

Administració de memòria: IA-32

Arquitectura IA-32

PROCESSADORS D'AQUEST TIPUS

- 80386
- 80486
- Pentium (PRO, II, III, 4 i totes les sèries derivades).

CARACTERÍSTIQUES COMUNS

- Bus de Dades de 32 bits.
- Bus d'Adreces de 32 bits (menys Petium 4 amb 36 bits).
- Coprocessador Matemàtic (FPU). Intern en tots menys al 80386.
- Poden treballar de 3 formes diferents:
 - Mode Real.
 - Mode Protegit.
 - Mode Virtual 8086.

Noves Instruccions

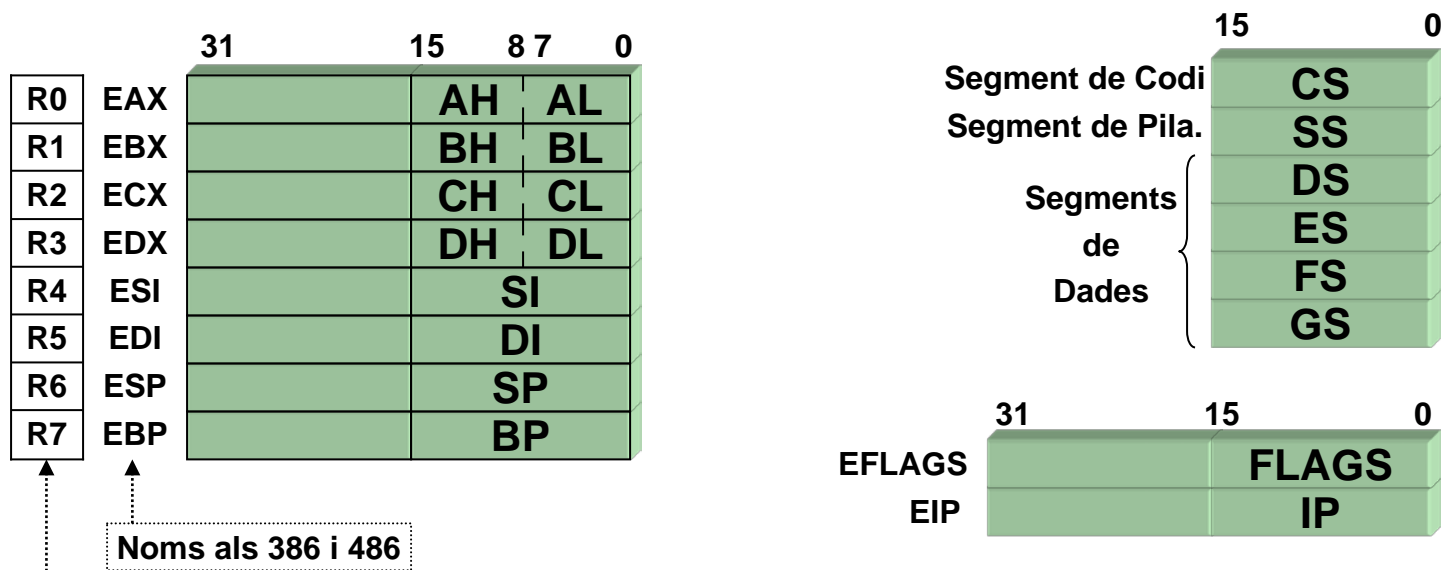
- El 80386 té 128 instruccions (49 més que el 8086).
- Als processadors següents s'incorporen instruccions pròpies de càlcul en coma flotant (ja que incorporen el coprocessador matemàtic, FPU).
- A partir dels Pentiums II, s'inclouen **instruccions MMX**.

Les instruccions MMX (*MultiMedia eXtension*) són un grup d'instruccions aritmètiques i lògiques senzilles que operen en paral·lel sobre un conjunt d'enters (4 de 16 bits o 8 de 8 bits). Estan pensades per operacions molt ràpides de la pantalla (amb els *pixels*).

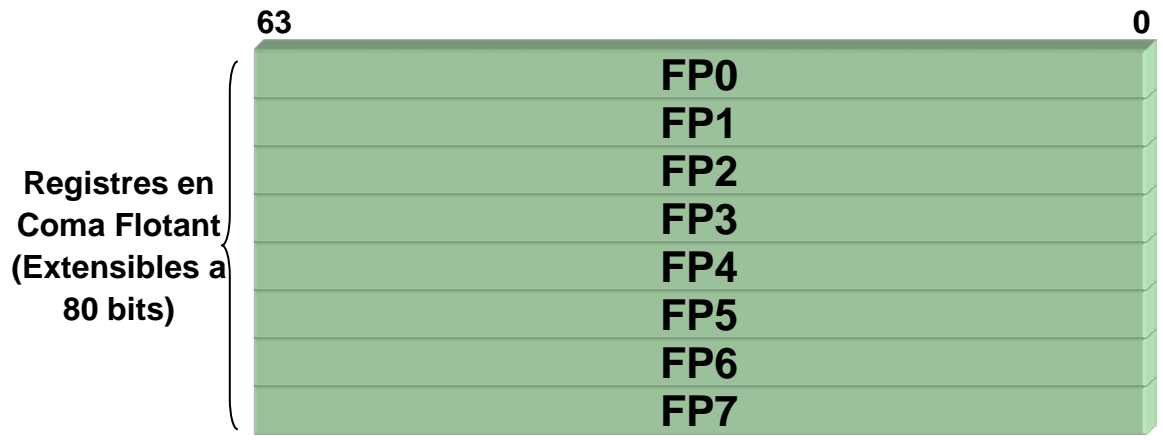
- A partir dels Pentiums III, s'inclouen instruccions tipus SIMD (Intel les anomena (**SSE**)).

Les instruccions SIMD (*Single Instruction Multiple Data*) són un conjunt d'instruccions que actuen en paral·lel sobre varies dades en coma flotant (4 dades de 32 bits al P.III i 2 dades de 64 bits al P.IV (a més)). De fet s'inclouen 8 registres nous de 128 bits, per aquestes instruccions (es diuen registres XMM).

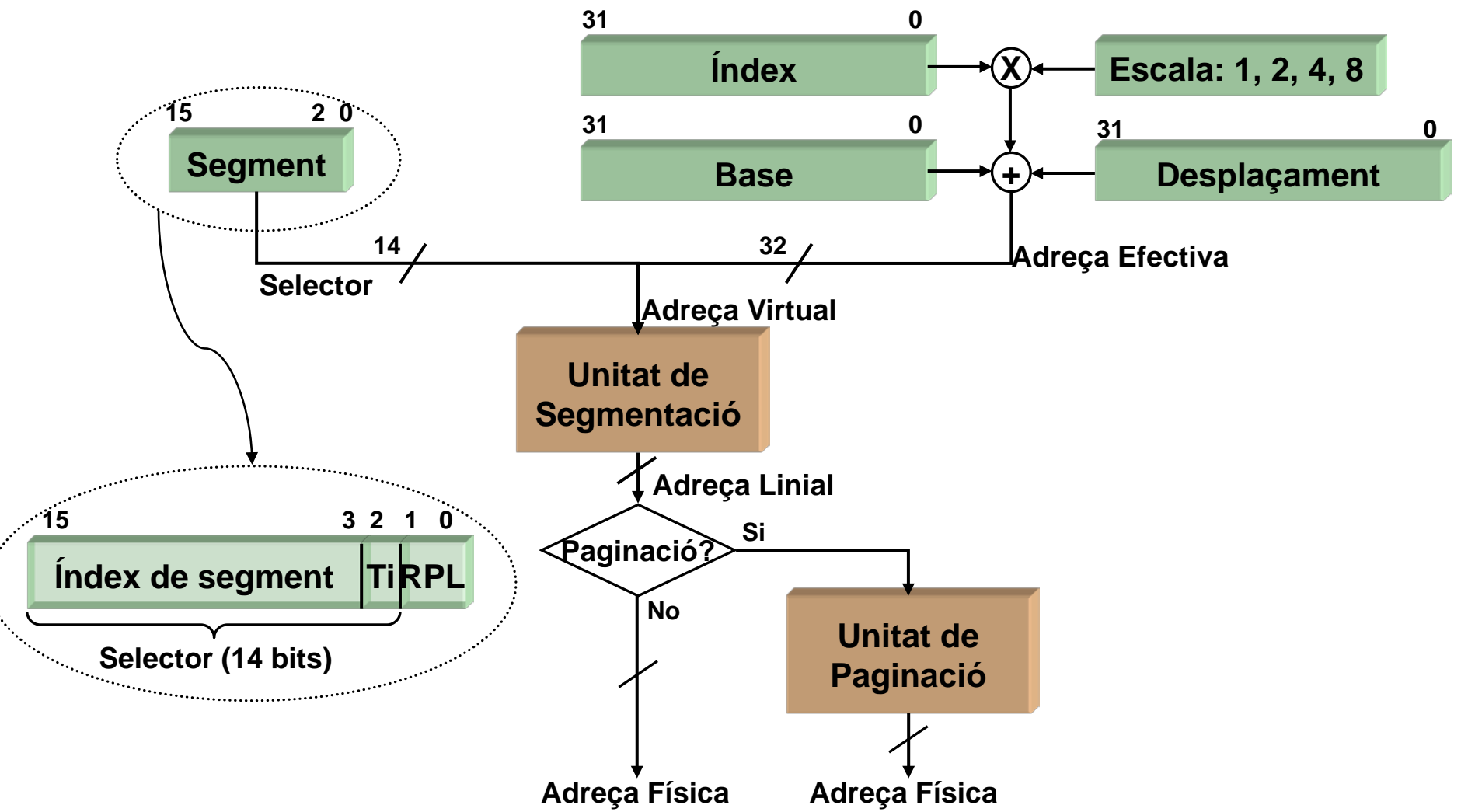
Model de Programació (Conjunt de Registres)



- A més hi ha Registres de
- Control (CR0..CR3)
 - Depuració (DR0..DR7)
 - Base GDT
 - Base LDT
 - Base Vector d'Interrupcions IDTR
 - Base Descriptor de Tasques TR
 - Ombra (hi ha 8)



Generació d'Adreces de Memòria



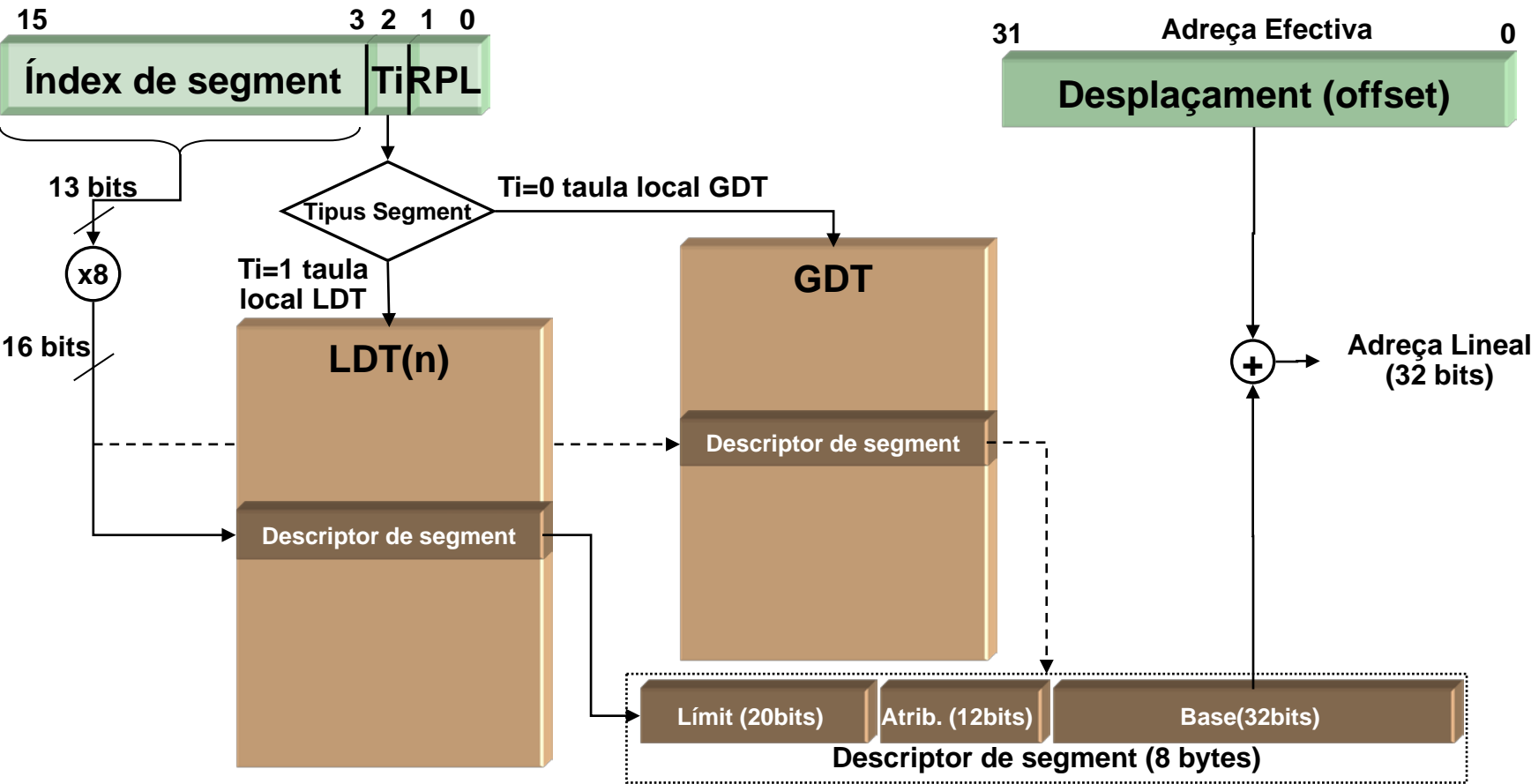
Modes d'Adreçament (Generació de l'Adreça Efectiva: EA)

- Immediat. Operant=Valor que hi ha a la mateixa instrucció (8 o 32 bits amb signe)
- Directe (EA = L'adreça de 32 bits està a la instrucció).
- Registre (EA = **REG**, és a dir Operant està al registre).
- Registre Indirecte (EA = [**REG**]).
- Base + (Índex · Escala) + Desplaçament. Combinats de les següents formes:

1.-	Base amb Desplaçament	$EA=[\text{REG}]+\text{Desp}$
2.-	Índex amb Desplaçament	$EA=[\text{REG}]\cdot\text{Escala}+\text{Desp}$
3.-	Base amb Índex	$EA=[\text{REG1}]+[\text{REG2}]\cdot\text{Escala}$
4.-	Base amb Índex i Desplaçament	$EA=[\text{REG1}]+[\text{REG2}]\cdot\text{Escala}+\text{Desp}$

- **REG**, **REG1** i **REG2**: són qualsevol registre entre EAX, EBX, ECX, EDX, ESP, EBP, ESI, EDI (menys ESP que no pot ser índex).
- Desplaçament (**Desp**): ha de ser un número de 8 o 32 bits amb signe.
- Escala: Es un factor d'escala que pot ser 1, 2, 4 o 8.
- EA: Adreça Efectiva.

MECANISME DE SEGMENTACIÓ



Cada Taula de Descriptors de Segments pot tenir fins 8k entrades (2^{13}) cada una de 8 Bytes. Es a dir, de mida fins a 64kB

$$\text{Adreça Lineal} = \text{Base} + \text{desplaçament} \text{ Si } \text{desplaçament} < \text{Límit}$$

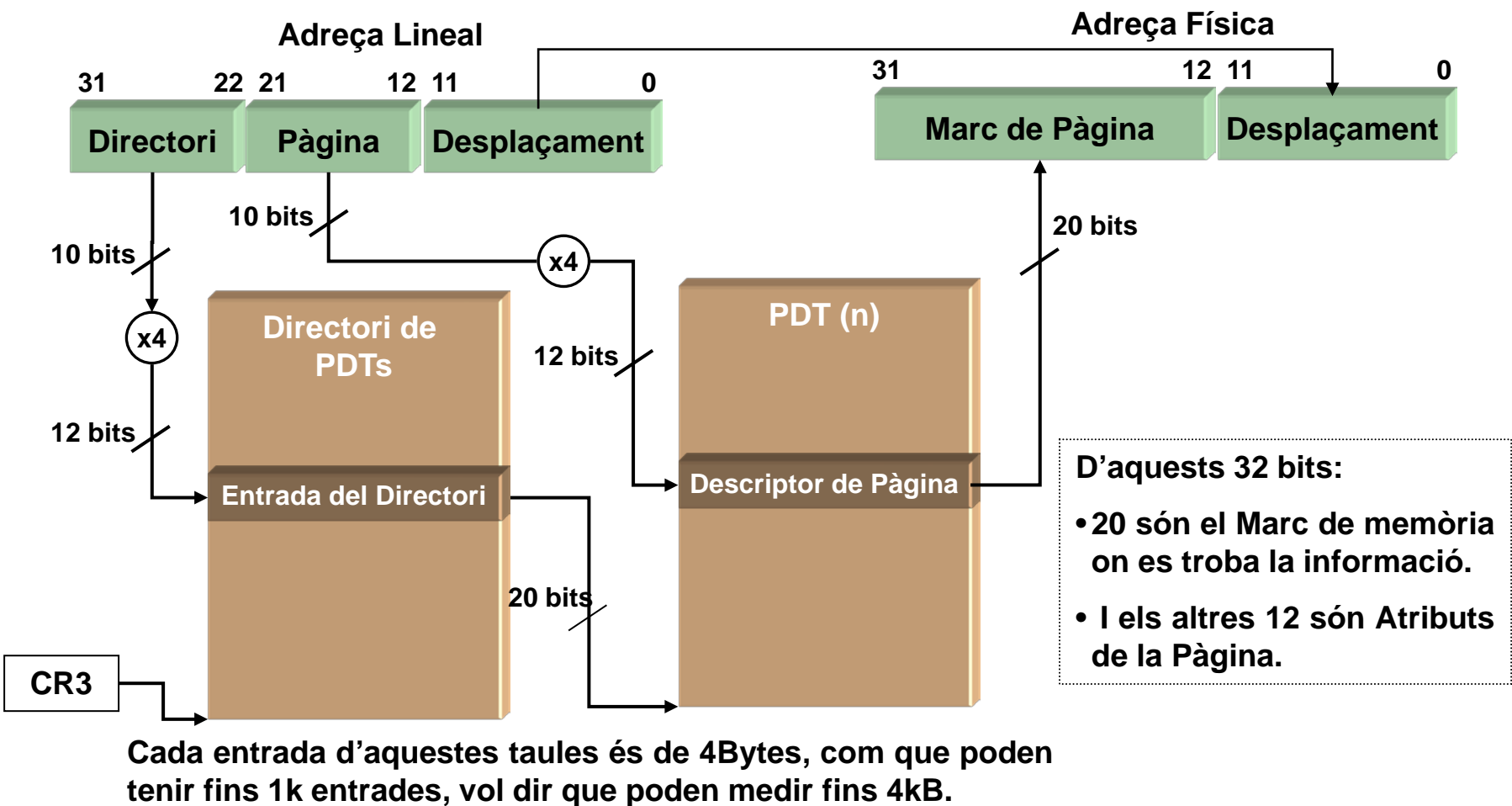
FORMAT DELS DESCRIPTORS DE SEGMENT

Els 8 bytes (64 bits) de cada descriptor de segment, es distribueix així:

- **32 bits que apunten a la base del segment.**
- **20 bits que indiquen el Límit (mida) del segment.** Si no hi ha paginació llavors aquesta és directament la mida en bytes, si hi ha paginació s'ha de multiplicar per 4kB que és la mida de cada pàgina.
- **12 bits de Atributs**, amb el següent significat (1 no es fa servir):
 - **G** (granularitat, 1 bit) **1 indica que SI hi ha paginació, 0 NO hi ha paginació.**
 - **P** (presencia, 1 bit) Indica si el segment es troba a memòria principal o no.
 - **D** (1 bit) Indica si és codi 286 o 386.
 - **DPL** (2 bits) **Nivell de privilegi del Descriptor.**
 - **DT** (1 bit) Segment memòria normal o de sistema.
 - **A** (1 bit) Indica si ja s'ha accedit o no al segment.
 - **Tipus** (3 bits) Tipus de segment (Lectura, L/E, Execució.....)
 - **AVL** (1 bit) Lliure per ser utilitzat pel programador.

MECANISME DE PAGINACIÓ

Un cop tenim l'adreça lineal, fem un procés de paginació per obtenir la adreça física (si G=1).

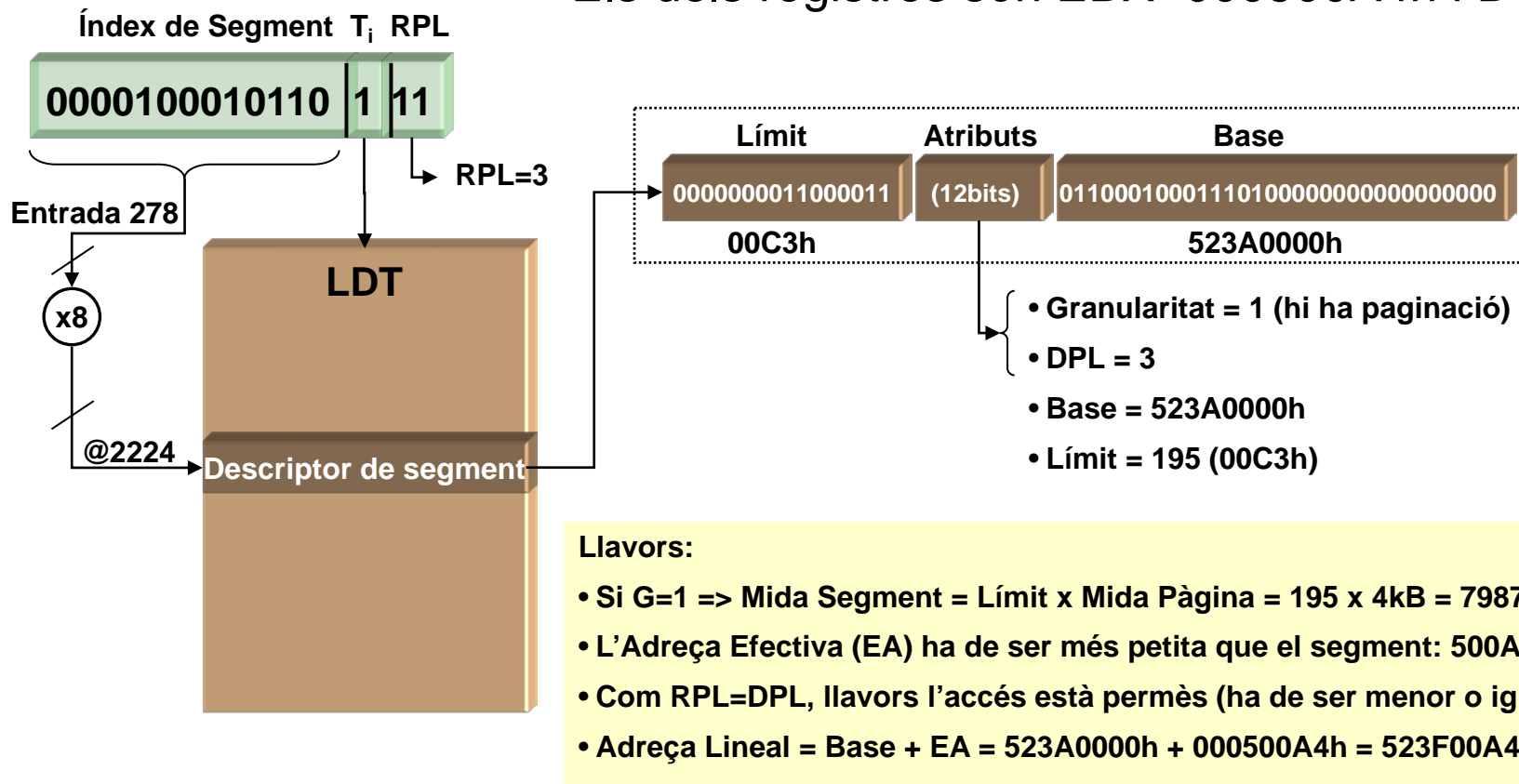


FORMAT DELS DESCRIPTORS DE PÀGINA

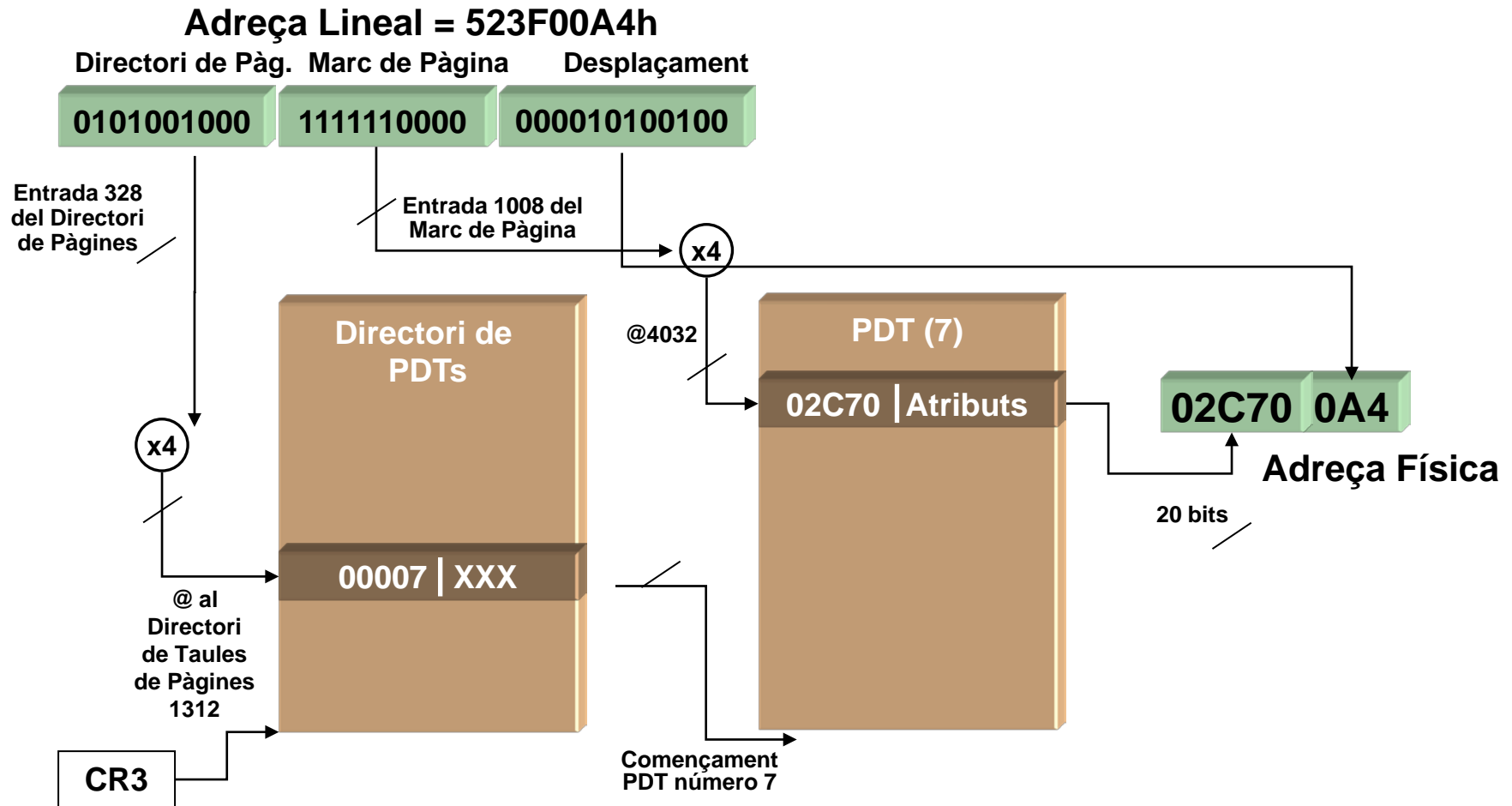
Els 4 bytes (64 bits) de cada descriptor de segment, es distribueix així:

- **20 bits que són els que indiquen el Marc de Pàgina per calcular l'adreça física.**
- **12 bits d'Atributs**, amb el següent significat (4 no es fan servir):
 - **P** (presencia, 1bit) Indica que la pàgina es troba a memòria principal o no.
 - R/W (1 bit) Read/Write.
 - D (Dirty, 1 bit) Dirty bit, la pàgina ha estat modificada a MP.
 - A (Accedit, 1 bit) Indica si ja s'ha accedit a aquesta pàgina.
 - U/S (1 bit) Usuari o Supervisor
 - AVL (3 bits) Lliures per ser utilitzats pel programador.

Exemple de Generació de l'Adreça Física (Segmentació)

Volem executar la instrucció **MOV EAX,[EBX]**Els dels registres són **EBX=000500A4h** i **DS=08B7h**Fins aquí tenim l'adreça lineal, com que hi ha paginació ($G=1$) s'ha de fer aquest procés

Exemple de Generació de l'Adreça Física (Paginació)



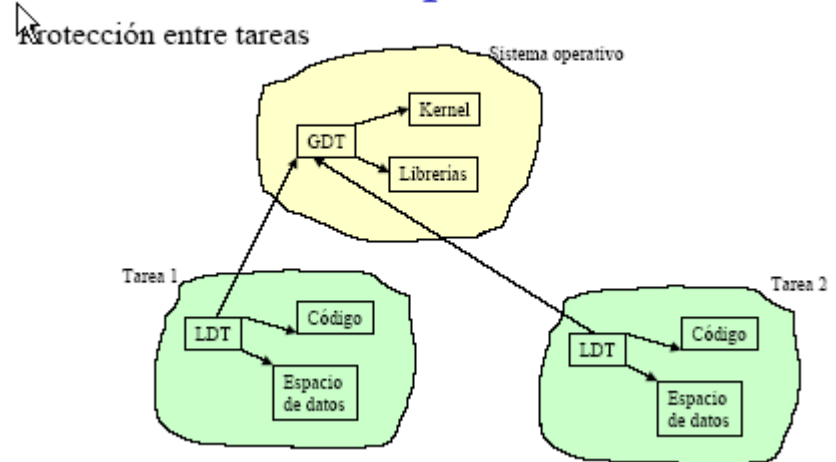
FORMAT DELS DESCRIPTORS DE PÀGINA

Mecanismos de protecció

2 tipus

- Protecció entre tasques:
 - Se assigna a cada tasca un espai de direccions virtual diferent
 - Cada tasca té una taula local de descriptors LDT
 - El sistema operatiu es mapeja en una taula global GDT
- Protecció dins d'una tasca
 - Quatre nivells de privilegi d'accés
 - se restringeix l'accés als dades segons el procés

Mecanismos de protecció



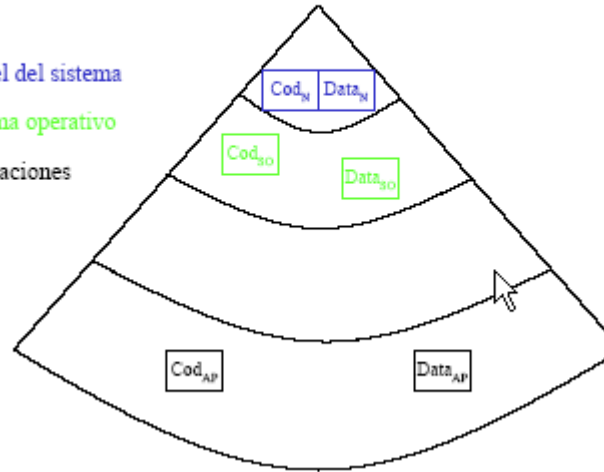
Mecanismos de protecció

Protecció dentro de una tarea

Kernel del sistema

Sistema operativo

Aplicaciones



Nivel 0

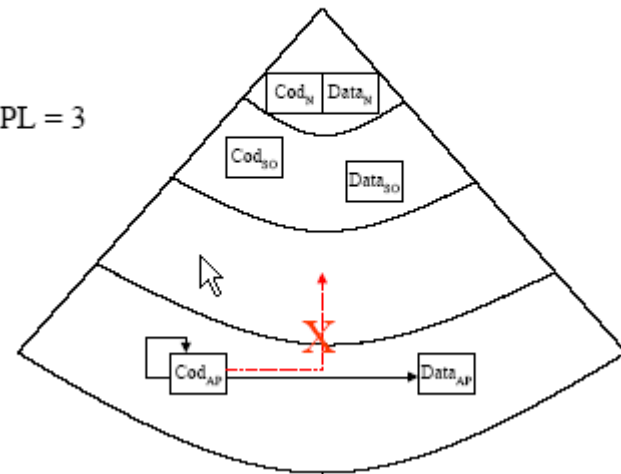
Nivel 1

Nivel 2

Mecanismos de protecció

Protecció dentro de una tarea

CPL = 3



Nivel 0

Nivel 1

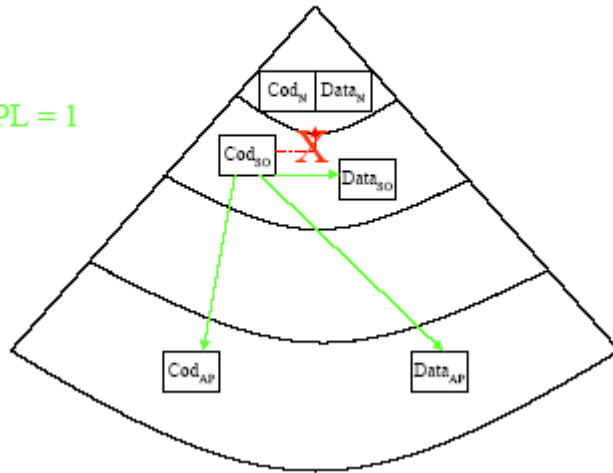
Nivel 2

Nivel 3

Mecanismos de protecció

Protecció dentro de una tarea

CPL = 1



Nivel 0

Nivel 1

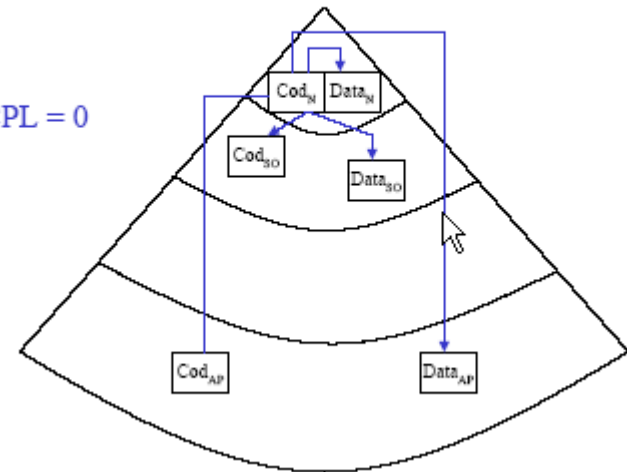
Nivel 2

Nivel 3

Mecanismos de protecció

Protecció dentro de una tarea

CPL = 0



Nivel 0

Nivel 1

Nivel 2

Nivel 3

Selección de niveles de privilegio

3 indicadores:

- **CPL** : Current privilege level
Esta en el campo RPL del selector de segmento CS
- **RPL** : Request privilege level
Esta en todos los selectores de segmento
- **DPL** : Data privilege level
En el campo de atributos de cada descriptor de segmento

Selección de niveles de privilegio

En todo acceso a un segmento se verifica:

si $CPL < DPL \Rightarrow$ acceso permitido
sino error

(si el privilegio de ejecución actual es mayor que el
del segmento accedido)

si $RPL \neq CPL \Rightarrow$ nuevo $CPL = \max(CPL, RPL)$

(si el privilegio de ejecución requerido es menor que el actual
entonces el CPL se debilita para adaptarse al nuevo privilegio)