

UNIT-II MEMORY MANAGEMENT

Page No.

- Q1 Write a short note on GDTR, IDTR & LDTR.

- (i) Global Descriptor Table Register :-

- GDTR is a 58-bit register located inside the 80386DX.

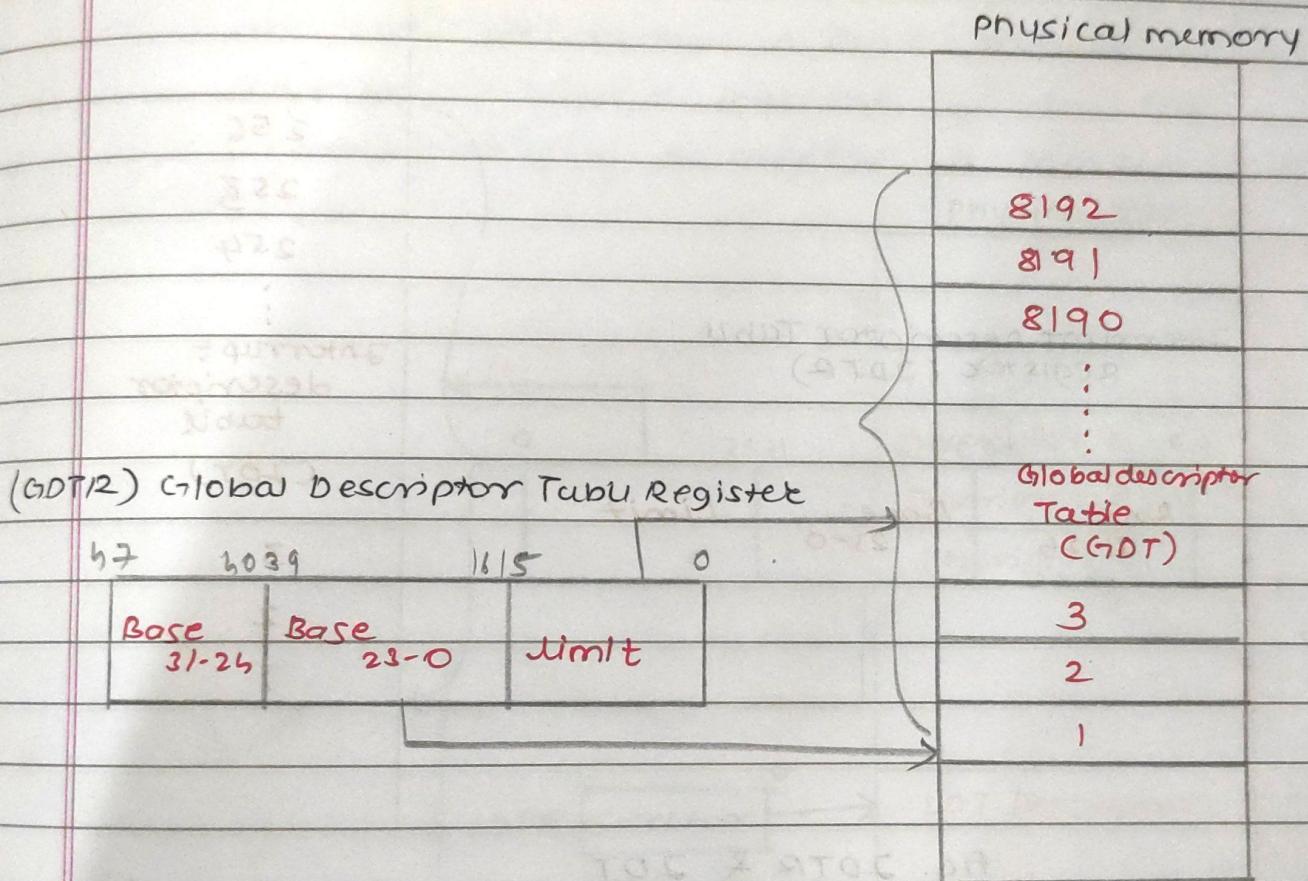


Fig. GDTR and GDT

- The lower two bytes (16 bit) of this register specifies the limit, for GDT. The value of limit is one less than the actual size of the table.
- The upper 5 bytes of GDTR specifies the 32-bit linear address of the base of the Global descriptor table.

(ii) Interrupt Descriptor Table Register (IDTR) :

- Like GDTR, IDTR holds the 16-bit limit & 32-bit linear address of the base of the interrupt descriptor table.

physical memory

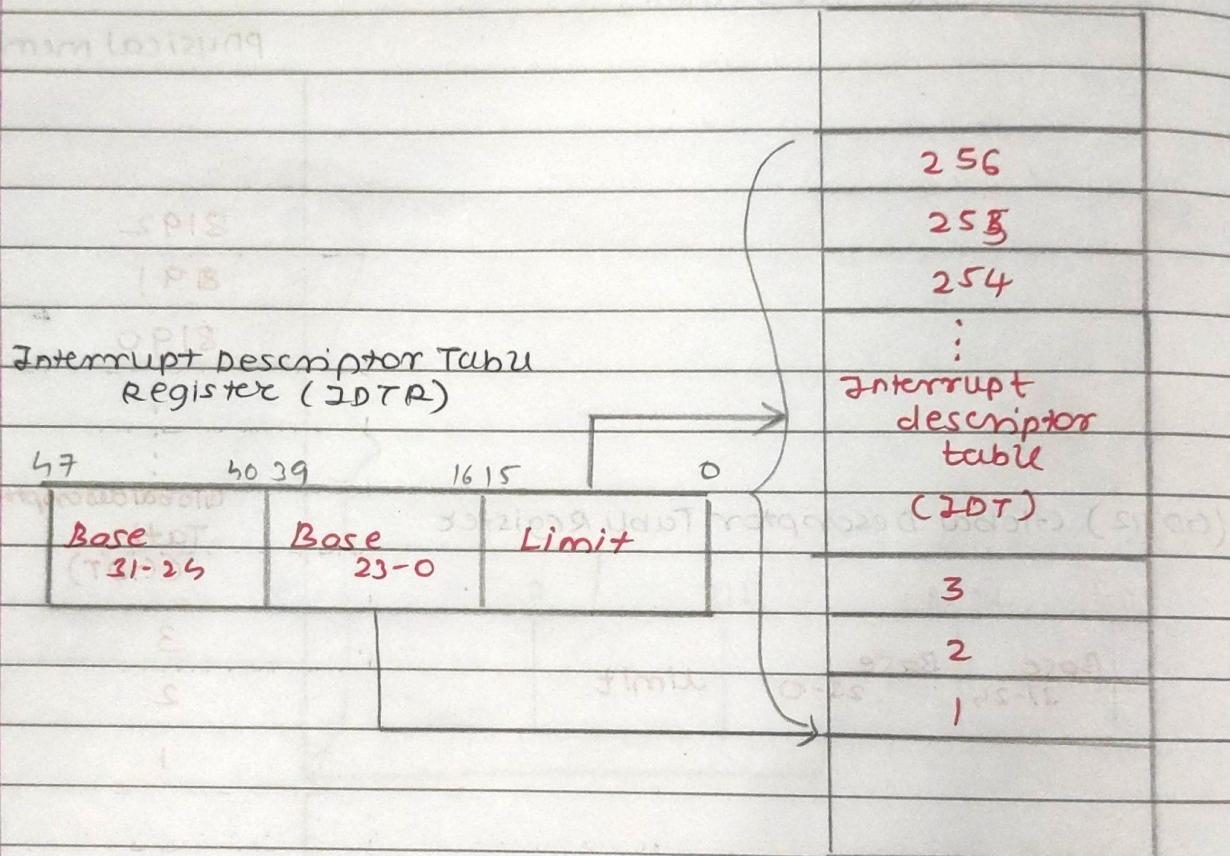


fig. IDTR & IDT

- IDTR is also 64-bit in length, with lower 2 bytes defines limits and upper 5 bytes defines the base address.
- Since limit field is of 2 bytes, the IDT can also be used up to 65536 bytes long. But the 80286DX only supports upto 256 interrupts or exceptions, therefore the size of IDT should not be set to support more than 256 interrupts.

(iii) Local Descriptor Table Register (LDTR):

- LDTR is a 16-bit register, unlike IDTR & GDTR.
- It specifies the address of LDT descriptor stored in GOT. IDTR holds a selector that points to an LDT descriptor in GDT. Whenever a selector is loaded into the LDTR, the corresponding descriptor is located in GDT physical memory.

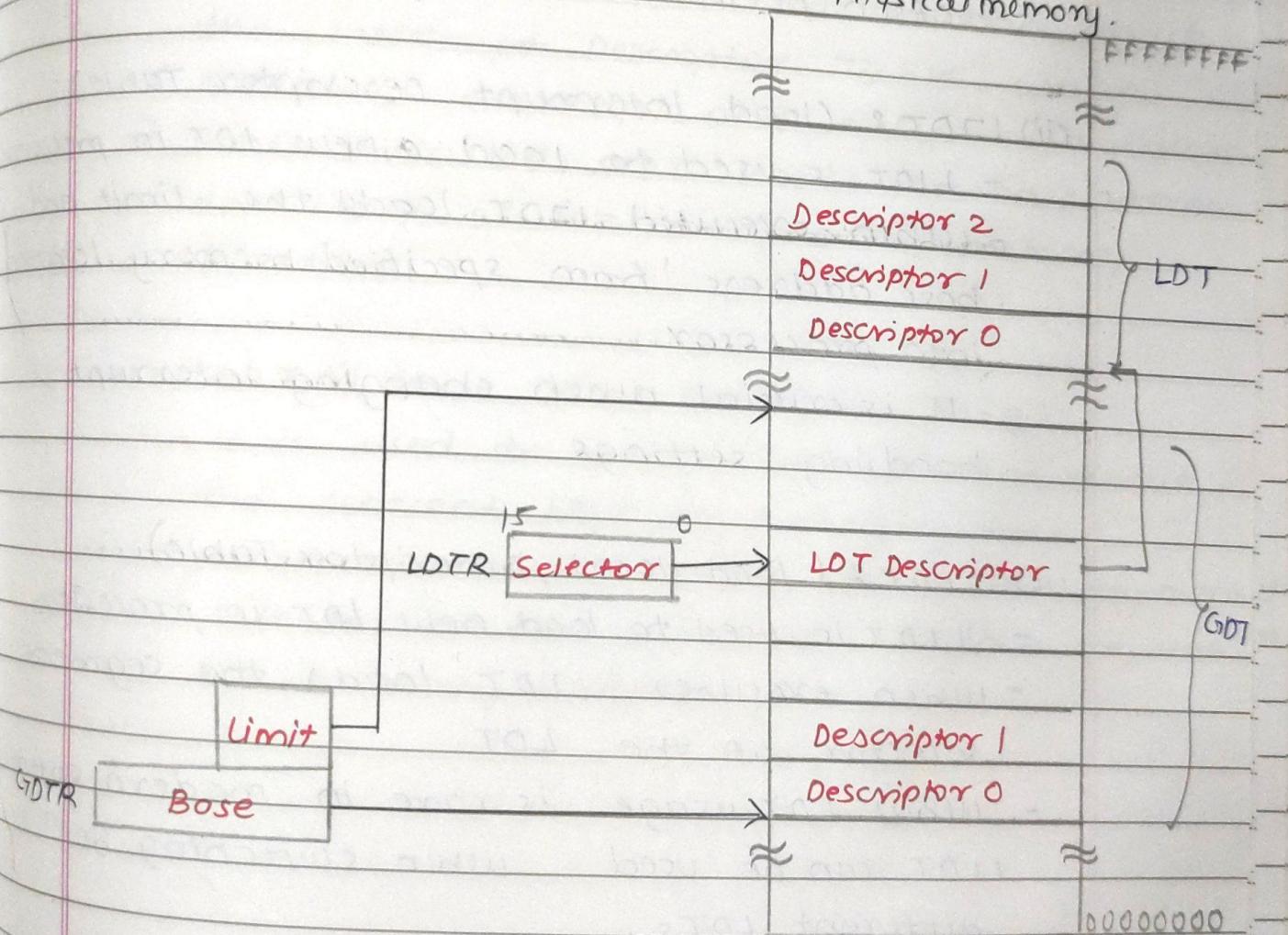


Fig. GDT & LDT

- The content of this descriptor defines the LDT.

(2) Instructions :-

(i) LGDT : (load Global Descriptor Table)

- LGDT is used to load a new GOT in processor.
- When executed, LGDT loads the base address and limit of GOT from a specified memory location (GOTR) into processor.
- It is crucial for setting up of changing memory segmentation settings, often performed during OS initialization.

(ii) LIOT : (load Interrupt Descriptor Table) :

- LIOT is used to load a new IDT in processor.
- When executed, LIOT loads the limit and base address from specified memory location into processor.
- It is crucial when changing interrupt handling settings.

(iii) LLDT : (load Local Descriptor Table)

- LLDT is used to load new LDT in processor.
- When executes, LLDT loads the segment selector of the LDT.
- While LDT usage is rare in modern systems, LLDT can be used when switching between different LDTs.

(ii) SGDT : (store global descriptor table)

- SGDT is used to obtain information about GDT,
- When executed SGDT ~~loads~~ stores the base address and limit of GDT in specified memory location.

(iii) SIDT : (store interrupt descriptor Table):

- SIDT is used to obtain information about the interrupt Descriptor Table , which contains interrupt and exception handlers.
- When executed SIDT stores the base address and limit of IDT in specified memory location

(iv) SLDT : (store local Descriptor Table):

- It is used to obtain information about the current LDT for managing task.
- When executed , SLDT stores the segment selector of the current LDT in a register or memory location.

③ With neat diagram explain ^{segment} page translation process in 80386.



- Segment translation is a process of converting logical address into a linear address.
- The selector is used to access descriptor in descriptor table. The 13-bit index part of selector is multiplied by 8 and used as a pointer to point desired descriptor in descriptor table.
- The index value is multiplied by 8 because each descriptor requires 8 bytes in descriptor table. The descriptor in descriptor table contains mainly base address, segment limit and access rights byte.
- The 80386 adds the base address from the descriptor to effective address or offset to generate a linear address.
- The Table Indicator (TI) bit in selector decides which descriptor table should be referred by selector : TI=0 (GDT) and TI=1 (LDT).
- The selector components of each logical address contains 2 bits (RP) which represents privilege level of the program section requesting access to a segment.

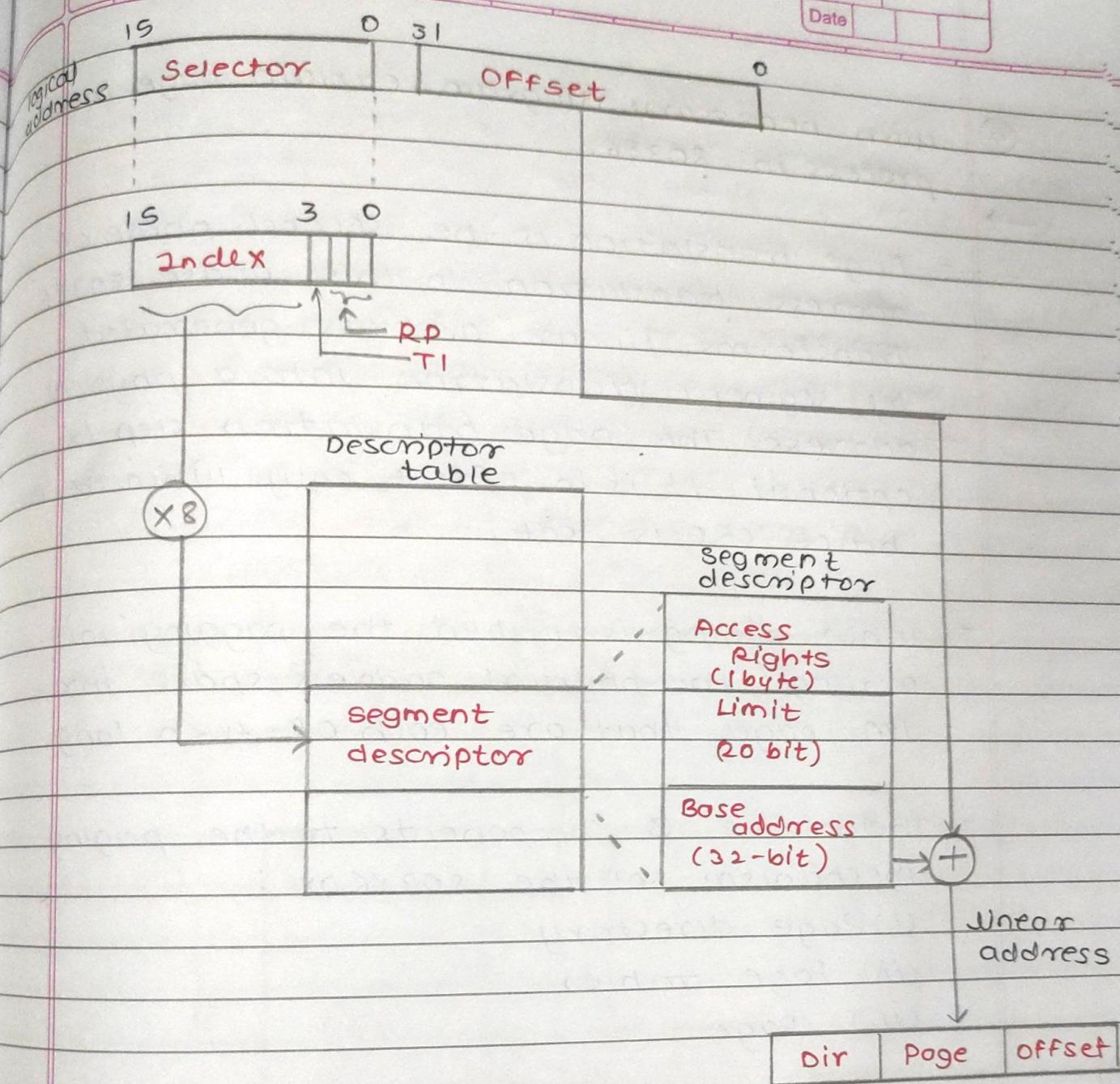


fig. Segmentation mechanism

- Level 0 is the most privileged and level 3 is least privileged. More privileged levels are numerically smaller than less privileged levels.
- When an executing program attempts to access a segment, the MMU compares the privilege level in selector with privilege level in segment descriptor.

(4) with necessary diagram, explain page translation process in 80386.



- Page translation is the second phase of address translation. In this phase 80386 transforms linear address generated by segment translation into a physical address. The page translation step is optional. It is in effect only when the PG bit of CR0 is set.
- When paging is enabled the paging unit arranges the physical address space into 1M pages that are each of 4KB long.
- There are 3 components to the paging mechanism of the 80386DX :
 - (i) Page directory
 - (ii) Page tables
 - (iii) Page.
- When paging is enabled 80386DX uses two levels of tables to translate the linear address into physical address.

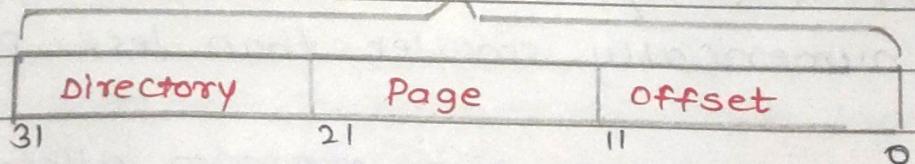


fig. Linear address format

- process internally divides linear address into 3 parts fields: Two field of 10 bits each and one field of 12 bits.
 - (i) The most significant 10 bits (DIR) are used as an index into page directory.
 - (ii) The next most significant 10 bits (PAGE) are used as an index into page table.
 - (iii) The least 12 bits (OFFSET) selection of 4KB page of memory from page frame determined by page tabu.

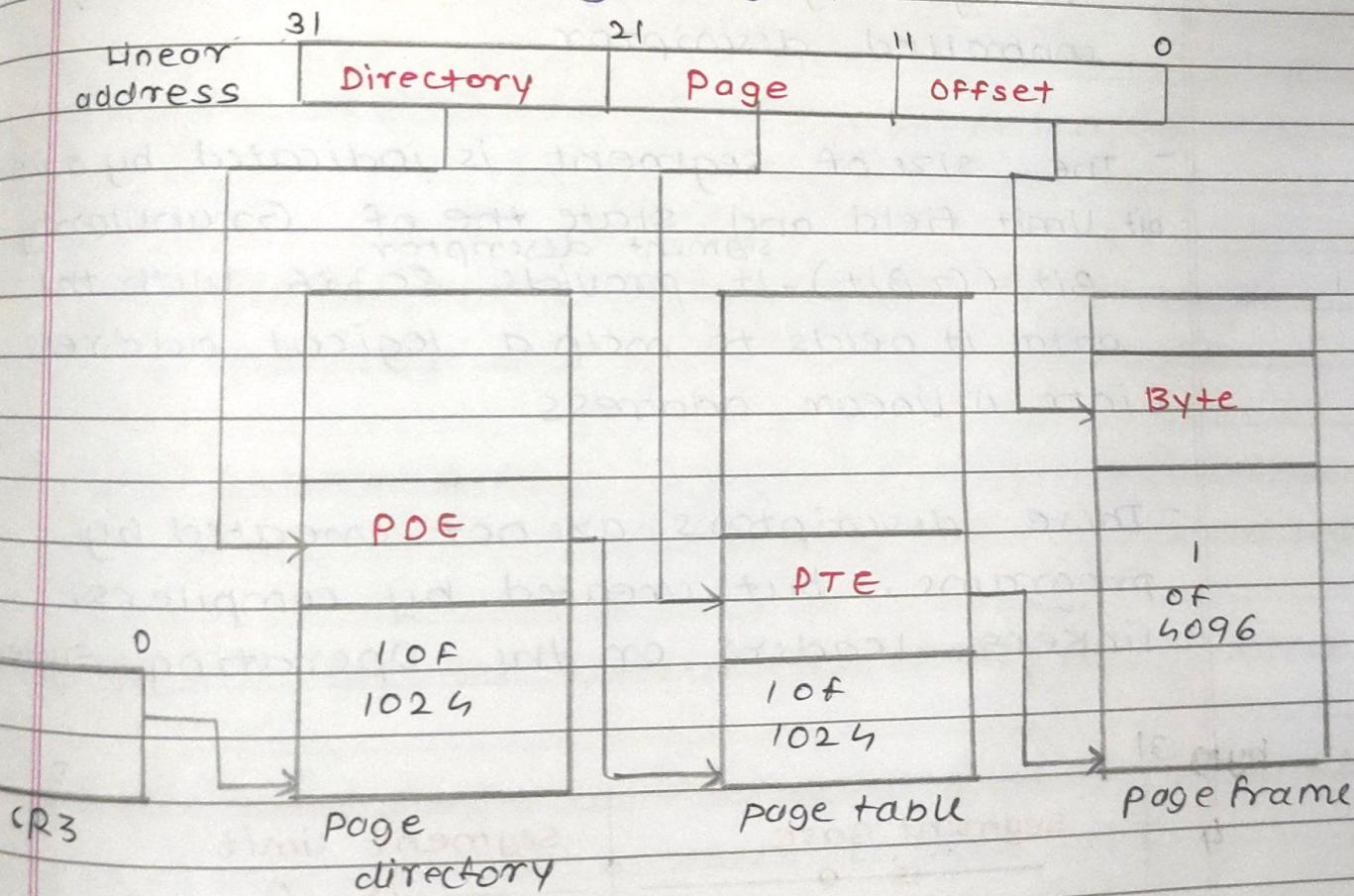


fig. Linear to physical address translation

- The physical address of current page directory is stored in **CR3** which is also referred to as **Page Directory Base Register (PDBR)**.

The descriptor in a page directory is referred to as a page directory Entry (PDE) and descriptor in a page table is referred to as a page table Entry (PTE).

⑤ Demonstrate general Descriptor format available in various descriptor table.

⇒ - A descriptor is a series of 8 bytes that describe and locate a memory segment. It contains 32-bit address that specifies the beginning of segment of memory controlled descriptor

- The size of segment is indicated by a 20-bit limit field and state the of Granularity bit (G Bit) - it provide 80386 with the data it needs to map a logical address into a linear address.

- These descriptors are not created by programs, but created by compilers, linkers, loaders or the operating systems.

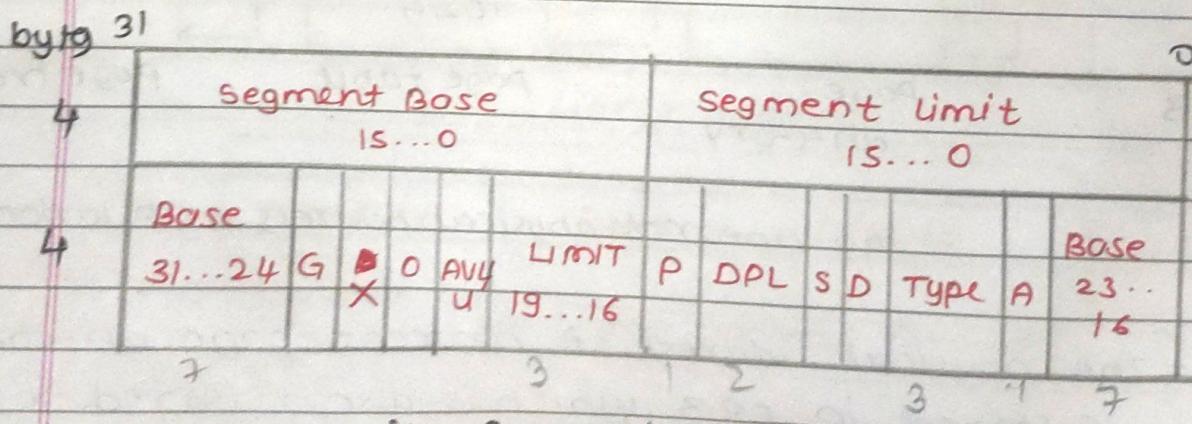


fig. General segment descriptor format

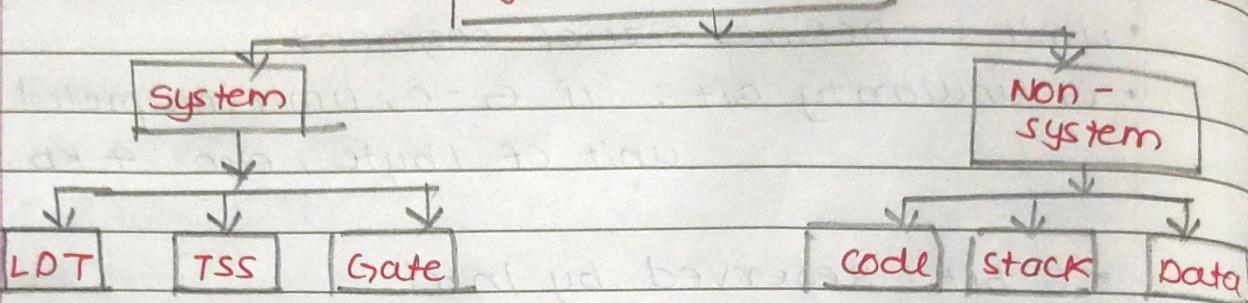
General Format Descriptor:

- Base : contains 32-bit base address of segments
- Limit : defines size of segment
- Granularity Bit : if G = 0, limit interpreted in a unit of 1 byte, else 4 KB.
- O bit : Reserved by Intel
- U (User bit) : available for user or OS.
- P (Present bit) : P bit is 1, when segment is loaded in physical memory.
- DPL (Descriptor Privileged Level) : DPL 0 is most privileged and DPL 3 is least privileged
- S (System Bit) : if S bit is 1 \rightarrow code or data segment
if it is 0 segment is system segm.
- Type : specifies specific descriptor among other
- A (Accessed Bit) : auto set when segment is loaded in segment register.

- ⑥ Enlist various types of system and non-system descriptor in the 80386. Explain their use in brief.



System Descriptor



System Descriptor :-

(i) Local Descriptor Table (LDT):

- The LDT is an optional table that can be used to store additional segment descriptor. It's similar to the GDT but is specific to a particular task or process.

(ii) Task State Segment (TSS):

- The TSS descriptor provides information about the task state, including stack pointer, privilege levels and other task related data. It's used during task switching and context switching.

(iii) Gate Descriptor:

- These are special descriptors used for interrupt gates, trap gates and call gates.
- They facilitate transitions betn different privilege levels and handle interrupts & exception.

Non-System Descriptors:

(i) Code Segment Descriptor:

It is used for executable code, such as program instructions. It specifies the base address, limit and rights for code segment.

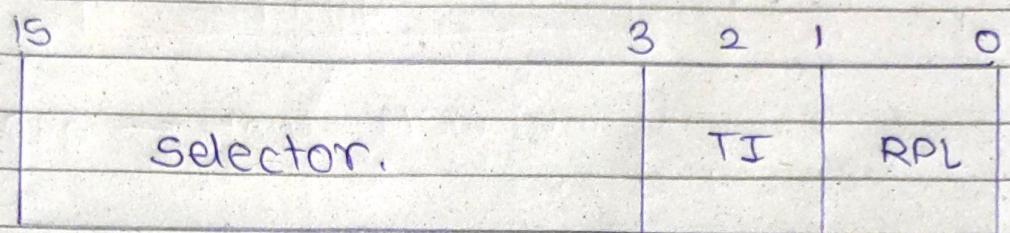
(ii) Stack Segment Descriptor:

It is used for stack operations. It provides information about the stack's location and size

(iii) Data Segment Descriptor:-

It is used for data storage, similar to the code segment descriptor, it defines the base address, limit and access rights for data segment.

* Demonstrate General Selector format in brief.



RPL (Requested privilege level) $TI = 0 \text{ GDT}$
 $TI = 1 \text{ LDT}$

- The Selector located in the segment register select's one of 18192 descriptor from one of two table of descriptor.
- descriptor table's are
 1. Global descriptor Table,
 2. Local descriptor Table.
- The selector contain total 16-bit's.
- first two bit (0-1) contain RPL.
 - the RPL is Requested privilege level it is used to represent privilege level of procedure that originate's a selector.
- TI bit :-
 The TI bit is Table Identity bit if bit is
 - 0 → then it select GDT
 - 1 → then it select LDT.



- * what is logical address, Linear Address and Physical Address?

⇒

Logic Address :

- This is the address generated by the CPU during program execution. It's also known as virtual address. The logical address space is typically larger than the physical address space.
- The operating system maps logic addresses to physical addresses through a process called address translation. In this first the logical address is converted to linear address by segmentation and then linear address is converted to physical address by page translation.

Linear Address :

- It is also sometimes referred to as virtual or intermediate address. The linear address is step closer to physical memory than the logical address. It's essentially a logical address after being translated by Memory Management Unit (MMU), but before being mapped to the physical address.
- The logic address is converted to linear address by segmentation mechanism of MMU and then converted to physical address by translation.

Physical address :

- This is the actual location in physical memory where data is stored. The physical address space is the range of addresses directly accessible by the hardware.
- It's the address that is ultimately sent to the memory module of data retrieval or storage. Physical address is generated by two levels of mechanism one is segmentation and other is translation.

- **GDTR** deals with segment descriptors for the entire system.
- **LDTR** handles process-specific segment descriptors (if an LDT is used).
- **IDTR** manages interrupt and exception descriptors.

OR

1. **GDTR (Global Descriptor Table Register):**

- Holds the base address and limit of the GDT.

- GDT defines memory segments for the entire system.
- Created by compilers, linkers, loaders, or the operating system.
- Segments described in the GDT include code segments, data segments, and system segments.
- The processor uses GDT descriptors to map logical addresses to linear addresses.
- GDT can contain up to 8192 descriptors.
- The first entry (INDEX=0) is not used by the processor.
- GDTR is set using the `lgdt` instruction.

2. **LDTR (Local Descriptor Table Register):**

- Holds the base address and limit of the LDT.
- LDT is specific to a particular task or process.
- LDT descriptors define memory segments for a specific context.
- LDT is set using the `lldt` instruction, which takes a selector pointing into the GDT.
- LDT is optional and not commonly used in modern operating systems.

3. **IDTR (Interrupt Descriptor Table Register):**

- Contains the base address and limit of the IDT.
- IDT is used for handling interrupts and exceptions.
- IDT descriptors define interrupt and exception handlers.
- IDT entries correspond to specific interrupt vectors.
- IDTR is set using the `lidt` instruction.