



# IA32

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica



# Introdução

- 80386DX, 80386SX
- 80486DX, 80486SX, 80486DX2, 80486DX4
- Intel Pentium, Pentium-MMX
- Intel Pentium Pro, Pentium II, Celeron
- Intel Pentium III, Celeron
- Intel Pentium 4, Pentium 4 D, Celeron
- Core Duo, Centrino
- Intel Core 2 Duo, Core 2 Quad, Centrino
  - EM64T, x86\_64
  - EM64T  $\neq$  IA64

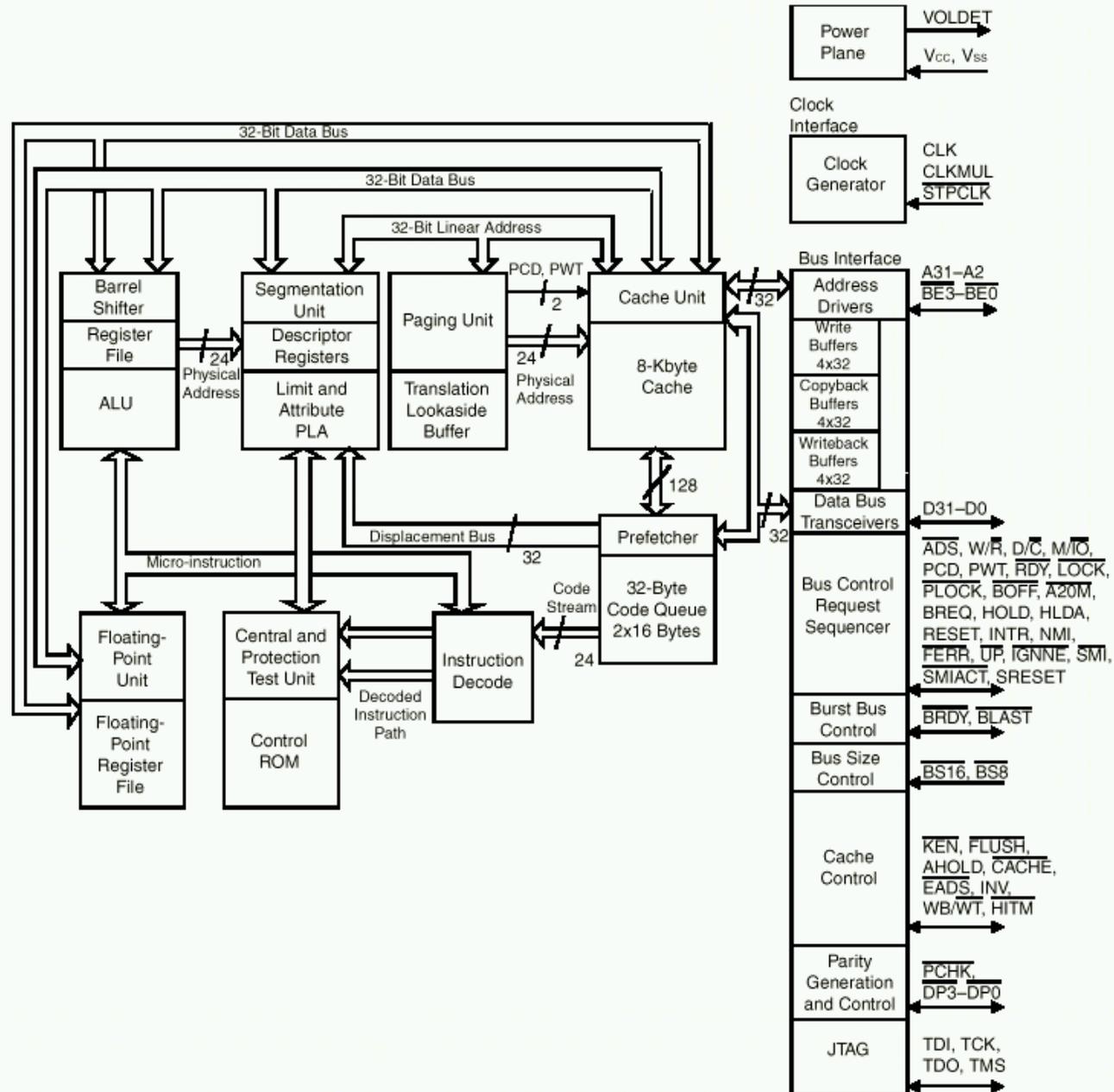


# Introdução

- AMD 5x86, K5, 6x86
- AMD K6, K6II, K6III
- AMD Athlon, Duron
- Athlon XP, Sempron, Turion
- AMD Athlon 64, Turion 64
  - AMD64, x86\_64
  - AMD64  $\approx$  EM64T
- AMD Athlon 64 X2, Turion 64 X2
  - AMD64, x86\_64
- Transmeta Crusoe
- Cyrix 5x86, M1

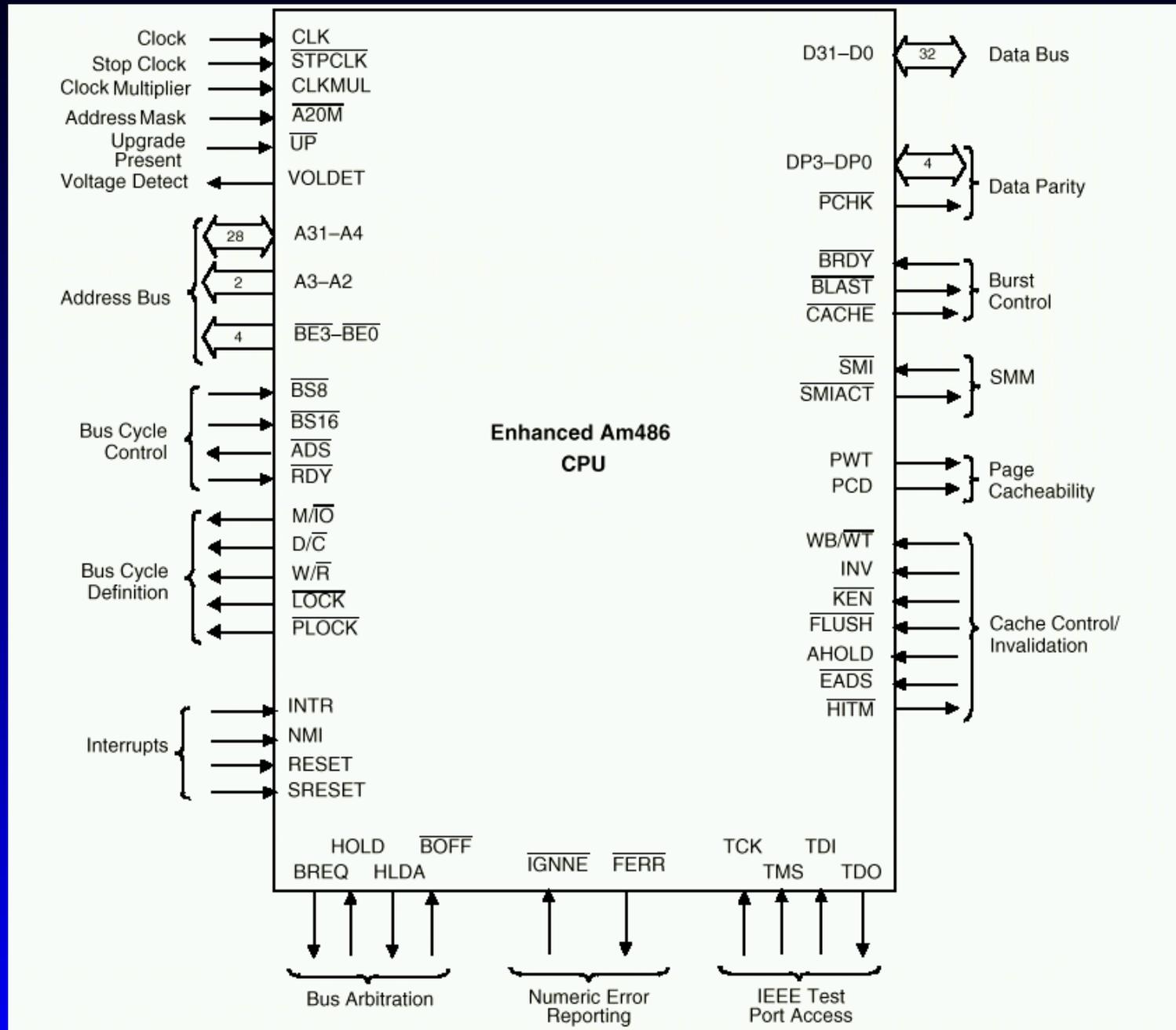


# Diagrama de Blocos DX4

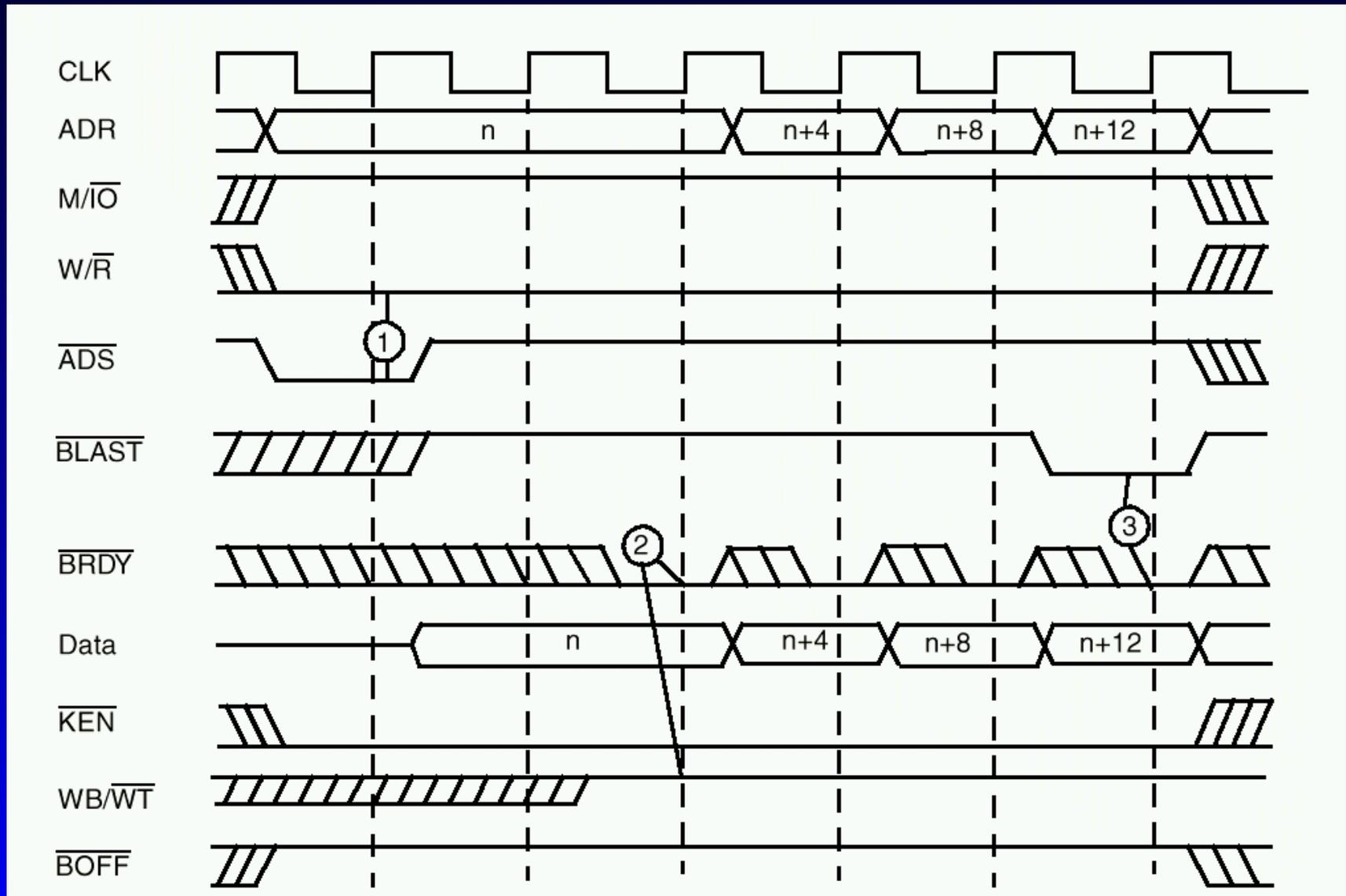




# Sinais DX4



# Ciclo de Leitura Externa DX4

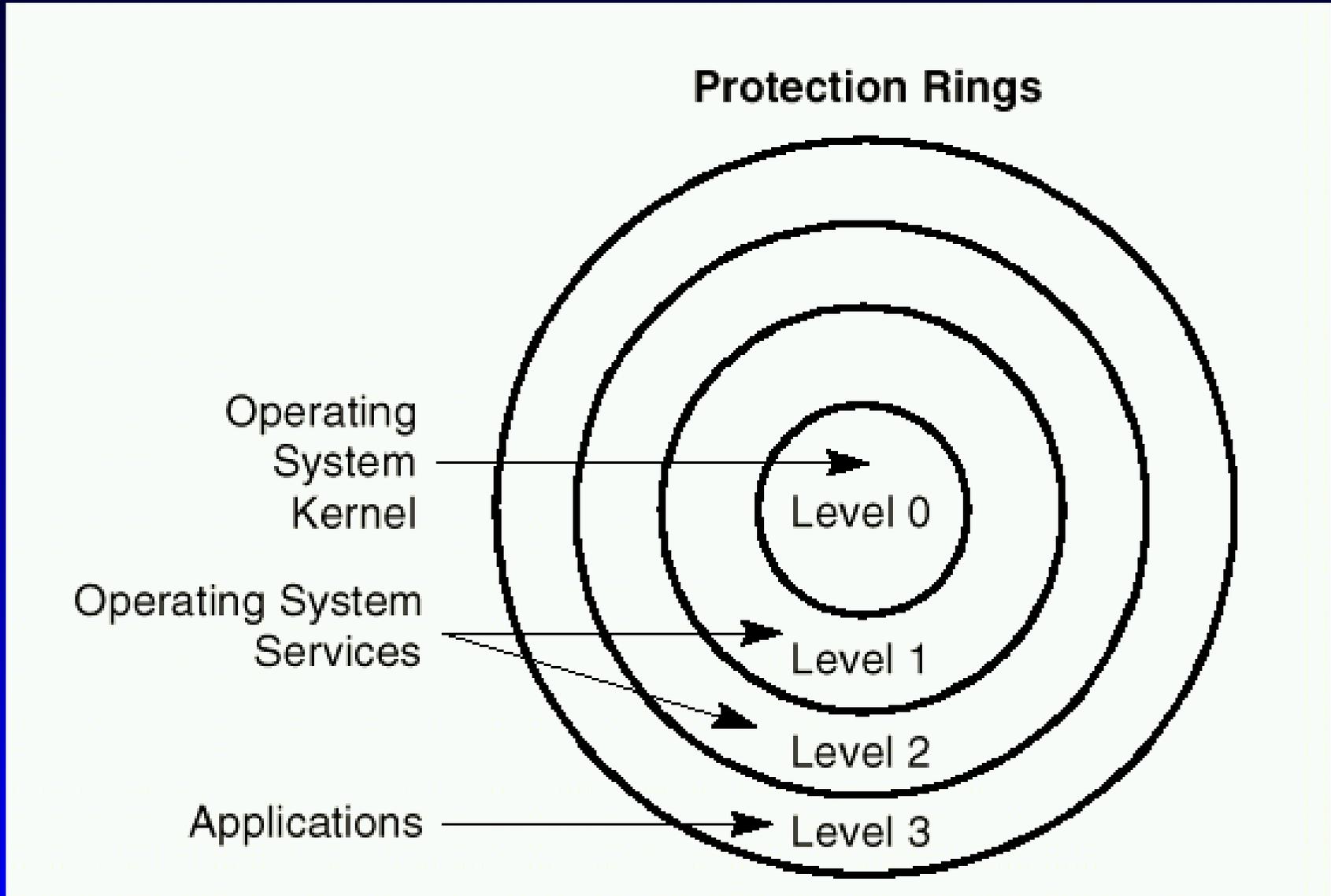




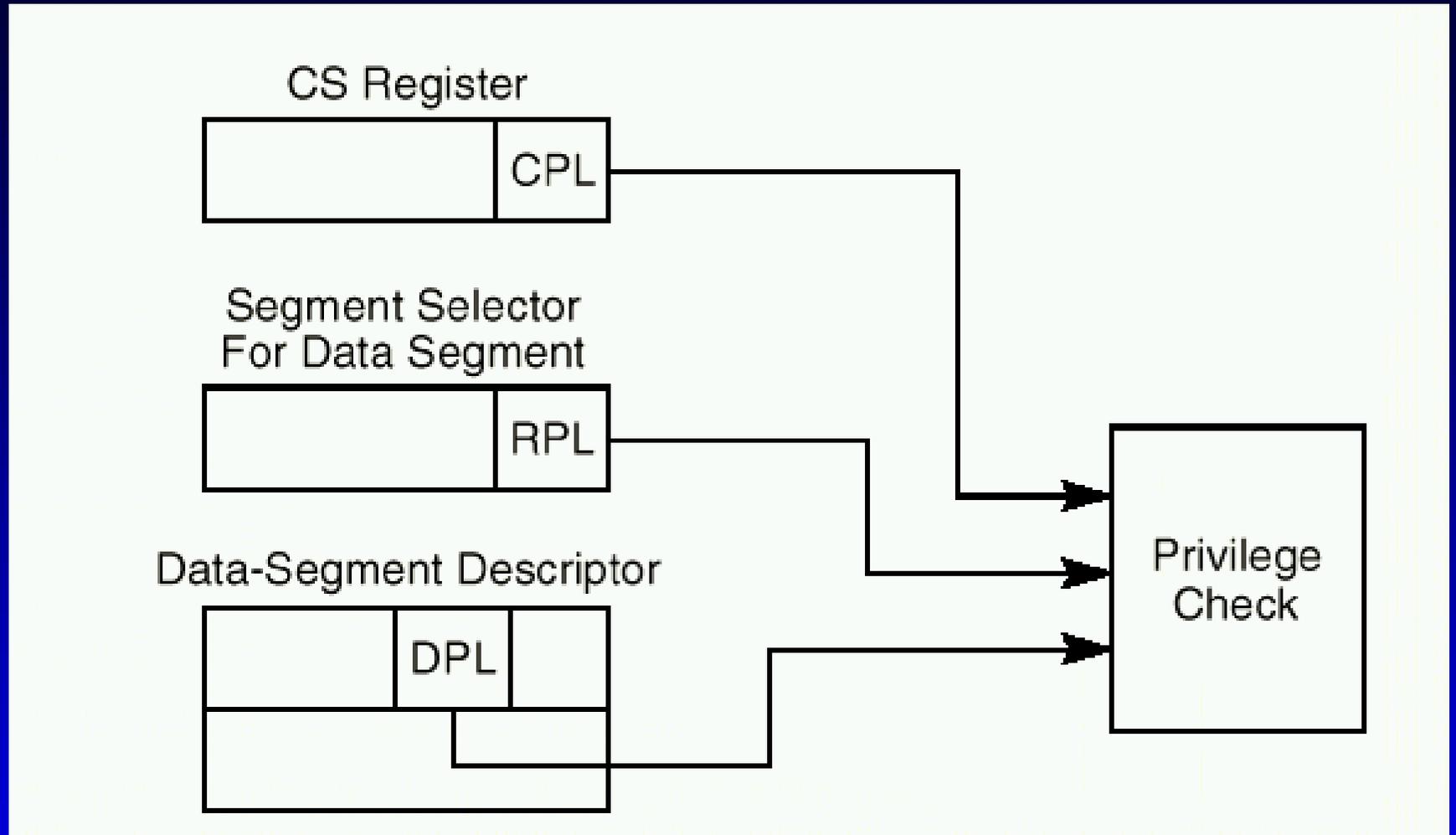
# Modos de Operação

- Modo protegido
  - Modo virtual 86
    - Emulação do modo real por uma tarefa de modo protegido
- Modo real
  - Modo original do 8086
  - Modo default
- Modo de gerenciamento de sistema
  - Introduzido com o 80386SL
  - Utilizado para função de economia de energia
  - Endereçamento similar ao modo real

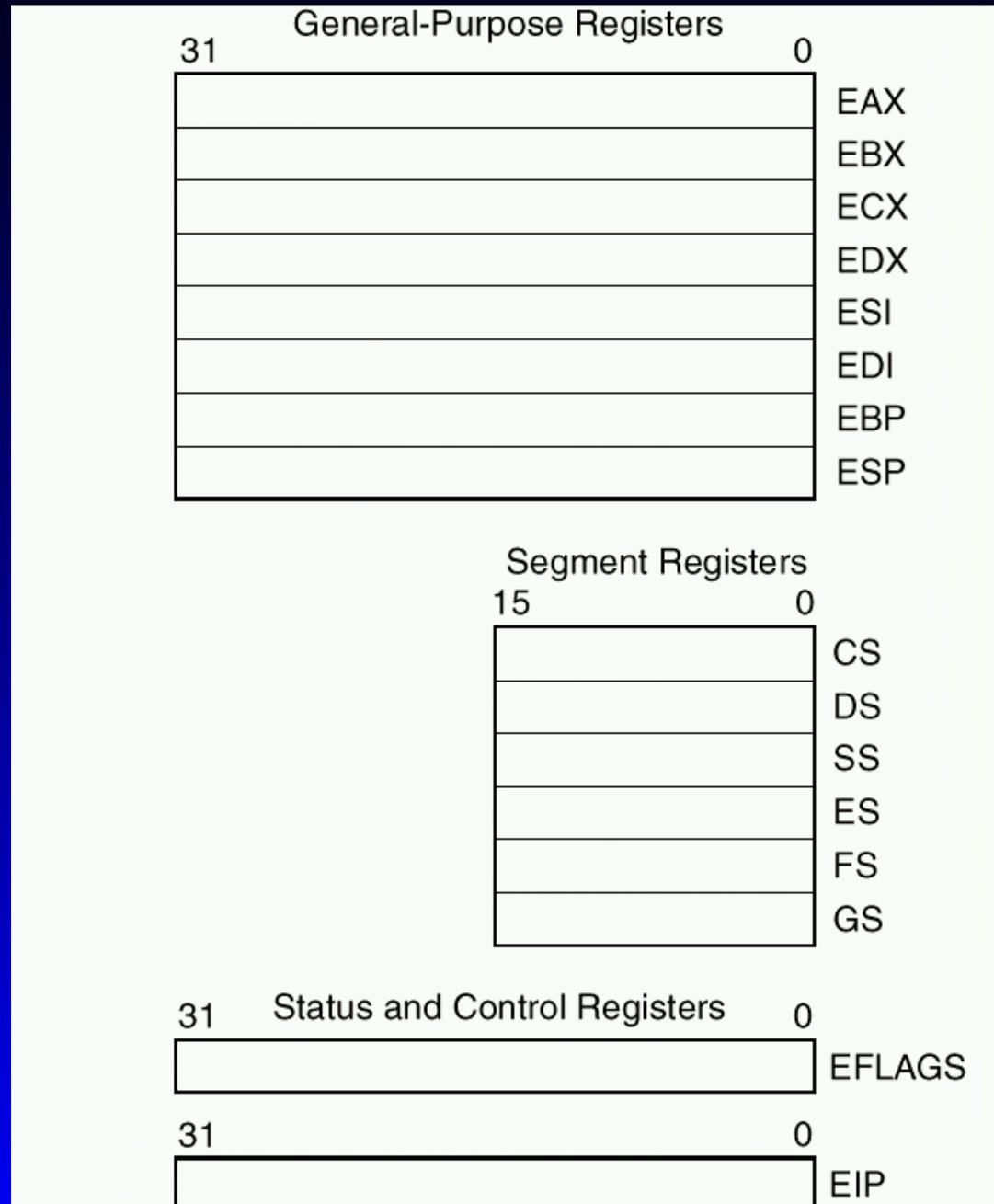
# Níveis de Privilégio



# Verificação de Privilégios



# Ambiente de Execução Básico



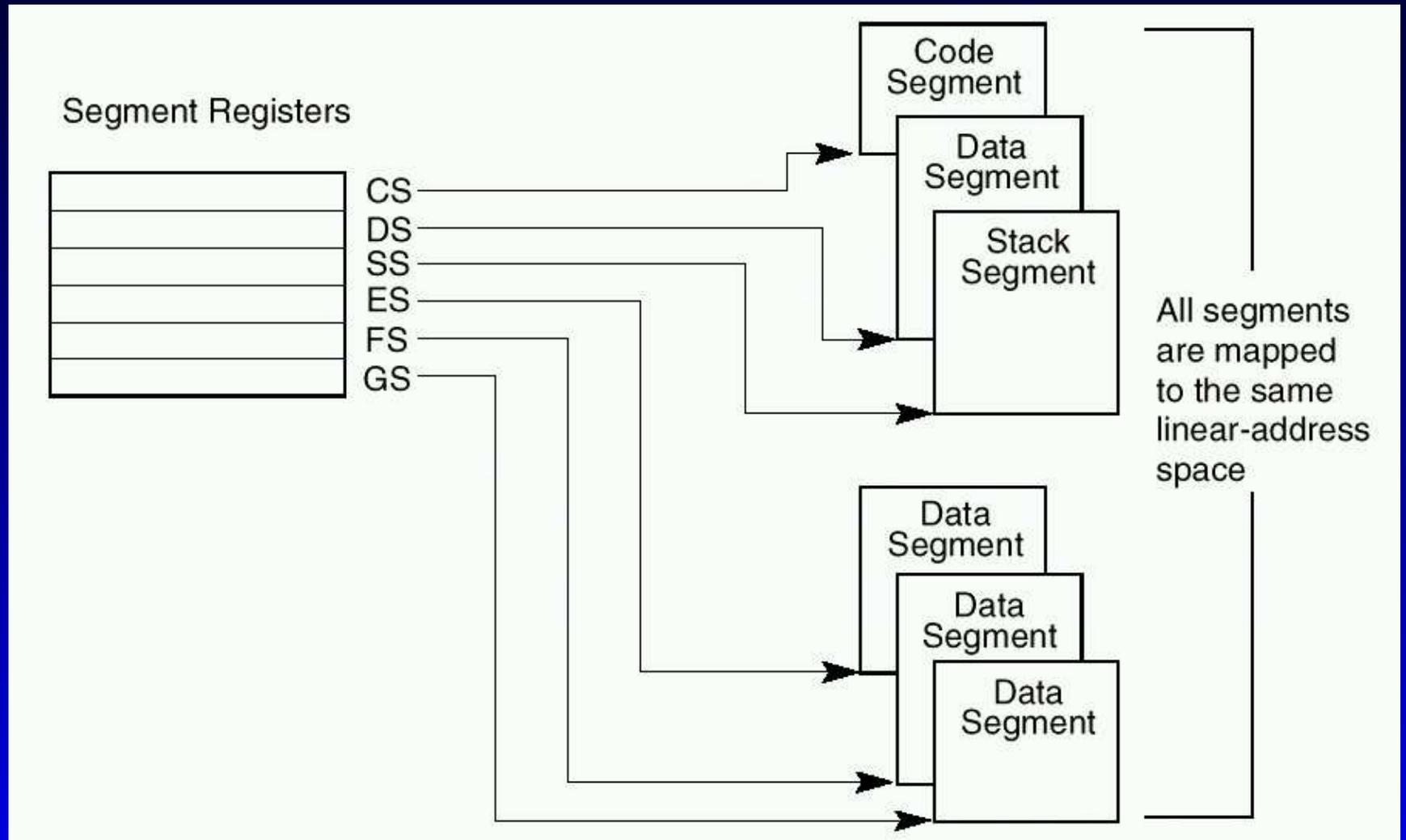
# Registradores de Uso Geral



## General-Purpose Registers

31	16	15	8	7	0	16-bit	32-bit
		AH			AL	AX	EAX
		BH			BL	BX	EBX
		CH			CL	CX	ECX
		DH			DL	DX	EDX
		BP					EBP
		SI					ESI
		DI					EDI
		SP					ESP

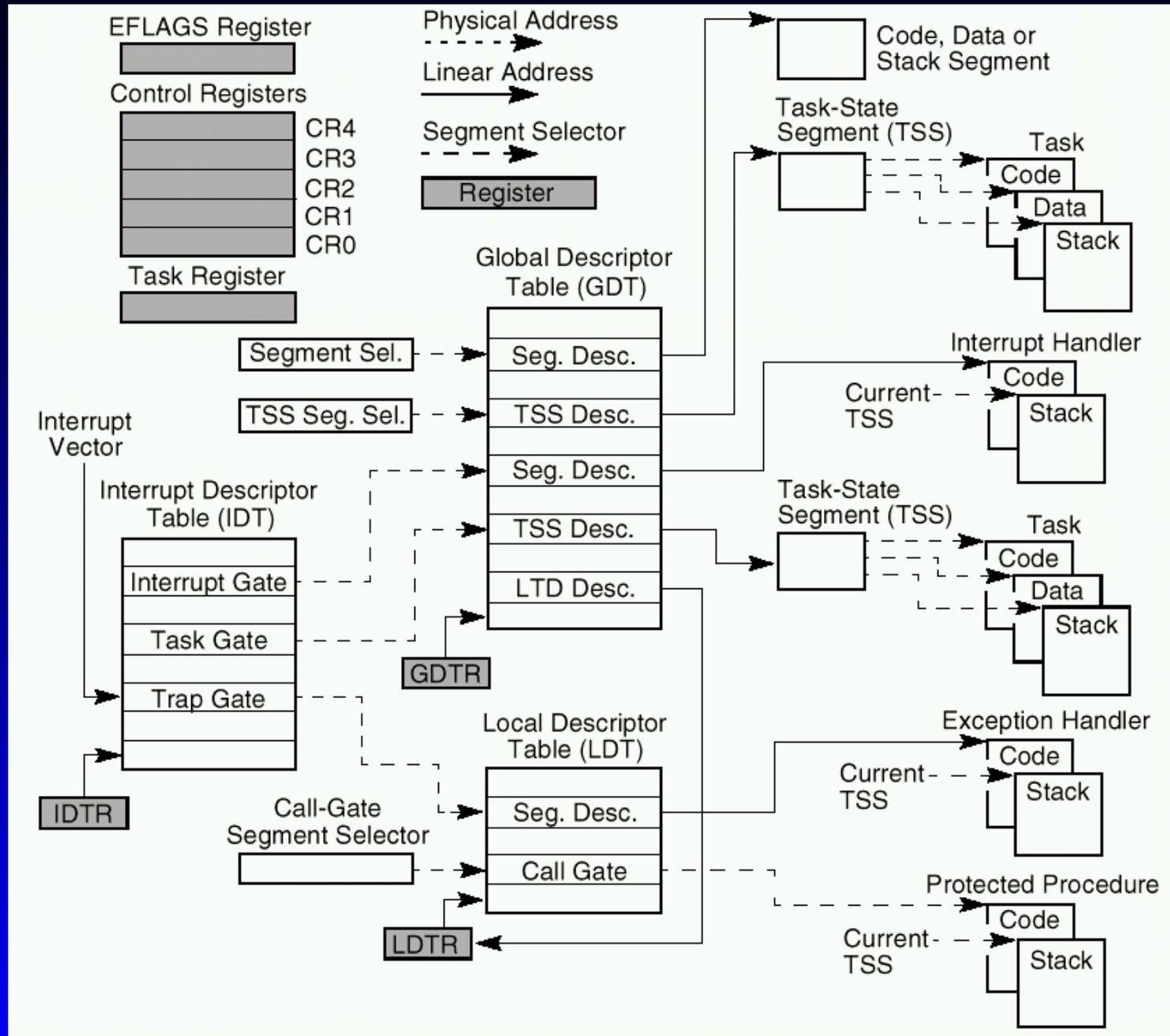
# Registradores de Segmento





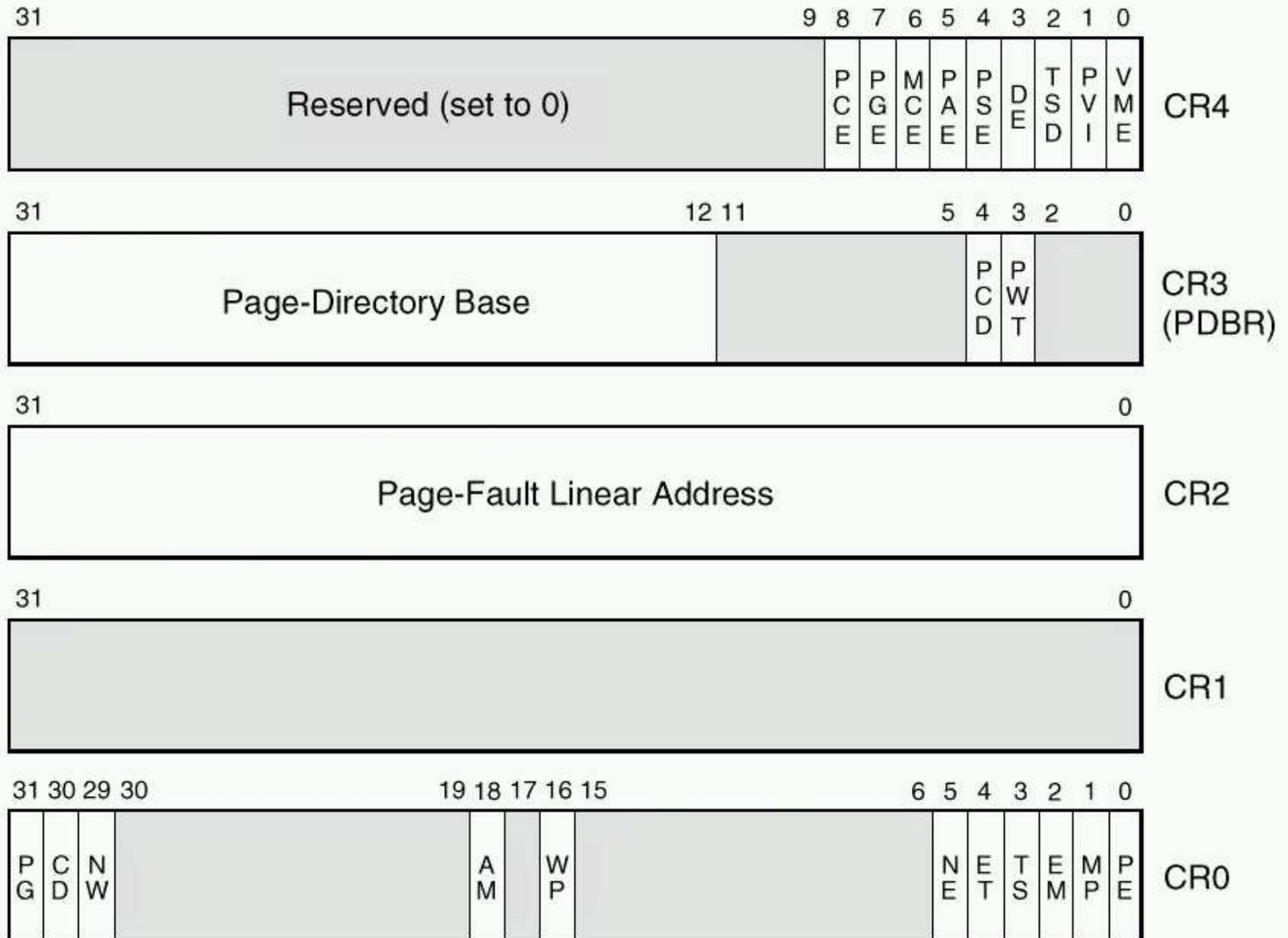


# Ambiente de Sistema





# Registradores de Controle





# Registradores de Controle

**PG:** Paging

**CD:** Cache disable

**NW:** Not write-through

**AM:** Alignment mask

**WP:** Write protect

**NE:** Numeric error

**ET:** Extension type

**TS:** Task switched

**EM:** Emulation

**MP:** Monitor coprocessor

**PE:** Protection enable



# Registradores de Controle

**PCD:** Page-level cache disable

**PWT:** Page-level write-through

**VME:** Virtual-8086 mode extensions

**PVI:** Protected-mode virtual interrupts

**TSD:** Time stamp disable

**DE:** Debugging extensions

**PSE:** Page size extensions

**PAE:** Physical address extension

**MCE:** Machine-check enable

**PGE:** Page global enable

**PCE:** Performance-monitoring counter enable



# Relocação

- Em sistemas multitarefa, cada tarefa terá um endereço inicial diferente. Como o programador/compilador/linker saberá o endereço inicial (e endereços de subrotinas, dados, etc...) da tarefa?
  - Problema da relocação
  - Gerar apenas código com endereços relativos
    - Código relocável
  - Relocação durante a carga da tarefa na memória

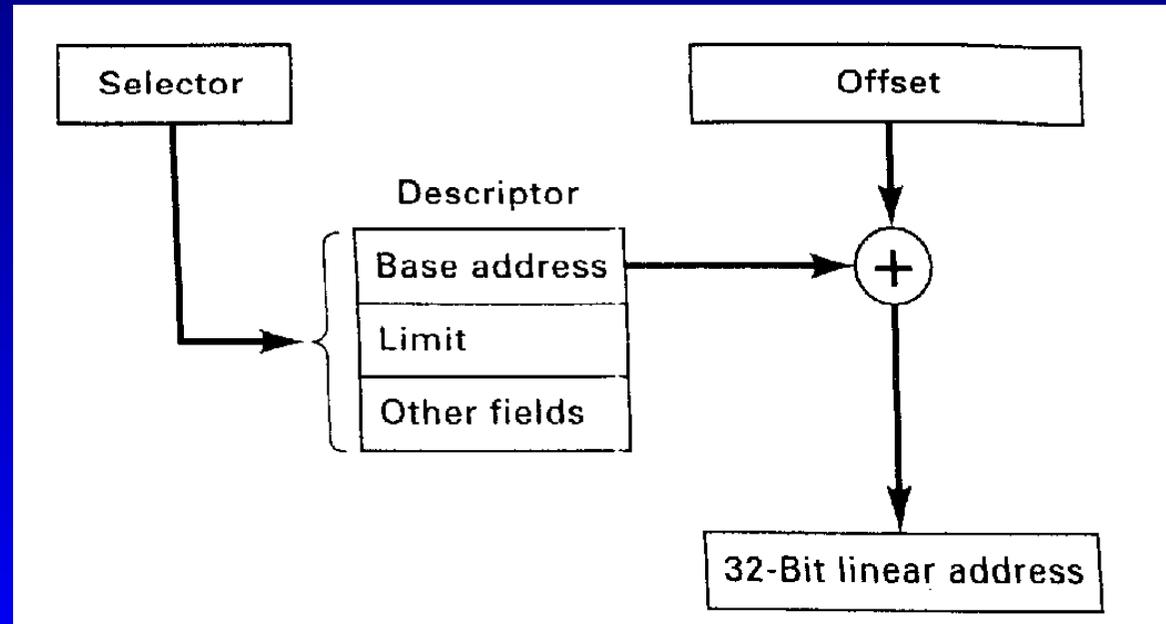


# Segmentação

- Associa-se a cada área de memória um valor de base e um valor de limite
- base=endereço inicial
- limite=endereço máximo ou tamanho
- Todas as tarefas são codificadas assumindo que a sua área de memória começa em 0
- Quando a tarefa é selecionada para executar, base e limite são carregados pelo sistema operacional em registradores especiais na CPU

# Segmentação

- Endereço físico = offset + base
  - Soma feita por hardware a cada acesso à memória
- Endereços são verificados quanto ao limite
  - Uma tarefa não pode acessar a área das outras





# Exemplo

- 64kB de memória total
- Programa 1
  - Tamanho de 16kB
  - Carregado no endereço físico 1000h
- Programa 2
  - Tamanho de 32kB
  - Carregado no endereço físico 6000h
- Ambos acessam o endereço lógico 3000h
- Tabela de segmentos no endereço físico 0400h

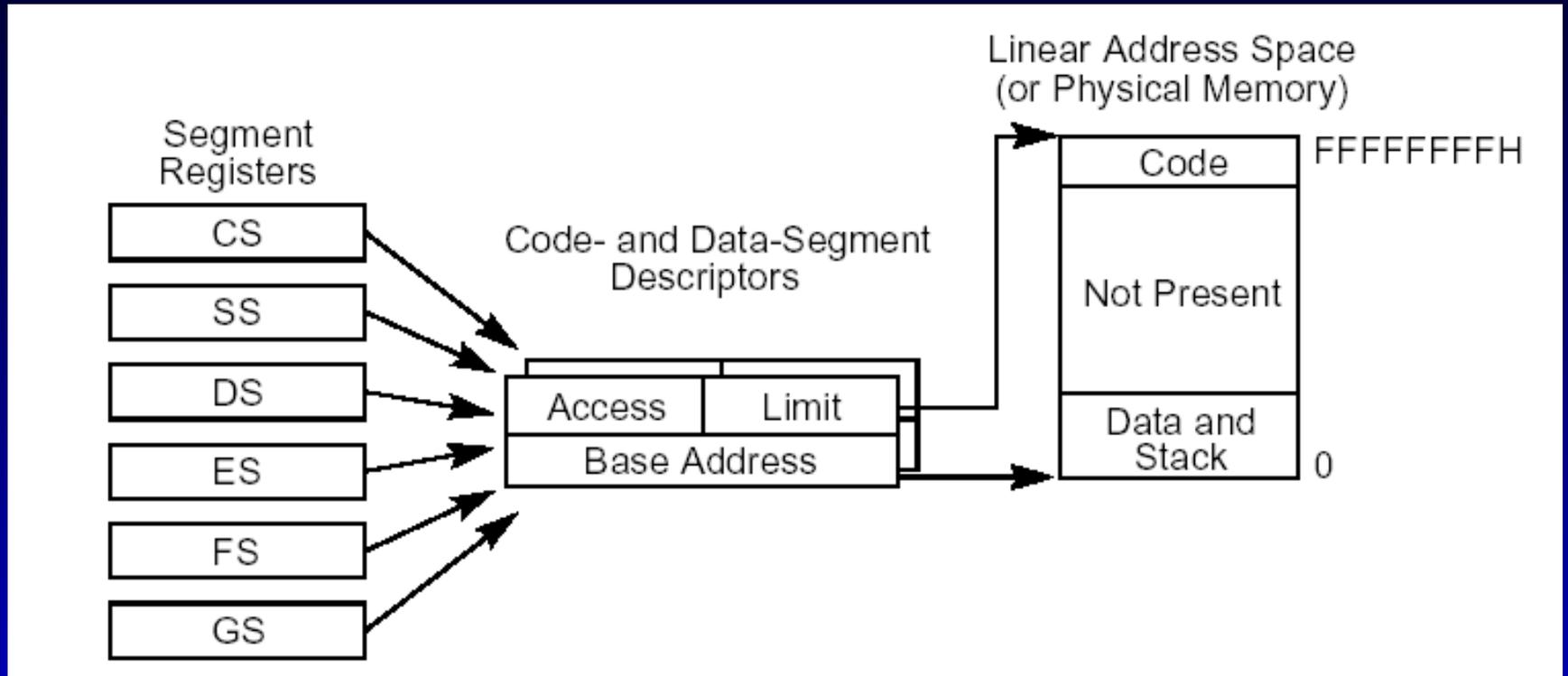


# Segmentação

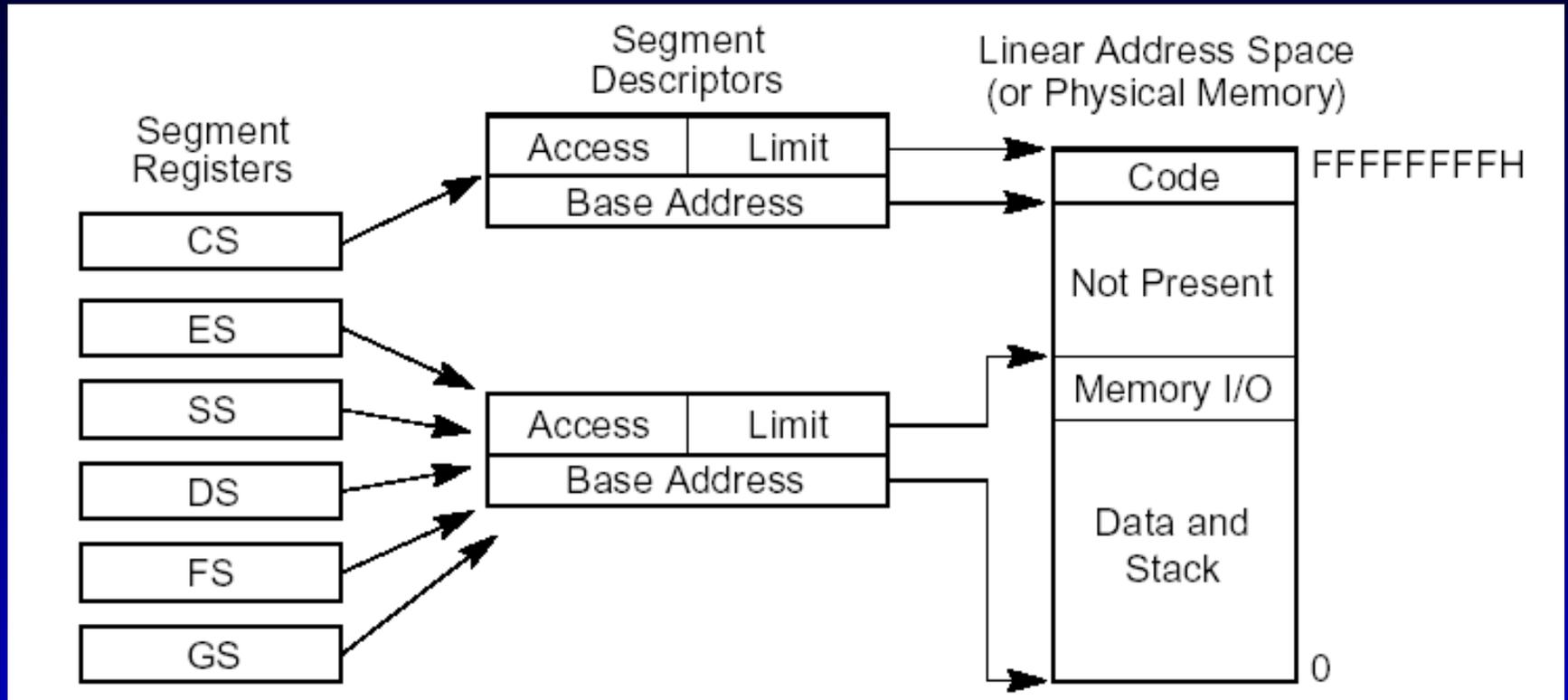
- O seletor e descritor de segmento são controlados pelo sistema operacional
- O offset é controlado pelo programa de aplicação
- Proteção
  - Usualmente o descritor de segmento tem alguns bits associados que são utilizados para determinar as características de proteção do segmento
  - Permissões de leitura, escrita, execução
- Cache de descritor de segmento
  - Evita dois acessos a memória
  - Alterado apenas quando o registrador de seletor de segmento é alterado



# Modelo Flat

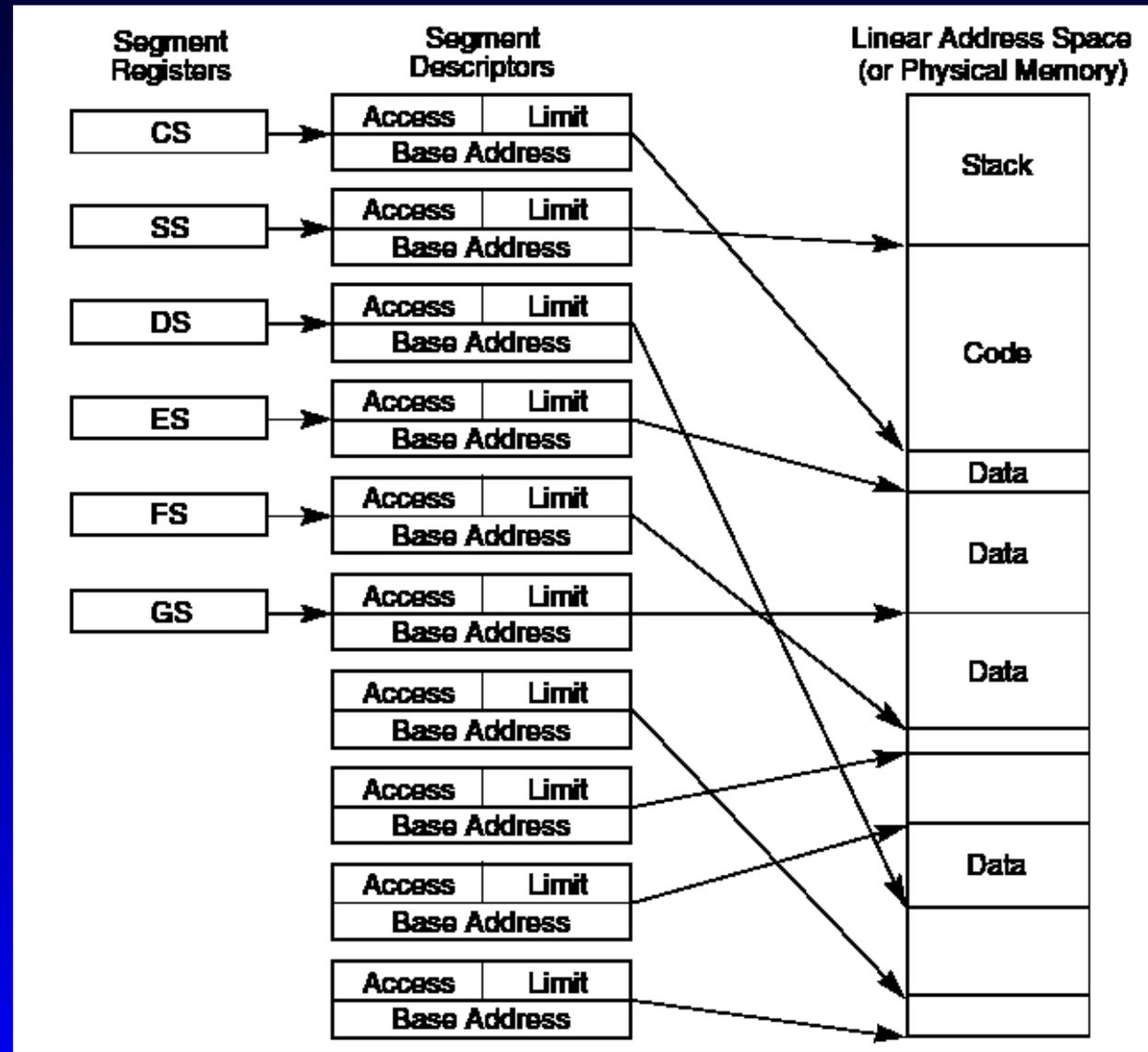


# Modelo Flat Protegido





# Multi-segmentos





# Segmentação na IA32

- Local Descriptor Table (LDT)
- Global Descriptor Table (GDT)

## System Table Registers

	47	16 15	0
GDTR	32-bit Linear Base Address		16-Bit Table Limit
IDTR	32-bit Linear Base Address		16-Bit Table Limit

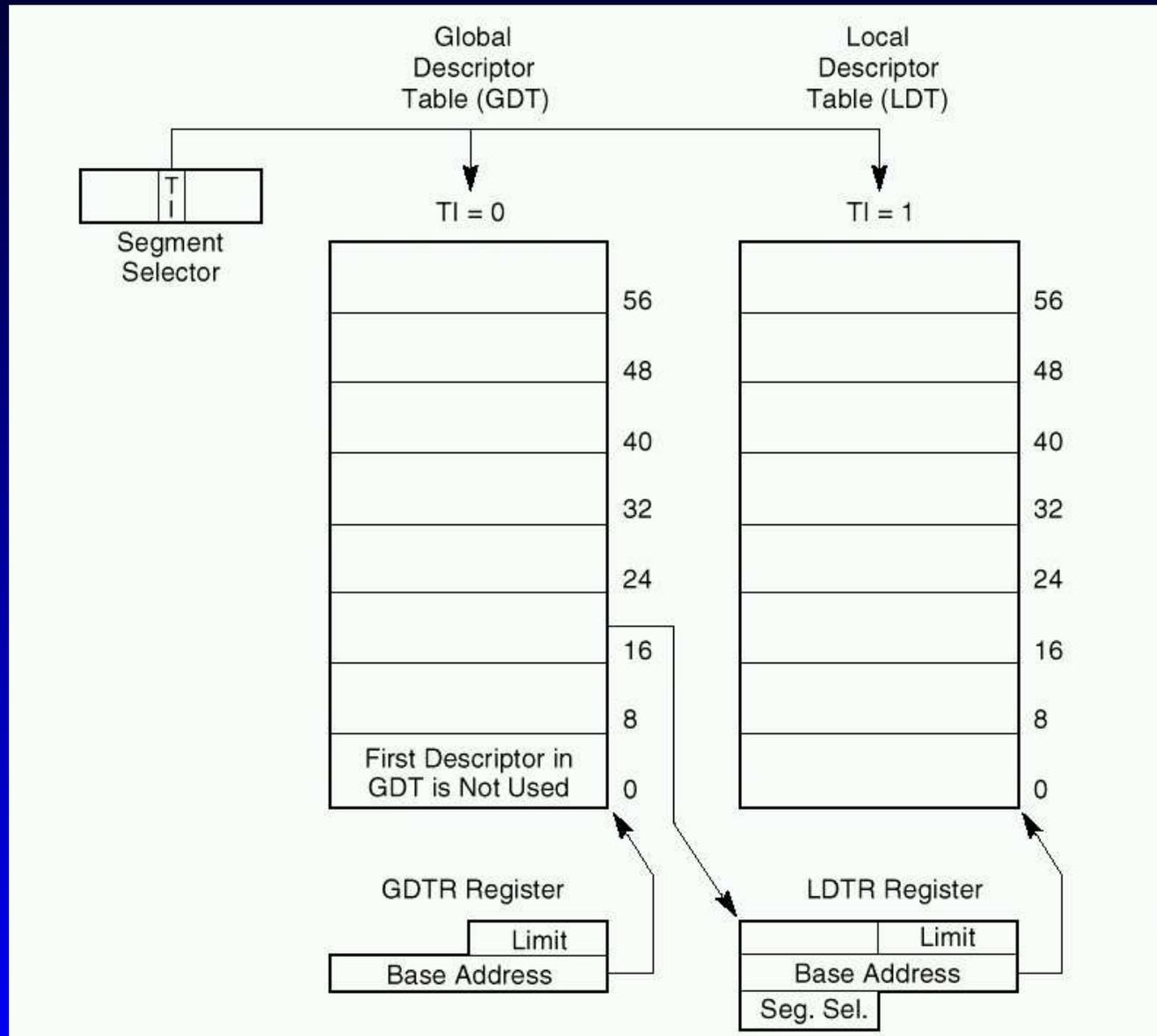
## System Segment 15 Registers 0

## Segment Descriptor Registers (Automatically Loaded)

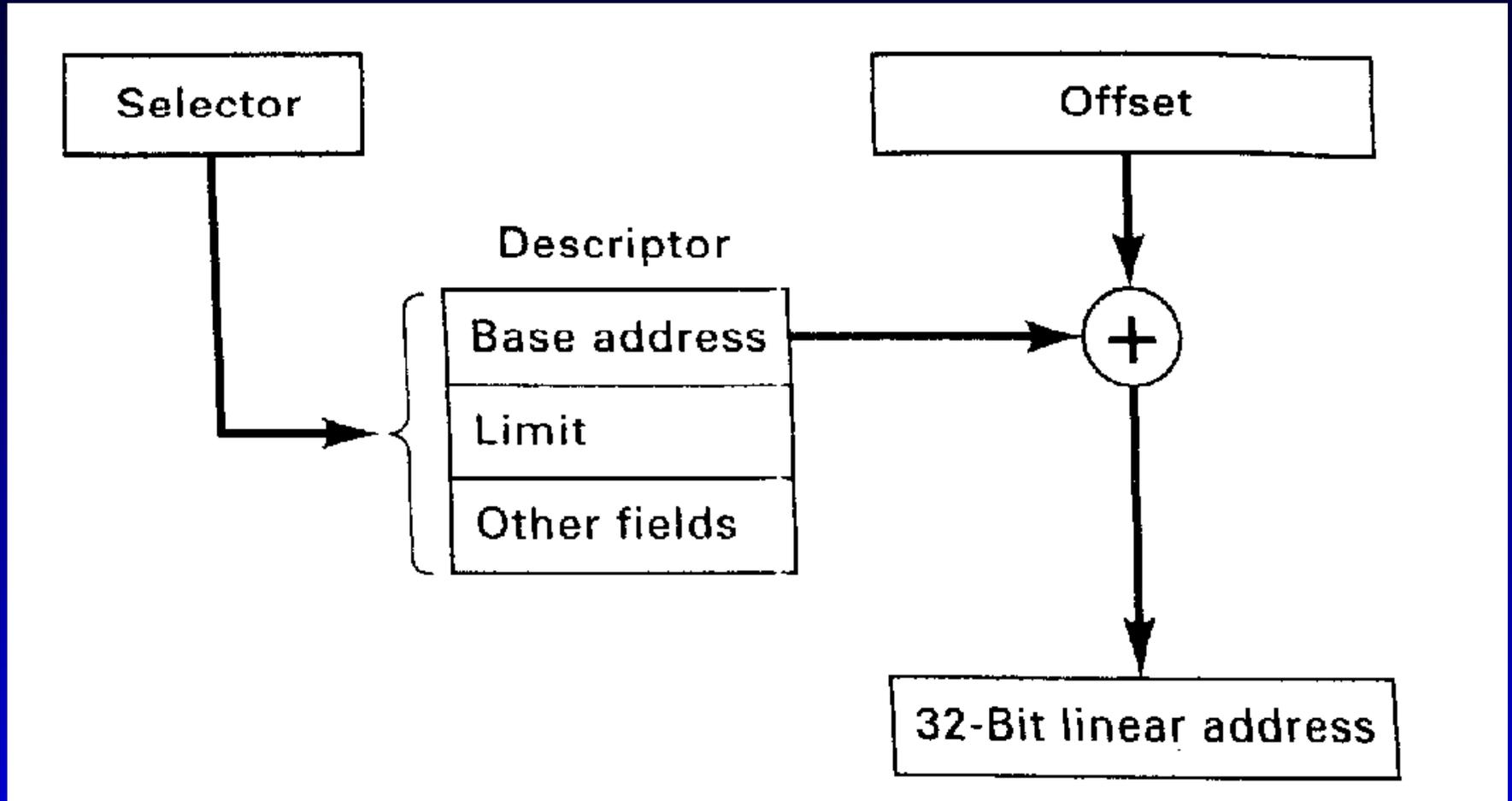
Task Register	Seg. Sel.	32-bit Linear Base Address	Segment Limit	Attributes	
LDTR	Seg. Sel.	32-bit Linear Base Address	Segment Limit		



# GDT e LDT



# Segmentação na IA32



# Seletor de Segmento na IA32



Table Indicator

0 = GDT

1 = LDT

Requested Privilege Level (RPL)



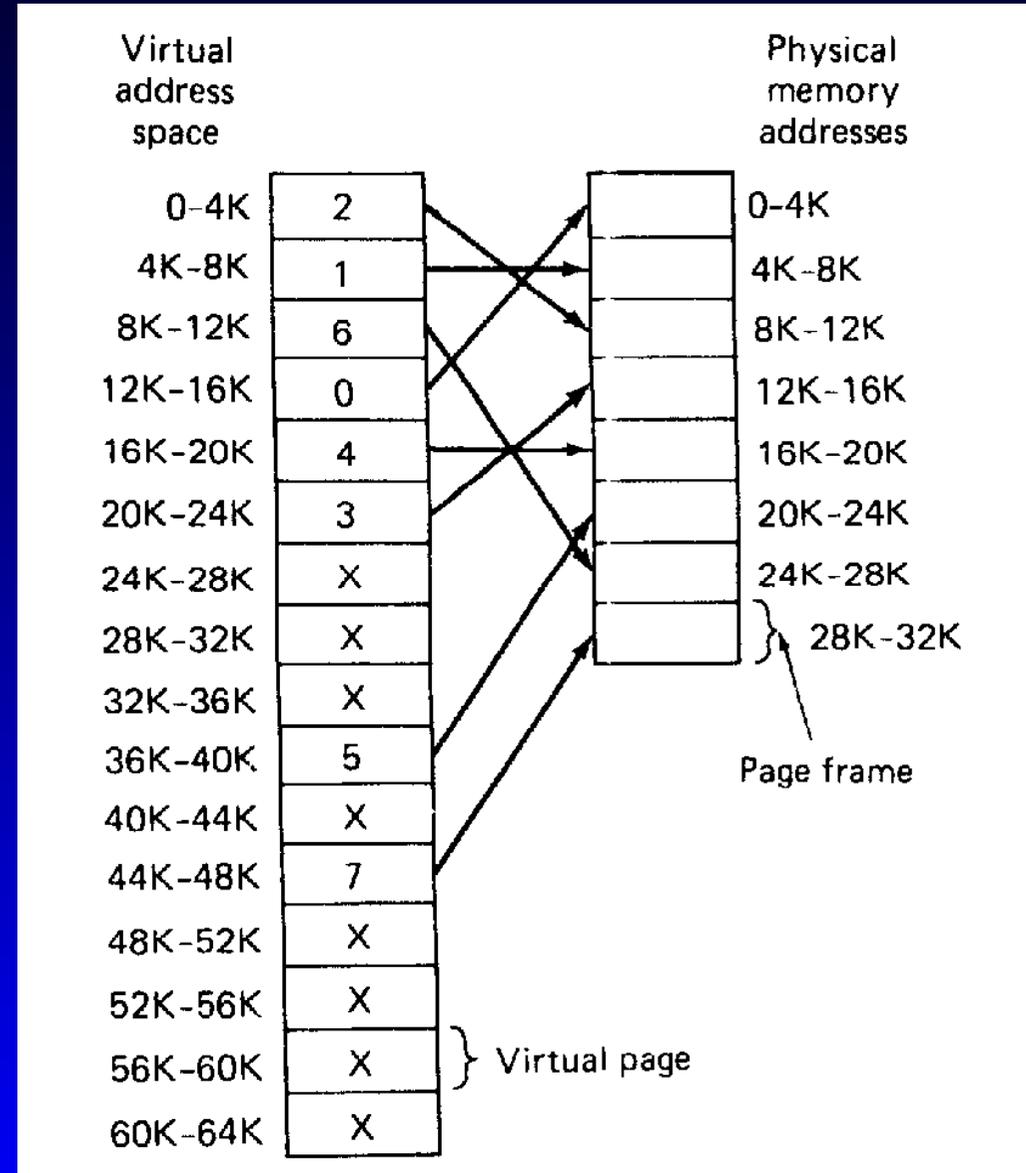


# Segmentação no Modo V86

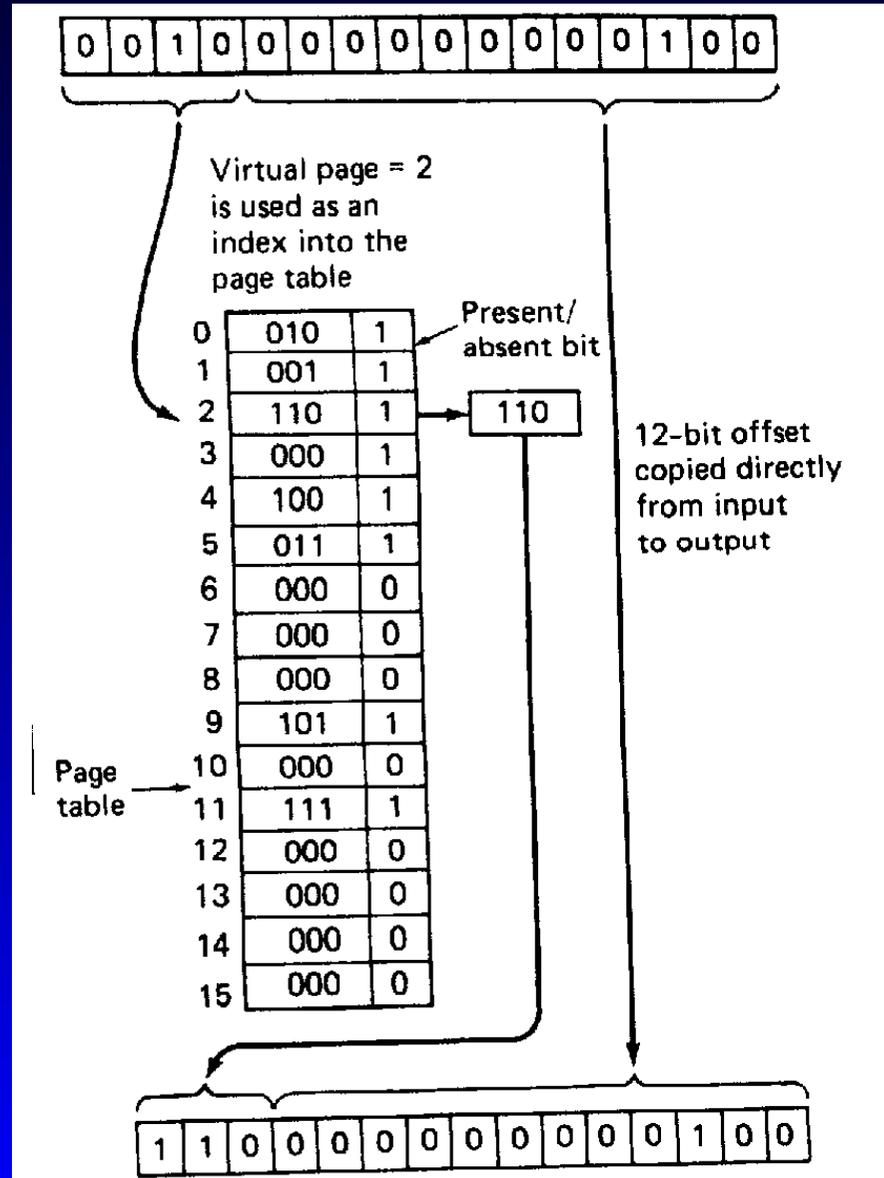
- Semelhante à segmentação no 8086
- Seletores são utilizados como base dos segmentos

# Memória Virtual

- Paginação
  - Falta de página
  - Page-table
- Endereço
  - Virtual
  - Físico



# Tabela de Páginas



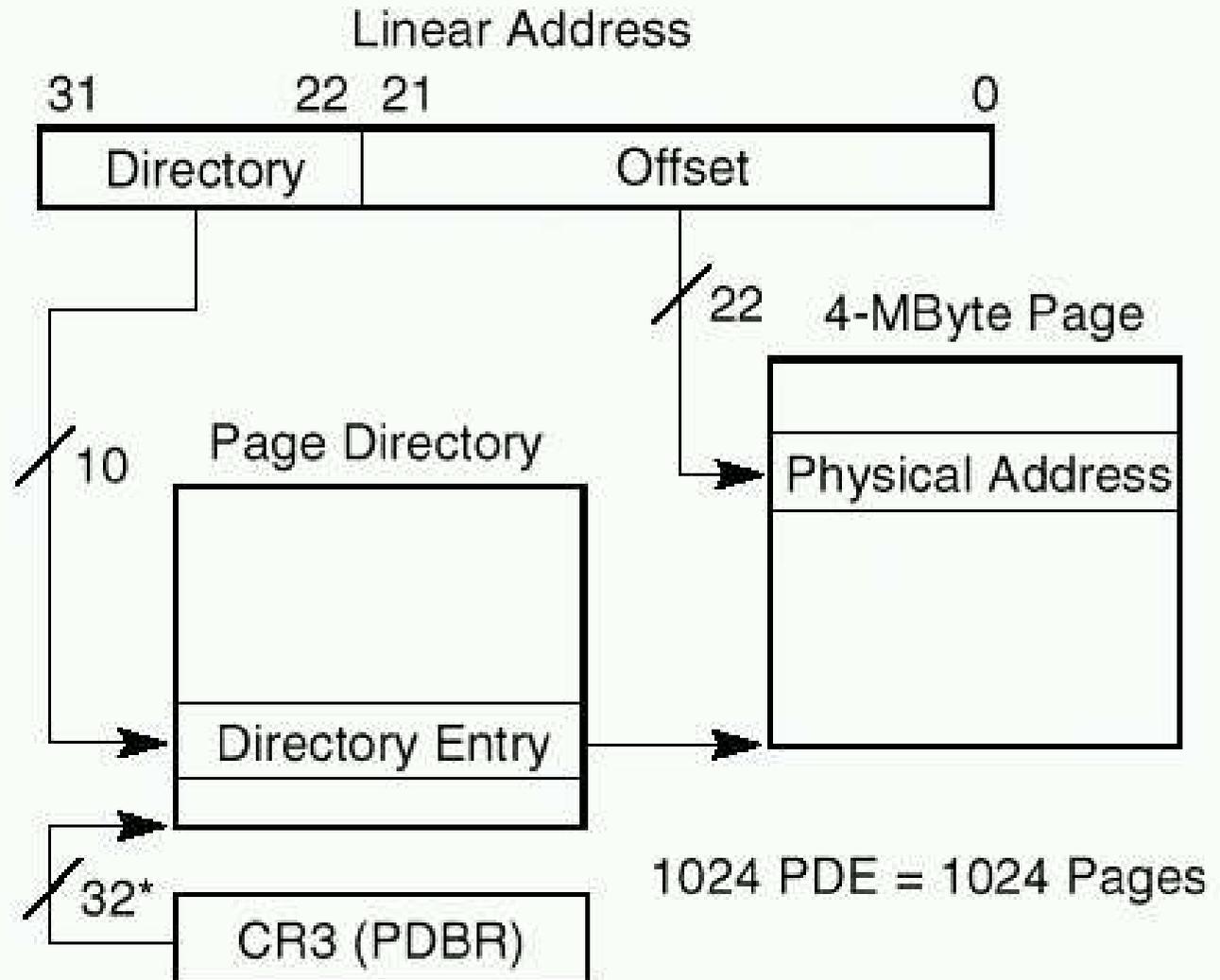


# Hardware de Paginação

- Usualmente a tabela de páginas é mantida em memória e não em registradores
  - Registrador de base da tabela de páginas
  - Translation look-aside buffer (TLB)
    - Evita dois acessos a memória
- Páginas também podem ter bits de proteção associados
  - Bit de presença/ausência



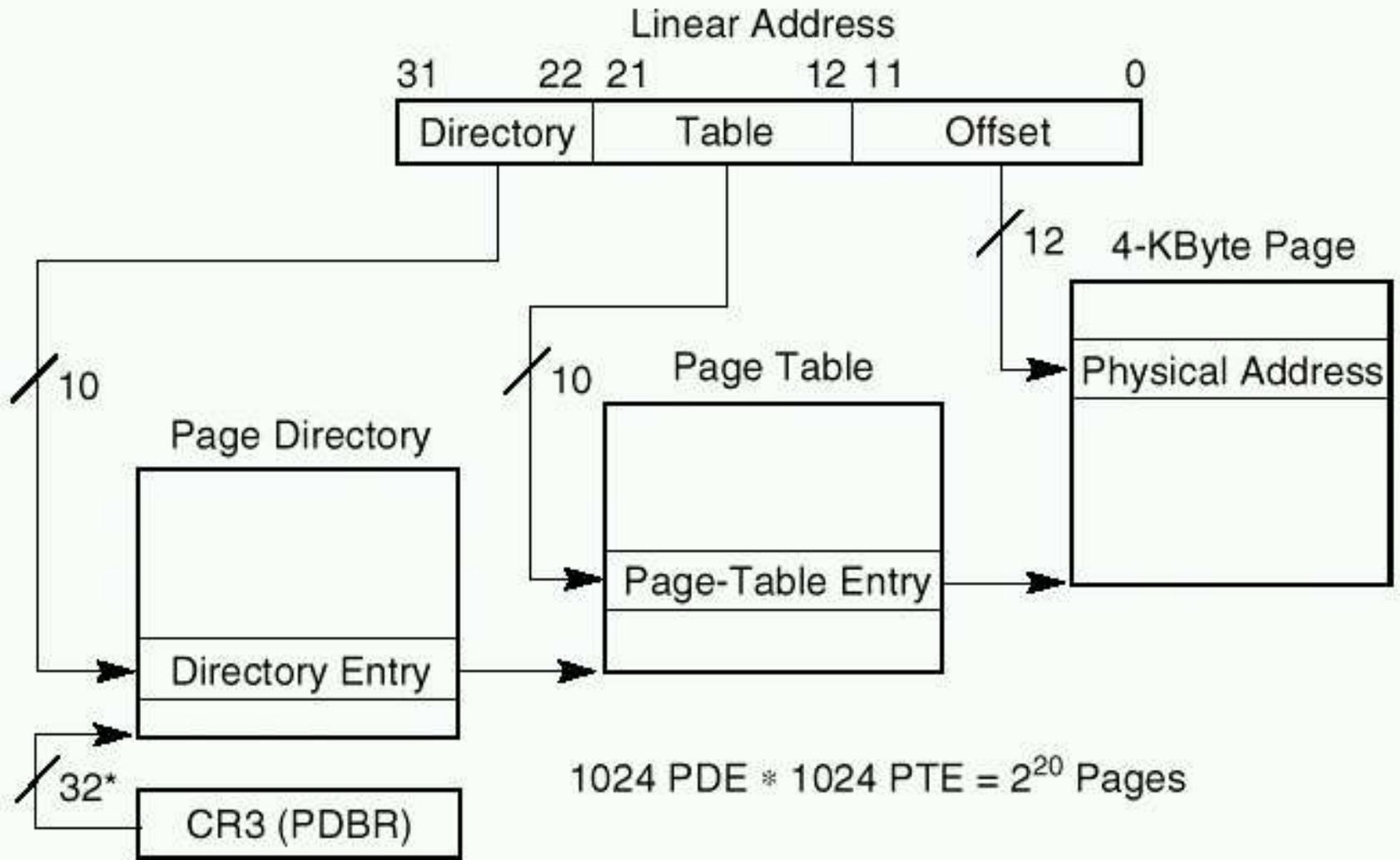
# Páginas de 4MB



\*32 bits aligned onto a 4-KByte boundary.



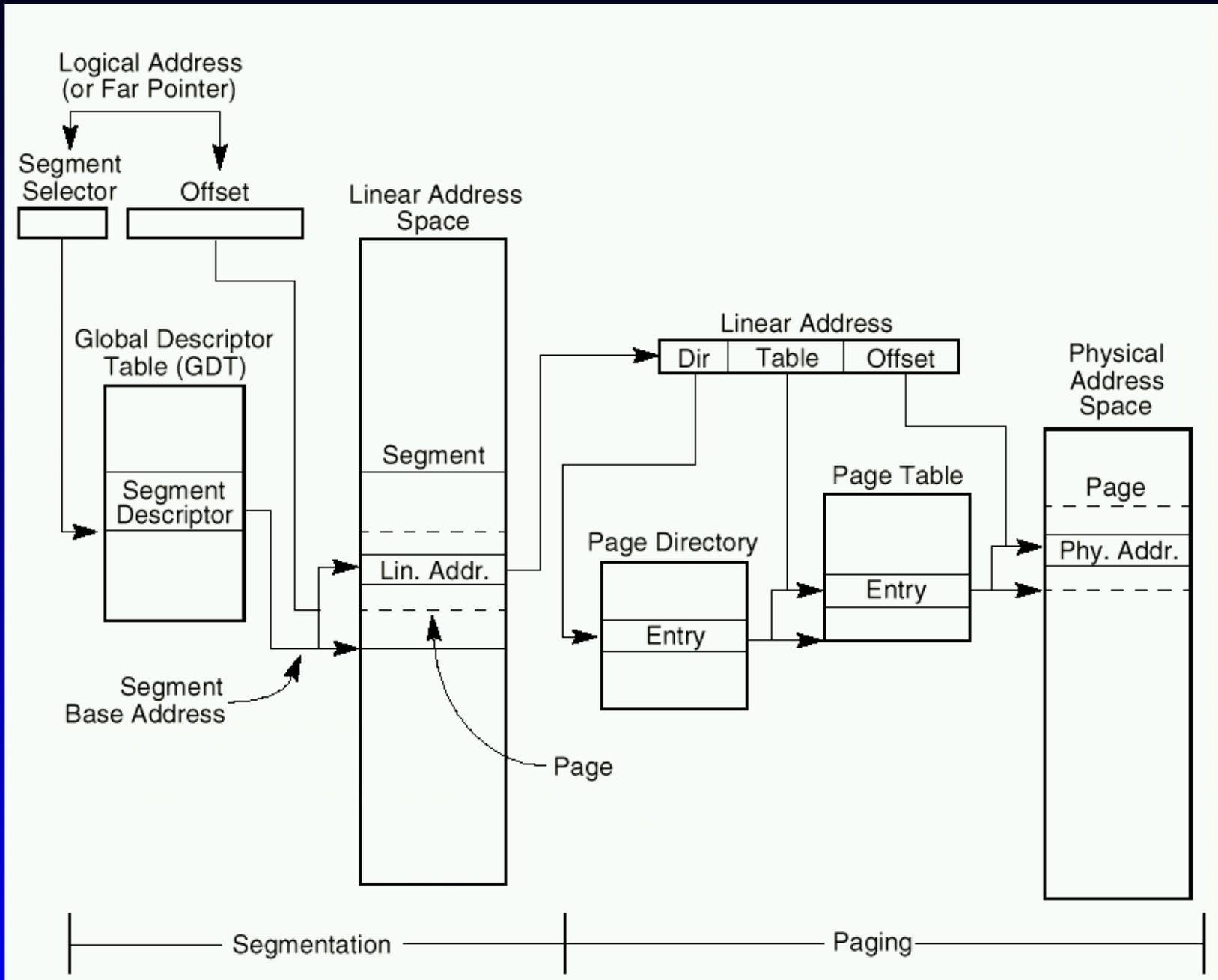
# Páginas de 4KB



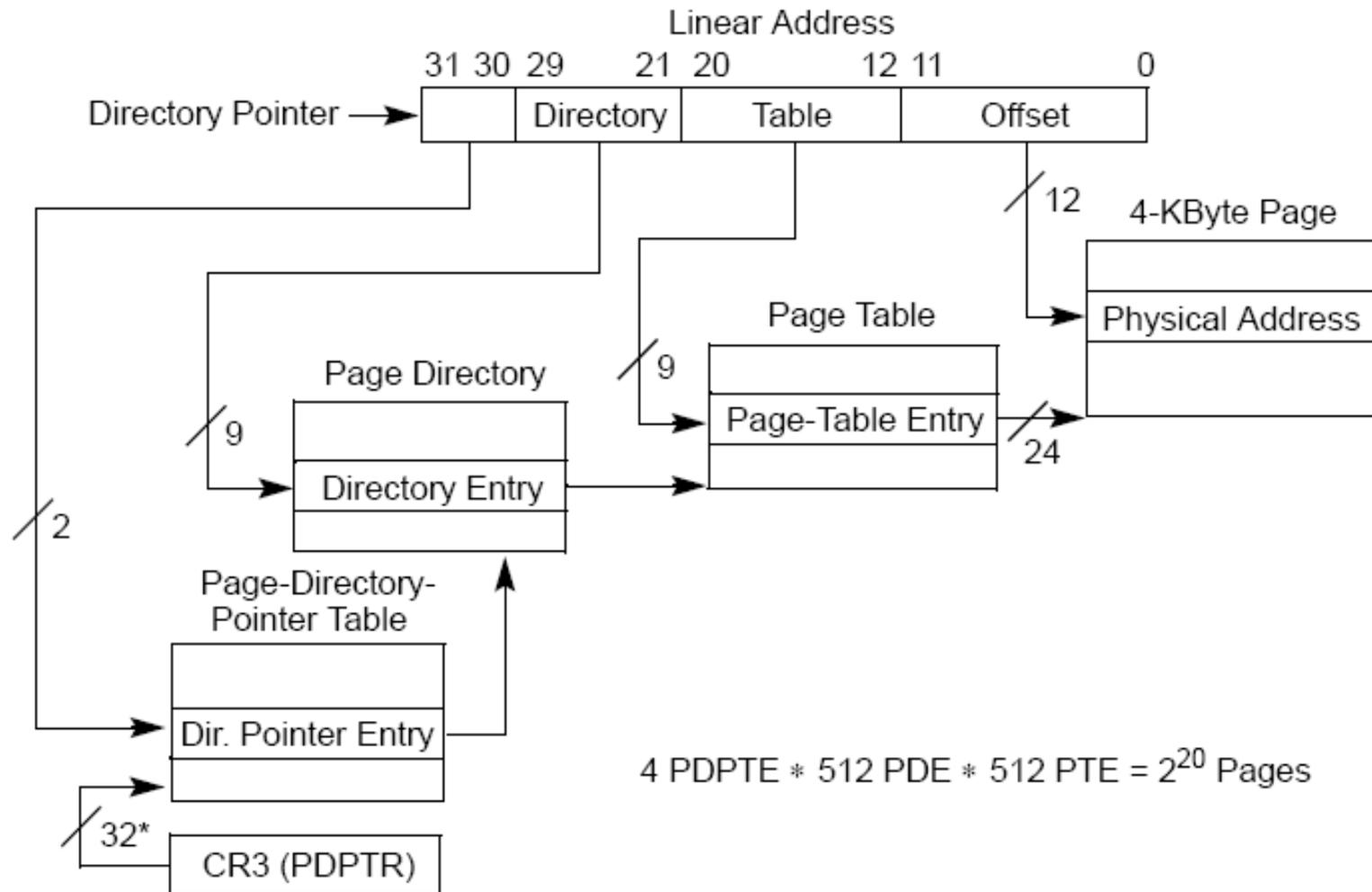
\*32 bits aligned onto a 4-KByte boundary.



# Segmentação e Paginação

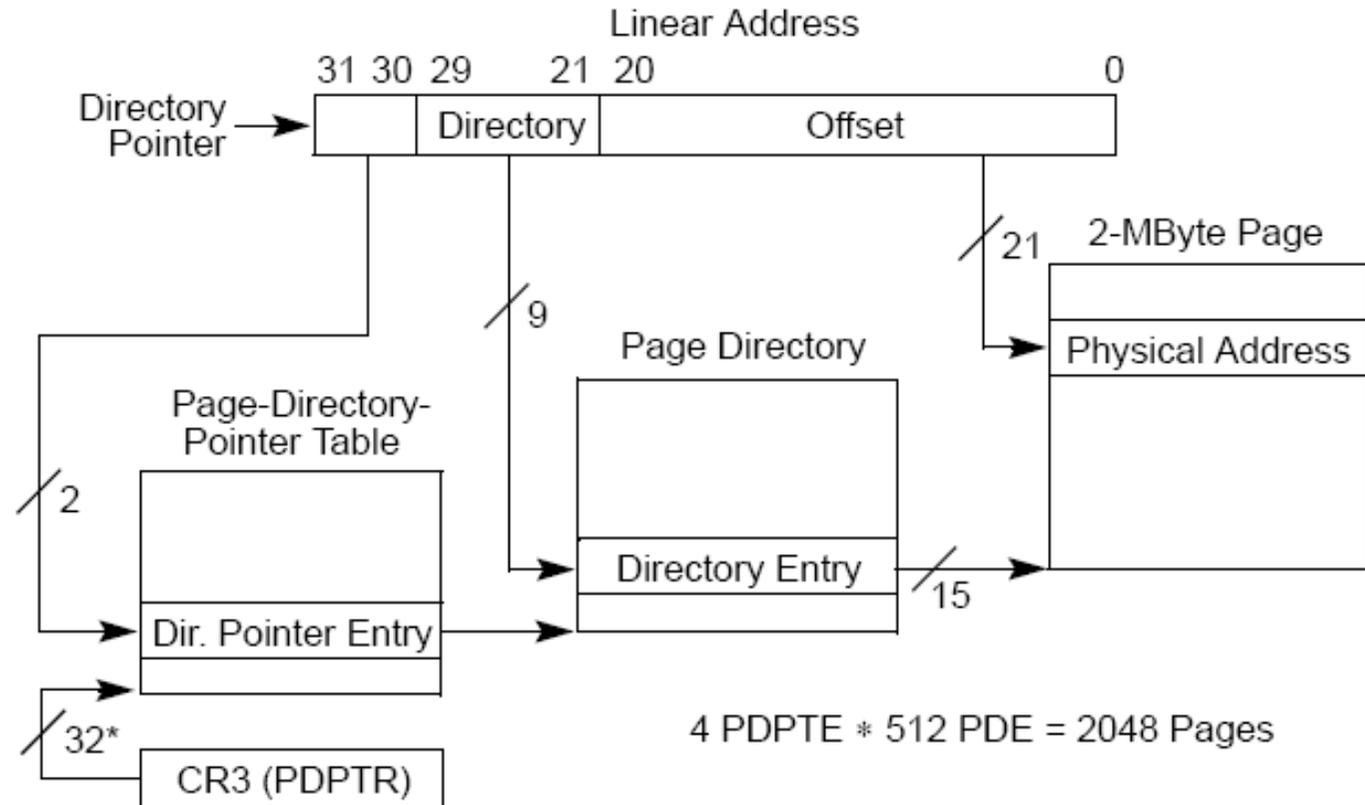


# Physical Address Extension



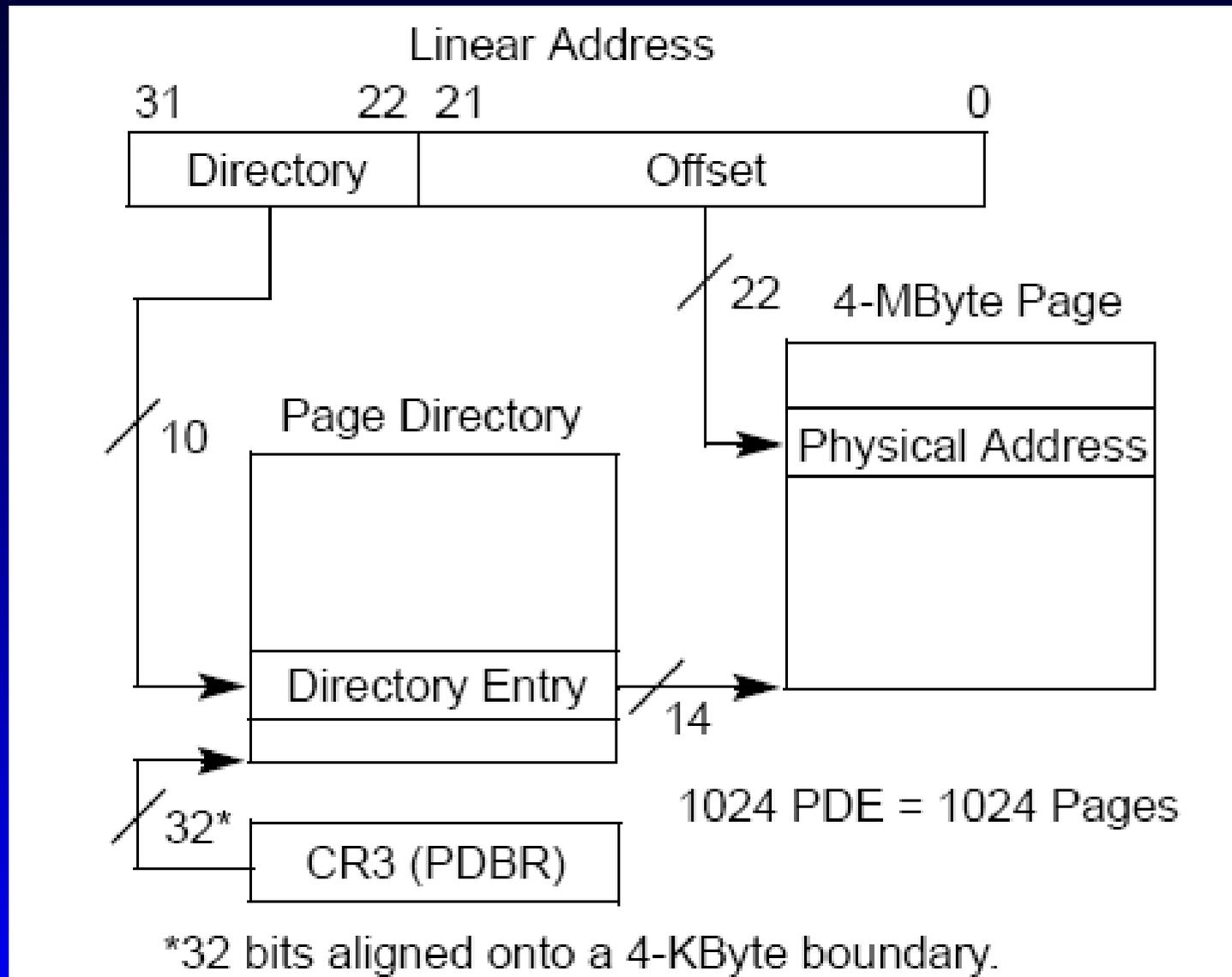
\*32 bits aligned onto a 32-byte boundary

# Páginas de 2MB



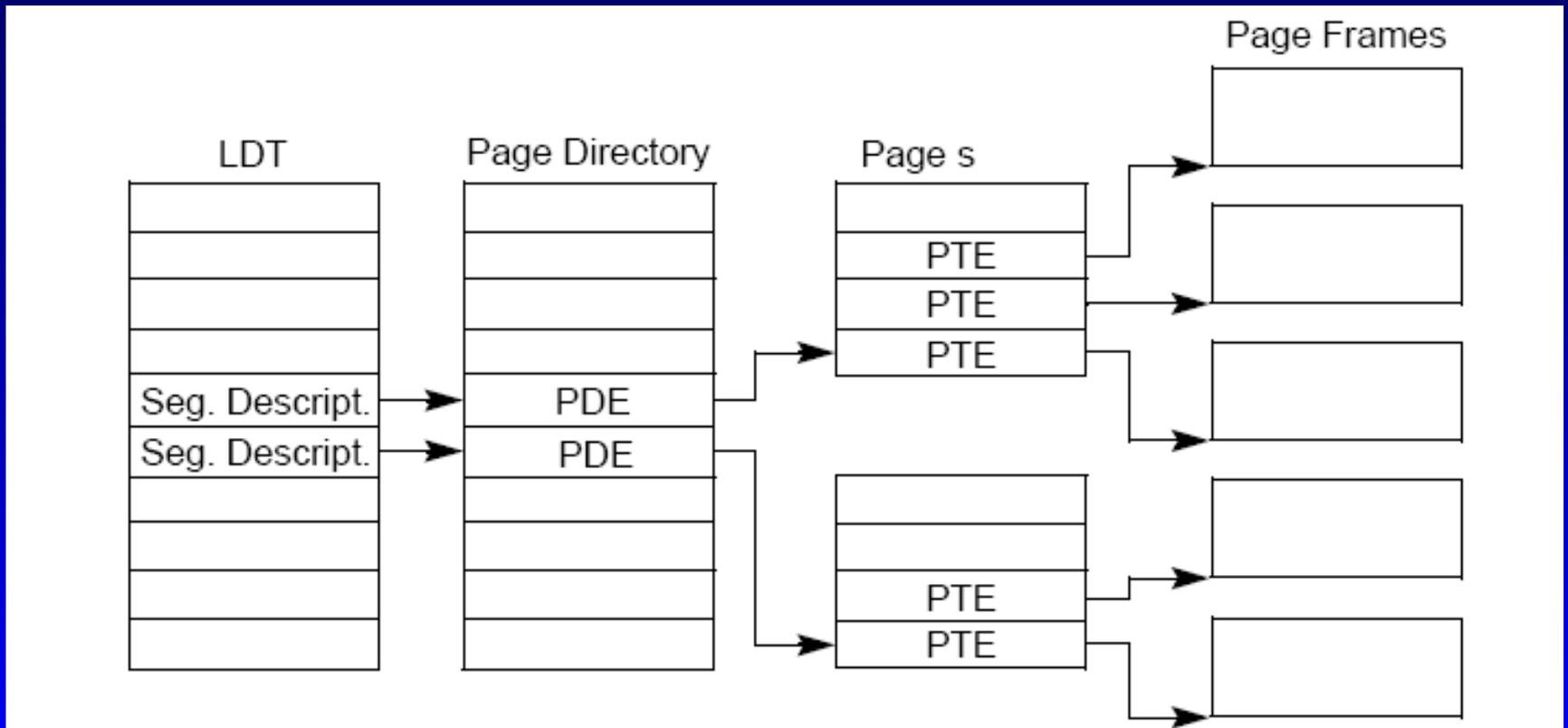
\*32 bits aligned onto a 32-byte boundary

# Page Size Extension



# Mapeamento Segmentos/Páginas

- Segmentos mapeados em páginas ou não
- Mapeamento de segmentos em páginas simplifica o gerenciamento

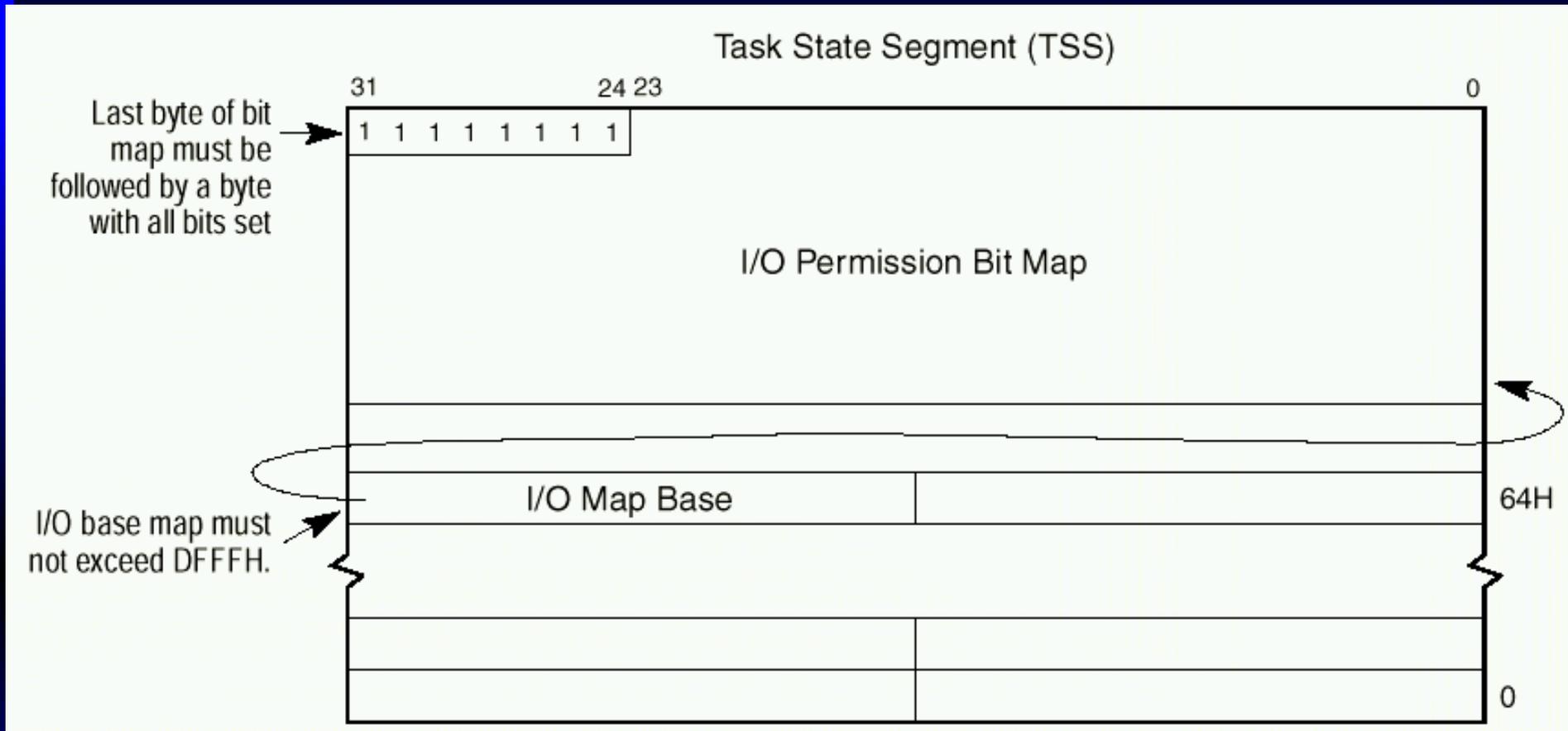




# Proteção de I/O

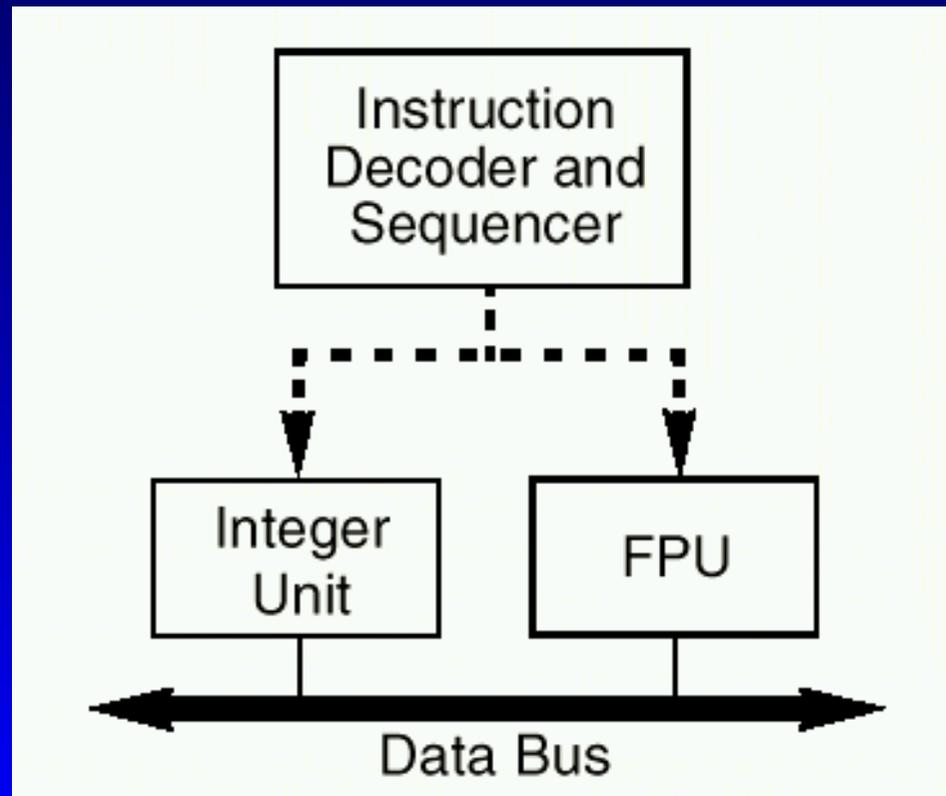
- Existe um nível de privilégio de I/O
  - Apenas processos com nível de privilégio menor ou igual podem executar instruções de I/O
- Associado a cada processo pode existir um bitmap de permissões de I/O
  - Apenas as portas habilitadas podem ser acessadas
- Hardware Virtual
  - Virtualização de dispositivos

# I/O Permission Bit Map



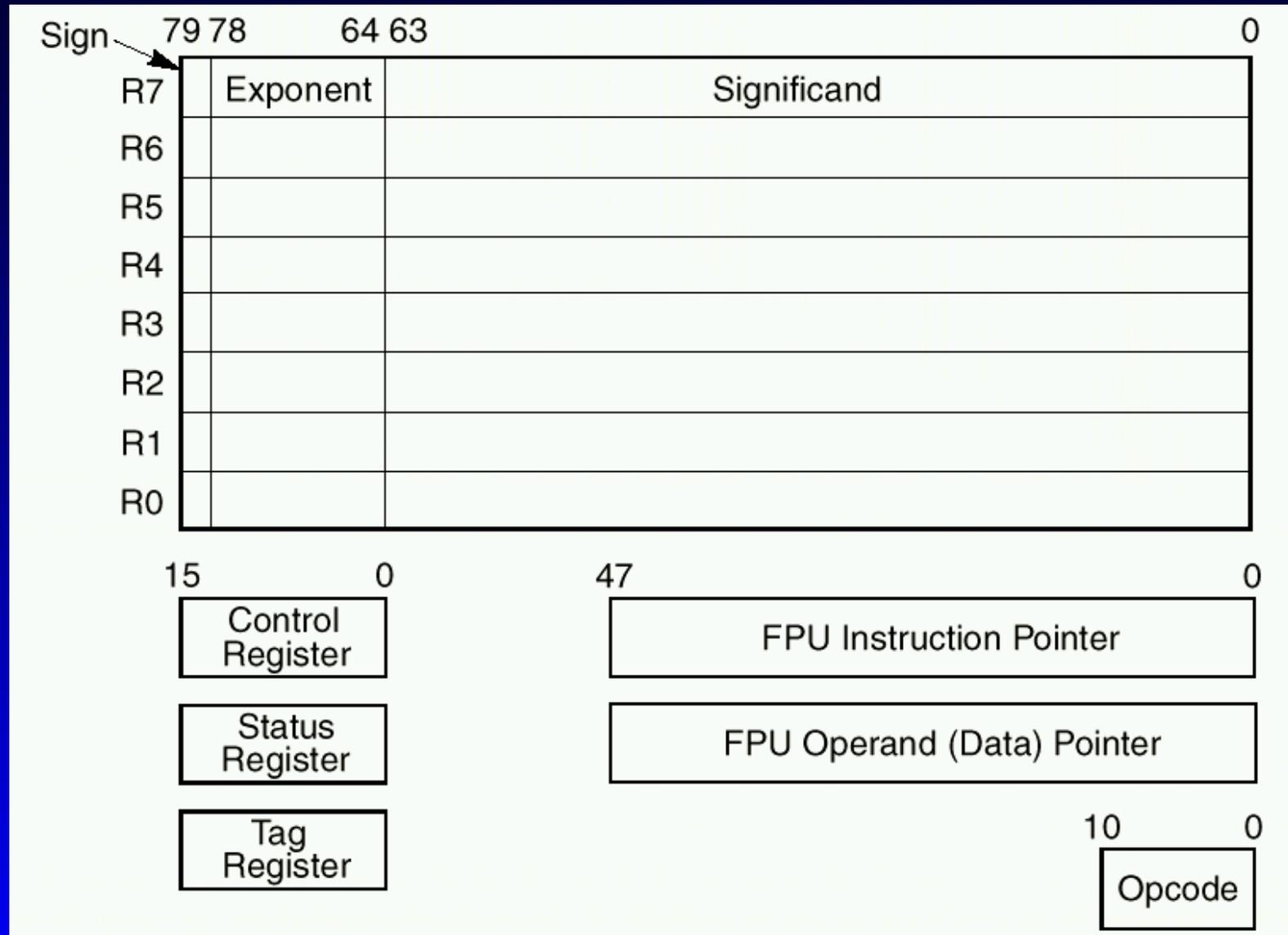
# Coprocessador (FPU)

- Registradores e opcodes ponto flutuante
- Na IA32 opera em notação polonesa reversa
- A partir do 80486DX passou a ser FPU



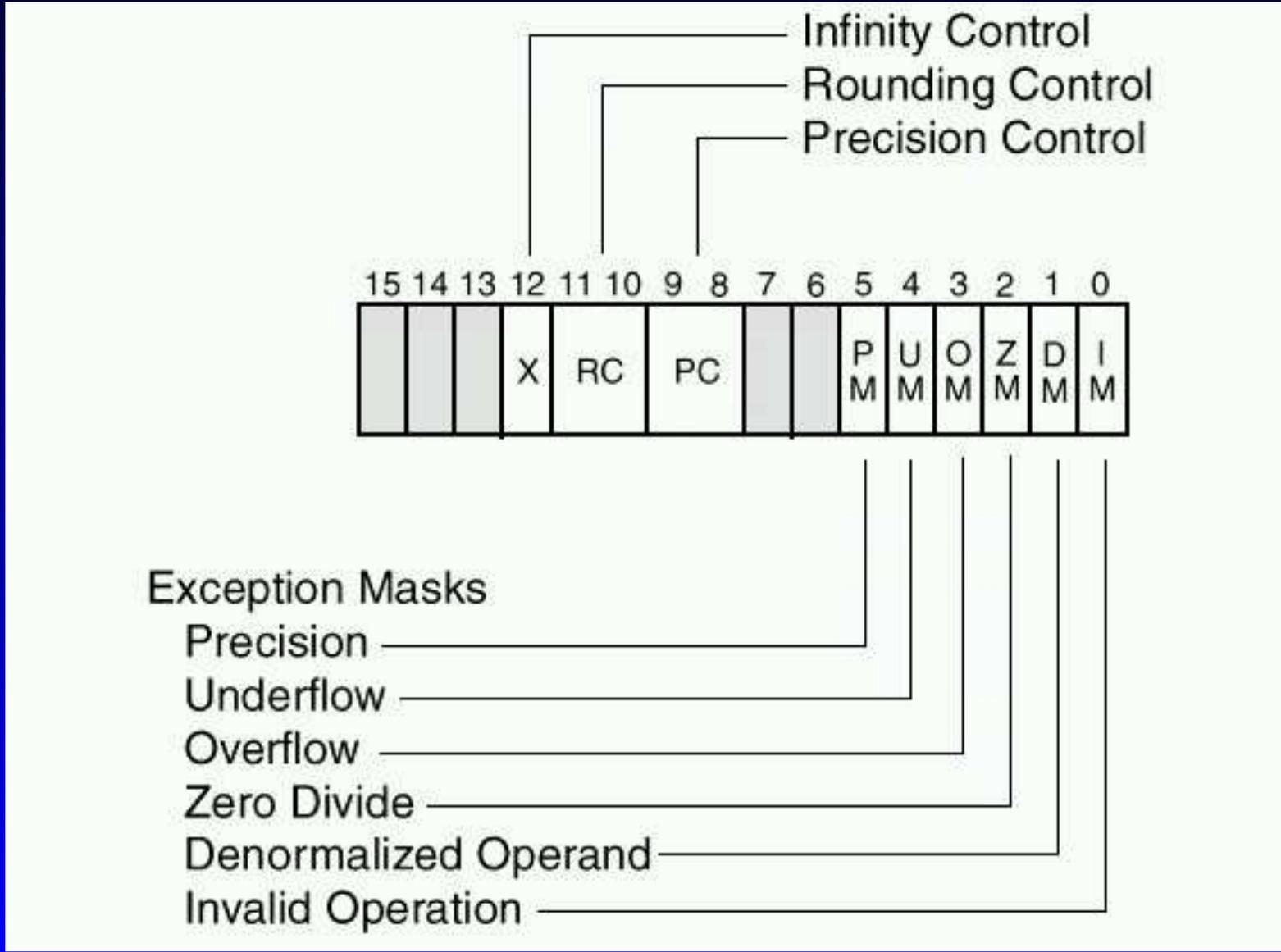


# Registradores FPU



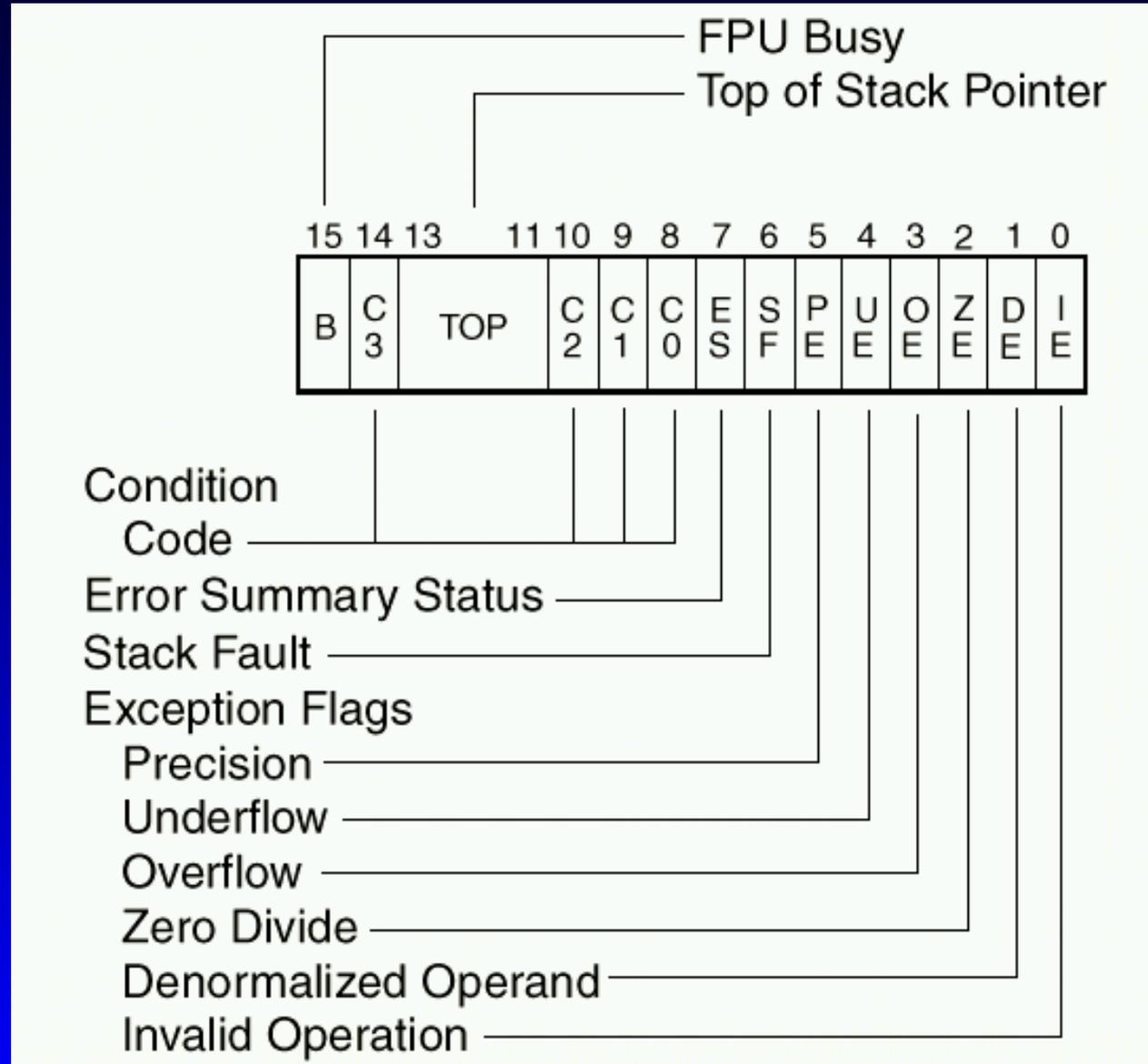


# Controle FPU





# Status FPU





# Exemplo FPU

```
.intel_syntax noprefix

.text

.global sum

# float sum(float a,float b)
sum:

    push    ebp
    mov     ebp,esp

    finit
    wait

    fld     dword ptr [ebp+8]        # a
    fadd    dword ptr [ebp+12]     # b
    fwait

    pop     ebp
    ret
```