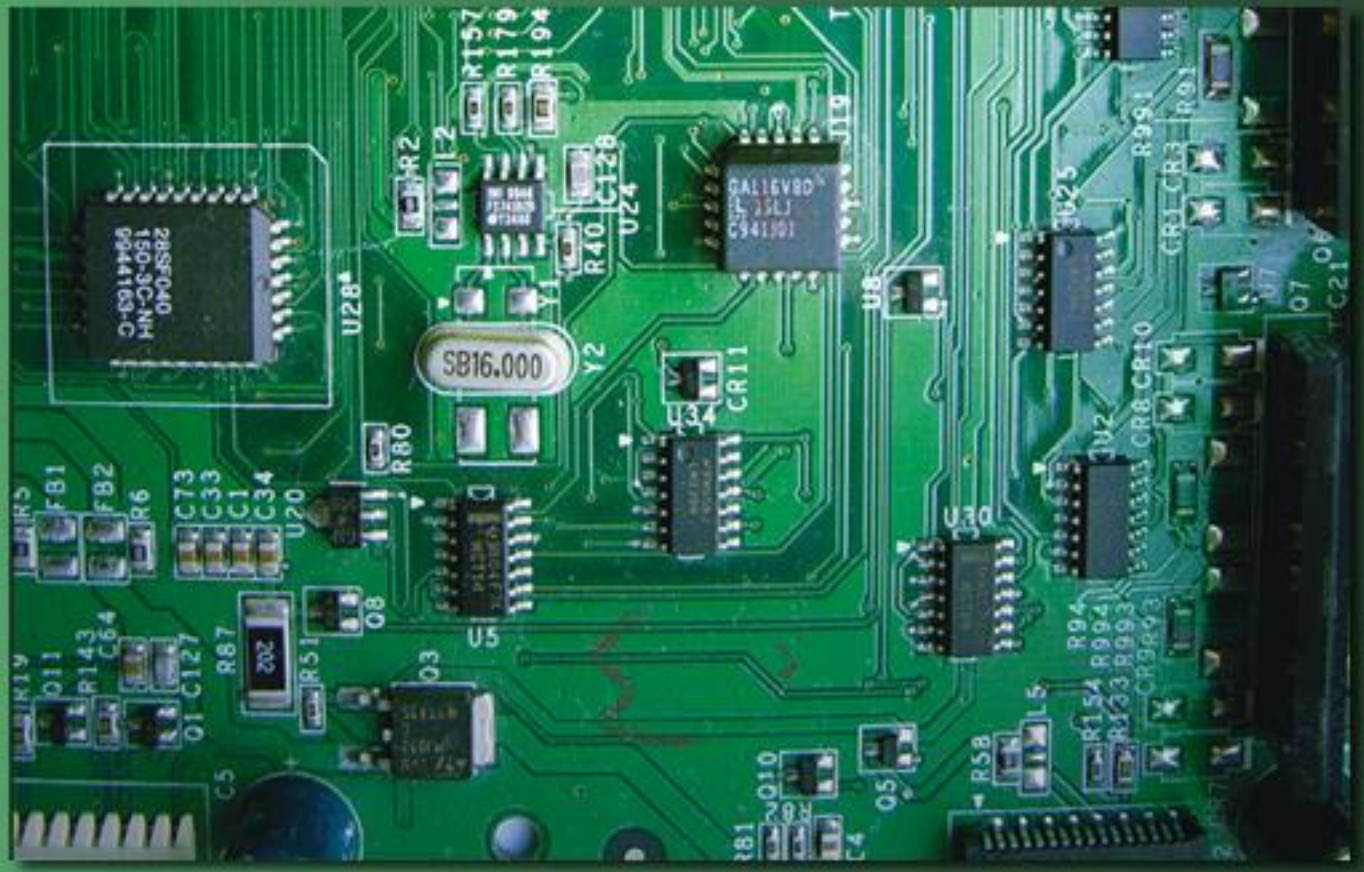
The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions

Architecture, Programming, and Interfacing



EIGHTH EDITION

Barry B. Brey

PEARSON

Introduction

- 8086 Microprocessor is an enhanced version of 8085 Microprocessor
- It is a 16-bit Microprocessor having 20 address lines and 16 data lines
- It provides up to 1MB storage.
- It consists of powerful instruction set, which provides operations like multiplication and division easily.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Features of 8086

- It has an instruction queue, which can store six instruction bytes from the memory resulting in faster processing.
- It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.

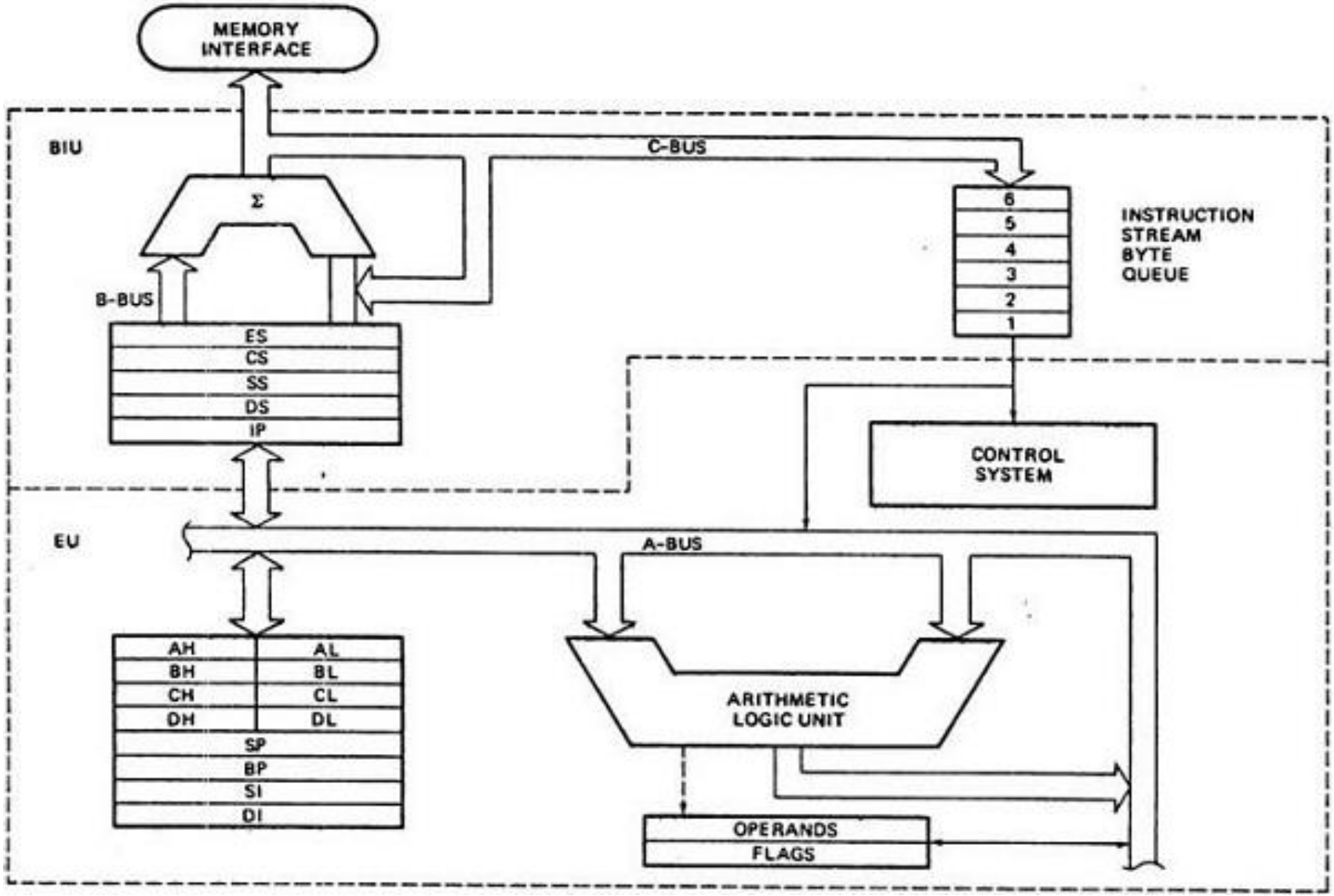
Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Features of 8086 (cont.)

- Fetch stage can prefetch up to 6 bytes of instructions and stores them in the queue.
- Execute stage executes these instructions.
- It has 256 vectored interrupts.
- It consists of 29,000 transistors.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Architecture of 8086



Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

EU (Execution Unit)

- Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions.
- Its function is to control operations on data using the instruction decoder & ALU.
- EU has no direct connection with system buses as shown in the above figure, it performs operations over data through BIU.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

EU (Execution Unit) (cont.)

- ALU: It handles all arithmetic and logical operations, like +, −, ×, /, OR, AND, NOT operations.
- Flag Register: is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups − Conditional Flags and Control Flags.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Conditional Flags

- **Carry flag** − This flag indicates an overflow condition for arithmetic operations.
- **Auxiliary flag** − When an operation is performed at ALU, it results in a carry/barrow from lower half (D0 – D3) to upper half(D4 – D7), then this flag is set, i.e. carry given by D3 bit to D4 is AF flag. The processor uses this flag to perform binary to BCD conversion.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Conditional Flags (cont.)

- **Parity flag** − This flag is used to indicate the parity of the result, i.e. when the lower order 8-bits of the result contains even number of 1's, then the Parity Flag is set. For odd number of 1's, the Parity Flag is reset.
- **Zero flag** − This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Conditional Flags (cont.)

- **Sign flag** − This flag holds the sign of the result, i.e. when the result of the operation is negative, then the sign flag is set to 1 else set to 0.
- **Overflow flag** − This flag represents the result when the system capacity is exceeded.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Control Flags

- **Trap flag**: It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- **Interrupt flag**: It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

Control Flags(cont.)

- **Direction flag**: It is used in string operation. When it is set to 1 then SI, DI will be decremented by 1, if it is 0, then SI, DI will be incremented by 1.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

BIU (Bus Interface Unit)

- Responsible for all data and addresses transfers on the buses for the EU like **sending addresses**, **fetching instructions** from the memory, **reading data** from the ports and the memory as well as **writing data** to the ports and the memory.
- EU has no direct connection with System Buses so this is possible with the BIU.
- EU and BIU are connected with the Internal Bus.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

BIU Functional Parts

- **Instruction queue**: BIU gets up to 6 bytes of next instructions and stores them in the **instruction queue**.
- When **EU** executes instructions and is ready for its next instruction, then it simply reads the instruction from this **instruction queue** resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called **pipelining**.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

BIU Functional Parts(cont.)

- **Segment register**: BIU has 4 segment buses, i.e. CS, DS, SS& ES.
- It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations.
- It also contains 1 pointer register IP, which holds the address of the **next instruction** to be executed by the EU.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

BIU Functional Parts(cont.)

- **CS**: It stands for Code Segment. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.
- **DS**: It stands for Data Segment. It consists of data used by the program and is accessed in the data segment by an offset address or the content of other register that holds the offset address.
- **SS**: It stands for Stack Segment. It handles memory to store data and addresses during execution.
- **ES**: It stands for Extra Segment. ES is additional data segment, which is used by the string to hold the extra destination data.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.