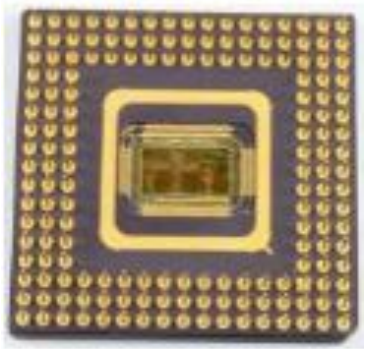
Microprocessors

Microcomputers

There are three main components of a Computer System.

1. **Central Processing Unit (CPU):** Also simply called as the microprocessor acts as the brain coordinating all activities within a computer.
2. **Memory**: The program instructions and data are primarily stored.
3. **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.



Microcomputers

Communicating between these components;

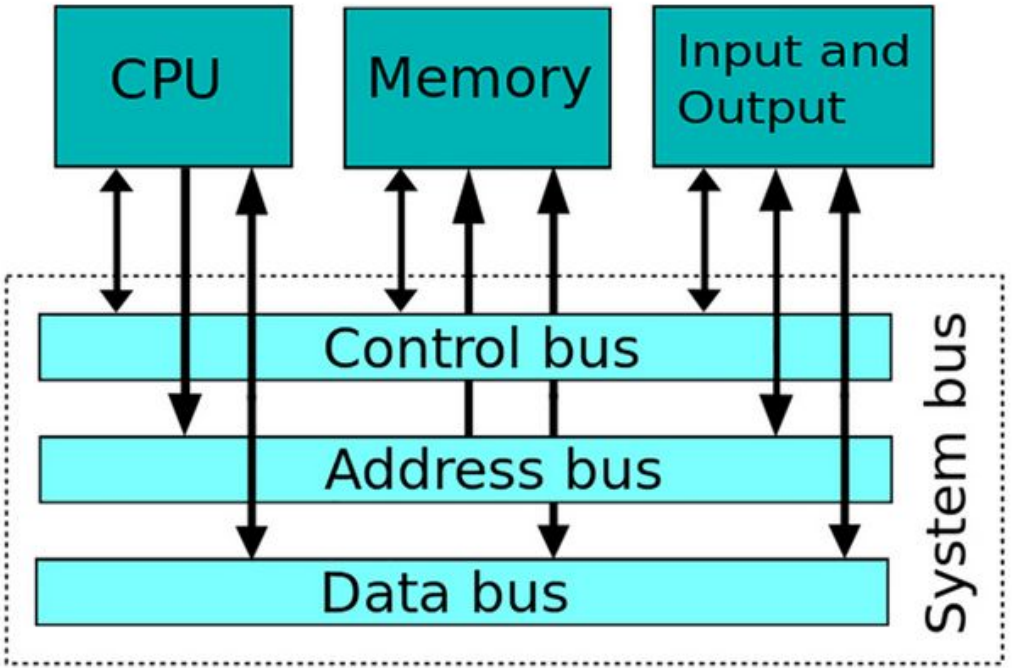
The CPU is connected to memory and I/O devices through a strip of wires called a **bus**. The bus inside a computer carries information from place to place.

There are three types of busses:

1. Address Bus: is used to identify the memory location or I/O device the processor intends to communicate with. The width of the Address Bus of 8086 microprocessor ranges from **20 bits**

2. Data Bus: is used by the CPU to get data from / to send data to the memory or the I/O devices. The size of data bus of 8086 microprocessor is **16 bits**.

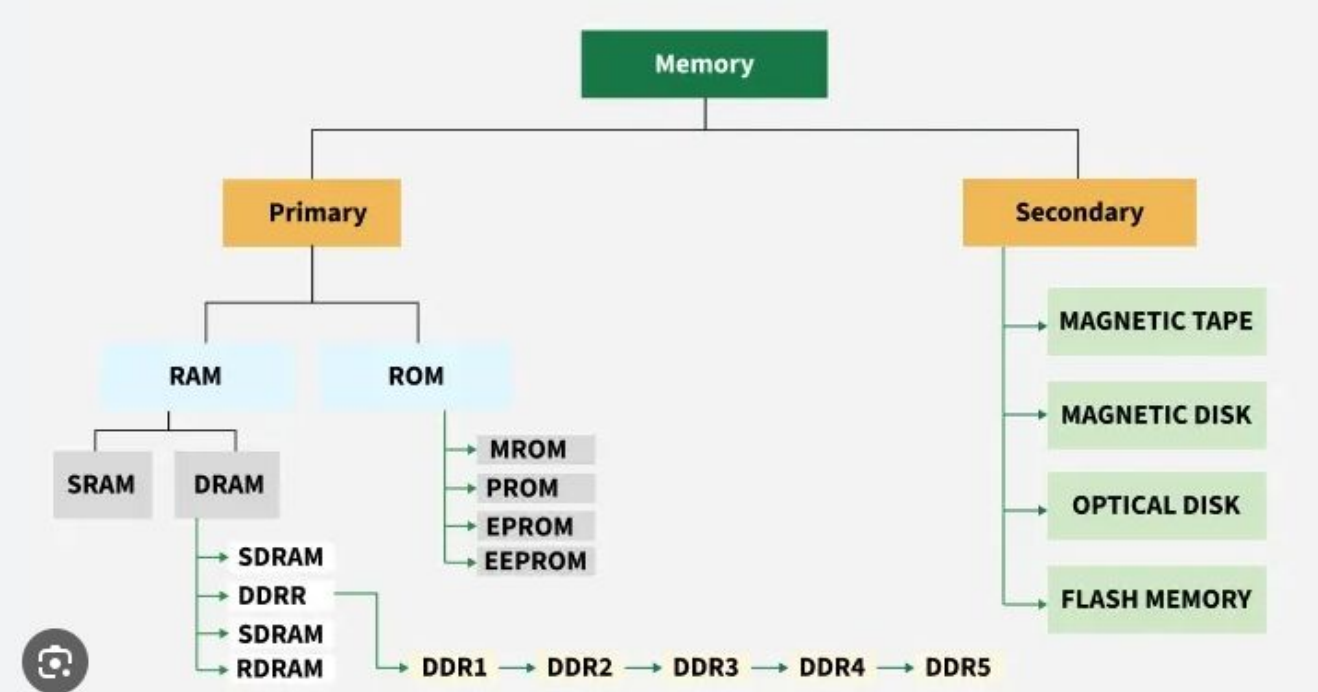
3. Control Bus: How can we tell if the address on the bus is memory address or an I/O device address? This is where the control bus comes in.



Microcomputers

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.



Brief History of the Computers

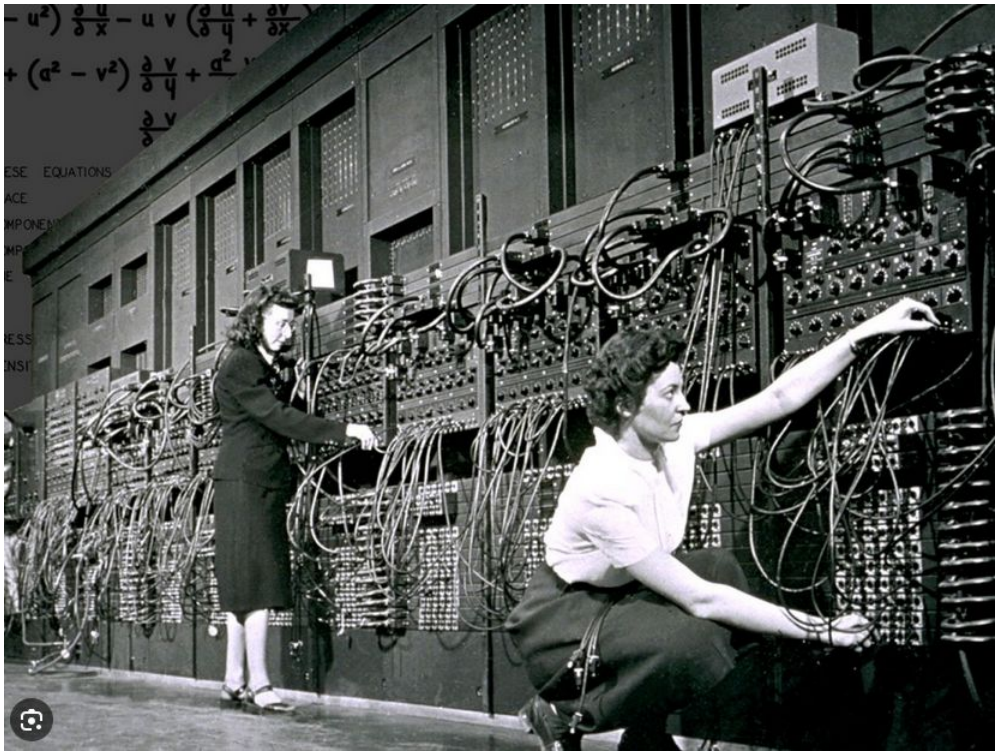
1946: The first generation of Computer **ENIAC** was started to be used based on the vacuum tube technology.

ENIAC was extremely large, occupying about 1,800 square feet and weighing almost 30 tons.

It contained approximately 17,500 vacuum tubes, 7,500 crystal diodes, 1,500 relays, and hundreds of thousands of resistors, capacitors, and inductors.

It could perform 5,000 additions, 357 multiplications, or 38 divisions in one second.

ENIAC was in operation until 1955.



Brief History of the Computers

1958: The first transistorized computer **TRADIC** was announced by IBM.

It contained about 800 transistors, which made it much smaller, more reliable, and more energy-efficient than its vacuum tube predecessors.

TRADIC used magnetic core memory. This type of memory allowed the computer to store data even when powered off.

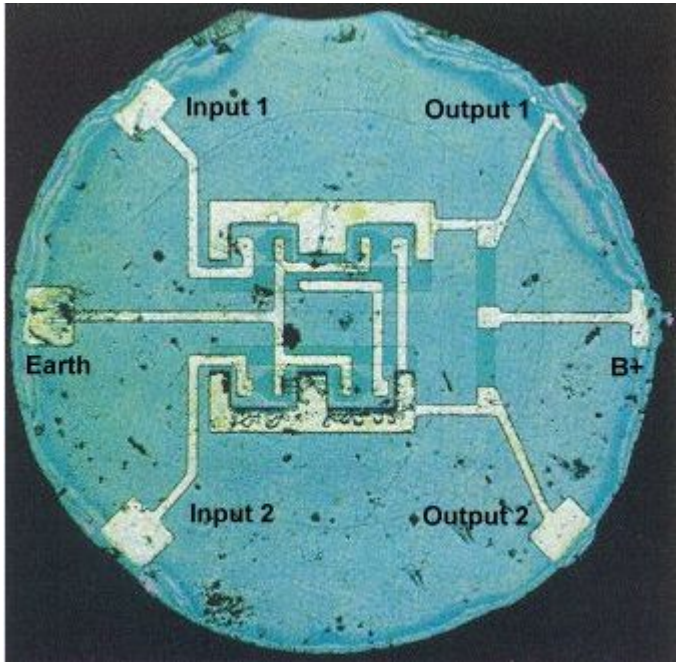
It could execute basic arithmetic and logical operations in microseconds.

Brief History of the Computers

1959: The first **IC (integrated circuit)** was invented.

1960s: ICs were started to be used in **CPU boards**.

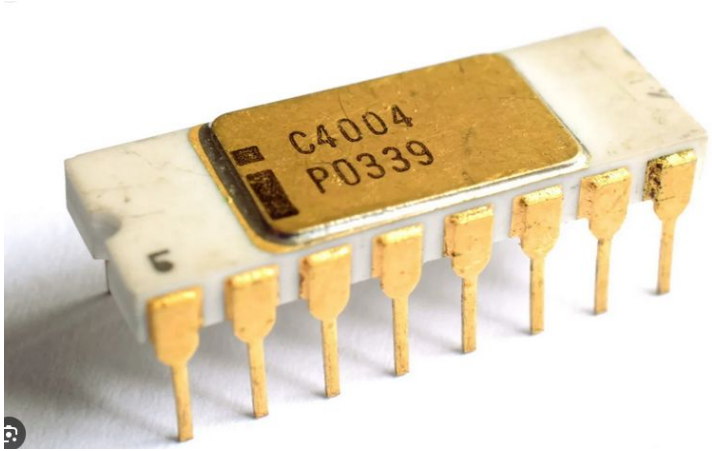
1970s: Entire CPU was put in a single chip.



Brief History of the Computers

1971 the first microprocessor of **Intel 4004**

- 4-bit data bus and 2300 transistors.
- Clock speed: 740 kHz.
- Instruction set: 45 instructions.
- address up to 640 bytes of program memory and up to 4 KB of data memory.
- Data memory was not on-chip; the 4004 interfaced with external memory for data storage.
- The 4004 was capable of executing approximately 60,000 instructions per second



Brief History of the Computers

1974: **Motorolla 6800** is introduced with 8-bit data bus and 16-bit address bus.

Late 1970s: Intel 8080/85 appeared with 8-bit data bus and 16-bit address bus and used from traffic light controllers to homemade computers.

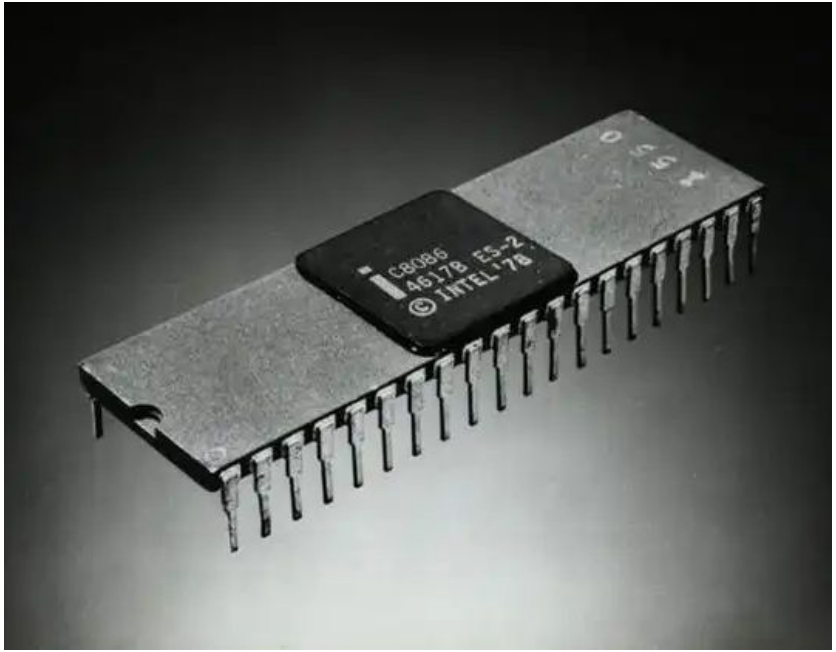
1978: **Intel 8086** is produced with 16-bit data bus and 20-bit address bus.

1981: First **PC** was introduced by IBM with **Intel 8088** microprocessor.

Brief History of the Computers

intel 8086

- 16-bit microprocessor.
- Clock speeds ranged from 5 MHz to 10 MHz.
- It could address up to 1 megabyte of memory.
- 20-bit address bus.
- supported a variety of data types and operations, including operations on strings.
- The 8086 introduced a form of **pipelining**, a technique that allows the overlap of instruction fetch and execution, improving the efficiency of instruction processing.



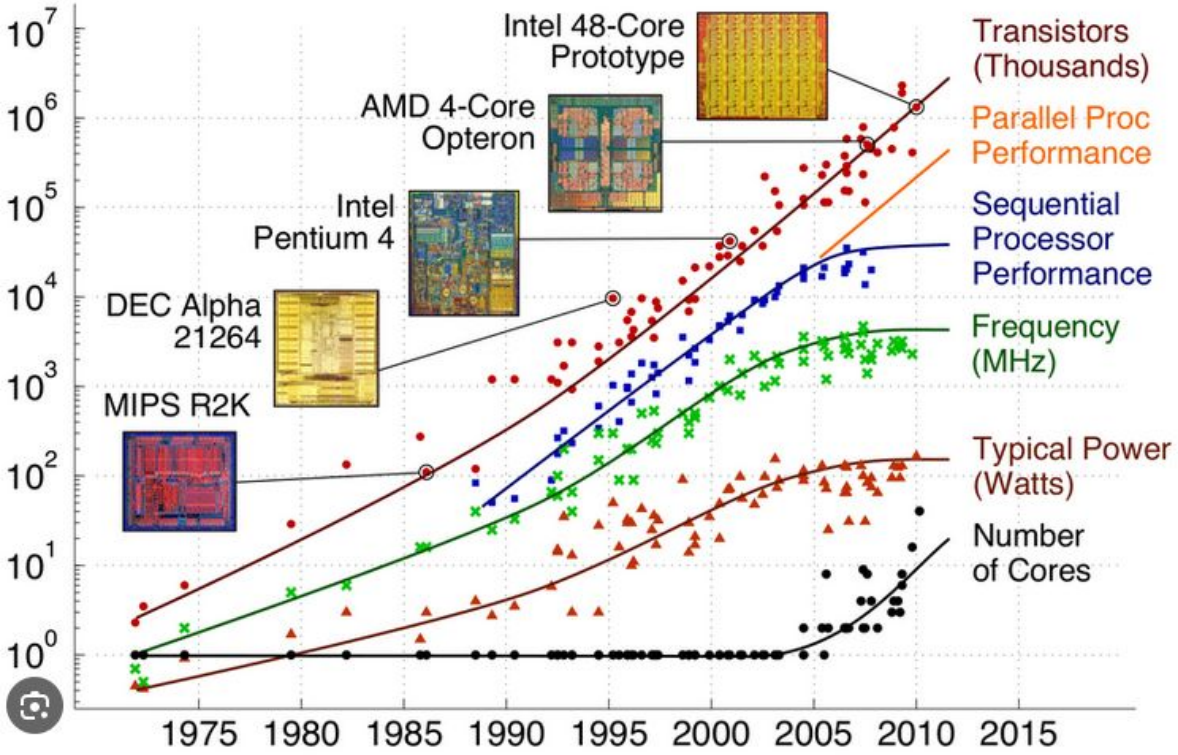
Evolution of Intel 80x86 Family Microprocessors

Processor	Year Intro.	Transistors	Clock Rate (MHz.)	External Data Bus	Internal Data Bus	Add. Bus
4004	1971	2,250	0.108	4	8	12
8008	1972	3,500	0.200	8	8	14
8080	1974	6,000	3	8	8	16
8085	1976	6,000	6	8	8	16
8086	1978	29,000	10	16	16	20
8088	1979	29,000	10	8	16	20
80286	1982	134,000	12.5	16	16	25
80386DX	1985	275,000	33	32	32	32
80386SX	1988	275,000	33	16	32	24
Pentium C	1993	3,100,000	66 -200	64	32	32
Pentium MMX	1997	4,500,000	300	64	32	32
Pentium Pro	1995	5,500,000	200	64	32	36
Pentium II	1997	7,500,000	233-450	64	32	36
Pentium III	1999	9,500,000	550-733	64	32	36
Itanium	2001	30,000,000	800-...	128	64	64

Evolution of Intel's Microprocessors

Product	8080	8085	8086	8088	80286	80386	80486
Year Introduced	1974	1976	1978	1979	1982	1985	1989
Clock rate (MHz)	2-3	3-8	5-10	5-8	6-16	16-33	25-50
No. transistors	4500	6500	29,000	29,000	130,000	275,000	1.2 million
Physical memory	64K	64K	1M	1M	16M	4G	4G
Internal data bus	8	8	16	16	16	32	32
External data bus	8	8	16	8	16	32	32
Address bus	16	16	20	20	24	32	32
Data type (bits)	8	8	8,16	8,16	8,16	8,16,32	8,16,32

Evolution of Microprocessors



The Modern Microprocessor: Striking Data & Trends

Computational density is reaching unprecedented levels.

- **Transistor Count:** Latest chips (e.g., Apple M2 Ultra, NVIDIA Blackwell) exceed 208 Billion transistors.
- **Process Node:** 2nm production (by TSMC/Samsung) is imminent. A human hair is ~40,000x wider.

AI is no longer software-only; it's now a core hardware feature.

- **Ubiquitous AI:** New smartphones and PCs (Intel Core Ultra, AMD Ryzen AI) feature dedicated NPUs (Neural Processing Units) for on-device AI tasks.
- **Data Center Demand:** Training large AI models (like ChatGPT) requires thousands of GPUs for months, costing millions in energy.

Performance is now directly tied to managing immense heat and power.

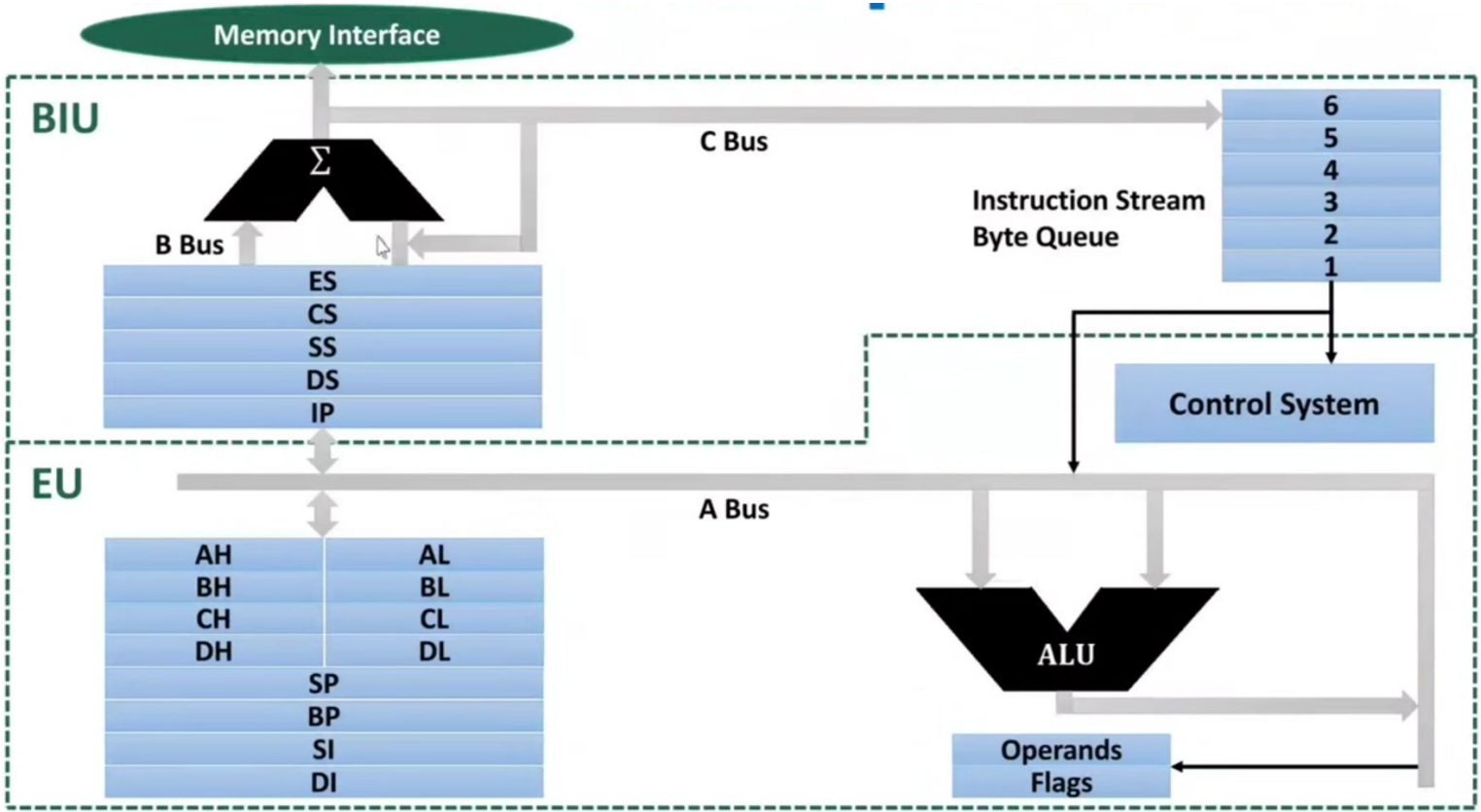
- **Mobile Chips (ARM):** Remarkably efficient, consuming under 5W.
- **Server/AI Chips (e.g., NVIDIA):** Can consume over 1000W (1kW)—enough to power a microwave.

The Modern Microprocessor: Striking Data & Trends

The era of "one architecture fits all" is over. Specialization is key.

- **Rise of ARM:** Apple's M-series chips demonstrated superior performance-per-watt, challenging decades of x86 (Intel/AMD) dominance in PCs.
- **Custom Silicon (ASICs):**
    - **Google:** TPU (Tensor Processing Unit) for AI.
    - **Amazon:** Graviton processors for AWS servers.
    - **Microsoft/OpenAI:** Designing their own AI chips.

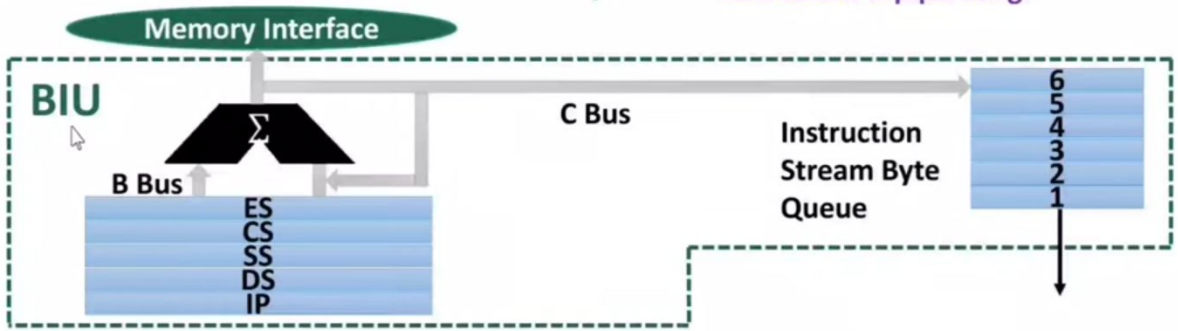
Architecture of 8086 Microprocessor



Bus interface unit (BIU) of 8086

- BIU is responsible for establishing communications with external peripheral devices and memory via system bus.
- Main Purpose of BIU serves is as followed
  - It fetches the instructions from Memory
  - It reads data from IO and Memory
  - It writes data into IO and Memory
  - It provides the address relocation facility
- BIU contains three main parts
  - Segment Registers
  - Instruction Queue
  - Instruction Pointer

- Instruction Queue
  - BIU prefetches six instruction bytes in advance from memory.
  - The prefetched instructions are stored in a group of high speed registers known instruction queue.
  - This Instruction queue works on FIFO order.
  - BIU and EU works in parallel.
  - The simultaneous operations of BIU and EU is possible only when the EU does not require the system bus.
  - The process of fetching the next instruction in advance while the EU is executing the current instruction is pipelining.



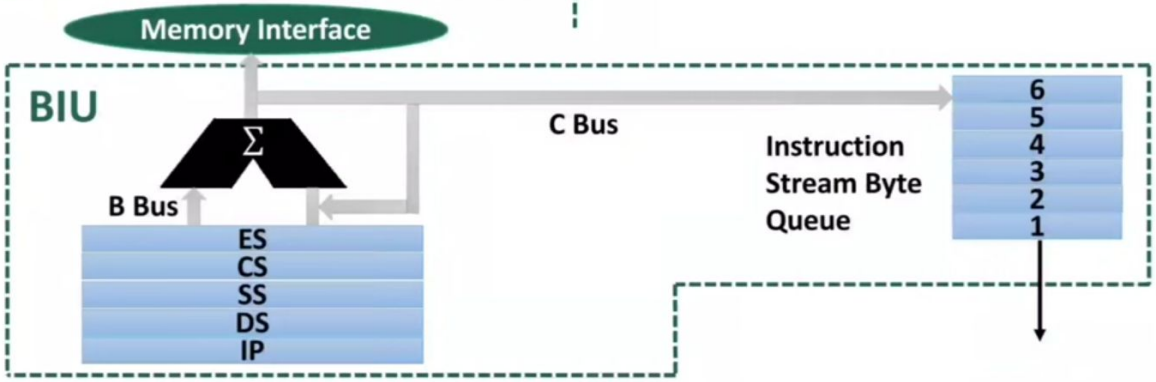
Bus interface unit (BIU) of 8086

- Segment Register of BIU
  - The 8086 MP has the capability of addressing 1MB memory, which is divided into 16 local segments.
  - Each segment contains 64KB memory
  - But any instant 8086 works with only four 64KB segments.
- Each segment associated with segment register
  - Code Segment Register CS – 16bits
  - Data Segment Register DS – 16bits
  - Stack Segment Register SS – 16bits
  - Extra Segment Register ES – 16bits

- Segment Register of BIU is used to store the starting address of Memory segment.
- BIU generates 20 bits address using segment register and Offset pointer registers.

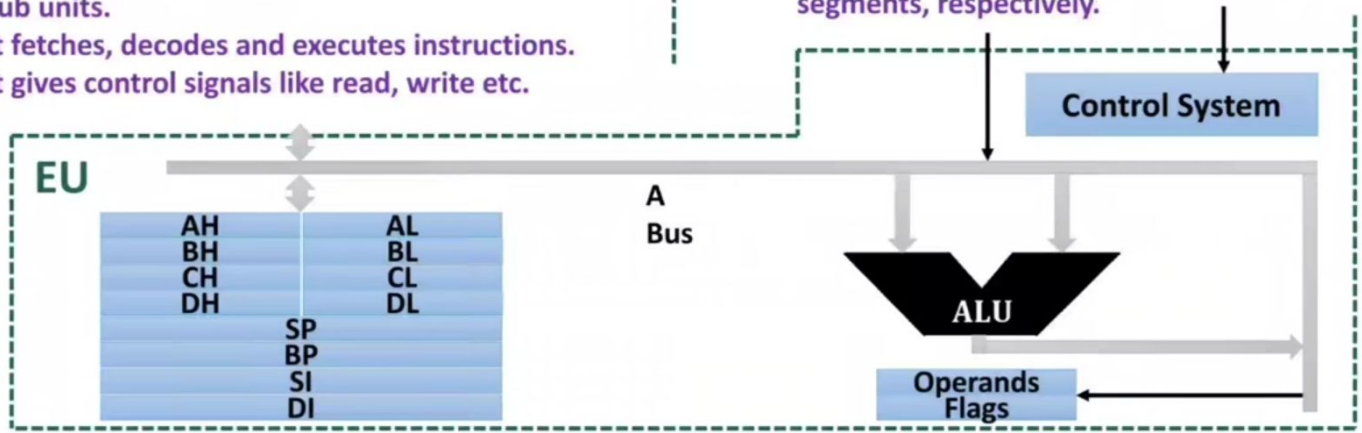
PA = SR X 10H + OP

- Instruction Pointer of BIU
  - IP holds the address of next instruction which is to be executed next.
  - It contains the OFFSET value of next Address.



Execution unit (EU) of 8086

- EU informs BIU from where the next instruction or data to be fetched.
- The EU performs following functions:
  - It picks up the instruction from the instruction queue of BIU.
  - It decode the instructions and then executes the instruction.
  - It updates the status of flag register.
- Control Unit
  - It is controlling and coordinating all the activities of sub units.
  - It fetches, decodes and executes instructions.
  - It gives control signals like read, write etc.
- ALU of EU
  - It performs all the arithmetic and logical operations.
  - Results may be stored in general purpose registers or index registers.
  - It updates Flag register after every instructions.
- General Purpose Registers & Pointer of EU
  - EU have four general purpose registers.
  - These registers are used for accessing data very fast.
  - EU have SP and BP for stack.
  - Index registers are SI and DI for data and Extra segments, respectively.

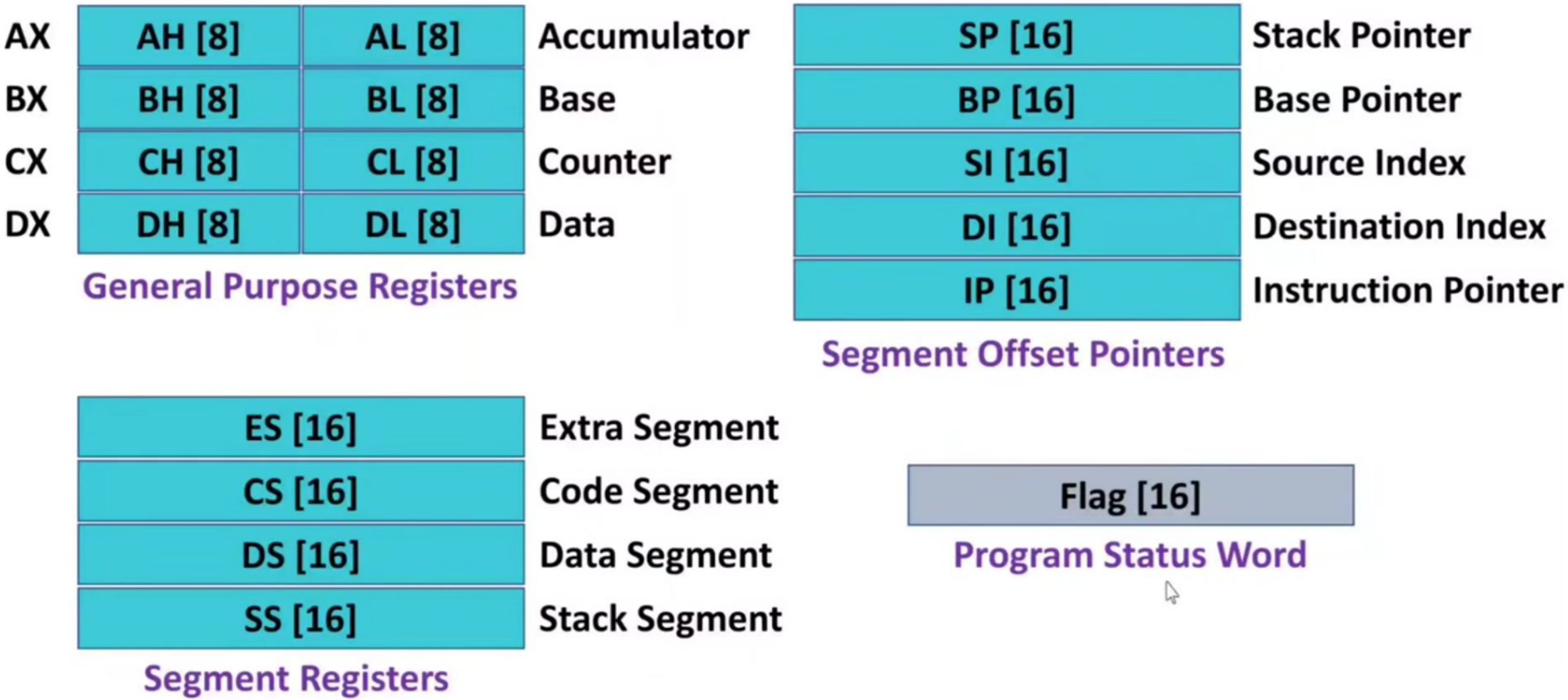


Pipelining in 8086

❖ How to increase Speed of execution of programs in microprocessor :

- ❑ Increasing data bus. [So, in single machine cycle we can transfer more data]
- ❑ Increasing System Clock. [So, more cycles can get executed in given time]
- ❑ Upgrading ALU. [Simplest explanation is 16 bits ALU performs more arithmetic and logical operations compared to 8 bits of ALU]
- ❑ Upgrading Memory technology. [Like now a days (2022), we have SSD & Cache in our computer]
- ❑ Upgrading Instruction Set. [Like in 8086 we have Multiplication and Division Instructions available which wasn't available with 8085]
- ❑ Pipelining [It helps to execute multiple instructions in parallel.]

The registers of the 8086

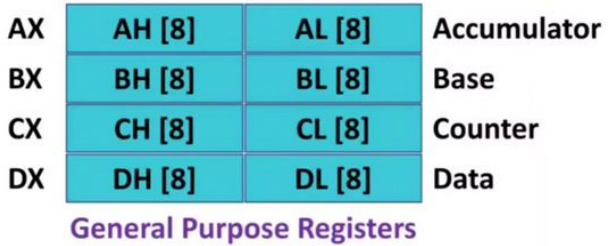


The registers of the 8086

General purpose registers

- **AX** (**accumulator)** is used for instructions such as multiplication, division, and some of the adjustment instructions. Generally considered to be a multipurpose register.
- **BX** (**base addressing register)** holds the offset address of a location in the memory system.
- **CX** (**counter loop operations)** holds the count for various instructions.
- **DX** (**data in I/O operations)** holds a part of the result from a multiplication or part of the dividend before a division.

General registers can be accessed as full **16 bits** (such as AX), or as the **high byte only** (AH) or **low byte only** (AL). The others are not!!

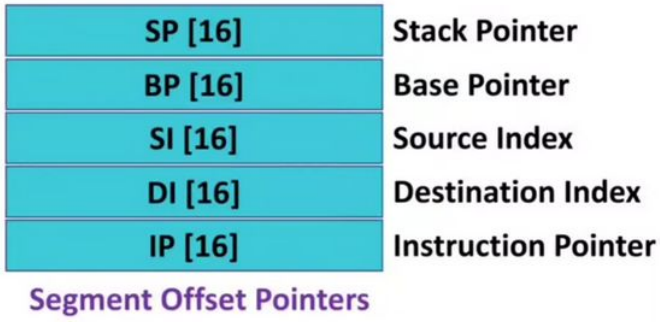


Category	Bits	Register Names
General	16	AX, BX, CX, DX
	8	AH, AL, BH, BL, CH, CL, DH, DL
Pointer	16	SP (stack pointer), BP(base pointer)
Index	16	SI (source index), DI(destination index)
Segment	16	CS (code segment), DS(data segment) SS (stack segment), ES(extra segment)
Instruction	16	IP (instruction pointer)
Flag	16	FR (flag register)

The registers of the 8086

Index registers

- **BP,** points to a memory location for memory data transfers.
- **DI**, addresses string destination data for the string instructions.
- **SI**. addresses source string data for the string instructions.
- **IP**, points to the next instruction in a program, is used by the microprocessor to find the next sequential instruction in a program located within the code segment.
- **SP,** addresses an area of memory called the stack.



Category	Bits	Register Names
General	16	AX, BX, CX, DX
	8	AH, AL, BH, BL, CH, CL, DH, DL
Pointer	16	SP (stack pointer), BP(base pointer)
Index	16	SI (source index), DI(destination index)
Segment	16	CS (code segment), DS(data segment) SS (stack segment), ES(extra segment)
Instruction	16	IP (instruction pointer)
Flag	16	FR (flag register)

The programming model of the 8086

Segment registers

- **CS**: contains the program code (instructions)
- **DS**; used to store data to be processed by the program
- **SS**; used to store information temporarily
- **ES**; used to store data like strings

Category	Bits	Register Names
General	16	AX, BX, CX, DX
	8	AH, AL, BH, BL, CH, CL, DH, DL
Pointer	16	SP (stack pointer), BP(base pointer)
Index	16	SI (source index), DI(destination index)
Segment	16	CS (code segment), DS(data segment) SS (stack segment), ES(extra segment)
Instruction	16	IP (instruction pointer)
Flag	16	FR (flag register)

Flag Register of 8086

Control Flags

Overflow Flag
1 = Overflow Occurred
0 = No Overflow

Direction Flag
1 = Auto Decrement
0 = Auto Increment
#Used with in string Instructions

Interrupt Flag
1 = Enable Interrupt
0 = Disable Interrupt

Trap Flag
1 = Perform Single Step
0 = Do Not perform Single Step

Sign Flag
1 = MSB of result is 1 [-ve Number]
0 = MSB of result is 0
#Used with signed operation

Zero Flag
1 = Result is Zero
0 = Result is not Zero

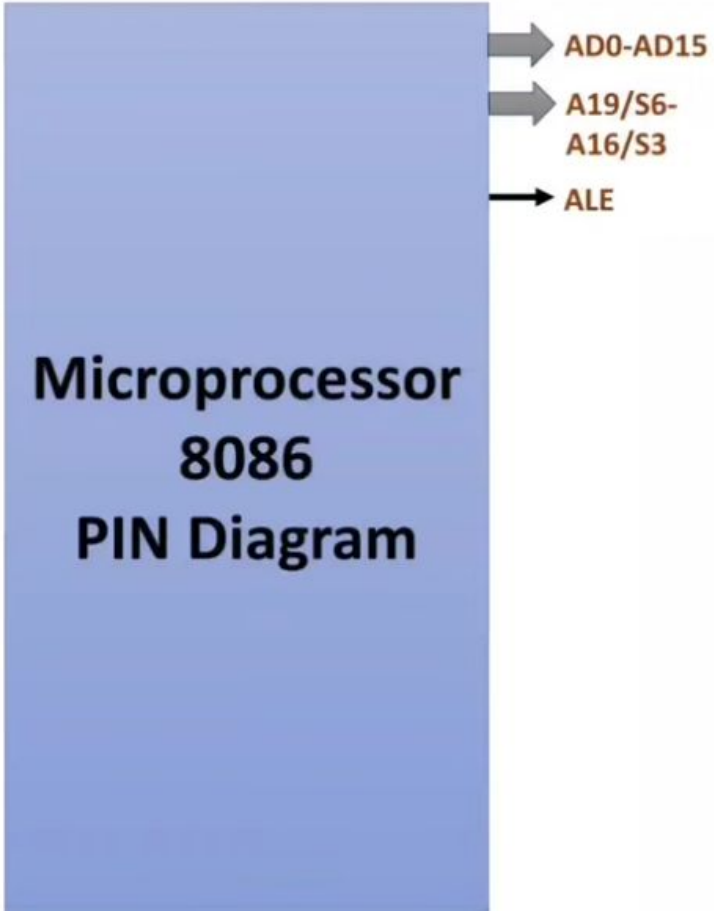
Auxiliary Carry Flag
1 = Carry from lower to higher Nibble
0 = No Auxiliary Carry
#Applicable in 8 bits Instructions

Parity Flag
1 = Even Parity
0 = Odd Parity

Carry Flag
0 = No Carry
1 = Carry from MSB

Category	Bits	Register Names
General	16	AX, BX, CX, DX
	8	AH, AL, BH, BL, CH, CL, DH, DL
Pointer	16	SP (stack pointer), BP(base pointer)
Index	16	SI (source index), DI(destination index)
Segment	16	CS (code segment), DS(data segment) SS (stack segment), ES(extra segment)
Instruction	16	IP (instruction pointer)
Flag	16	FR (flag register)

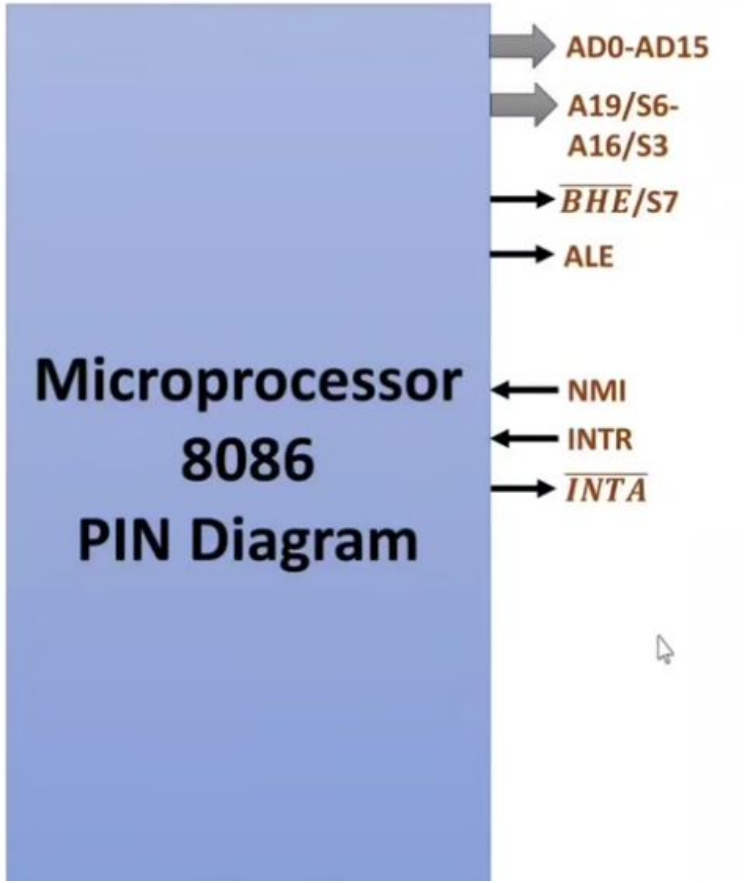
Pin Diagram of Microprocessor 8086



- **Address Data Bus – [AD0-AD15]**
  - 8086 has 20 lines for Address bus and 16 Data Lines.
  - Here, AD0 – AD15 are time multiplexed Address Data Lines and can be separated by ALE terminal.
  - If ALE = 1 then it carries Address [A0-A15] and If ALE = 0 then it carries data [D0-D15].
- **Address Status Lines – [A19/S6-A16/S3]**
  - 8086 has 4 time multiplexed Address status lines A19/S6 – A16/S3 and can be separated by ALE terminal.
  - If ALE = 1, then it carries Address [A19-A16] and If ALE = 0 then it carries Status signals [S6-S3].
- **Status signals [S7/S3]**
  - S3 & S4 indicates which segment is accessed by 8086 during current bus cycle.
  - S5 reflects IF flag register.
  - S6 is always zero.
  - S7 is always one.

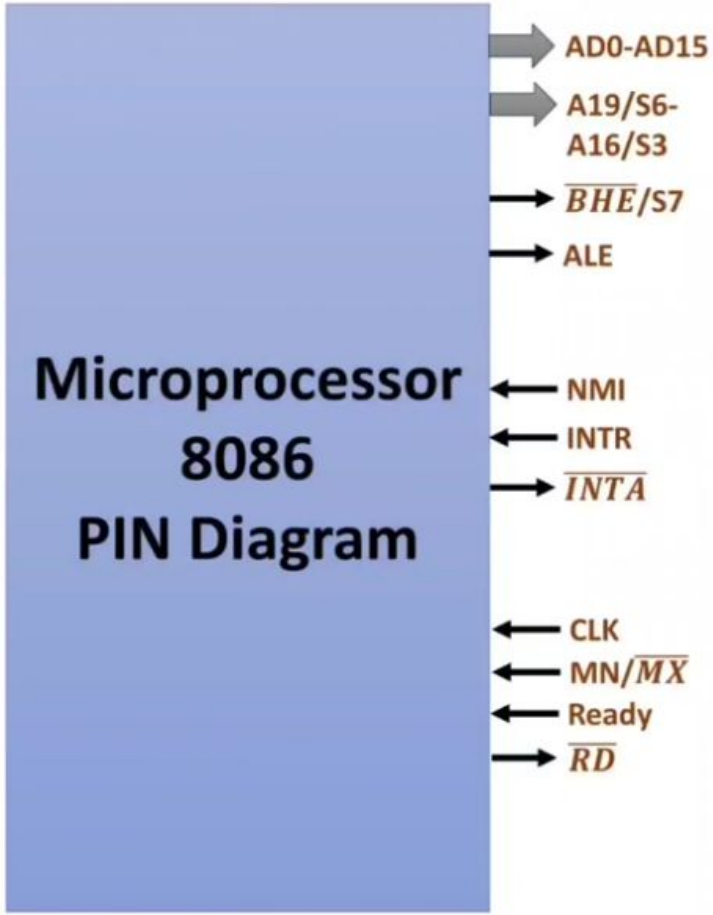
S4	S3	Segment Accessed
0	0	ES
0	1	SS
1	0	CS
1	1	DS

Pin Diagram of Microprocessor 8086



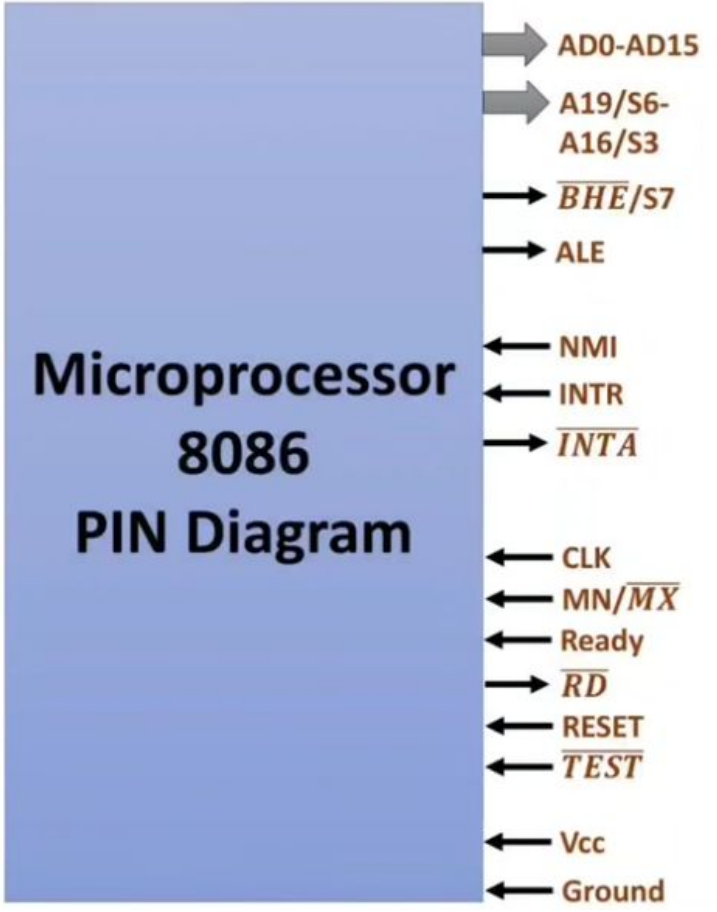
- Bus High Enable – [$\overline{BHE}$/S7]
  - This line is used to enable D15-D8 of D15-D0.
  - It is used by Microprocessor for memory banking.
- Non Maskable Interrupt – [NMI]
  - This line is used give highest priority interrupt to Microprocessor 8086.
  - It can not be disabled by the software.
  - It is positive edge triggered interrupt.
  - When it occurs, Type 2 Interrupt occurs in the 8086.
- Interrupt Request – [INTR]
  - This is level triggered hardware interrupt.
  - It depends of status of Interrupt Flag.
  - If IF = 1, INTR = 1, the 8086 gets interrupted.
  - If IF = 0, INTR = 1, then INTR is disabled.
- Interrupt Acknowledgment – [$\overline{INTA}$]
  - Along with Interrupt request on INTR, Microprocessor gives interrupt acknowledgment to peripherals.

Pin Diagram of Microprocessor 8086



- **System Clock – [CLK]**
  - Clock is given to 8086 for internal timings.
  - Clock with 8086 is 5MHz, 8MHz & 10Mhz.
- **Minimum & Maximum Mode [MN/$\overline{MX}$]**
  - For Minimum Mode connected with Vcc.
  - With Minimum Mode 8086 will work as single microprocessor.
  - For Maximum Mode connected with ground.
  - With Maximum mode 8086 will works with multiple coprocessor.
- **Ready Signal [READY]**
  - This input is used to insert the wait state into timing cycle of microprocessor 8086.
  - If the ready pin is at logic 1, it has no effect on the operation.
  - If it is logic 0, 8086 enters into the wait state like idle.
  - It is used to synchronize slow peripheral devices.
- **Read [$\overline{RD}$]**
  - If it is logic 0, Microprocessor reads data from Memory or IO devices.

Pin Diagram of Microprocessor 8086



- **System Reset – [RESET]**
  - This input will reset microprocessor 8086.
  - If it held logic 1 for minimum of 4 clock cycles then microprocessor will get RESET.
  - After RESET of 8086, CS and IP initialized to FFFFH and 0000H, so physical address will be FFFF0H.
  - Remaining registers initialized to zero.
- **Test [$\overline{TEST}$]**
  - It is used for synchronization.
  - When this input is logic 0, 8086 executes WAIT instruction.
- **Power Supply [Vcc & Ground]**
  - 8086 is given with 5 Volt DC supply.
  - Allowed variation is 10%.
  - 8086 has two ground pins.
  - This two ground pins are there to have less power dissipation.

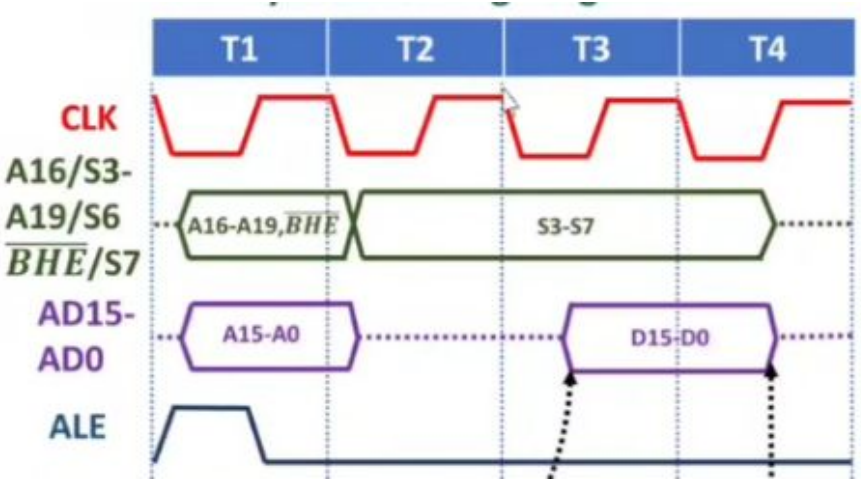
Multiplexed Address/Data Bus (AD0-AD15)

Multiplexed Address/Data Bus (AD0-AD15)

- **T1**: In the first clock cycle of a bus operation, the 8086 outputs the memory address on these lines.
- **T2, T3, T4:** In the subsequent cycles, the same lines are used to transmit or receive data.

The Role of **ALE** (Address Latch Enable):

- **ALE = 1 (High):** the information currently on the AD0-AD15 lines is a valid address (A0-A15).
  An external latch (like an 8282 or 74LS373 chip) uses this signal to "capture" and hold the address for the entire duration of the bus cycle.
- **ALE = 0 (Low)**: The AD0-AD15 lines are now being used for data (D0-D15).

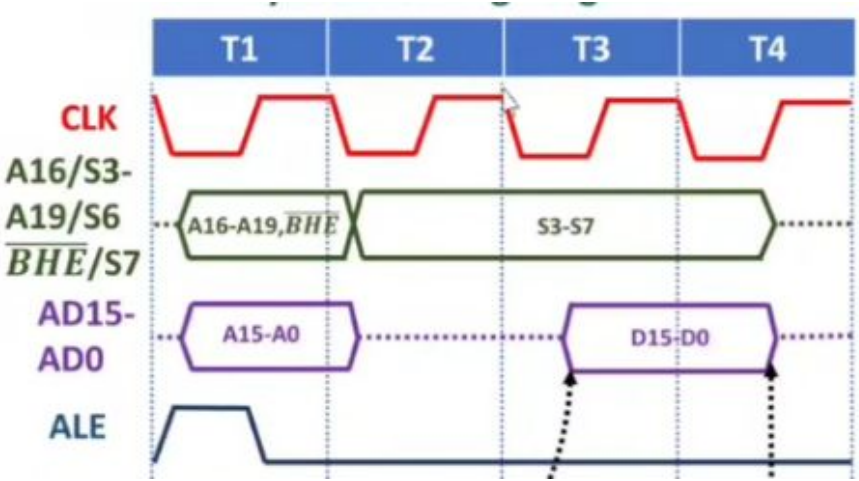


Multiplexed Address/Data Bus (AD0-AD15)

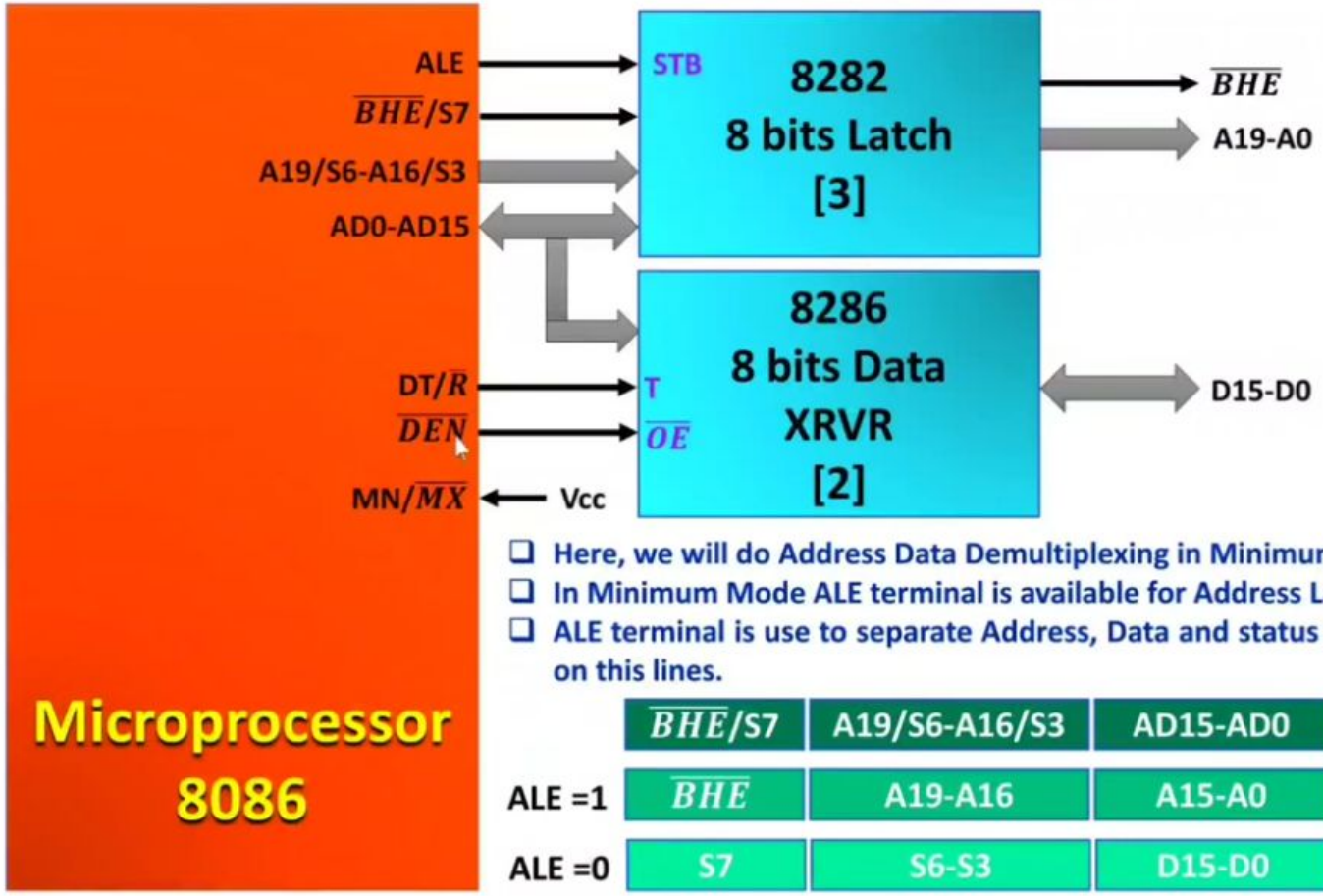
Multiplexed Address/Status Bus (A19/S6-A16/S3)

- **T1:** The lines carry the address bits A19-A16.
- **T2, T3, T4:** The lines carry the status signals S6-S3.

The Role of ALE: The same ALE signal is used to demultiplex these lines.
When ALE is **high**, external latches capture the address bits A19-A16.



Address Data Demultiplexing of 8086



- Here, we will do Address Data Demultiplexing in Minimum Mode. So connect MN/$\overline{MX}$ with Vcc.
- In Minimum Mode ALE terminal is available for Address Latch.
- ALE terminal is use to separate Address, Data and status signals. If ALE = 1, Address is available on this lines.

	$\overline{BHE}$/S7	A19/S6-A16/S3	AD15-AD0
ALE =1	$\overline{BHE}$	A19-A16	A15-A0
ALE =0	S7	S6-S3	D15-D0

S4	S3	Segment Accessed
0	0	ES
0	1	SS
1	0	CS
1	1	DS