CSE-3103: Microprocessor and Microcontroller

Dept. of Computer Science and Engineering
University of Dhaka

Prof. Sazzad M.S. Imran, PhD
Dept. of Electrical and Electronic Engineering
sazzadmsi.webnode.com

Real Mode vs. Protected Mode Operation

80286 and above → real or protected mode.
8086 and 8088 → real mode.

Real mode operation →

- addresses only first 1M byte of memory space.
- allows upward compatibility of application software written for 8086/8088.

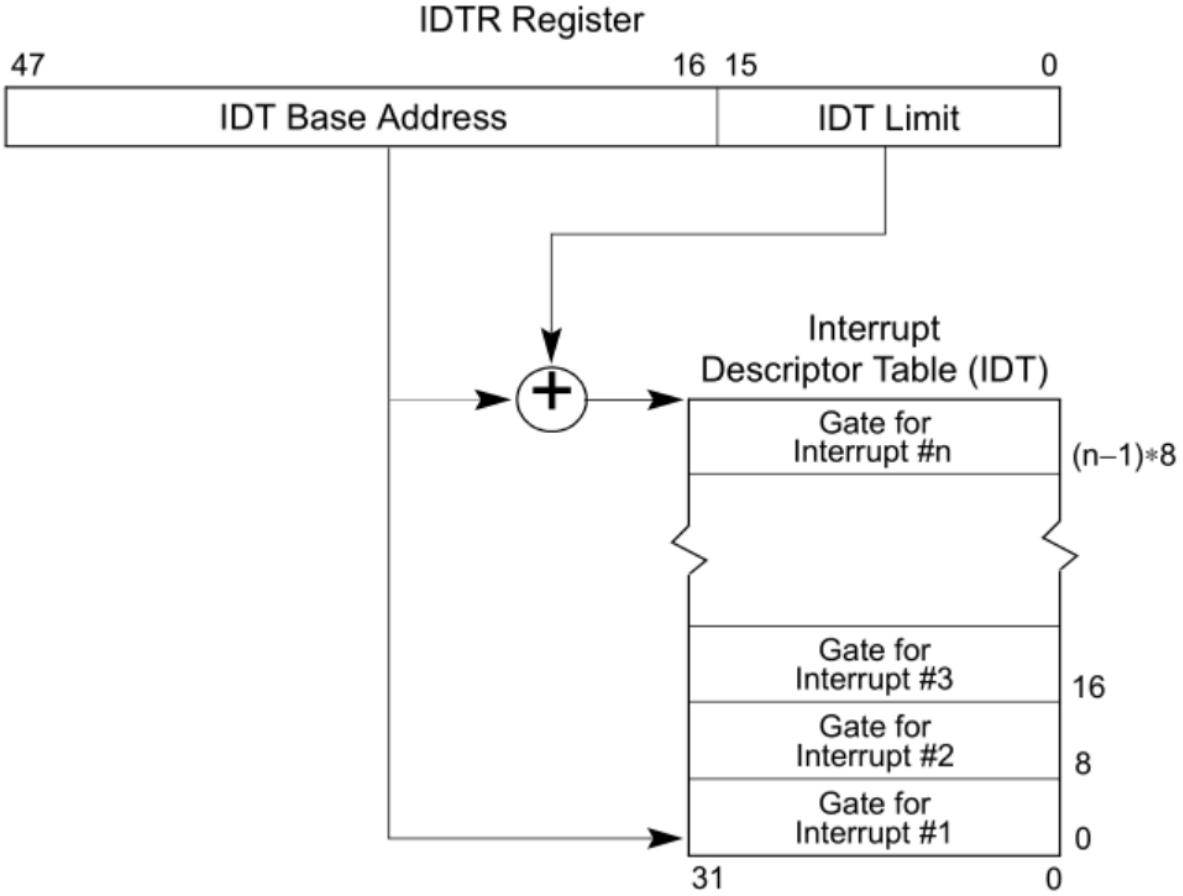
Protected mode operation →

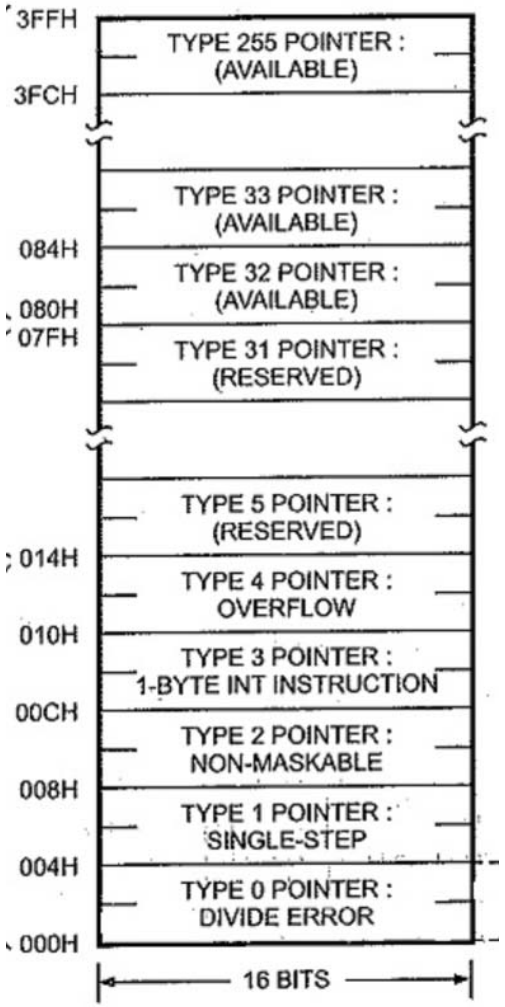
- addresses within and above first 1M byte of memory.
- 32-bit offset address is used to access information →
  - segment can be up to 4G bytes in length.

- segment register contains selector in place of segment address.
- selector →
  - selects descriptor from descriptor table.
- descriptor →
  - describes memory segment's location, length, access rights.

Real Mode vs. Protected Mode Operation

Protected mode operation →



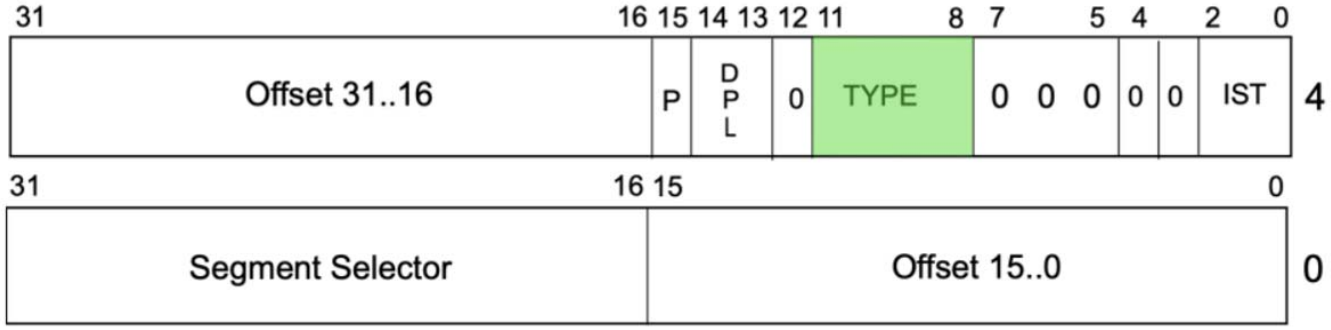


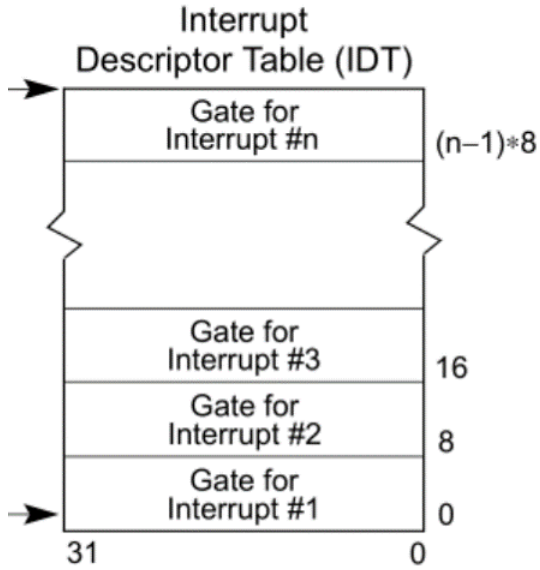
Operation of Protected Mode Interrupt

Interrupts have exactly same assignments as in real mode.
Set of 256 interrupt descriptors are used in place of interrupt vectors.
Descriptors are stored in interrupt descriptor table (IDT).
IDT is located at any memory location in system by IDTR.

Each descriptor contains 8 bytes.
Interrupt descriptor table is 256×8 bytes long.

Structure of each entry in IDT →

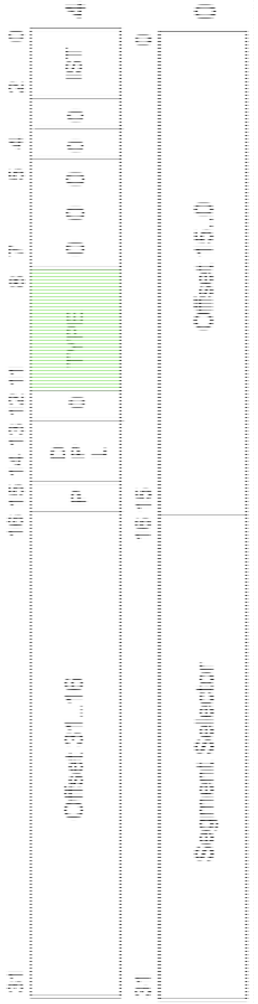




Operation of Protected Mode Interrupt

Structure of each entry in IDT →

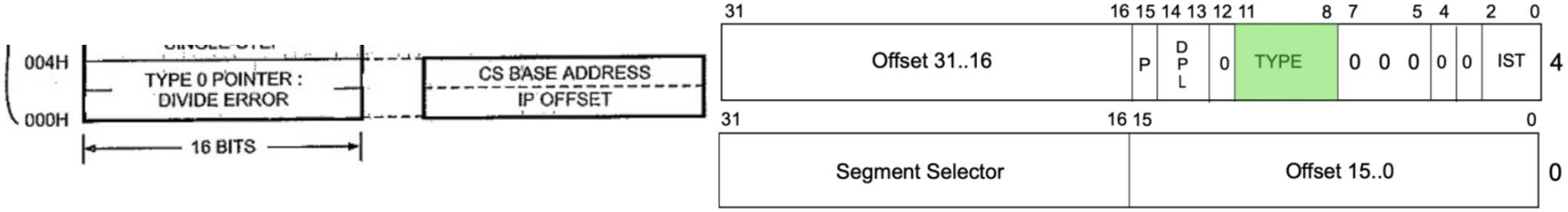
Offset	32-bit value, split in 2 parts. Represents address of entry point of ISP.
Selector	Segment selector, Must point to a valid code segment.
Gate Type	4-bit value. Defines gate type this Interrupt Descriptor represents. Task gate, interrupt gate, trap gate, call gate.
DPL	Descriptor Privilege Level. 2-bit value, Describes privilege level of interrupt.
P	Present bit. Must be set (1) for descriptor to be valid.
IST	Interrupt Stack Table. Introduced in x86-64 (long mode), Used when interrupt happens during stack corruption or overflow.



Operation of Protected Mode Interrupt

Real mode interrupt vectors can be converted into protected mode interrupts.

1) Copy interrupt procedure addresses from IVT.
2) Convert them to 32-bit offset addresses.
3) Store them in interrupt descriptors.



Global descriptor table = Single selector + Segment descriptor.
It identifies first 1M byte of memory as interrupt segment.

Protected mode interrupt functions like real mode interrupt.
We return from both interrupts by using IRET or IRETD instruction.

Microprocessor accesses IDT instead of IVT in protected mode.

Interrupt Flag Bits

IF = 1 →

allows INTR pin to cause interrupt.

IF = 0 →

prevents INTR pin from causing interrupt.

TF = 1 →

causes interrupt Type 1 to occur after each instruction executes.

TF = 0 →

normal program execution occurs.

IF is set by STI and cleared by CLI instructions.
No special instructions to set or clear TF.

Interrupt Flag Bits

```
; a procedure that sets the TRAP flag bit to enable trapping
TRON   PROC
        PUSHF           ; push current FLAGS register onto stack
        POP    AX       ; pop FLAGS into AX to modify flag bits
        OR     AH, 1    ; set trap flag bit 8 (TF = 1)
        PUSH AX         ; push modified FLAGS back to stack
        POPF            ; pop back into FLAGS register
        IRET            ; return from procedure
TRON   ENDP

; a procedure that clears the TRAP flag bit to disable trapping
TROFF   PROC
        PUSHF           ; push current FLAGS register onto stack
        POP    AX       ; pop FLAGS into AX to modify flag bits
        AND    AH, 0FEH ; clear trap flag bit 8 (TF = 0)
        PUSH AX         ; push modified FLAGS back on stack
        POPF            ; restore modified FLAGS register
        IRET            ; return from procedure
TROFF   ENDP
```

Hardware Interrupts

2 hardware interrupt inputs →

(i) Non-maskable interrupt (NMI) →

Type 2 interrupt occurs.

(ii) Interrupt request (INTR) →

Any interrupt vector can be chosen.

Uses interrupt type number 20H – FFH.

00H–1FH (0–31) are reserved for internal and software interrupts.

$\overline{\text{INTA}}$ signal →

used in response to INTR input,

apply vector type number to data bus connections D_7-D_0.

NMI →

edge-triggered input.

requests interrupt on +ve edge (0-to-1 transition).

after +ve edge, NMI pin must remain logic 1 until it is recognized.

before +ve edge, NMI pin must be logic 0 for at least 2 clocking periods.

Hardware Interrupts

INTR →

level-sensitive.
set by external event.

automatically disabled or cleared once it is accepted by microprocessor.

How interrupt vector type number 80H is applied to D_0–D_7 in response to INTR using Three-State Buffer for $\overline{\text{INTA}}$ →