LOKNETE DR. BALASAHEB VIKHE PATIL
(PADMA BHUSHAN AWARDEE)
PRAVARA RURAL EDUCATION SOCIETY'S
SIR VISVESVARAYA INSTITUTE OF TECHNOLOGY
PRAVARA TECHNICAL EDUCATION CAMPUS NASHIK
NASHIK

- SE Computer (22-23)

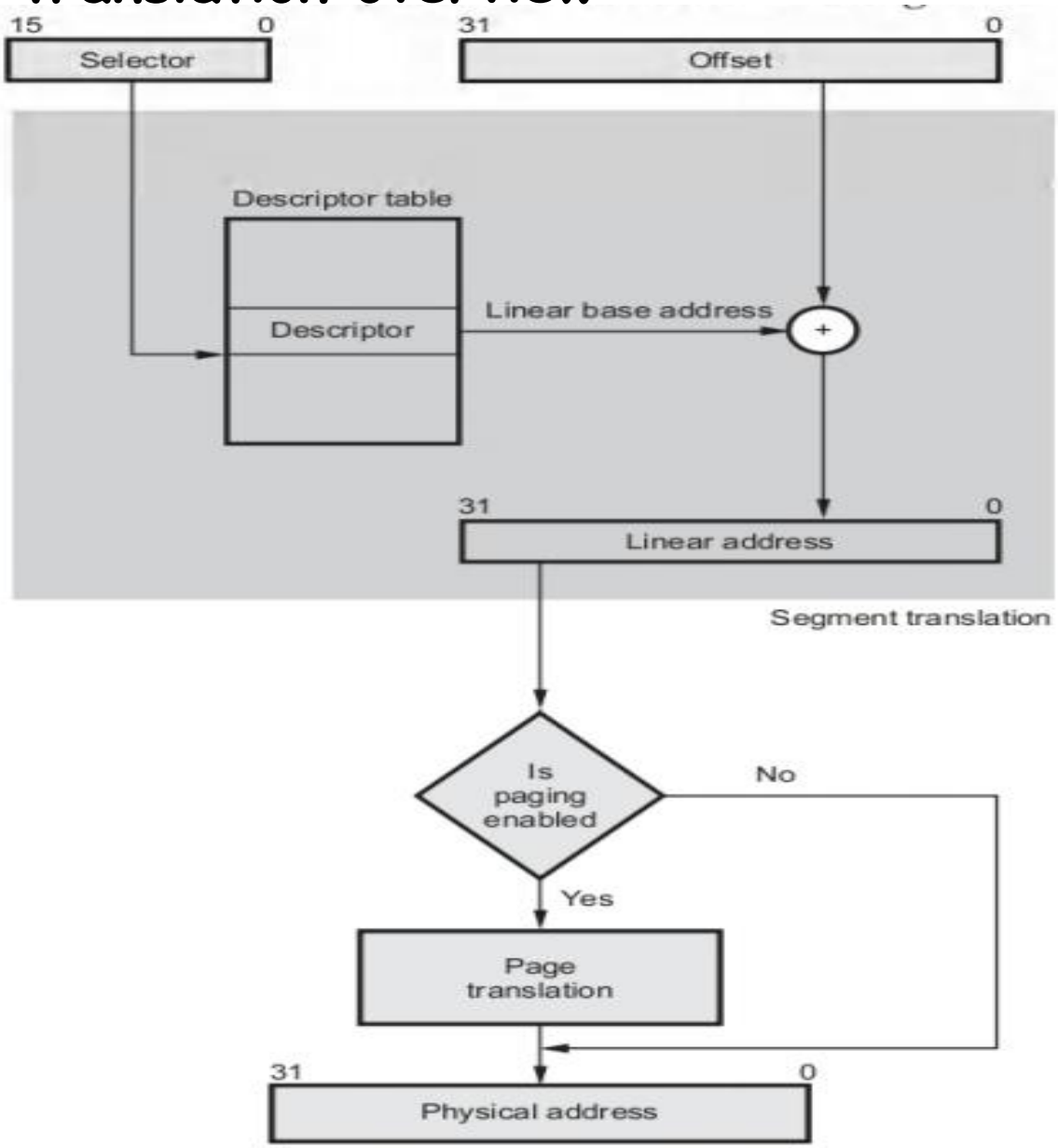
Microprocessor

(Course Code: 210254)

Unit III-Memory Management

Subject In-charge -Dr. Prerana N.Khairnar

Address translation overview



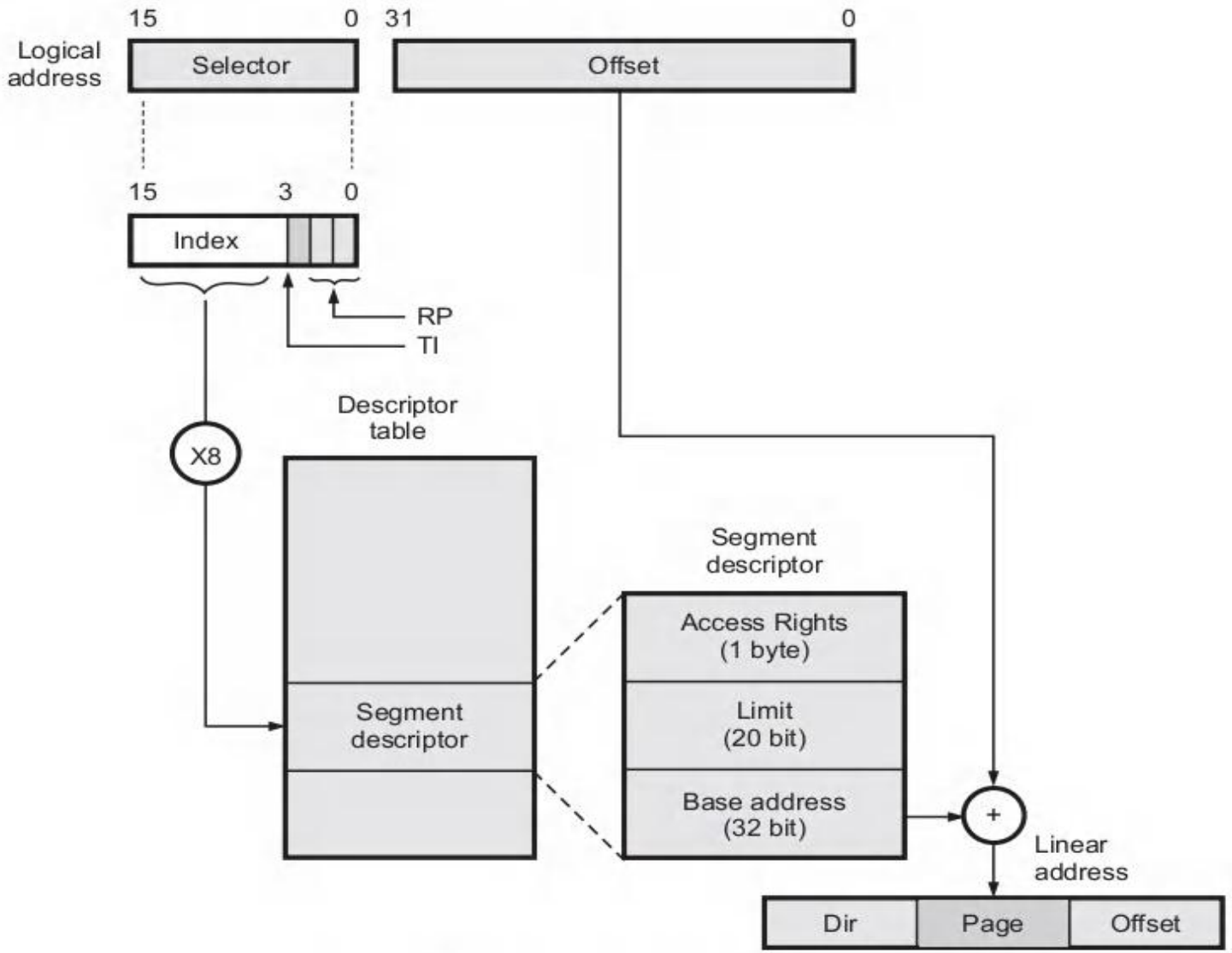
Segment translation, in which a logical address (consisting of a segment selector and segment offset) are converted to a linear address.

Page translation, in which a linear address is converted to a physical address. This step is optional, at the discretion of systems-software designers.

The linear base address from the descriptor is then added to the 32-bit offset to generate the 32-bit linear address. This process is known as **segmentation** or **segment translation**.

If paging unit is not enabled then the 32-bit linear address corresponds to the physical address. But if paging unit is enabled, paging mechanism translates the linear address space into the physical address space by **paging translation.**

Segment Translation(segmentation mechanism)



Selector

It shows how selector is used to access a descriptor in a descriptor table.

Index Part

- The 13-bit index part of selector is multiplied by 8 and used as a pointer to the desired descriptor in a descriptor table.
- The index value is multiplied by 8 because each descriptor requires 8 bytes in the descriptor table.
- The descriptor in the descriptor table contains mainly **base address, segment limit** and **access right byte.**
- The 80386 adds the base address from the descriptor to the effective address or offset to generate a linear address.

Requester's Privilege Level (RPL)

2 bits which represent the privilege level of the program

Level 0 is the most privileged and level 3 is the least privileged. More privileged levels are numerically smaller than less privileged levels.

If the segment selector has the same or greater privilege level, then the memory management unit allows the segment to be accessed.

If the selector privilege level is lower than the privilege level of the segment, the memory management unit denies the access and sends an interrupt signal to the CPU indicating a privilege level violation.

Table Indicator (TI)

Table Indicator (TI) bit decodes which descriptor table should be refered by the selector.

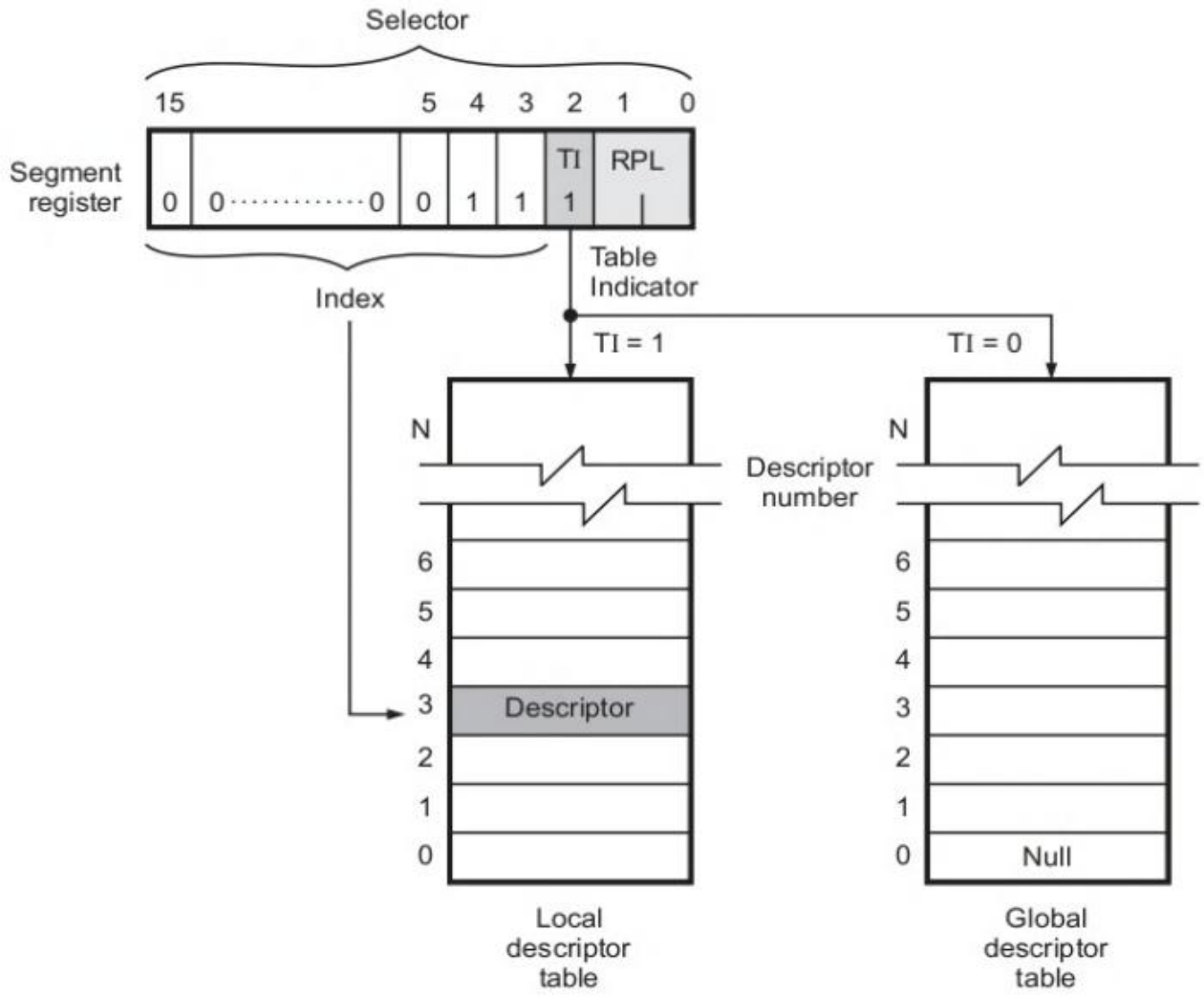
Global Descriptor table & Local Descriptor table

- TI=0, **Global Descriptor table**
- TI=1, **Local Descriptor table**

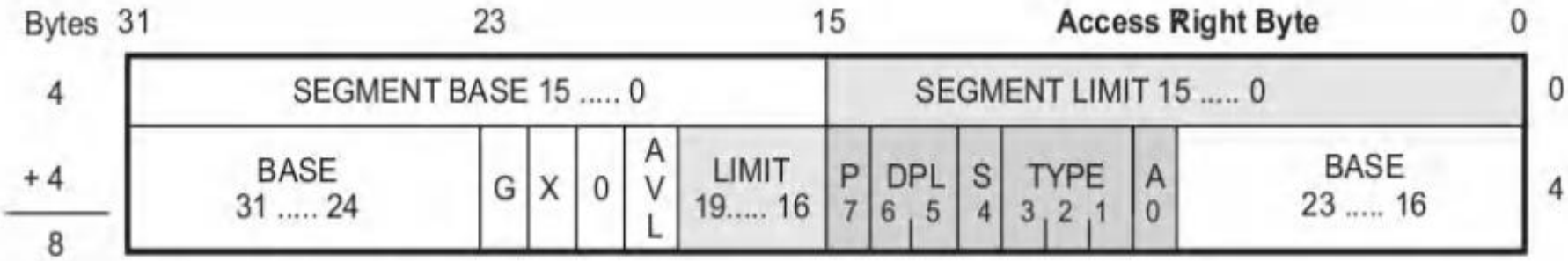
The Global Descriptor Table (GDT) is a general purpose table of descriptors, can be used by all programs to reference segments of memory.

Descriptor Table (LDT) are set up in the system for individual task or closely related group of tasks.

Selector & Descriptor table



Segment Descriptor



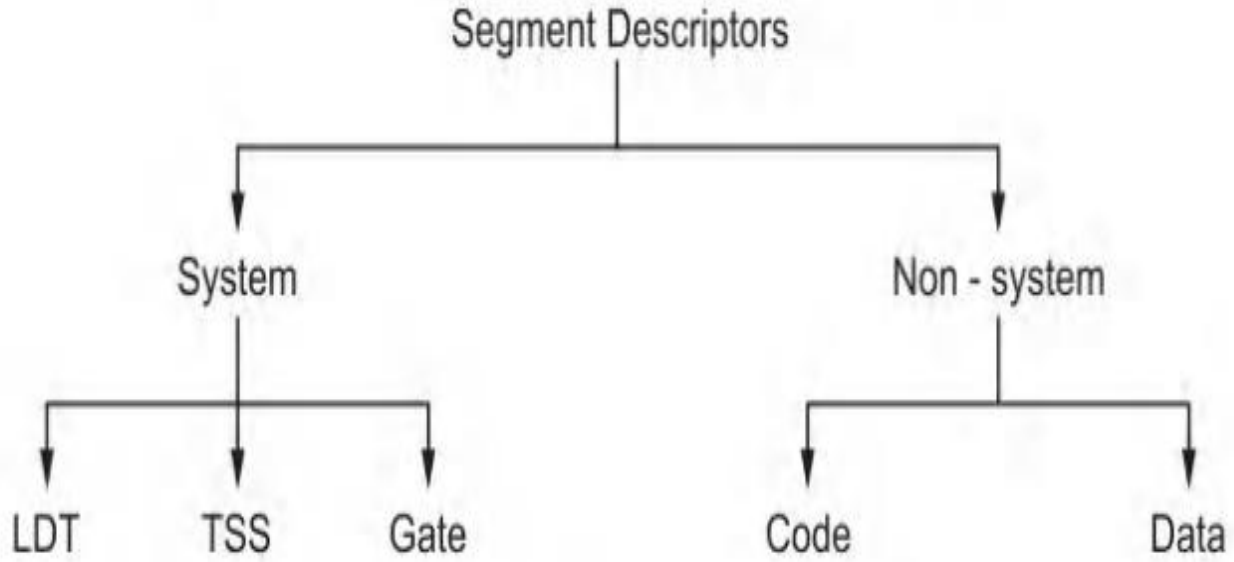
BASE	Base address of the segment
LIMIT	The length of the segment
P	Present bit : 1 = Present 0 = Not present
DPL	Descriptor privilege Level 0 - 3
S	Segment descriptor : 0 = System descriptor 1 = Code or Data segment descriptor
TYPE	Type of segment
A	Accessed bit
G	Granularity bit : 1 = Segment length is page granular 0 = Segment length is byte granular
0	Bit must be zero (0) for compatibility with future processors
AVL	Available field for user or OS

Note :
In a maximum - size segment (i.e. a segment with G = 1 and segment limit 19 ..... 0 = FFFFFH), the lowest 12 bits of the segment base should be zero. (i.e. segment base 11 ..... 000 = 000H).

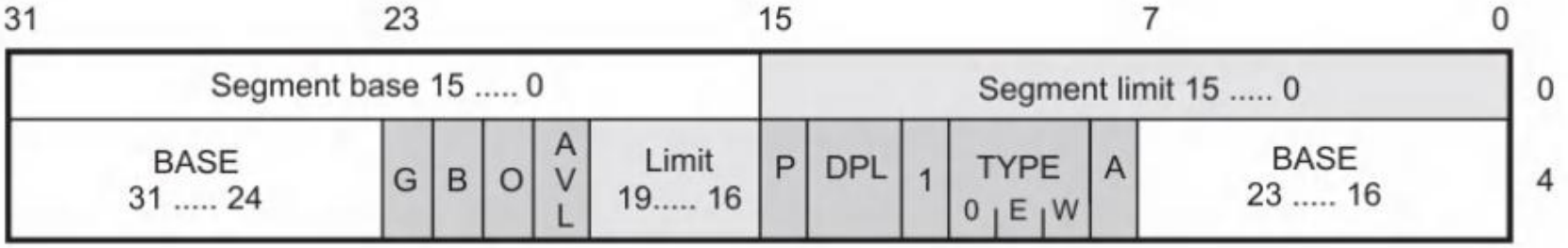
Segment Descriptor

- Describes a segment.
- Must be created for every segment.
- Is created by the programmer.
- Determines a base address of the segment (32-bit)
- Determines a size of the segment using limit field (20-bit)
- Determines a type of the segment. (4-bits)
- Determines a privilege level of the segment. (2-bits)
- Whether segment is physically present (1 bit)
- Whether segment has been accessed before (1 bit)
- Granularity of limit field (1 bit)
- Size of operands within segment (1 bit)
- Decides default operand size (1 bit)

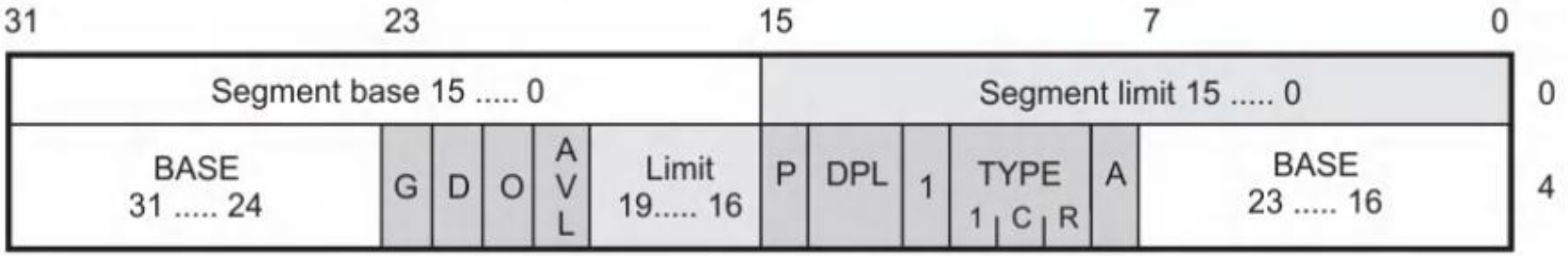
Types of Segment Descriptor



Non system Segment Descriptor



(a) Data segment descriptor

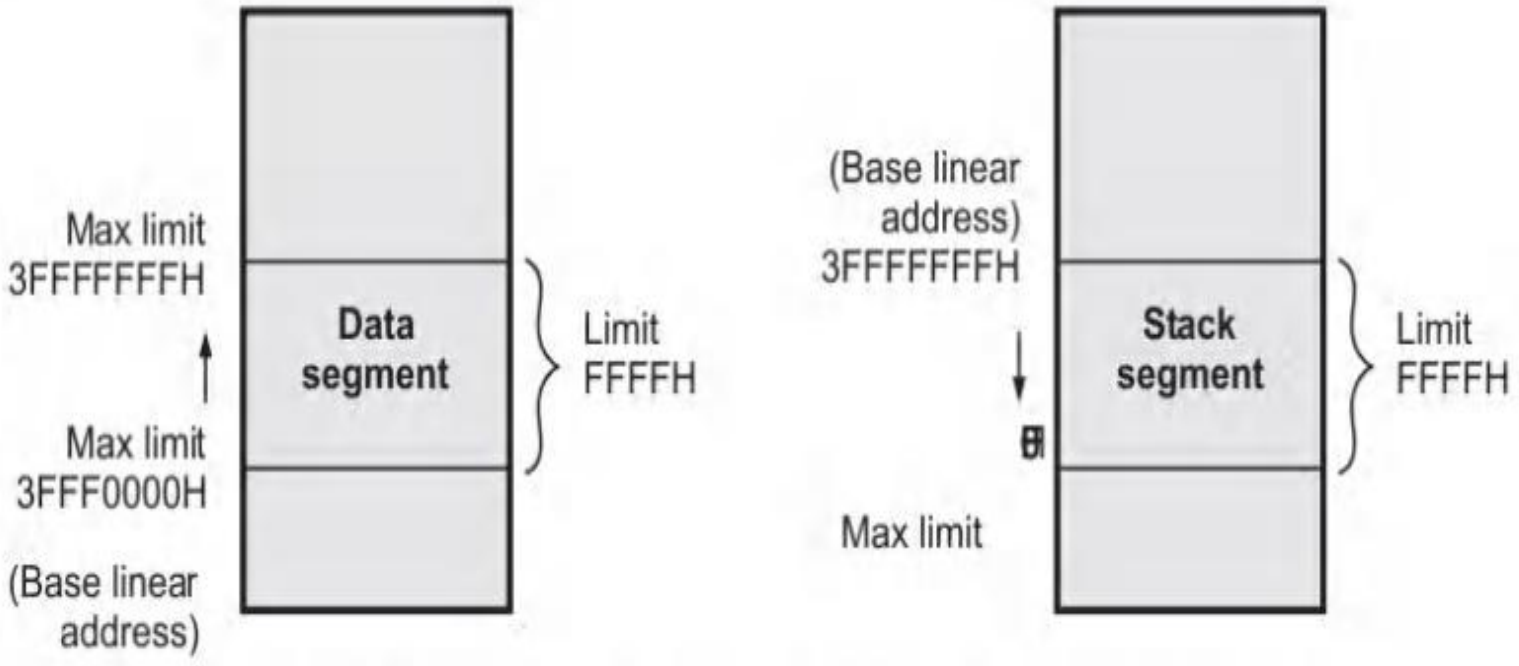


(b) Executable (code) segment descriptor

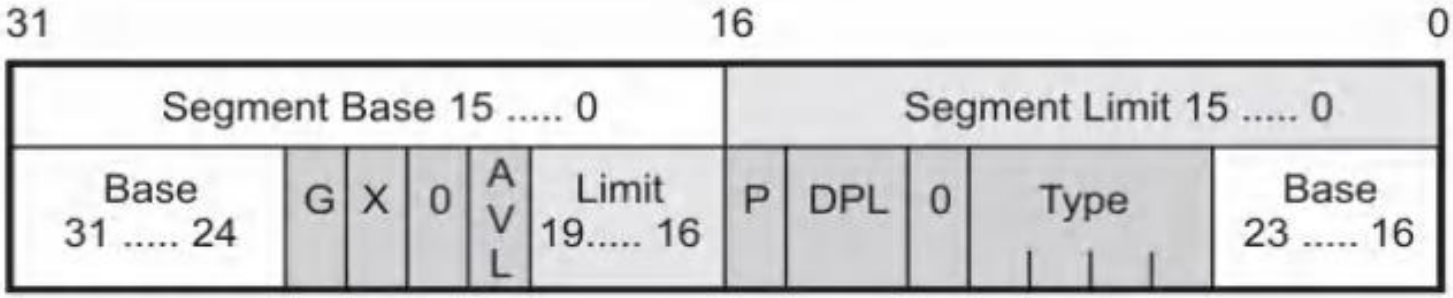
Bit Position	Name	Function	
7	Present (P)	P = 1	Segment is mapped into physical memory.
		P = 0	No mapping to physical memory exists, base and limit are not used.
6-5	Descriptor Privilege Level (DPL)		Segment privilege attribute used in privilege tests.
4	Segment Descriptor (S)	S = 1	Code or Data (includes stacks) segment descriptor
		S = 0	System segment descriptor or Gate descriptor
3	Executable (E)	E = 0	Descriptor type is data segment;
2	Expansion Direction (ED)	ED = 0	Expand up segment, offsets must be ≤ limit.
		ED = 1	Expand down segment, offsets must be > limit.
1	Writeable (W)	W = 0	Data segment may not be written into.
		W = 1	Data segment may be written into.

Note : If data segment (S = 1, E = 0)				
3	Executable (E)	E = 1	Descriptor type is code segment;	Code segment
2	Conforming (C)	C = 1	Code segment may only be executed when CPL ≥ DPL and CPL remains unchanged.	
1	Readable (R)	R = 0	Code segment may not be read.	
		R = 1	Code segment may be read.	
Note : If code segment (S = 1, E = 1)				
0	Accessed (A)	A = 0	Segment has not been accessed.	
		A = 1	Segment selector has been loaded into segment register or used by selector test instructions.	

Expansion direction for data and stack segment



System Segment Descriptor



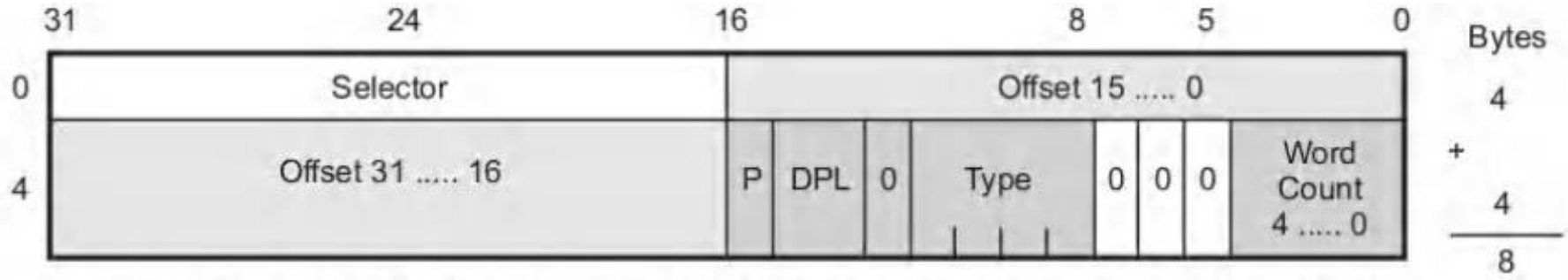
Type	Defines
0	Reserved by Intel
1	Available 80286 TSS
2	LDT
3	Busy 80286 TSS
4	80286 call gate
5	Task gate (for 80286 or Intel 80386DX task)
6	80286 Interrupt gate
7	80286 Trap gate

Type	Defines
8	Reserved by Intel
9	Available intel 80386DX TSS
A	Undefined (Intel reserved)
B	Busy intel 80386DX TSS
C	Intel 80386DX call gate
D	Undefined (Intel reserved)
E	Intel 80386DX interrupt gate
F	Intel 80386DX trap gate

LDT descriptors	TSS descriptor	Gate descriptors
(S = 0, Type = 2)	(S = 0, Type = 1, 3, 9, B)	(S = 0, TYPE = 4 – 7, C, F)
	multitasking environment	4 TYPES Call gate Task gate Interrupt gate Trap gate

Call gates are used to change privilege levels. Task gates are used to perform a task switch and interrupt and trap gates are used to specify interrupt service routines.

Gate descriptor formats



D word count : The number of double words to copy from caller's stack to the called procedure's stack. Only used with call gate.

Destination Selector :	(16 - bit)	Selector to the target code segment or Selector to the target task state segment for task gate
Destination Offset :	(32 - bit)	Entry point within the target code segment

Call gates are used to change privilege levels. Task gates are used to perform a task switch and interrupt and trap gates are used to specify interrupt service routines.

Descriptor tables

segment descriptors are grouped and placed one after the other in contiguous memory locations. This group arrangement is known as a **descriptor table.**

The maximum limit for the length of descriptor table is 64 kbytes

The Global Descriptor Table (GDT) : It is a general purpose table of descriptors, can be used by all programs to reference segments of memory. The GDT can have any type of segment descriptor except for descriptors which are used for serving interrupts.

The Interrupt Descriptor Table (IDT) : It holds the segment descriptors that define interrupt or exception handling routines. The IDT is a direct replacement for the interrupt vector table used in 8086 system.

A Local Descriptor Tables (LDT) : They are set up in the system for individual task or closely related group of tasks.

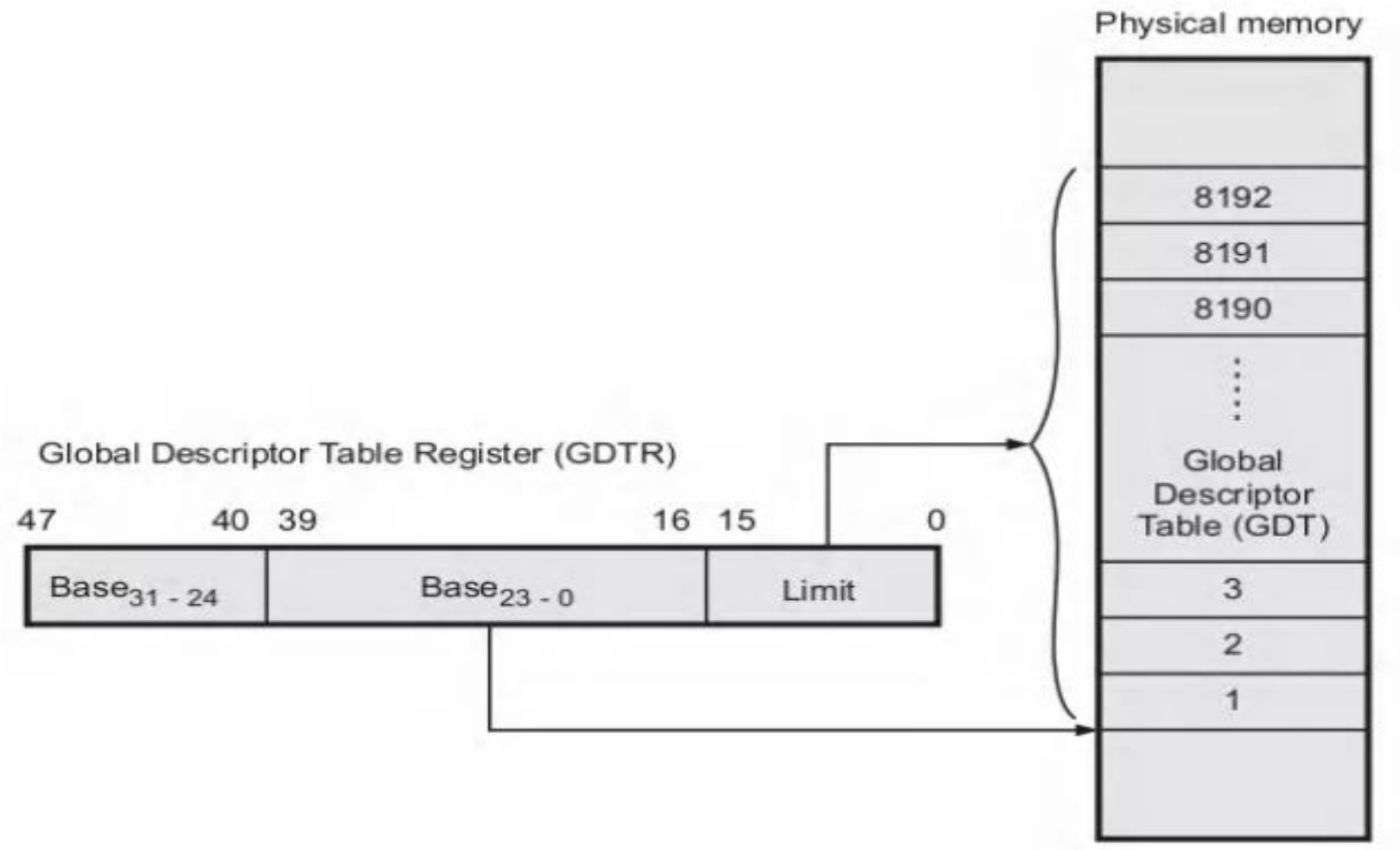
Global Descriptor Table (GDT)	Local Descriptor Table (LDT)	Interrupt Descriptor Table (IDT)
• Main table & most important one • The same GDT can be used by all programs to refer to the segment of memory. • processor in protected mode can have many LDT's but only one GDT. • It may contain special system descriptors.	• Multitasking system is defined on a per task basis. • Each task can have access to own private descriptor table(LDT) in addition to GDT. • It can also be shared with other tasks. • Each task can have its own segment of local memory. So there may be many LDT's in protected mode, say LDT-0 to LDT-n. • It can be smaller or larger than the GDT Function: Expand the total number of available descriptors	• Holds the descriptors that are used • Trap Gate Descriptor • Interrupt Gate Descriptor • Task Gate Descriptor • The user program can never select a descriptor in IDT like GDT or the LDT • It Maintains ISR • The default value that loads into IDTR as Base address=0, Limit = 03FFH

Descriptor Registers-GDTR, LDTR, IDTR

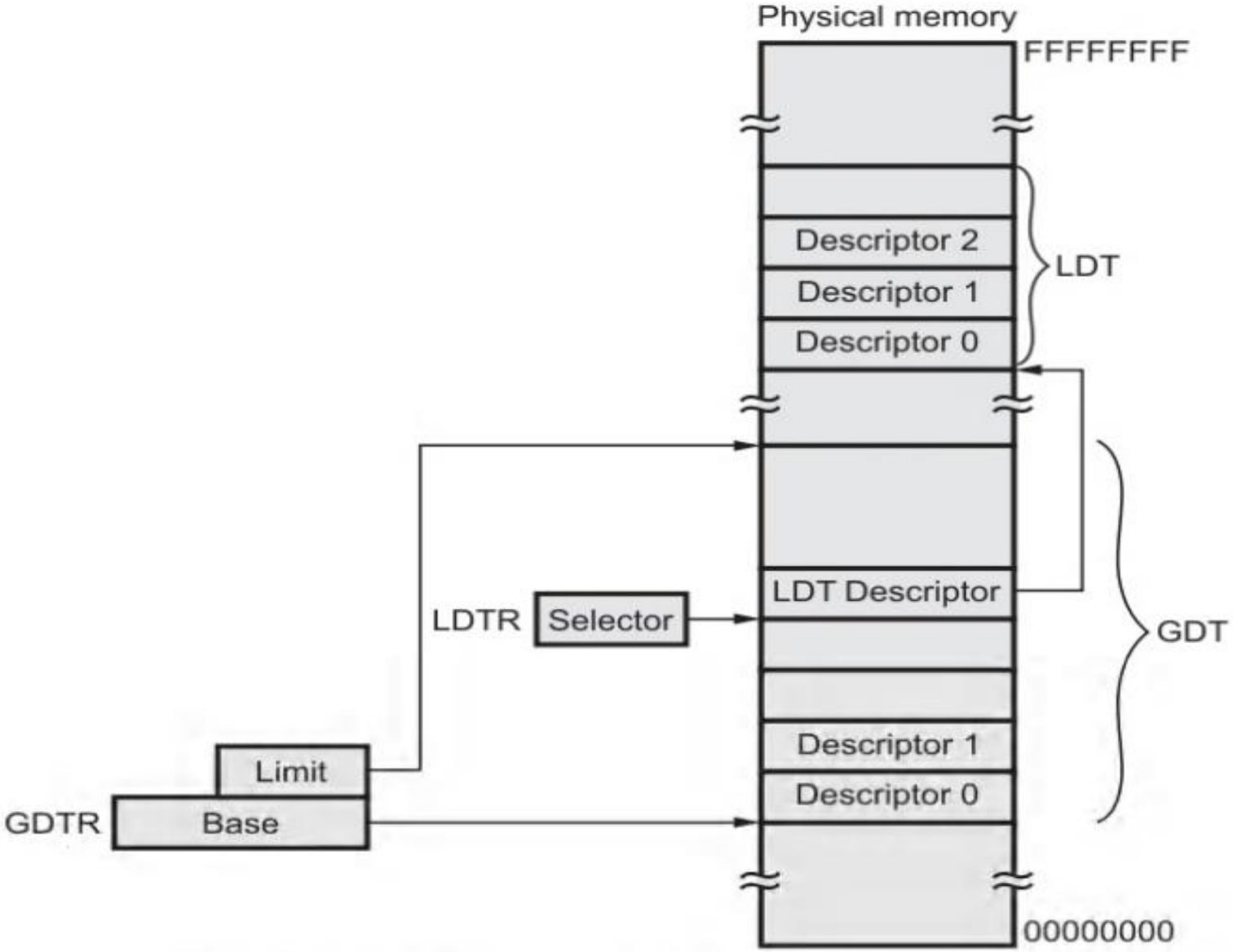
- Descriptor Registers=32 bit linear Address +table limit

GDTR	IDTR	LDTR
48 bit register	48 bit register	16 bit register
Limit is 16 bit, maximum size=65,536 bytes	Limit is 16 bit, maximum size=65,536 bytes But support 256 intrrupts only.	not specify limit(base address) specify address of LDT in GDT

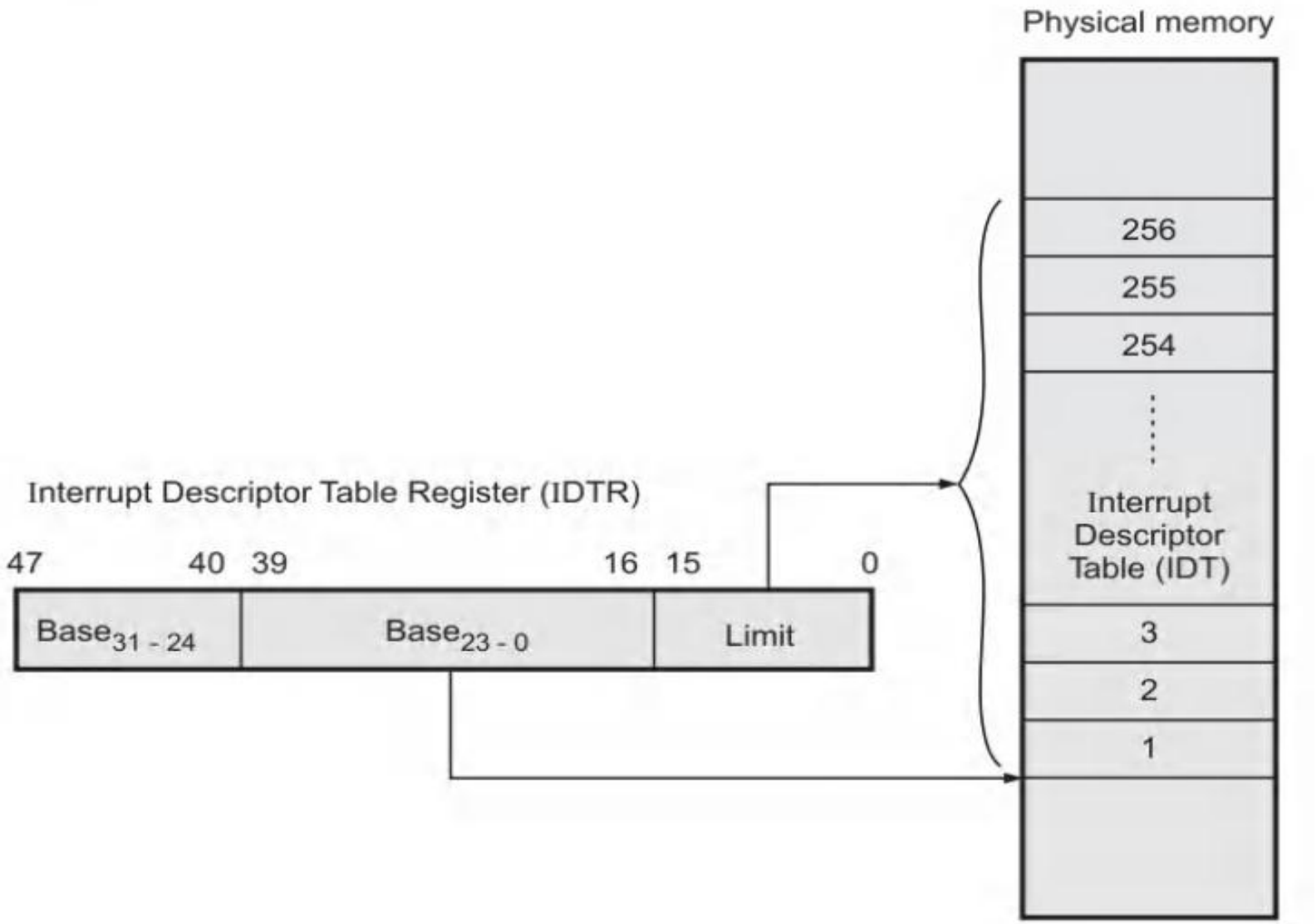
GDTR-Global Descriptor Table Register



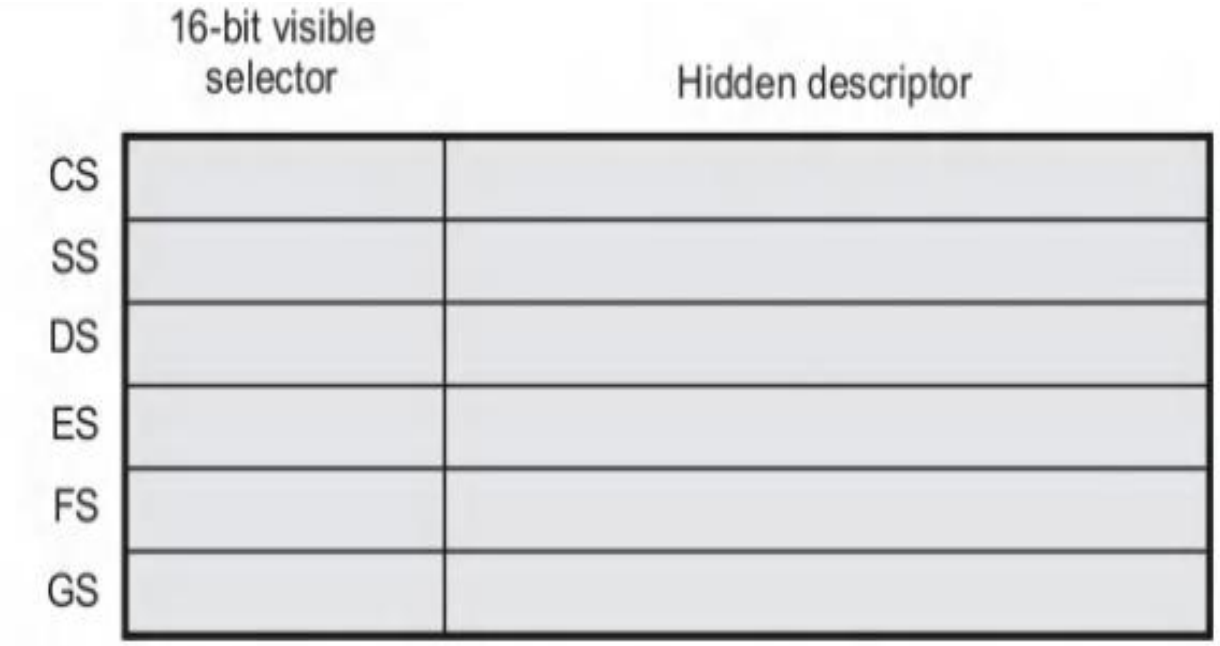
LDTR- Local Descriptor Table Register



IDTR-Intrrupt Descriptor Table Register



Segment register & segment descriptor cache



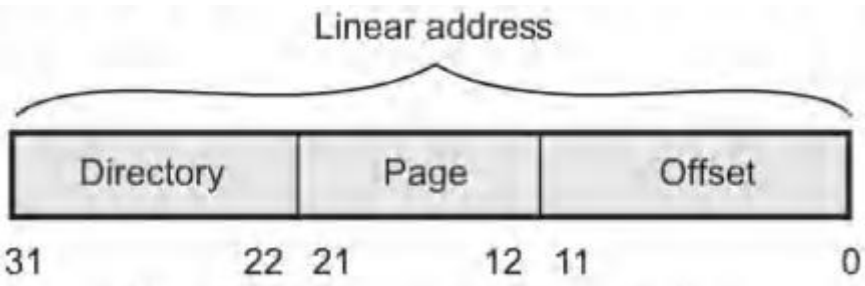
Segment register (visible portion) contents are manipulated by programs whereas segment descriptor cache register (hidden portion) contents are manipulated by processor.

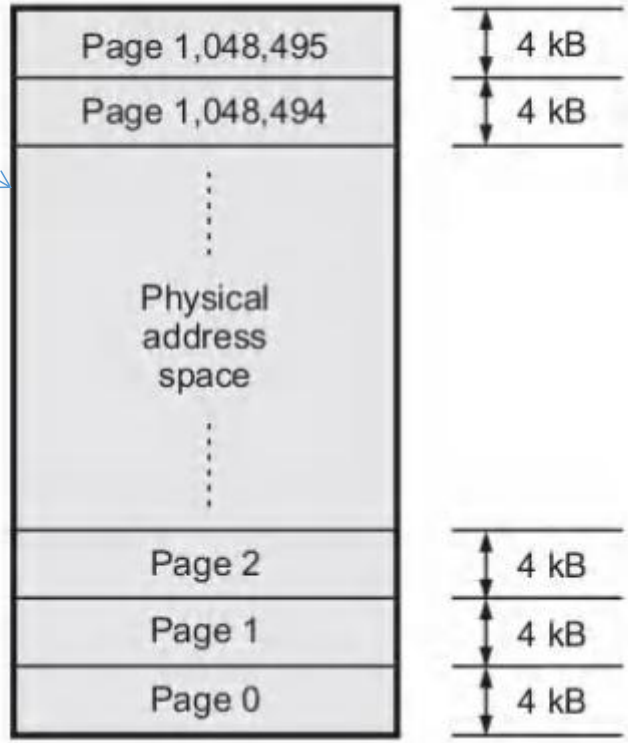
Page Translation

The page translation step is optional. Page translation is in effect only when the **PG** bit **of CR0 is set.**

Page translation is must if the operating system is to implement multiple virtual 8086 tasks, page-oriented protection, or page oriented virtual memory.

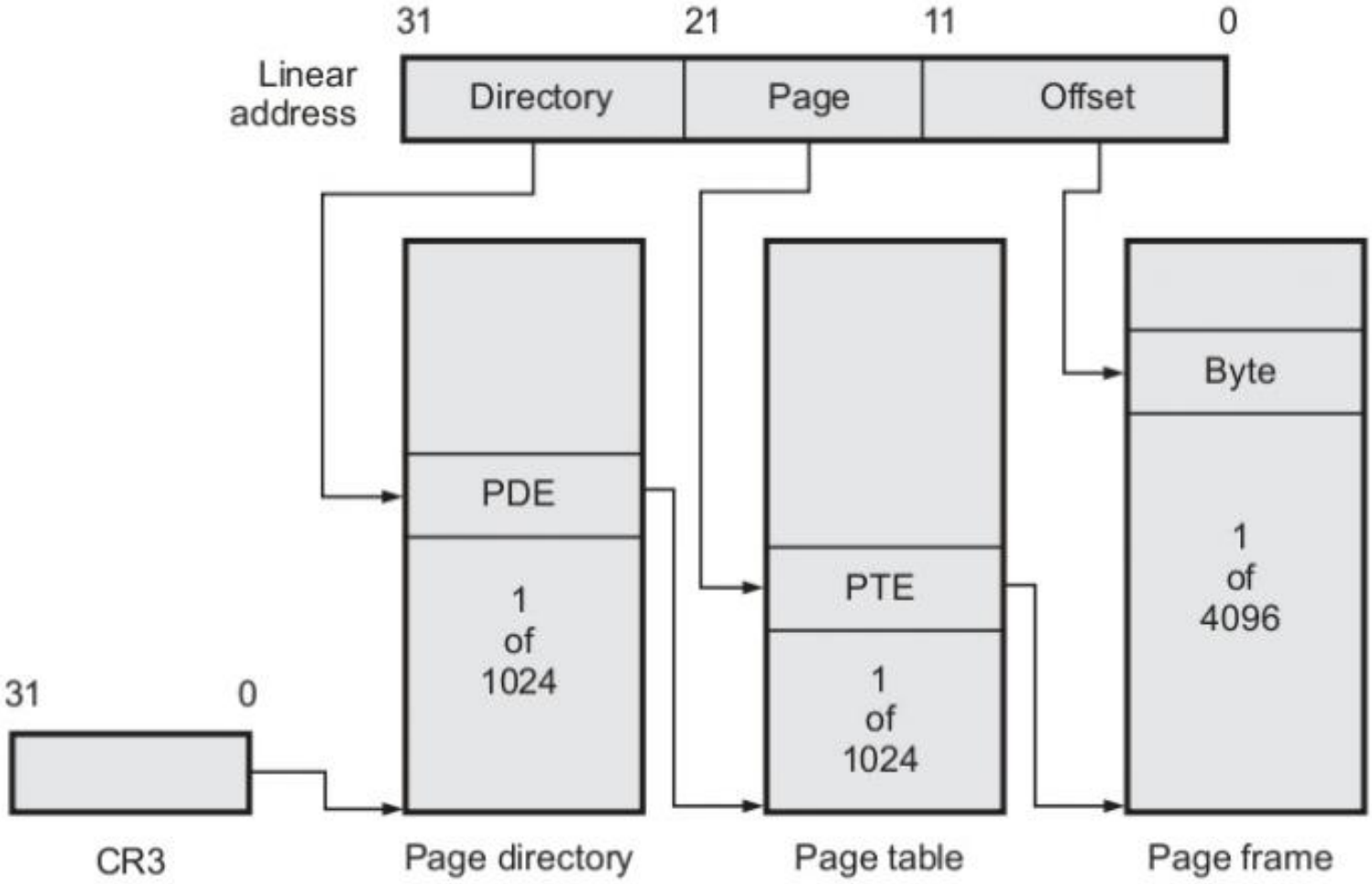
When paging is enabled, the paging unit arranges the physical address space into 1,048,496 pages that are each 4096 bytes





Linear to physical address translaion

Page Table Entry (PTE).
Page Directory Entry (PDE)



Page Table

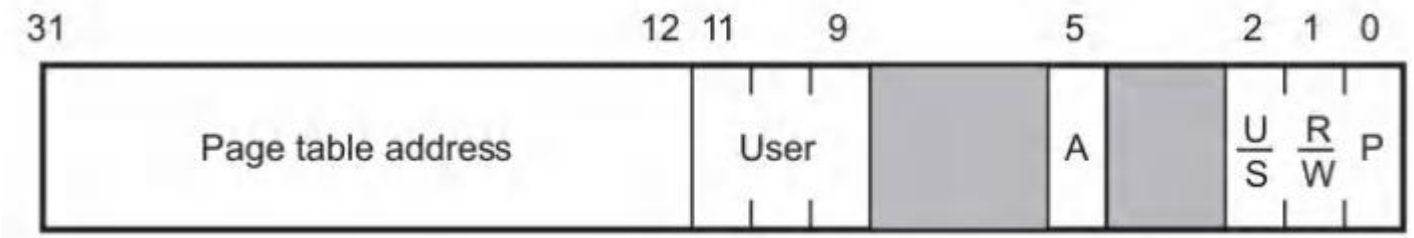
A page table is simply an array of 32-bit page specifiers. A page table is itself a page, and therefore contains 4 Kilobytes of memory or at most 1K 32-bit entries.

Two levels of tables are used to address a page of memory.

At the higher level is a **page directory**. The page directory addresses up to 1K **page tables** of the second level.

A page table of the second level addresses up to 1K pages.

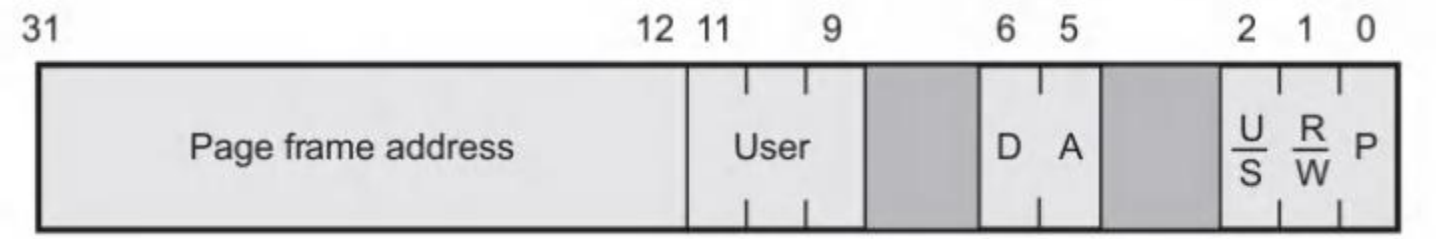
PDE Descriptor



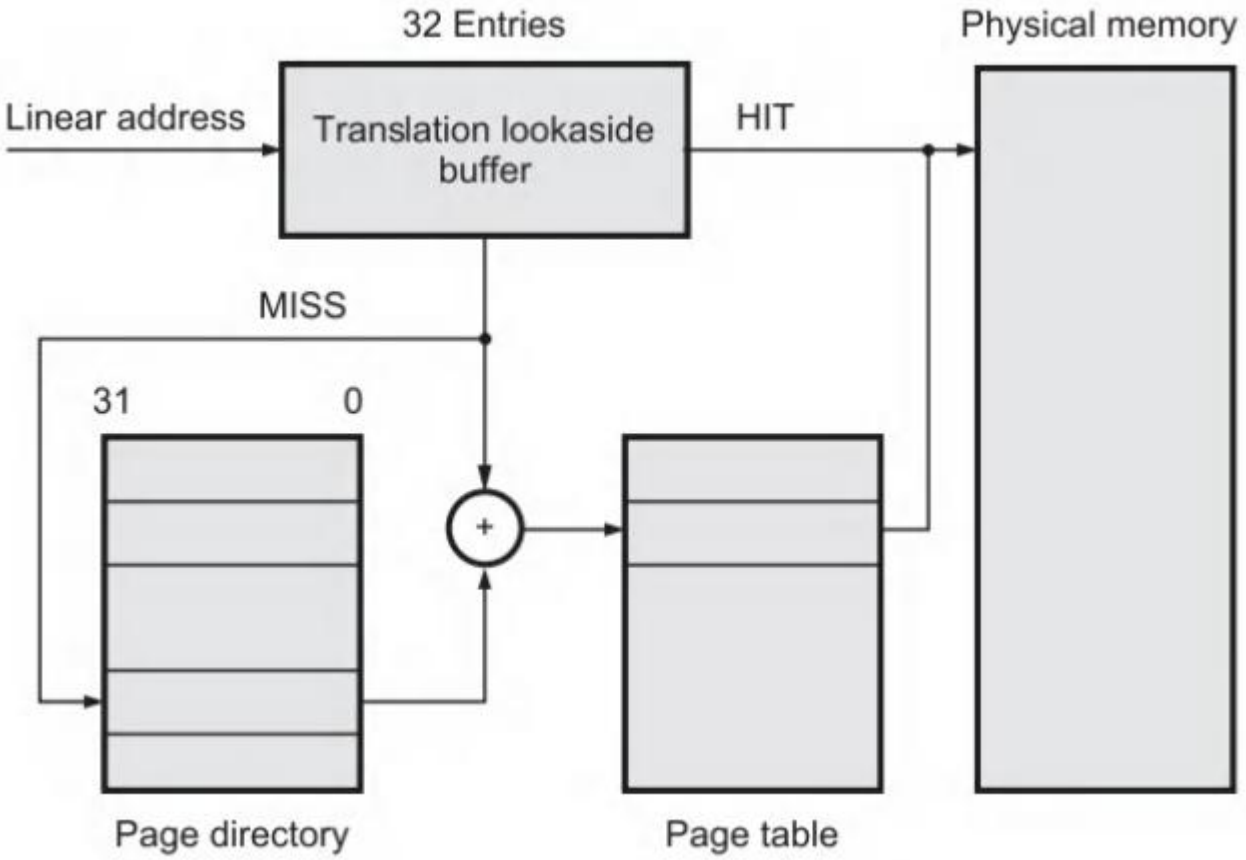
U/$\overline{S}$	R/$\overline{W}$	Permitted Level 3	Permitted Access Levels 0, 1, or 2
0	0	None	Read/Write
0	1	None	Read/Write
1	0	Read-Only	Read/Write
1	1	Read/Write	Read/Write

User/ $\overline{\text{Supervisor}}$ and Read/ $\overline{\text{Write}}$ Bits

PTE Descriptor



Translation Lookaside Buffer/Page Translation Cache



Combining Segment and Page Translation