CSE 4621
Microprocessor and Interfacing

Lecture-1

Class-1,2,3,4, and 5

Concept of Microcomputer systems & Introduction to 8086 Microprocessor

HISTORICAL PERSPECTIVE

- **1st generation: 1945 - 1955**
  - **Tubes, punchcards**
- **2nd generation : 1955 - 1965**
  - **transistors**
- **3rd generation: 1965 – 1980**
  - **Integrated circuits**
- **4th generation: 1980 –**
  - **PCs and workstations**

1st generation (1945-1955)

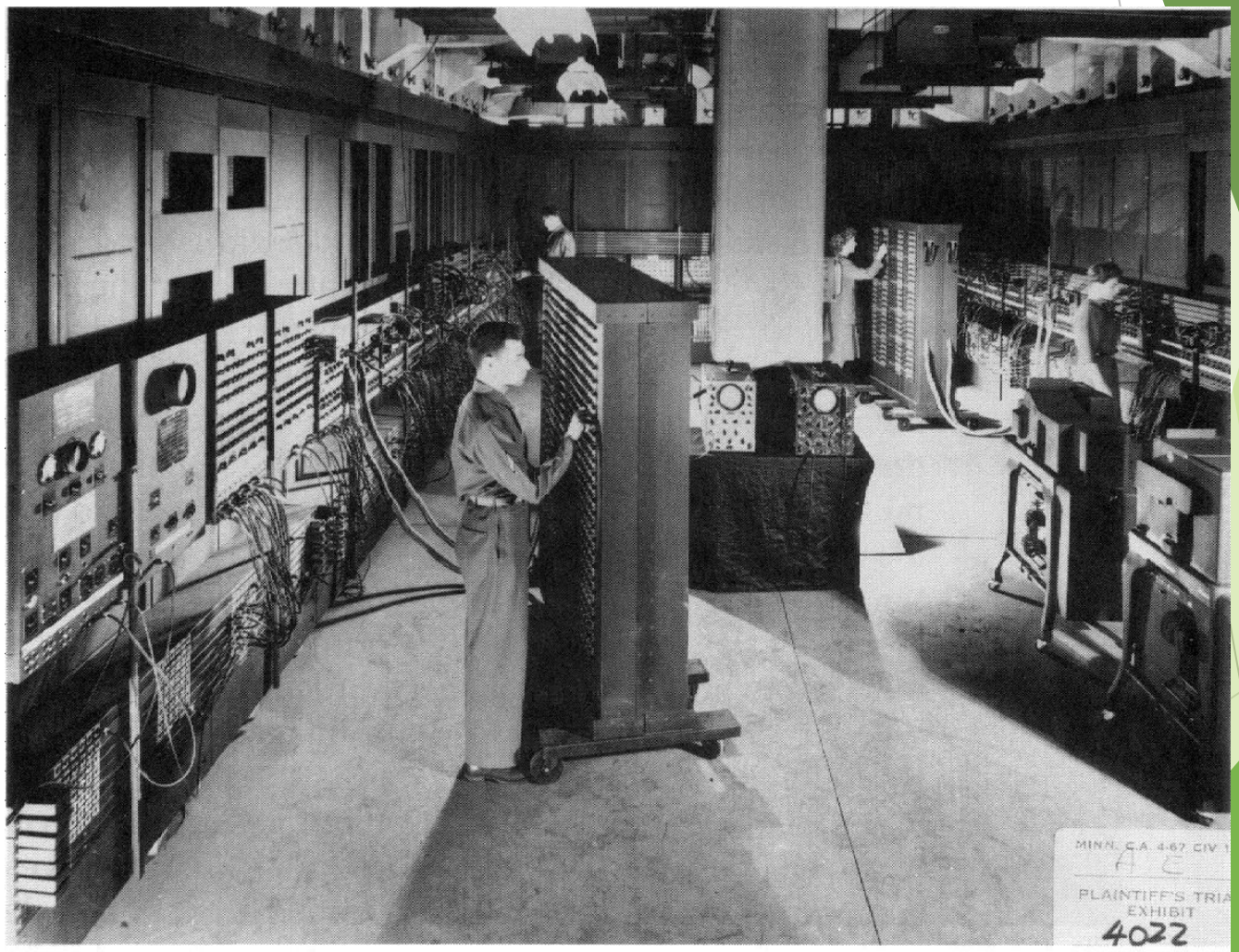
- **Programming was done in machine language**
- **No operating system**

ENIAC –
The first electronic computer (1946)

18,000 tubes

300 Tn

170 KWatt



2nd generation (1955-1965)

- Transistor-based
- Fairly reliable
- Only afforded by governments, universities or large companies (millions $)

2nd generation (1955-1965)

- **Program was first written on paper (FORTRAN) and then punched into cards**
- **Cards were then delivered to the user.**
- **Mostly used for scientific and technical calculations**
  - **Solving differential equations**

3rd generation (1965-1980)

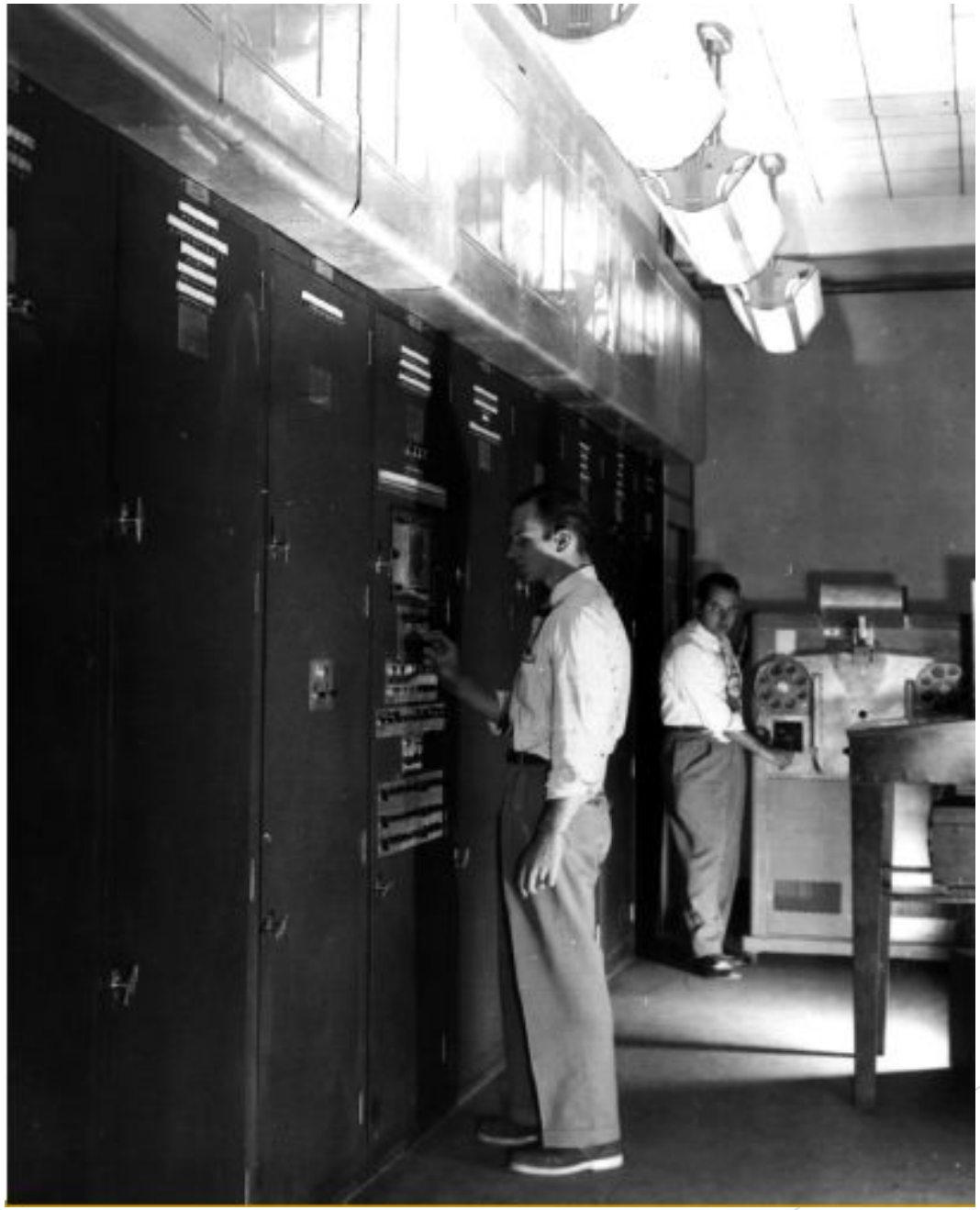
- **IC-based operation**
- **IBM develops compatible systems**
- **Tradeoffs in performance, memory, I/O etc).**
- **Greater MHz/$**

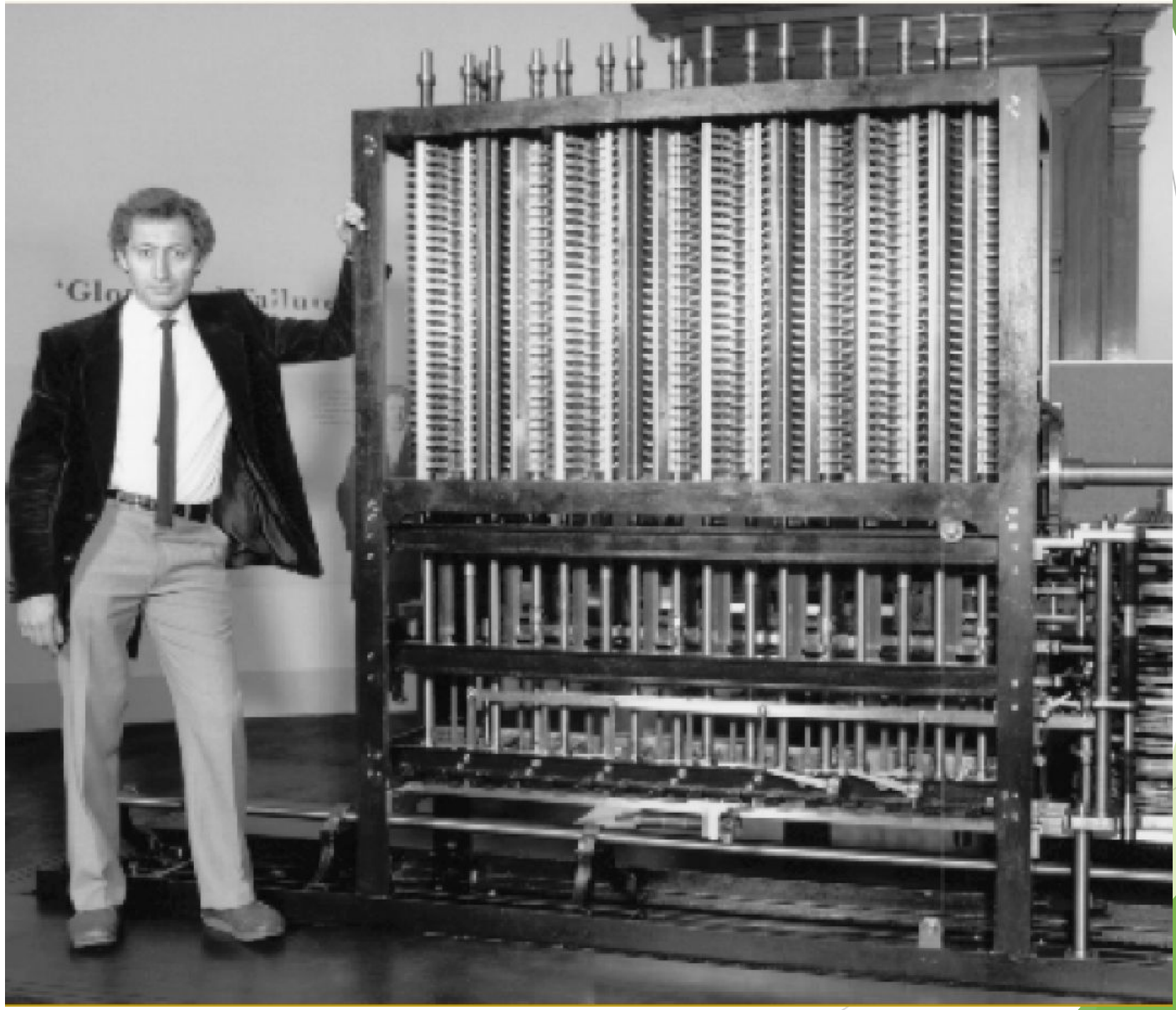
4th generation (1980-1990)

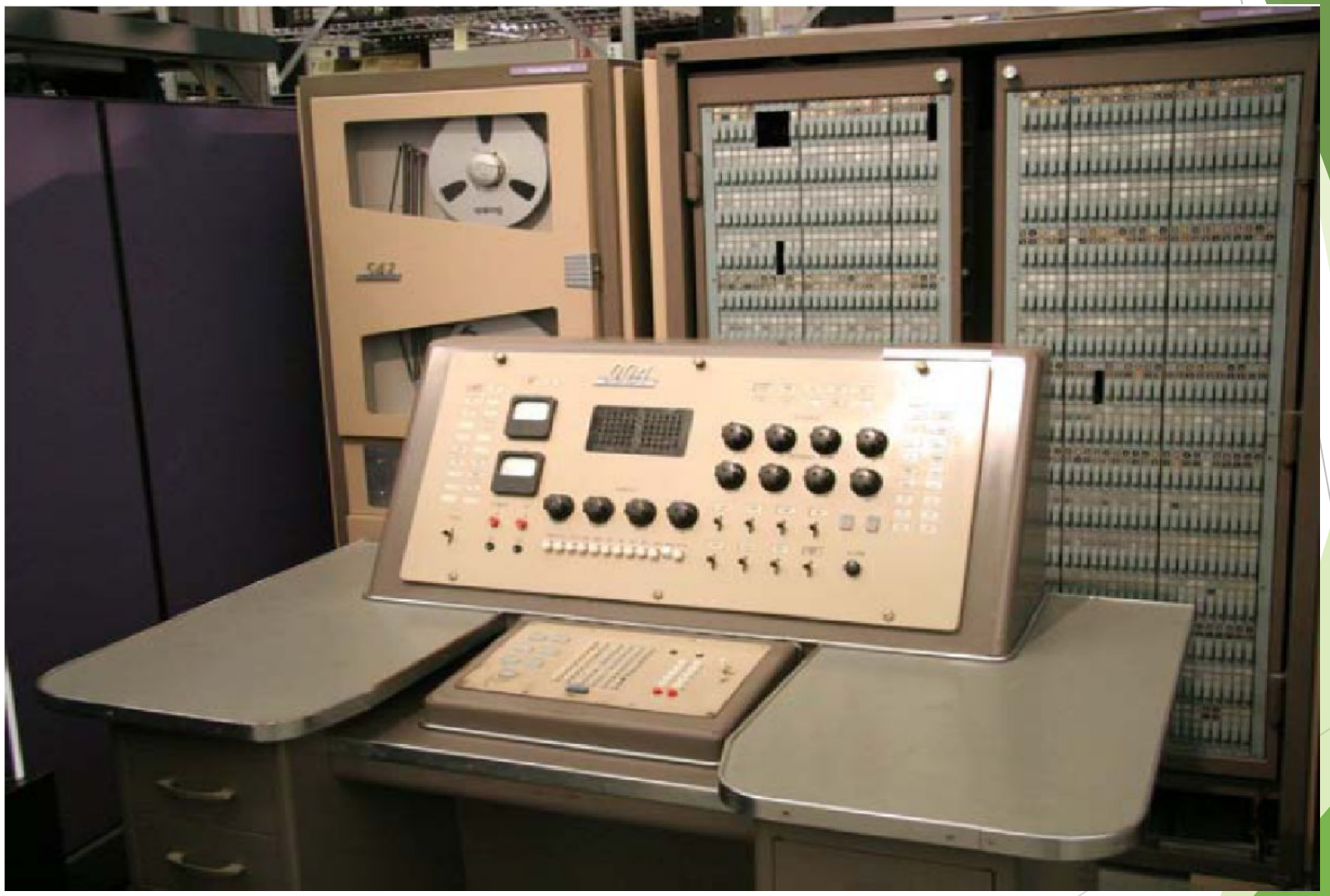
- LSI-based (large-scale integration) PCs
- Significantly cheaper
- User-friendly software
- 2 dominant operating systems:
  - MS DOS: IBM PC (8088, 80286, 80386, 80486)
  - UNIX: RISC workstations

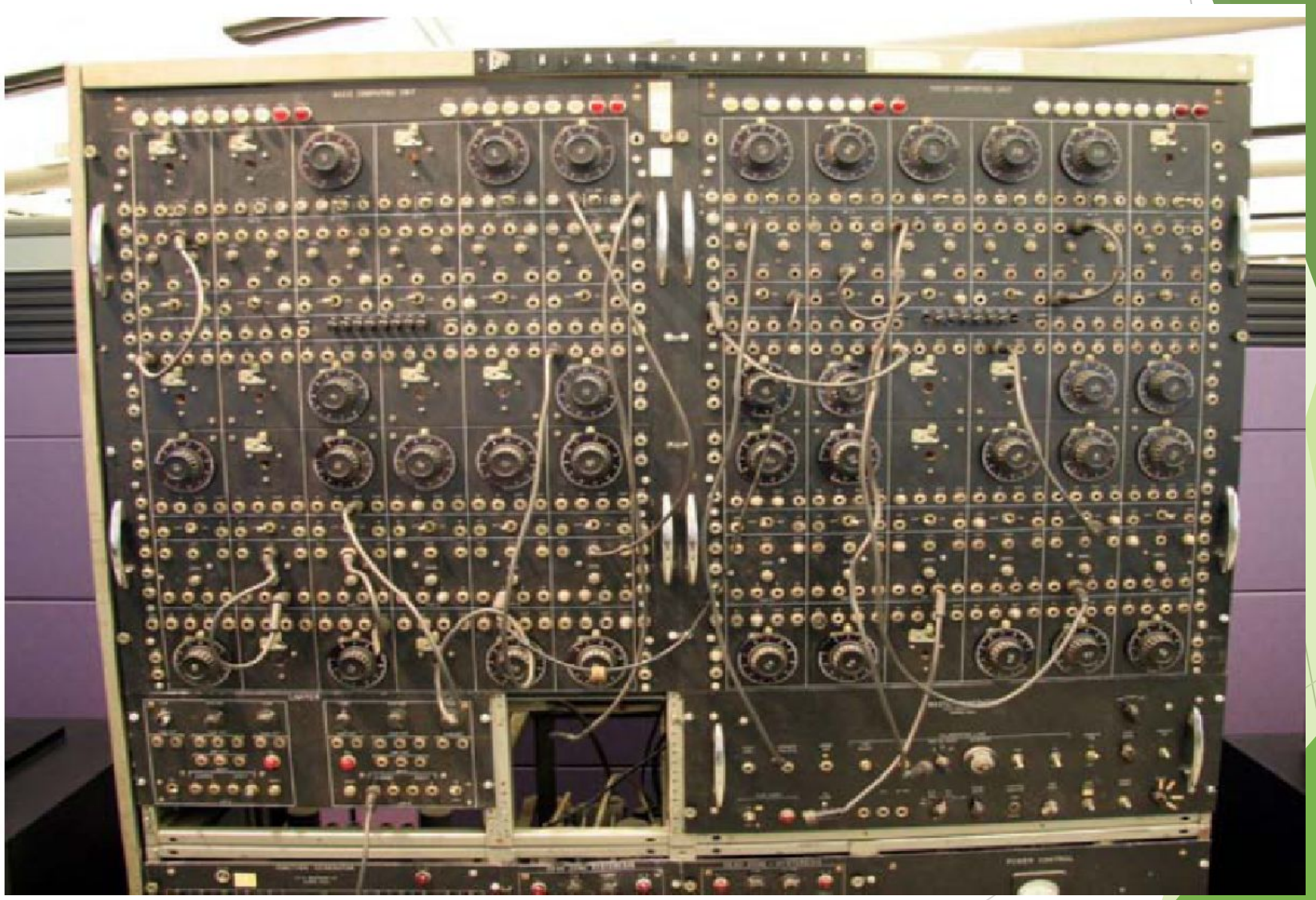
5th generation (1990-)

- **PC networks**
- **Network operating systems**
- **Each machine runs its own operating system**
- **Users don’t care where their programs are being executed**









A COMPUTER SYSTEM

A **computer** is a programmable machine that receives input, stores and manipulates data//information, and provides output ina useful format.

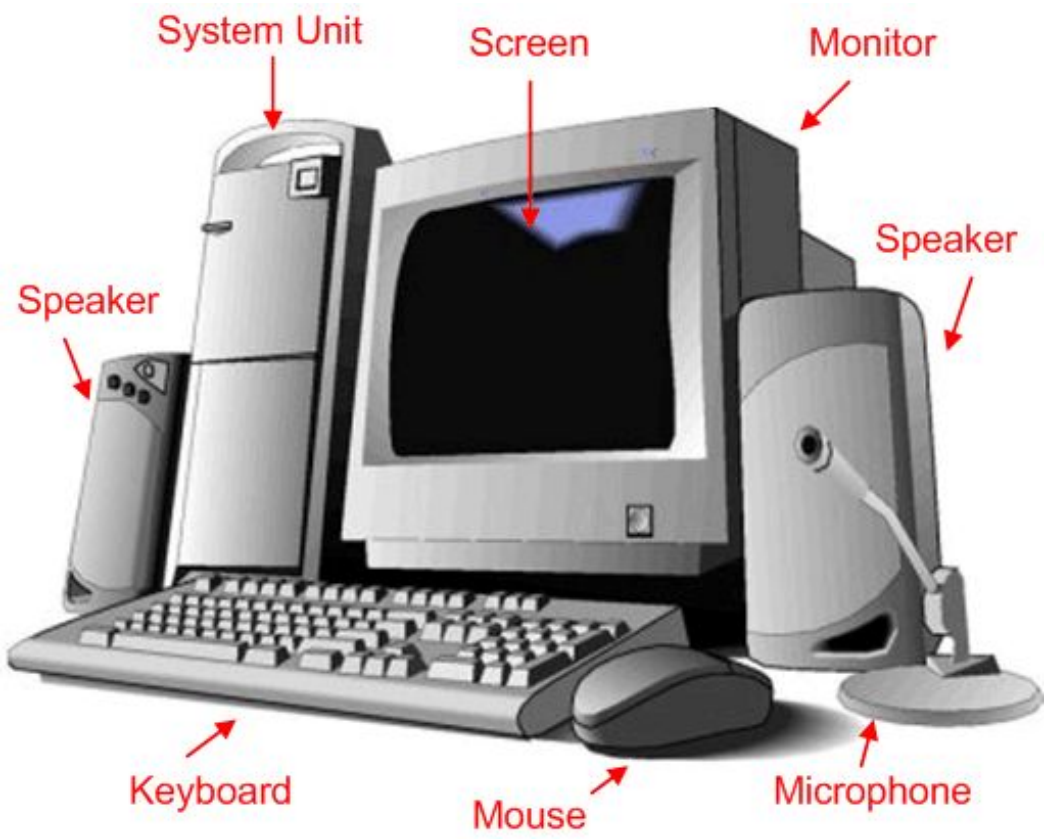
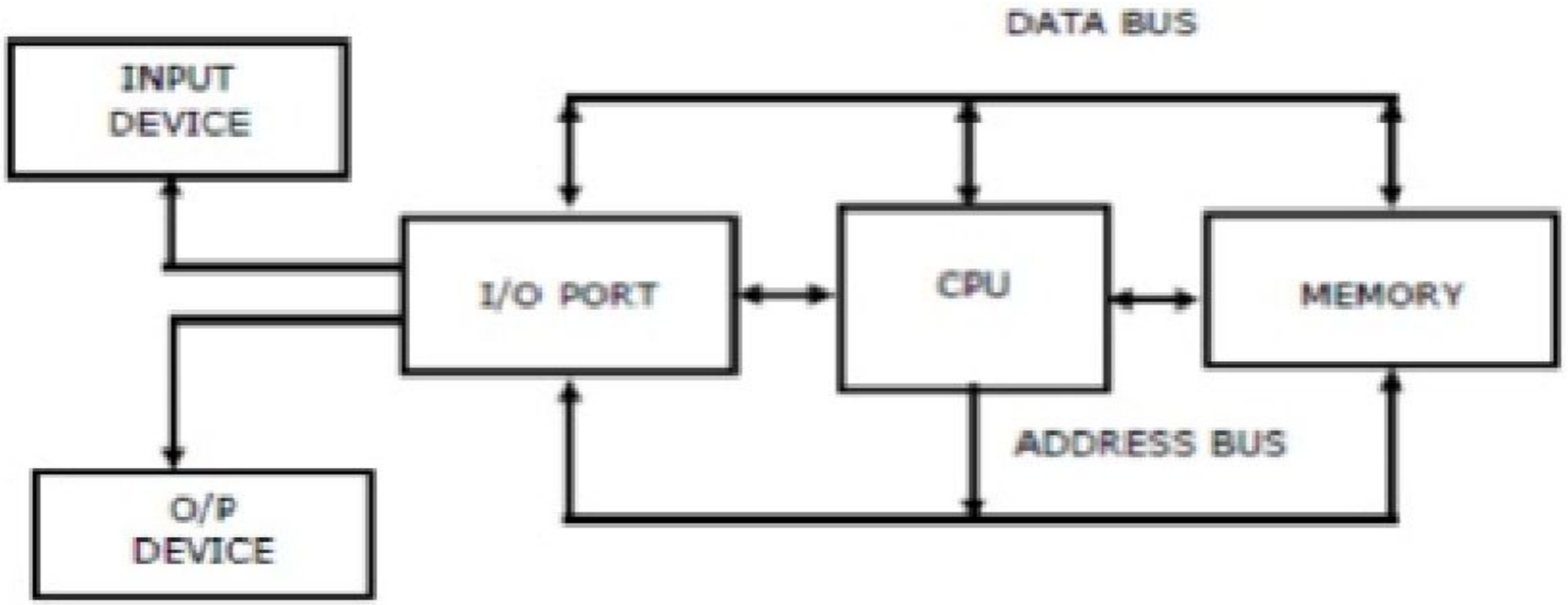


Diagram Of A Computer System

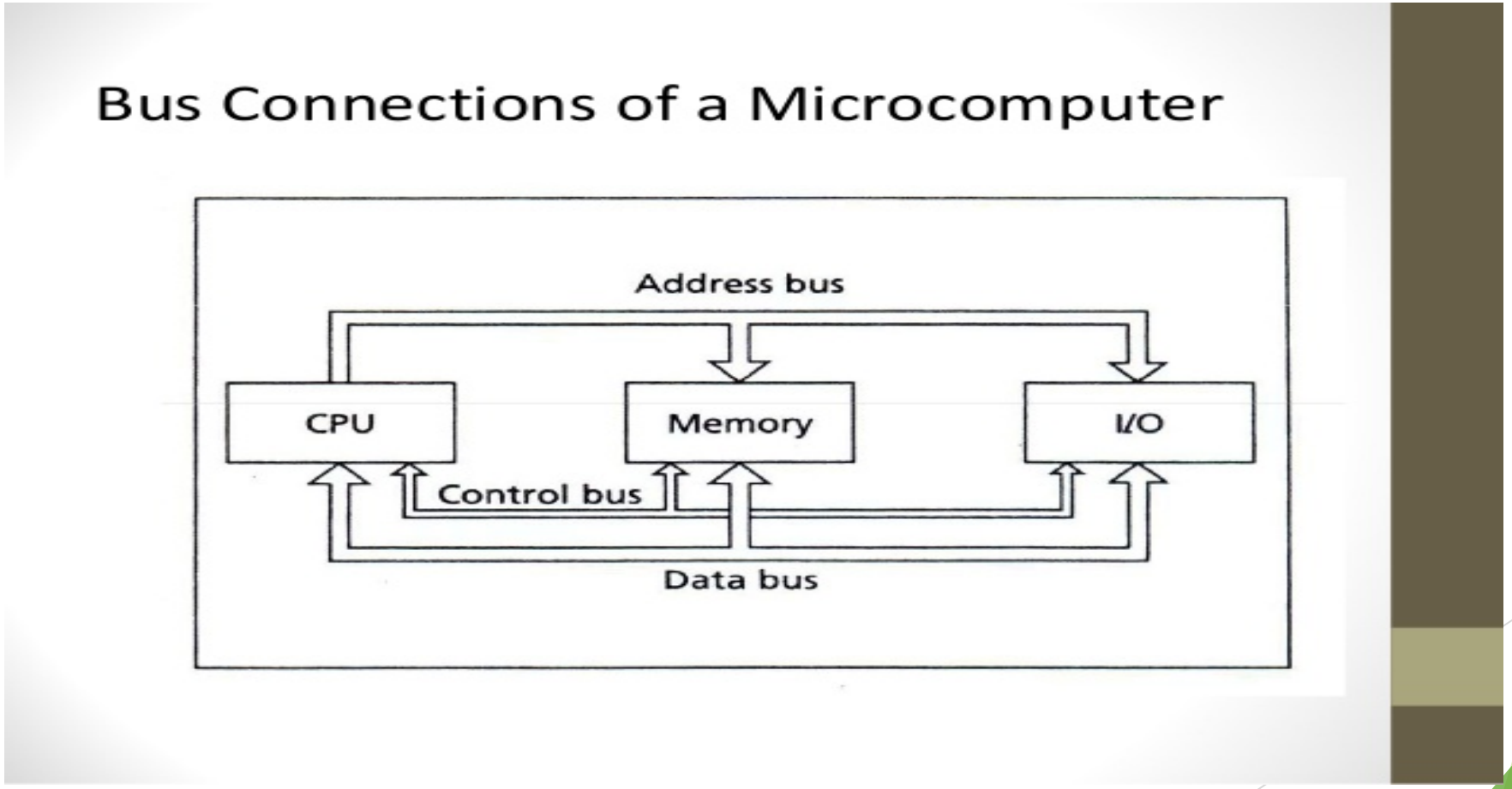
BLOCK DIAGRAM OF A BASIC COMPUTER SYSTEM

Basic computer system consist of a Central processing unit (CPU), memory (RAM and ROM), input/output (I/O) unit.



Block diagram of a basic computer system

BLOCK DIAGRAM OF A BASIC COMPUTER SYSTEM



Types of Computers

- Mainframes
- Minicomputer
- Microcomputer
  - CPU is usually a single integrated circuit called a **microprocessor**

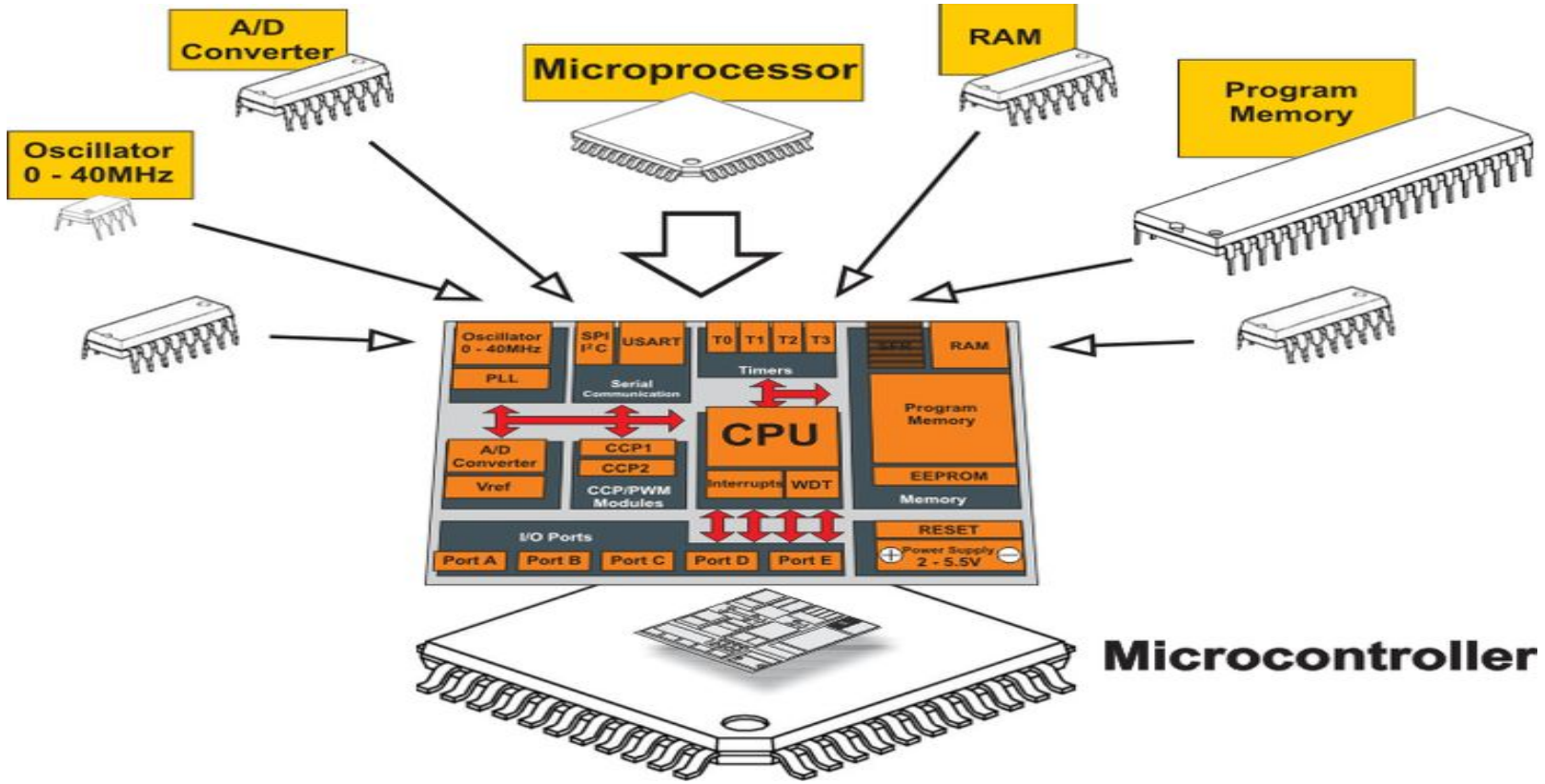
What is a Microprocessor?

- A Controlling unit of a micro-computer wrapped inside a small chip.
- Performs Arithmetic Logical Unit (ALU) operations and communicates with the other devices connected with it.
- It is a single Integrated Circuit in which several functions are combined.

What is a Micro Controller?

- If a microprocessor, its associated support circuitry, memory and peripheral I/O components are implemented on a single chip, it is a *microcontroller*.
- Microprocessor is used in Personal Computers whereas Micro Controller is used in an embedded system.

What is a Micro Controller?



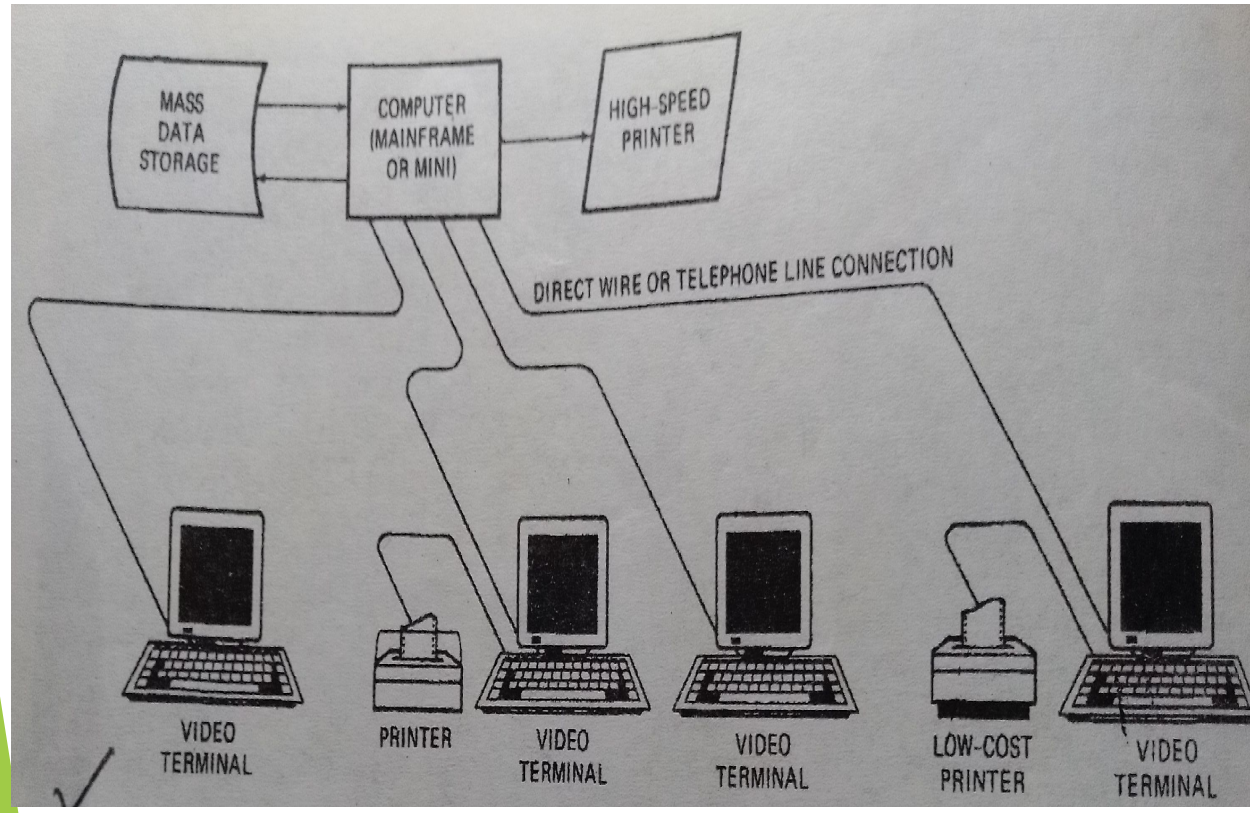
Microprocessor vs Micro Controller

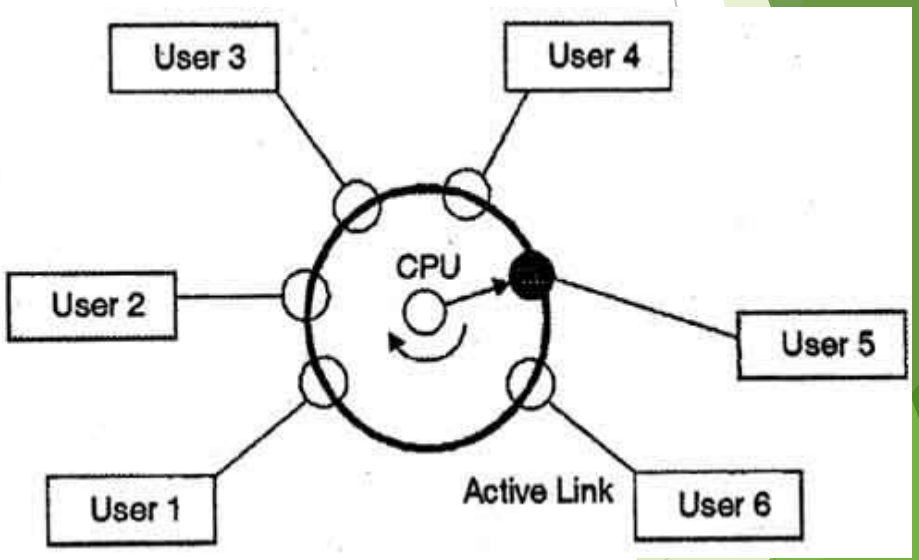
Microprocessor	Microcontroller
Microprocessor is the heart of Computer system.	Micro Controller is the heart of an embedded system.
It is only a processor, so memory and I/O components need to be connected externally	Micro Controller has a processor along with internal memory and I/O components.
Memory and I/O has to be connected externally, so the circuit becomes large.	Memory and I/O are already present, and the internal circuit is small.
You can't use it in compact systems	You can use it in compact systems.
Cost of the entire system is high	Cost of the entire system is low

How Computers and Microcomputers are Used:

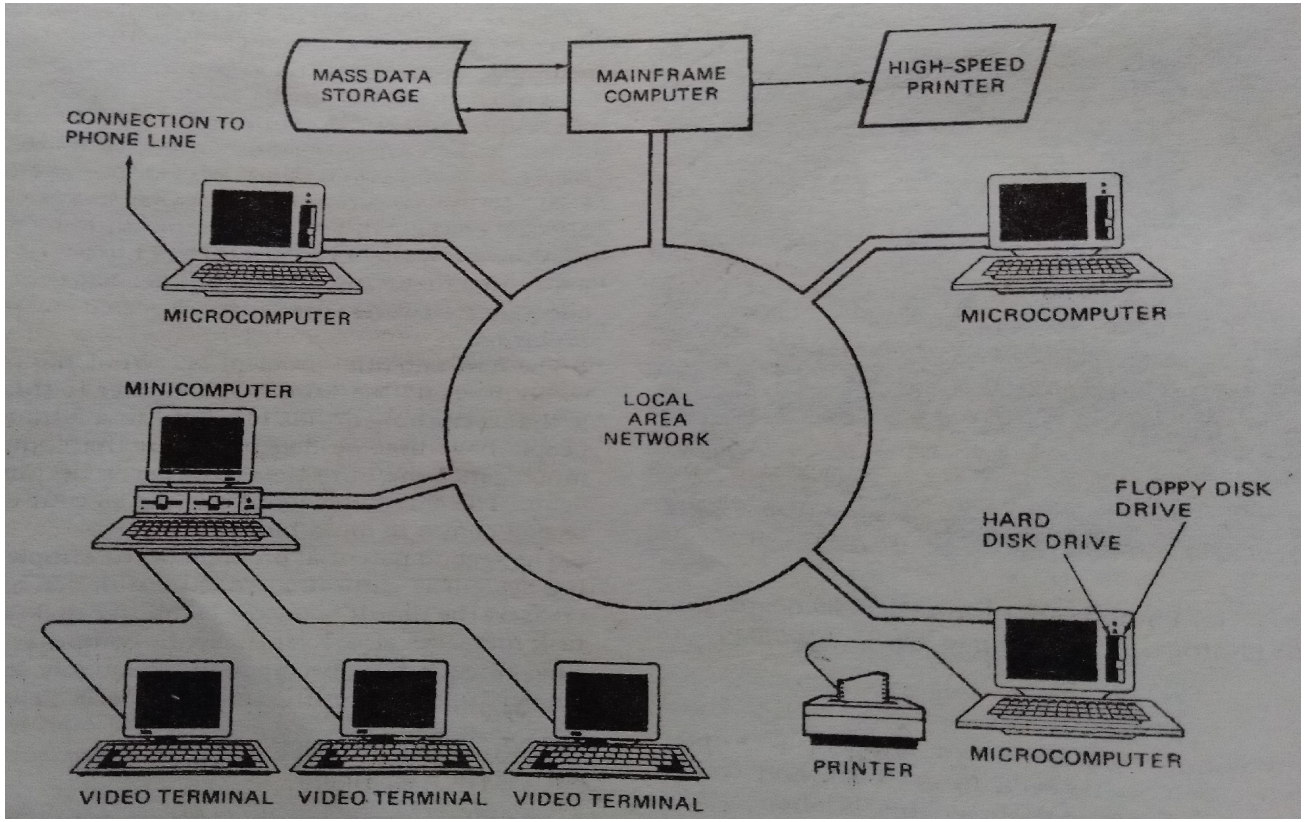
1. Timesharing and Multitasking Systems
2. Distributed Processing or Multiprocessing System

How Computers and Microcomputers are Used:
Computer Timesharing System





How Computers and Microcomputers are Used: Distributed Processing or Multiprocessing System

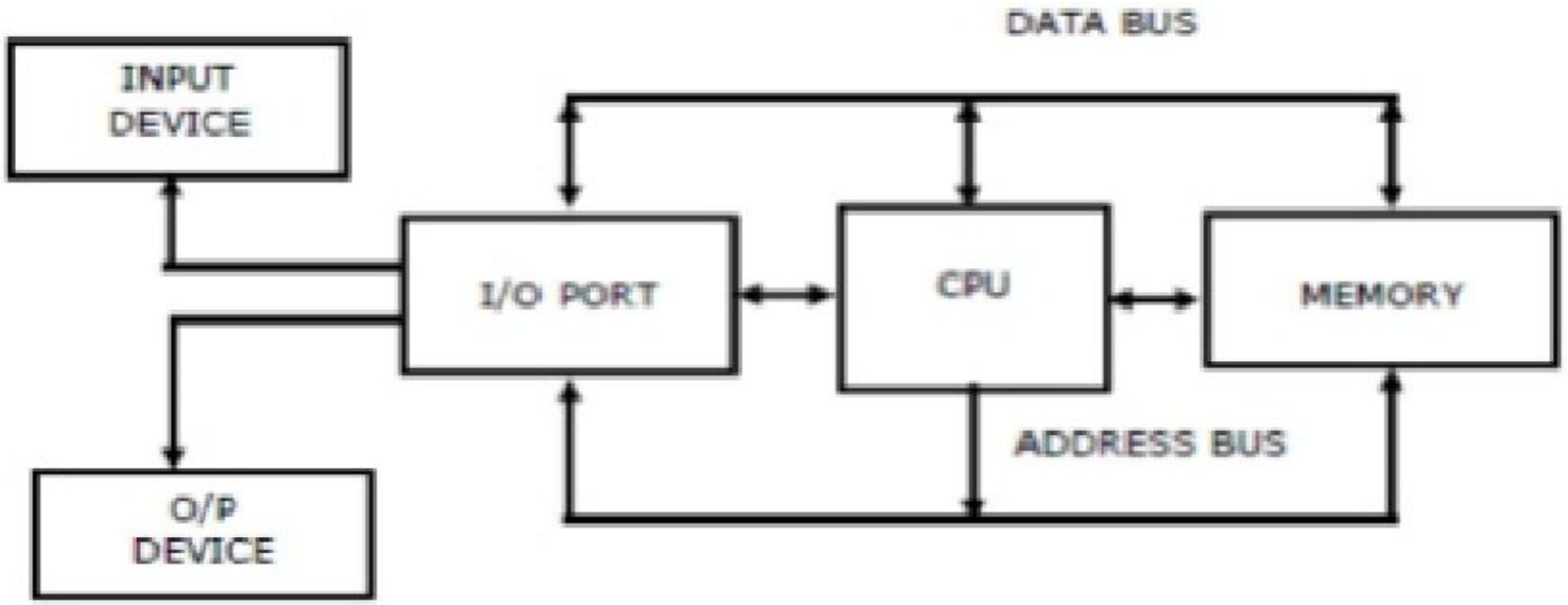


Computer Circuits

- **Integrated Circuit (IC) chips are used**
- Each IC chip may contain even thousands of transistors
- All IC operates on discreate voltage levels
- Computer Circuits consist of 3 parts:
  - CPU
  - Memory Circuits
  - I/O Circuits.

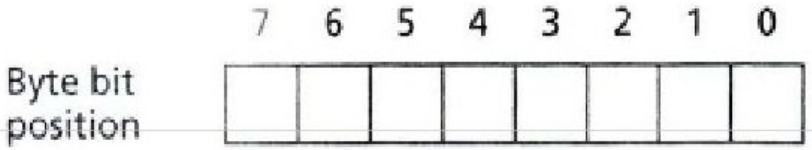
BLOCK DIAGRAM OF A BASIC COMPUTER SYSTEM

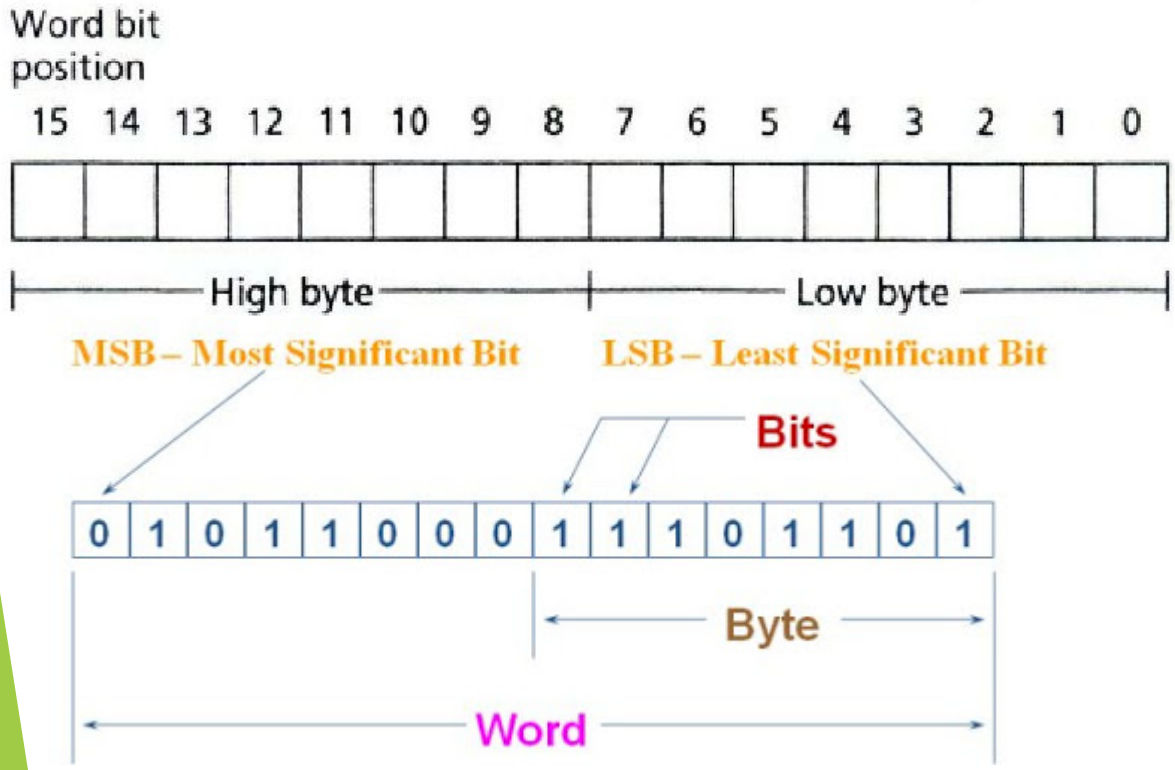
Basic computer system consist of a Central processing unit (CPU), memory (RAM and ROM), input/output (I/O) unit connected by 3 kinds of Buses.



Block diagram of a basic computer system

Memory





- Bytes and Word
- Address vs Content
- Bit Position
- Memory Operation
  1. Read (fetch)
  2. Write (store)
- RAM and ROM
- Firmware

Buses

- Processor communicates with memory and I/O circuits using signals
- Along a set of wires called Buses
- 3 kinds of signals
  1. Address signal
  2. Data signal
  3. Control signal
- 3 kind of buses
  1. Address bus
  2. Data bus
  3. Control bus

Buses:

- Address bus:
  - Consists of 16, 20, 24, 0r 32 parallel signal lines
  - Number of memory locations that CPU can address is determined by number of Address line
- Data bus:
  - Consists of 8, 16, 0r 32 parallel and bidirectional signal lines
  - May Output devices are connected to data bus
  - Only one at a time will have its output enabled
    - Must have 3 state outputs

Buses:

- Control bus:
  - Consists of 4 to 10 parallel signal lines
  - Typical control bus signals:
    - Memory read
    - Memory write
    - I/O read
    - I/O write

Input/output

- Peripherals
- I/O ports:
  - Actual physical device used to interface computer buses to external systems
  - Functions like shipping ports in a country
  - **Serial ports : 1 bit at a time**
  - **Parallel Ports : 8 or 16 bit at a time**
  - **Serial ports vs Parallel Ports**

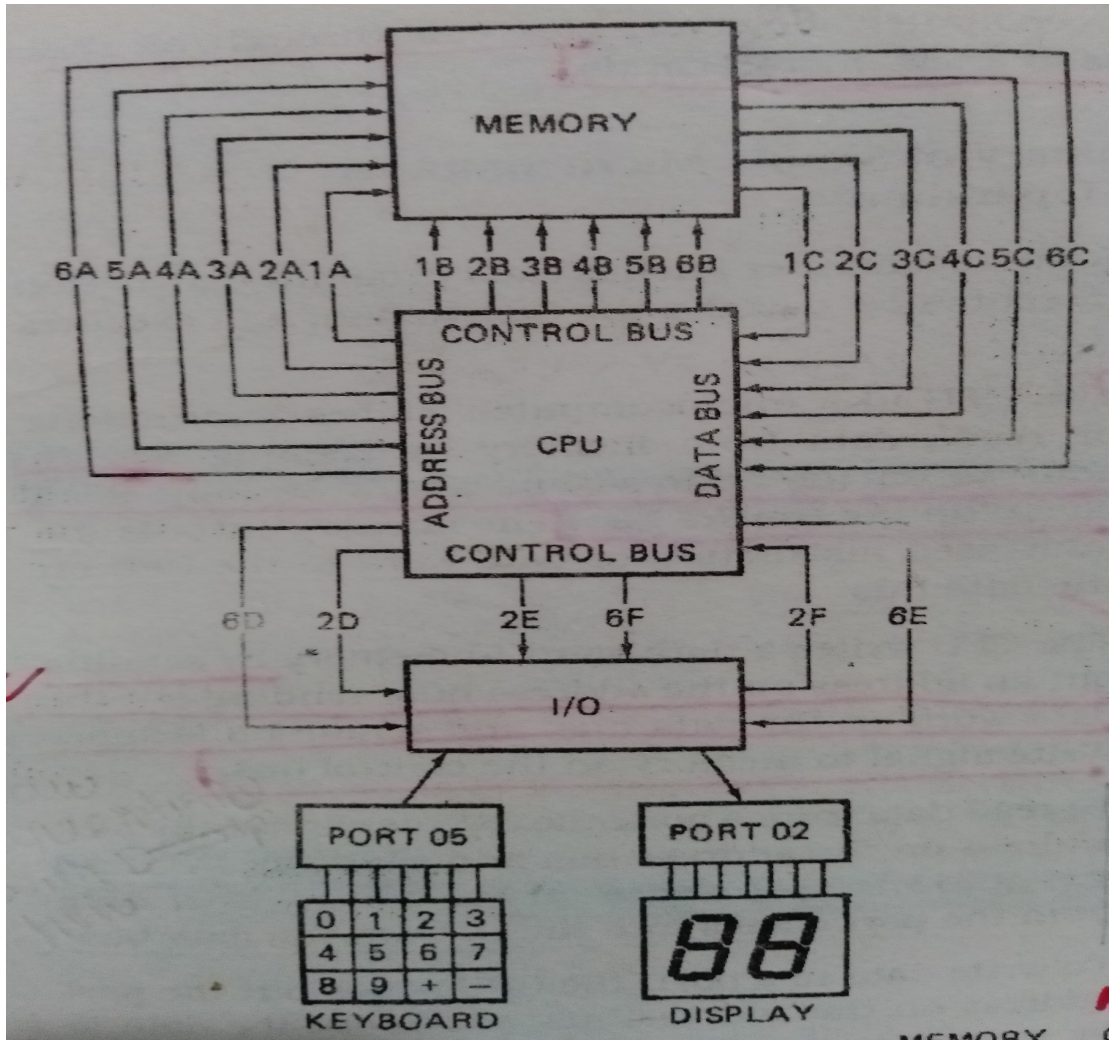
Central Processing Unit

- CPU fetches Binary coded instructions from memory usually in successive memory locations
- Decodes the instructions into series of simple actions
- Carries out actions in sequence of steps
- Instruction Pointer IP

Central Processing Unit

- Machine Instruction has 2 parts:
  - **Opcode**
  - **Operands**
- **To execute a machine instruction: Fetch -execute Cycle**
  - **Fetch:**
    - **Fetch an instruction from memory**
    - **Decode it to determine the operation**
    - **Fetch data from memory if necessary**
  - **Execute:**
    - **Perform the operation on data**
    - **Store the result in memory if needed**

Overview of Microcomputer structure and Operation:
Execution of a 3 Instruction Program



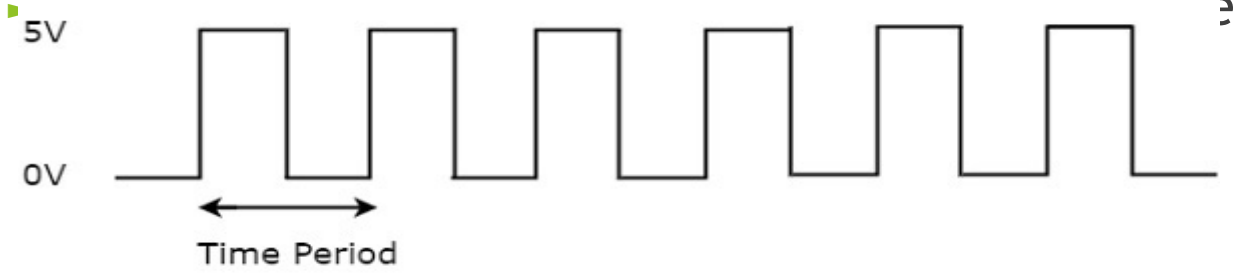
- Program:

1. Input a value from port 05
2. Add 7 to this value
3. Output tis result to port 02

Overview of Microcomputer structure and Operation: Timing: clock speed

- A clock circuit controls the processor by generation train of clock pulse
  - Clock Period: Interval between 2 pulse
  - Clock Speed/ Clock Rate: # of Pulse per second (MHz)
- All computer circuits are activated by clock pulse
- Each step in Fetch – Execution cycle require one or more Clock Periods
- Intel 8086 takes :
  - 4 clock periods -> read operation

eration



Microprocessor Evolution and Types

- Microprocessor have evolved in 3 major direction:
  1. Embedded Controller
  2. Bit-Slice Processor
  3. General Purpose CPU

Microprocessor Evolution and Types

- Common way to **categorize microprocessor: Data width**
  - Number of bits that their Arithmetic Logic Unit (ALU) can work with at a time, regardless of the number of address line or data bus line.
- What determines the size of a microprocessor?
  - Generally the bit-**size** of a processor is the **size** of its general purpose registers.
  - This often corresponds to the size of the Data bus and possibly the address bus, but doesn't necessarily.

Microprocessor Evolution and Types

- 16 bit microprocessor means:
  - Its ALU, internal registers and most of the Instructions are designed to work with 16 bit binary word.
  - 16 bit can be processed or transmitted in parallel.
  - A 16-bit microprocessor can process data and memory addresses that are represented by 16 bits.

Microprocessor Evolution and Types

NAME	YEAR	TRANSISTORS	DATA WIDTH	CLOCK SPEED
8080	1974	6,000	8 bits	2 MHz
8085	1976	6,500	8 bits	5 MHz
8086	1978	29,000	16 bits	5 MHz
8088	1979	29,000	8 bits	5 MHz
80286	1982	134,000	16 bits	6 MHz
80386	1985	275,000	32 bits	16 MHz
80486	1989	1,200,000	32 bits	25 MHz
PENTIUM	1993	3,100,000	32/64 bits	60 MHz
PENTIUM II	1997	7,500,000	64 bits	233 MHz
PENTIUM III	1999	9,500,000	64 bits	450 MHz
PENTIUM IV	2000	42,000,000	64 bits	1.5 GHz

Microprocessor Evolution and Types

Evolution of Intel Processor

4-bit	8-bit	16-bit	32-bit	64-bit
4004	8008	8086	80386	Dual Core
4040	8080	8088	80486	Core 2
	8085	80186	Pentium/80586	Core i7
		80188	PII	Core i5
		80286	PIII	Core i3
			PIV	
			Dual Core	

Microprocessor Evolution and Types

Processor	Year Of Introduction	No. Of Transistors	Intial Clock Speed	Address Bus	Data Bus(in bit)	Addressable Memory
4004	1971	2300	108 kHz	10 bit	4	640 bytes
8008	1972	3500	200 kHz	14 bit	8	16 k
8080	1974	6000	2 MHz	16 bit	8	64 k
8085	1976	6500	5 MHz	16 bit	8	64 k
8086	1978	29000	5 MHz	20 bit	16	1 M
8088	1979	29000	5 MHz	20 bit	8	1 M
80286	1982	134000	8 MHz	24 bit	16	16 M
80386	1985	275000	16 MHz	32 bit	32	4 G
80486	1989	1.2 M	25 MHz	32 bit	32	4 G
Pentium	1993	3.1 M	60 MHz	32 bit	32/64	4 G
Pentium Pro	1995	5.5 M	150 MHz	36 bit	32/64	64 G
Pentium II	1997	8.8 M	233 MHz	36 bit	64	64 G
Pentium III	1999	9.5 M	650 MHz	36 bit	64	64 G
Pentium 4	2000	42 M	1.4 GHz	36 bit	64	64 G

Microprocessor Evolution and Types:
Intel 8086 Microprocessor

Clock Speed	Data Width	Data Bus	Address Bus	Addressable Memory
5 MHz	16 bit	16 bit	20 bit	2^20 =1 Mega byte

- Intel 8086 microprocessor is a 16 bit microprocessor
- Can process data and memory addresses that are represented by 16 bits at a time.

Microprocessor Evolution and Types:
Intel 8088 Microprocessor

Clock Speed	Data Width	Data Bus	Address Bus	Addressable Memory
5 MHz	16 bit	8 bit	20 bit	2^20 =1 Mega byte

- Intel 8088 microprocessor is also 16 bit microprocessor but 8 bit data bus.
- 8086 and 8088 have same Instruction Set and forms basic set of instruction for other microprocessor in the family.
- Less expensive

Microprocessor Evolution and Types:
Intel 80286 Microprocessor

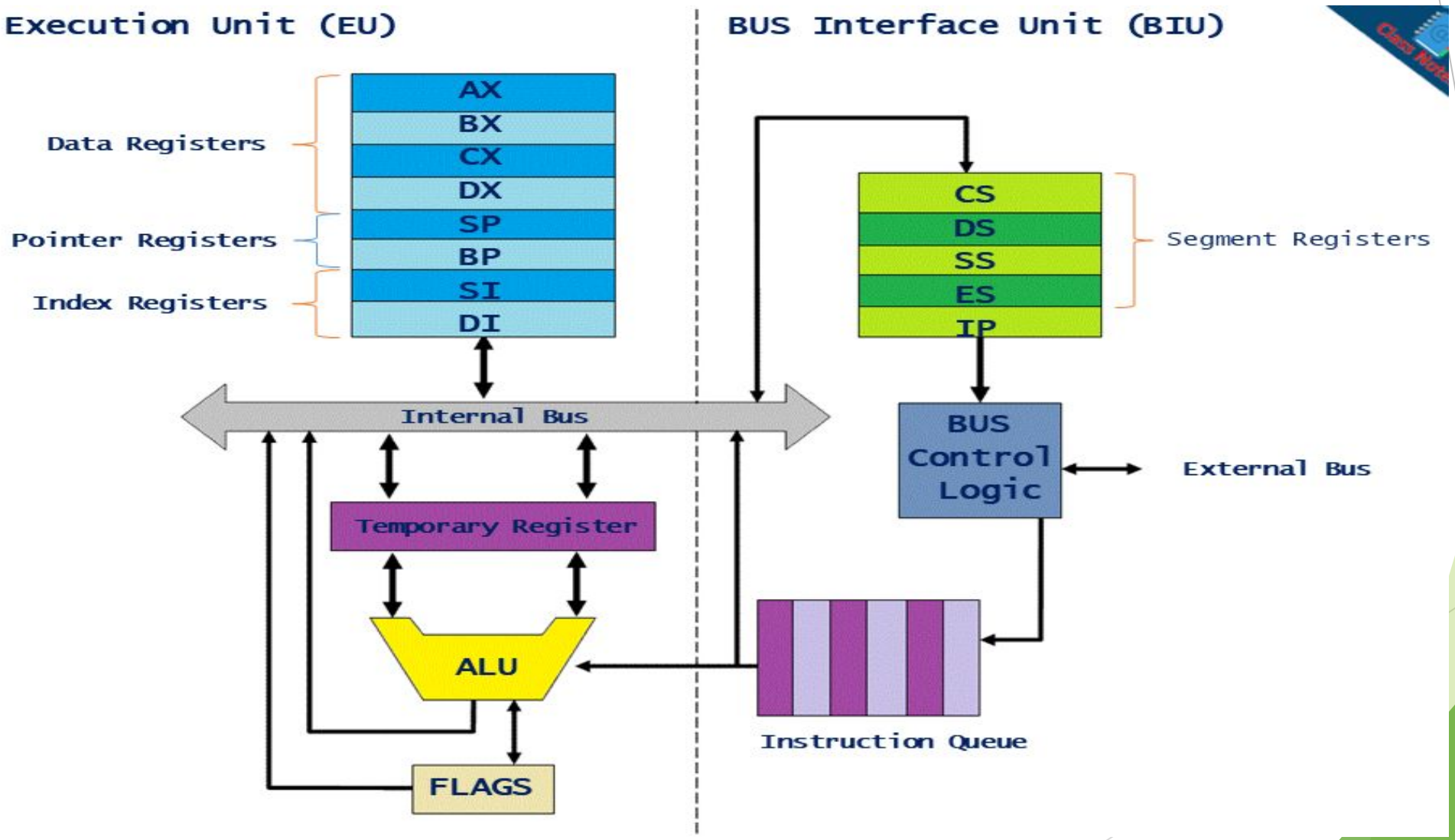
Clock Speed	Data Width	Data Bus	Address Bus	Addressable Memory
8 MHz	16 bit	16 bit	24 bit	2^24 =16 Mega byte

- Intel 80286 microprocessor is a also 16 bit microprocessor but faster than 8086
- 2 modes of operation:
  1. Real Address Mode
  2. Protected Virtual Mode/ Protected Mode:
     - Supports Multi-tasking and Memory Protection
- More Addressable Memory
- Virtual Memory in Protected Mode:
  - Can treat external storage as physical memory

Intel 8086 Internal Architecture

- 2 independent functional unit
- Bus Interface Unit (BIU):
  - Fetch Instruction from memory
  - Read data from port and memory
  - Write data to port and memory
- Execution Unit (EU):
  - Tells the BIU where to fetch
  - Decodes instruction
  - Executes instruction

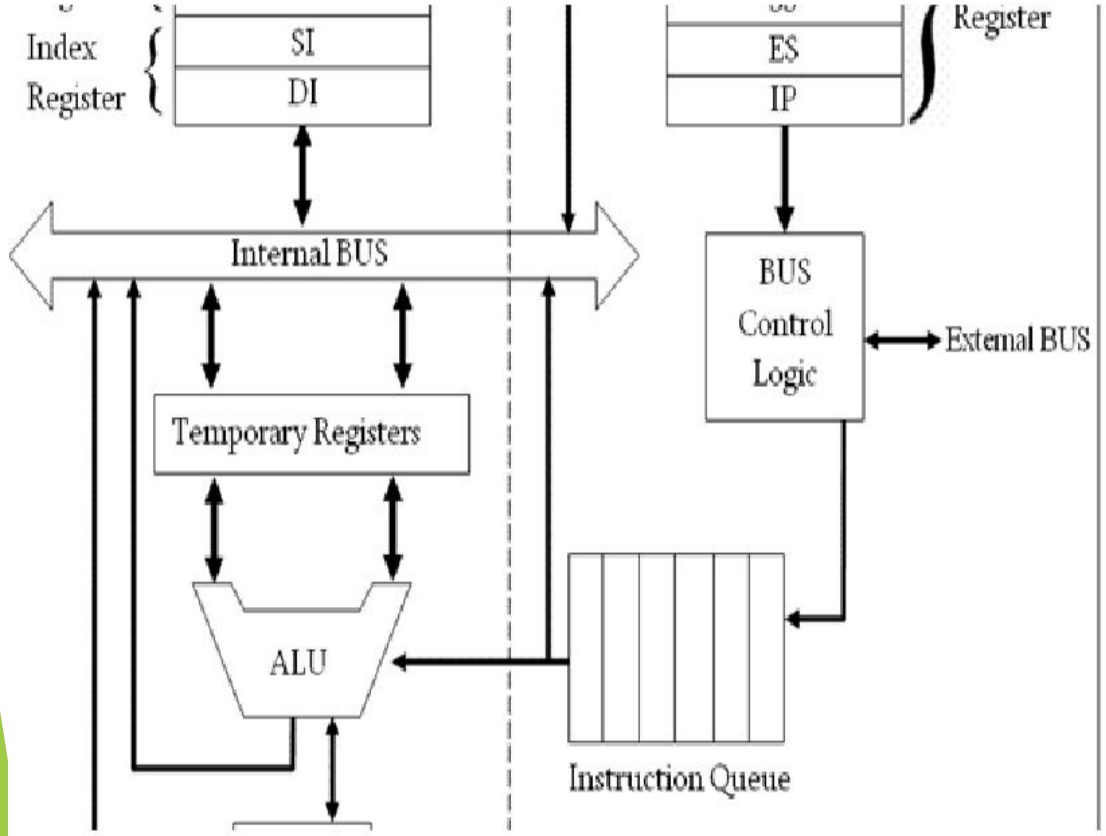
Intel 8086 Internal Architecture



Intel 8086 Internal Architecture:
Bus Interface Unit (BIU):

- The BIU facilitates communication between the EU and the memory or I/O circuits.
- The BIU is responsible for transmitting addresses, data, and control signals on the buses.
- The **instruction pointer (IP)** contains the address of the next instruction to be executed by the EU.

Intel 8086 Internal Architecture:
Bus Interface Unit (BIU):Instruction Prefetch



- While the EU is executing an instruction, the BIU fetches up to six bytes of the next instruction and places them in the instruction queue.

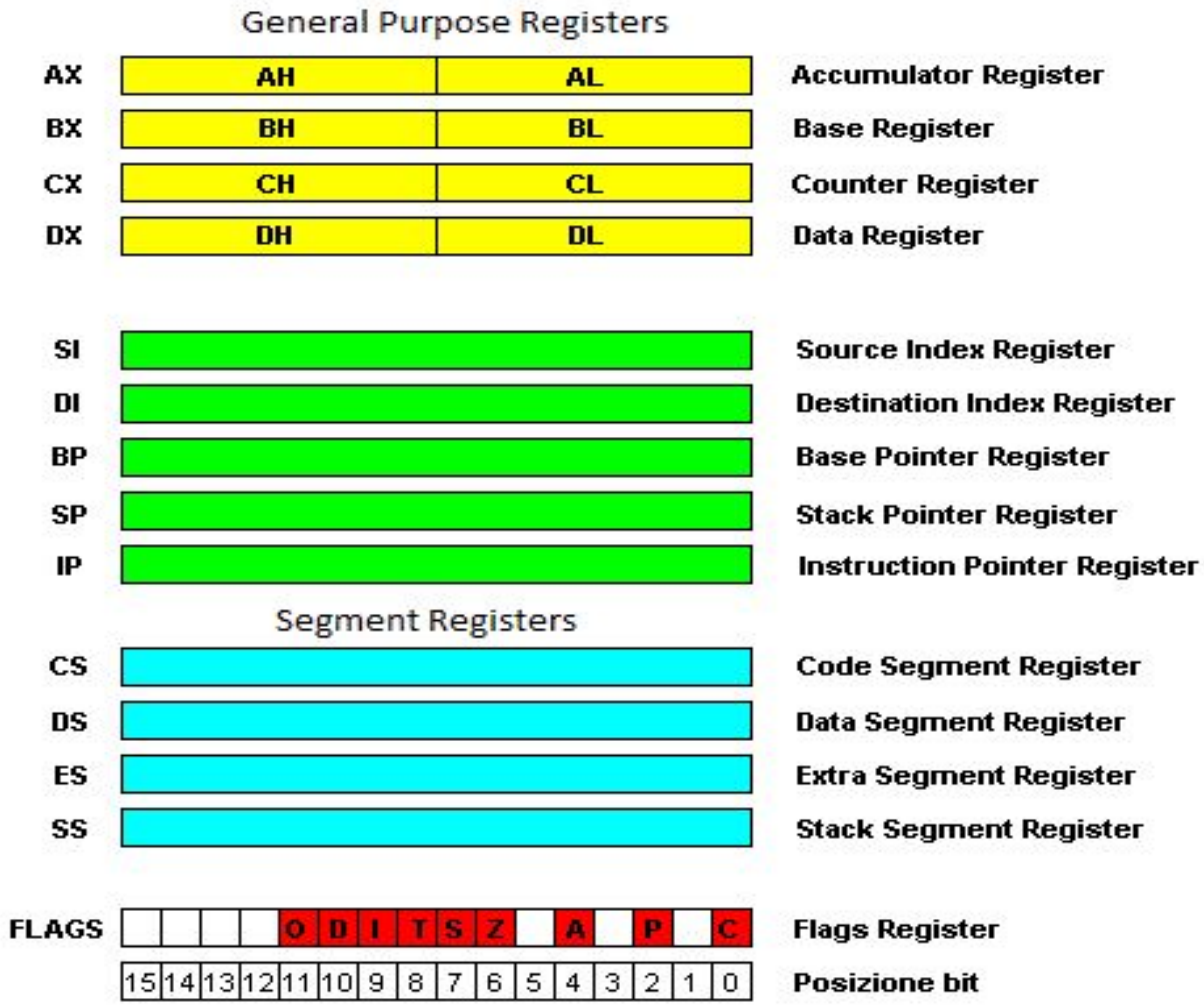
Intel 8086 Internal Architecture: Execution Unit

1. Control Circuitry
2. Instruction Decoder
3. ALU ():
   - Add, subtract, AND, OR XOR, increment, decrement, complement, shift binary numbers.
- The data for the operations are stored in circuits called registers.
- EU contains few registers for holding data and address

Intel 8086 Internal Architecture: Registers

- Information inside microprocessor is stored in register
- Total Fourteen registers: All of these registers are 16-bit long
- Classified based on their functions they perform:
  1. Data Registers: hold data for operation
     - Four (4) general data registers
  2. Address Registers: hold address of data or instruction
     - Segment Register
     - Pointer Register
     - Index Register
  3. A Status Register: keep current status of the processor :FLAGS register
- Temporary register: for holding operands

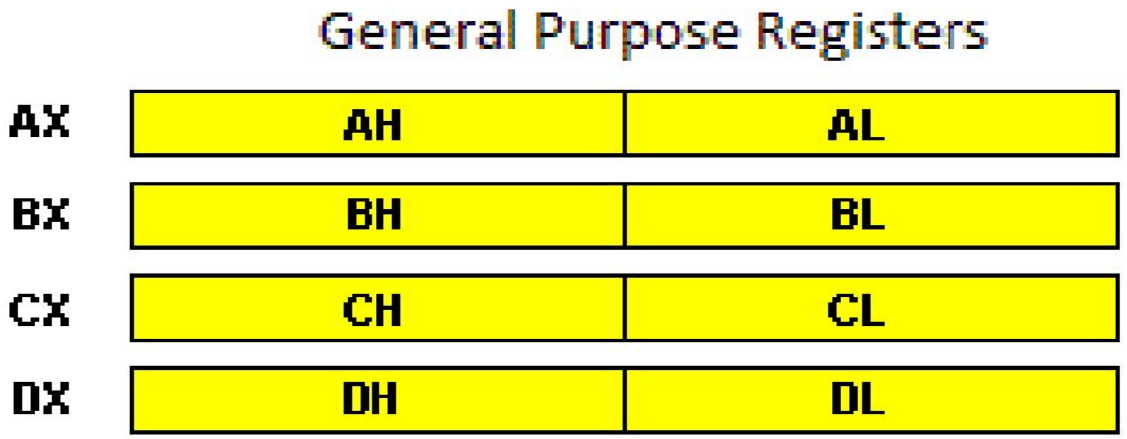
Intel 8086 Internal Architecture: Registers



Total Fourteen registers: All of these registers are 16-bit long

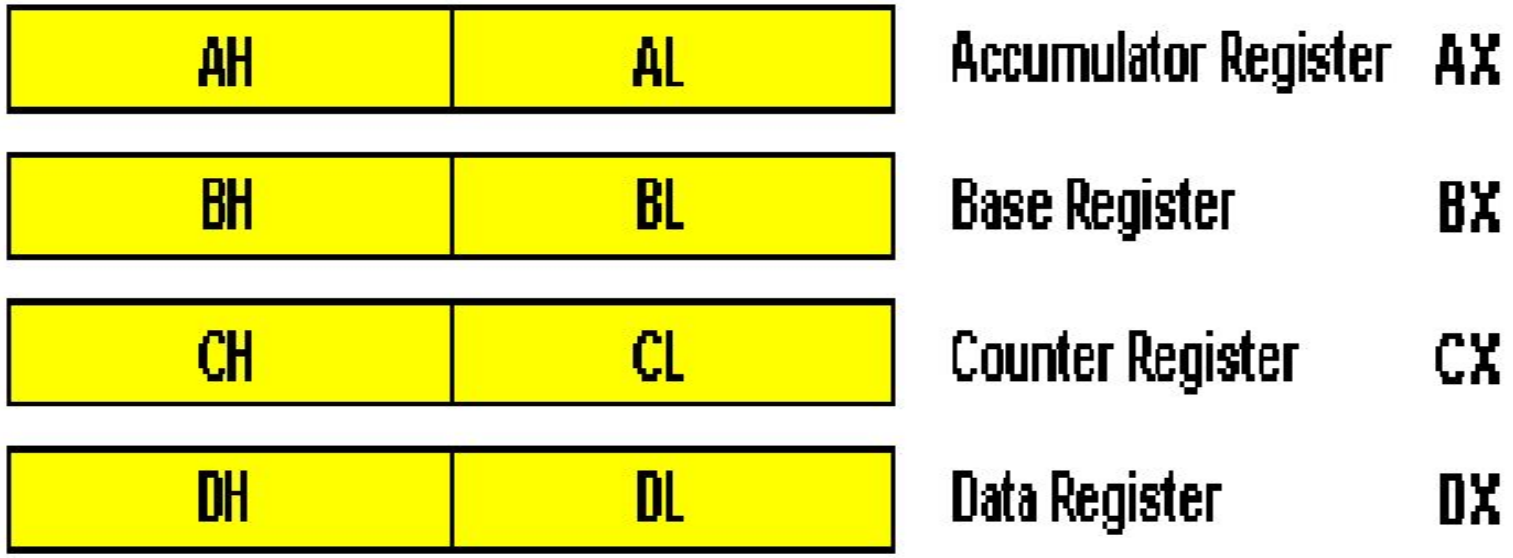
Intel 8086 Internal Architecture: Data Registers

- Hold data for general data manipulation
- **Why Register? Why not Memory?**
  - Even though the processor can operate on data stored in memory, the same instruction is faster if stored in registers, requires fewer clock cycles
- High and low bytes can be accessed separately



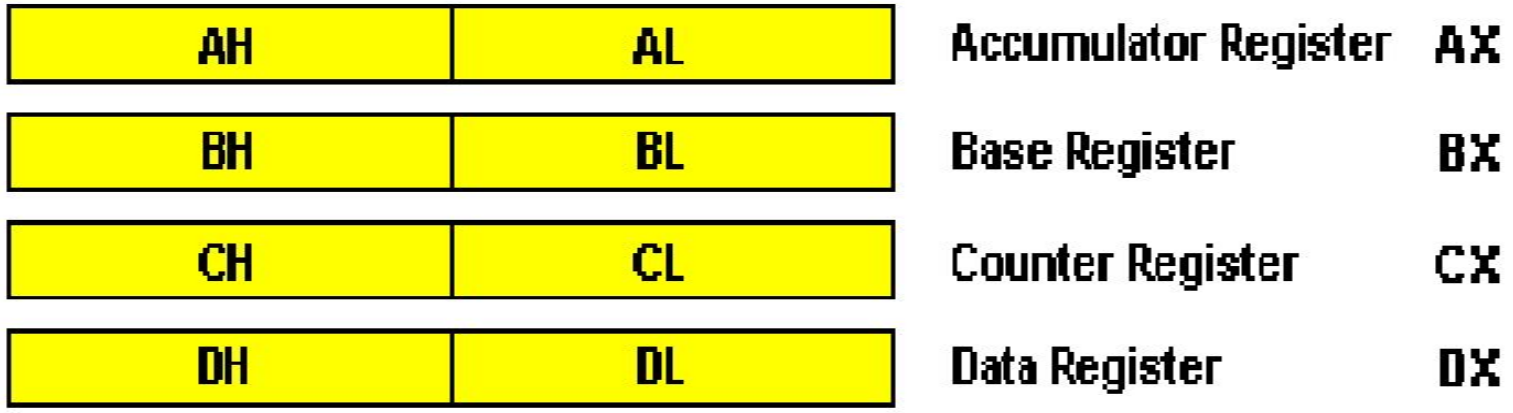
Intel 8086 Internal Architecture: Data Registers

- In addition to being general-purpose data registers, these 4 also performs following special functions.



Intel 8086 Internal Architecture: Data Registers

- AX (Accumulator):
  - preferred register to use arithmetic, logic and data transfer, since its use generate shortest machine code
  - In multiplication & division one of the number must be in AX or AL
  - Input & Output also requires to use of AX and AL



Intel 8086 Internal Architecture: Data Registers

- BX (Base Register):
  - Also serves as an address register
- CX (Count Register):
  - Program loop constructions
- DX (Data Register):
  - In multiplication & division
  - Also used in Input & Output

Intel 8086 Internal Architecture: Memory Segment

- The idea of memory segments is a direct consequence of using a 20-bit (physical) address in a 16-bit processor.
- Segmentation:

  Segmentation is the process in which the main **memory** of the computer is **divided into** different **segments**
- A **segment** is a logical unit of memory that may be up to 2^16=64 kilobytes long.

Intel 8086 Internal Architecture:
Segment: Why memory is divided into segments?

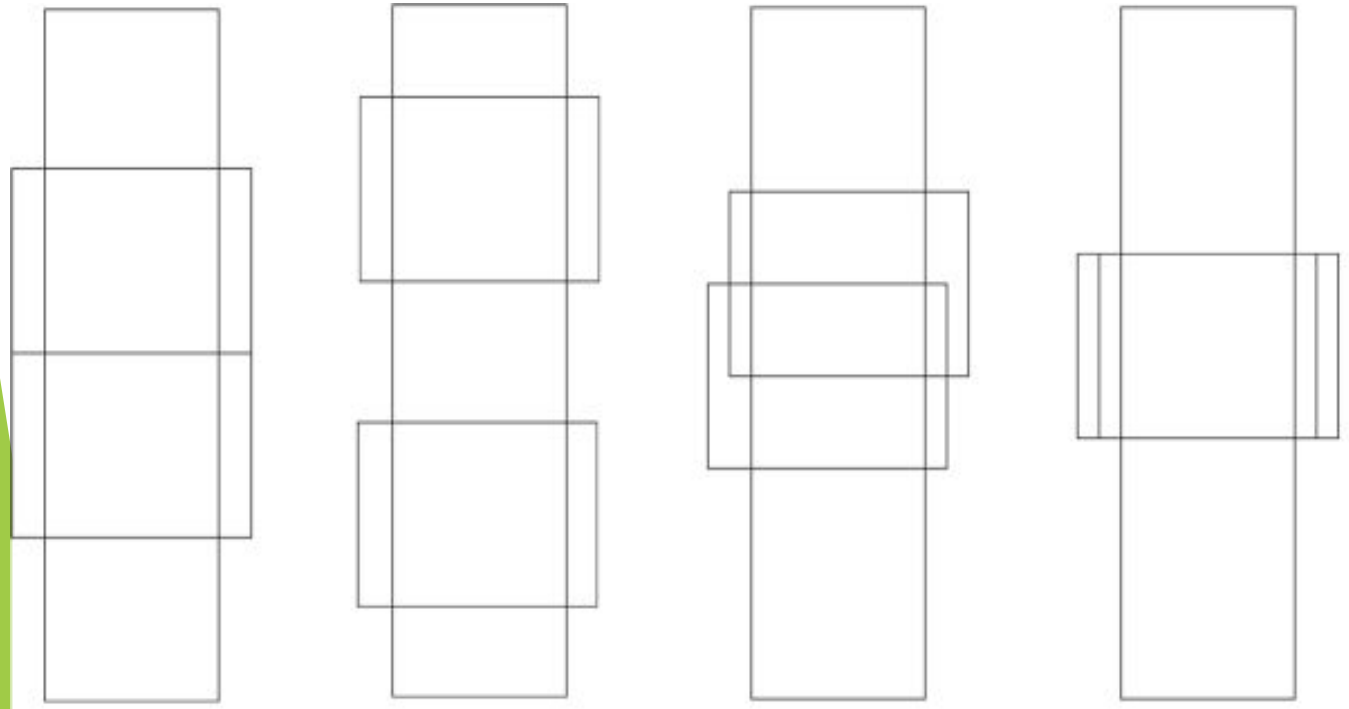
- 8086 BIU sends out 20 bit address, so it can address any of 2^20 = 1 Mega bytes in memory
- But, at any time 8086 can address any of 2^16= 64 Kbyte memory.
- Hence, memory is divided into (4)four 2^16= 64 Kbyte segments.
- At any given time 8086 can work with only this 4 (four) 64 Kbyte segments within 1Mbyte memory.
- Four segment registers hold the upper 16 bit of starting address of this four memory segments that 8086 is working at a particular time.

Intel 8086 Internal Architecture: Segment

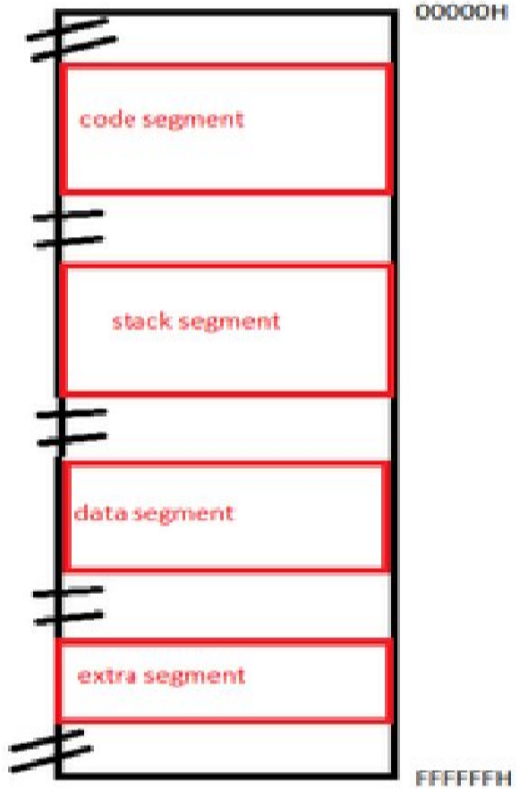
- 4 Segments:
  - Data Segment
  - Code Segment
  - Stack Segment
  - Extra Segment
- Each **segment** is made up of contiguous memory locations.
- Each **segment** has its own segment base address with zeros in the lowest 4 bits.
- A segment number is 16 bits, so the highest segment number is FFFFh.

Intel 8086 Internal Architecture: Segment

- Memory segments can be separated as shown



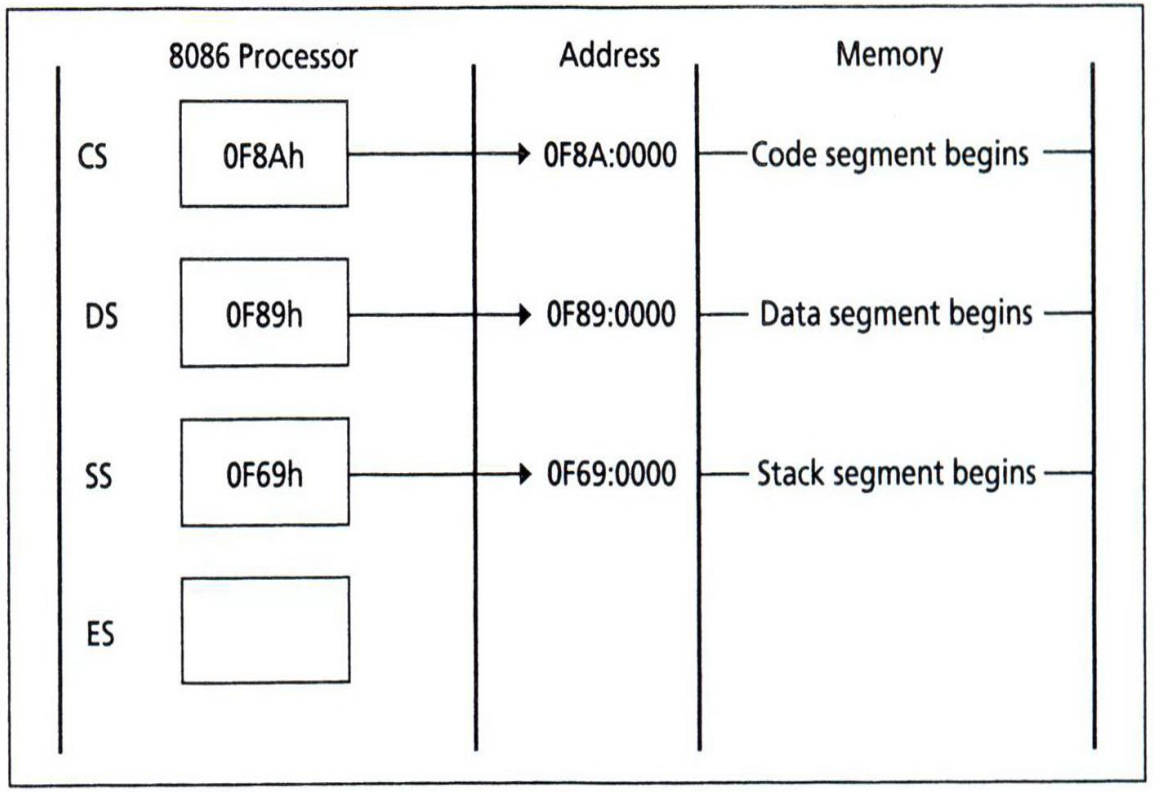
(a) Adjacent (b) Disjoint (c) Partially overlapped (d) Fully overlapped



1 MB memory of 8086

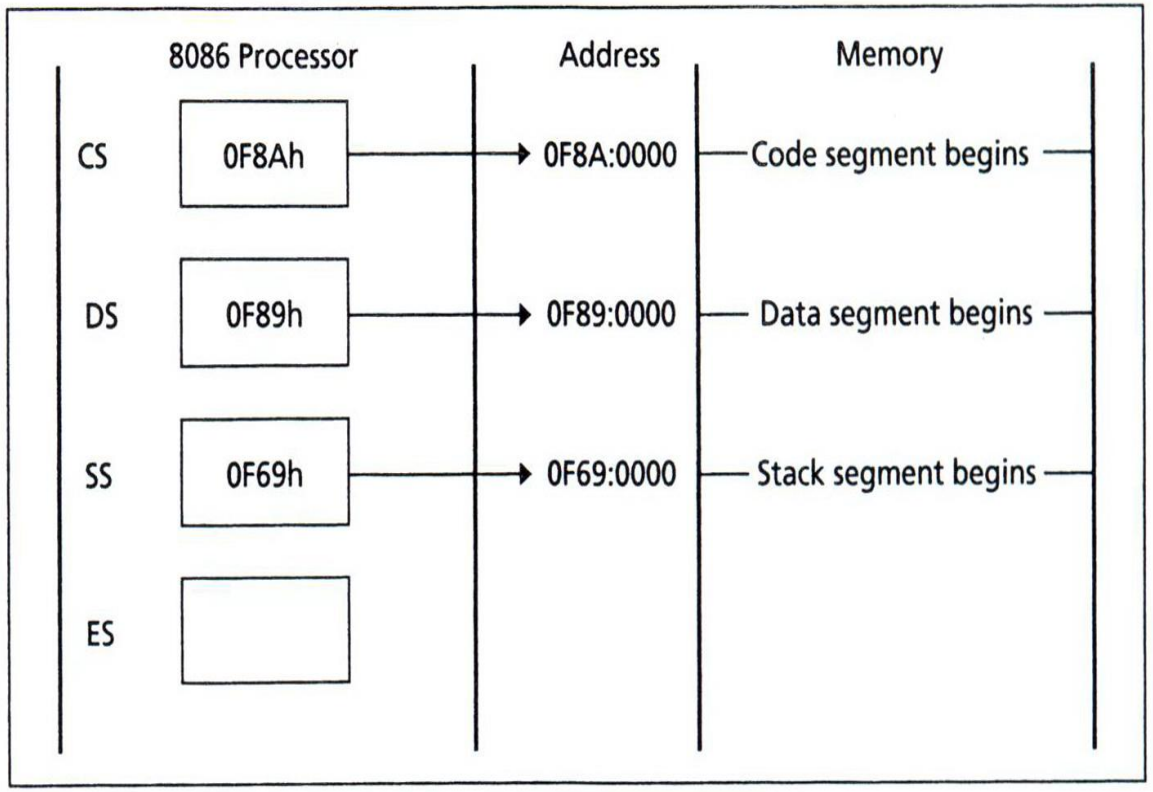
Intel 8086 Internal Architecture: Segments

- 4 segments registers in BIU holds the Upper 16 bits of starting address of four memory segments that the 8086 is currently working with.
- BIU always insert zeros for lowest 4 bit(nibble) of the 20 bit starting address for a segment
- 64 Kbyte segment can be located anywhere within the 1Mbyte address space but always start at an address with zeros in the lowest 4 bits



Intel 8086 Internal Architecture: Segments

- Segments start every 10h= 16 bytes
- And Stating address of any segment always ends with a hex digit 0.
- We call 16 bytes a paragraph
- An address that is divisible by 16 a paragraph boundary.

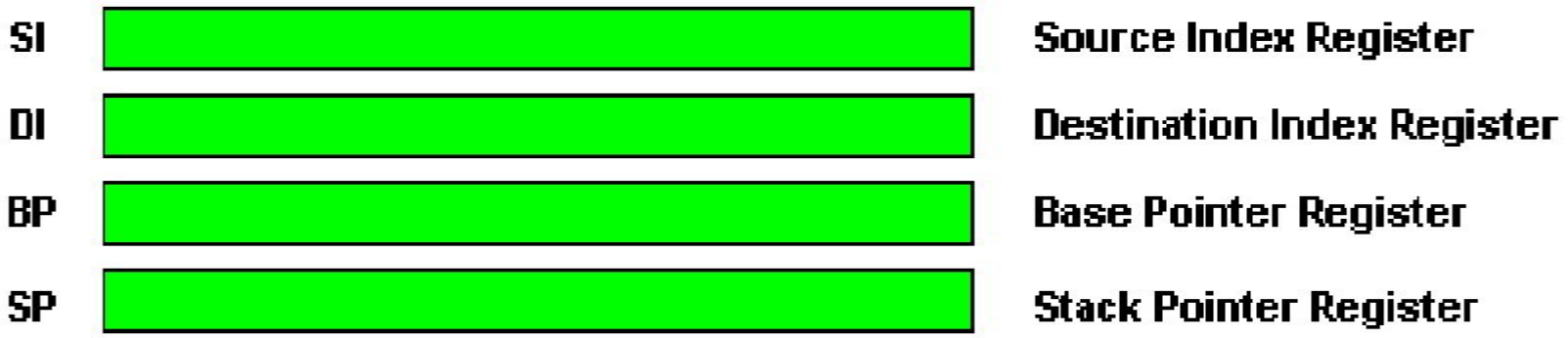


Intel 8086 Internal Architecture: Segment Registers

Memory segment	Segment register	Offset register
Code segment	Code segment Register (CSR)	Instruction Pointer (IP)
Data segment	Data segment Register (DSR)	Source index (SI)/ Destination index (DI)
Stack segment	Stack segment Register (SSR)	Stack Pointer (SP)/ Base Pointer (BP)
Extra segment	Extra segment Register (ESR)	Destination Index(DI)

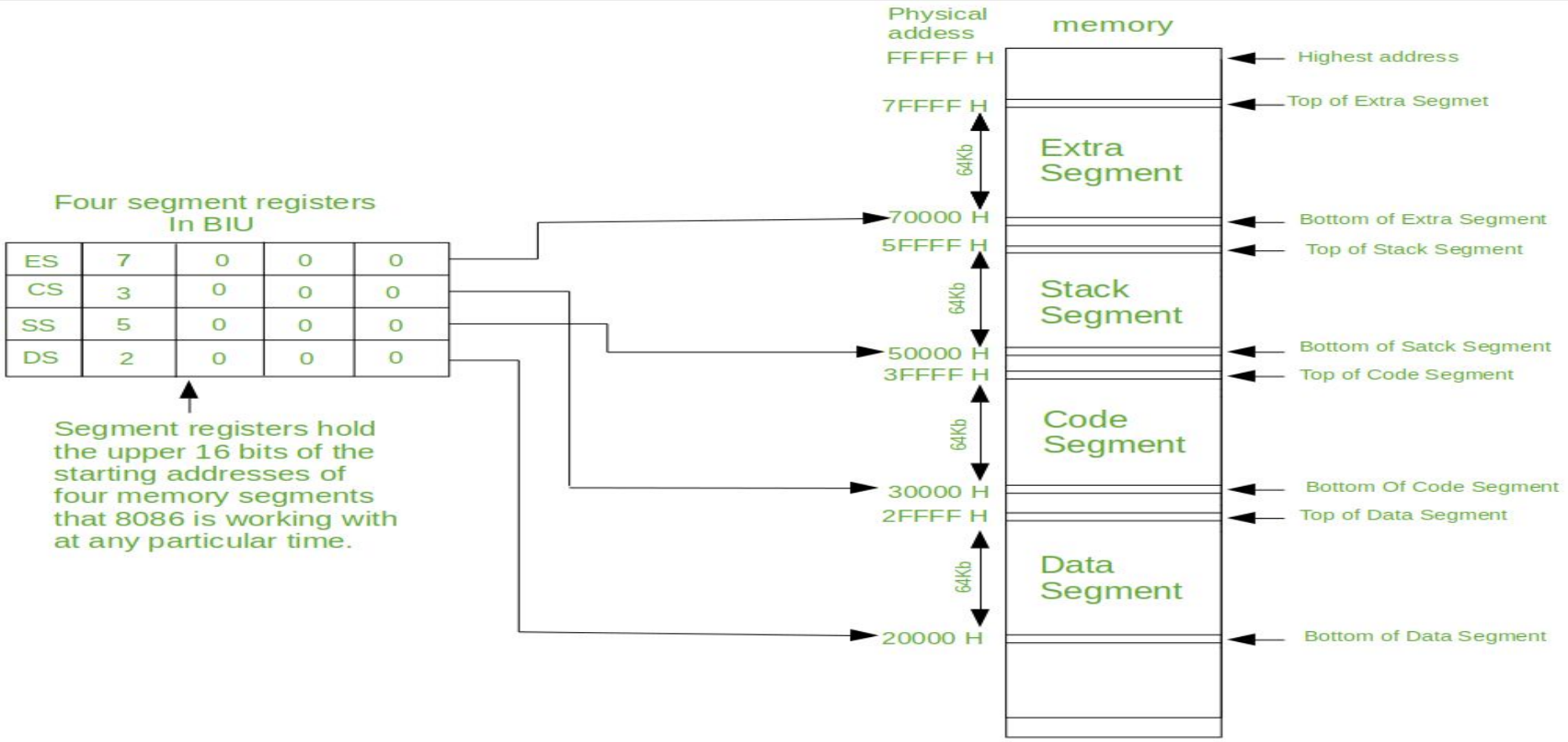
Intel 8086 Internal Architecture: Pointer and Index Registers

- Pointer Register->points to memory in Stack Segment
- Index Register -> points to memory in Data Segment
- Unlike segment registers, pointer and index registers can be used in arithmetic and other operations.

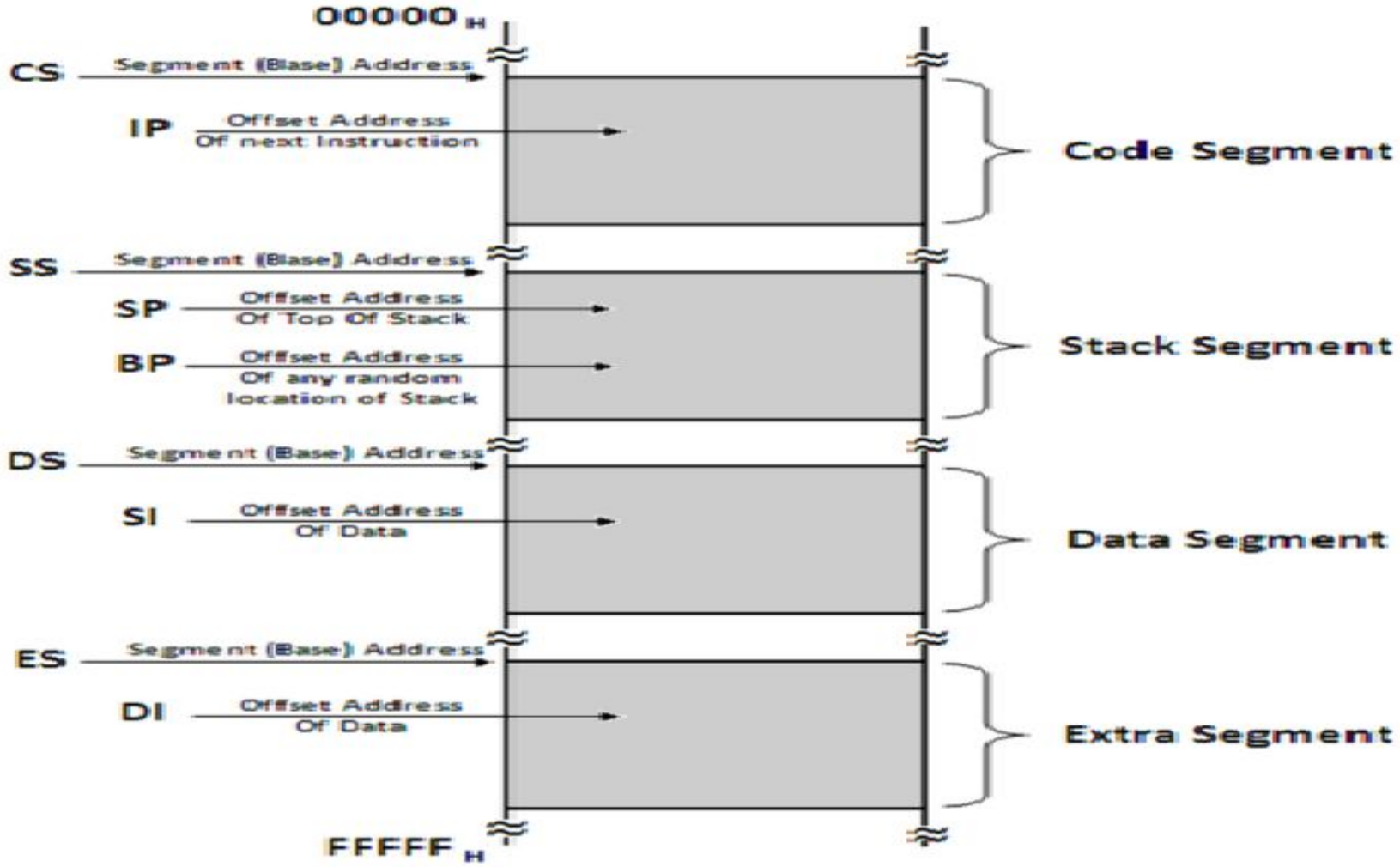


Intel 8086 Internal Architecture: Segment Registers

Segment registers store addresses of instructions and data in memory.



Intel 8086 Internal Architecture: Segment



Intel 8086 Internal Architecture: Physical Address

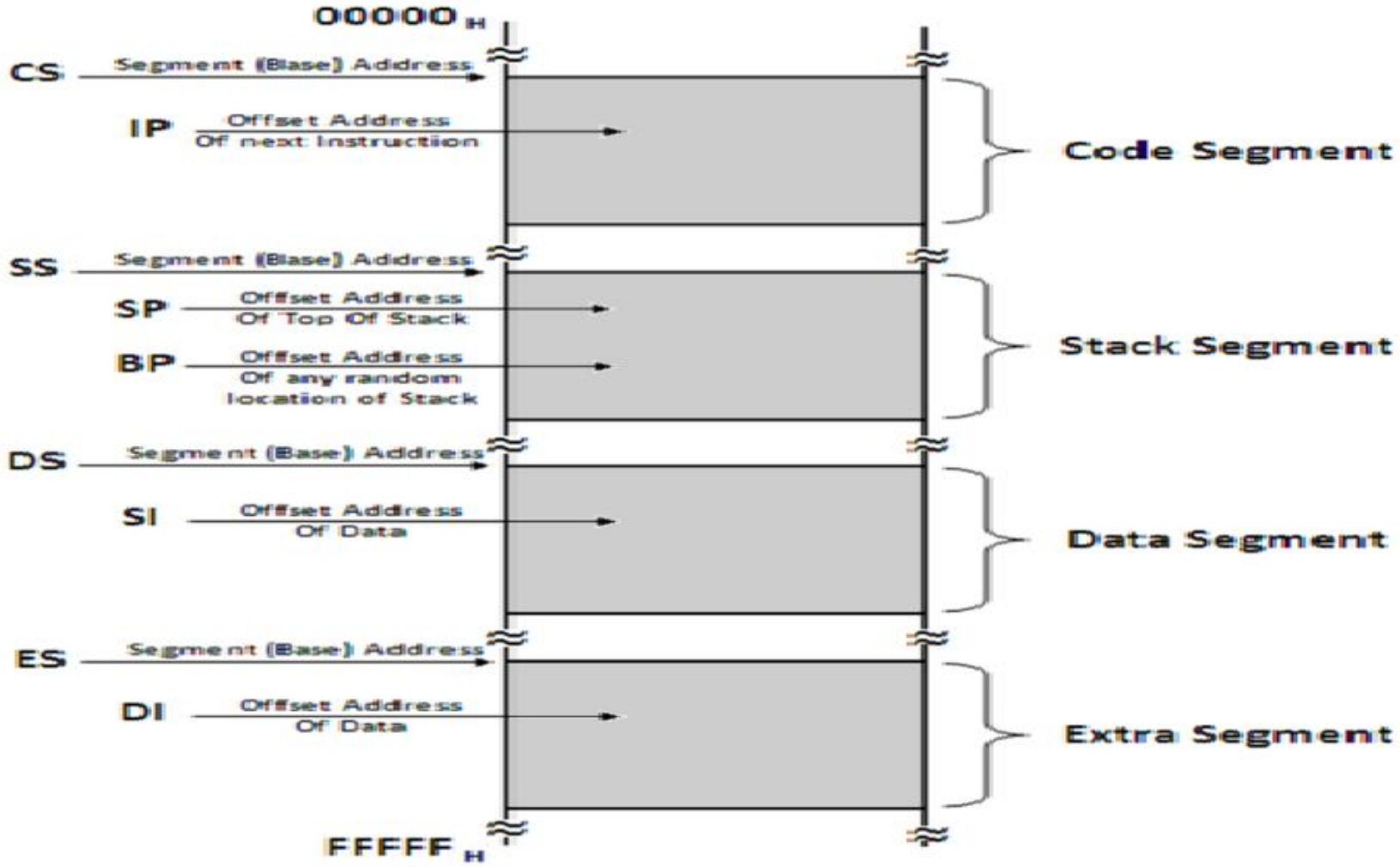
- 16 bit 8086 assigns a 20-bit physical address to its memory.
- It is possible to address 2^{20} = 1,048,576 bytes (one megabyte) of memory.

0000 0000 0000 0000 0000	00000h
0000 0000 0000 0000 0001	00001h
0000 0000 0000 0000 0010	00002h
0000 0000 0000 0000 0011	00003h
.....	
1111 1111 1111 1111 1110	FFFFEh
1111 1111 1111 1111 1111	FFFFFh

Intel 8086 Internal Architecture: Offset

- Within a segment, a memory location is specified by giving an offset.
- An offset is the number of bytes from the beginning of the segment to that particular memory location.
- With a 2^16=64 Kilobyte segment, the offset can be given as a 16-bit number.
- The first byte in a segment has offset 0.
- The last offset in a segment is FFFFh.

Intel 8086 Internal Architecture: Segment

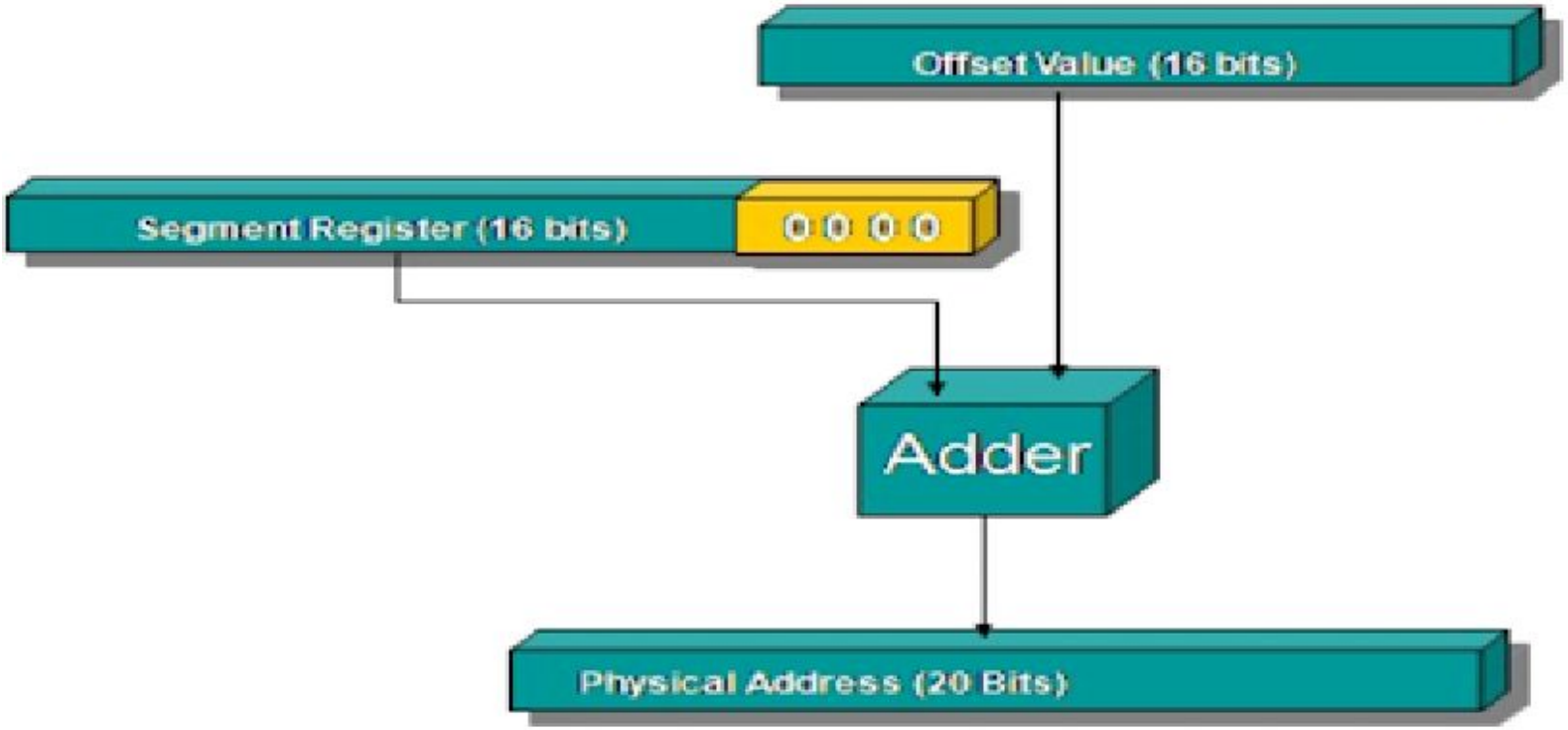


Intel 8086 Internal Architecture:
How 16 bit microprocessor works with 20 bit memory address?

- Segment : Offset also known as Logical Address
- To obtain a 20-bit physical address:
  - First shifts the segment address 4 bits to the left (this is equivalent to multiplying by 10h),
  - and then adds the offset.
- **Physical Address = Segment × 10h + Offset**
- A4FB:4872 = A4FB0h + 4872h = A9822h

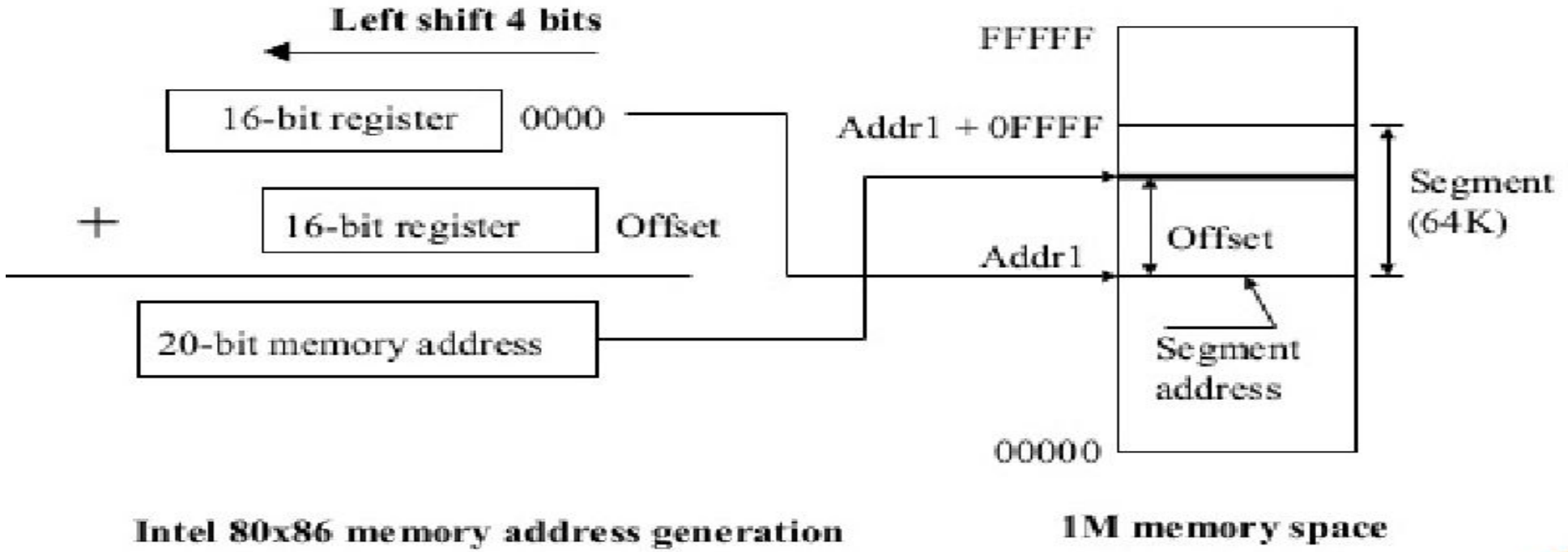
Intel 8086 Internal Architecture:

How 16 bit microprocessor works with 20 bit memory address?



Intel 8086 Internal Architecture:

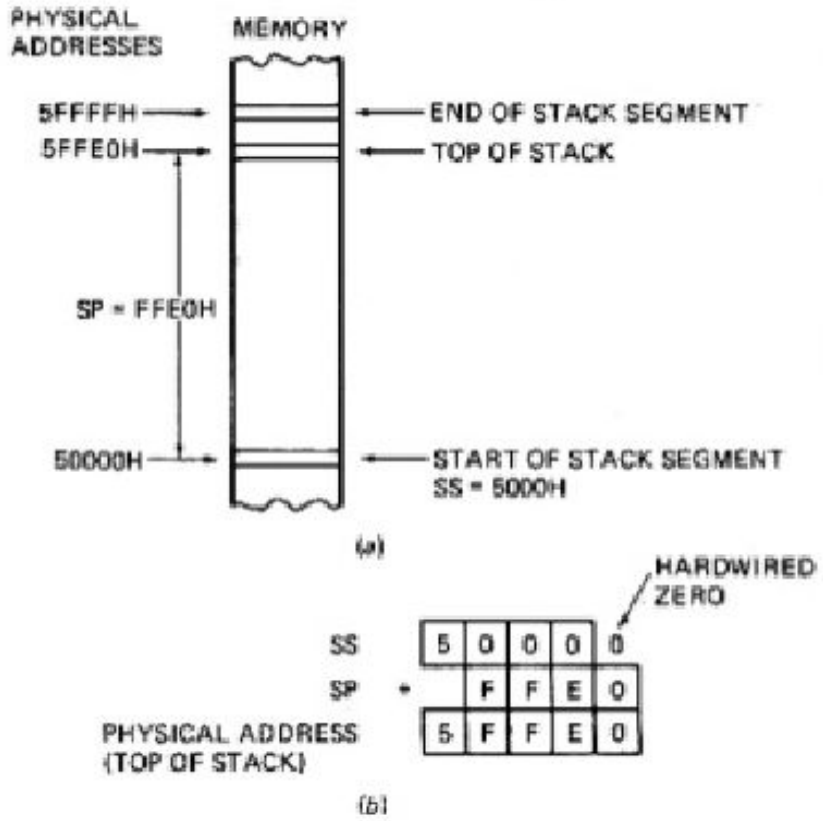
How 16 bit microprocessor works with 20 bit memory address?



Intel 80x86 memory address generation **1M memory space**

Intel 8086 Internal Architecture:

How 16 bit microprocessor works with 20 bit memory address?



- E.g. Let **SS** hold **5000h**, and **SP** hold **FFE0h**
- Now the actual address in the physical memory space is given by SS:SP and calculated as:
- SS is first shifted left four times

SS<<4 = 50000h

Then the offset in SP is added

50000h

+ FFE0h

Top of stack **5FFE0h**

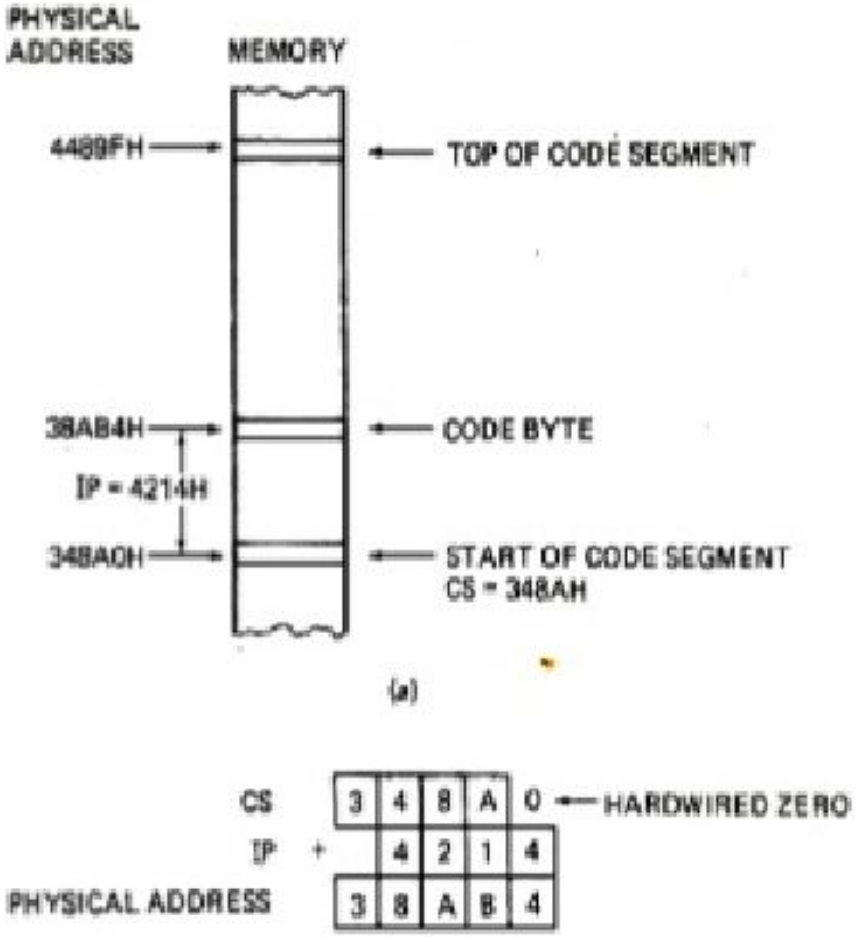
Intel 8086 Internal Architecture: IP (Instruction Pointer)

- To access instructions, 8086 uses CS(Code Segment) and IP (instruction pointer).
- CS contains the segment number or segment base address of the next instruction,
- IP contains the distance or offset from base address to the next instruction byte to be fetched.
- IP is updated each time an instruction is executed so that it will point to the next instruction.
- Unlike other registers, IP cannot be directly manipulated by an instruction; that is, an instruction may not contain IP as its operand.

Intel 8086 Internal Architecture: IP (Instruction Pointer)

- To access instructions, 8086 uses CS(Code Segment) and IP.
- CS contains the segment number of the next instruction, and IP contains the offset.
- IP contains the distance or offset from base address to the next instruction byte to be fetched.

Intel 8086 Internal Architecture: IP (Instruction Pointer)



- E.g. Let CS holds **348Ah**, and IP holds **4214h.** Now the actual address in the physical memory space is given by CS:IP and calculated as:
- CS is first shifted left four times

CS <<4= 348A0h

Then the offset in IP is added

348A0h

+ 4214h

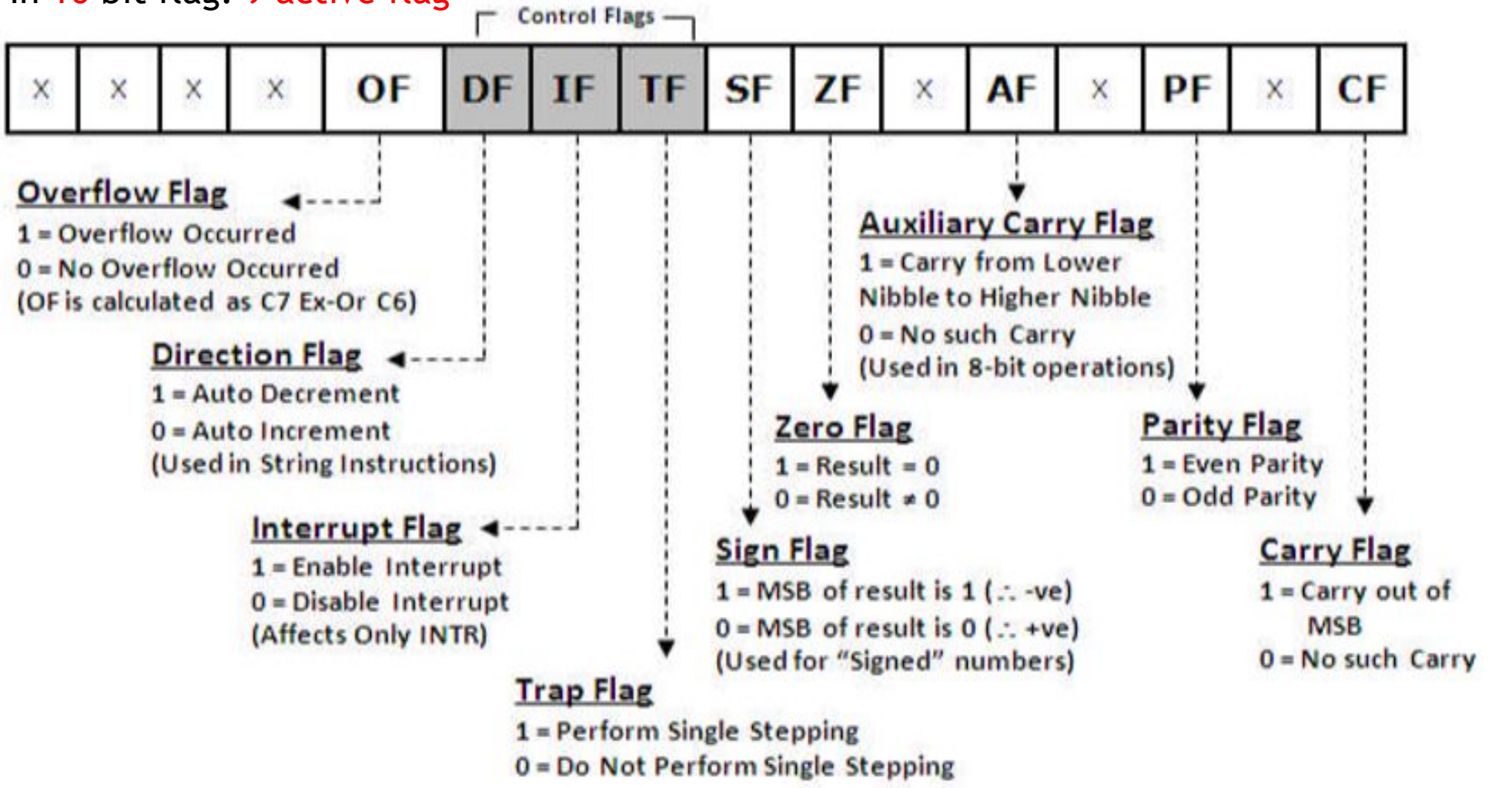
Actual address **38AB4h**

Intel 8086 Internal Architecture:
Flag Register

- A 16 bit register
- The purpose of FLAGS is to indicate the status of microprocessor by setting of individual bits called flags.
- 2 kinds of flags:
- Status flags:
  - When a subtraction operation results in a 0, ZF (zero flag) is set to 1 (true).
- Control flags: To enable or disable certain operation of microprocessor
  - Inputs from keyboard is ignored by processor if Interrupt flag is set zero.

Intel 8086 Internal Architecture:
Flag Register

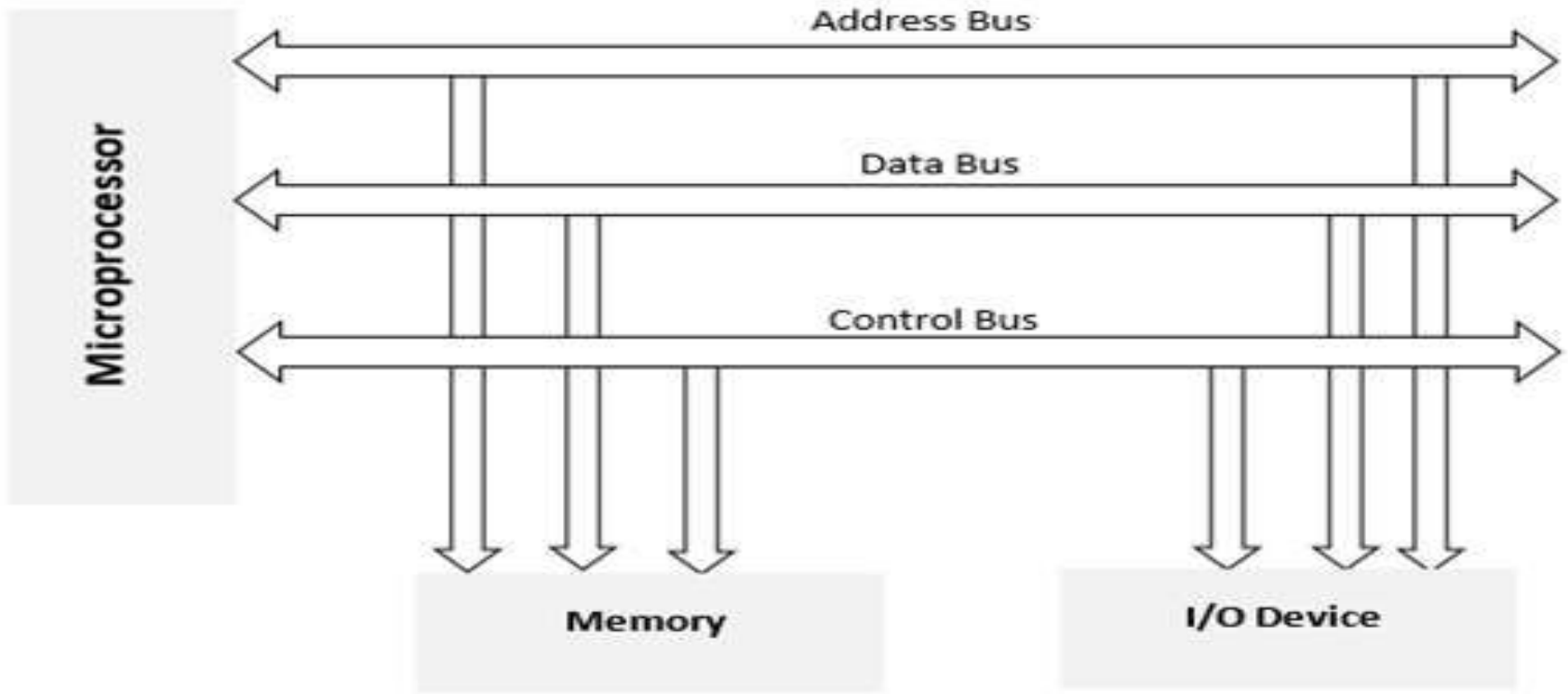
In 16 bit flag: 9 active flag



Interface

- **Interface** is the path for communication between two components.
- Interfacing is of two types,
  - memory interfacing
  - I/O interfacing.

Block Diagram of Memory and I/O Interfacing



Reference Book:

- Microprocessor & Interfacing, Author. V. Hall Chapter-2
- Assembly Language Programming and Organization of the IBM PC, Author: Ythe Yu, Charles Marut Chapter-1 and Chapter-3
- Memory Organization and Addressing - Prepared BY Shahadat Hussain Parvez