Low Level Programming
Lecture 3

International Faculty of Engineerig, Technical University of Łódź

IA Real Mode Addressing

To segment register CS, DS., SS, ES, FS, GS

	19		4	3			0
SEGMENT	16-bit SEGMENT number (a paragraph)			0	0	0	0

+

To general purpose register AX, BX, CX, DX, SI, DI, SP, BP

	19		16	15		0
OFFSET	0	0	0	0	16-bit OFFSET value	

=

	19	0
PHYSICAL ADDRESS	20-bit physical address (min = 0, max = 1 MB)	
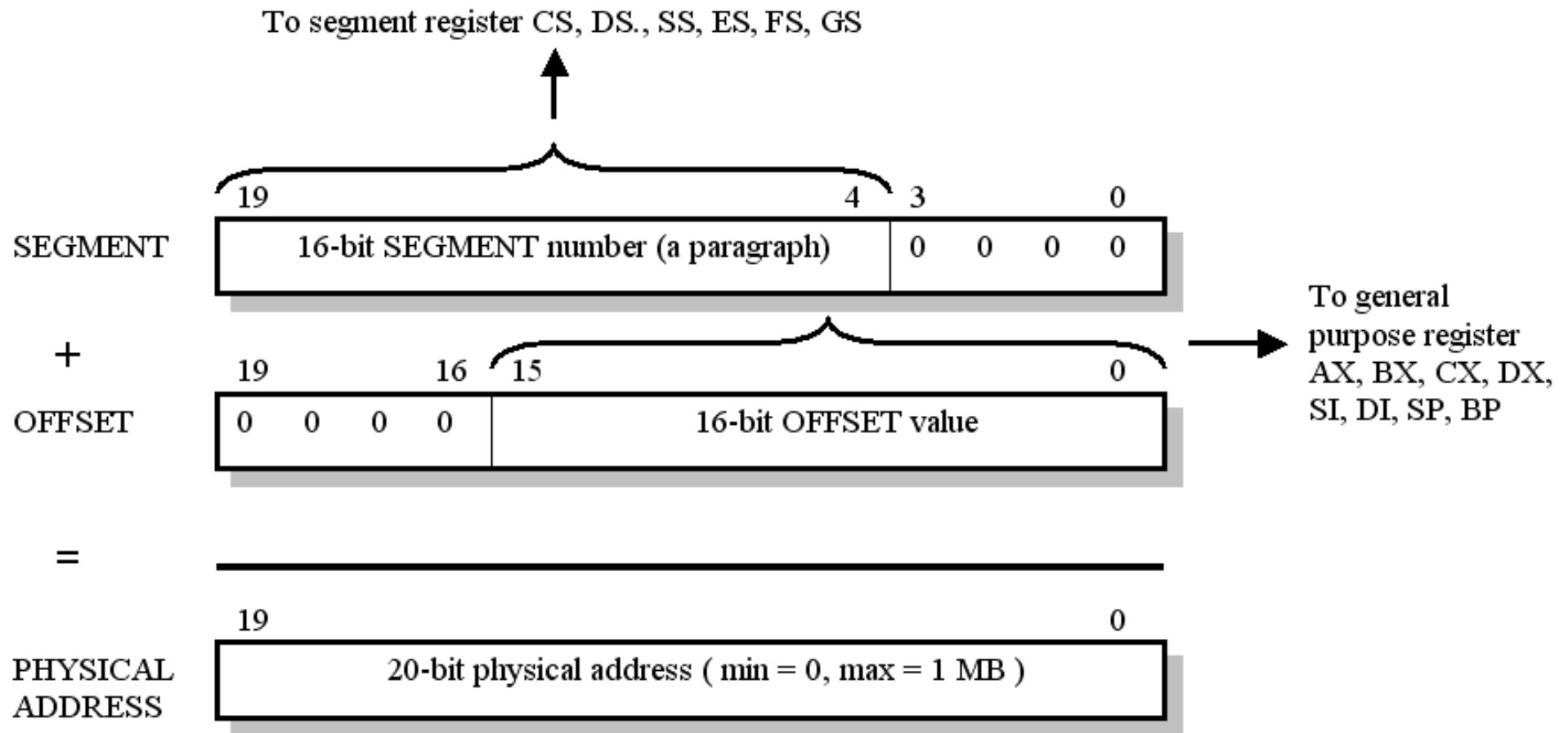
PHYSICAL ADDRESS = 16 * SEGMENT + OFFSET

IA Real Mode Addressing

Each [   ] gap is 16 bytes long, which means that segments overlap in operational memory
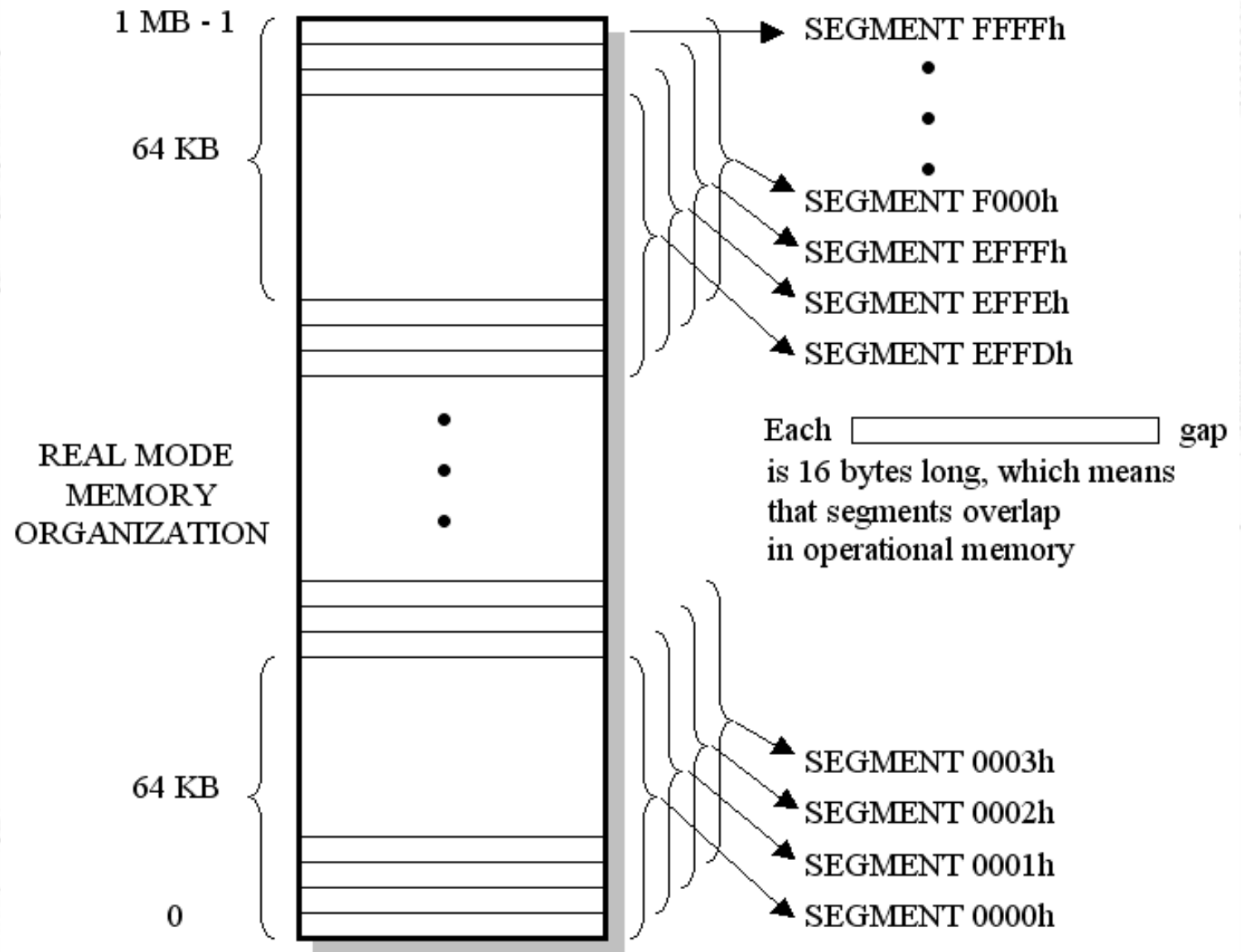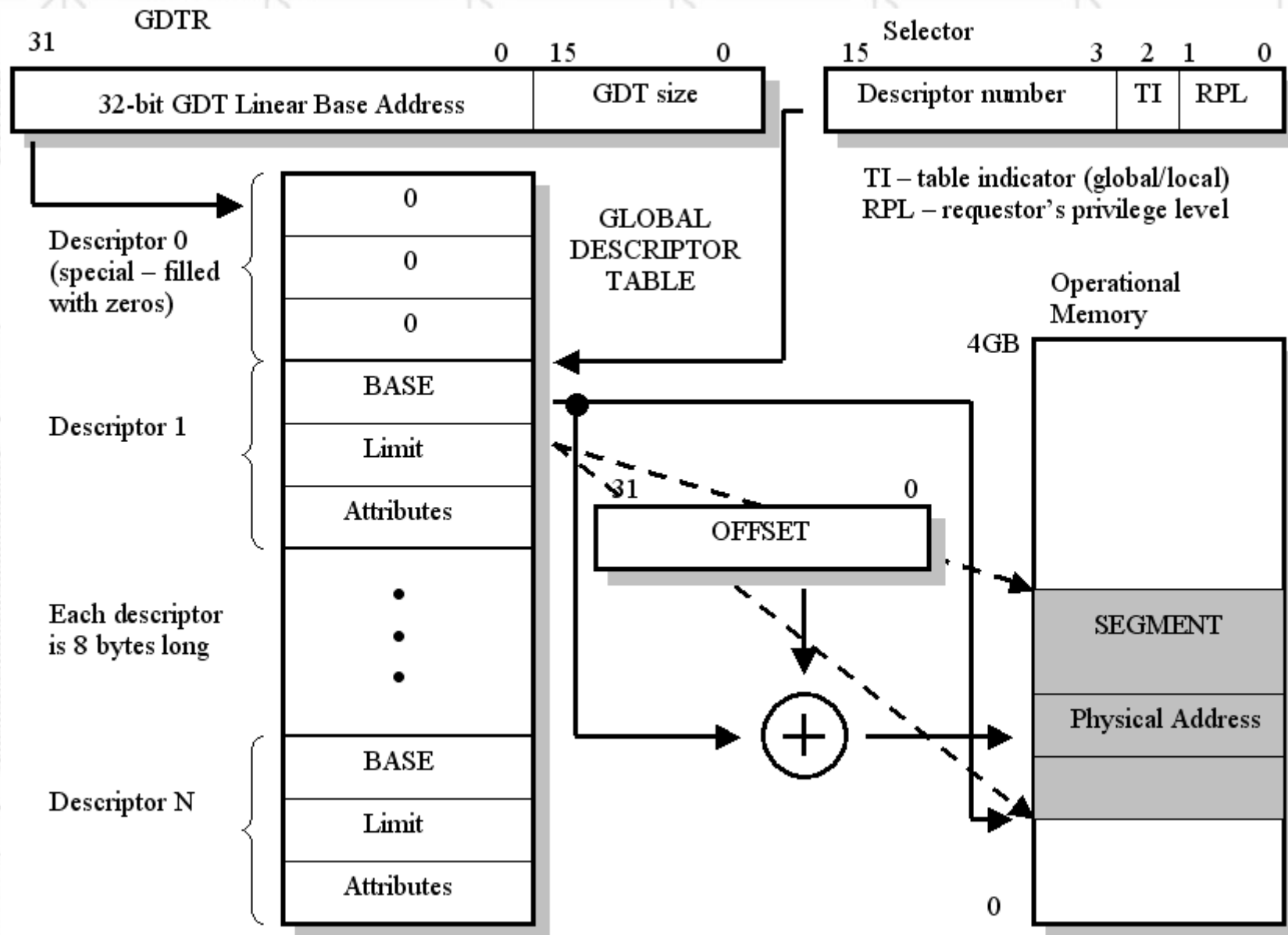
IA Protected Mode

Protected mode is an operational mode of IA-32 compatible processors. It was first introduced in the Intel's 80286 processor, and later extended with the release of other IA family processors. Protected mode allows a (operating) system software to utilize features such as:

- task separation
- safe multitasking
- restricting the application's software to have a direct access to hardware devices

.

IA Protected Mode Addressing

IA Protected Mode

Descriptor Format:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
63	*Base Address* 31..24								*G*	*D/B*	0	*AVL*	*Limit* 19..16				48
47	*P*	*DPL*		*S*	*Type*			*A*	*Base Address* 23..16								32
31	*Base Address* 15..0																16
15	*Limit* 15..0																0

Descriptor tables:
- **GDT** – global descriptor table stores the segments
of global system resources and routines used in protected mode
- **IDT** – interrupt descriptor table register stores the segments
of an interrupt routines used in protected mode
- **LDT** – local descriptor table register stores the segments of
a current task's local resources and routines used in protected mode
.

IA Protected Mode

- **Base Address:** 32-bit linear address where the segment starts
- **Limit:** segment's length
- **G:** granularity; 0 = 1 byte – maximum segment length = 1MB;
1 = 4 KB - maximum segment length = 4GB
- **D/B:** word size; 0 = 16-bit segment; 1 = 32 bit segment
- **S:** descriptor type; 0 = system descriptor; 1 = memory segment descriptor
- **Type**: segment type
0 = data segment (read only)
1 = data segment (read/write)
2 = extendable data segment (read only)
3 = extendable data segment (read/write)
4 = code segment (execute only)
5 = code segment (read/execute)
6 = conformable code segment (execute only)
7 = conformable code segment (read/execute)

IA Protected Mode

- **DPL:** descriptor privilege level
- **P:** segment present in operational memory
- **A:** segment recently accessed

1 = 4 KB - maximum segment length = 4GB

- **AVL:** abvailable to software (NOT USED – for future purposes)

Exemplary GDT Table

.

```
gdt_tab        dq  0                          ; ZERO
               dw -1, 0, 9a00h, 0             ; RM Code
               dw -1, 0, 9200h, 0             ; RM Data
               dw -1, 0, 9a00h, 0040h         ; PM Code
               dw -1, 0, 9200h, 0040h         ; PM Data
               dw 0ffh, 0, 9200h, 0040h       ; PM Stack
```

IA Protected Mode

Entering Protected Mode:

To enter the protected mode one has to accomplish the following steps:
- properly fill the descriptors in the GDT table
- clear the processor's interrupt flag not to allow interrupts to occur
- load processor's *Global Descriptor Table Register* (*GDTR*) with
  GDT address and limit
- switch on the processor to a protected mode (set PE flag in processor's
Control Word Register)
- make a *far jump* to a first instruction in the program's 32-bit code
segment, which is to be executed in the protected mode

IA Protected Mode

Leaving Protected Mode:

To leave the protected mode one has to accomplish the following steps:
- clear the processor's interrupt flag not to allow interrupts to occur
- set all segment registers to 16 – bit segment descriptors
- switch on the processor to a real mode (reset PE flag in processor's Control Word Register)
- make a *far jump* to appropriate instruction in the program's 16-bit code segment

Thank You for today's lecture