A HISTORICAL BACKGROUND:

The idea of computing system is not new-it has been around long before modern electrical and electronic devices were developed. **Babylonians** invented the calculating device **Abacus**, the first mechanical calculator. The **Abacus, which** used strings of beads to perform calculations .It was used extensively and is still in use today. It was not improved until 1642. The mathematician **Blaise Pascal** invented a calculator in 1642 that was constructed of gears and wheels. The motor –driven adding machines were came after the advent of electric motors in 1800's.These are all based on the mechanical calculator developed by **Pascal.**

In the early 1870s, **Bomar** introduced the first small hand –held electronic calculator. In 1889, **Herman Hollerith** developed the punched card for storing data. In 1896, **Hellerith** formed a company called the **Tabulating Machine Company**, which developed a line of machines that was used punched cards for tabulation. After a number of mergers, the Tabulating Machine Company was formed into **International Business machines Corporation**, now it is referred more commonly as **IBM**. The punched cards used in computer systems are often called Hollerith cards.

The first Electronic calculating machine was invented by **Konrad Zuse** (German) in 1941. It has recently been discovered that the first electronic computer was placed into operation in 1943.This first electronic computing system, which used **Vacuum** tubes, was invented by **Alan Turing** .It was a fixed program computer system, today which is often called a special- purpose computer.

The first general –purpose programmable electronic computer system was developed in 1946 at the **University of Pennsylvania**. The first modern computer was called the **ENIAC** (**E**lectronics **N**umerical **I**ntegrator **a**nd **C**alculator).The ENIAC was a huge machine, containing over 17,000 vacuum tubes and over 500 miles of wires. This machine weighted over 30 tons, yet performed only about 1, 00,000 operations per second. Another problem with the ENIAC was the life of the vacuum tube components, which required frequent maintenance.

After development of **Transistors** in 1948 at **Bell Labs** and invention of the Integrated Circuits in 1958 by Jack Kilby of **Texas Instruments**, the **Intel Corporation** introduced the first **Microprocessor 4004 in 1971**.The device that started the Microprocessor revolution that continues today at an ever-accelerating pace.

INTRODUCTION TO 8086 MICROPROCESSOR:

Internal Architecture of 8086 has two blocks BIU and EU. The BIU performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands. The instruction bytes are transferred to the instruction queue. EU executes instructions from the instruction system byte queue. Both units operate asynchronously to give the 8086 an overlapping instruction fetch and execution mechanism which is called as Pipelining. This results in efficient use of the system bus and system performance. BIU contains Instruction queue, Segment registers, Instruction pointer, Address adder. EU contains Control circuitry, Instruction decoder, ALU, Pointer and Index register, Flag register. Bus Interfacr Unit**:** It provides a full 16 bit bidirectional data bus and 20 bit address bus. The bus interface unit is responsible for performing all external bus operations.

DEVELOPMENT OF MICROPROCESSORS (µP):

Intel introduced its first **4-bit** microprocessor **4004** in **1971**.The 4004 instruction set contained only 45 instructions. It execute instructions at the slow rate of 50 Kilo-instructions per second(KIPS).This was slow when compared to the 100,000 instructions executed per second by the ENIAC computer in1946.The 4-bit microprocessor is used in early video game systems and small microprocessor based control systems. Intel released **4040(4-bit)** microprocessor, an updated version of the earlier 4004.The 4040 microprocessor operated at a higher speed, although it lacked improvements in word width and memory size.

In **1972 Intel** Corporation released **8-bit microprocessor 8008**, which is an extended version of 4-bit microprocessor. The 8008 microprocessor is used in more advanced systems because it has an expanded **memory size (16KB**) and contained a additional instructions(48) .It executes 5,00,000 instructions per second. But it is some what small memory size, slow speed and instruction set limited its usefulness. These microprocessors

could not survive as general purpose microprocessors due to their design and performance limitations.

The launch of the first general purpose **8-bit microprocessor 8080** in **1974** by **Intel** is considered to be the first major stepping stone towards the development of advanced microprocessors. It executes the instruction s10 times faster than the 8008.Also the 8080 was compatible with TTL, where as the 8008 was not directly compatible. This made interfacing much easier and less expensive. The memory size of **8080 is 64KB**.

In **1977 Intel** Corporation introduced an updated version of the 8080 microprocessor –the **8085**.This was to be the last **8-bit** general purpose microprocessor developed by Intel. The 8085 executed software at a higher speed. It executes 769230 instructions per second. The main advantages of 8085 were its internal clock generator, internal system controller and higher clock frequency.

The main limitations of the 8-bit microprocessor were their low speed, low memory addressing capability, limited number of general purpose registers and a less powerful instruction set. All these limitations of the 8-bit microprocessors pushed the designers to build more powerful processors in terms of advanced architecture, more processing capability, larger memory addressing capability and a more powerful instruction set. The 8086 was a result of such development design efforts.

In the family of **16-bit microprocessors**, **Intel's 8086** were the first one to be launched in **1978**.The introduction of the 16-bit microprocessor was a result of the increasing demand for more powerful and high speed computational resources. The 8086 microprocessor has a much powerful instruction set along with the architectural developments which imparts substantial programming flexibility and improvement in speed over the 8-bit microprocessors. A year or so later Intel released the **8088** microprocessor. It is also a **16-bit** microprocessor. The memory size of the **8086/8088** processor is **1MB**,which was 16 times more memory than 8085.One another feature found in 8086 was a small 6-byte instruction queue, that prefetched a few instructions before they were executed. Note that these types of microprocessors were called CISC (Complex Instruction Set Computers) because of the number and complexity of instructions. The popularity of the ***Intel family*** was ensured in **1981**, when **IBM** Corporation decided to use the ***8088*** **microprocessor in its Personal Computer.**

In **1983**, Intel introduced the **80286 microprocessor;** an updated version of 8086.It is also a **16-bit** microprocessor. It was almost identical to the 8086 and 8088, except memory size. The memory size of **80286 is 16MB**.

In **1986, Intel** released **80386** microprocessor keeping in mind faster microprocessor speeds, more memory size and wider data paths. It is **32-bit** microprocessor. So, it contained 32-bit data bus. The memory size is **4GB** (32-bit address bus).

Some of the advanced processors developed by Intel Corporation are 80486, Pentium, PentiumII, PentiumII Xeaon, Pentium III, Pentium-4, etc.

ARCHITECTURE OF 8086 MICROPROCESSOR:

Features:

Intel released its first 16-bit microprocessor 8086 in 1978.The Intel 8086 is a 16-bit processor ,which is fabricated using HMOS technology and it has 40 pins ,packaged in DIP .

The 8086 is a 16-bit microprocessor. The term 16-bit means that its arithmetic logic unit, internal registers and most of its instructions are designed to work with 16-bit binary words.

The 8086 has a 16-bit data bus, so it can read data from or write data to memory and ports either 16 bits or 8 bits at a time. The 8088, however has an 8-bit data bus, so it can only read data from or write data to memory and ports 8 bits at a time.

The 8086 has a 20-bit address bus, so it can directly access 2^{20} or 10, 48,576 (1MB) memory locations. Each of the 10, 48,576 (1MB) memory locations is byte wide. Therefore, a 16-bit word is stored in two consecutive memory locations. The 8088 also has a 20-bit address bus, so it can also address 2^{20} memory locations.

The 8086 can generate 16-bit I/O address, hence it can access 2^{16}=65536 I/O ports.

The 8086 provides fourteen 16-bit registers.

The 8086 has multiplexed address and data bus which reduces the number of pins needed, but does slow down the transfer of data.

The 8086 requires clock with a 33% duty cycle to provide optimized internal timing.

The 8086 microprocessor available in three clock rates: 5 MHz (8086), 8 MHz (8086-2) and 10 MHz (8085-1).

The Intel 8086 is designed to operate in two modes: minimum mode and maximum mode.

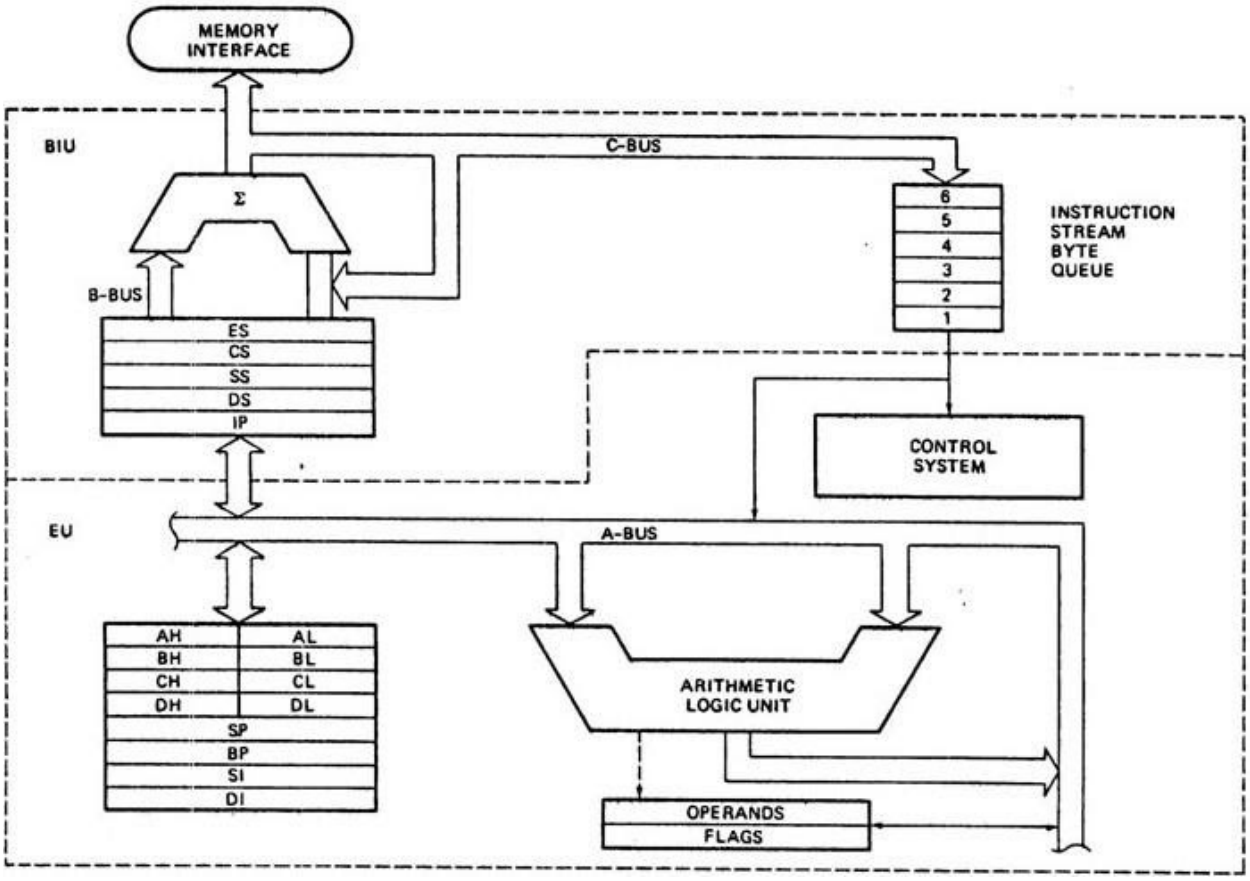
Minimum mode: When only one 8086 CPU is to be used in a microcomputer system, the 8086 is used in the minimum mode of operation.

Maximum mode: More than one processor (multiprocessor) is used in the system, the 8086 is used in the maximum mode of operation.

An interesting feature of the 8086 is that it fetches up to 6 instruction bytes from memory and queue stores them in order to speed up instruction execution.

It requires +5V single power supply.

The 8086 processor is divided into two independent functional parts, the Bus Interfacing Unit (BIU), and the Execution Unit (EU). Dividing the work between these two units speeds up processing. The BIU sends out addresses, fetches instructions from memory, reads data from ports and memory, and writes data to ports and memory. In other words, the BIU handles all transfers of data and addresses on the buses for the execution. The EU of the 8086 tells the BIU where to fetch instructions or data from, decodes instructions, and executes instructions.



1. **Bus Interfacing Unit:** The bus interfacing unit in 8086 provides the interface to the outside world. This unit is responsible for performing all external bus operations like fetches instructions from memory, reads data from ports and memory, and writes data to ports and memory.

The queue: To speed up program execution, the BIU fetches *six instruction bytes* ahead of time from the memory. The BIU stores these perfected bytes in a first-in-first-out register set called a **queue**. When the EU is ready for its next instruction, it simply reads the instruction byte for the instruction from the queue in the BIU. This is much faster than sending out an address to the system memory and waiting for memory to send back the next instruction byte or bytes. The process is analogous to the way a bricklayer's assistant fetches bricks ahead of time and keeps a queue of bricks lined up so that the brick layer can just reach out and grab a brick when necessary. Except in the cases of JMP and CALL instructions, where the queue must be damped and then reloaded starting from a new address. So, the queue operates on the principle first in first out (FIFO).So that the execution unit gets the instructions for execution in the order they are fetched. This prefetch-and-queue scheme greatly speeds up processing. Fetches the next instruction while the current instruction executes is called **pipelining**.

2. **Execution Unit:** The EU of 8086 tells the BIU from where to fetch instructions or data, decodes instructions and executes instructions. As shown in figure, the EU contains: control unit, decoder, ALU and a registers.

Control unit: This directs internal operations.

Decoder: A decoder in the EU translates instructions fetched from memory into a series of actions which the EU carries out.

ALU: The EU has a 16-bit arithmetic logic unit which can add, subtract, AND, OR, XOR, increment, decrement, complement and shift binary numbers.

REGISTER ORGANIZATION OF *8086* MICROPROCESSOR:

The 8086 has four groups of the user accessible internal registers. They are the instruction pointer, four data registers, four pointer and index register, four segment registers.

The 8086 has a total of fourteen 16-bit registers including a 16 bit register called the ***status register***, with 9 of bits implemented for status and control flags. Most of the registers contain data/instruction offsets within 64 KB memory segment. There are four different 64 KB segments for instructions, stack, data and extra data. To specify where in 1 MB of processor memory these 4 segments are located the processor uses four segment registers:

Code segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions.

Stack segment (SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment. SS register can be changed directly using POP instruction.

Data segment (DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions.

Extra segment (ES) is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions. It is possible to change default segments used by general and index registers by prefixing instructions with a CS, SS, DS or ES prefix.

All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. The general registers are:

Accumulator register 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. Accumulator can be used for I/O operations and string manipulation.

Base register 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. BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

Count register 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. Count register can be used in Loop, shift/rotate instructions and as a counter in string manipulation,.

Data register 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. Data register can be used as a port number in I/O operations. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

The following registers are both general and index registers:

Stack Pointer (SP) is a 16-bit register pointing to program stack.

Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data address in string manipulation instructions.

Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data address in string manipulation instructions.

Other registers:

Instruction Pointer (IP) is a 16-bit register.

Flag register is a 16-bit register containing 9 one bit flags.

Overflow Flag (OF) - set if the result is too large positive number, or is too small negative number to fit into destination operand.

Direction Flag (DF) - if set then string manipulation instructions will auto-decrement index registers. If cleared then the index registers will be auto-incremented.

Interrupt-enable Flag (IF) - setting this bit enables maskable interrupts.

Single-step Flag (TF) - if set then single-step interrupt will occur after the next instruction.

Sign Flag (SF) - set if the most significant bit of the result is set.

Zero Flag (ZF) - set if the result is zero

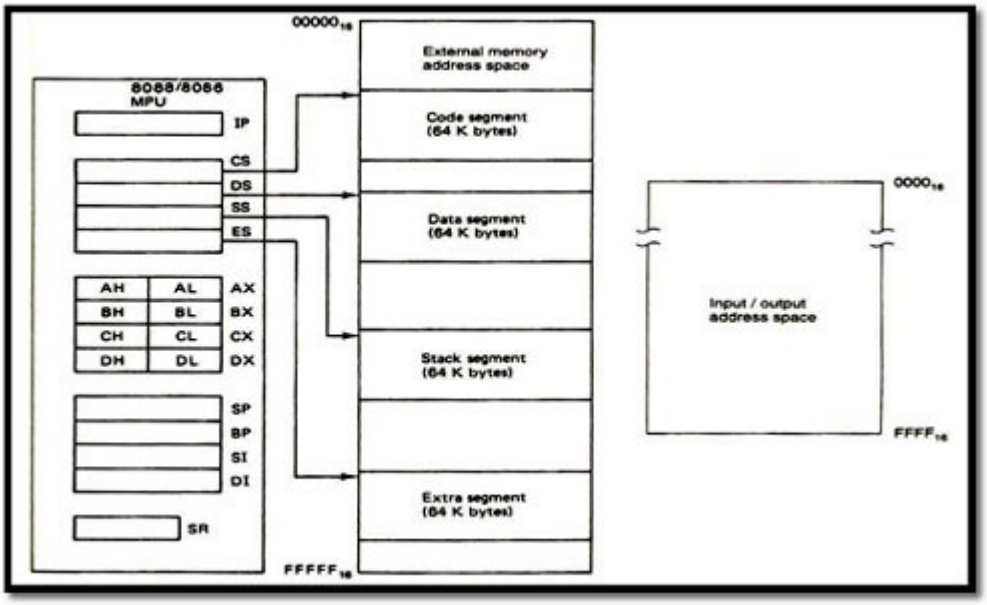
Auxiliary carry Flag (AF) - set if there was a carry from or borrow to bits 0-3 in the AL register.

Parity Flag (PF) - set if parity (the number of "1" bits) in the low-order byte of the result is even.

Carry Flag (CF) - set if there was a carry from or borrow to the most significant bit during last result calculation.

MEMORY ORGANIZATION

The size of address bus of 8086 is 20 and is able to address 1 Mbytes of physical memory. The complete 1 Mbytes memory can be divided into 16 segments, each of 16 Kbytes size. The addresses of the segment may be assigned as 0000*H* to *F*000*H* respectively. The offset values are from 0000*H* to *FFFFFH*. If the segmentation is done as per above mentioned way, the segments are called non-overlapping segments. In some cases segment may overlap also. Suppose a segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts before this 64 Kbytes location of the first segment, the two segments are said to be overlapping segment. The area of memory from the start of the second segment to the possible end of the first segment is called as overlapped segment.



SYSTEM BUS STRUCTURE

BASIC 8086 CONFIGURATION

Minimum mode versus maximum mode

There are 2 available modes of operation for the 8086/8088 microprocessors: minimum and maximum mode. Minimum mode operation is obtained by connecting the mode selection pin MN/MX bar to +5 v, and maximum mode is selected by grounding this pin. Both modes enable different control structures for the 8086/8088 microprocessors. The mode of operation provided by minimum mode is familiar to that of the 8085 A, the most recent intel 8 bit up, whereas maximum mode is new and unique and designed to be used whenever a coprocessor exists in a system.

Minimum mode operation

Minimum mode operation is the least expensive way to operate the 8086/8088 microprocessors. It costs less because aa the control signals for the memory and i/o are generated by the microprocessor. These control signals are identical to those of the intel 8085 A, an earlier 8 bit microprocessor. The minimum mode allows the 8085 A, a 8 bit peripherals to be used with the 8086/8077 without any special considerations.

Maximum mode operation

Maximum mode operation differs from Minimum mode operation in that some of the control

signals must be externally generated. This requires the addition of an external bus controller-8288 bus controller. There are not enough pins on the 8086/8088 for the bus control during maximum mode because new pins and new features have replaced some of them. Maximum mode is used only when the system contains external processors such as the 8087 arithmetic coprocessor

8086 INTERFACING

When the minimum mode operation is selected, the 8086 provides all control signals needed to implement the memory and I/O interface. The minimum mode signal can be divided into thee following basic groups:

1. Addresss bus/data bus
2. Status
3. Control
4. Interrupt
5. DMA

Address/data bus:

These lines serve 2 functions. As an address bus is 20 bits long and consists of signals lines A0 through A19,A19represents the MSB and A0,LSB. A 20 bit address gives the 8086 a 1 Mbyte memory address space. It has an independent I/O address space which is 64 K bytes in length. The 16 databus lines D0 through D15 are actually multiplexed with address lines A0 through A15 respectively. By multiplexed, we mean that bus work as an address bus during first machine cycle and as a data bus during next machine cycles,

D15 is the MSB and D0 LSB. When acting as a data bus, they carry read/write data for memory, input/output data for I/O devices, and the interrupt type codes from an

interrupt controller.

MINIMUM-MODE AND MAXIMUM-MODE

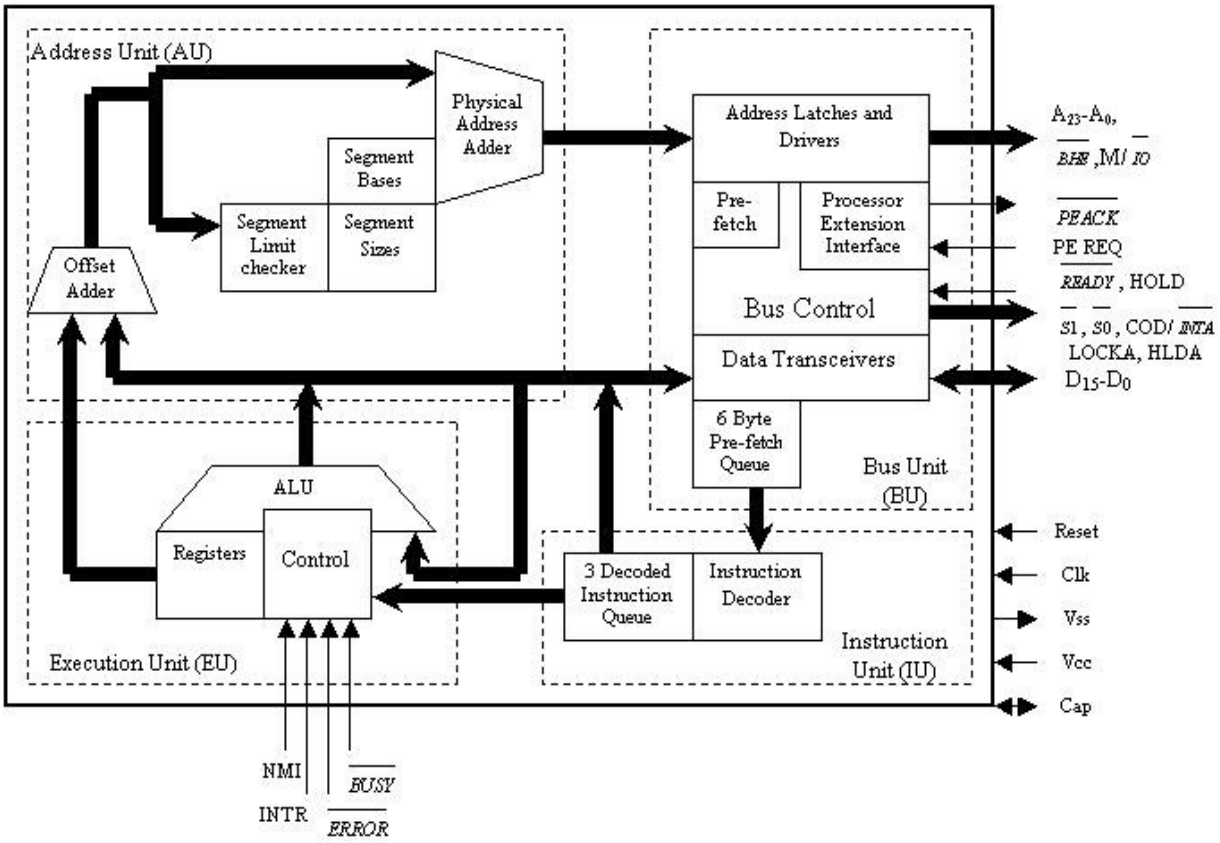
The 8086 can be configured to work in either of two modes:

• The minimum mode is selected by applying logic 1 to the MN/MX input lead. It is typically used for smaller single microprocessor systems.

• The maximum mode is selected by applying logic 0 to the MN/MX input lead. It is typically used for larger multiple microprocessor systems.

ARCHITECTURE OF INTEL 80286

The 80286 was designed for multi-user systems with multitasking applications, including communications and real-time process control. It had 134,000 transistors and consisted of four independent units: address unit, bus unit, instruction unit and execution unit. These were organized into a pipeline, significantly increasing performance. It was produced in a 68-pin package including PLCC (Plastic Leaded Chip Carrier), LCC (Leadless chip carrier) and PGA (Pin Grid Array) packages.



The Intel 80286 had a 24-bit address bus and was able to address up to 16 MB of RAM, compared to 1 MB for its predecessor. However cost and initial rarity of software using the memory above 1 MB meant that 80286 computers were rarely shipped with more than one megabyte of RAM.

80286 Architecture contains 4 separate processing units:

Bus Unit (BU):

It has address latches, data transceivers, bus interface and circuitry, instruction pre-fetcher, processor extension interface and 6 byte instruction queue.

Functions :

- To perform all memory and I/O read and write.
- To pre-fetch the instruction bytes.
- To control the transfer of data to and from processor extension devices like 80287 math co-processor.
- Whenever BU is not using the buses for the operation, it pre-fetches the instruction bytes and put them is a 6 byte pre-fetch queue.

Instruction Unit (IU):

It has 3 decoded instruction queue and instruction decoder.

Functions :

- It fully decodes up to three prefetched instructions and holds them in a queue.
- So that EU can access them.
- It helps the processor to speed up, as pipelining of instruction is done.

Execution Unit (EU):

It includes ALU, registers and the Control unit. Registers are general purpose, index, pointer, flag register and 16 –bit Machine Status Word (MSW).

Functions :

- To sequentially execute the instructions received from the instruction unit.
- ALU result is either stored in register bank or sent over the data bus.

Address Unit (AU):

It consists of segment registers, offset address and a physical address adder.

Functions :

- Compute the physical address that will be sent out to the memory or I/O by BU.
- 80286 operate in two different modes
  1. real address mode
  2. Protected virtual address mode.
- When used in Real address mode, AU computes the address with segment base and offset like 8086. Segment register are CS, DS, ES and SS hold base address. IP, BP, SI, DI , SP hold offset.
- Maximum physical space allowed in this mode is 1MB.
- When 80286 operate in protected mode, the address unit acts as MMU.
- All 24 address lines used and can access up to 16MB of physical memory.
- If descriptor table scheme is used it can address up to 1GB of virtual memory.

80386 Architecture

internal architecture of the 80386 includes six functional units that operate in parallel.

- The The parallel operation is called as pipeline processing.

Fetching, decoding execution, memory management, and bus access for several instructions are performed simultaneously.

- The six functional units of the 80386 are
  1. Bus Interface Unit
  2. Code Pre-fetch Unit

3. Instruction Decoder Unit
4. Execution Unit
5. Segmentation Unit
6. Paging Unit

- The **Bus Interface Unit** connects the 80386 with memory and I/O. Based on internal requests for fetching instructions and transferring data from the code pre-fetch unit, the 80386 generates the address, data and control signals for the current bus cycles.

- The **code pre-fetch unit** pre-fetches instructions when the bus interface unit is not executing the bus cycles. It then stores them in a 16-byte instruction queue for decoding by the instruction decode unit.

0 The **instruction decode unit** translates instructions from the pre-fetch queue into micro-codes. The decoded instructions are then stored in an instruction queue (FIFO) for processing by the execution unit.

1 The **execution unit** processes the instructions from the instruction queue. It contains a control unit, a data unit and a protection test unit.

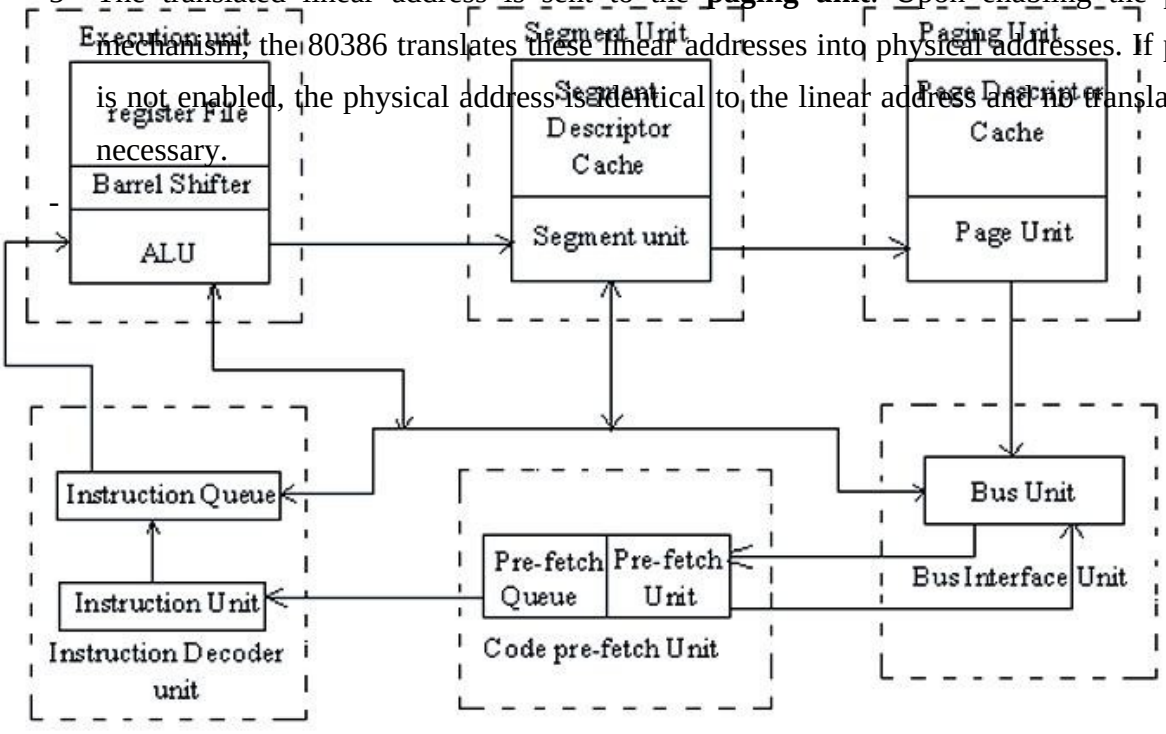
2 The **control unit** contains microcode and parallel hardware for fast multiply, divide and effective address calculation. The unit includes a 32-bit ALU, 8 general purpose registers and a 64-bit barrel shifter for performing multiple bit shifts in one clock. The data unit carries out data operations requested by the control unit.

3 The **protection test unit** checks for segmentation violations under the control of microcode.

4 The **segmentation unit** calculates and translates the logical address into linear addresses at the request of the execution unit.

5 The translated linear address is sent to the **paging unit**. Upon enabling the paging mechanism, the 80386 translates these linear addresses into physical addresses. If paging is not enabled, the physical address is identical to the linear address and no translation is necessary.

-

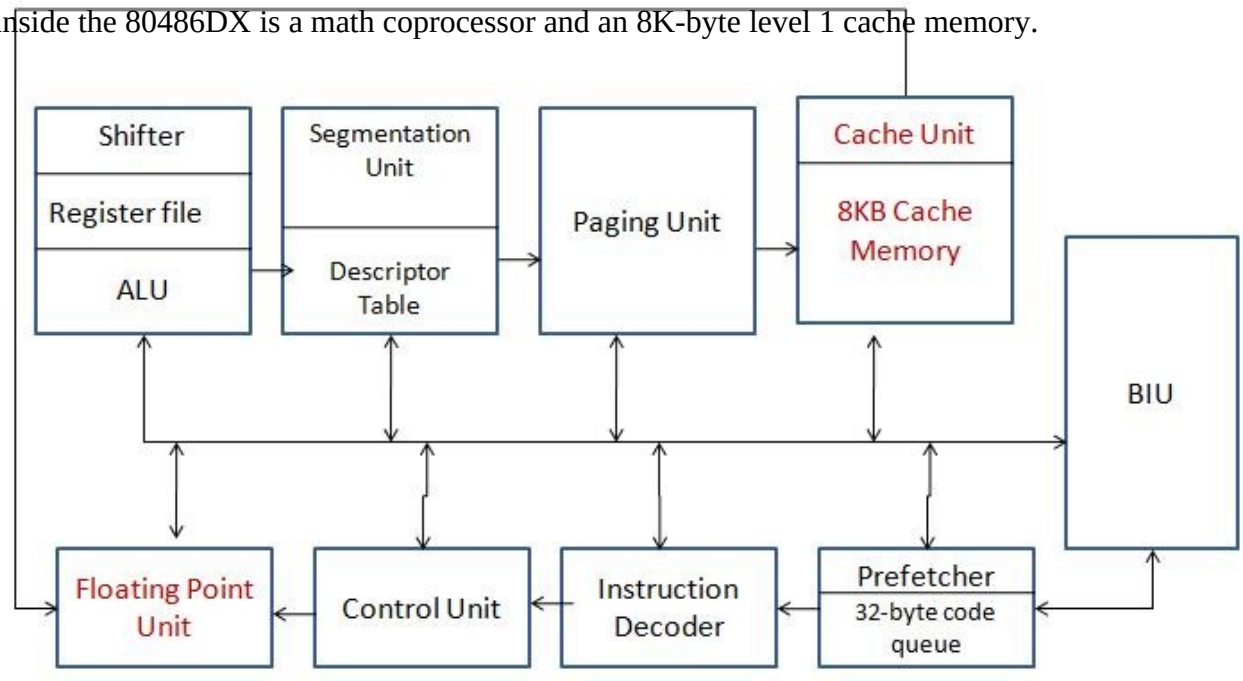


80486 Architecture

The 32-bit 80486 is the next evolutionary step up from the 80386.

- One of the most obvious feature included in a 80486 is a built in math coprocessor. This coprocessor is essentially the same as the 80387 processor used with a 80386, but being integrated on the chip allows it to execute math instructions about three times as fast as a 80386/387 combination.
- 80486 is an 8Kbyte code and data cache.
- To make room for the additional signals, the 80486 is packaged in a 168 pin, pin grid array package instead of the 132 pin PGA used for the 80386.
- Operates on 25MHz, 33 MHz, 50 MHz, 60 MHz, 66 MHz or 100MHz.
- It consists of parity generator/checker unit in order to implement parity detection and generation for memory reads and writes.
- Supports burst memory reads and writes to implement fast cache fills.
- Three mode of operation: real, protected and virtual 8086 mode.
- The 80486 microprocessor is a highly integrated device, containing well over 1.2 million transistors.

The architecture of the 80486DX is almost identical to the 80386. Added to the 80386 architecture inside the 80486DX is a math coprocessor and an 8K-byte level 1 cache memory.



BIU:

- BIU generates address, data and control signals for a bus cycle it is supported with an additional parity detection/generation for memory reads and writes.
- During memory write operation, the 486 generates even parity bit for each byte outputs these bits.
- These bits will be stored in a separate parity memory bank.
- During read operation, stored parity bits will be read from the parity memory.
- 80486 checks the parities of data bytes read and compare them with the DP0 – DP3 signals and generates parity check error, if it occurs.

- It pre-fetches the instruction bytes in advance and holds them in a 32–byte code queue. Instruction Decoder :
- Decodes the instructions in the queue and passes the control and protection test unit. Execution Unit:
- Executes the instruction with the help of Barrel Shifter, ALU and Register bank. Segmentation Unit and Paging Unit :
- They are part of MMU (which manages virtual memory of system). Helpful in generation of Physical Address.
- Work same as they work in 80386.
- Responsible for performing the floating point operations.
- The 80486 has a built in math co-processor . It performs the floating point operations.
- It executes math instructions 3 times faster than the 386/387 combination.

Cache Unit :

- 8KB cache
- Additional high speed cache memory provides a way of improving overall system performance.
- It Contains the recently used instructions, data or both.
- The main aim is that the microprocessor unit access code and data in the cache most of time, instead from the main memory.