Lecture 2

The Microprocessor and its Architecture

Contents

- Internal architecture of the Microprocessor:
  - The programmer's model, i.e. The registers model
  - The processor model (organization)
- Real mode memory addressing using memory segmentation
- Protected mode memory addressing using memory segmentation
- Memory paging mechanism

Objectives for this Chapter

- Describe the function and purpose of program-visible registers
- Describe the Flags register and the purpose of each flag bit
- Describe how memory is accessed using segmentation in both the real mode and the protected mode
- Describe the program-invisible registers
- Describe the structures and operation of the memory paging mechanism
- Describe the processor model

TABLE 1–6 The Intel family of microprocessor bus and memory sizes.

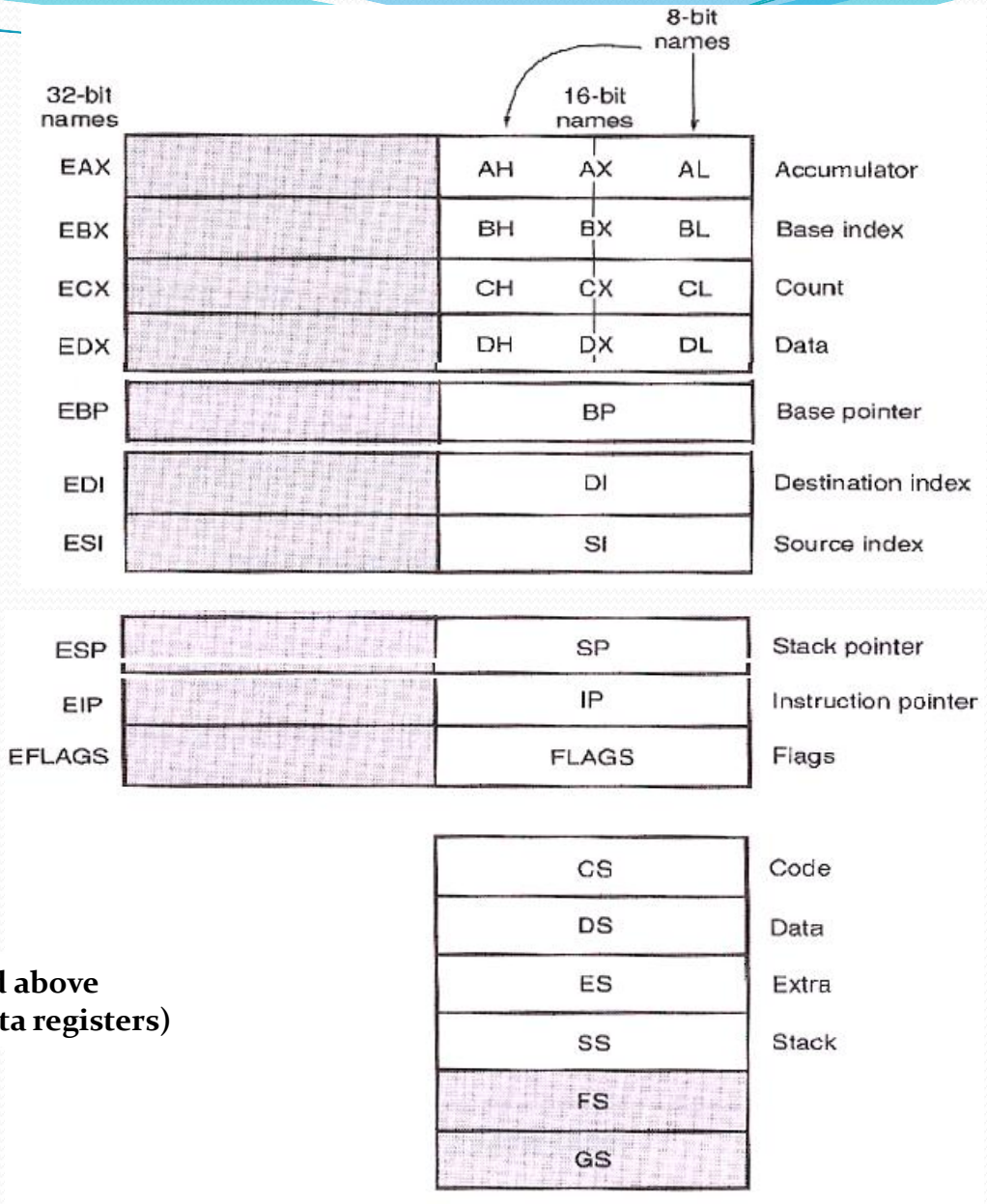
Microprocessor	*Data Bus Width*	*Address Bus Width*	*Memory Size*
8086	16	20	1M
8088	8	20	1M
80186	16	20	1M
80188	8	20	1M
80286	16	24	16M
80386SX	16	24	16M
80386DX	32	32	4G
80386EX	16	26	64M
80486	32	32	4G
Pentium	64	32	4G
Pentium Pro–Pentium 4	64	32	4G
Pentium Pro–Pentium 4 (if extended addressing is enabled)	64	36	64G

Programming Model

General Purpose Registers

Special Purpose Registers

Segment Registers



General-Purpose Registers

- The top portion of the programming model contains the general purpose registers: EAX, EBX, ECX, EDX, EBP, ESI, and EDI.
- Although general in nature, each have a special purpose and name:
- Can carry both Data & Address offsets
- EAX – Accumulator

  Used also as AX (16 bit), AH (8 bit), and AL (8 bit)
- EBX – Base Index often used to hold offset address of a memory location (BX, BH, and BL)

General-Purpose Registers (continued)

- ECX – count, for shifts, rotates, and loops (CX, CH, and CL)
- EDX – data, used with multiply and divide (DX, DH, and DL)
- EBP – base pointer used to address stack data (BP)
- ESI – source index (SI) for memory locations, e.g. with string instructions
- EDI – destination index (DI) for memory locations string operations

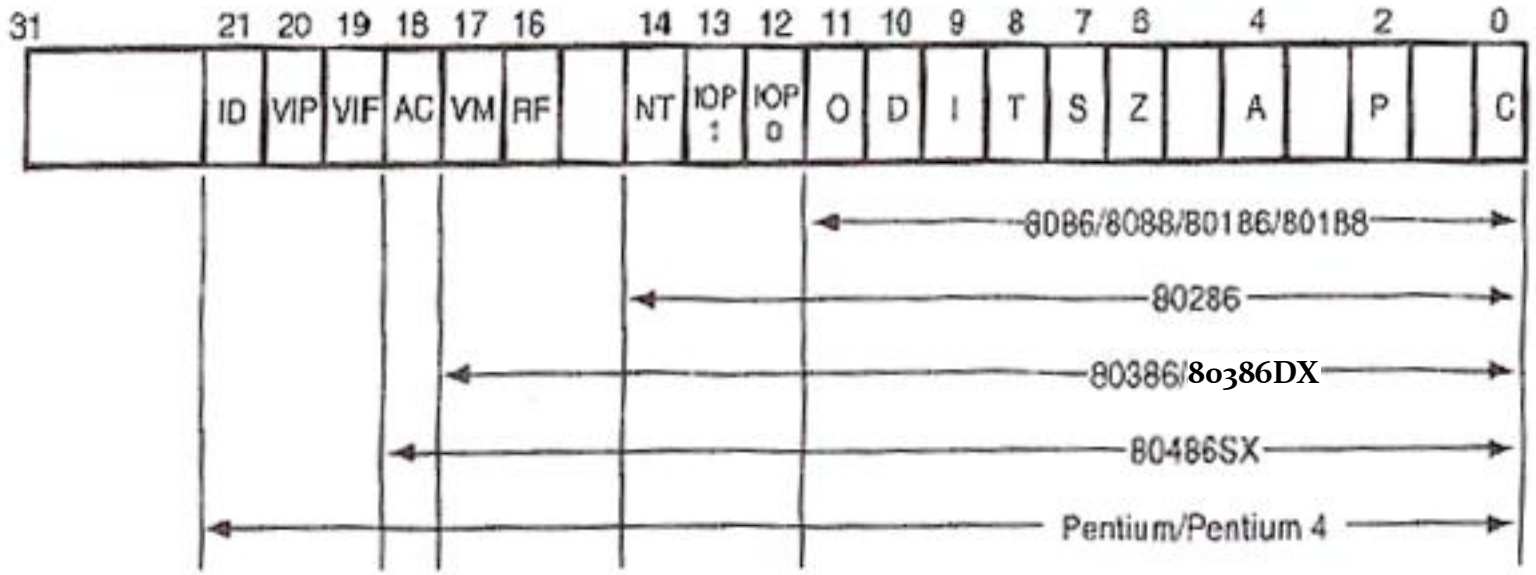
Special-Purpose Registers

- ESP, EIP, and EFLAGS

  Each has a specific task

  - ESP – Stack pointer: addresses the stack segment used with functions (procedures) (SP)
  - EIP – Instruction Pointer: addresses the next instruction in a program in the code segment (IP)
  - EFLAGS – indicates latest conditions of the microprocessor (FLAGS)

EFLAGS


31
21
20
19
18
17
16
14
13
12
11
10
9
8
7
6
4
2
0
ID
VIP
VIF
AC
VM
RF
NT
IOP 1
IOP 0
O
D
I
T
S
Z
A
P
C
8086/8088/80186/80188
80286
80386/80386DX
80486SX
Pentium/Pentium 4

The Flags

Basic Flag Bits (8086 etc.)

- C – Carry/borrow of result also show error conditions
- P – the parity flag odd or even parity (little used today)
- A – auxiliary flag used with BCD arithmetic, e.g. using DAA and DAS (decimal adjust after add/sub)
- Z – zero
- S – sign (-ve (S=1)or +ve (S=0))
- O – Overflow
- D – direction - Determines auto incr/dec direction for string instructions (STD, CLD)
- I – interrupt- Enables (using STI) or disables (using CLI) the processing of hardware interrupts arriving at the INTR input pin
- T – Trap- (turns trapping on/off for program debugging)

Newer Flag Bits

- IOPL – 2-bit I/O privilege level in protected mode
- NT – nested task
- RF – resume flag (used with debugging)
- VM – virtual mode: multiple DOS programs each with a 1 MB memory partition
- AC – alignment check: detects addressing on wrong boundary for words/double words
- VIF – virtual interrupt flag
- VIP – virtual interrupt pending
- ID = CPUID instruction available

  The instruction gives info on CPU version and manufacturer

Segment Registers

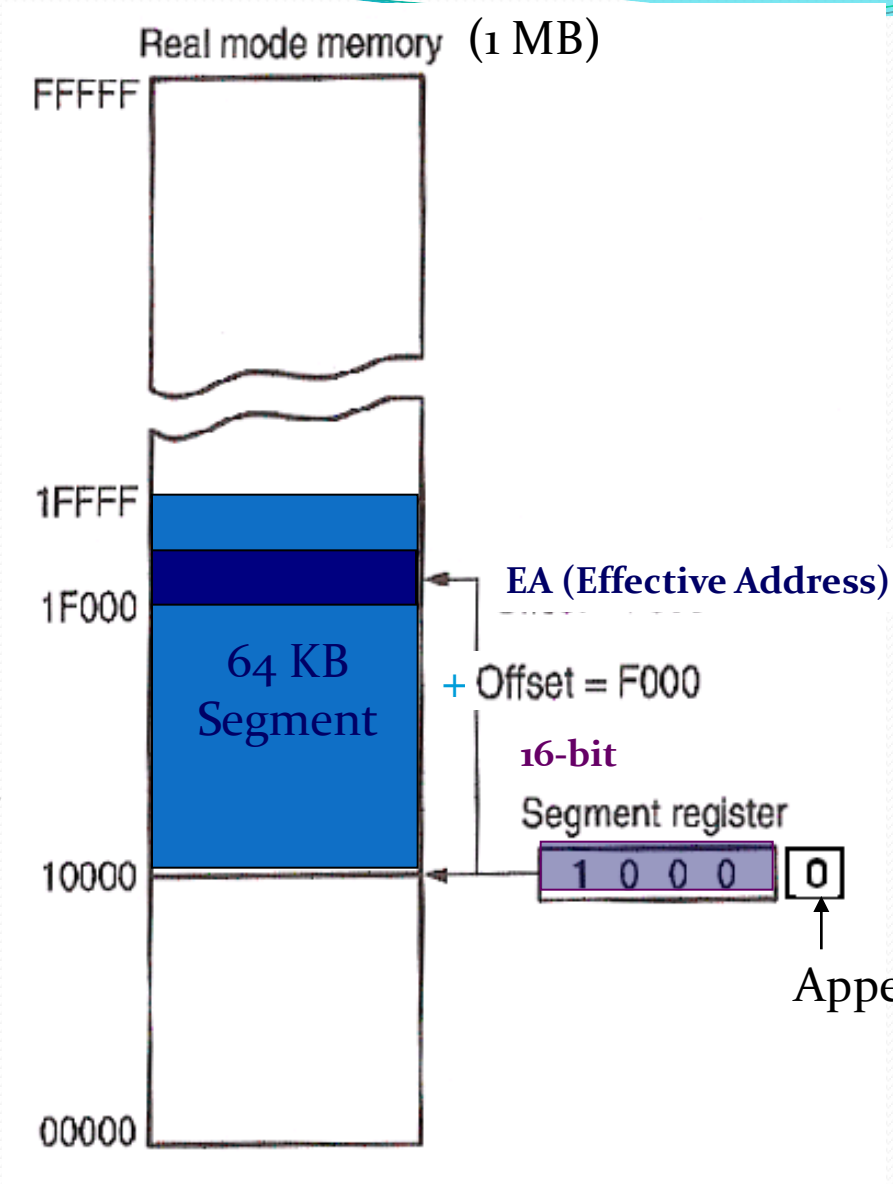
- The segment registers are:
  - CS (code),
  - DS (data),
  - ES (extra data. used with string instructions),
  - SS (stack),
  - FS, and GS.
- Segment registers define a section of memory (segment) for a program.
- A segment is either 64K (2^{16}) bytes of fixed length (in the real mode) or up to 4G (2^{32}) bytes of variable length (in the protected mode).
- All code (programs) reside in the code segment.

Real Mode Memory Addressing

- The only mode available on for 8088-8086
- Real mode memory is the first 1M (2^{20}) bytes of the memory system (real, conventional, DOS memory)
- All real mode 20-bit addresses are a combination of a segment address (in a segment register) plus an offset address (in another register)
- The segment register address (16-bits) is appended with a 0H or 0000_2 (or multiplied by 10H) to form a 20-bit start of segment address
- The effective memory address =

  this 20-bit segment address + a 16-bit offset address in a register
- Segment length is fixed = 2^{16} = 64K bytes

20-bit (5-byte)
Physical
Memory address



Effective Address Calculations

- EA = segment register (SR) x 10H plus offset

  (a) SR: 1000H

  10000 + 0023 = 10023

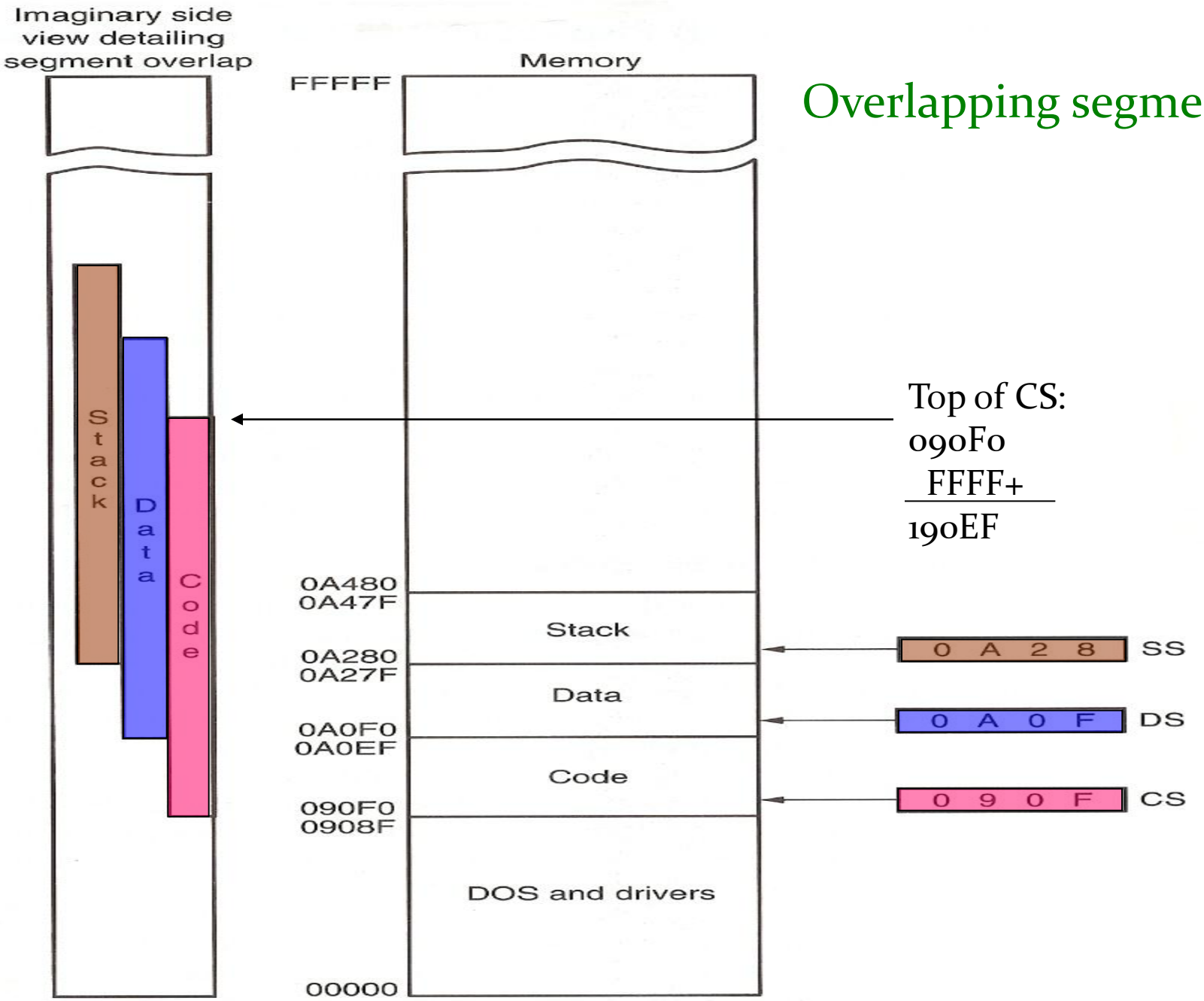
  (b) SR: AAF0H

  AAF00 + 0134 = AB034

  (c) SR: 1200H

  12000 + FFF0 = 21FF0

Overlapping segments



Defaults

- CS for program (code)
- SS for stack
- DS for data
- ES for string destination

- Default offset addresses that go with them:

Segment	Offset (16-bit) 8080, 8086, 80286	Offset (32-bit) 80386 and above	Purpose
CS	IP	EIP	Program
SS	SP, BP	ESP, EBP	Stack
DS	BX, DI, SI, 8-bit or 16-bit #	EAX, EBX, ECX, EDX, ESI, EDI, 8-bit or 32-bit #	Data
ES	DI	EDI	String destination

Segmentation: Pros and Cons

Advantages:

- Allows easy and efficient relocation of code and data
- A program can be located anywhere in memory without any change to the program
- Program writing needs not worry about actual memory structure of the computer used to execute it
- To relocate code or data only the segment number needs to be changed

Disadvantages:

- Complex hardware and for address generation
- Software: Programs limited by segment size (only 64KB with the 8086)

Limitations of the above real mode segmentation scheme

- Segment size is fixed at 64 KB
- Segment can not begin at an arbitrary memory address…

  With 20-bit memory addressing, can only begin at addresses starting with 0H, i.e. at 16 byte intervals
- Difficult to use with 24 or 32-bit memory addressing with segment registers remaining at 16-bits

80286 and above use 24, 32, 36 bit addresses but still 16-bit segment registers

→ Use memory segmentation in the protected mode