EE222 – Microprocessor Systems

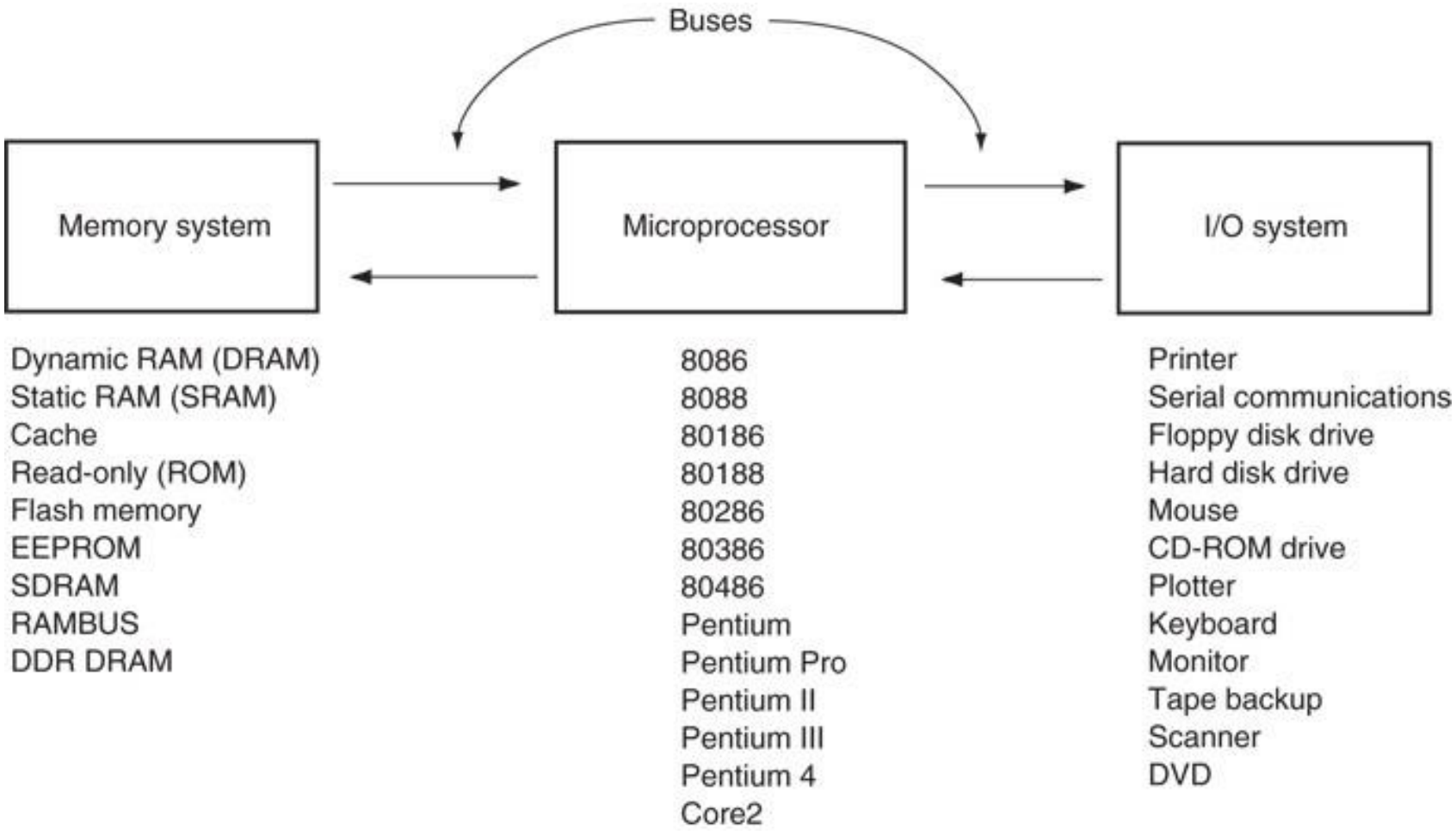
Memory System

Arbab Latif

Spring 2022

Resources:
The Intel Microprocessors: Architecture, Programming, and Interfacing, Eighth Edition Barry B. Brey (**Section 1.2, 2.2**)

Figure 1–6 The block diagram of a microprocessor-based computer system.



The Memory and I/O System

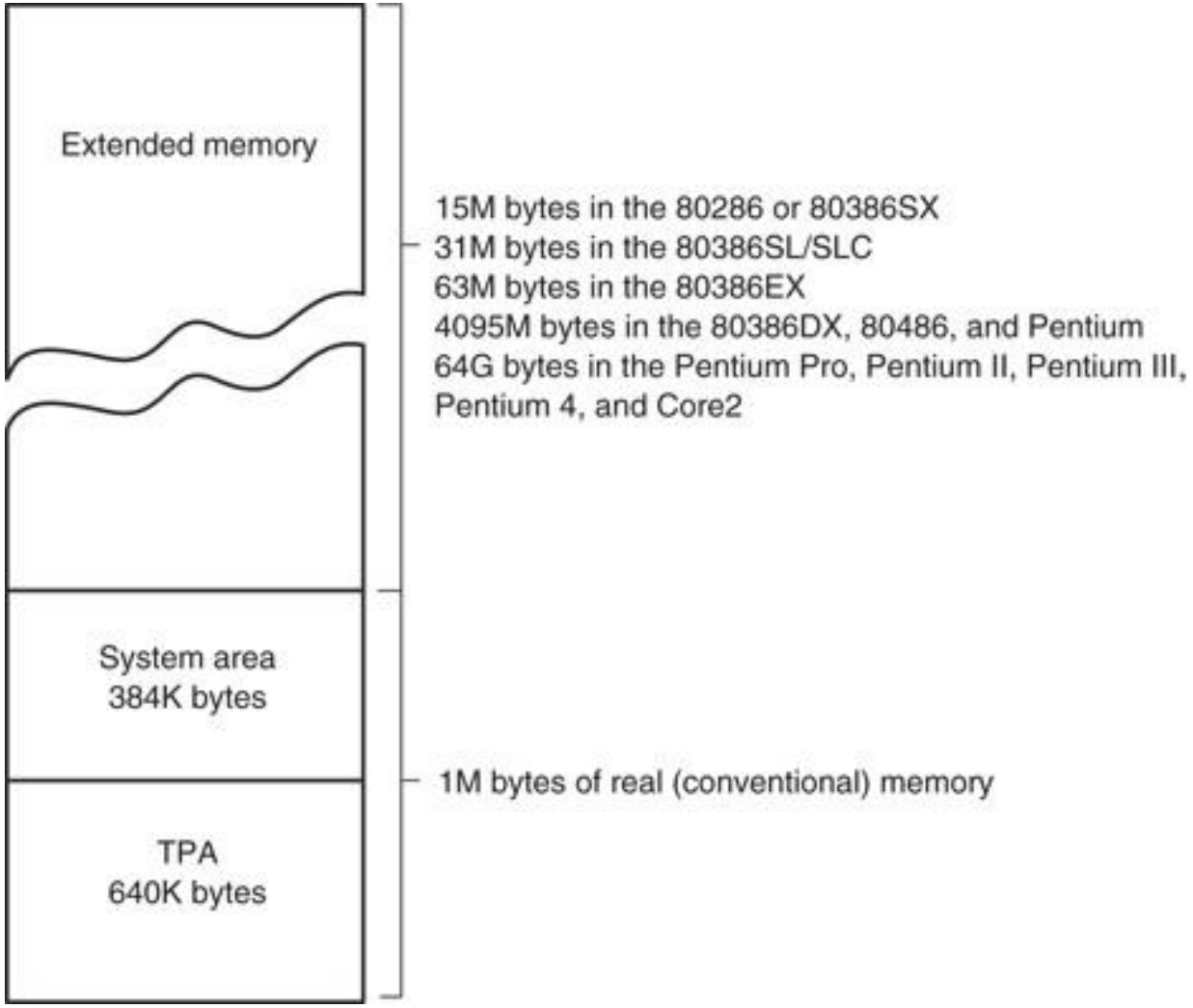


Figure 1–7 The memory map of a personal computer.

- Main memory system divided into three parts:
  - TPA (transient program area)
  - system area
  - XMS (extended memory system)
- Type of microprocessor present determines whether an extended memory system exists.
- First 1M byte of memory often called the real or conventional memory system.
  - Intel microprocessors designed to function in this area using real mode operation

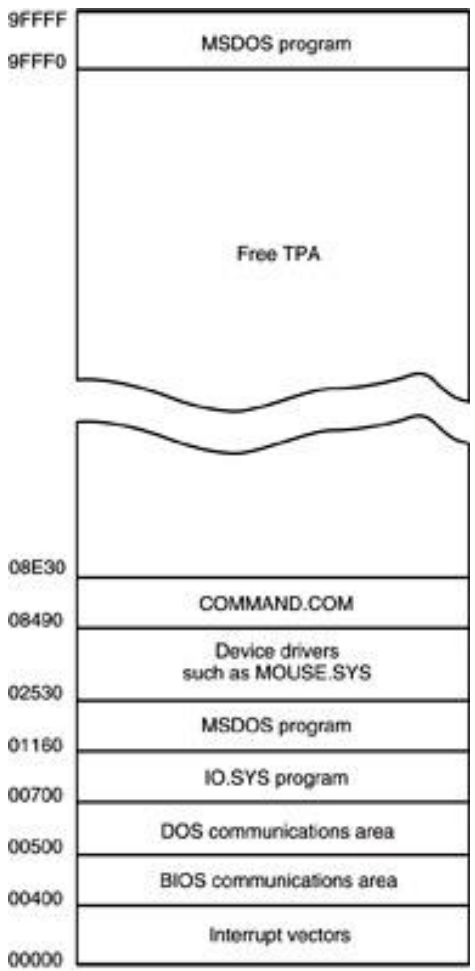
- Pentium and ATX class machines feature addition of the PCI (**peripheral component interconnect**) bus.
- Many 80486 systems use **VESA** local, VL bus to interface disk and video to the microprocessor at the local bus level.
  - allows 32-bit interfaces to function at same clocking speed as the microprocessor
  - recent modification supporting 64-bit data bus has generated little interest

- USB (**universal serial bus**).
  - intended to connect peripheral devices to the microprocessor through a serial data path and a twisted pair of wires
  - Data transfer rates are 10 Mbps for USB1.
  - Increase to 480 Mbps in USB2.
  - Increase to 5 Gbps in USB3.
    AGP (**advanced graphics port**) for video cards.
  - The port transfers data between video card and microprocessor at higher speeds.
  - 66 MHz, with 64-bit data path
- New buses are serial ATA interface (**SATA**) for hard disk drives; PCI Express bus for the video card.

The TPA

- The transient program area (TPA) holds the DOS (**disk operating system**) operating system; other programs that control the computer system.
  - the TPA is a DOS concept and not really applicable in Windows
  - also stores any currently active or inactive DOS application programs
  - length of the TPA is 640K bytes

Figure 1–8 The memory map of the TPA in a personal computer. (Note that this map will vary between systems.)



- DOS memory map shows how areas of TPA are used for system programs, data and drivers.
  - also shows a large area of memory available for application programs
  - hexadecimal number to left of each area represents the memory addresses that begin and end each data area

- Interrupt vectors access DOS, BIOS (basic I/O system), and applications.
- Areas contain transient data to access I/O devices and internal features of the system.
  - these are stored in the TPA so they can be changed as DOS operates
- The IO.SYS loads into the TPA from the disk whenever an MSDOS system is started.
- IO.SYS contains programs that allow DOS to use keyboard, video display, printer, and other I/O devices often found in computers.
- The IO.SYS program links DOS to the programs stored on the system BIOS ROM.

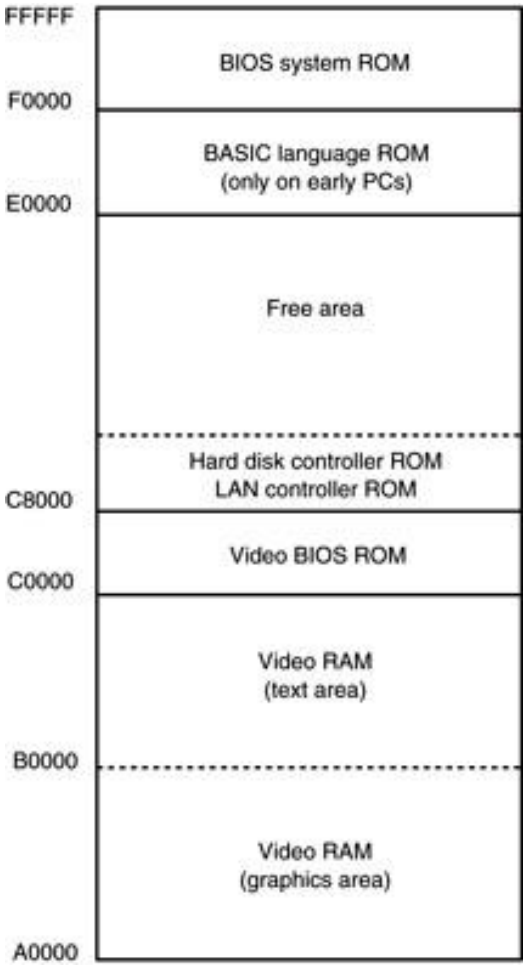
- **Drivers** are programs that control installable I/O devices.
  - mouse, disk cache, hand scanner, CD-ROM memory (**Compact Disk Read-Only Memory**), DVD (**Digital Versatile Disk**), or installable devices, as well as programs
- Installable drivers control or drive devices or programs added to the computer system.
- DOS drivers normally have an extension of .SYS; MOUSE.SYS.
- DOS version 3.2 and later files have an extension of .EXE; EMM386.EXE.

- COMMAND.COM (**command processor**) controls operation of the computer from the keyboard when operated in the DOS mode.
- COMMAND.COM processes DOS commands as they are typed from the keyboard.
- If COMMAND.COM is erased, the computer cannot be used from the keyboard in DOS mode.
  - never erase COMMAND.COM, IO.SYS, or MSDOS.SYS to make room for other software
  - your computer will not function

The System Area

- Smaller than the TPA; just as important.
- The **system area** contains programs on read-only (ROM) or flash memory, and areas of read/write (RAM) memory for data storage.
- Figure 1–9 shows the system area of a typical personal computer system.
- As with the map of the TPA, this map also includes the hexadecimal memory addresses of the various areas.

Figure 1–9 The system area of a typical personal computer.



- First area of system space contains video display RAM and video control programs on ROM or flash memory.
  - area starts at location A0000H and extends to C7FFFH
  - size/amount of memory depends on type of video display adapter attached

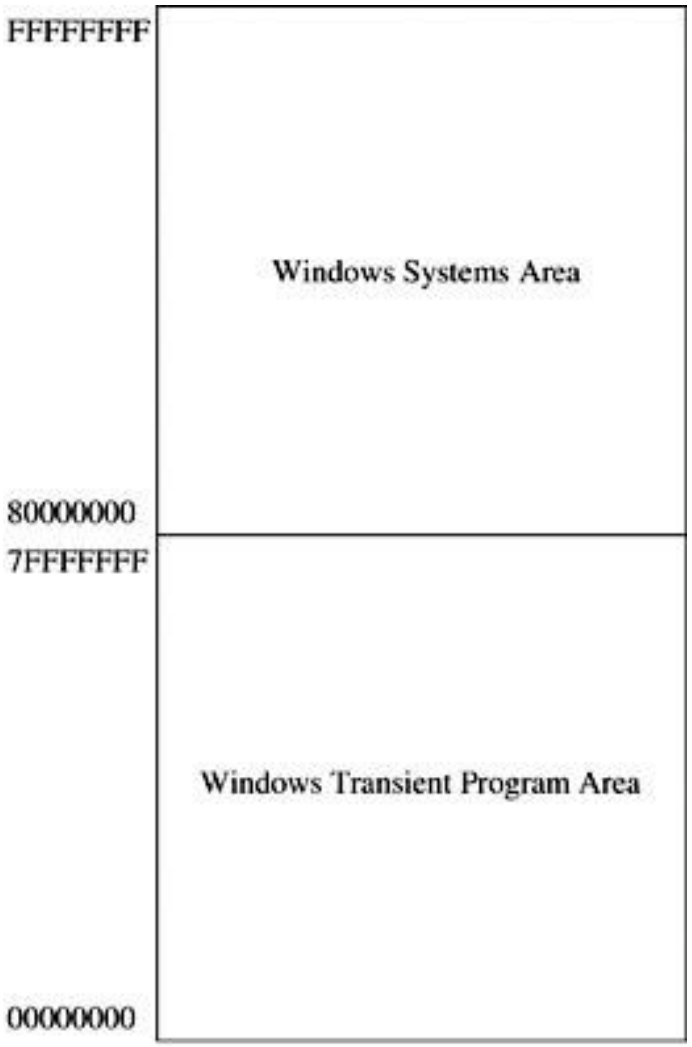
- Display adapters generally have video RAM at A0000H–AFFFFH.
  - stores graphical or bit-mapped data
- Memory at B0000H–BFFFFH stores text data.
- The video BIOS on a ROM or flash memory, is at locations C0000H–C7FFFH.
  - contains programs to control DOS video display
- C8000H–DFFFFH is often open or free.
  - used for expanded memory system (EMS) in PC or XT system; upper memory system in an AT

- System BIOS ROM is located in the top 64K bytes of the system area (F0000H–FFFFFH).
  - controls operation of basic I/O devices connected to the computer system
  - does not control operation of video
- The first part of the system BIOS (F0000H–F7FFFH) often contains programs that set up the computer.
- Second part contains procedures that control the basic I/O system.

Windows Systems

- Modern computers use a different memory map with Windows than DOS memory maps.
- The Windows memory map in Figure 1–10 has two main areas; a TPA and system area.
- The difference between it and the DOS memory map are sizes and locations of these areas.

Figure 1–10 The memory map used by Windows XP.



- TPA is first 2G bytes from locations 00000000H to 7FFFFFFFH.
- Every Windows program can use up to 2G bytes of memory located at linear addresses 00000000H through 7FFFFFFFH.
- System area is last 2G bytes from 80000000H to FFFFFFFFH.

I/O Space

- I/O devices allow the microprocessor to communicate with the outside world.
- I/O (input/output) space in a computer system extends from I/O port 0000H to port FFFFH.
  - **I/O port address** is similar to a memory address
  - instead of memory, it addresses an I/O device
- Figure 1–11 shows the I/O map found in many personal computer systems.

Figure 1–11 Some I/O locations in a typical personal computer.

- Access to most I/O devices should always be made through Windows, DOS, or BIOS function calls.
- The map shown is provided as a guide to illustrate the I/O space in the system.

The Microprocessor

- Called the CPU (**central processing unit**).
- The controlling element in a computer system.
- Controls memory and I/O through connections called buses.
  - buses select an I/O or memory device, transfer data between I/O devices or memory and the microprocessor, control I/O and memory systems
- Memory and I/O controlled via instructions stored in memory, executed by the microprocessor.

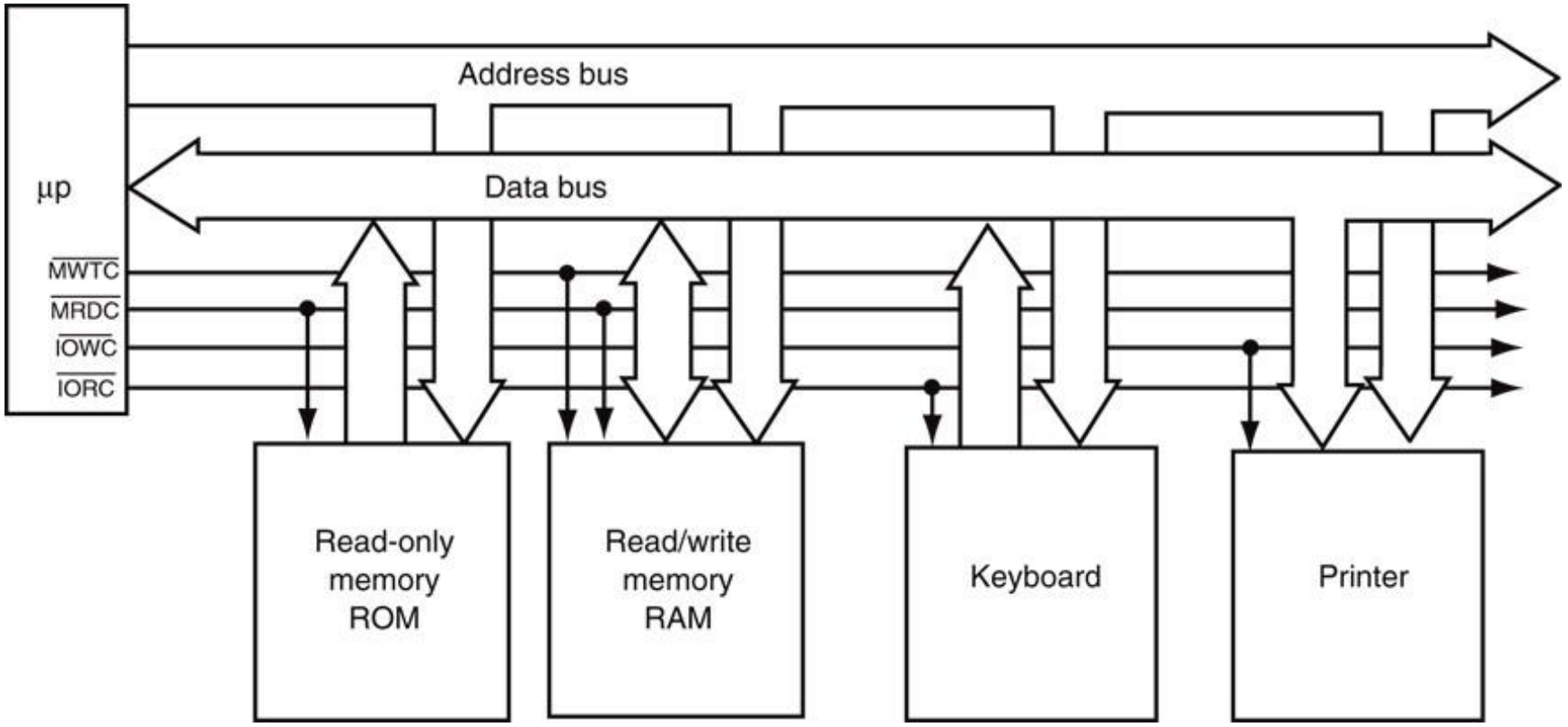
- Microprocessor performs three main tasks:
  - data transfer between itself and the memory or I/O systems
  - simple arithmetic and logic operations
  - program flow via simple decisions
- Power of the microprocessor is capability to execute billions of millions of instructions per second from a program or software (**group of instructions**) stored in the memory system.
  - stored programs make the microprocessor and computer system very powerful devices

- Another powerful feature is the ability to make simple decisions based upon numerical facts.
  - a microprocessor can decide if a number is zero, positive, and so forth
- These decisions allow the microprocessor to modify the program flow, so programs appear to think through these simple decisions.

Buses

- A common group of wires that interconnect components in a computer system.
- Transfer address, data, & control information between microprocessor, memory and I/O.
- Three buses exist for this transfer of information: address, data, and control.
- Figure 1–12 shows how these buses interconnect various system components.

Figure 1–12 The block diagram of a computer system showing the address, data, and control bus structure.



- The address bus requests a memory location from the memory or an I/O location from the I/O devices.
  - if I/O is addressed, the address bus contains a 16-bit I/O address from 0000H through FFFFH.
  - if memory is addressed, the bus contains a memory address, varying in width by type of microprocessor.
- 64-bit extensions to Pentium provide 40 address pins, allowing up to 1T byte of memory to be accessed.

- Advantage of a wider data bus is speed in applications using wide data.
- Figure 1–13 shows memory widths and sizes of 8086 through Core2 microprocessors.
- In all Intel microprocessors family members, memory is numbered by byte.
- Pentium through Core2 microprocessors contain a 64-bit-wide data bus.

Figure 1–13a The physical memory systems of the 8086 through the Core2 microprocessors.

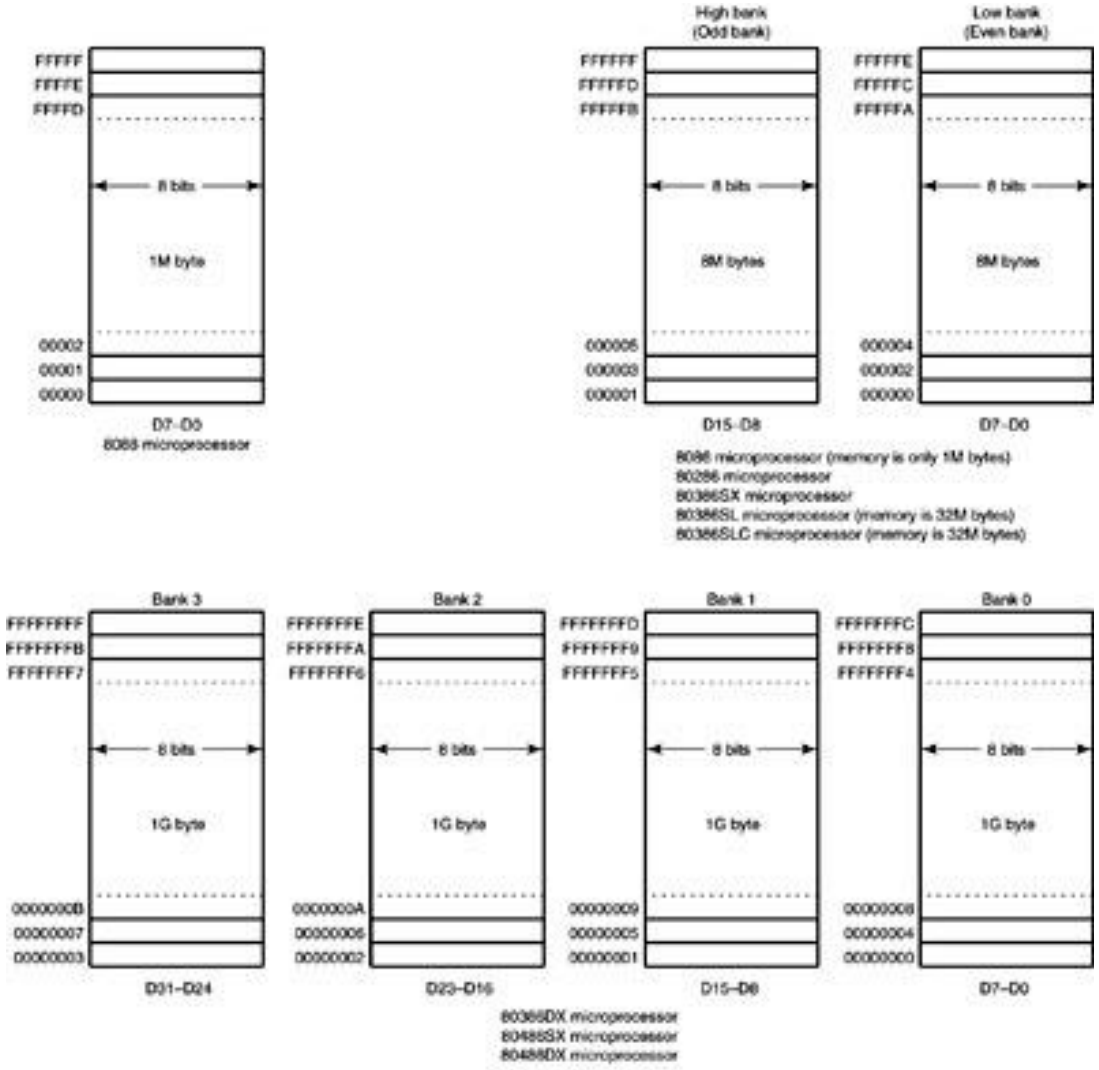
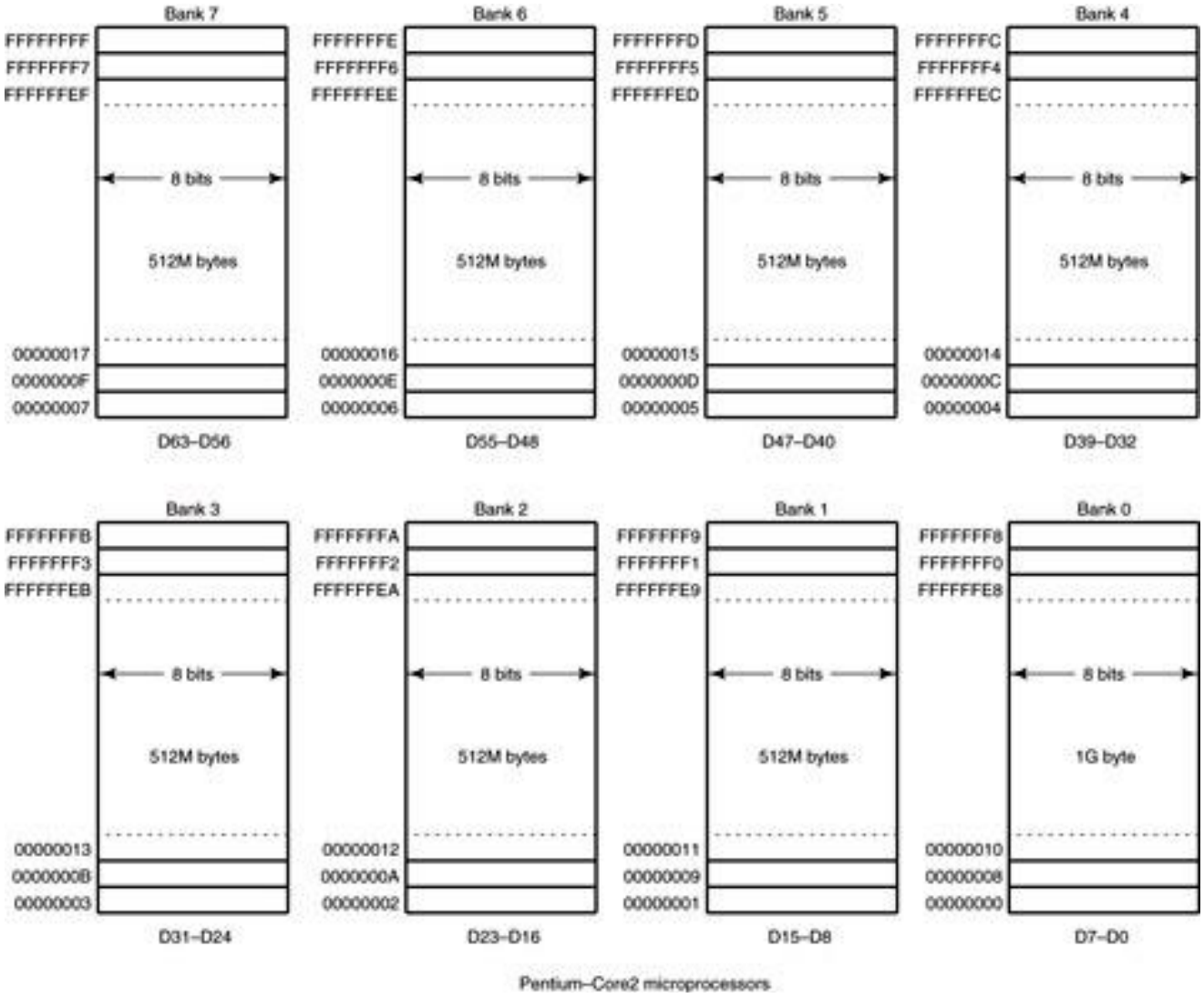


Figure 1–13b The physical memory systems of the 8086 through the Core2 microprocessors.



- Control bus lines select and cause memory or I/O to perform a read or write operation.
- In most computer systems, there are four control bus connections:
- $\overline{MRDC}$ (**memory read control**)
- $\overline{MWTC}$ (**memory write control**)
- $\overline{IORC}$ (**I/O read control**)
- $\overline{IOWC}$ (**I/O write control**).
- overbar indicates the control signal is active-low; (active when logic zero appears on control line)

- The microprocessor reads a memory location by sending the memory an address through the address bus.
- Next, it sends a memory read control signal to cause the memory to read data.
- Data read from memory are passed to the microprocessor through the data bus.
- Whenever a memory write, I/O write, or I/O read occurs, the same sequence ensues.

Word-Sized Data

- A word (16-bits) is formed with two bytes of data.
- The least significant byte always stored in the lowest-numbered memory location.
- Most significant byte is stored in the highest.
- This method of storing a number is called the **little endian** format.

EE222 – Microprocessor Systems

Real Mode Memory System

Arbab Latif

Spring 2022

Resources:
The Intel Microprocessors: Architecture, Programming, and Interfacing, Eighth Edition Barry B. Brey (**Section 2.2**)

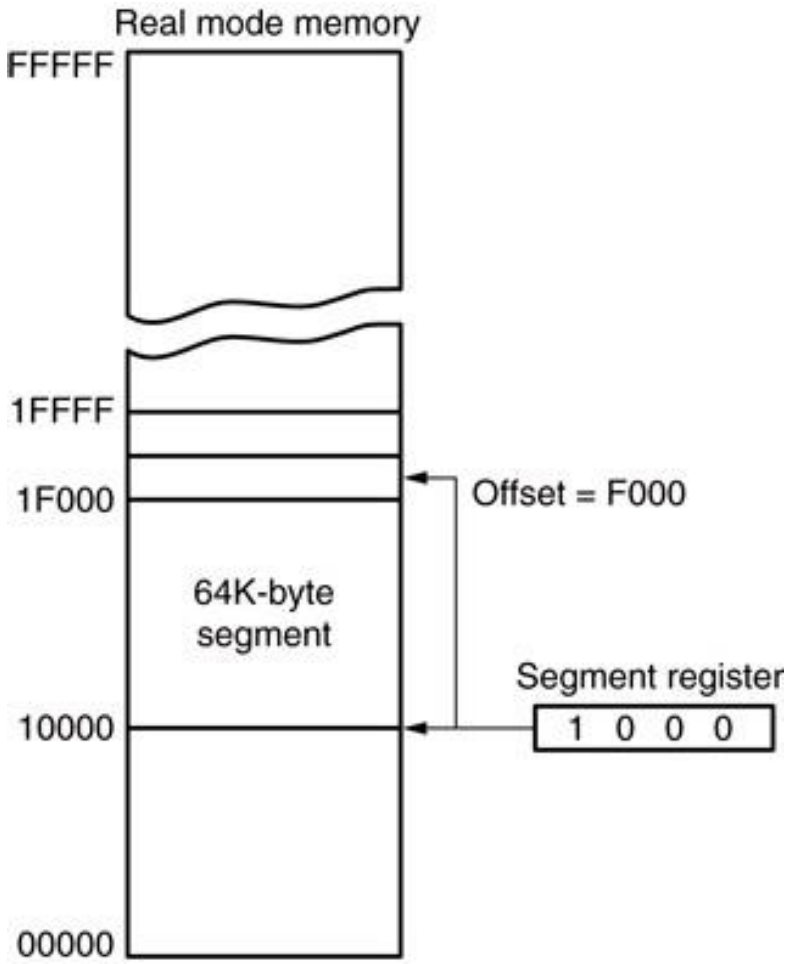
2–2 REAL MODE MEMORY ADDRESSING

- 80286 and above operate in either the real or protected mode.
- **Real mode operation** allows addressing of only the first 1M byte of memory space—even in Pentium 4 or Core2 microprocessor.
  - the first 1M byte of memory is called the **real memory**, **conventional memory**, or **DOS memory** system

Segments and Offsets

- All real mode memory addresses must consist of a segment address plus an offset address.
  - **segment address** defines the beginning address of any 64K-byte memory segment
  - **offset address** selects any location within the 64K byte memory segment
- Figure 2–3 shows how the **segment plus offset** addressing scheme selects a memory location.

Figure 2–3 The real mode memory-addressing scheme, using a segment address plus an offset.



- **this shows a memory segment beginning at 10000H, ending at location IFFFFH**
  - **64K bytes in length**
- **also shows how an offset address, called a displacement, of F000H selects location 1F000H in the memory**

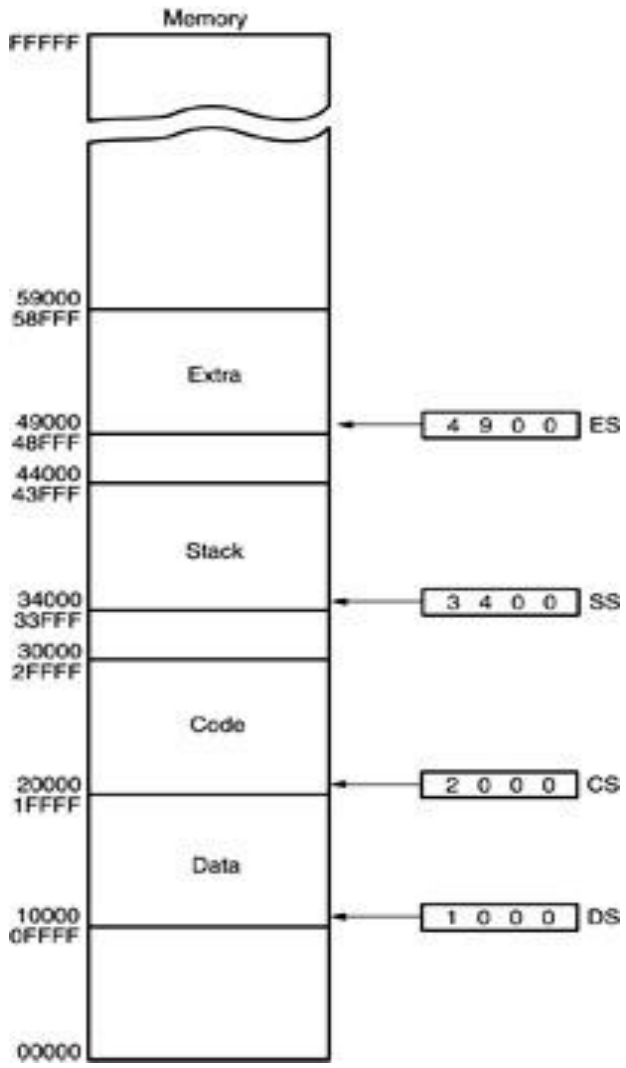
- Once the beginning address is known, the **ending address** is found by adding FFFFH.
  - because a real mode segment of memory is64K in length
- The offset address is always added to the segment starting address to locate the data.
- Segment and offset address is sometimes written as 1000:2000.
  - a segment address of 1000H; an offset of 2000H

Default Segment and Offset Registers

- The microprocessor has rules that apply to segments whenever memory is addressed.
  - these define the segment and offset register combination
- The **code segment** register defines the start of the code segment.
- The **instruction pointer** locates the next instruction within the code segment.

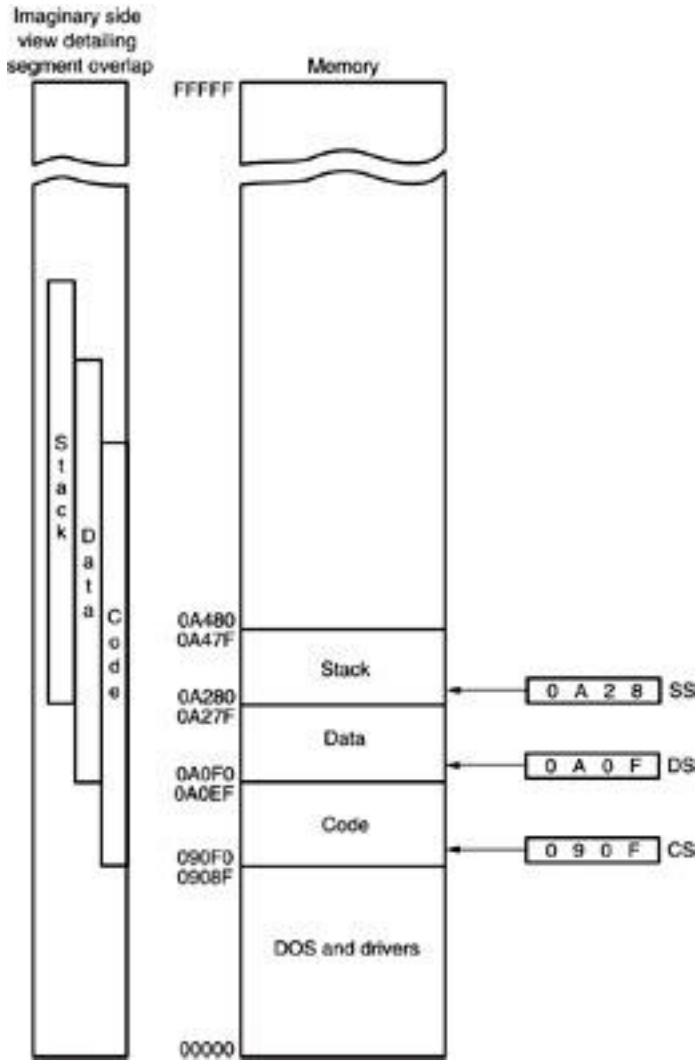
- Another of the default combinations is the **stack**.
  - stack data are referenced through the stack segment at the memory location addressed by either the stack pointer (SP/ESP) or the pointer (BP/EBP)
- Figure 2–4 shows a system that contains four memory segments.
  - a memory segment can touch or overlap if 64K bytes of memory are not required for a segment

Figure 2–4 A memory system showing the placement of four memory segments.



- **think of segments as windows that can be moved over any area of memory to access data or code**
- **a program can have more than four or six segments,**
  - **but only access four or six segments at a time**

Figure 2–5 An application program containing a code, data, and stack segment loaded into a DOS system memory.



- **a program placed in memory by DOS is loaded in the TPA at the first available area of memory above drivers and other TPA programs**
- **area is indicated by a free-pointer maintained by DOS**
- **program loading is handled automatically by the program loader within DOS**

TABLE 2–3 Default 16-bit segment and offset combinations.

Segment	*Offset*	*Special Purpose*
CS	IP	Instruction address
SS	SP or BP	Stack address
DS	BX, DI, SI, an 8- or 16-bit number	Data address
ES	DI for string instructions	String destination address

TABLE 2–4 Default 32-bit segment and offset combinations.

Segment	*Offset*	*Special Purpose*
CS	EIP	Instruction address
SS	ESP or EBP	Stack address
DS	EAX, EBX, ECX, EDX, ESI, EDI, an 8- or 32-bit number	Data address
ES	EDI for string instructions	String destination address
FS	No default	General address
GS	No default	General address

Segment and Offset Addressing Scheme Allows Relocation

- Segment plus offset addressing allows DOS programs to be relocated in memory.
- A **relocatable program** is one that can be placed into any area of memory and executed without change.
- **Relocatable data** are data that can be placed in any area of memory and used without any change to the program.

- Because memory is addressed within a segment by an offset address, the memory segment can be moved to any place in the memory system without changing any of the offset addresses.
- Only the contents of the segment register must be changed to address the program in the new area of memory.
- Windows programs are written assuming that the first 2G of memory are available for code and data.