Microprocessor Systems and Interfacing

EEE 342

Nesruminallah

nesruminallah@cuilahore.edu.pk

Introduction to 8086 & Programming Model

CLO	Bloom Taxonomy	Specific Outcome
CLO1	C2	Comprehend the theoretical knowledge of microprocessor and microcontroller using hardware architecture

- Outline
  - Introduction to MPU System
  - History of Microprocessors
  - Basic Concepts of Microprocessor Architecture
  - Registers and Flags

Introduction

- MICROCOMPUTERS AND MICROPROCESSORS
- There are three major parts of a Computer System.
  - ***Central Processing Unit (CPU):*** Also simply called as the microprocessor acts as the brain coordinating all activities within a computer.
  - ***The Memory:*** The program instructions and data are primarily stored.
  - ***The Input/output (I/O) Devices:*** Allow the computer to input information for processing and then output the results. I/O Devices are also known as computer peripherals.
- The integrated Circuit (IC) chip containing the CPU is called the ***microprocessor*** and the entire computer including the microprocessor, memory and I/O is called a ***microcomputer***.

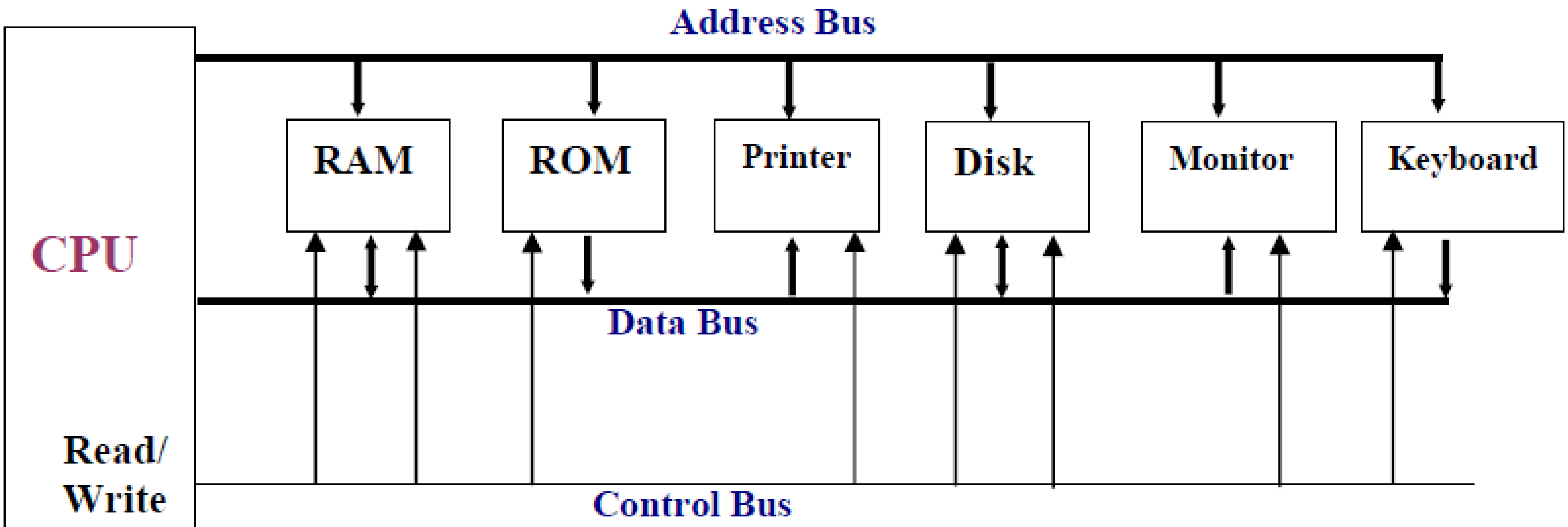
Introduction

- The CPU is connected to memory and I/O devices through a strip of wires called a ***bus***. The bus inside a computer carries information from place to place. In every computer there are three types of busses:
  - ***Address Bus:*** The address bus is used to identify the memory location or I/O device the processor intends to communicate with
  - ***Data Bus:*** Data bus is used by the CPU to get data from / to send data to the memory or the I/O devices.
  - ***Control Bus:*** Each time the processor outputs an address it also activates one of the four control bus signals: Memory Read, Memory Write, I/O Read and I/O Write.

Introduction

- The address and control bus contains output lines only, therefore it is unidirectional, but the data bus is ***bidirectional.***
- There two types of memory used in microcomputers:
  - ***RAM*** (**Random Access Memory/ Read-Write memory)** is used by the computer for the temporary storage of the programs that is running. Data is lost when the computer is turned off. So known as ***volatile*** memory.
  - ***ROM*** (**Read Only Memory**) the information in ROM is permanent and not lost when the power is turned off. Therefore, it is called ***nonvolatile*** memory.
- Note that RAM is sometimes referred as ***primary storage***, where magnetic /optical disks are called ***secondary storage.***

Introduction



Internal organisation of a microcomputer

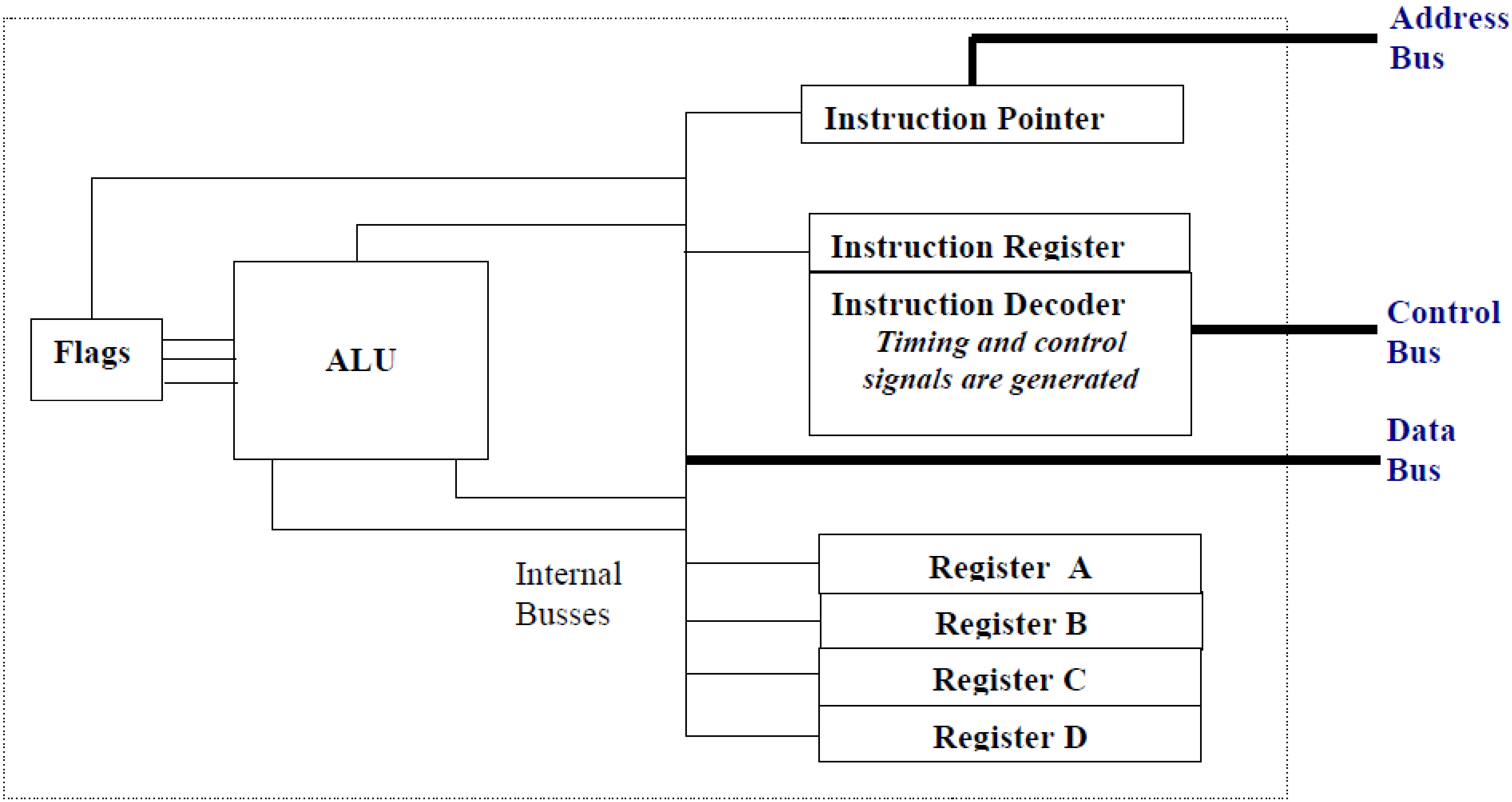
Introduction

- **Inside the CPU:**
- A program stored in the memory provides instructions to the CPU to perform a specific action. This action can be a simple addition. It is function of the CPU to ***fetch*** the program instructions from the memory and ***execute*** them.
- 1. The CPU contains a number of ***registers*** to store information inside the CPU temporarily.
  - Registers inside the CPU can be 8-bit, 16-bit, 32-bit or even 64-bit depending on the CPU.
- 2. The CPU also contains ***Arithmetic and Logic Unit (ALU).*** The ALU performs arithmetic (add, subtract, multiply, divide) and logic (AND, OR, NOT) functions.

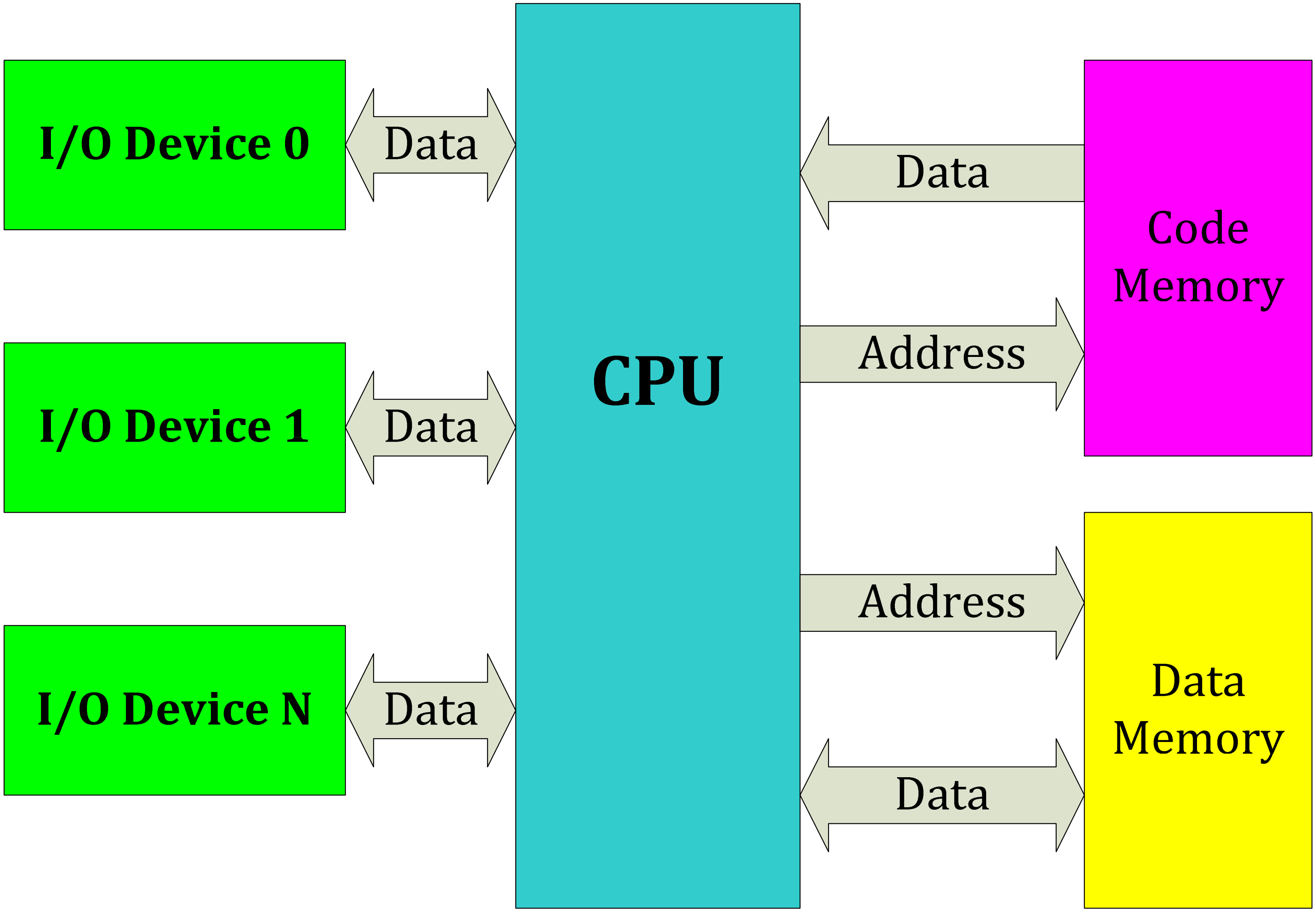
Introduction

- 3. The CPU contains a program counter also known as the ***Instruction Pointer*** to point the address of the next instruction to be executed.
- 4. ***Instruction Decoder*** is a kind of dictionary which is used to interpret the meaning of the instruction fetched into the CPU. Appropriate control signals are generated according to the meaning of the instruction.

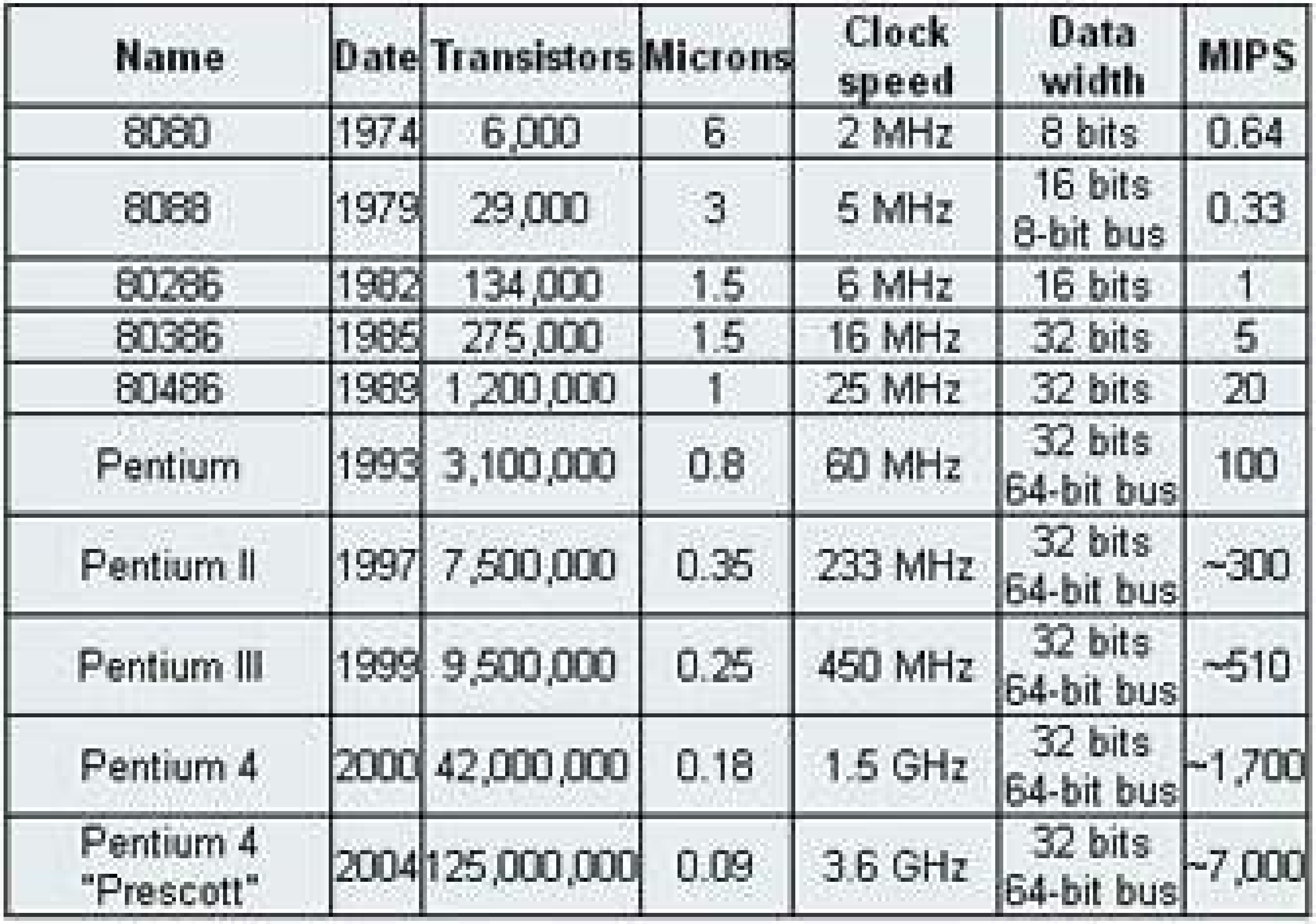
Introduction



Introduction

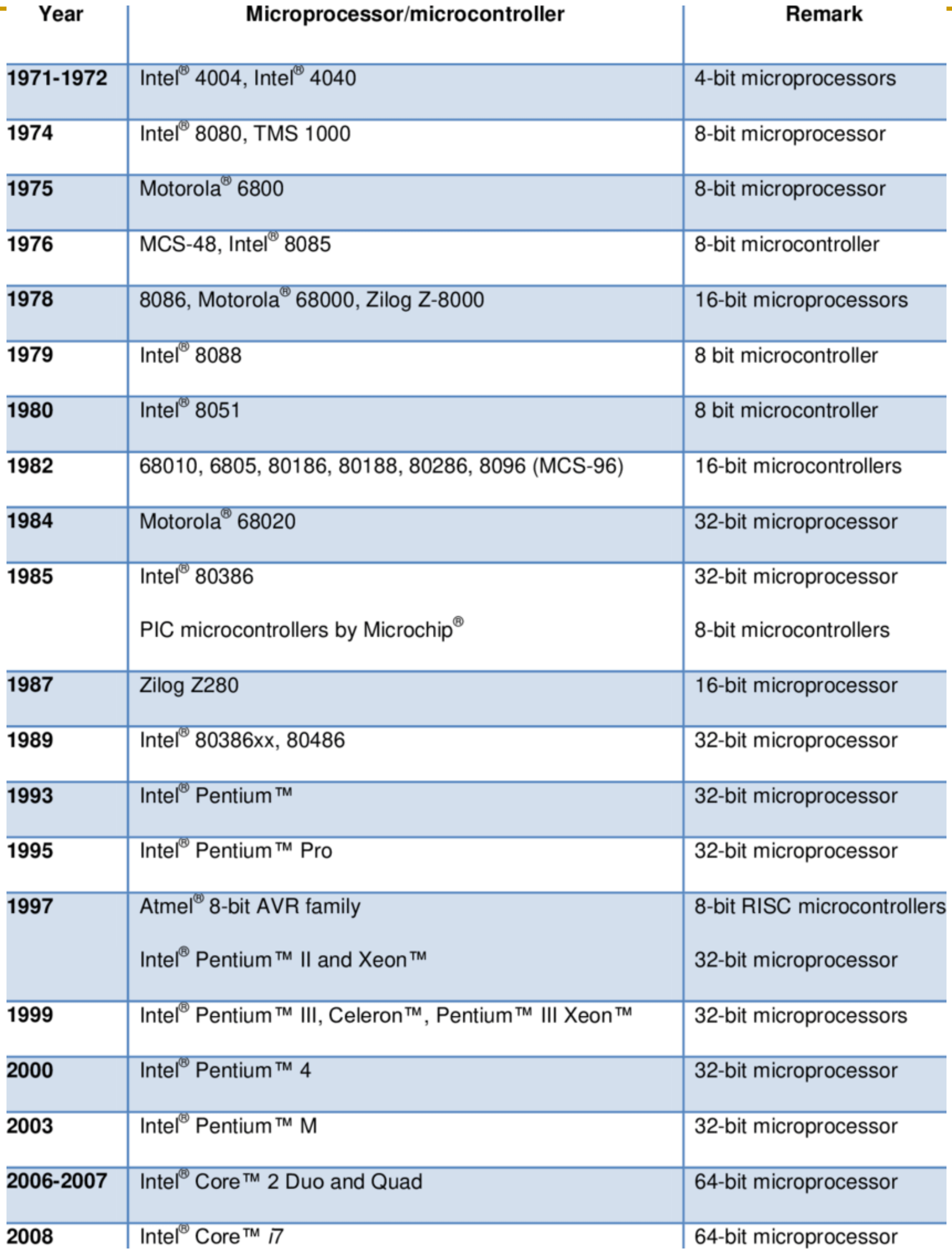


History



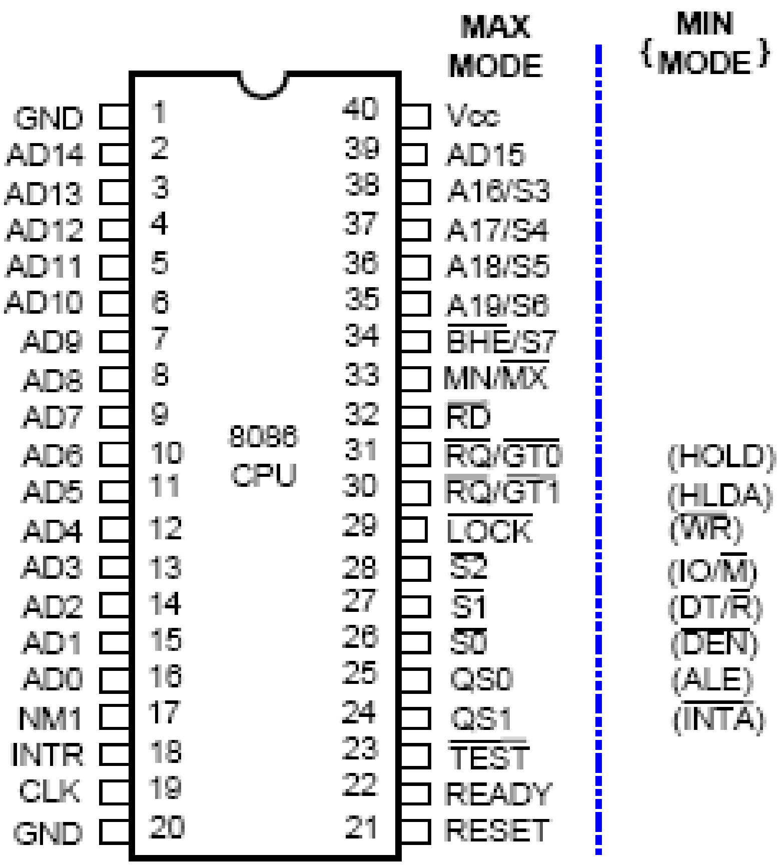
Name	Date	Transistors	Microns	Clock speed	Data width	MIPS
8080	1974	6,000	6	2 MHz	8 bits	0.64
8088	1979	29,000	3	5 MHz	16 bits 8-bit bus	0.33
80286	1982	134,000	1.5	6 MHz	16 bits	1
80386	1985	275,000	1.5	16 MHz	32 bits	5
80486	1989	1,200,000	1	25 MHz	32 bits	20
Pentium	1993	3,100,000	0.8	60 MHz	32 bits 64-bit bus	100
Pentium II	1997	7,500,000	0.35	233 MHz	32 bits 64-bit bus	~300
Pentium III	1999	9,500,000	0.25	450 MHz	32 bits 64-bit bus	~510
Pentium 4	2000	42,000,000	0.18	1.5 GHz	32 bits 64-bit bus	~1,700
Pentium 4 "Prescott"	2004	125,000,000	0.09	3.6 GHz	32 bits 64-bit bus	~7,000

History

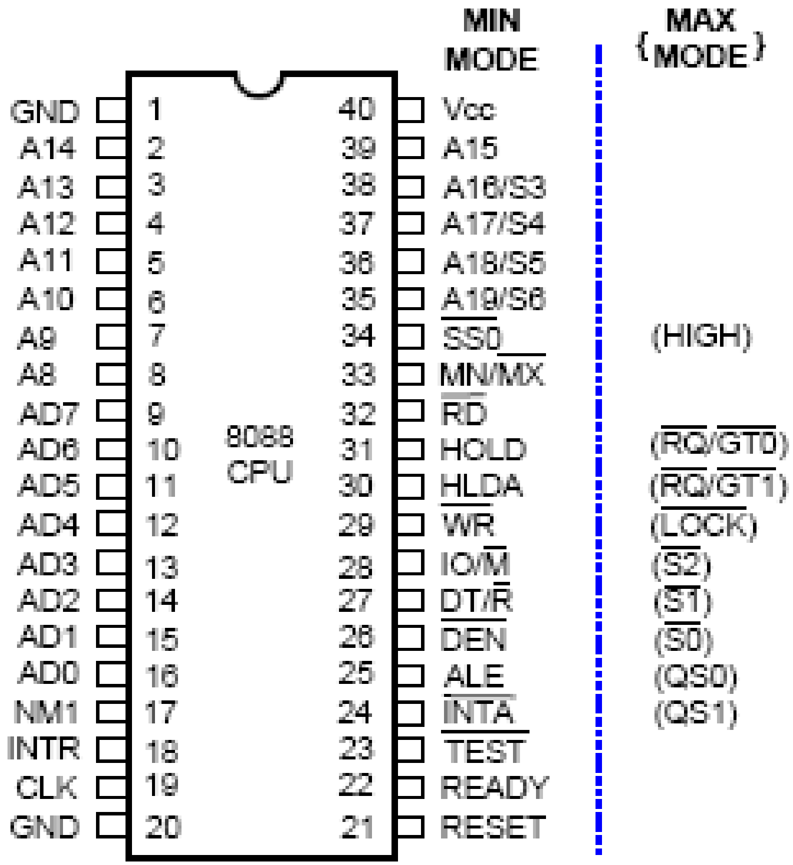


Year	Microprocessor/microcontroller	Remark
1971-1972	Intel® 4004, Intel® 4040	4-bit microprocessors
1974	Intel® 8080, TMS 1000	8-bit microprocessor
1975	Motorola® 6800	8-bit microprocessor
1976	MCS-48, Intel® 8085	8-bit microcontroller
1978	8086, Motorola® 68000, Zilog Z-8000	16-bit microprocessors
1979	Intel® 8088	8 bit microcontroller
1980	Intel® 8051	8 bit microcontroller
1982	68010, 6805, 80186, 80188, 80286, 8096 (MCS-96)	16-bit microcontrollers
1984	Motorola® 68020	32-bit microprocessor
1985	Intel® 80386 PIC microcontrollers by Microchip®	32-bit microprocessor 8-bit microcontrollers
1987	Zilog Z280	16-bit microprocessor
1989	Intel® 80386xx, 80486	32-bit microprocessor
1993	Intel® Pentium™	32-bit microprocessor
1995	Intel® Pentium™ Pro	32-bit microprocessor
1997	Atmel® 8-bit AVR family Intel® Pentium™ II and Xeon™	8-bit RISC microcontrollers 32-bit microprocessor
1999	Intel® Pentium™ III, Celeron™, Pentium™ III Xeon™	32-bit microprocessors
2000	Intel® Pentium™ 4	32-bit microprocessor
2003	Intel® Pentium™ M	32-bit microprocessor
2006-2007	Intel® Core™ 2 Duo and Quad	64-bit microprocessor
2008	Intel® Core™ i7	64-bit microprocessor

Pinout of 8086 & 8088

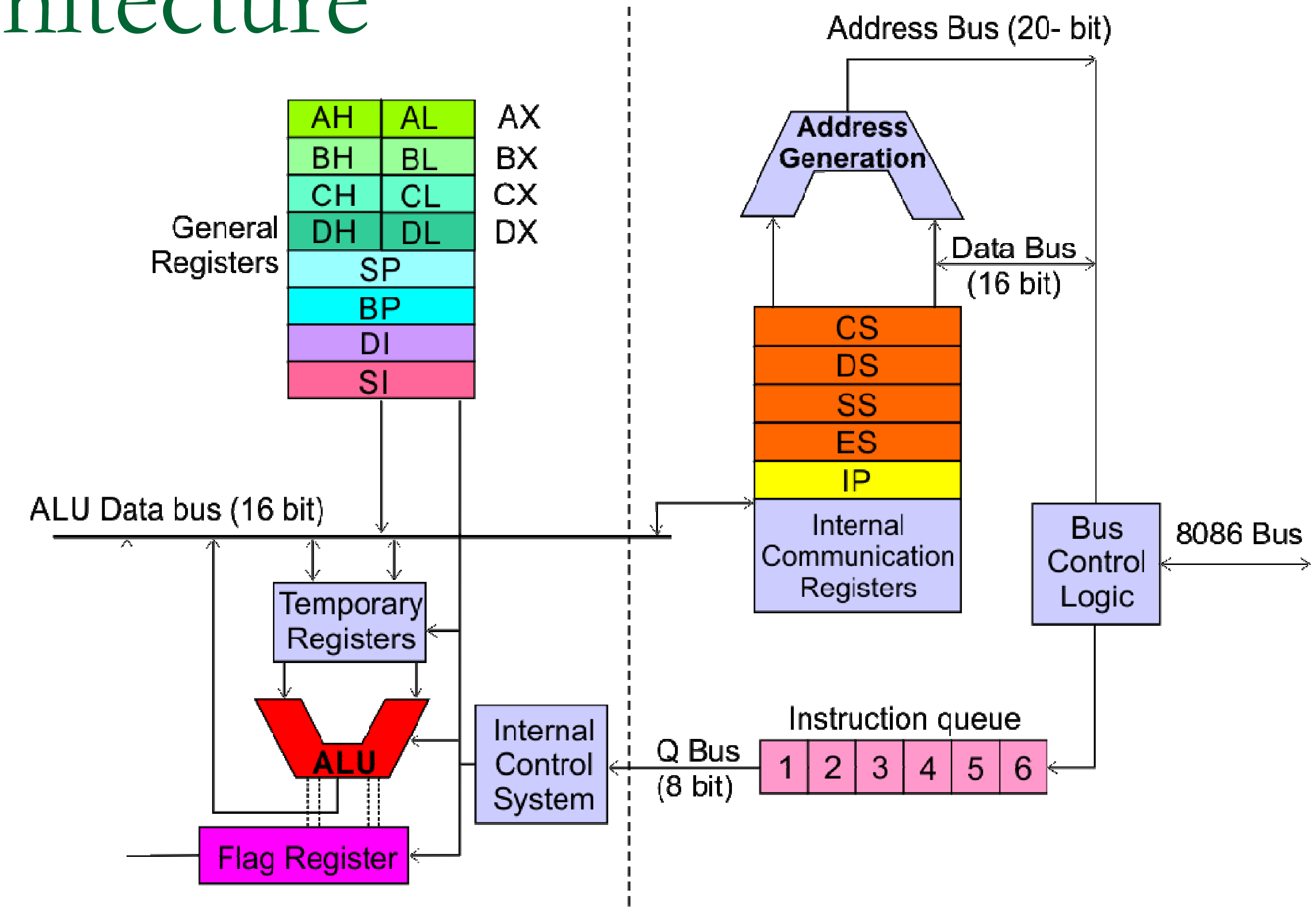


8086 pin diagram



8088 pin diagram

Architecture



Execution Unit (EU)

EU executes instructions that have already been fetched by the BIU.

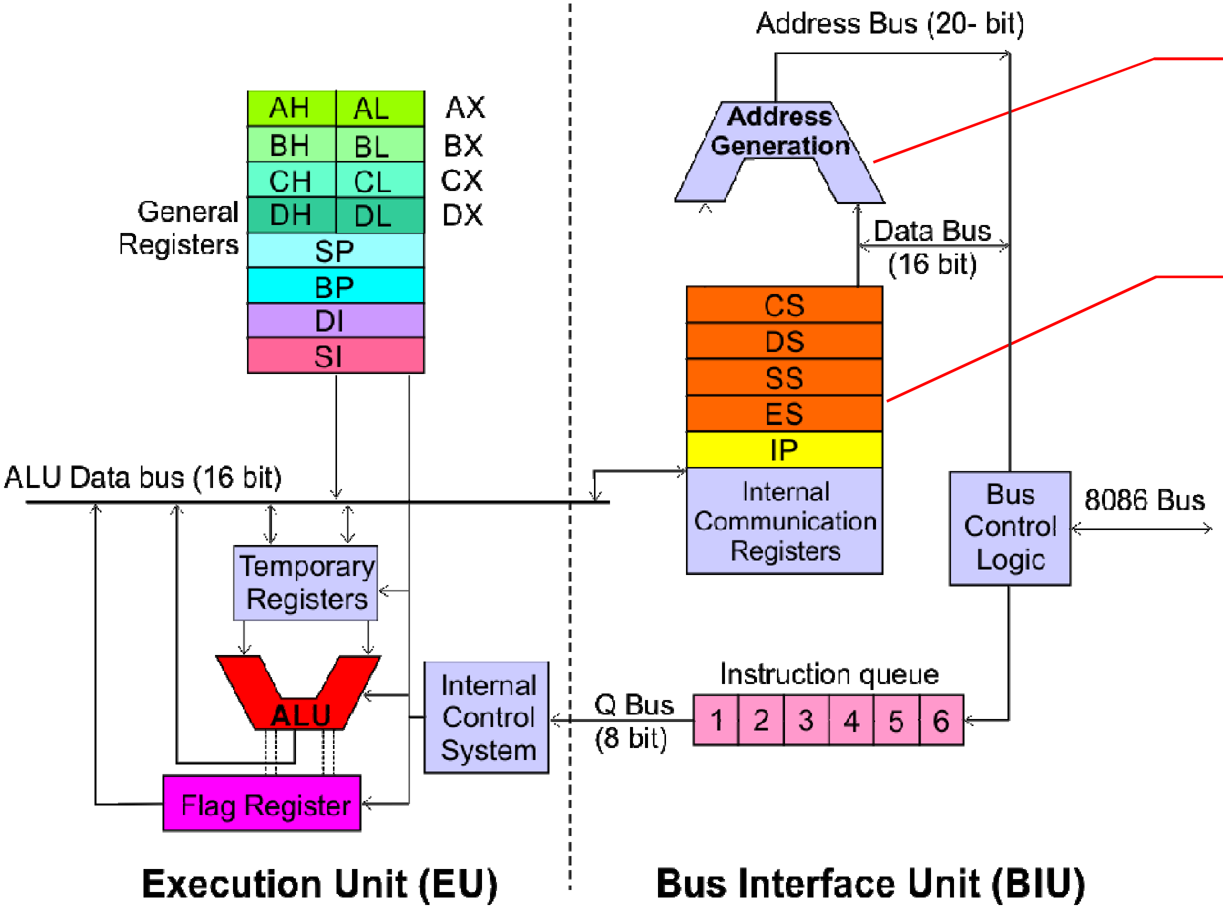
BIU and EU functions separately.

Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/ O ports.

Architecture

Bus Interface Unit (BIU)

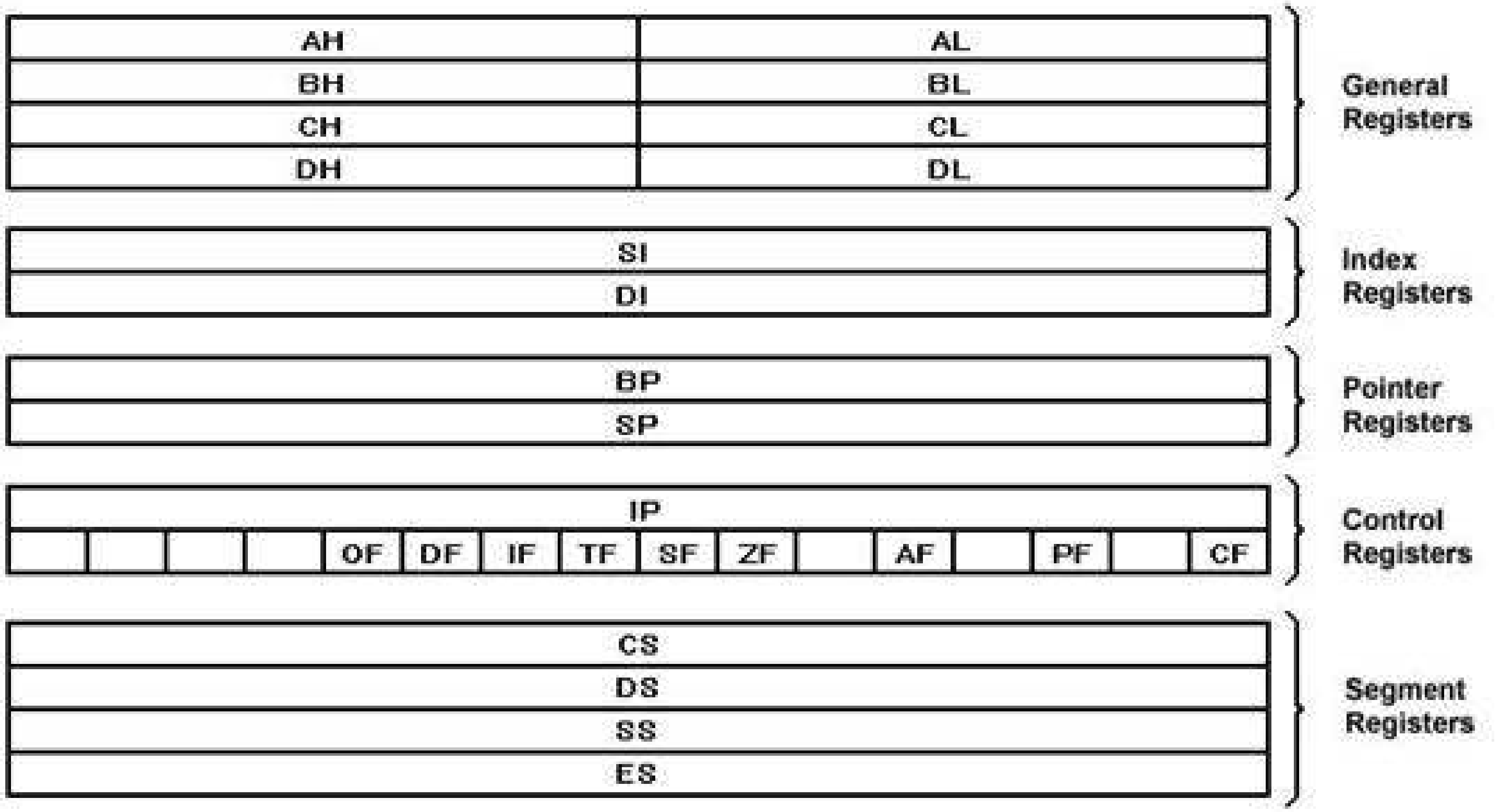


Dedicated Adder to generate 20 bit address

Four 16-bit segment registers

Code Segment (CS)
Data Segment (DS)
Stack Segment (SS)
Extra Segment (ES)

Microprocessor Architecture



AH	AL
BH	BL
CH	CL
DH	DL

General Registers

SI

DI

Index Registers

BP

SP

Pointer Registers

IP

				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Control Registers

CS

DS

SS

ES

Segment Registers

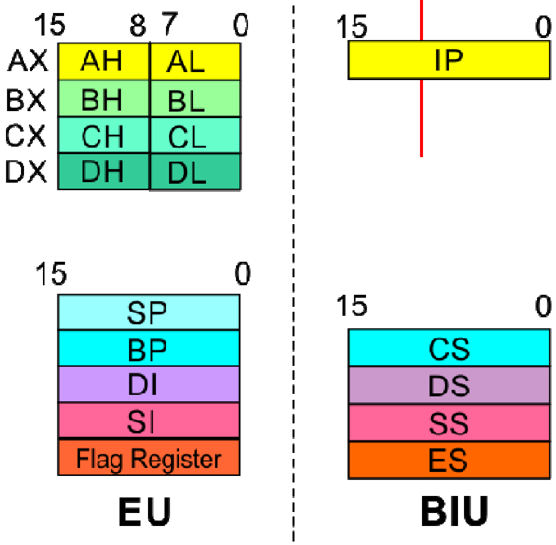
Architecture

Bus Interface Unit (BIU)

Segment Registers

Data Segment Register

- **16-bit**
- **Points to the current data segment; operands for most instructions are fetched from this segment.**
- **The 16-bit contents of the Source Index (SI) or Destination Index (DI) or a 16-bit displacement are used as offset for computing the 20-bit physical address.**



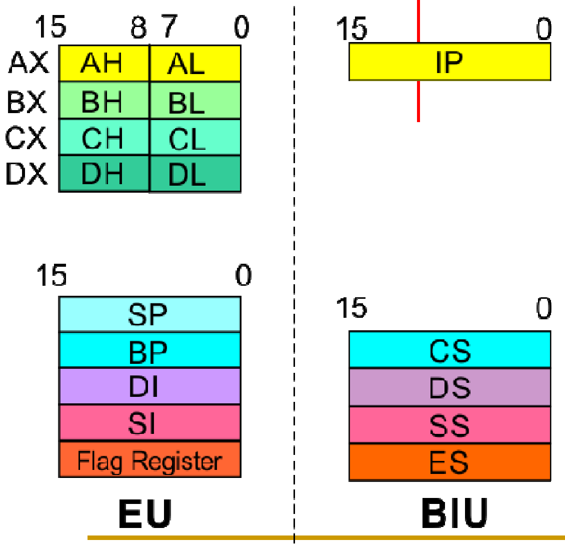
Architecture

Bus Interface Unit (BIU)

Segment Registers

Stack Segment Register

- 16-bit
- Points to the current stack.
- The 20-bit physical stack address is calculated from the Stack Segment (SS) and the Stack Pointer (SP) for stack instructions such as PUSH and POP.
- In based addressing mode, the 20-bit physical stack address is calculated from the Stack segment (SS) and the Base Pointer (BP).



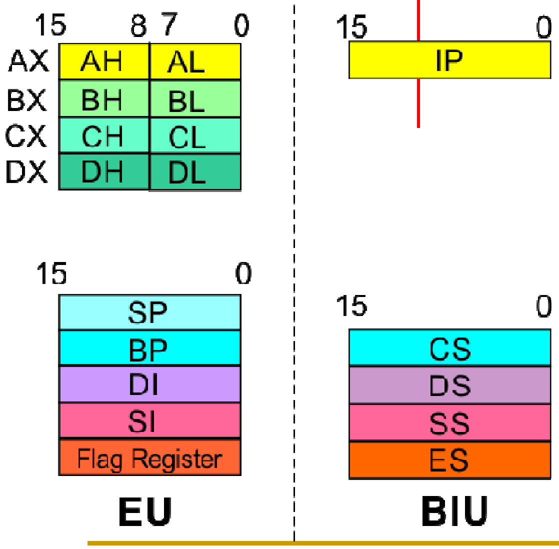
Architecture

Bus Interface Unit (BIU)

Segment Registers

Extra Segment Register

- **16-bit**
- **Points to the extra segment in which data (in excess of 64K pointed to by the DS) is stored.**
- **String instructions use the ES and DI to determine the 20-bit physical address for the destination.**



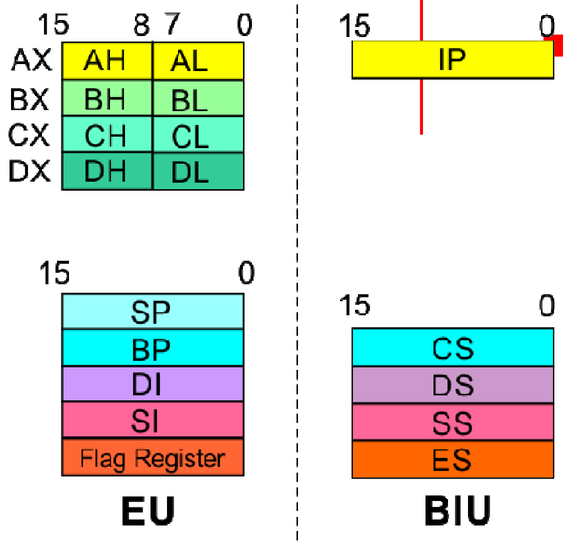
Architecture

Bus Interface Unit (BIU)

Segment Registers

Instruction Pointer

- **16-bit**
- **Always points to the next instruction to be executed within the currently executing code segment.**
- **So, this register contains the 16-bit offset address pointing to the next instruction code within the 64Kb of the code segment area.**
- **Its content is automatically incremented as the execution of the next instruction takes place.**



Multipurpose Registers

- AX (accumulator) 16-bit register
  - The accumulator is used for instructions such as multiplication, division, and some of the adjustment instructions. For these instructions, the accumulator has a special purpose, but is generally considered to be a multipurpose register.
- BX (base index) register
  - Sometimes holds the offset address of a location in the memory system in all versions of the microprocessor.
- CX, CH, or CL, is a (count) register
  - General-purpose register that also holds the count for various instructions.

Multipurpose Registers

- CX, CH, or CL, is a (count) register
  - General-purpose register that also holds the count for various instructions.
- DX , DH, or DL, is a (data) register
  - It holds a part of the result from a multiplication or part of the dividend before a division.
- DI (destination index)
  - It often addresses string destination data for the string instructions.
- SI (source index)
  - The source index register often addresses source string data for the string instructions.

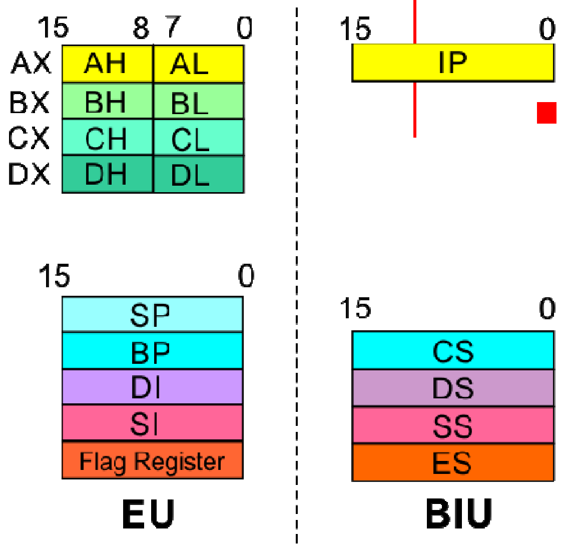
Special-Purpose Registers.

- The special-purpose registers include IP, SP, FLAGS and the segment registers include CS, DS, ES, SS, FS, and GS.
- IP (instruction pointer)
  - It addresses the next instruction in a section of memory defined as a code segment
- SP (stack pointer)
  - The stack memory stores data through this pointer.

Architecture

Execution Unit (EU)

EU Registers



Accumulator Register (AX)

- Consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX.
- AL in this case contains the low order byte of the word, and AH contains the high-order byte.
- The I/O instructions use the AX or AL for inputting / outputting 16 or 8 bit data to or from an I/O port.
- Multiplication and Division instructions also use the AX or AL.

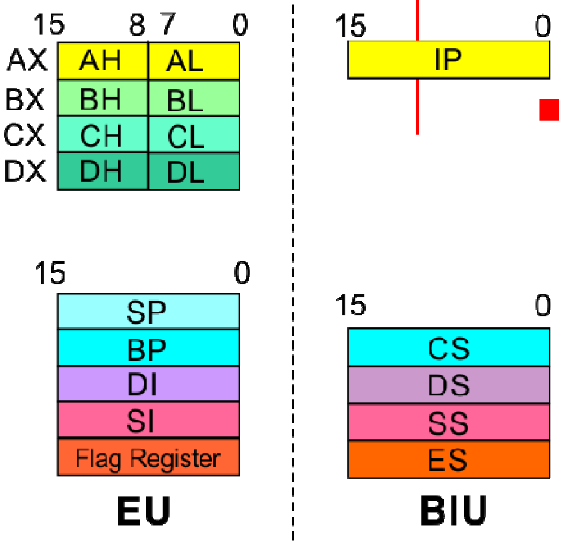
Architecture

Execution Unit (EU)

EU Registers

Base Register (BX)

- Consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX.
- BL in this case contains the low-order byte of the word, and BH contains the high-order byte.
- This is the only general purpose register whose contents can be used for addressing the 8086 memory.
- All memory references utilizing this register content for addressing use DS as the default segment register.



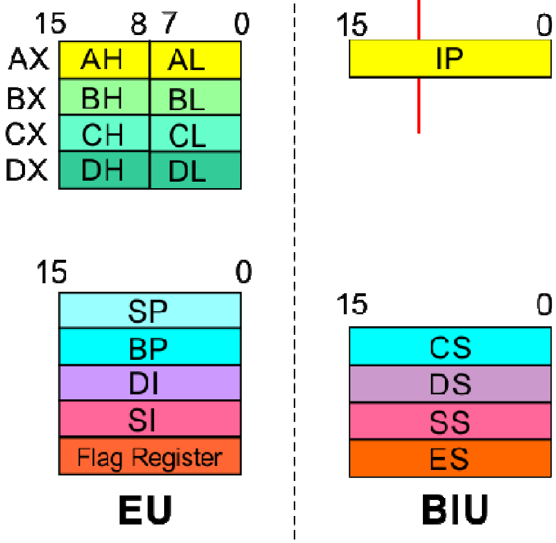
Architecture

Execution Unit (EU)

EU Registers

Counter Register (CX)

- **Consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX.**
- **When combined, CL register contains the low order byte of the word, and CH contains the high-order byte.**
- **Instructions such as SHIFT, ROTATE and LOOP use the contents of CX as a counter.**



Example:

The instruction LOOP START automatically decrements CX by 1 without affecting flags and will check if [CX] = 0.

If it is zero, 8086 executes the next instruction; otherwise the 8086 branches to the label START.

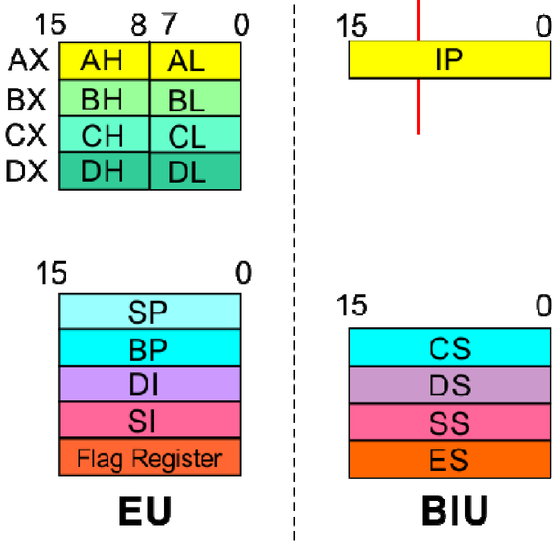
Architecture

Execution Unit (EU)

EU Registers

Data Register (DX)

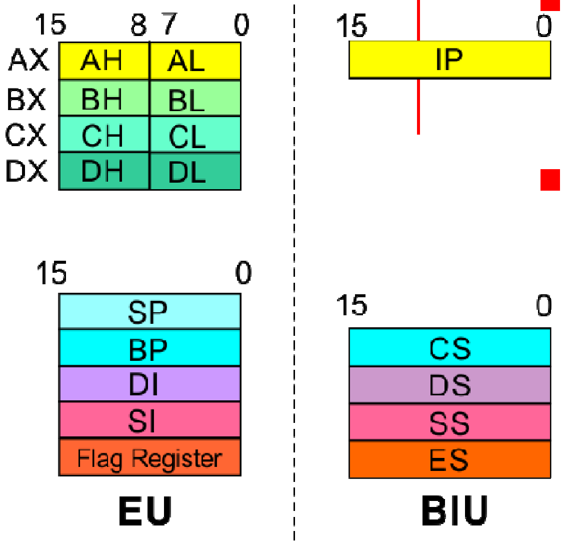
- Consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX.
- When combined, DL register contains the low order byte of the word, and DH contains the high-order byte.
- Used to hold the high 16-bit result (data) in 16 X 16 multiplication or the high 16-bit dividend (data) before a 32 ÷ 16 division and the 16-bit reminder after division.



Architecture

Execution Unit (EU)

EU Registers



Stack Pointer (SP) and Base Pointer (BP)

- **SP and BP are used to access data in the stack segment.**
- **SP is used as an offset from the current SS during execution of instructions that involve the stack segment in the external memory.**
- **SP contents are automatically updated (incremented/ decremented) due to execution of a POP or PUSH instruction.**
- **BP contains an offset address in the current SS, which is used by instructions utilizing the based addressing mode.**

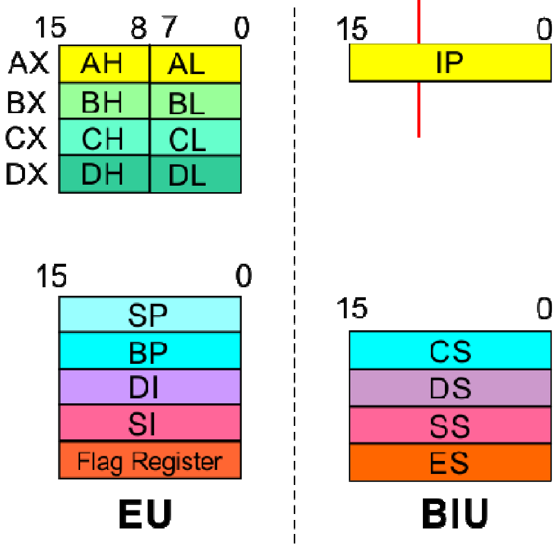
Architecture

Execution Unit (EU)

EU Registers

Source Index (SI) and Destination Index (DI)

- Used in indexed addressing.
- Instructions that process data strings use the SI and DI registers together with DS and ES respectively in order to distinguish between the source and destination addresses.



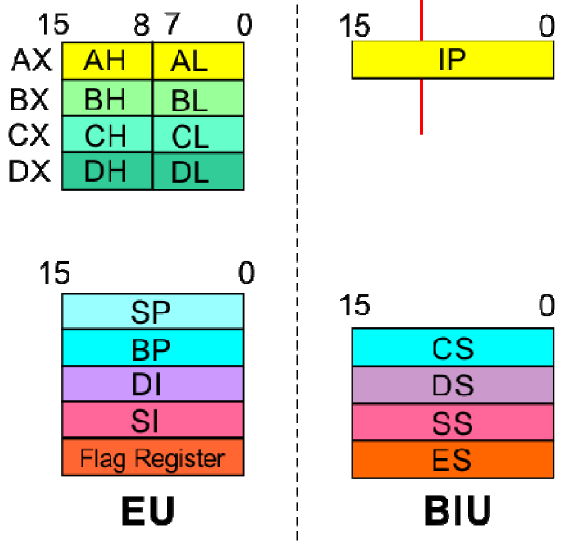
Architecture

Execution Unit (EU)

EU Registers

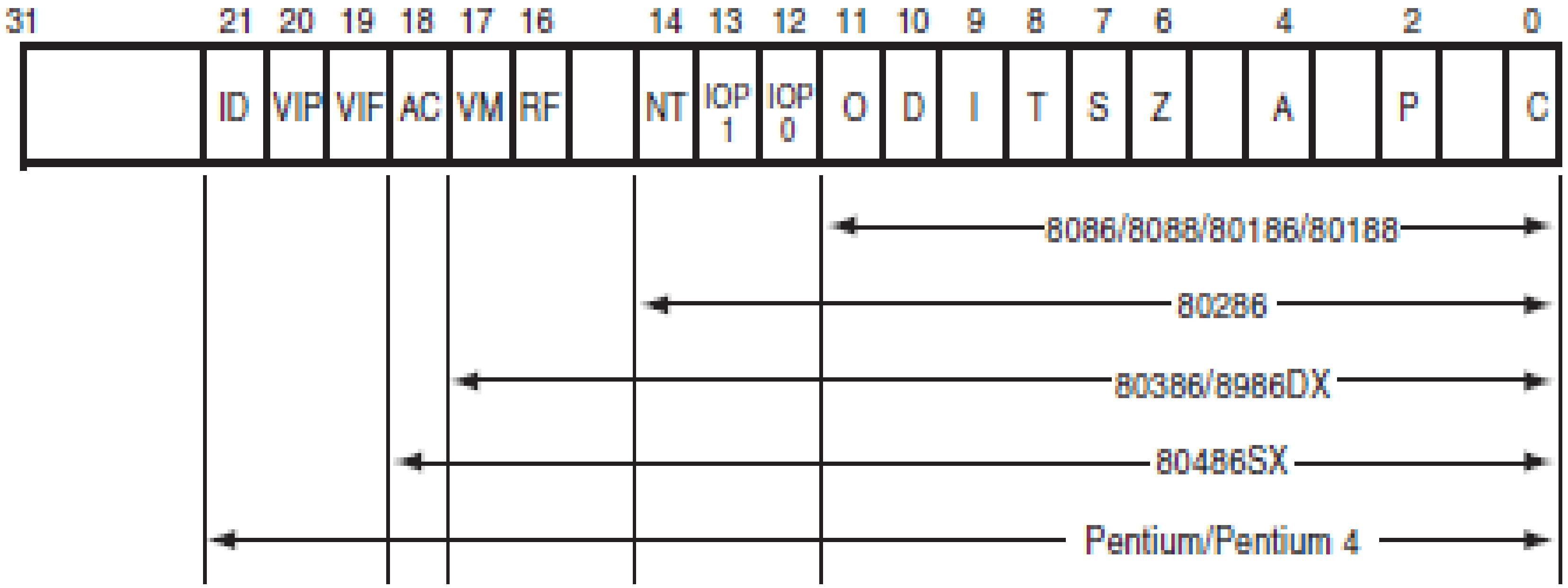
Source Index (SI) and Destination Index (DI)

- Used in indexed addressing.
- Instructions that process data strings use the SI and DI registers together with DS and ES respectively in order to distinguish between the source and destination addresses.



Flags

- It indicate the condition of the microprocessor and control its operation.



Flags

- It indicate the condition of the microprocessor and control its operation.
- C (carry)
  - Carry holds the carry after addition or the borrow after subtraction. The carry flag also indicates error conditions, as dictated by some programs and procedures.
- P (parity)
  - Parity is a logic 0 for odd parity and a logic 1 for even parity. Parity is the count of ones in a number expressed as even or odd.

Flags

- 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
- Z (zero)
  - The zero flag shows that the result of an arithmetic or logic operation is zero.
- S (sign)
  - The sign flag holds the arithmetic sign of the result after an arithmetic or logic instruction executes.
-

Flags

- T (trap)
  - The trap flag enables trapping through an on-chip debugging feature.
- I (interrupt)
  - The interrupt flag controls the operation of the INTR (interrupt request) input pin.
- D (direction)
  - The direction flag selects either the increment or decrement mode for the DI and/or SI registers during string instructions.

Flags

- (overflow)
  - Overflows occur when signed numbers are added or subtracted. An overflow indicates that the result has exceeded the capacity of the machine
- NT (nested task)
  - The nested task flag indicates that the current task is nested within another task in protected mode operation. This flag is set when the task is nested by software.
- RF (resume)
  - The resume flag is used with debugging to control the resumption of execution after the next instruction.

Architecture

Execution Unit (EU)

Flag Register

Auxiliary Carry Flag

This is set, if there is a carry from the lowest nibble, i.e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

Carry Flag

This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction.

Sign Flag

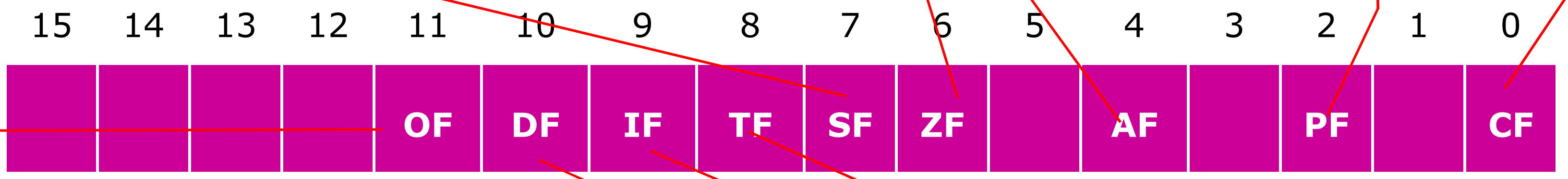
This flag is set, when the result of any computation is negative

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.



Over flow Flag

This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

Tarp Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag

This is used by string manipulation instructions. If this flag bit is '0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts; clearing IF disables these interrupts.

Segment Register

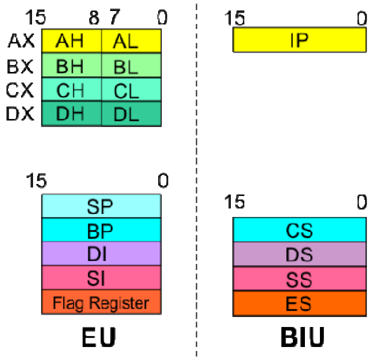
- Additional registers, called segment registers, generate memory addresses when combined with other registers in the microprocessor.
- CS (code)
  - The code segment is a section of memory that holds the code (programs and procedures) used by the microprocessor.
- DS (data)
  - The data segment is a section of memory that contains most data used by a program.

Segment Register

- ES (extra)
  - The extra segment is an additional data segment that is used by some of the string instructions to hold destination data.
- SS (stack)
  - The stack segment defines the area of memory used for the stack.

Architecture

8086 registers categorized into 4 groups



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Sl.No.	Type	Register width	Name of register
1	General purpose register	16 bit	AX, BX, CX, DX
		8 bit	AL, AH, BL, BH, CL, CH, DL, DH
2	Pointer register	16 bit	SP, BP
3	Index register	16 bit	SI, DI
4	Instruction Pointer	16 bit	IP
5	Segment register	16 bit	CS, DS, SS, ES
6	Flag (PSW)	16 bit	Flag register

Architecture

Registers and Special Functions

Register	Name of the Register	Special Function
AX	16-bit Accumulator	Stores the 16-bit results of arithmetic and logic operations
AL	8-bit Accumulator	Stores the 8-bit results of arithmetic and logic operations
BX	Base register	Used to hold base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions
DX	Data Register	Used to hold data for multiplication and division operations
SP	Stack Pointer	Used to hold the offset address of top stack memory
BP	Base Pointer	Used to hold the base value in base addressing using SS register to access data from stack memory
SI	Source Index	Used to hold index value of source operand (data) for string instructions
DI	Data Index	Used to hold the index value of destination operand (data) for string operations