Chapter 2 The Microprocessor and its Architecture

Assoc. Prof. Dr. Gazi Erkan BOSTANCI

Slides are mainly based on The Intel Microprocessors by Barry B. Brey, 2008

- We are going to look at the microprocessor as aprogrammable device by first looking at its internal programming model and then how its memory space is addressed.
- Real addressing mode will be covered. Protected and flat modes are beyond the scope of our course.

Internal MP architecture

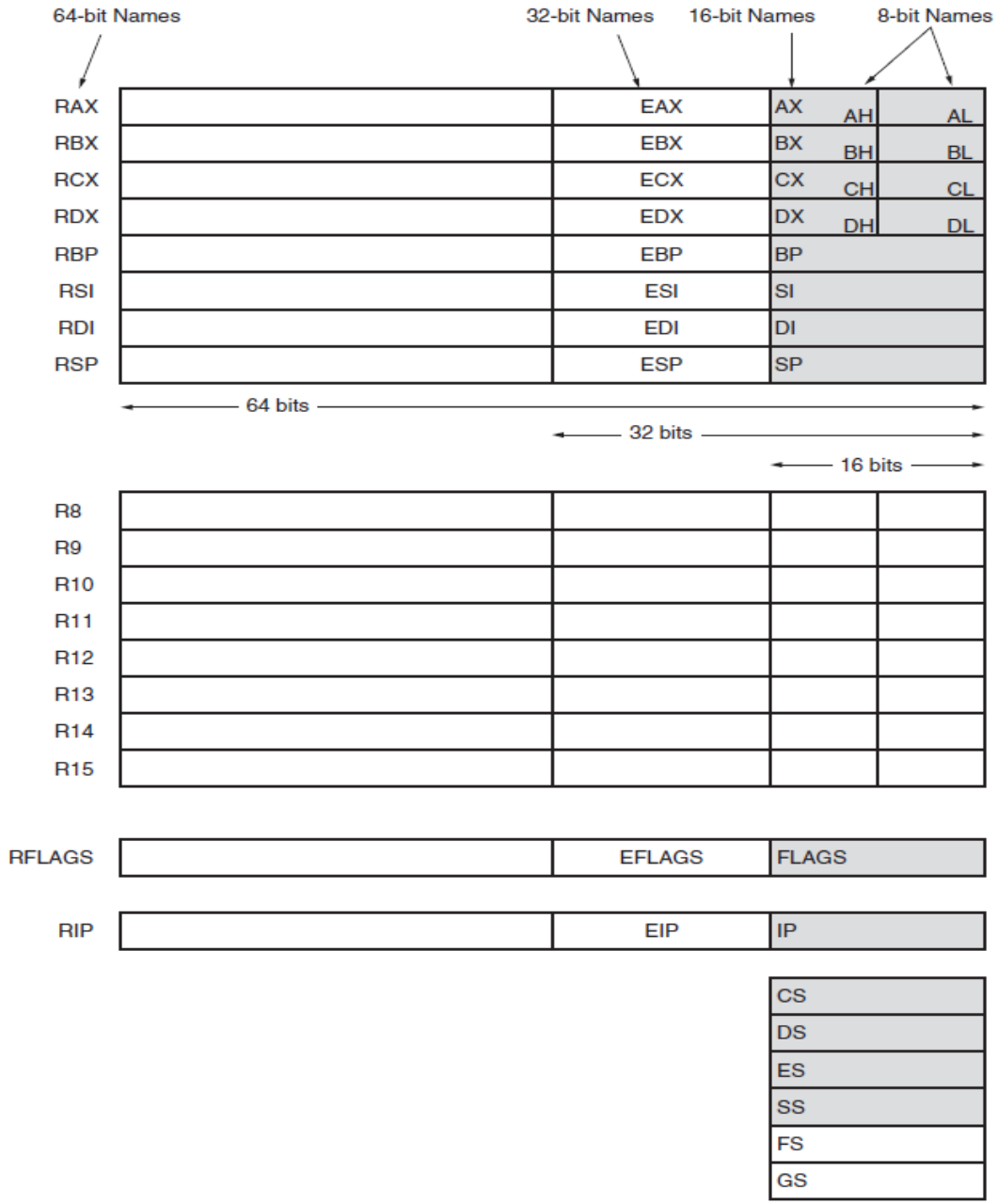
- Before a program is written or any instruction is investigated, the internal configuration of the microprocessor must be known.

❖In multiple core MP, each core contains the same programming model. The only difference is that each core runs a separate task or thread simultaneously.

The Programming Model

- The programming model of 8086 through Core2 is considered to be program visible, because its registers are used during application programming and are specified by the instructions.
- Other registers are program invisible because they cannot be addressed directly with application programming, but may be indirectly used during system programming.

- The programming model contains 8, 16, 32 and 64bit registers.
  - 8 bit registers: AH, AL, BH, BL, CH, CL, DH and DL
  - These registers are referred to when an instruction is formed using these two letter designations.
    - ADD AL, AH; adds the 8 bit contents of AH to AL
  - 16 bit registers: AX, BX, CX, DX, SP, BP, DI, SI, IP, FLAGS, CS, DS, ES, SS, FS, GS
    - The first four registers contain 2 x 8 bit registers.
  - 32 bit registers: EAX, EBX, ECX, EDX, EDI, ESI, ….
  - 64 bit registers are designated as RAX, RBX, …
- Some registers are general or multipurpose registers, while some have special purposes. The multipurpose registers include AX, BX, CX, DX, DI and SI.


64-bit Names
32-bit Names
16-bit Names
8-bit Names
RAX
RBX
RCX
RDX
RBP
RSI
RDI
RSP
EAX
EBX
ECX
EDX
EBP
ESI
EDI
ESP
AX
AH
AL
BX
BH
BL
CX
CH
CL
DX
DH
DL
BP
SI
DI
SP
64 bits
32 bits
16 bits
R8
R9
R10
R11
R12
R13
R14
R15
RFLAGS
EFLAGS
FLAGS
RIP
EIP
IP
CS
DS
ES
SS
FS
GS

- There are also additional 64 bit registers that are called R8 to R15.
- These registers can be addressed as byte, word, double word or quadword; but only the rightmost 8bit as a byte. R8 to R15 have no provision for directly addressing bits 8-15 as a byte.

Register Size	Override	Bits Accessed	Example
8 bits	B	7-0	MOV R9B, R10B
16 bits	W	15-0	MOV R10W, AX
32 bits	D	31-0	MOV R14D, R15D
64 bits	----	63-0	MOV R13, R12

Multipurpose Registers

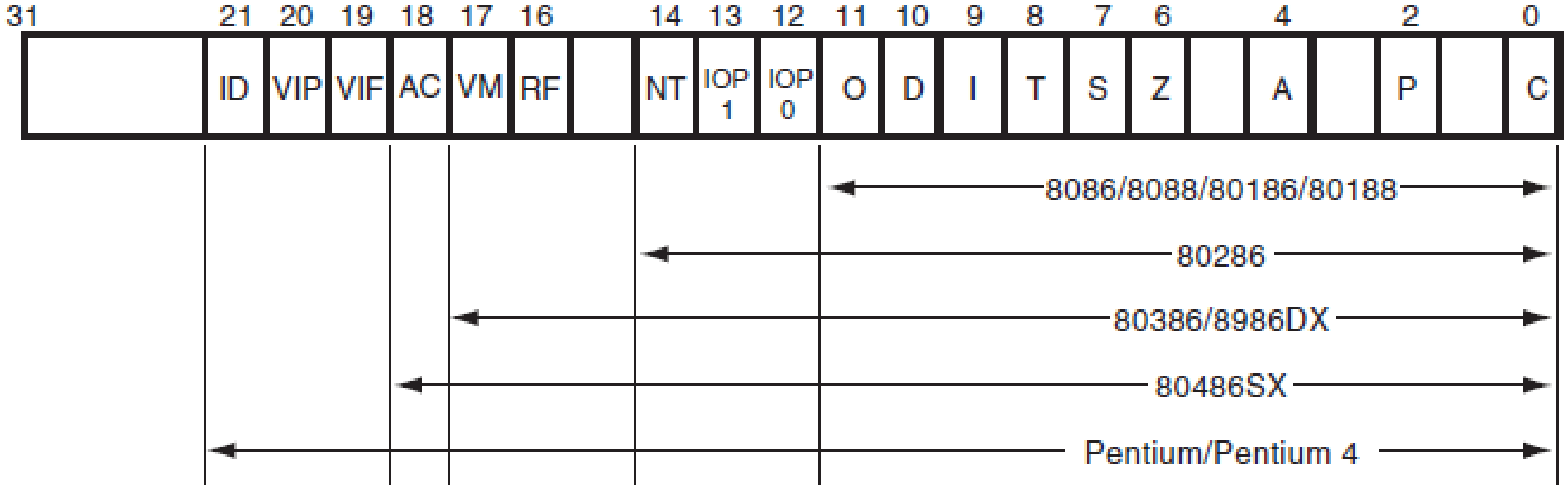
- AX (Accumulator): 64 bits RAX, 32 bits EAX, 16 bits AX, 8 bits AH or AL
  - If an 8-16bit register is addressed, only that portion of the 32 bit is changed without affecting the remaining bits.
  - This register is used for instructions such as multiplication, division and some adjustment instructions.
- BX (Base index): BX is addressable as RBX, EBX, BX, BH and BL.
  - This register holds the offset address of a location in the memory system.
- CX (Count): Addressable as RCX, ECX, CX, CH and CL.
  - It holds the count for various instructions. Instructions that use a count are the repeated string instructions, shift, rotate and loop instructions.
    - Shift and Rotate CL
    - Repeated string CX
    - LOOP /LOOPD CX or ECX
  - In 64 bit mode LOOP uses RCX register for the loop counter.

- DX (Data): RDX, EDX, DX, DH and DL.
  - It holds part of result from a multiplication or part of the dividend before a division. It can also address memory data.
- BP (Base Pointer): RBP, EBP or BP.
  - It points to a memory location for memory data transfers.
- DI (Destination Index): RDI, EDI or DI.
  - It addresses the string destination data for string instructions.
- SI (Source Index): RSI, ESI or SI.
  - It addresses the source string data for string instructions.
- R8-R15: These registers are only found in Pentium 4 and Core2 if 64 bit extensions are enabled.

Special Purpose Registers

- IP (Instruction Pointer): This register addresses the next instruction in a section of memory defined as code segment.
  - The instruction pointer can be modified with a jump or call instruction. (IP, EIP, RIP)
- SP (Stack Pointer): It addresses the stack area of the memory. The stack memory stores data through this pointer. (SP, ESP)

- RFLAGS: These flags indicate the condition of the MP and control its operation.



- The rightmost five flags and the overflow flag are changed by most arithmetic and logic operations, although data transfers do not affect them.
- C (Carry): It holds carry after addition or borrow after subtraction. It also indicates error conditions.
- P (Parity): Parity is logic 0 for odd parity, 1 for even parity. Parity is the count of 1’s in a number. This was mainly used for earlier systems. Today parity checking is performed by data communication equipment rather than the MP.

- A (Auxiliary carry): The auxiliary carry holds the carry (half carry) after addition or the borrow after subtraction between bit positions 3 and 4 of the result.
  - This flag is tested by DAA and DAS instructions to adjust the value of AL after a BCD addition or subtraction.
- Z (Zero): This flag shows that the result of an artihmetic or logic operation is zero.
  - IF Z==1 then the result is zero.
- S (Sign): This flag holds the arithmetic sign of the result after an arithmetic or logic instruction executes.
  - If S==1 then sign bit (leftmost bit) is set and negative otherwise the bit is cleared and positive

- T (Trap): The trap flag enables trapping through an on-chip debugging feature. Visual C++ also use trap feature for debugging.
- I (Interrupt): This flag controls the operation of the INTR (interrupt request) input pin.
  - If I==1, interrupt is enabled.
  - The state of this bit is controlled by STI and CLI instructions (set and clear).
- D (Direction): This flag selects either the increment or decrement mode for the DI and/or SI registers during string instructions.
  - If D==1, the registers are automatically decremented, otherwise automatically incremented.
  - Set and cleared by STD and CLD instructions.

- O (Overflow): Overflows ocur when signed numbers are added or subtracted. An overflow indicates that the result exceeds the capacity of the machine.
  - For instance, if 7FH (+127) is added -using an 8bit addition- to 01H (+1), the result is 80H (-128). This result represents an overflow condition.
  - For unsigned operations, this flag is ignored.
- IOPL (I/O Privilege Level): IOPL is used in proted mode to select the privilege level of I/O devices.
  - If the current privilege level of the task or program is lower or more trusted than the IOPL, I/O executes without hindrance.
  - If the IOPL is lower than then program is suspended with an interrupt.
    - 00 is the highest.
    - 11 is the lowest.

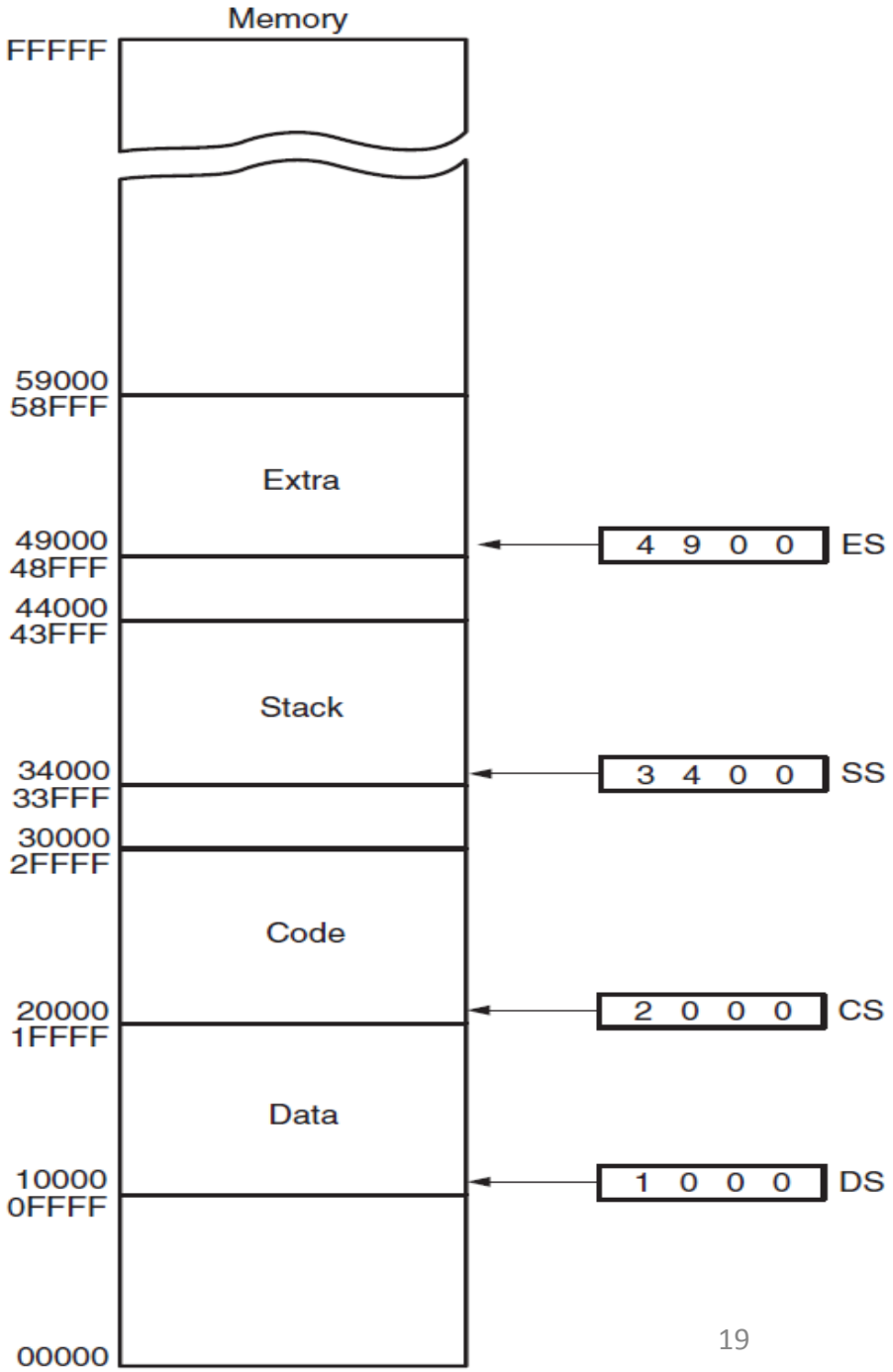
- NT (Nested Task): The nested task flag indicates that the current task is nested within another task in the protected mode.
- RF (Resume): This flag is used with debugging to control the resumptionof execution after the next instruction.
- VM (Virtual Mode): It selects the Virtual Mode which allow multiple DOS memory partitions that are 1Mbyte in length to coexist in the memory system.
  - It allows the system program to execute multiple DOS programs (to simulate DOS in a modern Windows environment.)
- AC (Alignment Checks): This flag activates if a word or doubleword is addressed on a non-word or non-doubleword boundary.

- VIF (Virtual Interrupt): This is a copy of the interrupt flag in Pentium 4.
- VIP (Virtual Interrupt Pending): Available to P4 MPs. This is used in multitasking environments to provide the OS with virtual interrupt flags.
- ID (Identification): This flag indicates P4 microprocessors support the CPUID instruction. This instruction provides the system with version number and manufacturer information.

Segment Registers

- These additional registers generate memory addreses when combined with other registers.
- CS (Code Segment): The code segment is a section of memory that holds the code (programs and procedures) used by the MP.
- DS (Data Segment): The data segment is a section of memory that contains most data used by a program. Data in this segment is addressed by an offset address o the contents of other registers that hold the offset address.
- ES (Extra Segment): Additional data segment used by some of the string instructions to hold destination data.

- SS (Stack Segment): This register defines the stack memory. Stack entry point is determined by the stack segment and stack pointer registers. BP register also addresses data within stack segment.
- FS and GS: Supplemental registers.
  - Windows uses these registers for internal operations, but no details are given.
  - On Linux FS is used for exception handling chain, GS is free.



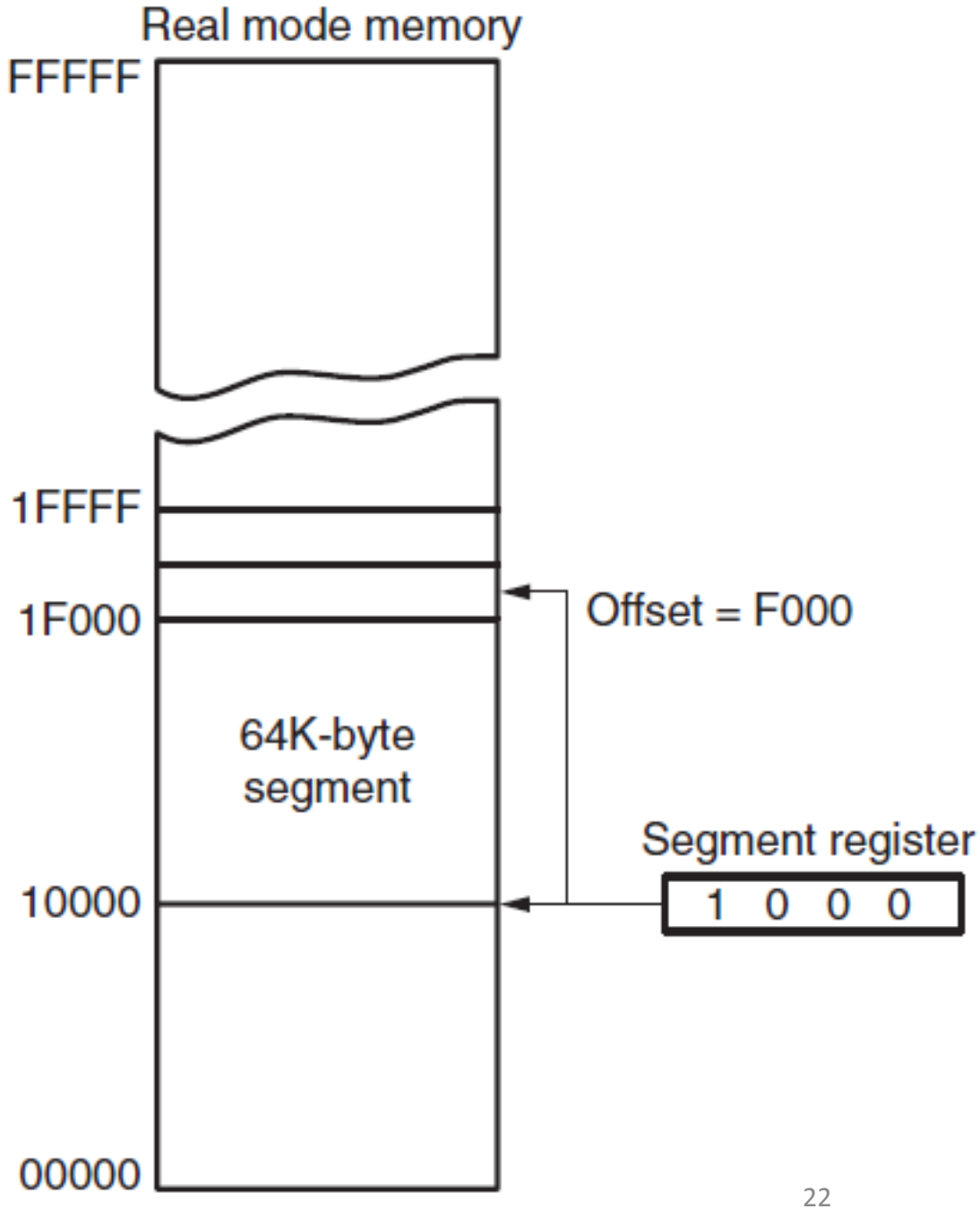
Real Mode Memory Addressing

- Real mode operation allows the MP to address only the first 1Mbyte of memory space.
- Windows does not use real mode.

Segments and Offsets

- A combination of segment address and an offset address accesses a memory location in the real mode.
- The segment address, located within one of the segment registers, defines the beginning address of any 64Kbyte memory segment.
- The offset address selects any location within the 64Kbyte memory segment.

- The memory segment starts with 10000H and ends at 1FFFFH (64Kbytes).
- Note that the segment register contains 1000H, but it addresses a starting segment at location 10000H.
- In the real mode, each segment register is internally appended with 0 on its rightmost end. The MP must generate a 20bit address to access a location within the first 1Mbyte memory.
- Because the length of real mode segment is 64K, once the beginning address is known, the ending address is found by adding FFFFH (= 65535).



Segment Register	Starting Address	Ending Address
2000H	20000H	2FFFFH
2001H	20010H	3000FH
2100H	21000H	30FFFH

- The offset address is always added to the starting address of the segment to locate the data. The segment and offset address is sometimes written as 1000:2000 for a segment address of 1000H with offset of 2000H.

Default Segment and Offset Registers

- The MP has a set of rules that apply to segments whenever memory is addressed.
- For instance, the code segment register is always used with the instruction pointer.
  - CS:IP
  - The code segment defines the start of the code segment and instruction pointer locates the next instruction in this segment.
  - If CS=1400H and IP=1200H, the address of the next instruction is 14000H+1200H = 15200H
- Another default combination is the stack. (SS:SP). BP can also be used here.
  - For SS=2000H and BP=3000H then 23000H is the address to access in the stack segment.

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

- The segment and offset addressing may look complicated. However, it offers an advantage to the system.
- This complicated scheme of segment plus offset addressing allows DOS programs to be relocated in the memory system. It also allows programs written to function in the real mode to operate in a protected mode system.
  - 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. The segment and offset addressing scheme allows both programs and data to be relocated without changing a thing in a program or data.
- This is ideal for use in a general-purpose computer system in which not all machines contain the same memory areas. The personal computer memory structure is different from machine to machine, requiring relocatable software and data.
- This is also important for implementing swapping operation or paging for virtual memory.