

0 Processador Pentium

Introdução

-  A família de processadores Intel
-  O processador Pentium
-  Registos do processador Pentium
-  Arquitectura de memória do processador Pentium
 - ✓ Modo real
 - ✓ Modo protegido

Intel - Família x86 (e x86-64)

Processador	Ano	Capacidade Registos (bits)	Bus de dados (bits)	Bus de endereços (bits)	Características
8086	1978	16	16	20	Usado nos 1ºs PCs, pode endereçar 4 segmentos de 64K
8088	1979	16	8	20	Versão mais barata do 8086
80286	1982	16	16	24	Modo protegido na arquitectura de memória
80386	1985	32	32	32	IA-32
80486	1989	32	32	32	Pipeline, co-processador matemático incorporado
Pentium (P5)	1993	32	64	32	2 pipelines, super-escalar, previsão de saltos

Intel - Família x86 (e x86-64)

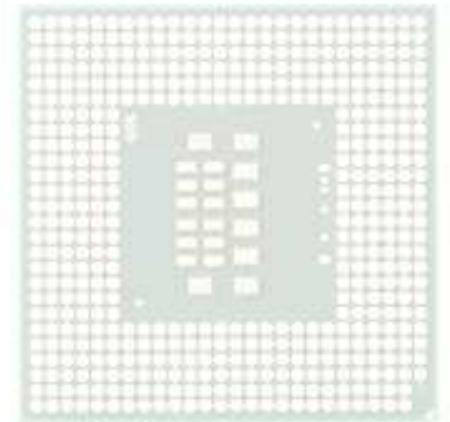
Processador	Ano	Capacidade Registos	Bus de dados	Bus de endereços	Características
Pentium (P5)	1993	32	64	32	2 pipelines, super-escalar, previsão de saltos
Pentium II (P6)	1997	32	64	36	Super-escalar de 3 inst., execução fora de ordem e especulativa, instr. MMX de raiz, gestão de energia, cache L2
Pentium III (P6)	1999	32	64	36	Instruções SSE
Pentium IV (NetBurst ou P7)	2000	32 ou 64	64	36	Pipeline grande, instruções SSE2 e SSE3, 1º linha Intel c/ dual-cores, 1º linha Intel com x86-64, Hyper-threading (HThreading)

Intel - Família x86 (e x86-64)

Processador	Ano	Capacidade Registos	Bus de dados	Bus de endereços	Características
Core e Core 2 (Core)	2006	32 (Core) 64 (Core 2)	128	36	Relógios mais baixos, mais memória cache, super-escalar 4 inst, pipeline menor, sem HThreading, quad-cores
Atom	2008	32 ou 64	64	36	Baixa voltagem Volta o Hthreading Execução em ordem
Core i7, Core i5, Core i3 (Nehalem)	2009 2010	64	128	36	Volta o Hthreading, controlador de memória no CPU, novas tecnologias para o FSB (Quick-path, Direct Media Interface), GPUs incorporados (portáteis), cache L3, ...

Intel - Família x86 (e x86-64)

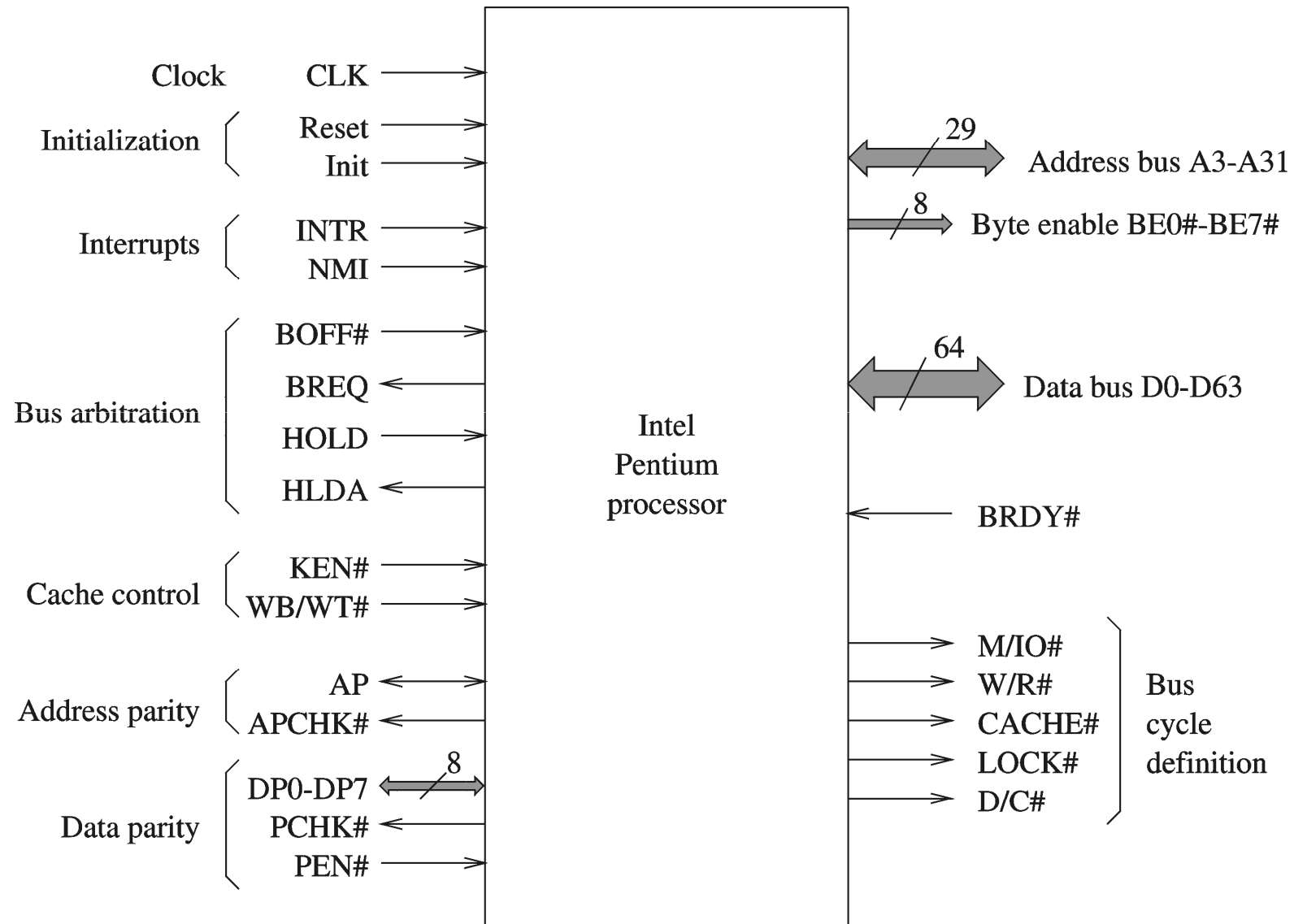
- 💻 Todos os modelos são compatíveis com o 8086
 - ✓ Existem instruções para manipular dados de 8 e 16bits em todos os modelos: 80386, Pentium, Core 2, Core i7, ...



Intel - Família Itanium (IA-64)

-  O ISA IA-64 (introduzido em 2001) é implementado pela família de processadores Intel Itanium
-  Não é compatível com o IA-32 (também conhecido por x86) nem com a sua extensão a 64 bits (x86-64)
 - ✓ A compatibilidade só é possível através de um emulador implementado em hardware
-  Processador RISC de 64 bits
 - ✓ Bus de dados de 128 bits - bus de endereços de 44 bits
-  Falaremos deste processador em detalhe mais tarde

O Processador Pentium



O Processador Pentium



Bus de dados

- ✓ D0-D63 → 64 bits
- ✓ BE0#-BE7# → indica que bytes (dos 8 que estão no bus de dados) devem ser lidos ou escritos



Bus de endereços

- ✓ A3-A31 → Só 29 linhas para endereços de 32 bits?
- ✓ Os acessos são alinhados a múltiplos de 8 pois o bus de dados é de 64 bits (8 bytes)



Paridade dos dados

- ✓ DP0-DP7 → Paridade dos bytes recebidos
- ✓ PCHK# → Resultado do teste de paridade
 - ✓ O teste só é realizado para os bytes activos (enabled)
- ✓ PEN# → Activa/Desactiva o teste de paridade

O Processador Pentium

Paridade dos endereços

- ✓ AP → Paridade do endereço
- ✓ APCHK# → Resultado do teste de paridade

Definição do ciclo do bus

- ✓ M/IO# → Indica se a operação é sobre memória ou I/O
- ✓ W/R# → Indica se a operação é de escrita ou leitura
- ✓ D/C# → Indica se a operação é sobre dados ou código
- ✓ CACHE# → Indica que os dados vão ou vêm da cache interna
- ✓ LOCK# → Acesso exclusivo ao bus

Interrupções

- ✓ INTR → Recepção de interrupção mascarável
- ✓ NMI → Recepção de interrupção não-mascarável

O Processador Pentium

Relógio

- ✓ CLOCK → Sinal de relógio

Ciclos de espera

- ✓ Utilizado por dispositivos externos para informar o CPU da necessidade de introduzir ciclos de espera

Arbitragem do bus

- ✓ BREQ → Pedido de acesso ao bus
- ✓ BOFF# → Sinal de interrupção dos ciclos de bus pendentes
 - ✓ São retomadas quando o sinal é desactivado
- ✓ HOLD → Termina ciclos pendentes e activa HLDA para oferecer o controlo do bus a outro mestre

O Processador Pentium

Controlo da cache

- ✓ KEN# → Indica se a cache interna pode receber dados (uma linha) no ciclo corrente
- ✓ WB/WT# → Determina a política de propagação de escritas na cache dinamicamente

Inicialização

- ✓ RESET → Reset do estado do CPU
 - ✓ “Apaga” os dados caches internas
- ✓ INIT → Equivalente a RESET mas mantém os dados nas caches

Registos do Processador Pentium

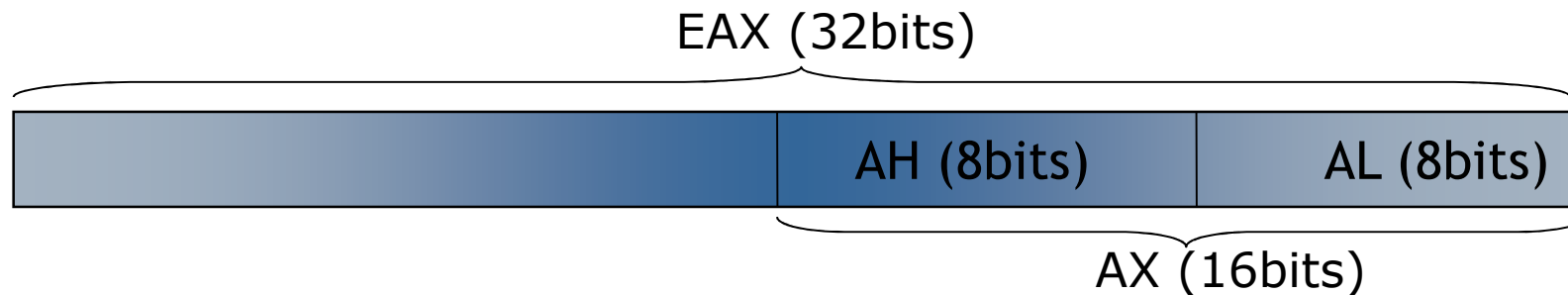
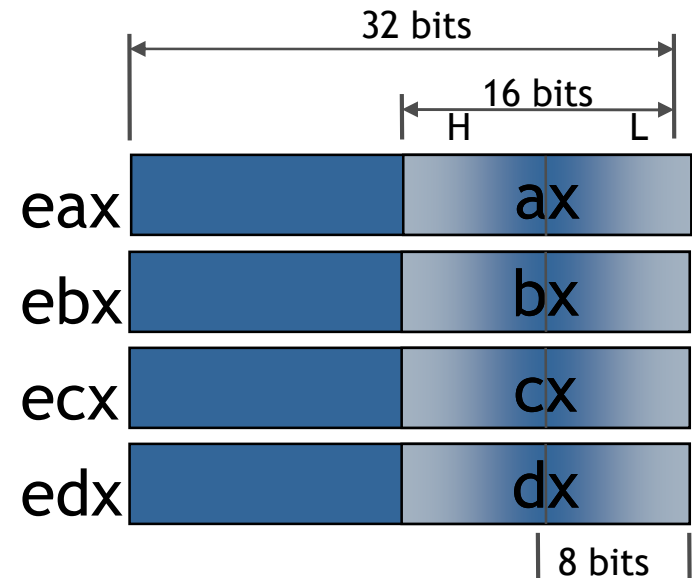
 4 registos de uso geral: *EAX*, *EBX*, *ECX*, *EDX*

✓ Pode aceder-se a sub-partes dos registos

✓ Exemplo: *EAX*

✓ *AX* - 16 bits

✓ *AL* e *AH* - 8 bits



Registos do Processador Pentium

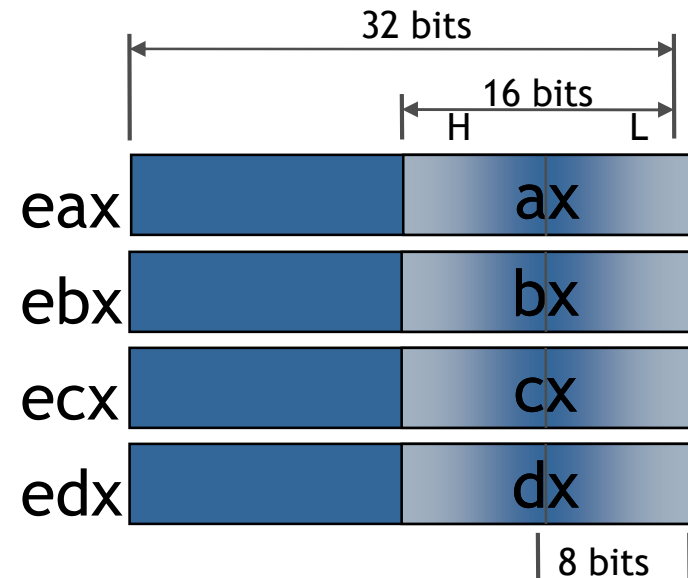
🖥️ 4 registos de uso geral: *EAX*, *EBX*, *ECX*, *EDX*

✓ Pode-se aceder a subpartes dos registos

✓ Exemplo: *EAX*

✓ *AX* - 16 bits

✓ *AL* e *AH* - 8 bits



🖥️ Podem ser utilizados implicitamente por certas instruções

✓ *Por exemplo a instrução LOOP decrementa o ECX automaticamente*

🖥️ Continuam a existir instruções com dados de 16 e 8bits, com os mesmos códigos do 8086

Registos do Processador Pentium

 2 registos de índice: ESI (Source Index) e EDI (Destination Index)

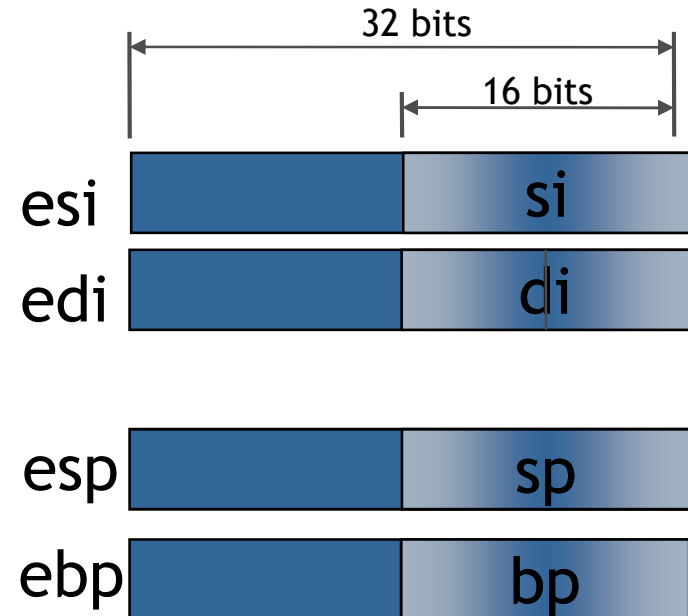
- ✓ Utilizados implicitamente por instruções de manipulação de strings

- ✓ Podem ser utilizados para uso geral

 2 registos de endereços: ESP (Stack Pointer) e EBP (Base Pointer)

- ✓ Usados na manipulação da pilha (stack)

- ✓ ESP usado implicitamente pelas instruções push e pop



Registos do Processador Pentium

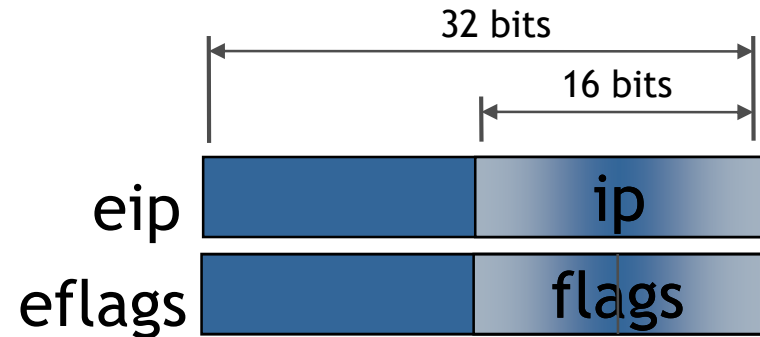


Registo de flags: EFLAGS

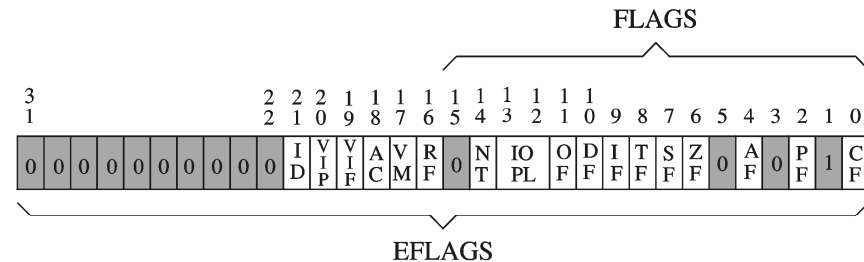
- ✓ Flags de estado
 - ✓ Bits de condição
- ✓ Flags de controlo e sistema



Program Counter: EIP (Instruction Pointer)



Flags register



Status flags

CF = Carry flag
 PF = Parity flag
 AF = Auxiliary carry flag
 ZF = Zero flag
 SF = Sign flag
 OF = Overflow flag

Control flags

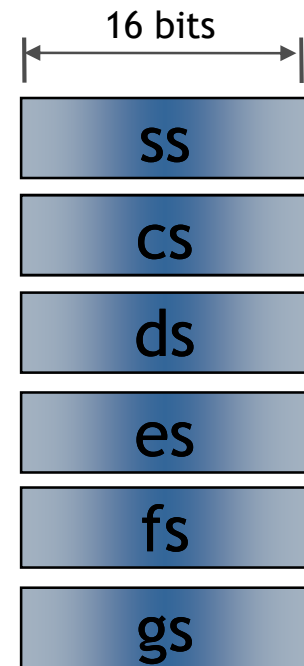
DF = Direction flag

System flags

TF = Trap flag
 IF = Interrupt flag
 IOPL = I/O privilege level
 NT = Nested task
 RF = Resume flag
 VM = Virtual 8086 mode
 AC = Alignment check
 VIF = Virtual interrupt flag
 VIP = Virtual interrupt pending
 ID = ID flag

Registos do Processador Pentium

-  Registos para suporte à segmentação:
SS (Stack Segment), CS (Code Segment),
DS (Data Segment), ES, FS e GS (Extra
Data Segments)
- ✓ Falaremos de segmentação mais tarde



Arquitectura de Memória

 O Pentium suporta duas arquitecturas de memória

- ✓ Modo real (real mode)
 - ✓ Endereçamento a 16 bits
 - ✓ Executa programas do 8086
 - ✓ O Pentium serve apenas como um 8086 mais rápido
- ✓ Modo protegido (protected mode)
 - ✓ Endereçamento a 32 bits
 - ✓ Modo nativo do Pentium
 - ✓ Suporta segmentação e paginação
 - ✓ Falaremos de ambos no capítulo sobre gestão de memória

Endereços Lógicos e Físicos



Endereço lógico, virtuais ou efectivo:

- ✓ Utilizado pelo programa a nível das instruções de referência a memória
- ✓ A dimensão máxima do espaço de endereçamento virtual é determinada pela capacidade dos registos de endereços e modos de endereçamento



Endereços físicos ou reais:

- ✓ Definidos pelas linhas de endereço do bus que dão acesso às células físicas da memória central
- ✓ Espaço de endereçamento real é limitado pela número de linhas no bus e pela capacidade da memória central realmente instalada em cada computador

Arquitectura de Memória - Modo Real

🖥️ O endereço lógico (EL) tem duas componentes

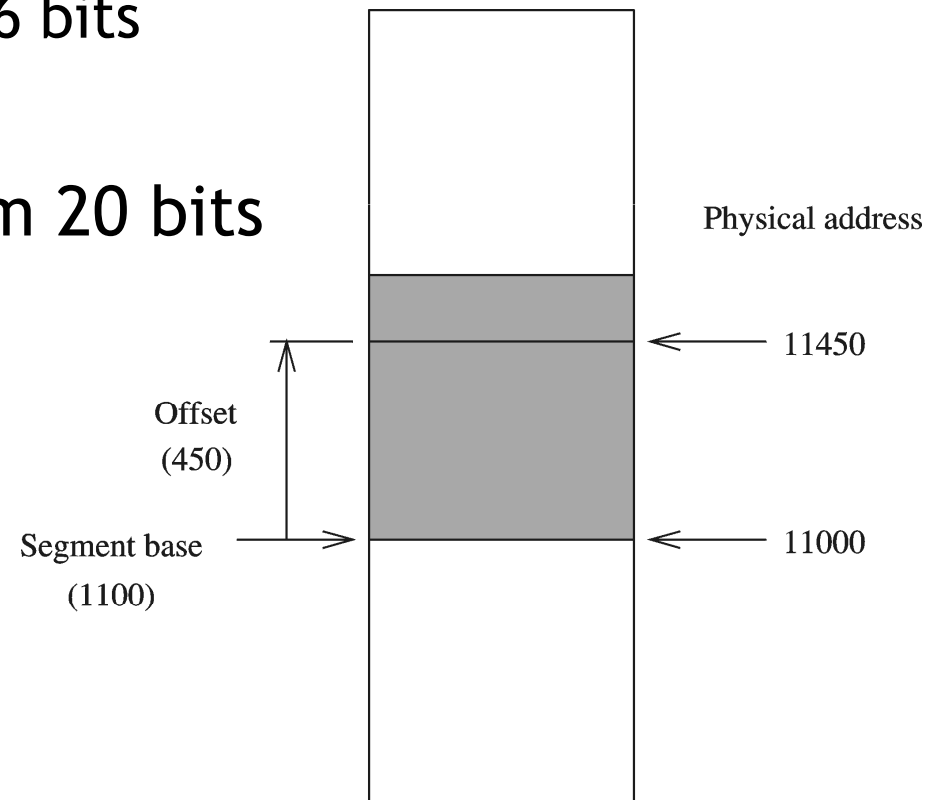
- ✓ Base - 16 bits

- ✓ Dado pelo registo de segmentação

- ✓ Deslocamento (offset) - 16 bits

🖥️ O endereço físico (EF) tem 20 bits

$$\begin{array}{ccccc} \text{Offset} & + & 2^4 * \text{RegSeg} & \rightarrow & \text{EF} \\ \downarrow & & \underbrace{(4\text{bits}) * (16\text{bits})} & & \downarrow \\ (16\text{bits}) & + & (20\text{bits}) & \rightarrow & (20\text{bits}) \end{array}$$

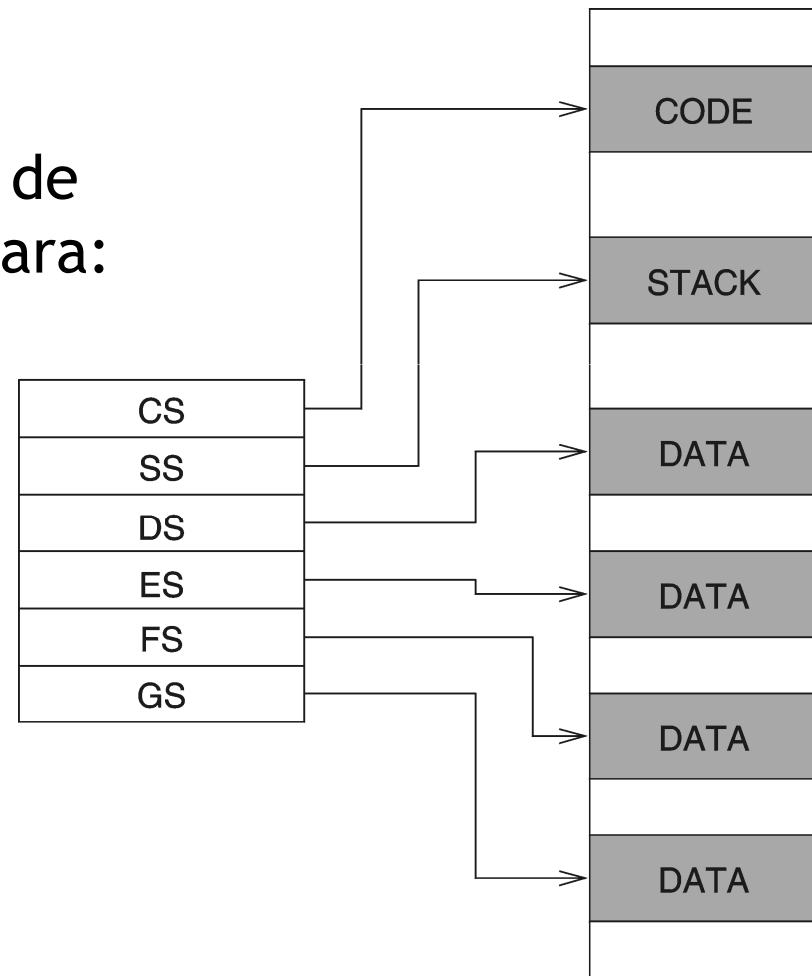


Arquitectura de Memória - Modo Real



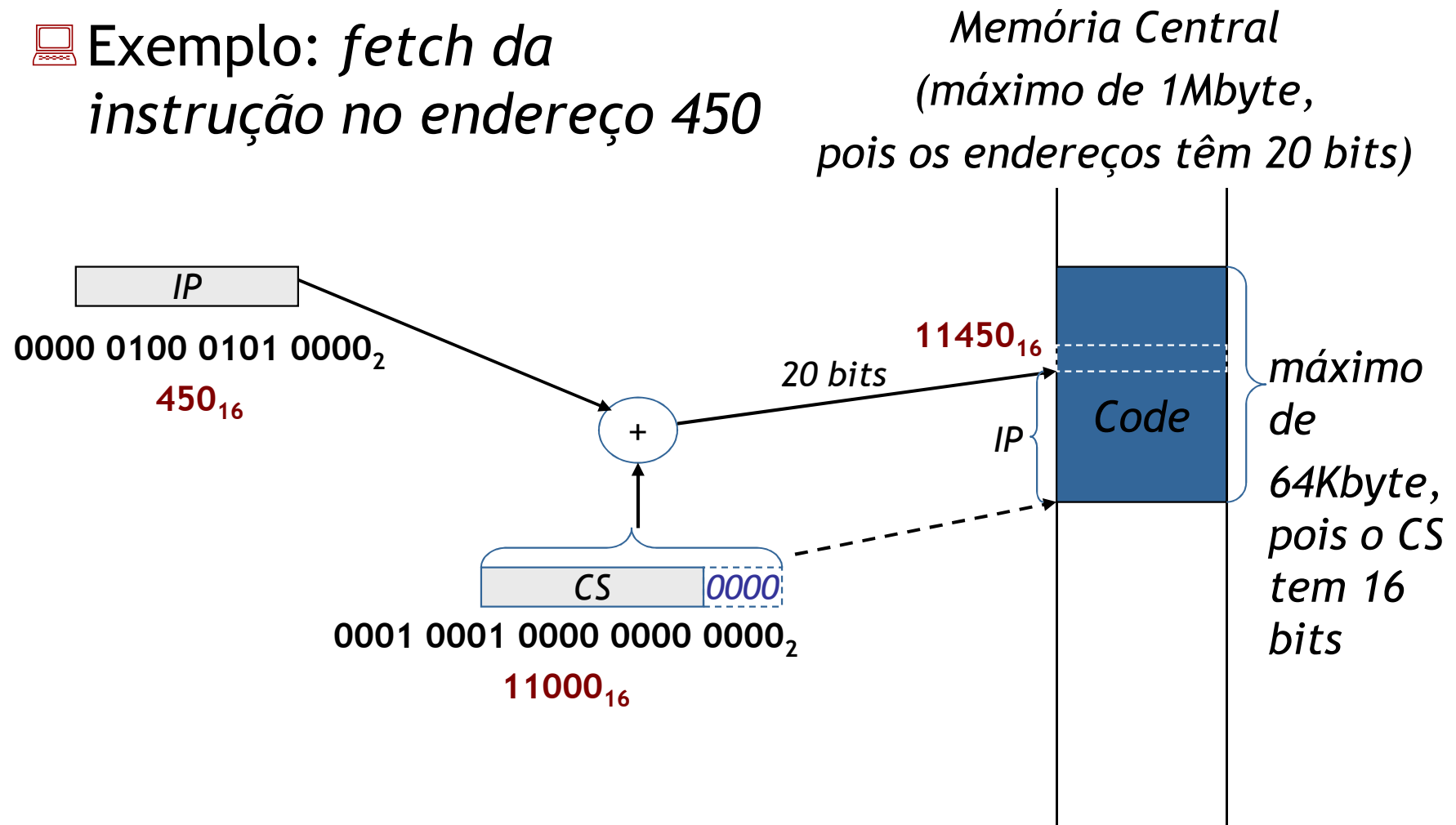
Um programa pode utilizar os seis registos de segmentação disponíveis

- ✓ Permite usufruir de espaços de endereçamento separados para:
 - ✓ Pilha (Stack)
 - ✓ Dados
 - ✓ Código



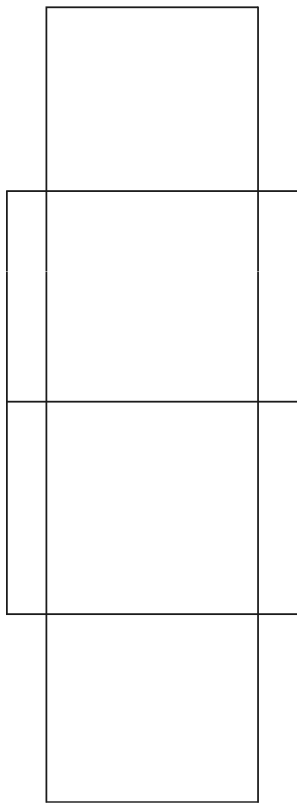
Arquitetura de Memória - Modo Real

 Exemplo: *fetch da instrução no endereço 450*

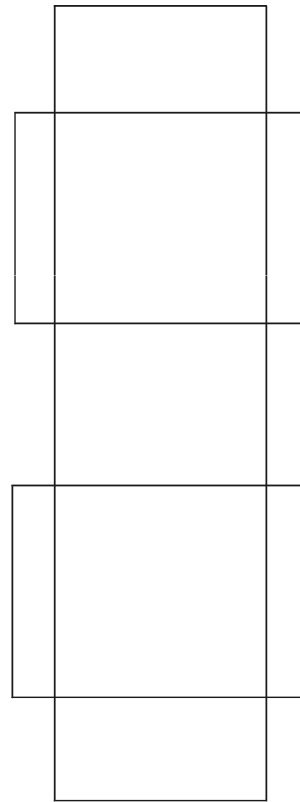


Arquitetura de Memória - Modo Real

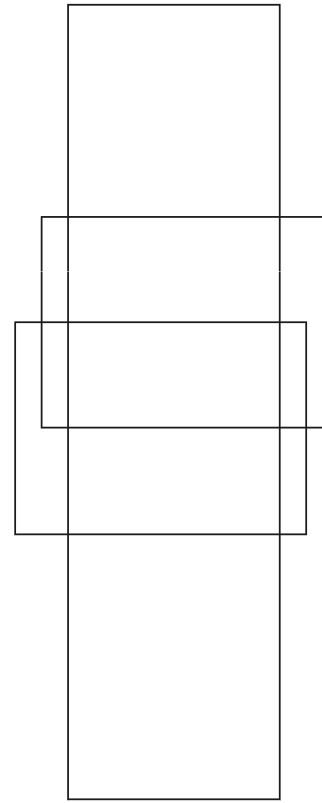
 Os segmentos pode ser disjuntos, adjacentes, parcial ou totalmente sobrepostos



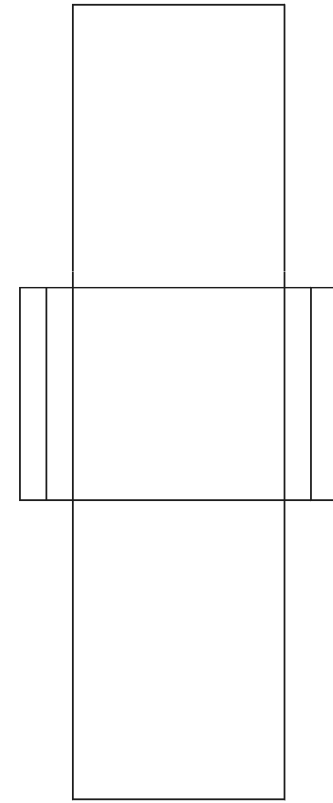
(a) Adjacent



(b) Disjoint



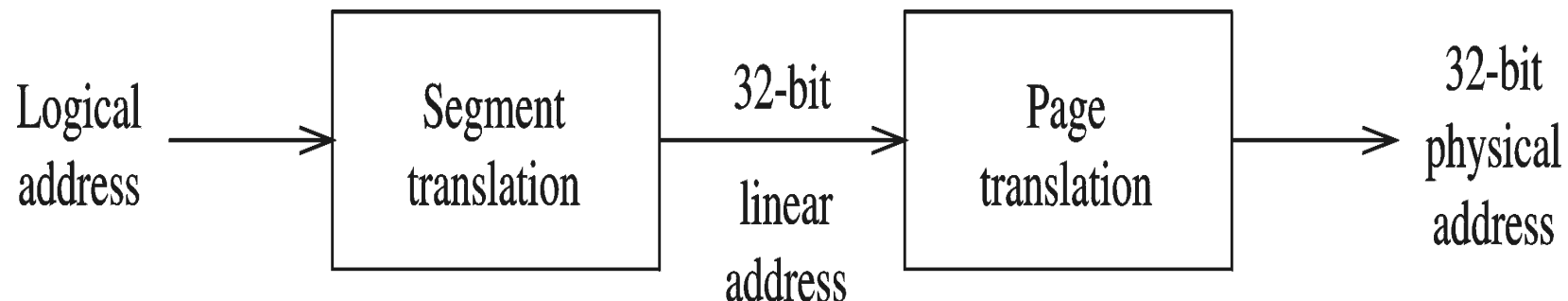
(c) Partially overlapped



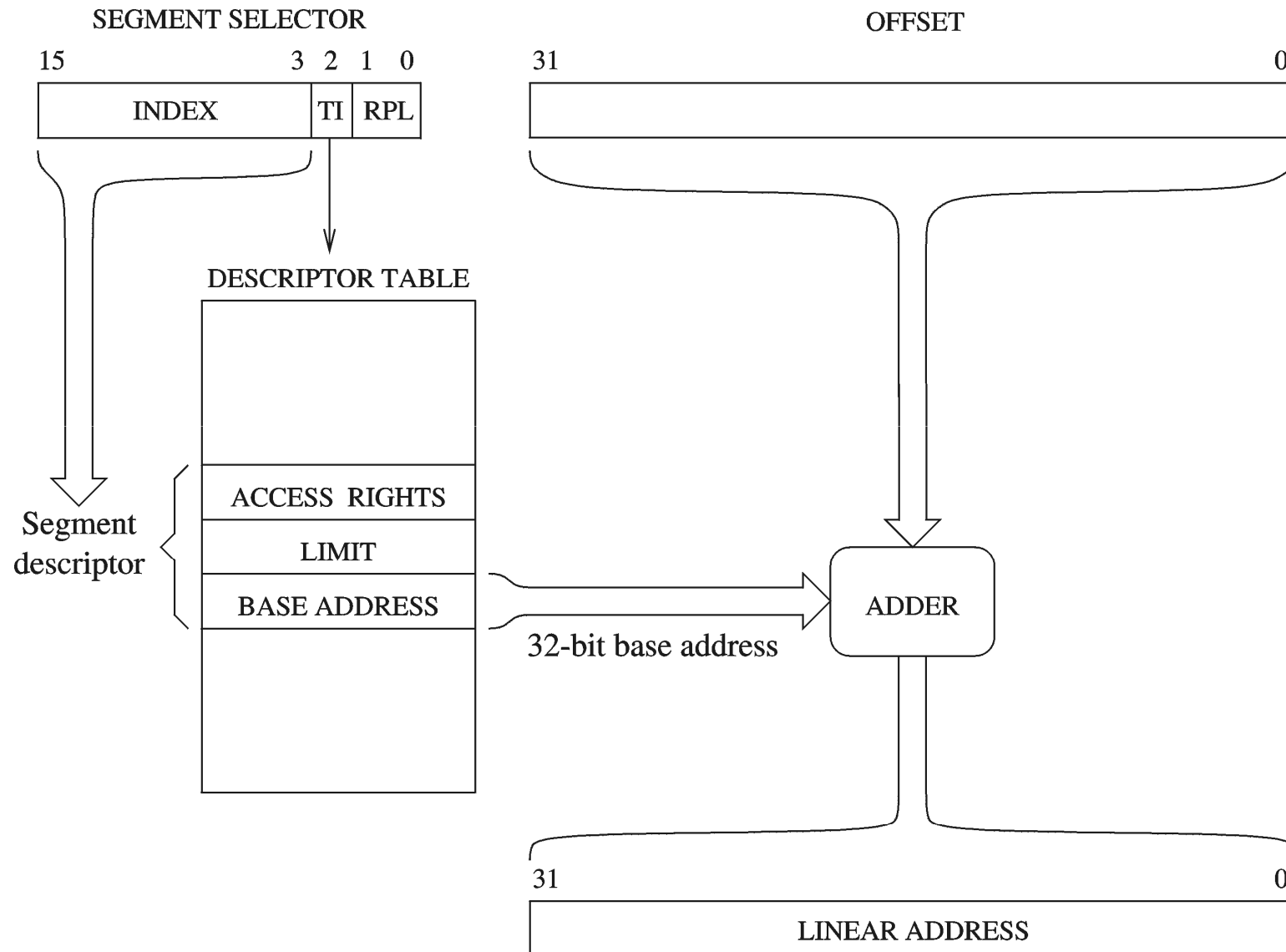
(d) Fully overlapped

Arquitetura de Memória - Modo Protegido

- 🖥️ Mecanismo de segmentação mais sofisticado
- 🖥️ A unidade de segmentação traduz endereços lógicos de 32 bits para endereços lineares de 32 bits
 - ✓ Endereço linear → endereço dentro do segmento
- 🖥️ A unidade de paginação traduz endereços lineares em endereços físicos de 32 bits
 - ✓ Se não houver paginação o endereço linear é o end. físico
 - ✓ Versões mais recentes usam mais indirecções para suportar endereços físicos maiores



Arquitetura de Memória - Modo Protegido



Arquitetura de Memória - Modo Protegido

 Um segmento é composto por:

- ✓ Uma parte visível (selector do segmento)
 - ✓ Acessível através das instruções mov, pop, lds, lss, ...
- ✓ Uma parte invisível (descriptor do segmento)
 - ✓ Carregada automaticamente aquando do carregamento da parte visível

Visible part	Invisible part	
Segment selector	Segment base address, size, access rights, etc.	CS
Segment selector	Segment base address, size, access rights, etc.	SS
Segment selector	Segment base address, size, access rights, etc.	DS
Segment selector	Segment base address, size, access rights, etc.	ES
Segment selector	Segment base address, size, access rights, etc.	FS
Segment selector	Segment base address, size, access rights, etc.	GS

Arquitectura de Memória - Modo Protegido



Selector de segmento:

- ✓ Índice (index) - indica qual é o segmento que se quer utilizar como base na tradução
- ✓ Indicador de tabela (TI) - indica que tabela de segmentos utilizar
 - ✓ Cada programa tem acesso a duas tabelas: uma local à sua execução e outra global a todos os programas
- ✓ Nível de acesso do pedido (RPL) - indica qual é o nível de acesso do programa que pretende aceder ao segmento
 - ✓ Quanto menor o valor maior o nível de acesso

Arquitectura de Memória - Modo Protegido

Descritor de segmento

- ✓ Endereço base (Base Address) - 32 bits - Contém o endereço inicial do segmento no espaço de endereçamento de 4GB
- ✓ Limite do segmento (Limit) - 20 bits - Indica o tamanho do segmento
 - ✓ Consoante o valor de um bit de granularidade (G) no selector, o limite pode pertencer os intervalos:
 - ✓ [1byte, 2bytes, 3bytes, ..., 1Mbyte]
 - ✓ [4Kbytes, 8Kbytes, 12Kbytes, ..., 4Gbytes]
- ✓ Bits de controlo de acesso (Access Rights) - definem as permissões de acesso e características do segmento

Arquitectura de Memória - Modo Protegido



Tabelas de descritores de segmentos

- ✓ Tabela global (GDT)
 - ✓ Existe apenas uma no sistema
 - ✓ Contém informação gerida pela sistema de operação
 - ✓ Disponível a todos os programas (ou tarefas)
- ✓ Tabela Local (LDT)
 - ✓ Existem várias
 - ✓ Contém descritores de segmentos com dados de programas
- ✓ Ambas têm $2^{13} = 8192$ entradas de 8 bytes
- ✓ Registos indicam onde estão localizadas na memória
 - ✓ Registo base (GDTR ou LDTR) e registo limite (GDTL e LDTL)
- ✓ Tabela de Interrupções (IDT)
 - ✓ Utilizado no processamento de interrupções
 - ✓ A discutir mais tarde

Arquitetura de Memória - Modo Protegido

Modelos de segmentação

✓ Flat

- ✓ A segmentação não é utilizada
 - ✓ Todos os registos de segmentos contêm o mesmo valor, 0, e portanto os segmentos estão sobrepostos
- ✓ Passa a existir apenas um segmento de 4GB
- ✓ Abordagem seguida, por exemplo, pelo Windows

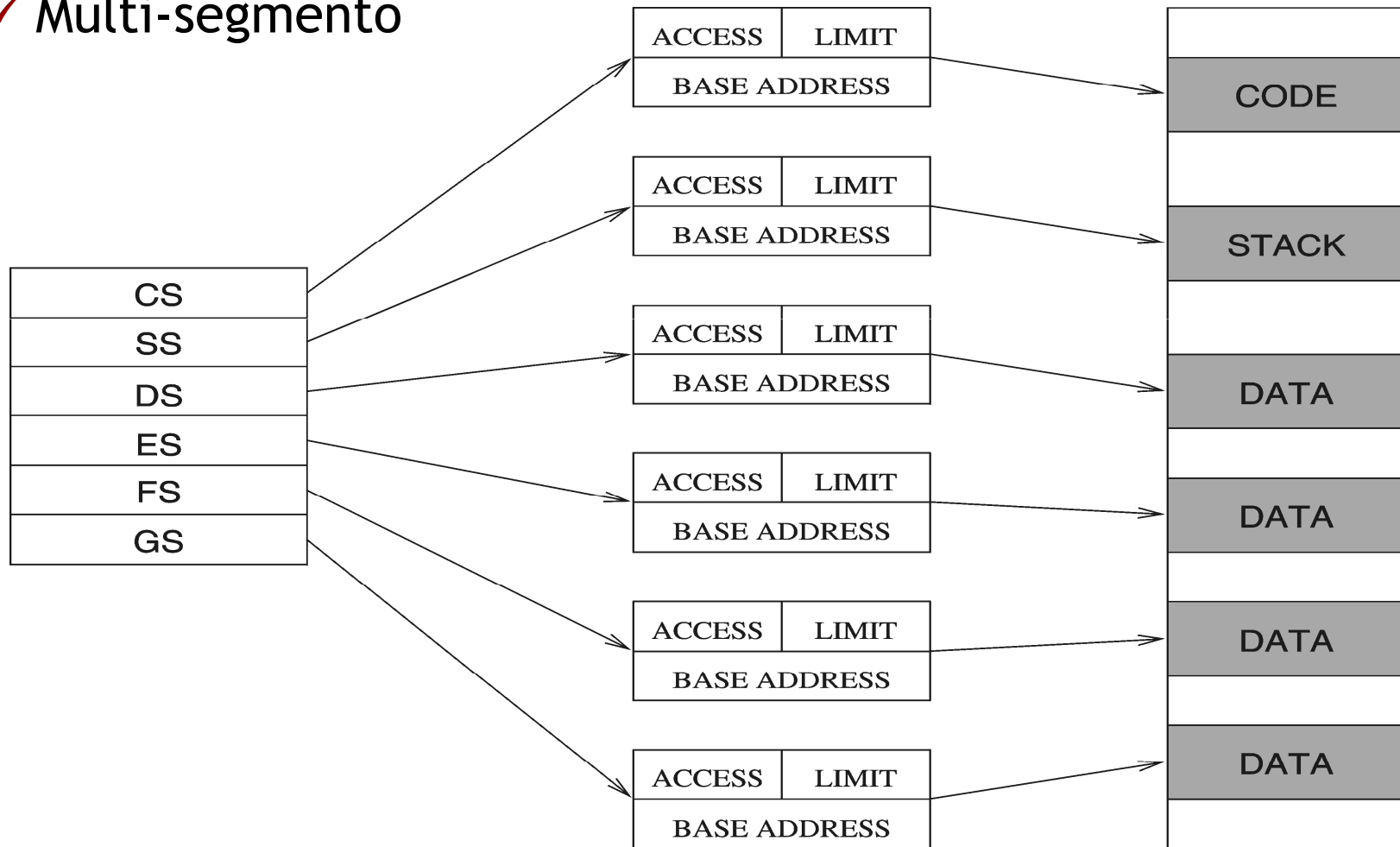
✓ Multi-segmento

- ✓ Cada tarefa pode utilizar os 6 registos de segmentos
- ✓ Caso queira mais do que 6 segmentos estão ficarão na tabela de descritores de segmentos e terão de ser carregados para os registos quando necessário

Arquitetura de Memória - Modo Protegido

Modelos de segmentação

✓ Multi-segmento



Arquitetura de Memória - Modo Protegido



Segmentos utilizados por omissão

- ✓ Fetch

- ✓ Base → CS

- ✓ Deslocamento → IP ou EIP

- ✓ Operações sobre a pilha: push e pop

- ✓ Base → SS

- ✓ Deslocamento → SP ou ESP

- ✓ Operações sobre dados

- ✓ Base → DS

- ✓ Deslocamento → depende do modo de endereçamento



Notação:

- ✓ Base:Deslocamento

- ✓ Exemplo: CS:EIP

Intel64/AMD64 (X86-64)

 Qual é o ISA que temos nas máquinas Intel actuais?

- ✓ O x86-64 que é uma extensão para 64 bits do IA-32
- ✓ Outra?
 - ✓ Além de dados de 8, 16 e 32 bits agora temos de lidar com dados de 64 bits
 - ✓ Os registos EAX, ..., EDX, ESI e EDI têm agora extensões a 64 bits: RAX, ..., RDI
 - ✓ O mesmo é válido para os registos de endereços que são de 64 bits: RBP, RSP, RIP
 - ✓ O registo das flags também foi estendido: RFLAGS
 - ✓ Temos 8 novos registos de dados: r8 a r15

 Que máquinas são a 64 bits?

- ✓ AMD Opteron, Athlon64, Turion64, Phenom, ...
- ✓ Intel Core 2, i3, i5 e i7 e alguns Atom e Pentium

Intel64/AMD64 (X86-64)

