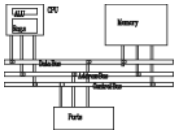
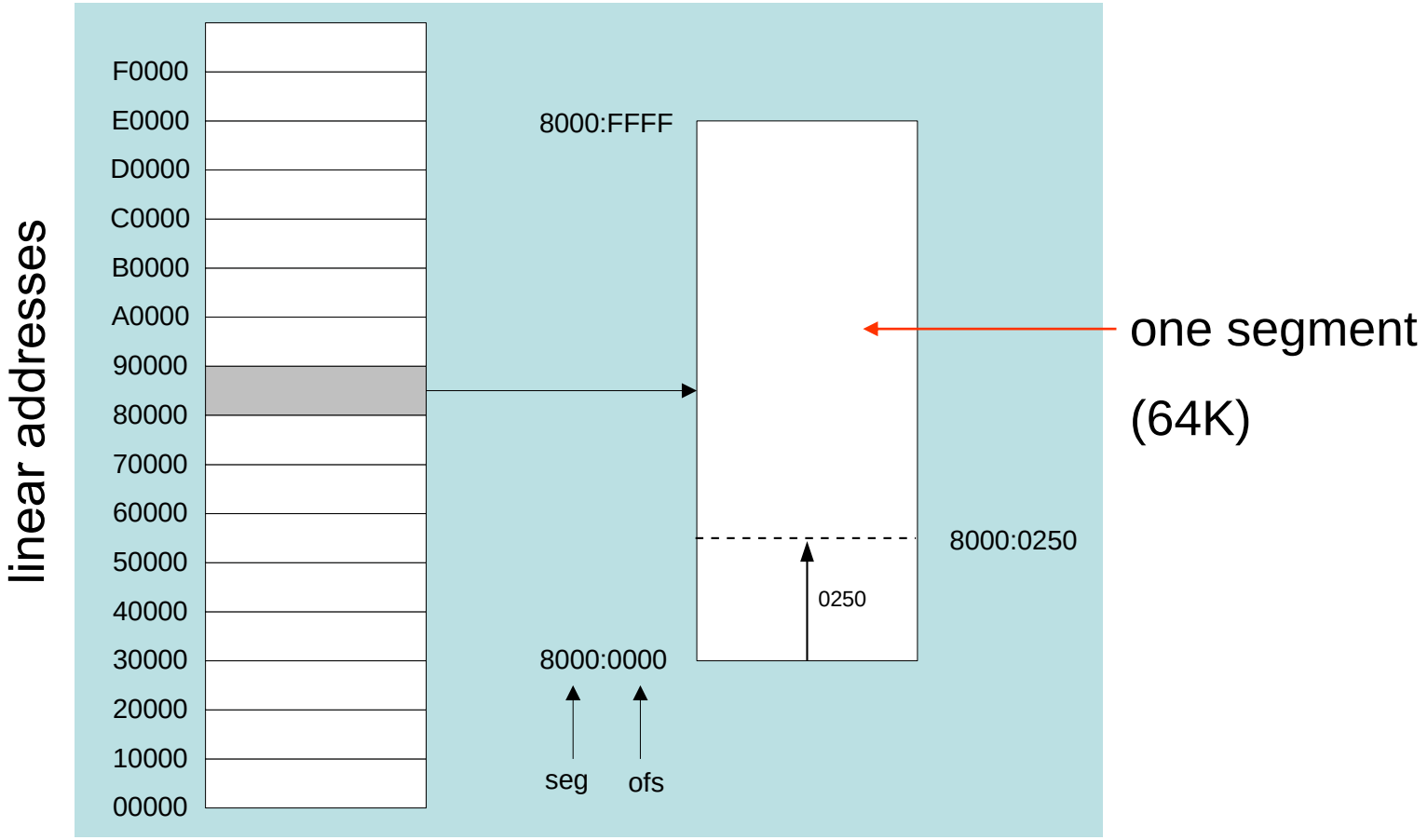
Real-address mode



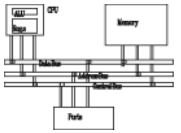
- 1 MB RAM maximum addressable (20-bit address)
- Application programs can access any area of memory
- Single tasking
- Supported by MS-DOS operating system

Real-address Mode: Segmented memory

Segmented memory addressing: absolute (linear) address is a combination of a 16-bit segment value added to a 16-bit offset



Real-address Mode: Calculating linear addresses



- Given a segment address, multiply it by 16 (add a hexadecimal zero), and add it to the offset
- Example: convert 08F1:0100 to a linear address

Adjusted Segment value:	**0**	**8**	**F**	**1**	**0**
Add the offset:		**0**	**1**	**0**	**0**
Linear address:	**0**	**9**	**0**	**1**	**0**

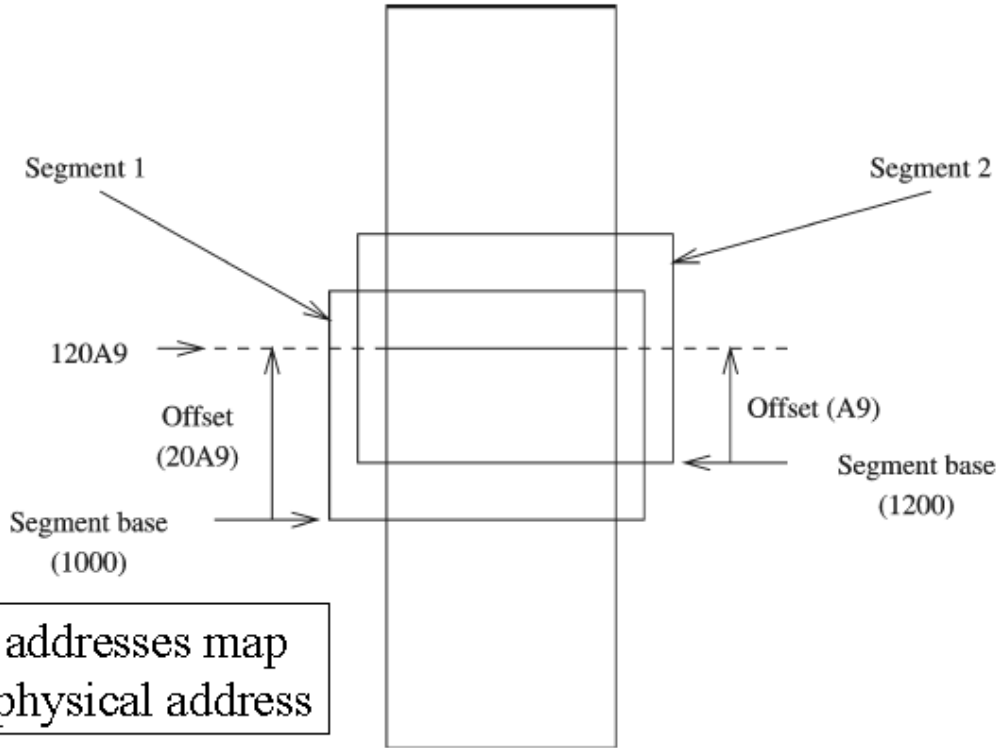
- A typical program has three segments: code, data and stack. Segment registers CS, DS and SS are used to store them separately.

Real-address Mode: Example

What linear address corresponds to the segment/offset address 028F:0030?

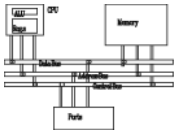
028F0 + 0030 = 02920

Always use hexadecimal notation for addresses.

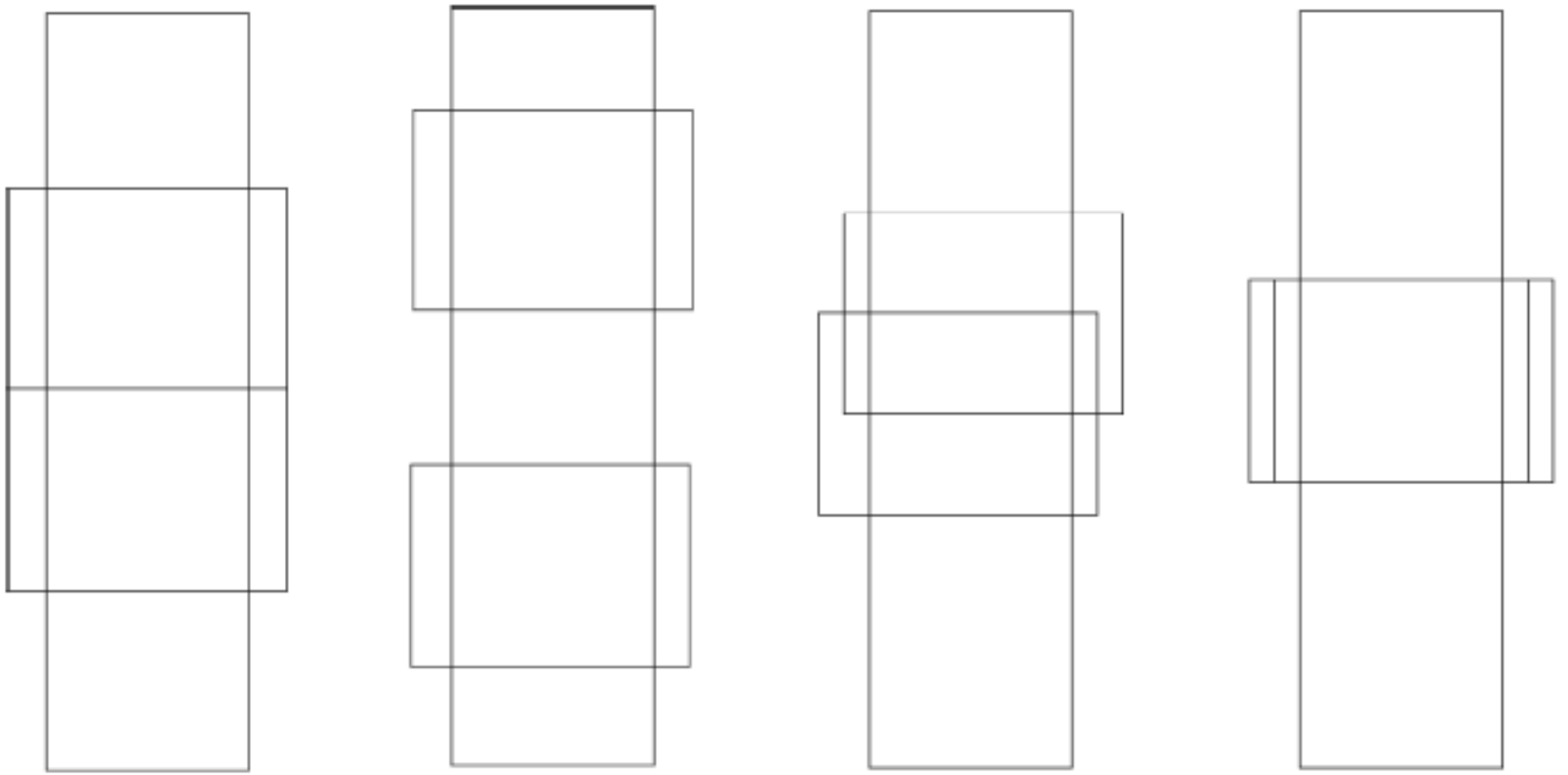


Two logical addresses map to the same physical address

Real-address Mode: Example

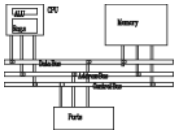


- Segment Overlapping



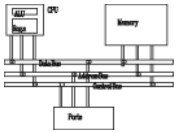
(a) Adjacent (b) Disjoint (c) Partially overlapped (d) Fully overlapped

Protected mode



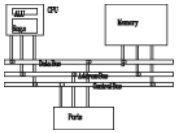
- 4 GB addressable RAM (32-bit address)
  - (00000000 to FFFFFFFFh)
- Each program assigned a memory partition which is protected from other programs
- Designed for multitasking
- Supported by Linux & MS-Windows

Protected mode

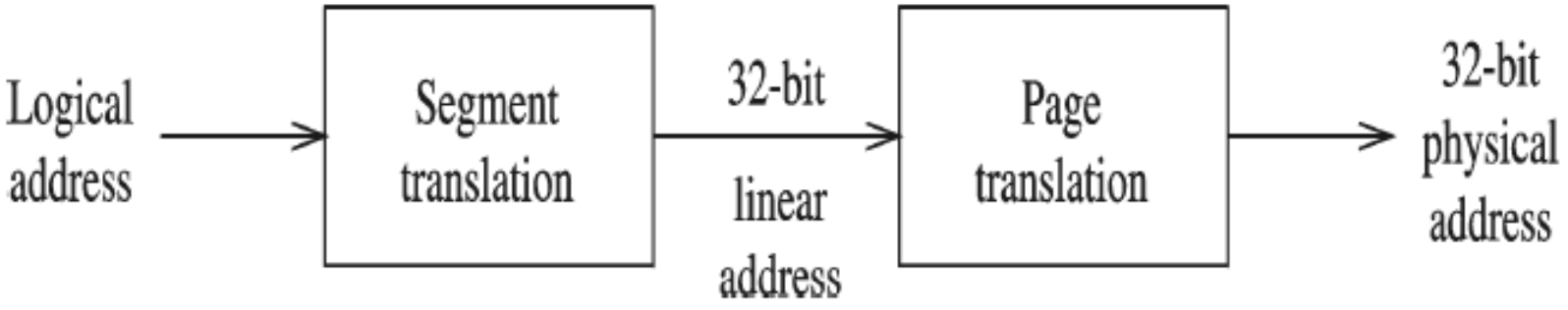


- Started with 80286 up to Pentium, all processors use 2 modes for memory address management
  - Real mode
    - Uses 16-bit addresses
    - Runs 8086 programs
    - Pentium acts as a faster 8086
  - Protected mode
    - 32-bit mode
    - Native mode of Pentium
    - Supports segmentation and paging

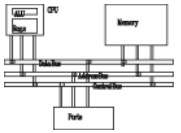
Protected mode



- Supports sophisticated segmentation
- Segment unit translates 32-bit logical address to 32-bit linear address
- Paging unit translates 32-bit linear address to 32-bit physical address
  - If no paging is used
    - **Linear address = physical address**

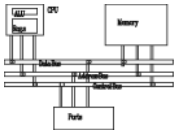


Protected mode

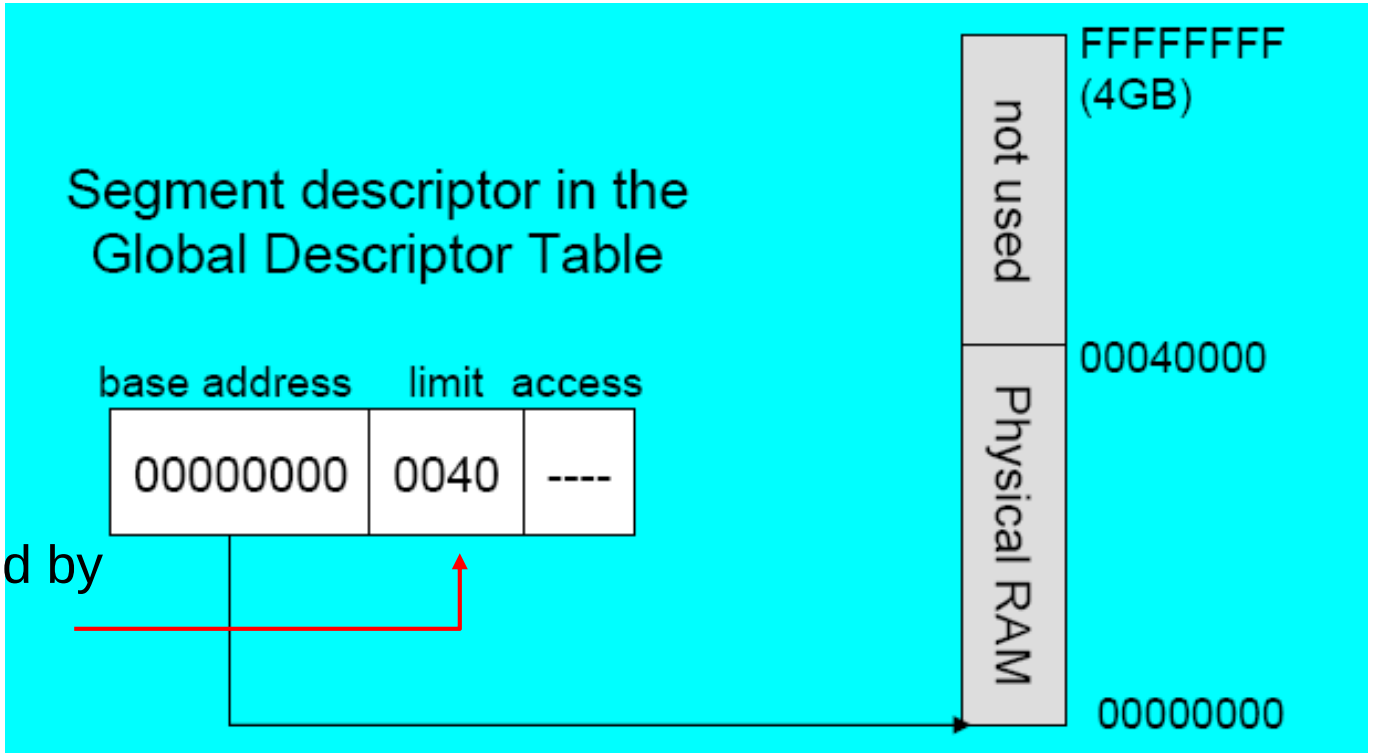


- In this mode there is a Segment Descriptor Table
- Typical Program structure follows:
  - Code, Data, and Stack areas
  - CS, DS, SS segment descriptors
  - Global Descriptor Table (GDT)
  - Local Descriptor Table (LDT)
- MASM Programs use the Microsoft flat memory model

Flat segmentation model

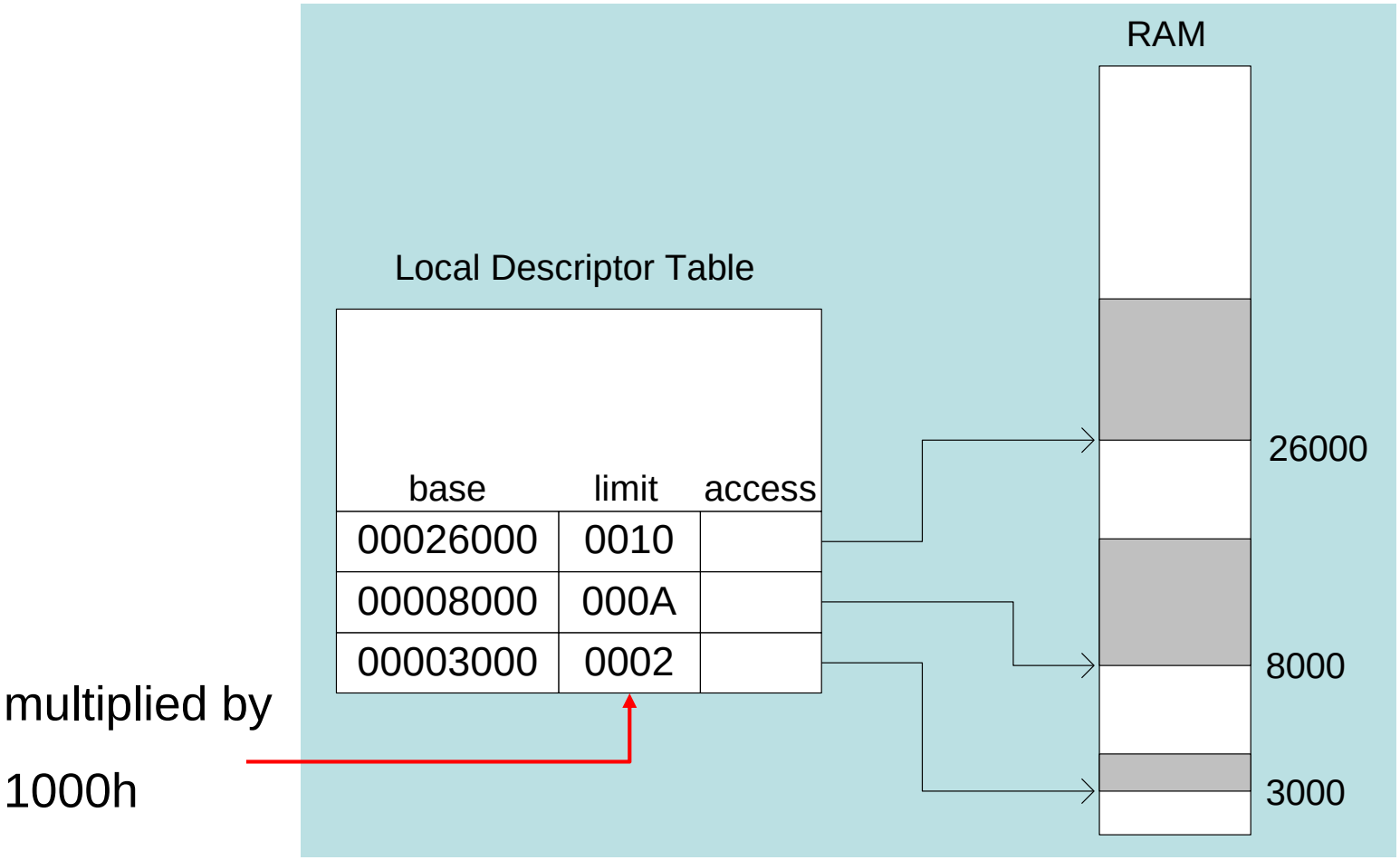


- All segments are mapped to the entire 32-bit physical address space, at least two, one for data and one for code
- Global Descriptor Table (GDT)

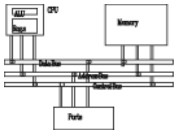


Multi-segment model

- Each program has a local descriptor table (LDT)
  - holds descriptor for each segment used by the program

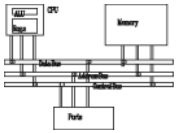


Translating Addresses

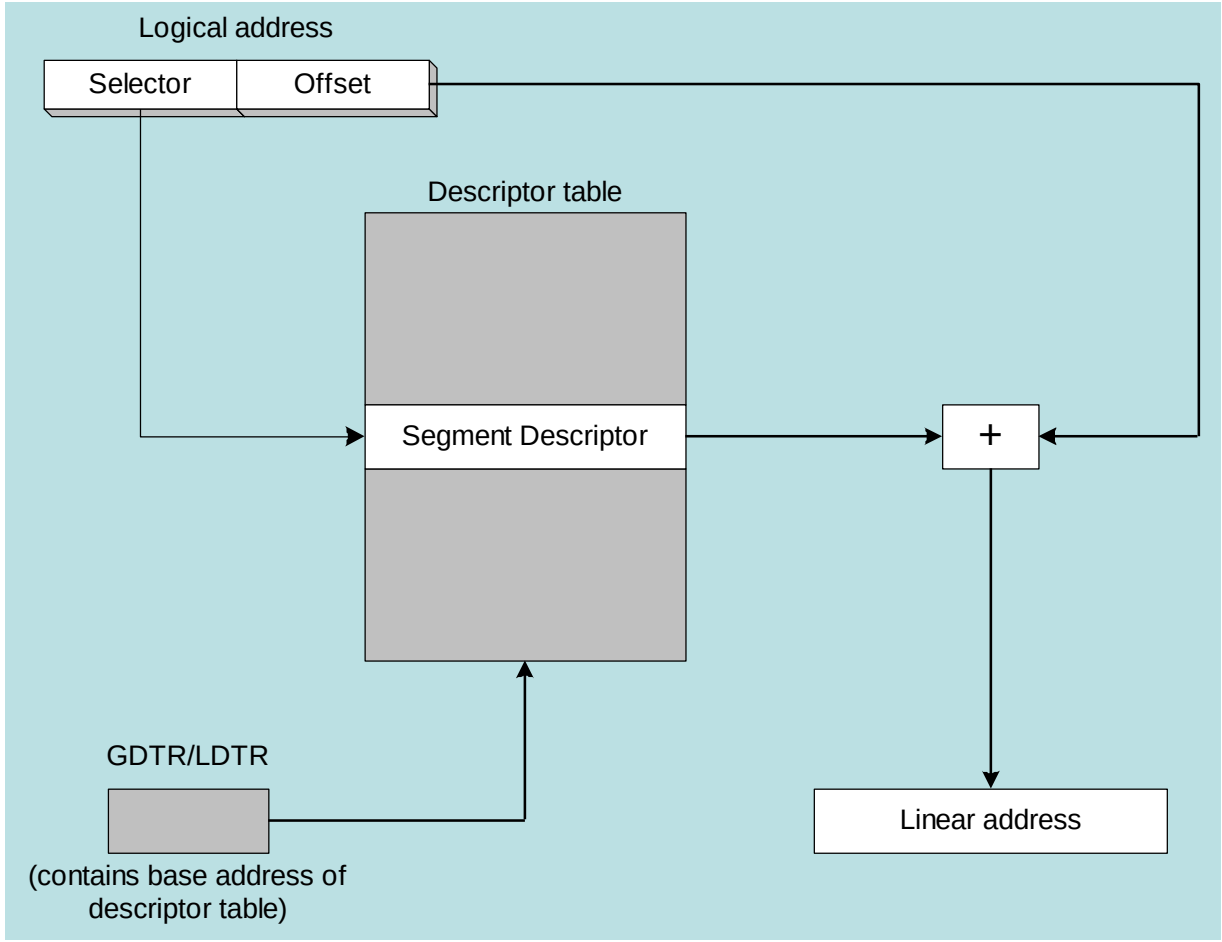


- The processor uses a one- or two-step process to convert a variable's logical address into a unique memory location.
- The first step combines a segment value with a variable’s offset to create a linear address.
- The second optional step, called page translation, converts a linear address to a physical address.

Converting Logical to Linear Address

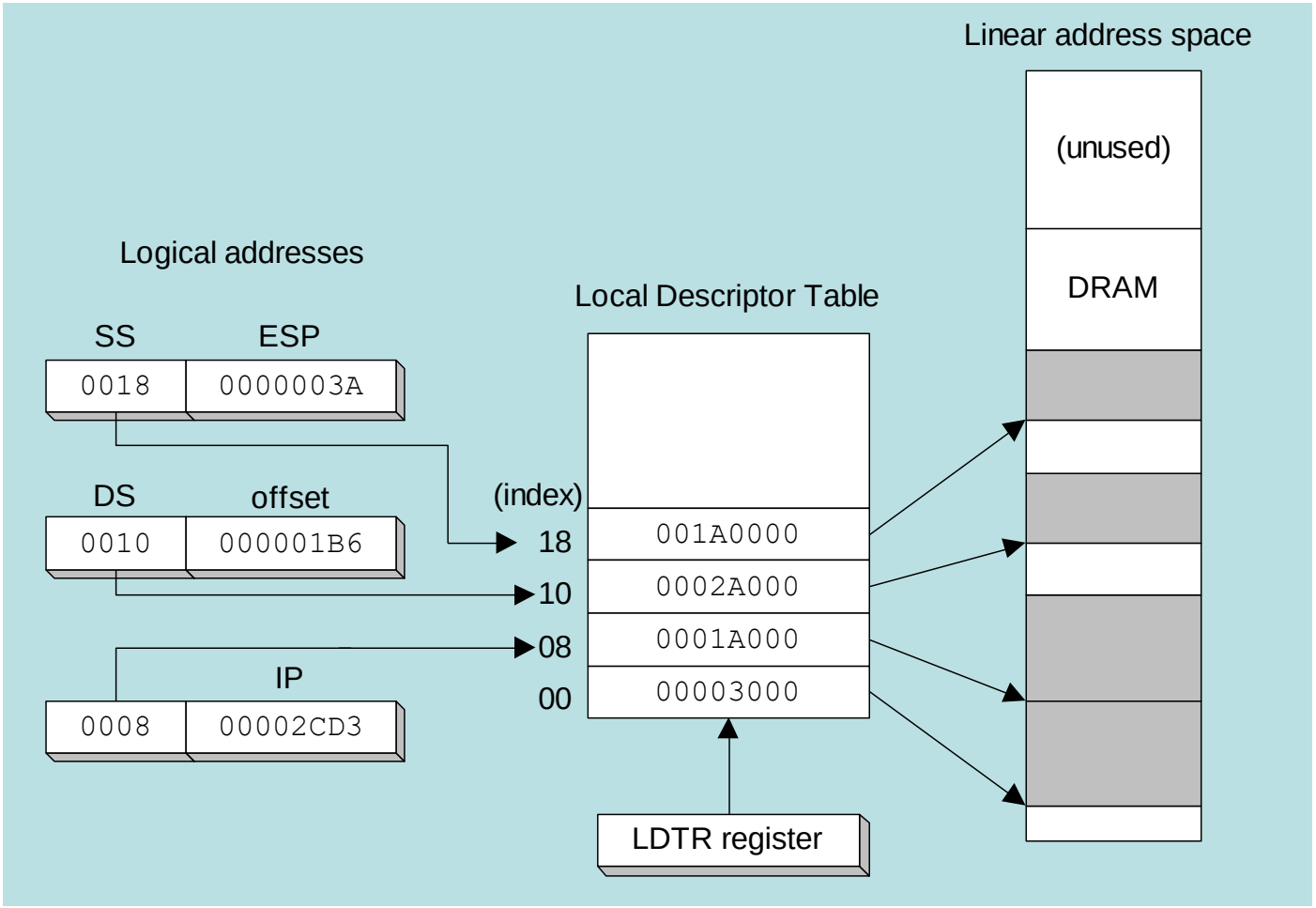


The segment selector points to a segment descriptor, which contains the base address of a memory segment. The 32-bit offset from the logical address is added to the segment's base address, generating a 32-bit linear address.

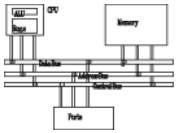


Indexing into a Descriptor Table

Each segment descriptor indexes into the program's local descriptor table (LDT). Each table entry is mapped to a linear address:

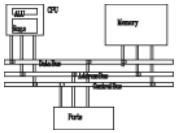


Paging



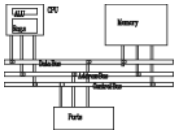
- Virtual memory uses disk as part of the memory, thus allowing sum of all programs can be larger than physical memory
- Only part of a program must be kept in memory, while the remaining parts are kept on disk.
- The memory used by the program is divided into small units called pages (4096-byte).
- As the program runs, the processor selectively unloads inactive pages from memory and loads other pages that are immediately required.

Paging



- OS maintains **page directory** and **page tables**
- **Page Translation:** CPU converts the linear address into a physical address
- **Page Fault:** Occurs when a needed page is not in memory, and the CPU interrupts the program
- **Virtual Memory Manager (VMM):** OS utility that manages the loading and unloading of pages
- OS copies the page into memory, program resumes execution

Page Translation



A linear address is divided into a page directory field, page table field, and page frame offset. The CPU uses all three to calculate the physical address.