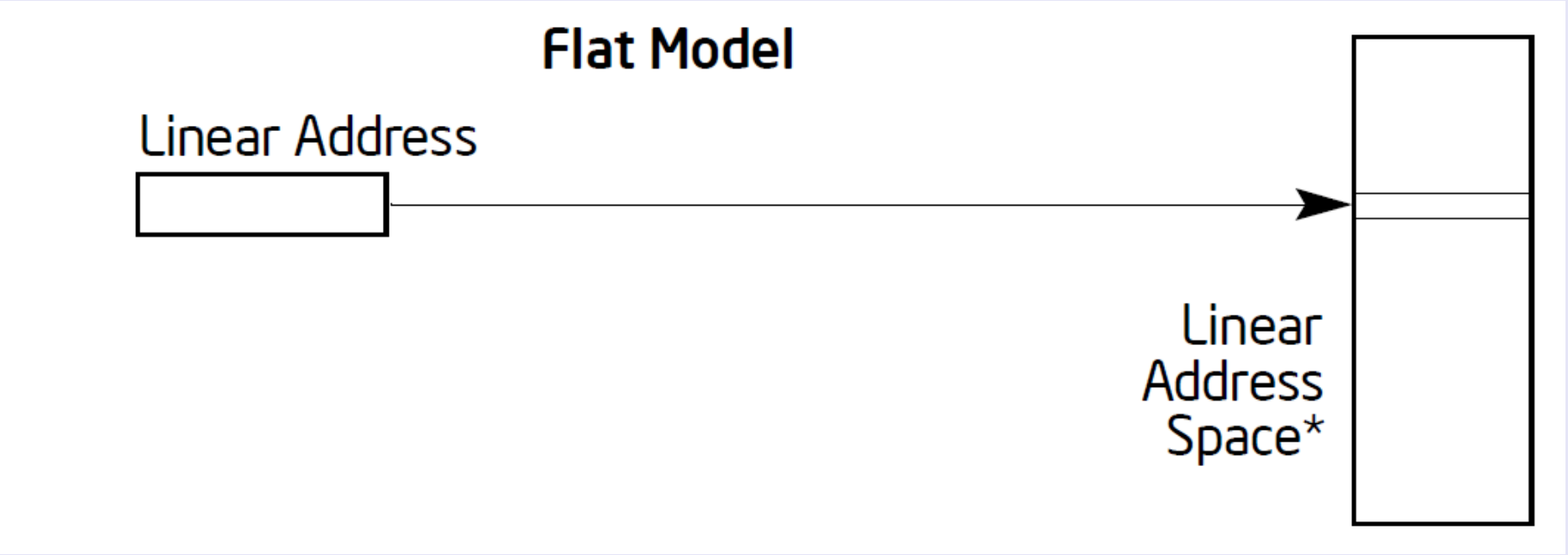
Архитектура ЭВМ и основы ОС

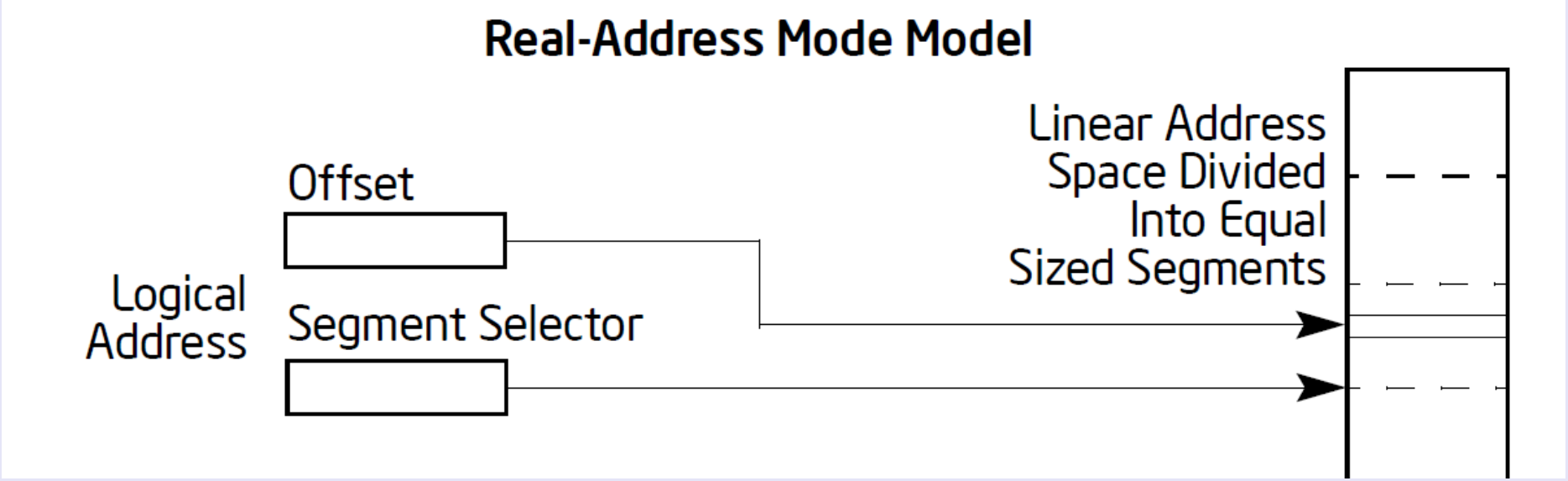
Управление памятью в процессорах Intel

Flat

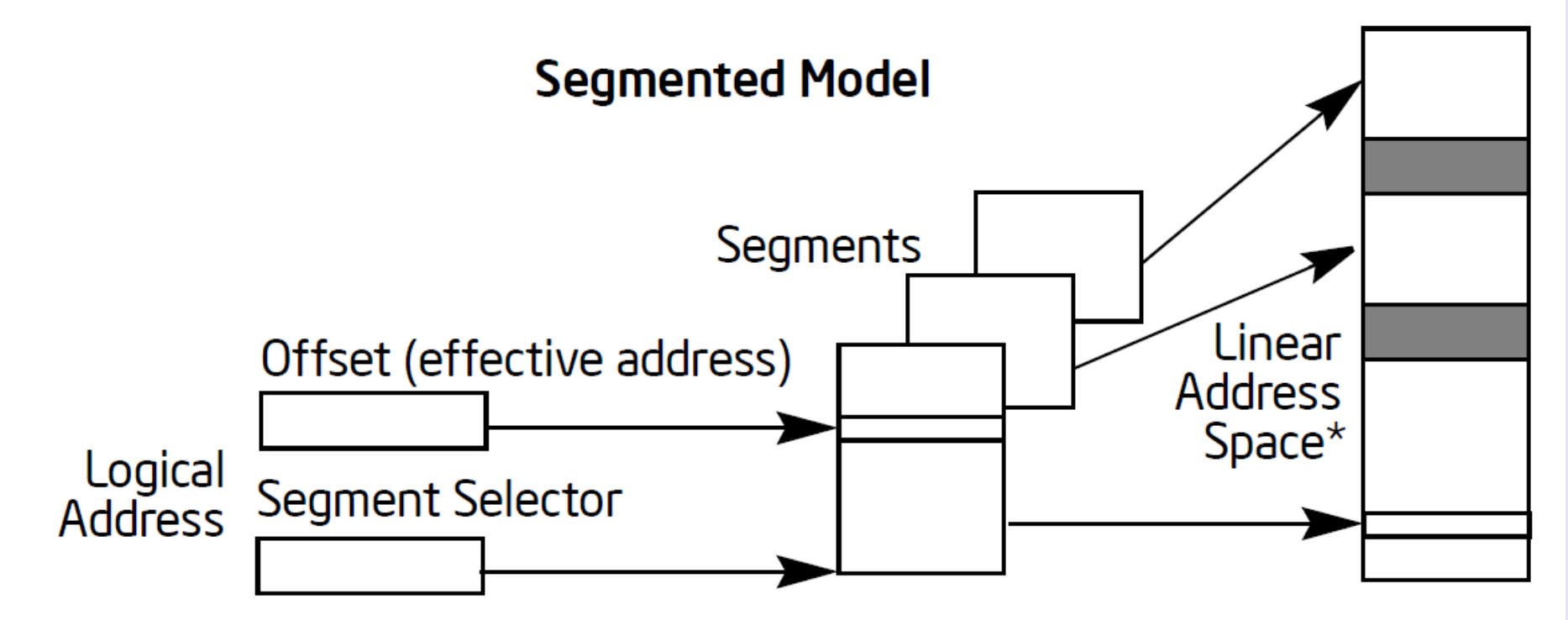


*) использованы материалы Intel

Real


Real-Address Mode Model
Linear Address
Space Divided
Into Equal
Sized Segments
Offset
Logical
Address
Segment Selector

Segmented


Segmented Model
Segments
Offset (effective address)
Logical
Address
Segment Selector
Linear
Address
Space*

Адреса

Логический адрес – тот адрес, которым оперирует процесс (Селектор:смещение)

Линейный адрес – получается из (g/l)dt, не является физическим адресом.

Например

(Таблица10:страница10:смещение12)

Физический адрес: адрес на шине памяти

Регистры управления памятью

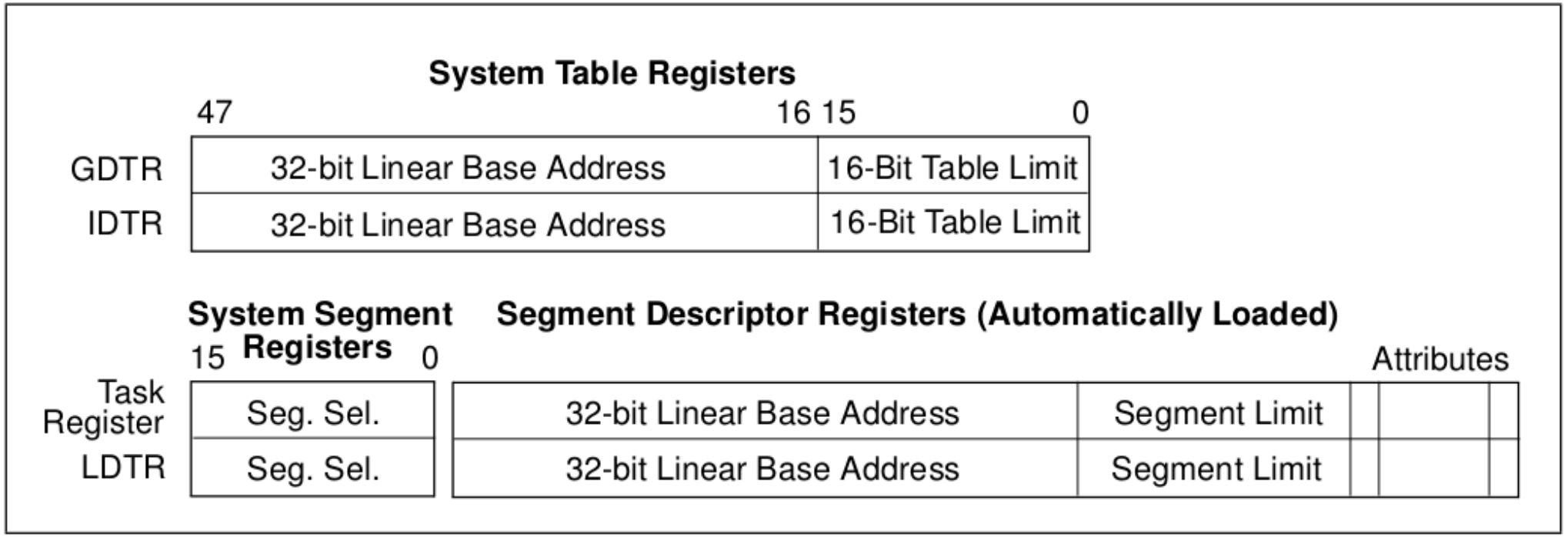


Figure 2-4. Memory Management Registers

Специальные регистры

- Cr0 – Contains system control flags that control operating mode and states of the processor

- CR1 – res
- CR2 – Contains the page-fault linear address (the linear address that caused a page fault)
- CR3 – Contains the physical address of the base of the page directory and two flags (PCD and PWT)
    - PCD – Page-level Cache Disable
    - PWT – Page-level Writes Transparent (L1,L2)
- CR4 – Contains a group of flags that enable several architectural extensions (SIMD)

Специальные регистры

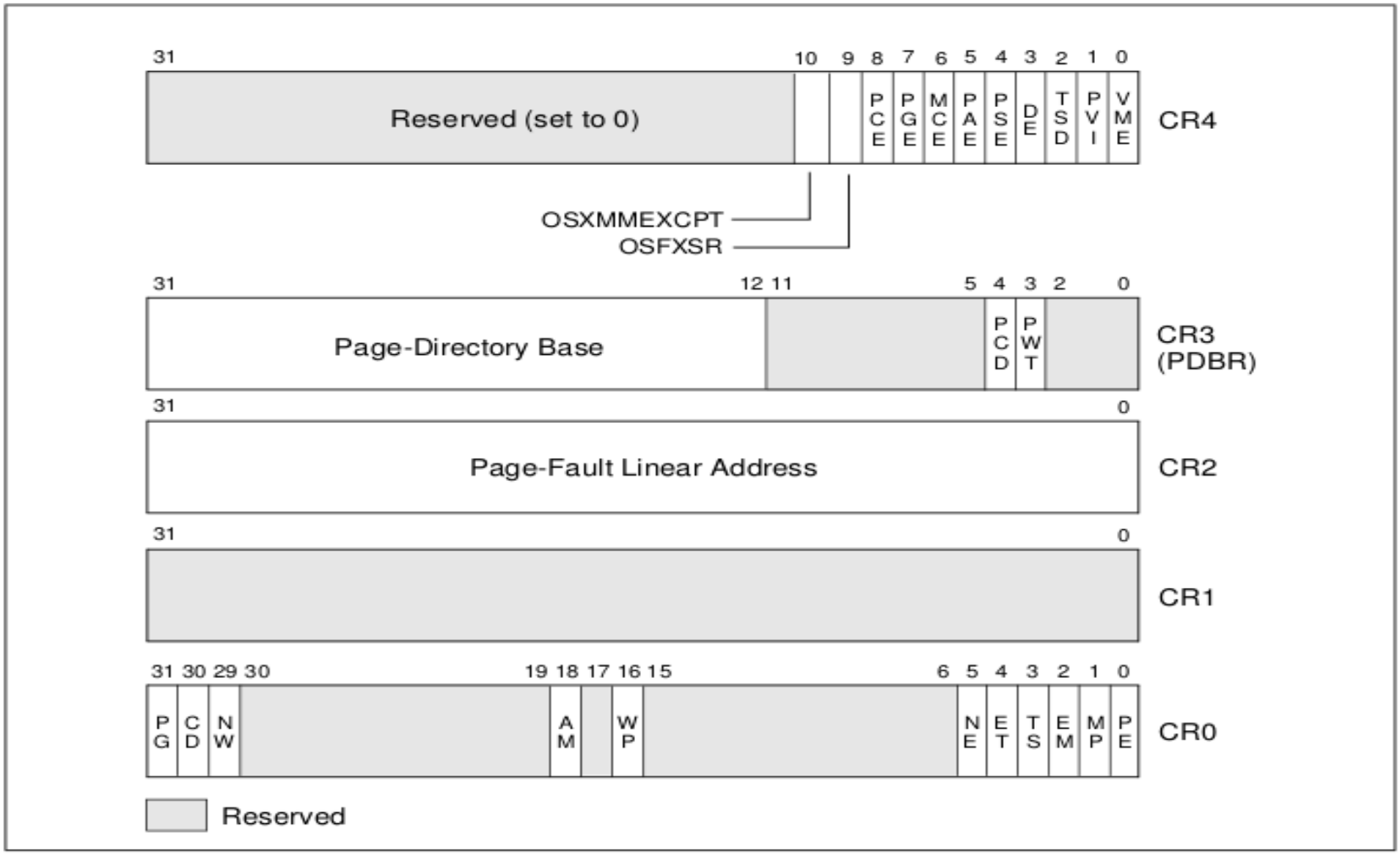
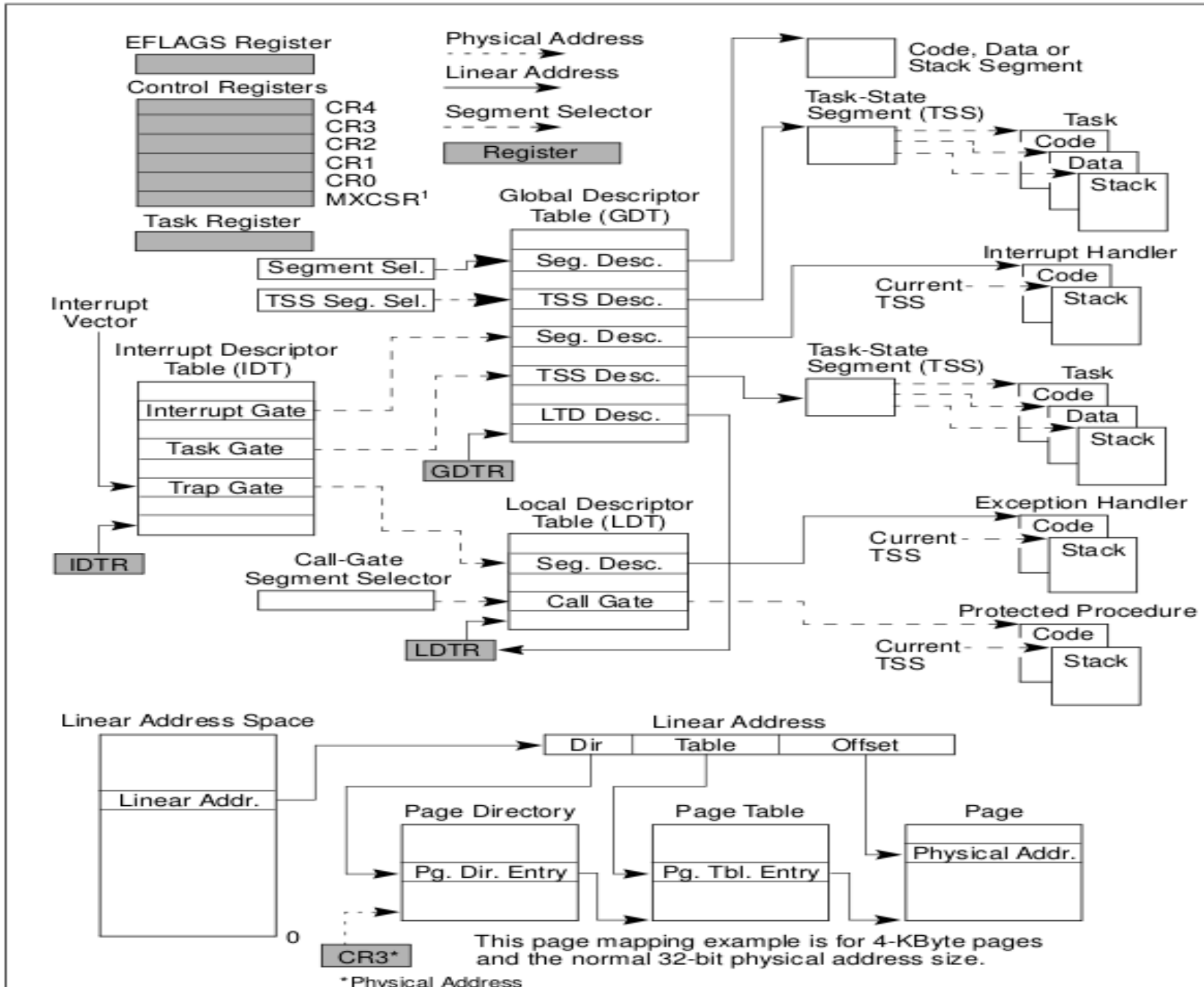
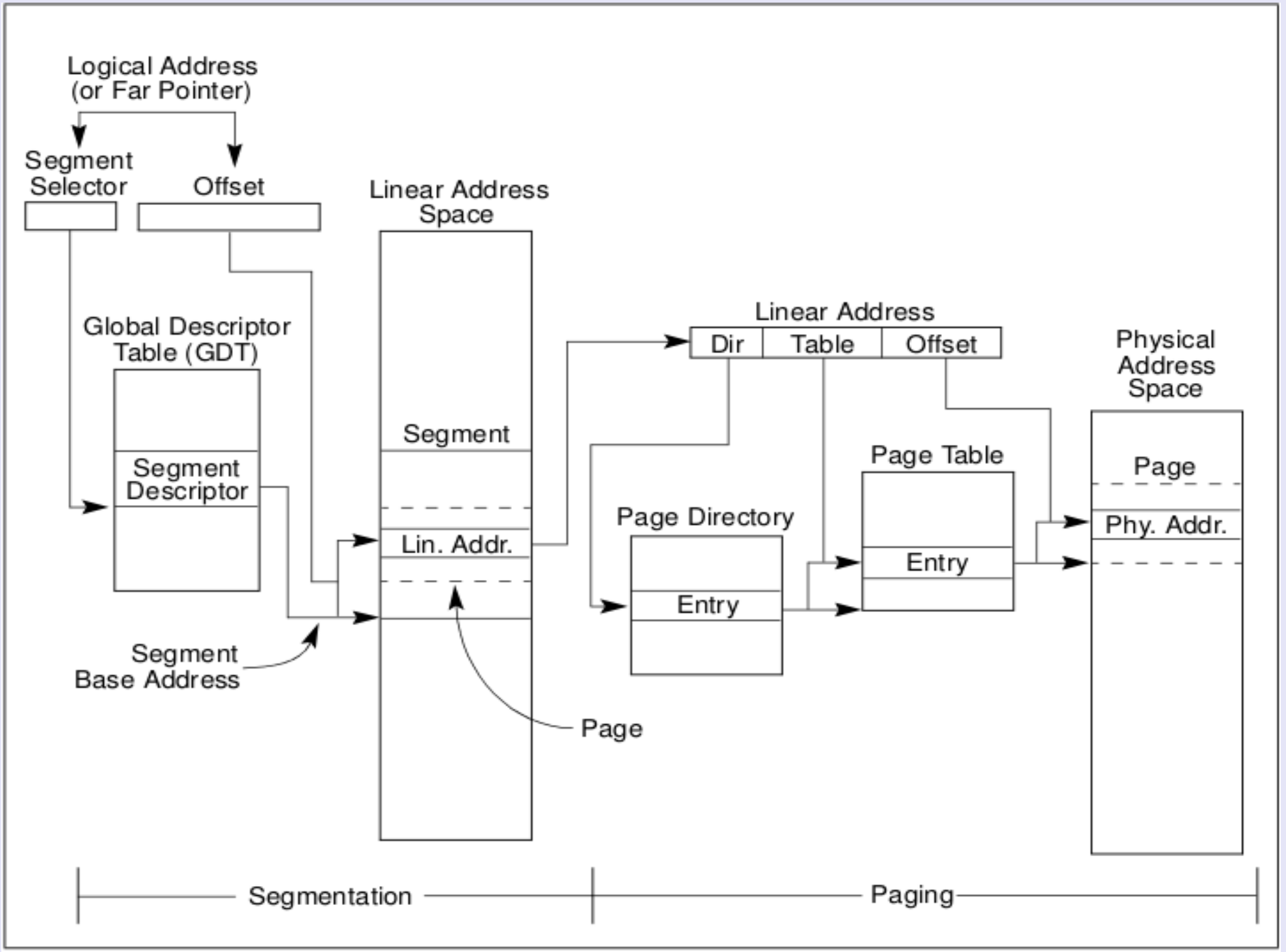


Figure 2-5. Control Registers


EFLAGS Register
Control Registers
CR4
CR3
CR2
CR1
CR0
MXCSR¹
Task Register
Physical Address
Linear Address
Segment Selector
Register
Code, Data or Stack Segment
Task-State Segment (TSS)
Task
Code
Data
Stack
Global Descriptor Table (GDT)
Seg. Desc.
TSS Desc.
Seg. Desc.
TSS Desc.
LTD Desc.
Segment Sel.
TSS Seg. Sel.
Interrupt Handler
Code
Stack
Current TSS
Interrupt Vector
Interrupt Descriptor Table (IDT)
Interrupt Gate
Task Gate
Trap Gate
GDTR
IDTR
Task-State Segment (TSS)
Task
Code
Data
Stack
Local Descriptor Table (LDT)
Seg. Desc.
Call Gate
Call-Gate Segment Selector
LDTR
Exception Handler
Code
Stack
Current TSS
Protected Procedure
Code
Stack
Current TSS
Linear Address Space
Linear Addr.
0
Linear Address
Dir
Table
Offset
Page Directory
Pg. Dir. Entry
Page Table
Pg. Tbl. Entry
Page
Physical Addr.
CR3*
This page mapping example is for 4-KByte pages and the normal 32-bit physical address size.
*Physical Address


Logical Address
(or Far Pointer)
Segment
Selector
Offset
Linear Address
Space
Global Descriptor
Table (GDT)
Segment
Descriptor
Segment
Lin. Addr.
Linear Address
Dir
Table
Offset
Physical
Address
Space
Page Table
Page
Page Directory
Phy. Addr.
Entry
Entry
Segment
Base Address
Page
Segmentation
Paging

Segmentation

a mechanism of isolating individual code, data, and stack modules so that multiple programs (or tasks) can run on the same processor without interfering with one another.

Mandatory

Paging

Paging provides a mechanism for implementing a conventional demand-paged, virtual-memory system where sections of a program's execution environment are mapped into physical memory as needed. Paging can also be used to provide isolation between multiple tasks

Optional

A couple important words...

If paging is not used, the linear address space of the processor is mapped

directly into the physical address space

of processor. The physical address space is defined as the range of addresses that the processor can generate on its address bus.

Flat model

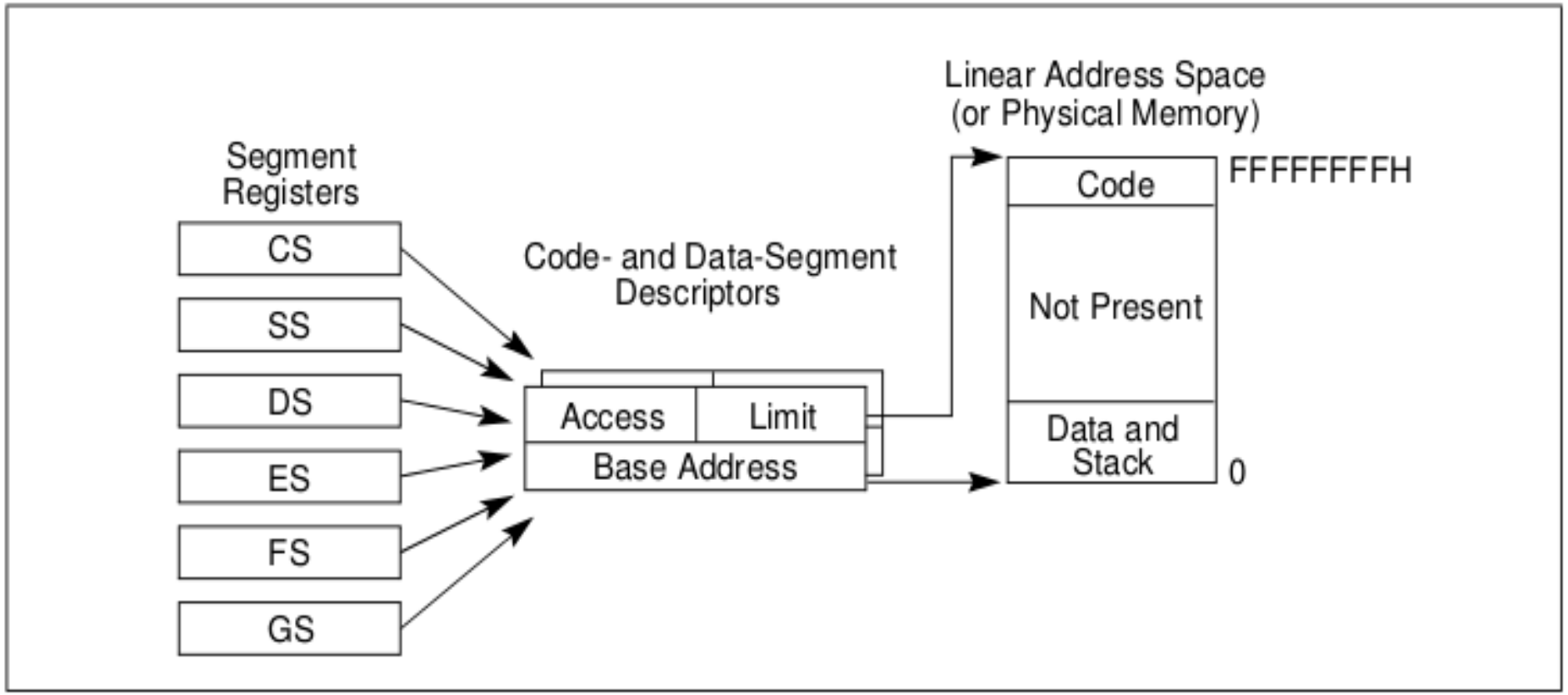


Figure 3-2. Flat Model

Protected Flat model

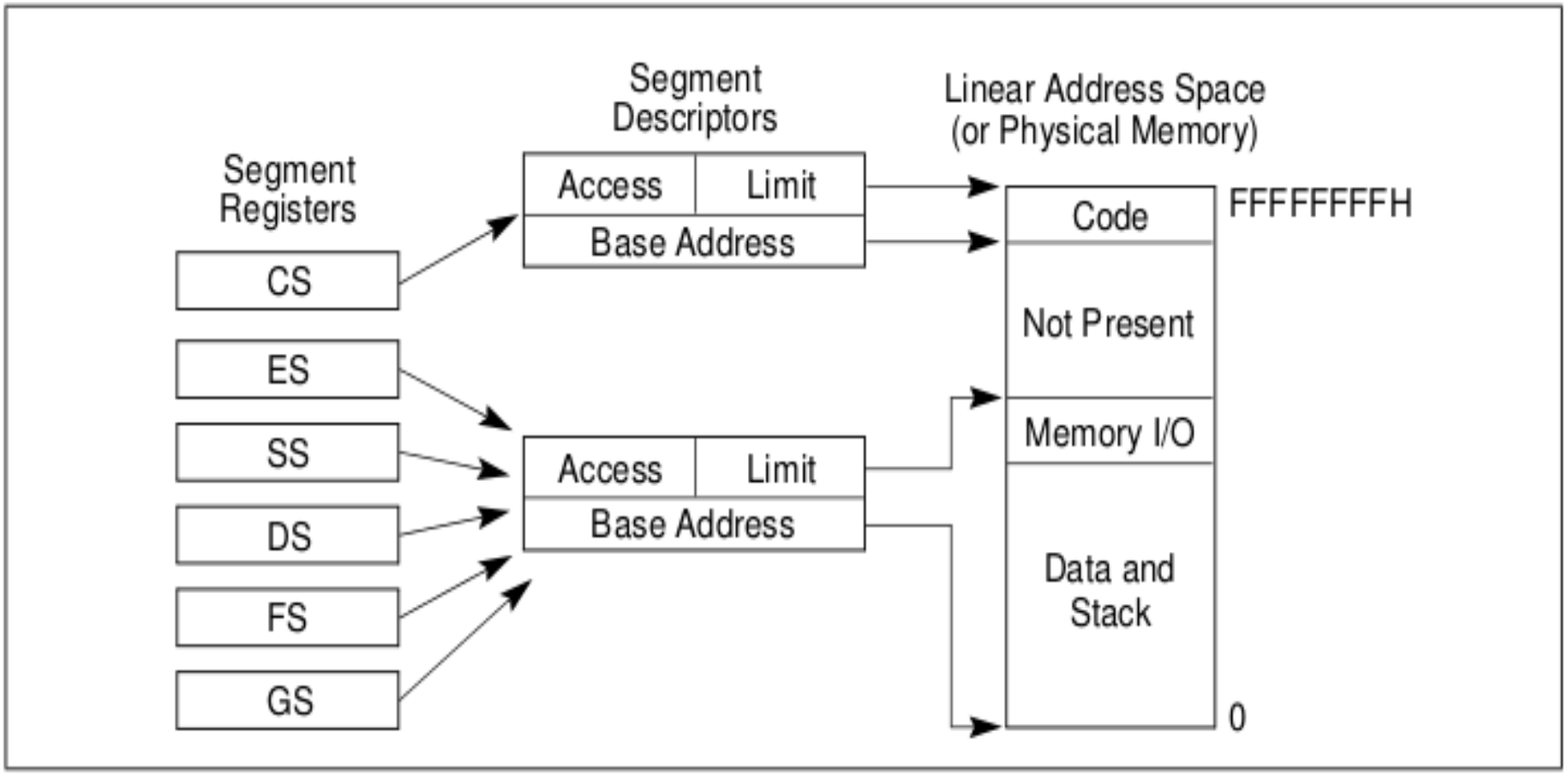


Figure 3-3. Protected Flat Model

Segment selector

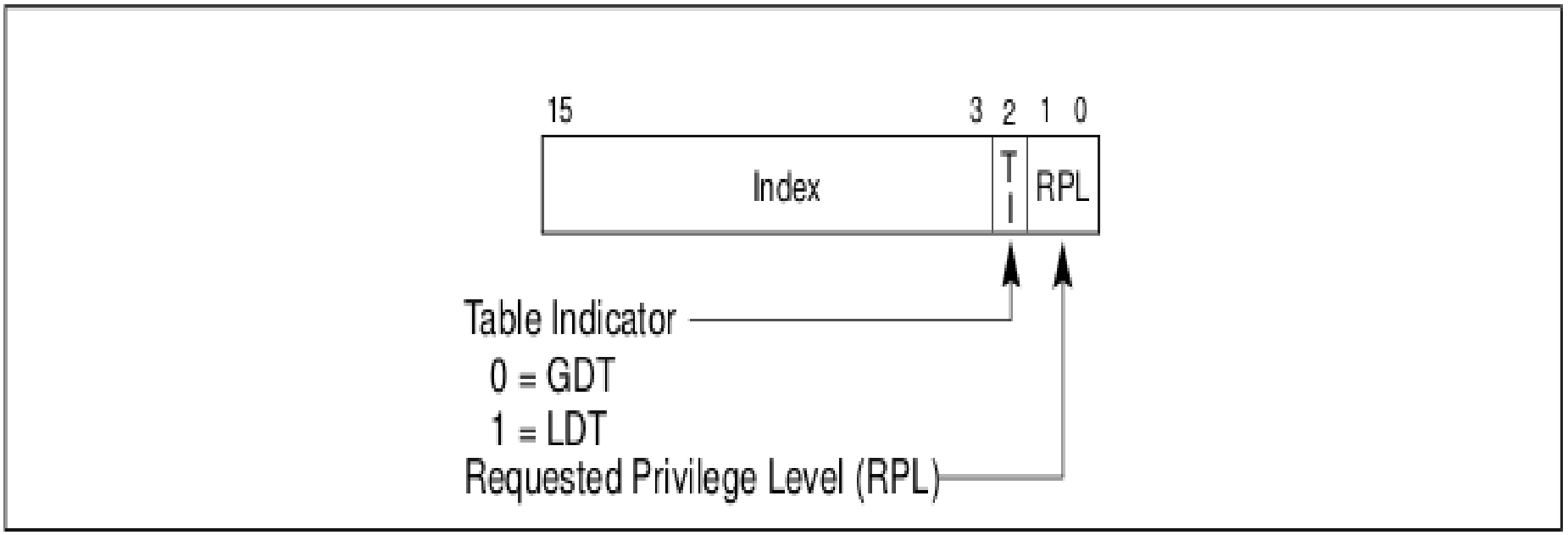
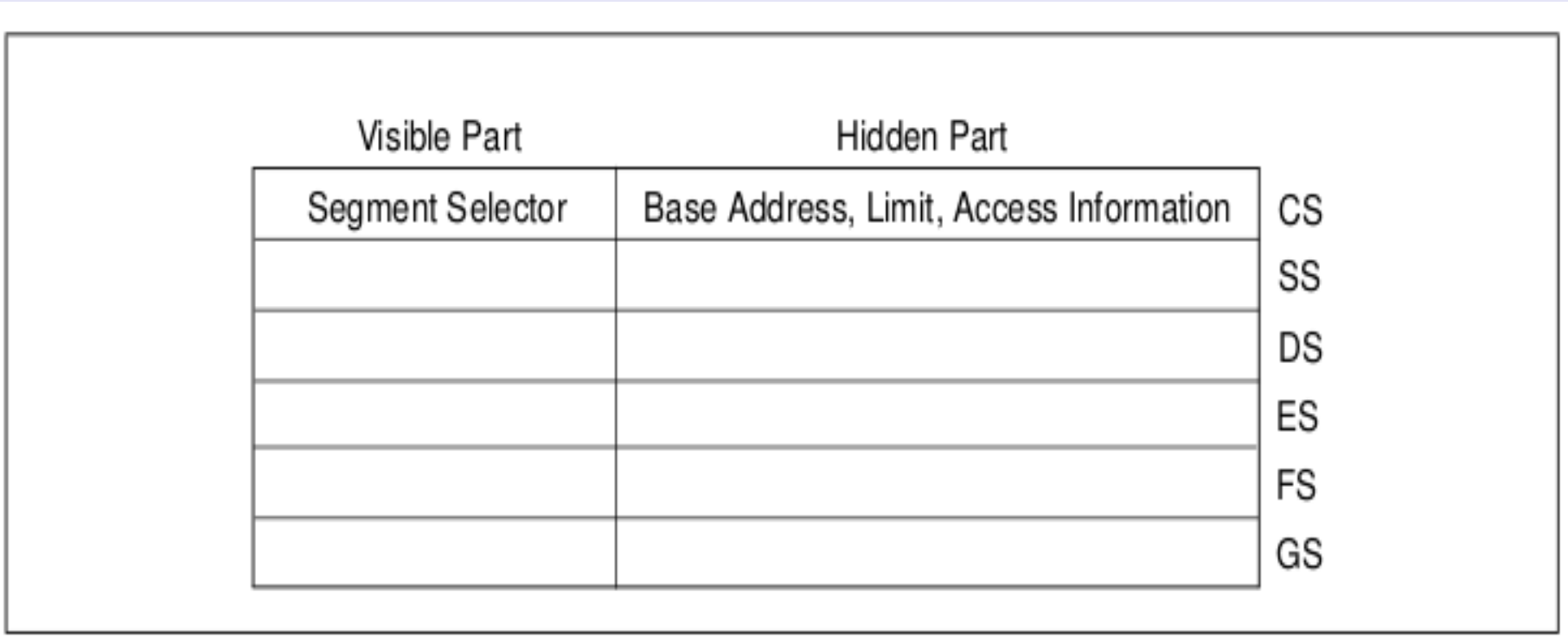


Figure 3-6. Segment Selector

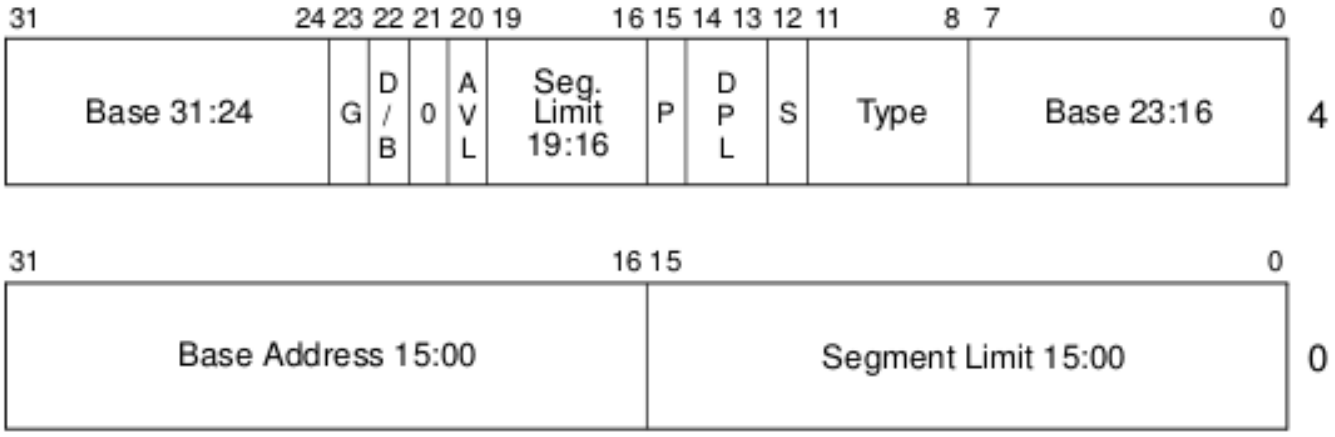
Немного магии...



Visible Part	Hidden Part	
Segment Selector	Base Address, Limit, Access Information	CS
		SS
		DS
		ES
		FS
		GS

Figure 3-7. Segment Registers

Дескриптор сегмента



AVL — Available for use by system software
BASE — Segment base address
D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
DPL — Descriptor privilege level
G — Granularity
LIMIT — Segment Limit
P — Segment present
S — Descriptor type (0 = system; 1 = code or data)
TYPE — Segment type

Figure 3-8. Segment Descriptor

Paging options

Paging is controlled by three flags in the processor's control registers:

- PG (paging) flag, bit 31 of CR0 (beginning with the x386 processor).
- PSE (page size extensions) flag, bit 4 of CR4 (4-MByte pages or 2-MByte).
- PAE (physical address extension) flag, bit 5 of CR4 (It relies on page directories and page tables to reference physical addresses above FFFFFFFFH).

Флажки и структура памяти

Table 3-3. Page Sizes and Physical Address Sizes

PG Flag, CR0	PAE Flag, CR4	PSE Flag, CR4	PS Flag, PDE	Page Size	Physical Address Size
0	X	X	X	—	Paging Disabled
1	0	0	X	4 KBytes	32 Bits
1	0	1	0	4 KBytes	32 Bits
1	0	1	1	4 MBytes	32 Bits
1	1	X	0	4 KBytes	36 Bits
1	1	X	1	2 MBytes	36 Bits

Трансляция линейного адреса страницы 4Kb

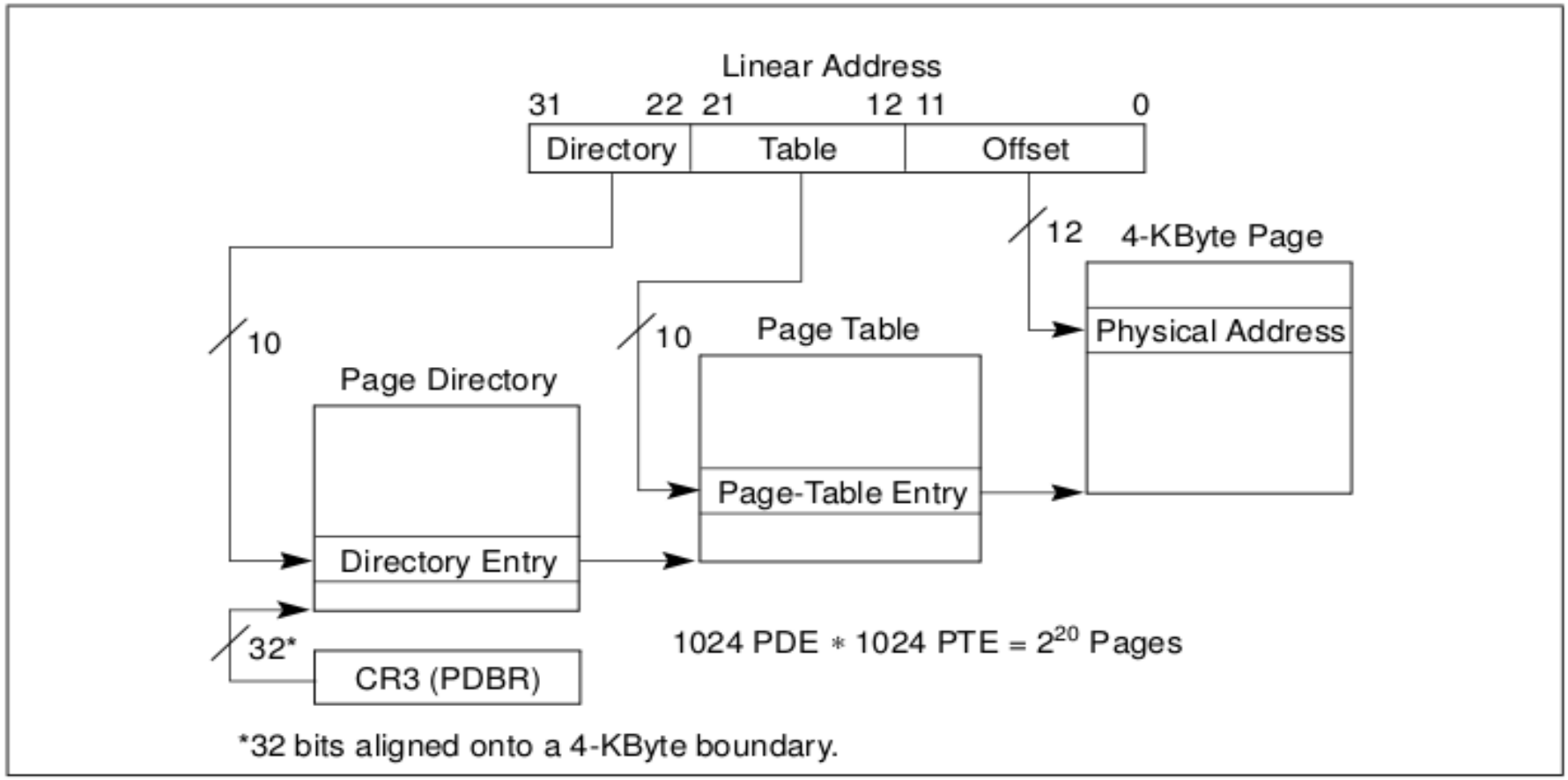


Figure 3-12. Linear Address Translation (4-KByte Pages)

Трансляция линейного адреса страницы 4Mb

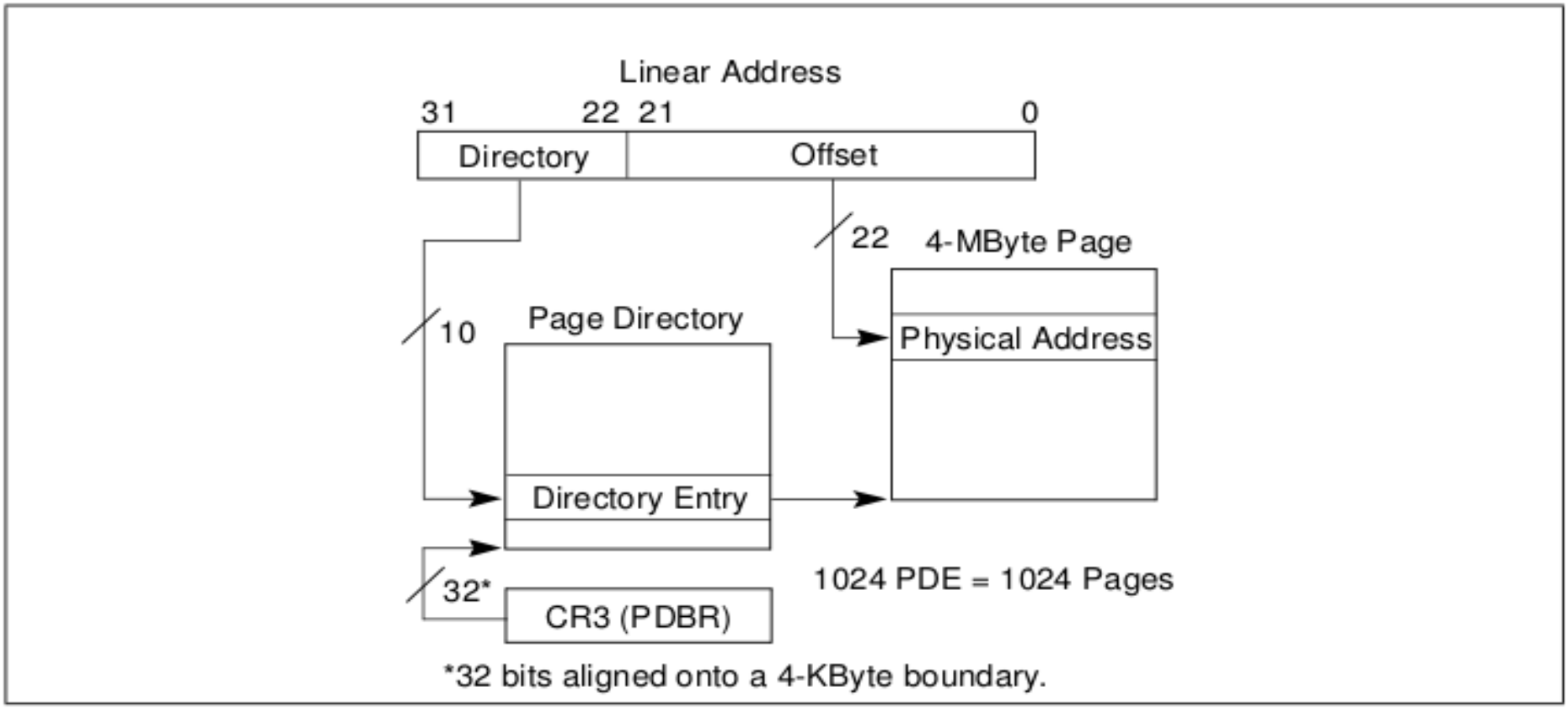


Figure 3-13. Linear Address Translation (4-MByte Pages)

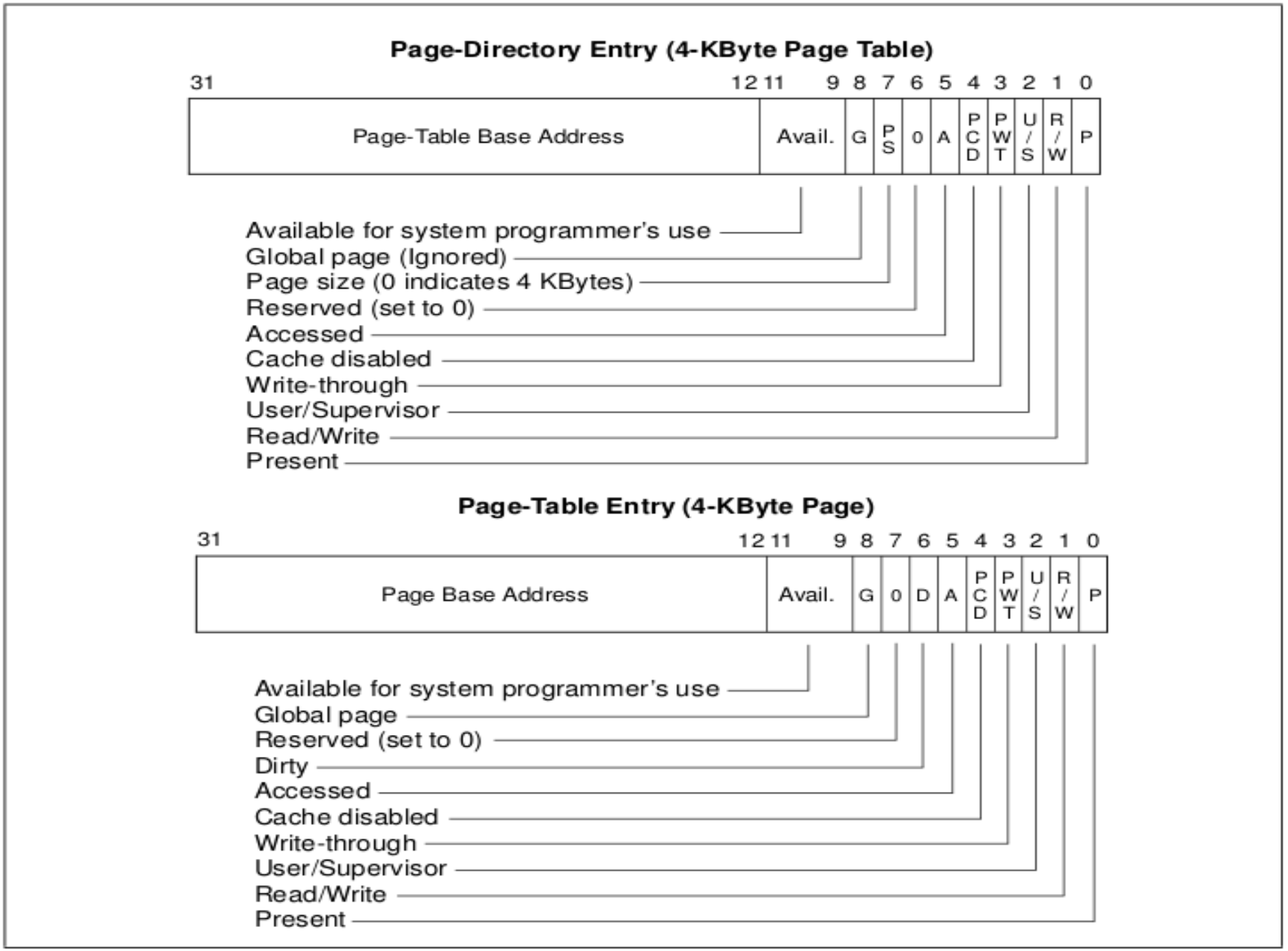
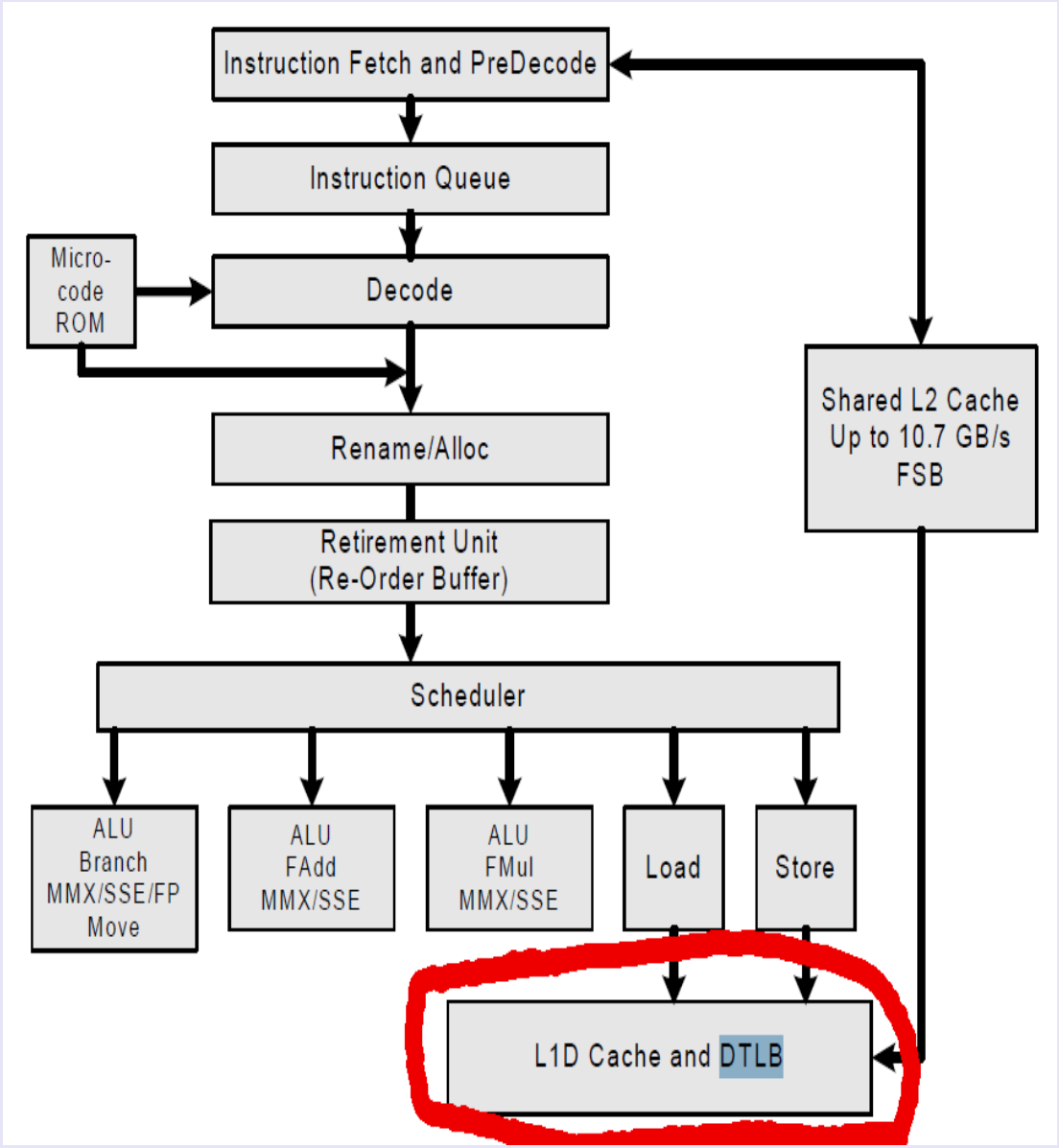


Figure 3-14. Format of Page-Directory and Page-Table Entries for 4-KByte Pages and 32-Bit Physical Addresses

Translation Lookaside Buffer (TLB)

{страничный кэш}



page addr → ph.addr

очистка при смене контекста памяти

возможно кэширование TLB

многоуровневые TLB