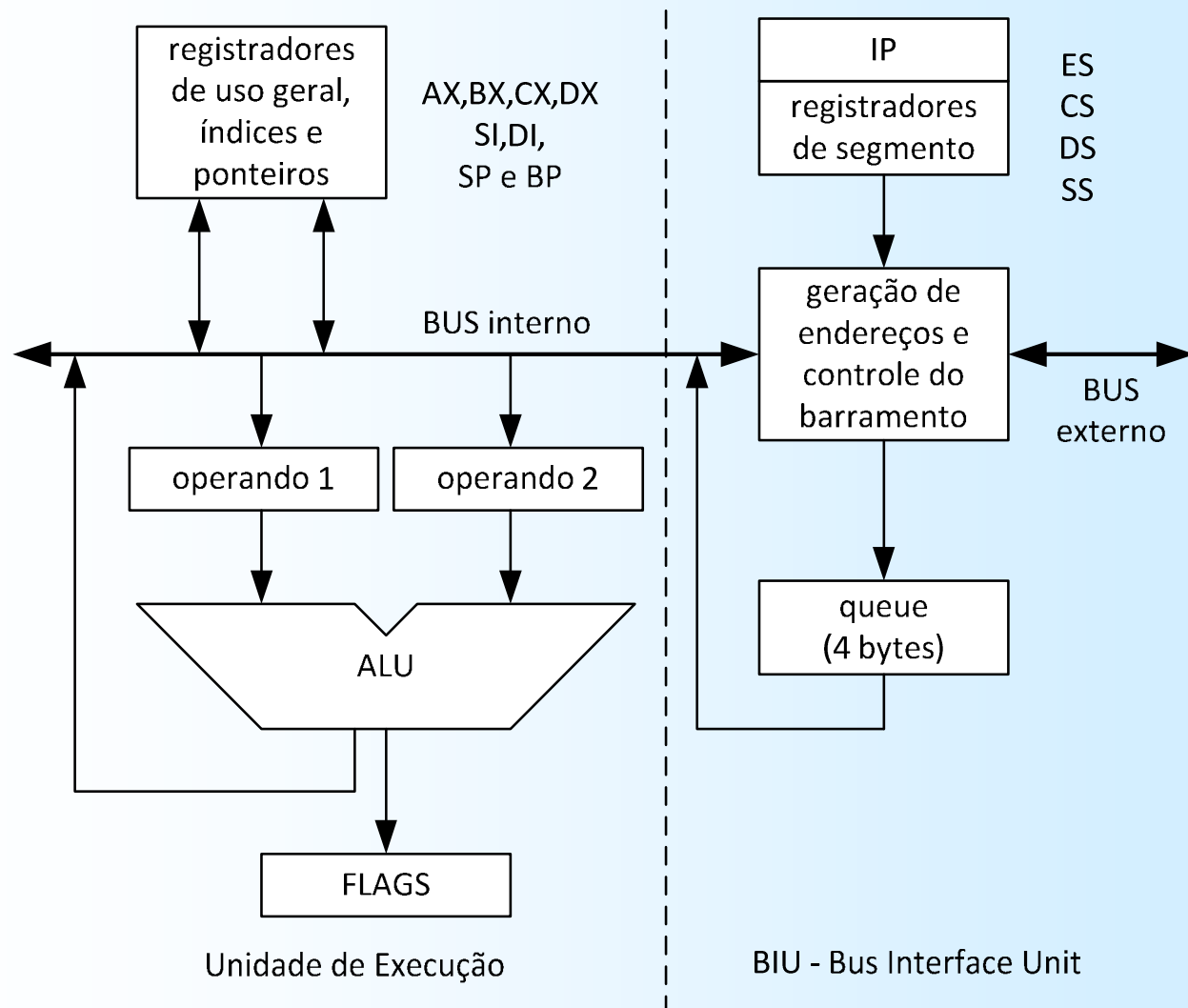
- Intel 8080
 - RAM endereçável até 64K
 - Registradores de 8-bits
 - Sistema operacional CP/M
 - Arquitetura de barramento S-100
 - Discos flexíveis de 8-polegadas!
- Intel 8086/8088
 - IBM-PC Usava 8088
 - RAM endereçável até 1 MB
 - Registradores de 16-bits
 - Barramento de dados de 16-bits (8-bits para 8088)
 - Unidade de ponto-flutuante separada (8087)

8088

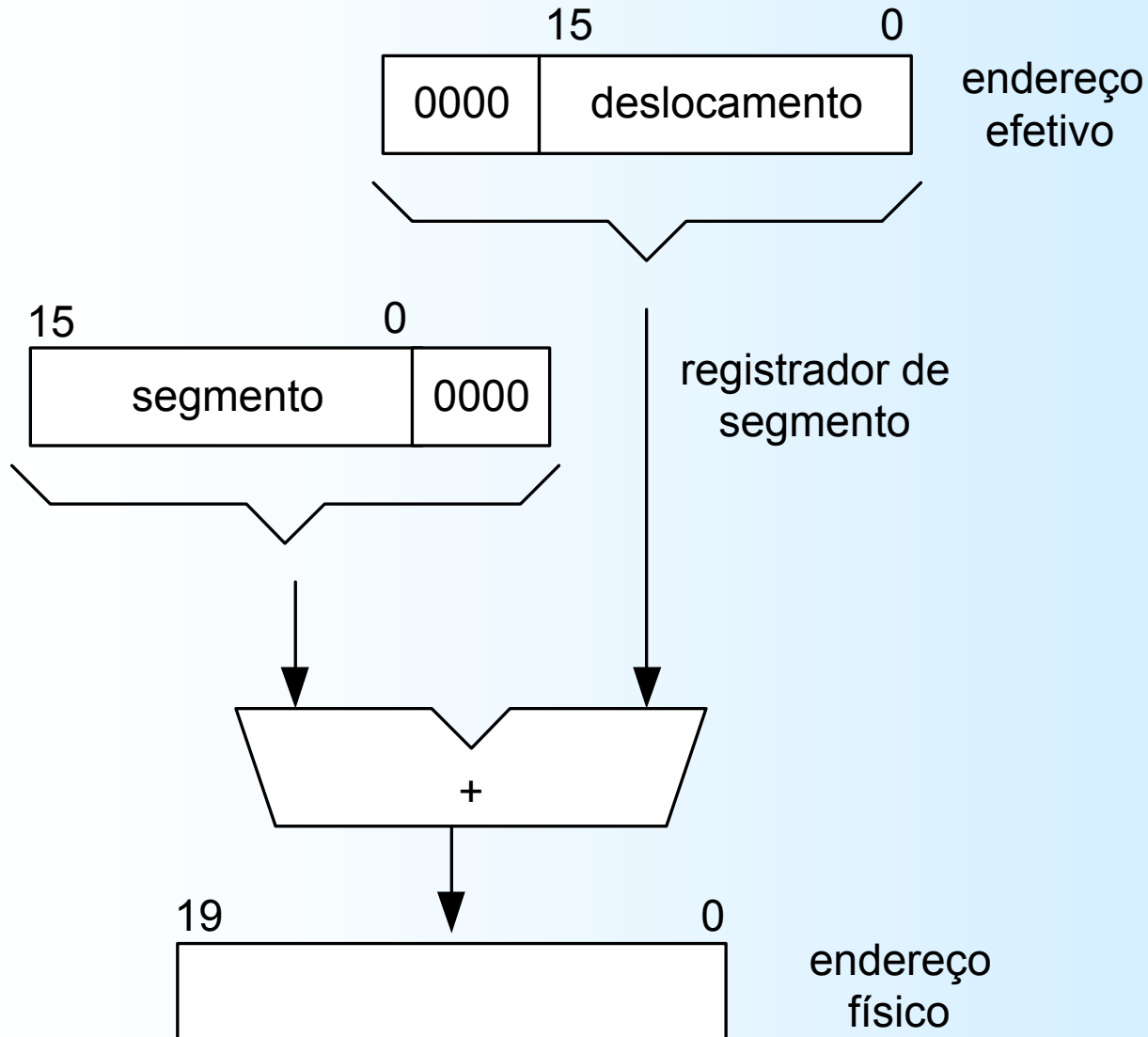


S2'	S1'	S0'	TIPO DE CICLO DE BARRAMENTO
0	0	0	Reconhecimento de interrupção
0	0	1	Leitura de porto de E/S
0	1	0	Escrita de porto de E/S
0	1	1	Halt
1	0	0	Acesso a código
1	0	1	Leitura de memória
1	1	0	Escrita de memória
1	1	1	Liberação de barramento

Arquitetura do 8086/8088



Cálculo do endereço físico do 8086/8088



Registradores do 8086/8088

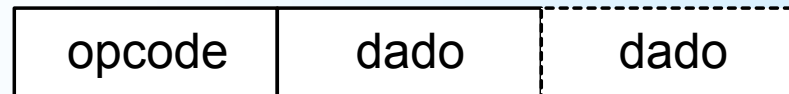
	← 16 bits →		
AX	AH	AL	Acumulador principal
BX	BH	BL	Aritmética, apontadores
CX	CH	CL	Loops
DX	DH	DL	Multiplicação e Divisão
SI			Apontador fonte para operações com cadeia de caracteres
DI			Apontador destino para operações com cadeia de caracteres
BP			Apontador de Base
SP			Apontador de Pilha
CS			Segmento de Código
DS			Segmento de Dados
SS			Segmento de Pilha
ES			Segmento Extra
IP			Contador de Programa
FLAGS			Palavra de Status

Endereçamento

- **Endereçamento: como um dado valor é especificado na instrução ?**
 - **Várias formas....**
 - **Valor constante**
 - **Valor em um registrador**
 - **Valor na memória (vária formas novamente...)**

Endereçamento imediato

- O operando é um dado constante contido na própria instrução



- Ex: `mov cl, 10h`
 - o dado especificado na instrução (constante) é carregado no registrador CL.

Endereçamento de registrador

O operando é contido num registrador

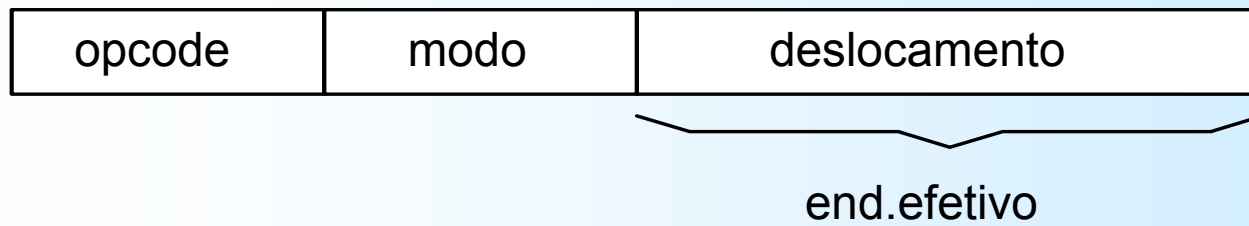


Ex: `mov bx, ax`

- o conteúdo do registrador AX é carregado no registrador BX.

Dado contido na memória

- Endereçamento direto – o endereço efetivo de memória é o deslocamento

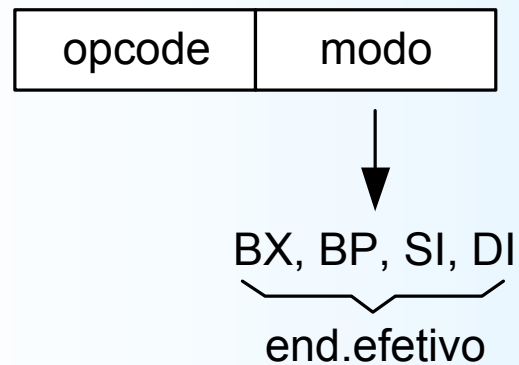


Ex: mov al, (2000h)

- o conteúdo de memória no endereço DS:2000h é transferido para o registrador AL.

Dado contido na memória

- Endereçamento indireto – o endereço efetivo é contido num registrador

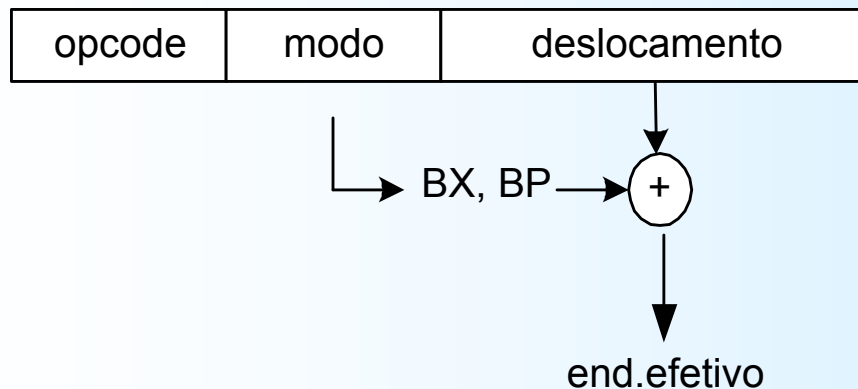


Ex: `mov al, (bx)`

- o conteúdo de memória no endereço DS: BX é carregado no registrador AL.

Dado contido na memória

- Endereçamento por base – o endereço efetivo é a soma de um registrador e o deslocamento

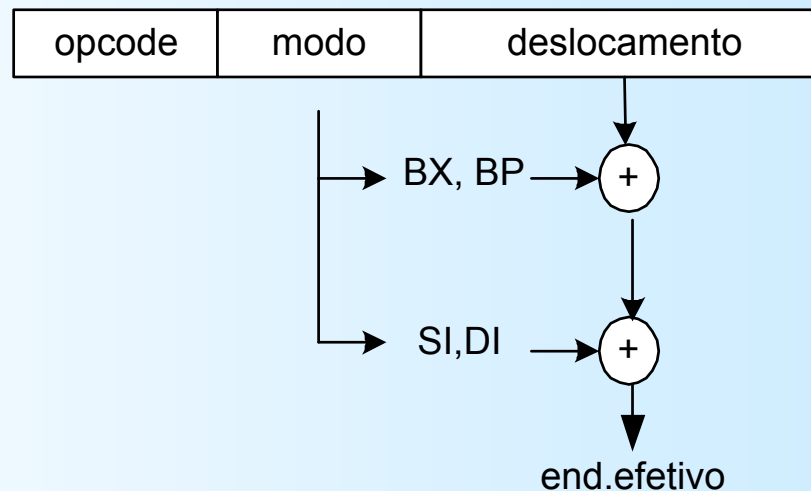


Ex: `mov cl, (bx + 04)`

- o conteúdo de memória no endereço DS: `BX + 04` é carregado no registrador CL.

Dado contido na memória

- Endereçamento indexado – o endereço efetivo é a soma de dois registradores, com ou sem deslocamento



Ex: `mov ah, (bx + si + 10h)`

- o conteúdo de memória dado pelo endereço DS: BX + SI + 10H é carregado no registrador AH.

Dado contido na memória

Endereçamento de string (cadeia) – os endereços de memória são implícitos, usando DS:SI para fonte e ES:DI para destino, para instruções de cadeia



opcode

Ex: movsb

Significado: copia um byte do endereço dado por DS:SI para o endereço ES:DI.

Endereçamento de instrução

- Endereçamento relativo – o endereço é relativo ao contador de programa, e é usado para desvio

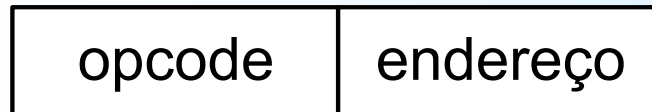
opcode	deslocamento
--------	--------------

Ex: jc 20h

Significado: desvia de uma instrução no endereço 20H à frente, caso ocorra o vai-um (carry = 1).

Endereçamento de entrada/saída

- Endereçamento de porto de entrada/saída(E/S) modo direto



Ex1: in al, 20h

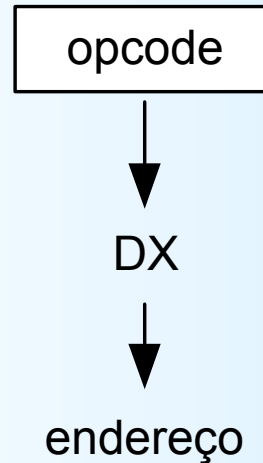
- o byte de dados presente na porta 20H é carregado no registrador AL.

Ex2: out 50h, ax

- a palavra de 16 bits contida em AX é transferida para a porta 50H.

Endereçamento de entrada/saída

Endereçamento de porto de E/S, indireto, por registrador DX



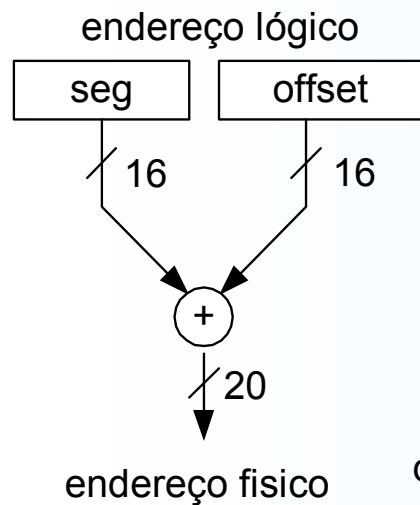
Ex1: in al, dx

- o byte contido na porta cujo endereço é dado pelo conteúdo do registrador DX, é carregado no registrador AL.

Ex2: out dx, al

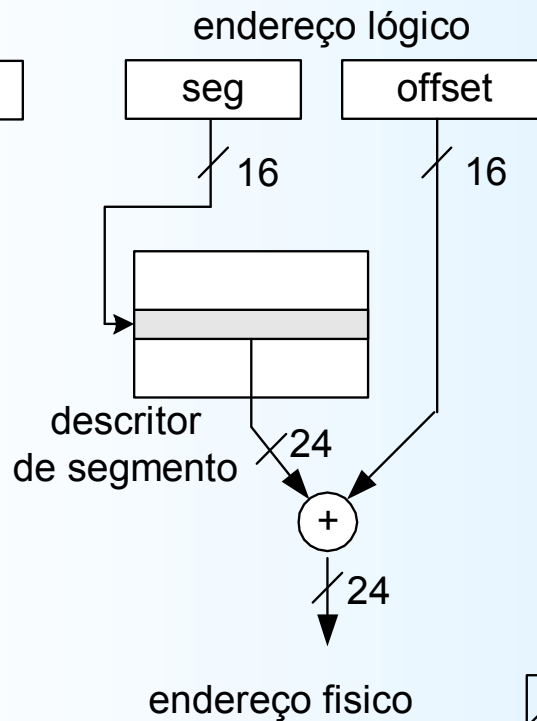
- o byte contido no registrador AL é carregado para a porta cujo endereço é dado pelo conteúdo do registrador DX.

Modo Real
8086/8088

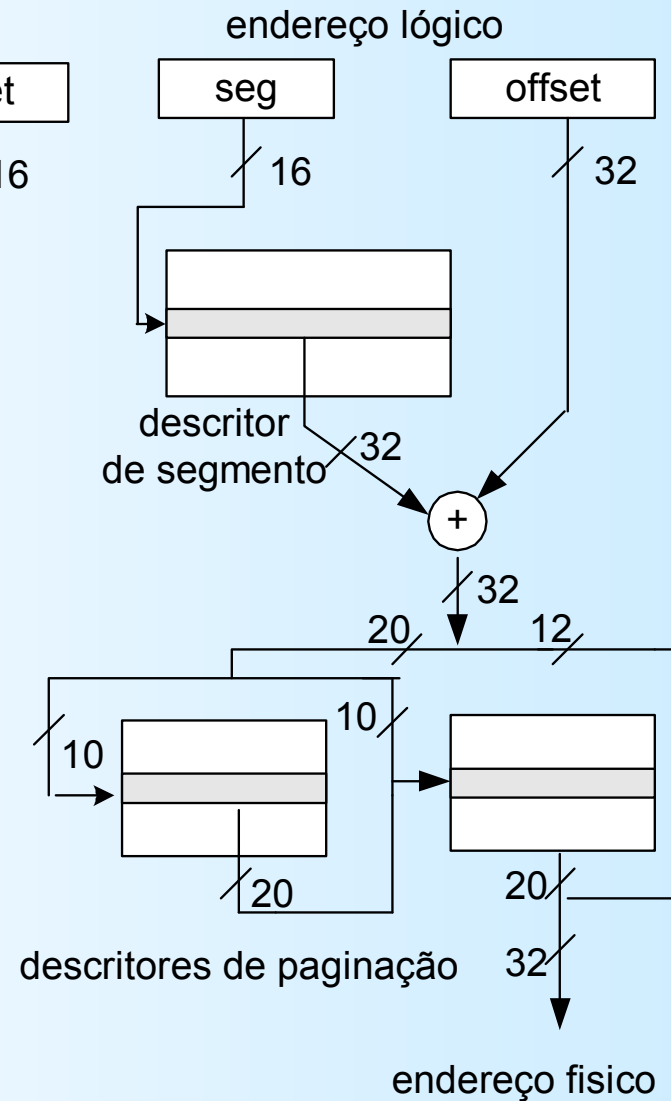


modo protegido

80286



386, 486, Pentium



Cálculo de endereço

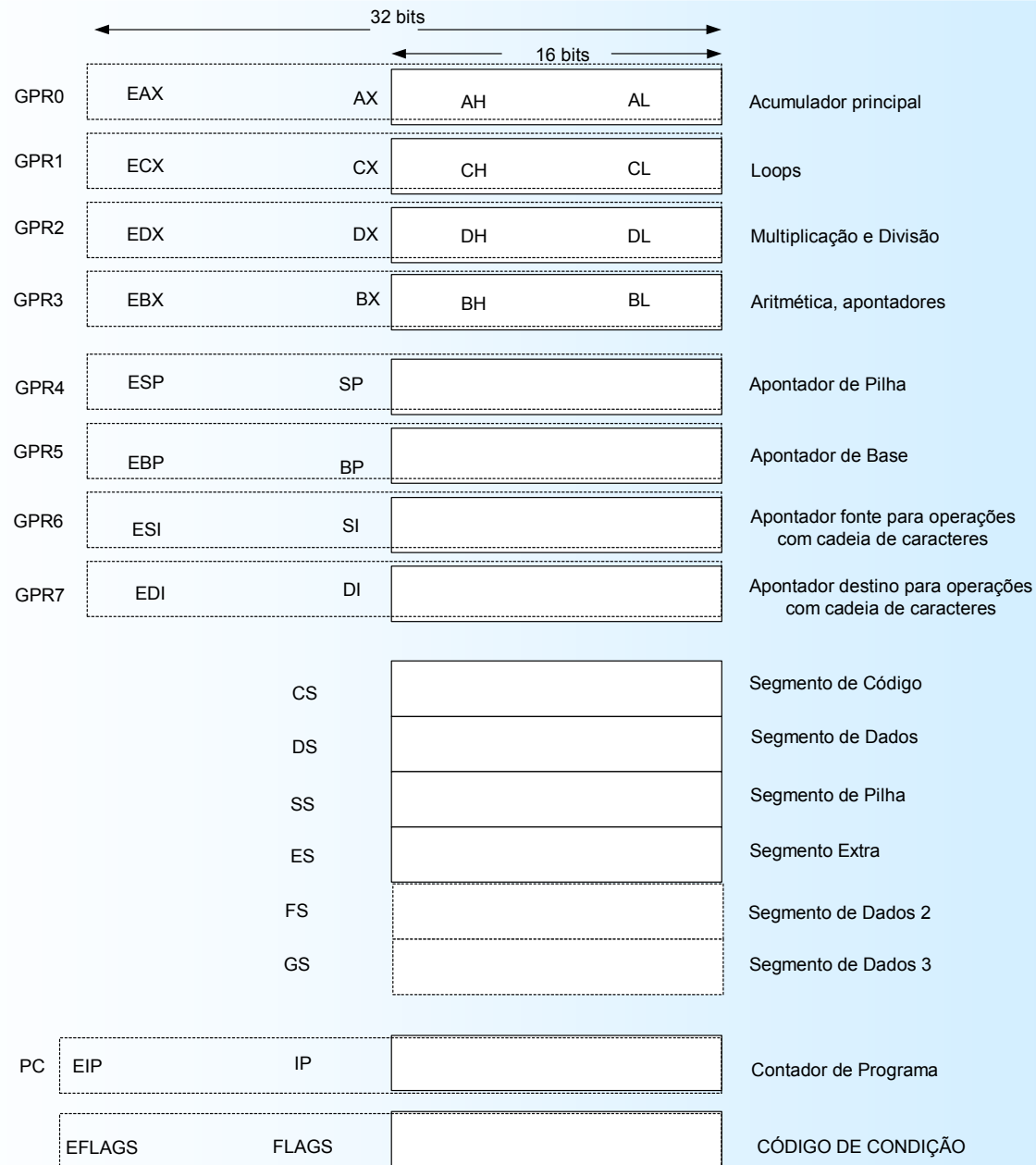
O IBM-AT

- Intel 80286
 - RAM endereçável até 16 MB
 - Memória protegida
 - Várias vezes mais rápido que 8086
 - Introduziu o barramento IDE
 - Unidade de ponto-flutuante 80287

Família Intel IA-32

- Intel386
 - RAM endereçável até 4 GB, registradores de 32-bits, paginação (memória virtual)
- Intel486
 - Pipeline de instruções
- Pentium
 - superescalar, barramento de endereço de 32-bits, caminho de dados interno de 64-bits

IA32 Regis- trado- res



Família Intel P6

- Pentium Pro
 - Técnicas avançadas de otimização em microcódigo
- Pentium II
 - Conjunto de instruções MMX (multimídia)
- Pentium III
 - Instruções SIMD (streaming extensions)
- Pentium 4 e Xeon
 - Micro-arquitetura Intel NetBurst, sintonizadas para multimídia

Arquitetura Intel Net Burst

- Tecnologia Hyper Pipeline
- Barramento de sistema de 400MHz
- Cache de execução de rastreo
- mecanismo de execução rápida
- Cache de transferência rápida
- Execução dinâmica avançada
- Unidade melhorada de ponto-flutuante e multimídia
- Streaming SIMD Extensions 2

Processadores de 64-bits

- Intel64
 - Espaço de endereços lineares de 64-bits
 - Intel: Pentium Extreme, Xeon, Celeron D, Pentium D, Core 2, and Core i7
- Modo IA-32
 - Modo de compatibilidade para aplicações antigas de 16 e 32 bits
- Modo IA-64
 - Usa endereços e operandos de 64 bits

Tecnologias Modernas

- Multithreading (Intel = HyperThreading)
 - Duas threads executando simultaneamente em um único processador
- Processamento Multicore
 - Múltiplos núcleos de processamento em um único chip
 - Processadores independentes, mas memória compartilhada (cache, RAM, etc)

Famílias de Processadores Intel Atuais

- Pentium & Celeron: dual core
- Core 2 Duo - 2 núcleos de processamento
- Core 2 Quad - 4 núcleos de processamento
- Core i7 – 4 núcleos de processamento / 8 threads

CISC e RISC

- CISC – complex instruction set
 - Conjunto grande de instruções
 - Operações de alto-nível
 - Requer interpretador de microcódigo
 - **exemplos:**
 - **família Intel IA-32**
- RISC – reduced instruction set
 - Instruções atômicas, simples
 - Conjunto pequeno de instruções
 - Executadas diretamente pelo hardware
 - **exemplos:**
 - **MIPS, ARM (Advanced RISC Machines)**

Gerenciamento de memória IA-32

- Modo de endereçamento real
- Cálculo de endereçamento linear
- Modo protegido
- Modelo Multi-segmento
- Paginação

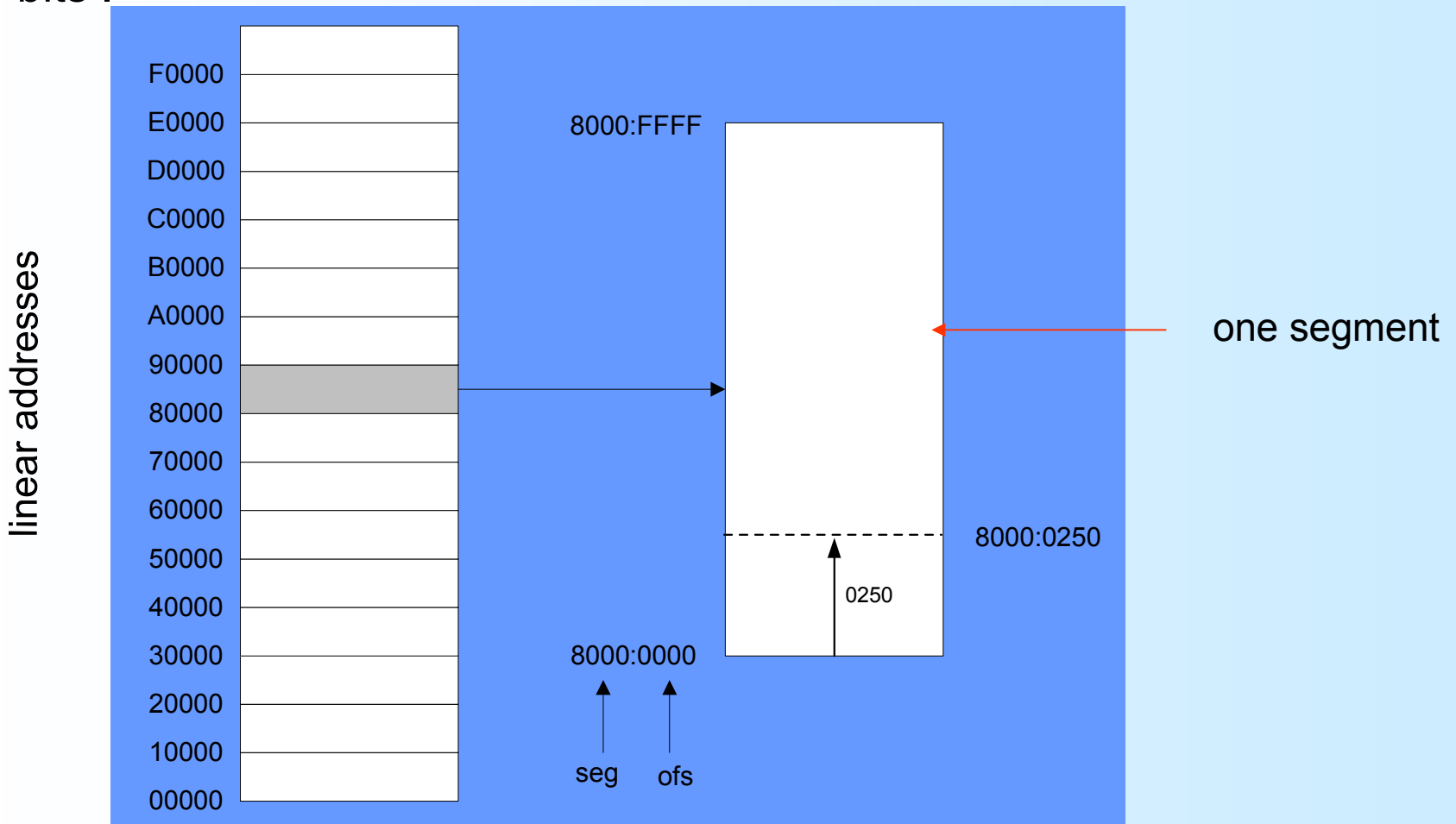
Modo de endereçamento real

- Endereçamento máximo de RAM de 1 MB
- Programas de aplicação podem acessar qualquer área de memória
- Única tarefa
- Suportado pelo sistema operacional MS-DOS

Sistema antigo, mantido apenas p/ compatibilidade de código

Memória Segmentada

Endereçamento segmentado: endereço absoluto (linear) é uma combinação de um valor de segmento de 16-bits adicionado a um offset (deslocamento) de 16-bits .



Cálculo do endereço linear

- Dado um endereço de segmento, multiplicá-lo por 16 (adicionar um zero hexadecimal), e adicionar o offset
- Exemplo: converter 08F1:0100 a um endereço linear

Adjusted Segment value: 0 8 F 1 0

Add the offset: 0 1 0 0

Linear address: 0 9 0 1 0

Sistema antigo, mantido apenas p/ compatibilidade de código

Sua vez . . .

Que endereço linear corresponde o endereço segmento/offset 028F:0030?

$$028F0 + 0030 = 02920$$

Usar notação hexadecimal para endereçamento.

Sistema antigo, mantido apenas p/ compatibilidade de código

Sua vez . . .

Que endereço de segmento corresponde ao endereço linear 28F30h?

Muitos diferentes endereços (segmento-offset) podem produzir o endereço linear 28F30h. Por exemplo:

28F0:0030, 28F3:0000, 28B0:0430, . . .

Sistema antigo, mantido apenas p/ compatibilidade de código

Modo protegido (1 de 2)

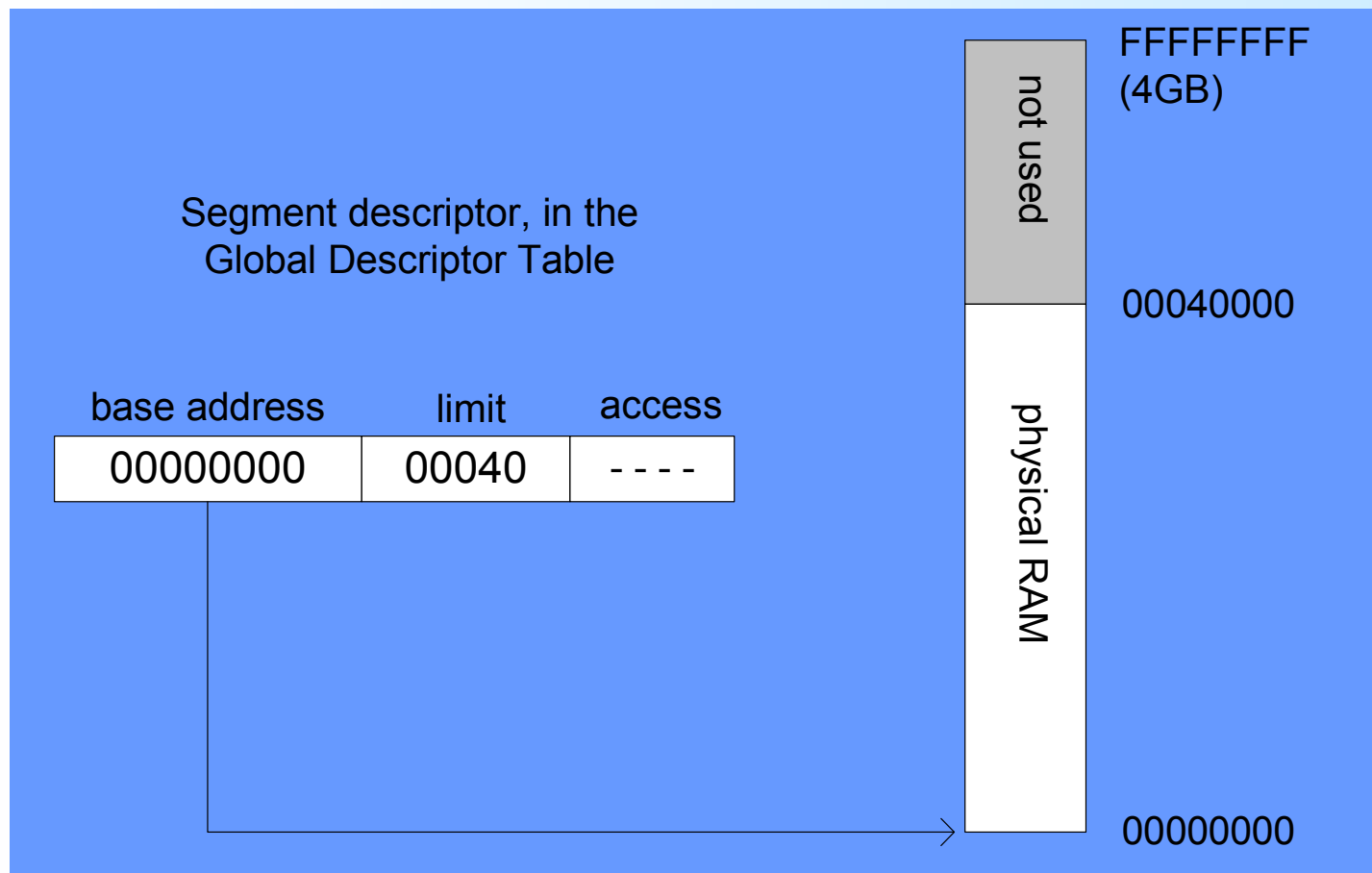
- RAM endereçável até 4 GB
 - (00000000 a FFFFFFFFh)
- Cada programa é associado a uma partição de memória que é protegida de outros programas
- Projetado para multitarefas
- Suportado por Linux & MS-Windows

Modo protegido (2 de 2)

- Tabelas de descrição de segmentos
- Estrutura de programa
 - Áreas de código, dados e pilha
 - Descritores de segmentos CS, DS e SS
 - Tabela de descrição global (GDT)
- Programas MASM usam o modelo de memória *flat*

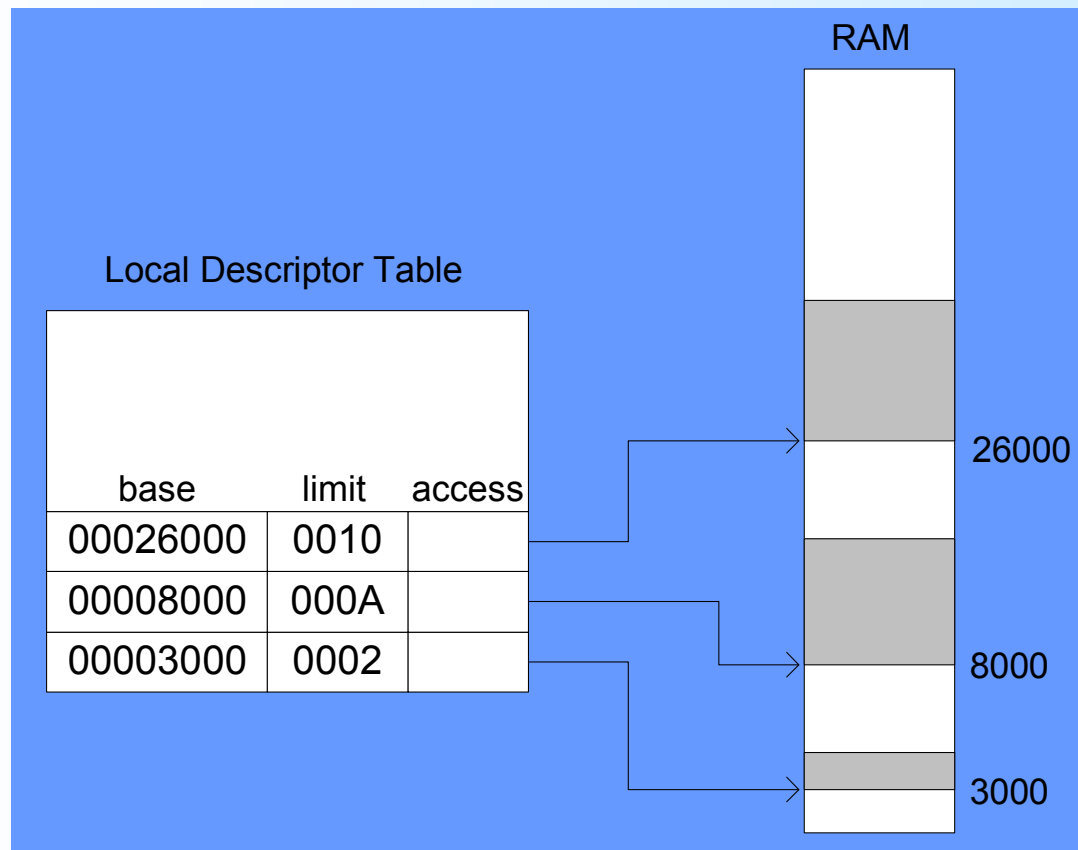
Modelo de segmento Flat

- Uma única tabela de descrição global (GDT).
- Todos os segmentos são mapeados no espaço de 32-bits



Modelo de multi-segmento

- Cada programa tem uma tabela de descrição local (LDT)
 - Contem a descrição para cada segmento usado por um programa



Paginação

- Suportada diretamente pela CPU
- Divide cada segmento em blocos de 4096-bytes chamados páginas
- A soma de todos os programas pode ser maior que a memória física
- Parte do programa em execução fica na memória, e parte no disco
- Gerenciamento de memória virtual (VMM) – utilitário do SO que gerencia a carga e descarga de páginas
- Falta de Página – emitida pela CPU quando uma página deve ser carregada do disco

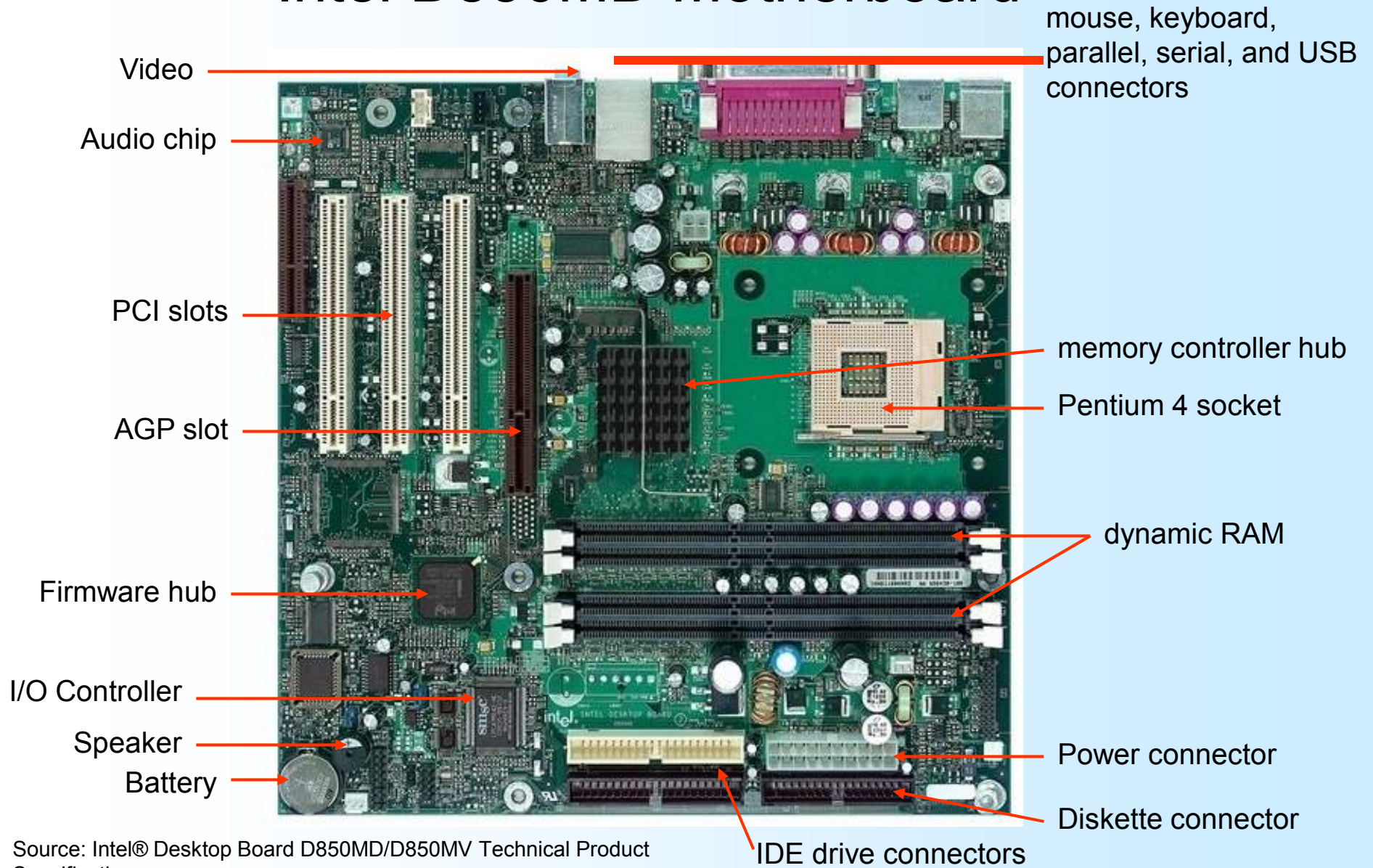
Componentes de um microcomputador IA-32

- Placa-mãe
- Memória
- Saída de Video
- Portos de entrada-saída

Placa-mãe (motherboard)

- Soquete de CPU
- Slots de memória cache externa
- Slots de memória principal
- Chips BIOS
- Chip de sintetização de som (opcional)
- Chip de controle de Vídeo (opcional)
- Conectores IDE, paralelo, serial, USB, vídeo, teclado, joystick, rede e mouse
- Conectores de barramento PCI (expansão)

Intel D850MD Motherboard



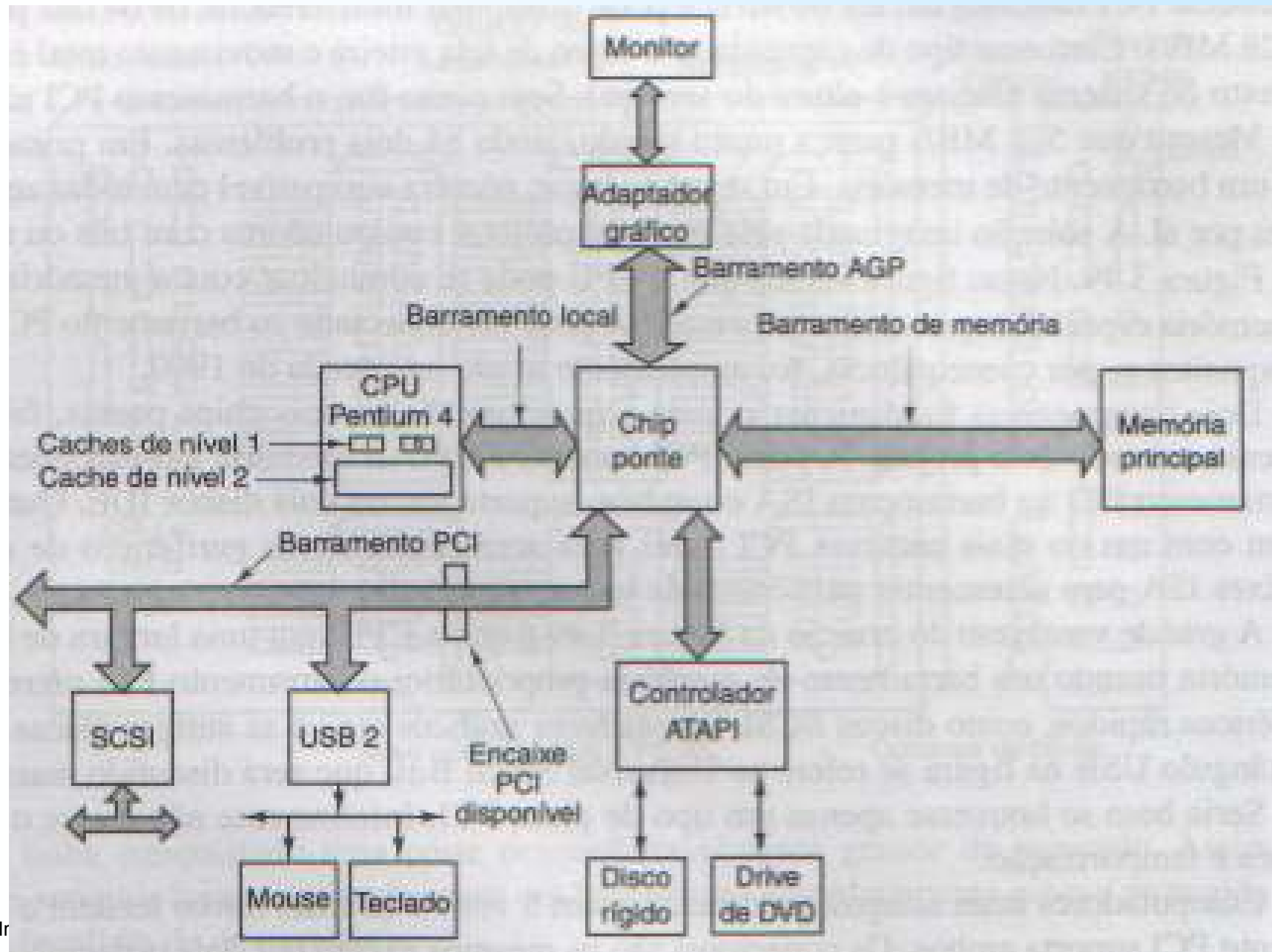
Source: Intel® Desktop Board D850MD/D850MV Technical Product Specification

Irvine, Kip R. Assembly Language for Intel-Based Computers 5/e, 2007.

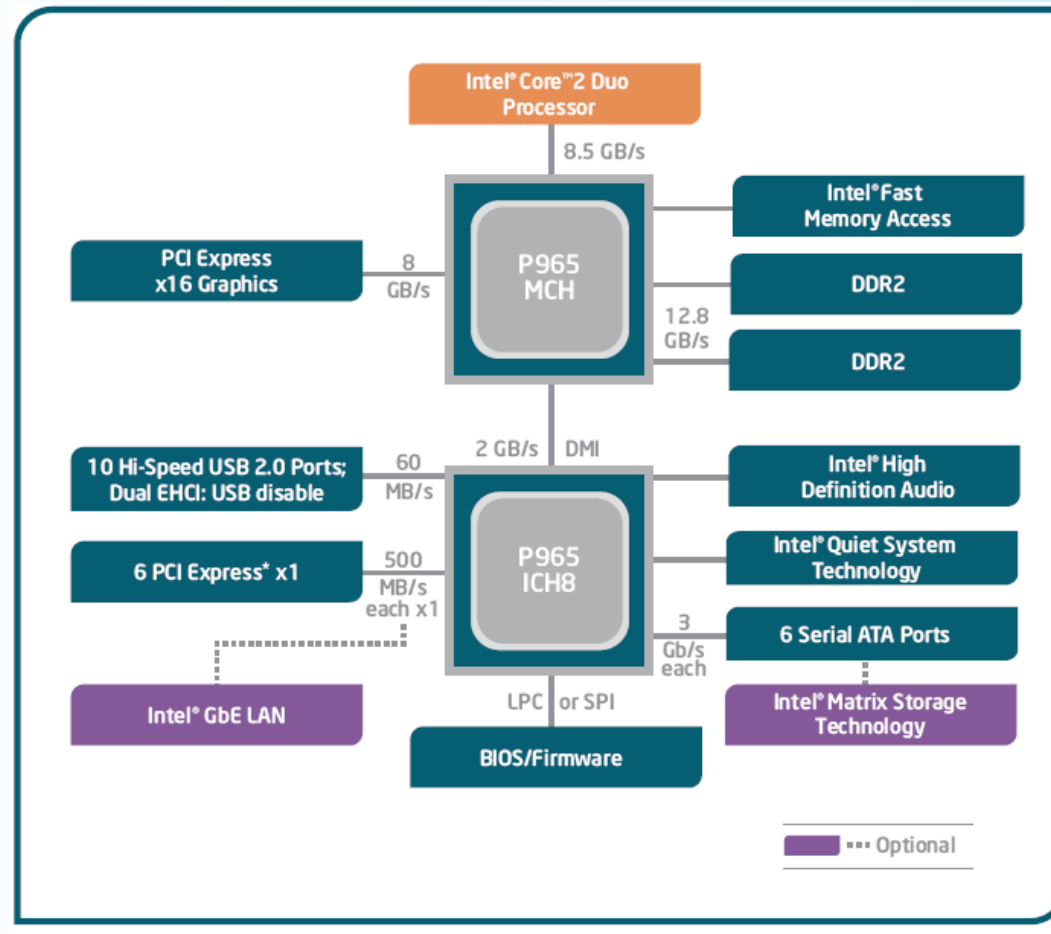
[Web site](#)

[Examples](#)

Estrutura de barramento de um Pentium 4



Intel 965 Express Chipset



Device Interfaces

- ATA host adapters
 - intelligent drive electronics (hard drive, CDROM)
- SATA (Serial ATA)
 - inexpensive, fast, bidirectional
- FireWire
 - high speed (800 MB/sec), many devices at once
- Bluetooth
 - small amounts of data, short distances, low power usage
- Wi-Fi (wireless Ethernet)
 - IEEE 802.11 standard, faster than Bluetooth

Memória

- ROM
 - Somente de leitura (read-only)
- EPROM
 - ROM programável e apagável (erasable programmable)

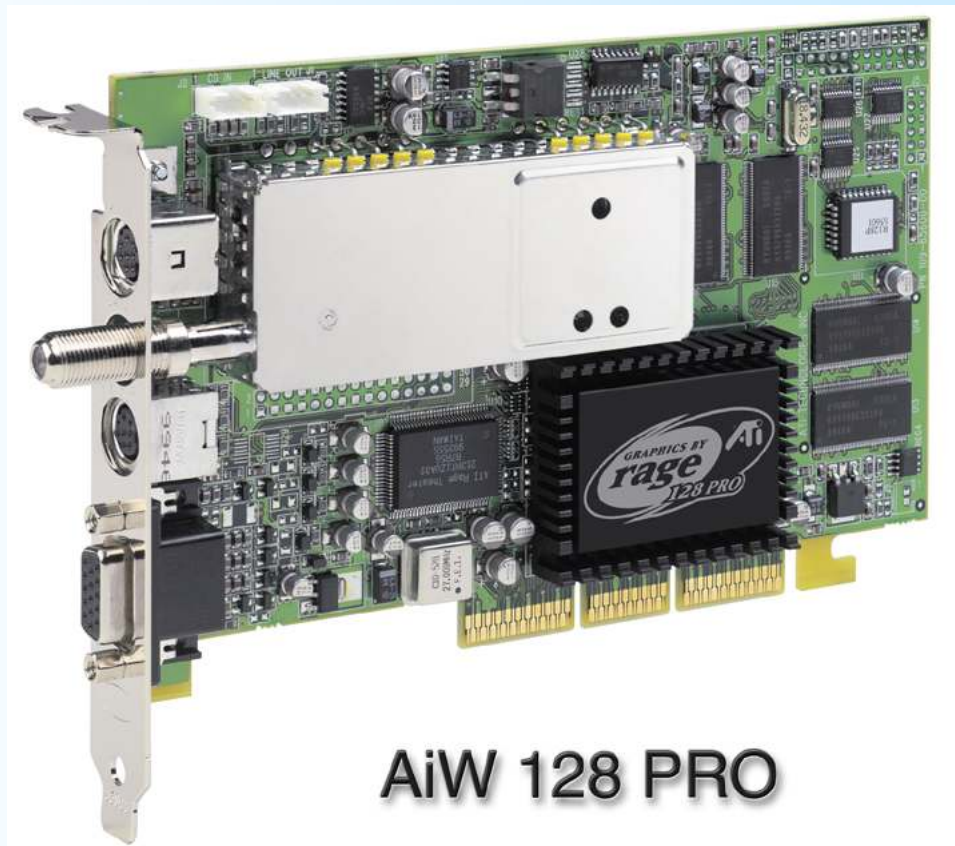


- RAM Dinâmica (DRAM)
 - Custo baixo, mas deve ser reavivada constantemente (refresh)
- RAM Estática (SRAM)
 - Custo alto; usada para memória cache; não necessita reavivamento
- RAM de Vídeo (VRAM)
 - Dois acessos (dual port); otimizada para reavivamento constante do vídeo
- CMOS RAM
 - complementary metal-oxide semiconductor

Saída de Vídeo

- Controlador de vídeo
 - Na placa-mãe ou em cartão de expansão
 - AGP
 - Memória de Vídeo (VRAM)
- Mostrador de Vídeo CRT
 - usa varredura de rastreio
 - Retraço horizontal
 - Retraço vertical
- Monitores LCD
 - Usa a tecnologia de cristais líquidos

Ex: Placa Gráfica / Controladora de Vídeo (ATI Corp.)



AiW 128 PRO

[Web site](#)

[Examples](#)

Portas de entrada-saída

- USB (universal serial bus)
 - Conexão a dispositivos de alta velocidade
 - Até 12 megabits/segundo
 - Hub USB conecta múltiplos dispositivos
 - *enumeração*: para o computador distingüir os dispositivos
 - suporta conexão quente (*hot connections*)
- Paralelo
 - Cabo curto, alta velocidade
 - Comum para impressoras
 - bidirecional, transferência de dados em paralelo
 - Chip controladora Intel 8255

Portas de entrada-saída (cont)

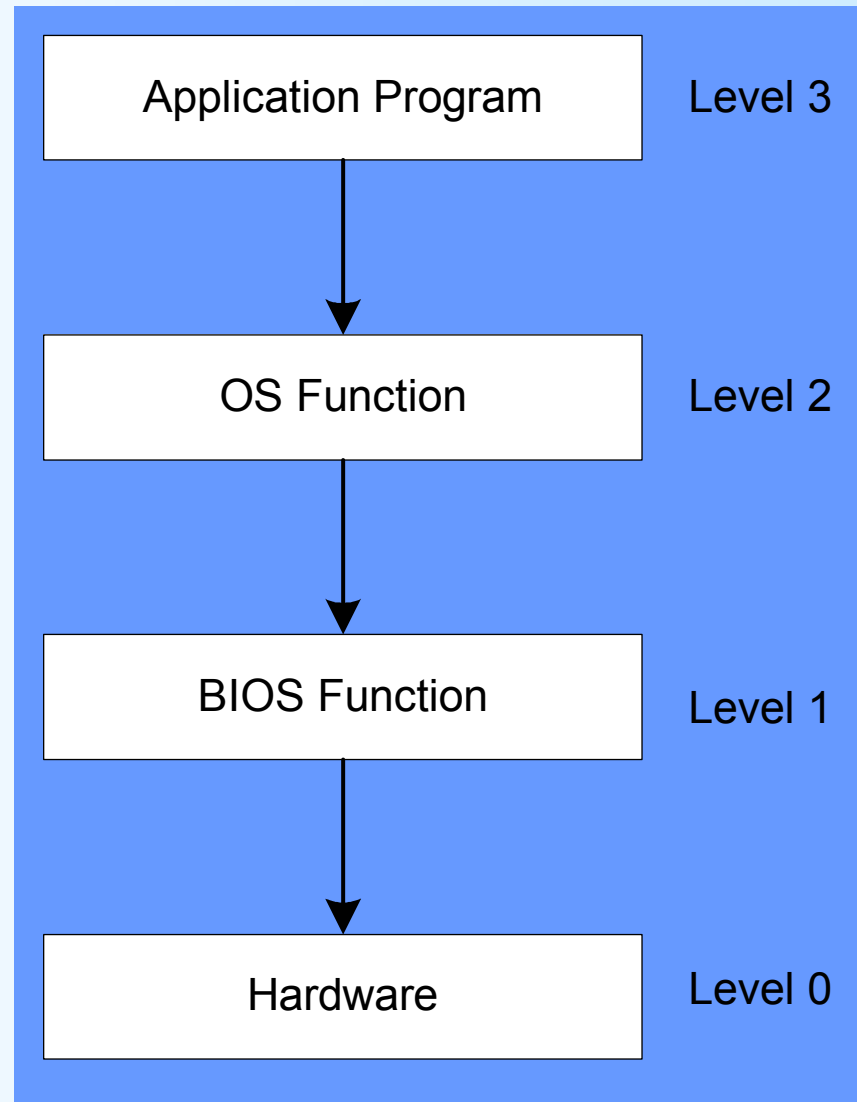
- Serial
 - Porta serial RS-232
 - Um bit por vez
 - Usa cabos longos e modems
 - 16550 UART (universal asynchronous receiver transmitter)

Níveis de entrada-saída

- **Nível 3:** chama uma função da biblioteca (C++, Java)
 - fácil; abstraído do hardware; detalhes ocultos
 - Desempenho lento
- **Nível 2:** Chama uma função do sistema operacional
 - Específico a um SO; independente do dispositivo
 - Desempenho médio
- **Nível 1:** Chama uma função do BIOS (basic input-output system)
 - Pode produzir diferentes resultados em diferentes sistemas
 - Requer conhecimento de hardware
 - Usualmente tem bom desempenho
- **Nível 0:** Comunica diretamente com o hardware
 - Pode não ser permitido por certos sistemas operacionais

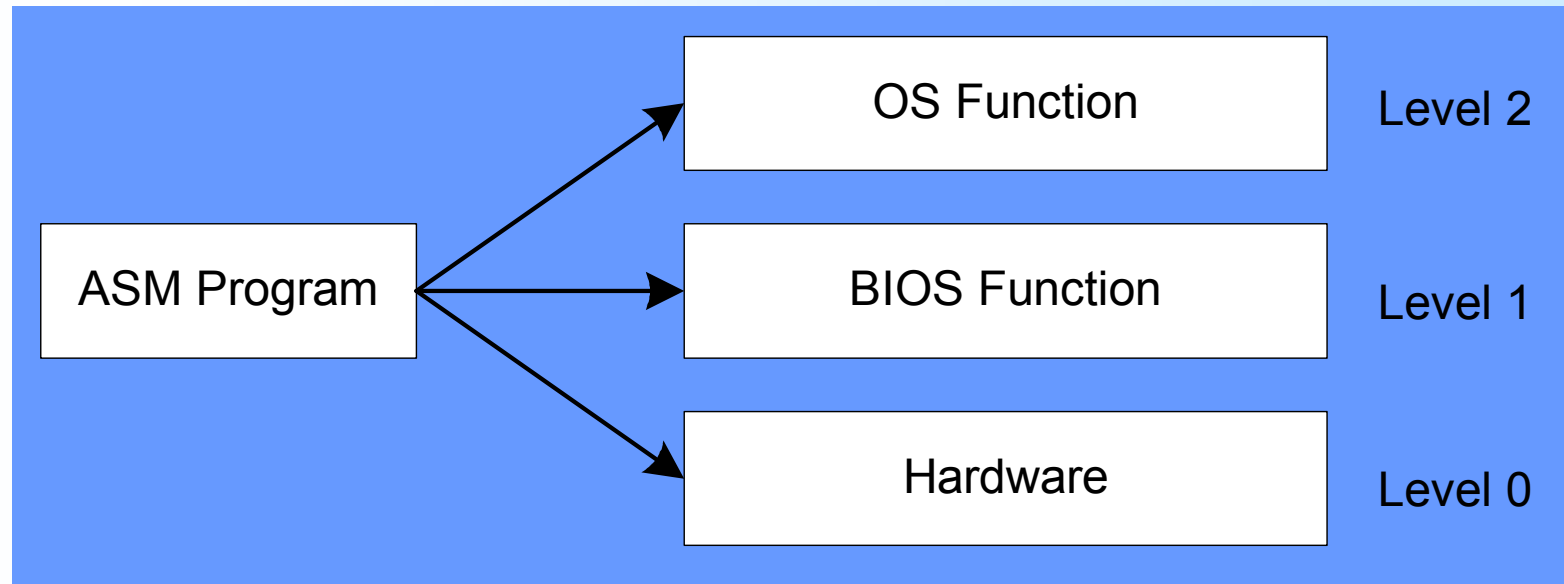
Mostrando uma cadeia de caracteres

Quando um programa escrito em linguagem de alto nível (ex: C) mostra uma cadeia de caracteres, ocorrem os seguintes passos:



Nível de programação ASM

Os programas em ASM podem realizar entrada-saída em um dos seguintes níveis:



Ex: Programação em Assembly IA-32

- **Objetivo:** acender uma lâmpada de acordo c/ temperatura
 - 1) **lâmpada verde:** temperatura < 30 graus
 - 2) **lâmpada amarela:** 30 < temperatura < 40 graus
 - 3) **lâmpada vermelha:** temperatura > 40 graus

Obs:

endereço do sensor de temperatura: 0FF0H;

endereço da saída para as lâmpadas: 0FF1H;

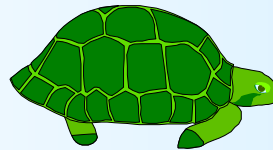
código para selecionar a lâmpadas: **verde** = 05H

amarela = 10H

vermelha = 15H

Exemplo de programa ASM

Rótulo (label)	mnemônico	operandos	comentário
	mov	dx, 0FF0h	;endereço da porta de entrada do sensor
	in	al, dx	; ler valor do sensor (temperatura)
	mov	dx,0FF1h	; endereço da porta de saída p/ lâmpadas
	cmp	al,1Eh ;30 graus	
	jb	verde	
	cmp	al, 28h ;40 graus	
	jbe	amarela	
	mov	al, 15h	
	out	dx,al	;acende vermelha
	jmp	retorna	
amarela:	mov	al, 10h	
	out	dx, al	;acende amarela
	jmp	retorna	
verde:	mov	al, 05h	;acende verde
	out	dx,al	
retorna:	ret		



42 69 6E 61 72 79