

Syllabus: UNIT- II

Overview of 8086 architecture, Internal Architecture- Register Organization, Memory Segmentation, Flag Register, Pin Configuration, Physical Memory Organization, Advanced Processors – 80286, 80386 and Pentium Processors.

OVERVIEW OF 8086 ARCHITECTURE**Features:**

- 8086 is an enhanced version of 8085 designed by INTEL in 1976.
- It is a 16 bit processor which has 16 data lines and 20 address lines.
- It provides up to 1MB storage.
- It consists of powerful instruction set which provides operations like multiplication and division easily.
- It supports two modes of operation.
 - ◆ Maximum mode: Suitable for system having multiple processors.
 - ◆ Minimum mode: Suitable for system having single processor.
- 8086 has an instruction queue which is capable of storing up to 6 instruction bytes from the memory which results in faster processing.
- It has a 16-bit ALU, 16-bit registers, internal data bus and 16-bit external data bus which allows faster processing.
- It is available in 3 versions based on the frequency of operation.
 1. 8086 : 5 MHz
 2. 8086-2 : 8 MHz
 3. 8086-1 : 10 MHz
- It uses two stages of pipelining, fetch stage and execute stage which improves performance.
- Fetch stage can prefetch up to 6 bytes of instructions and stores them in queue and execute stage executes these instructions.
- It has 256 vectored interrupts and consists of approximately 29,000 transistors.

Comparison between 8085 and 8086:

	8085	8086
1. Size (Data Bus)	8-Bit processor	16-Bit processor
2. Address Bus	16-Bits long.	20-Bits long.
3. Memory	64KB	Provides storage up to 1MB
4. Instruction Queue	No instruction queue. Only one instruction can be fetched at a time.	There is an instruction queue which can store up to 6 instruction bytes.
5. Pipelining	Does not use pipelining.	It uses two stages of pipelining, fetch stage and execute stage which improves performance
6. Cost	Low cost	High cost
7. Mode of operation	One mode	Min & Max mode

8086 ARCHITECTURE/BLOCK DIAGRAM:

Architecture is divided into two parts,

- Bus Interface Unit (BIU)
- Execution Unit (EU)

Bus Interface Unit:

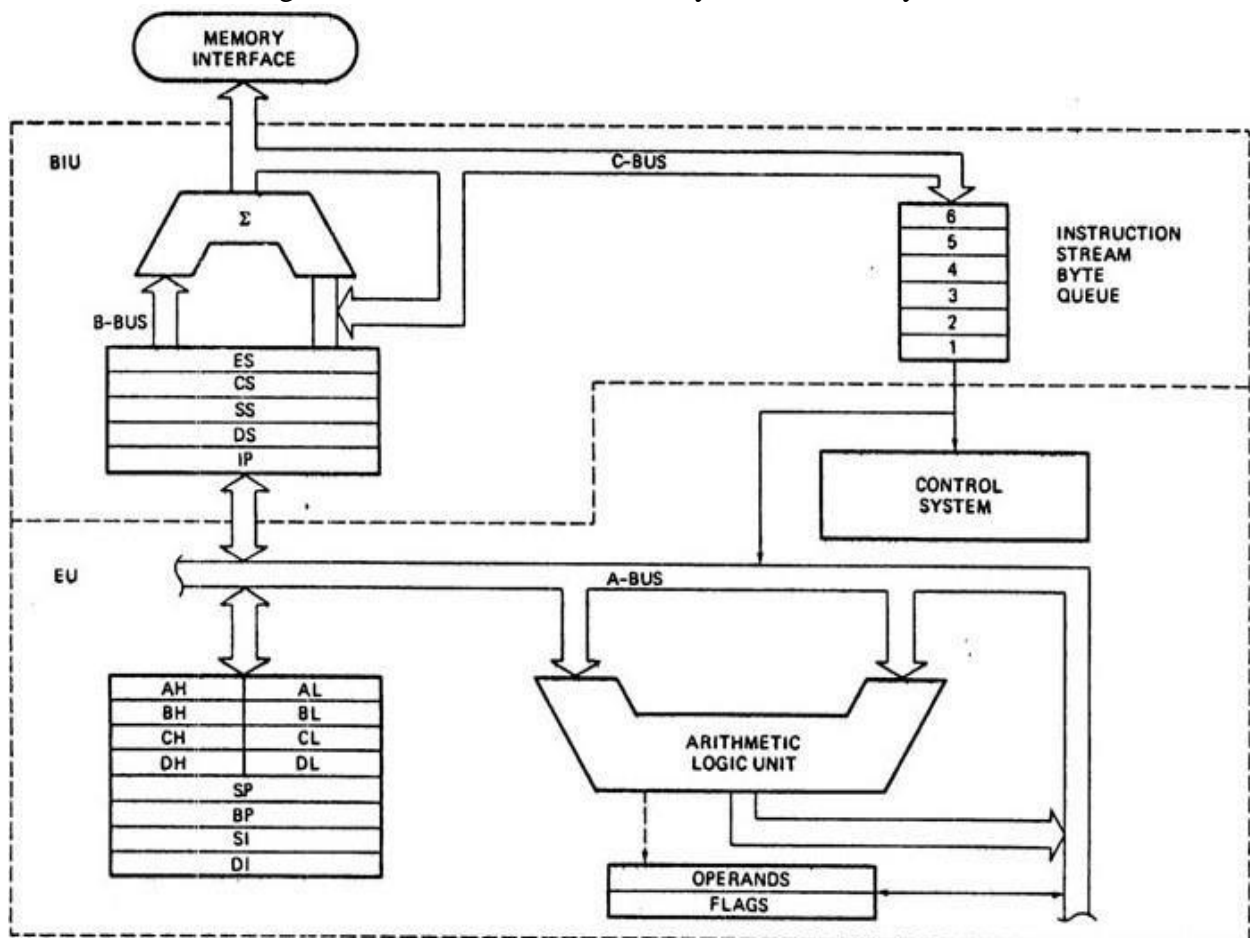
It handles all data and addresses on the buses for execution unit such as,

- Sending out addresses
- Fetch instructions from memory
- Read data from ports and memory
- Write data to ports and memory

BIU contains many functional units or parts.

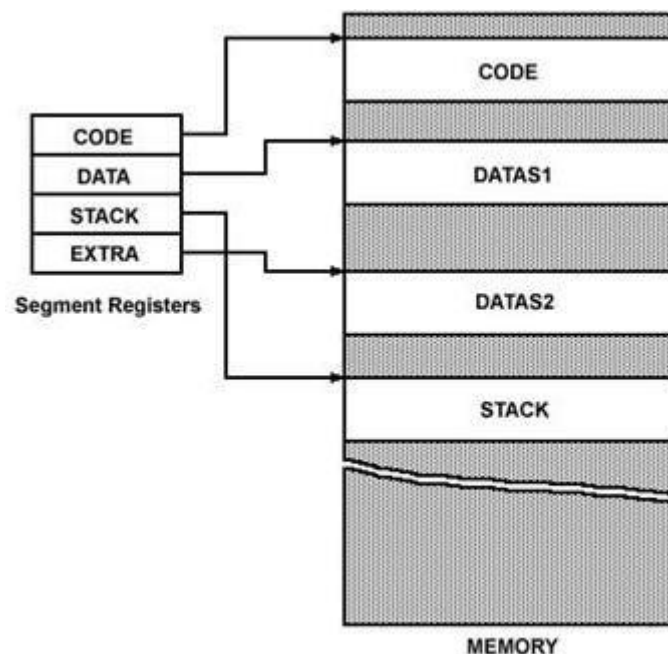
1. Instruction Queue:

- To increase the execution speed, BIU fetches up to 6 instruction bytes from the memory.
- The pre-fetched instruction bytes are held for execution unit in a first in first out group of register called instruction queue.
- When EU is ready for its next instruction, it reads the instruction from the queue.
- It is faster than fetching the instructions from the memory which includes sending address and fetch instruction byte from memory.



2. Segment Registers:

- BIU contains four 16-bit segment registers.
- They are Code segment (CS) register, Data segment (DS) register; Stack segment (SS) register and Extra segment (ES) register.
- These registers are used to hold the upper 16-bits of starting address for each of the segments.
- Part of a segment starting address stored in a segment register is called segment base.
- Code segment register is used for addressing a memory location in code segment in memory where executable program is stored.
- Data segment register is used for addressing a memory location in data segment where program data is stored.
- Stack segment register contains starting address of stack segment which stores program and data while a subprogram executes.
- Extra segment is called extra data segment used by some of the strings to hold extra destination data.



3. Instruction Pointer (IP):

Next register below the segment registers is the instruction pointer. It holds 16-bit address of next code byte within the code segment.

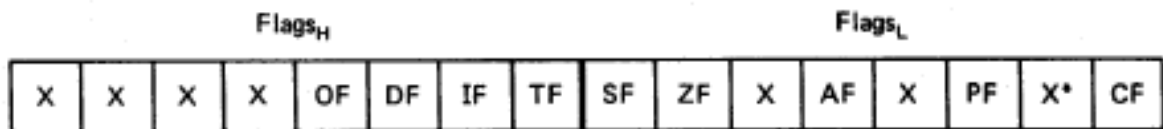
Execution Unit:

This unit tells the BIU where to fetch instructions or data from, decodes and executes instructions.

Functional units of EU are,

1. Control Circuitry: Used to perform various internal operations.
2. Instruction Decoder: Which translates instructions fetched from memory to generate different internal or external control signals required for performing operation.

3. ALU: It is a 16-bit register to perform arithmetic operations such as addition, subtraction and logical operations such as AND, OR, XOR, INC, DEC etc.
4. Flag Register: 16-bit flag register which indicates some condition produced by execution of an instruction or controls certain operations of EU.
 - Flags are modified automatically by CPU after mathematical operations.
 - It has 9 flags divided into 2 categories: Conditional flags and control flags.
 - Conditional flags are carry flag, zero flag, sign flag, auxiliary flag, parity flag and overflow flag.
 - Control flags are trap flag, interrupt flag and direction flag.



*Bits marked X are undefined.

5. General purpose registers:
 - EU has 8 general purpose register AH,AL, BH,BL, CH,CL,DH,DL.
 - They can be used individually for temporary storage of 8-bit data and also as register pairs as AX, BX, CX and DX to store 16-bit data.
 - AX: Accumulator, stores results of 16-bit operations.
 - BX: Used as base address. Holds starting base location of a memory region within a data segment.
 - CX: Defined as a counter. Mainly used in loop instructions to store loop count.
 - DX: Used as destination operand in few instructions and as implicit operand in few instructions.
6. Stack Pointer: Contains 16-bit offset from start of segment to memory location to memory location where a word is recently stored on stack. It is called top of the stack.
7. Other Pointers and Index registers:
 - EU contains a 16-bit Source Index (SI) register, Base Pointer (BP) register and Destination Index (DI) register.
 - They can be mainly used for temporary storage of 16-bit data like general purpose registers.

Note: Execution Unit receives program codes and data from BIU and executes these instructions and stores results in general registers.

EU has no connection to system buses; it receives and transmits all its data through BIU.

Overall Operation:

1. BIU sends contents of IP on to address bus.
2. IP is incremented by 1 to prepare for next instruction fetch.
3. Instruction is passed to Queue.
4. EU takes instructions from queue immediately and begins execution.
5. While EU is executing this instruction, BIU processed to fetch new instruction.
6. BIU fetches a new instruction whenever Queue has room for one or two additional bytes. This is called pipelining.

7. Advantage of Pipelined architecture is EU can execute instructions continuously instead of waiting for BIU to fetch a new instruction.

Note: EU enters into wait state under the following circumstances,

- Whenever instruction requires access to memory not Queue.
- When instruction to be executed is Jump
- When execution of instruction is slow.

REGISTER ORGANIZATION:

The 8086 has four groups of the user accessible internal registers.

· These are

- Instruction pointer(IP)
- Four General purpose registers (AX,BX,CX,DX)
- Four pointer registers (SP,BP,SI,DI)
- Four segment registers (CS,DS,SS,ES)
- Flag Register(FR)

The 8086 has a total of fourteen 16-bit registers including a 16 bit register called the *status register (flag 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 addressable memory these 4 segments are located the processor uses four segment registers:

EU Registers	AX	AH	AL	Accumulator
	BX	BH	BL	Base Register
	CX	CH	CL	Count Register
	DX	DH	DL	Data Register
		SP		Stack Pointer
		BP		Base Pointer
		SI		Source Index Register
		DI		Destination Index Register
	FR			Flag Register
BIU Registers	CS			Code Segment Register
	DS			Data Segment Register
	SS			Stack Segment Register
	ES			Extra Segment Register
	IP			Instruction Pointer

Segment Registers

1) **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.

2) **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.

3) **Data and Extra segment (DS and ES)** 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, and DX) and index register (SI, DI) is located in the data and Extra segment.

Data Registers

1) **AX (Accumulator)**- It 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.

2) **BX (Base register)** · It 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 offset for data segment.

3) **CX (Count register)** · It 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.
- 8086 has the LOOP instruction which is used for counter purpose when it is executed CX/CL is automatically decremented by 1.

EX MOV CL, 05H

START NOP

LOOP START (here CL is automatically decremented by 1 without DCR instruction)

4) **DX (Data register)**-It 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.

- DX 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.

Pointer registers

1. **Stack Pointer (SP)** is a 16-bit register is used to hold the offset address for stack segment.

2. **Base Pointer (BP)** is a 16-bit register is used to hold the offset address for stack segment.

i. BP register is usually used for based, based indexed or register indirect addressing.

ii. The difference between SP and BP is that the SP is used internally to store the address in case of interrupt and the CALL instruction.

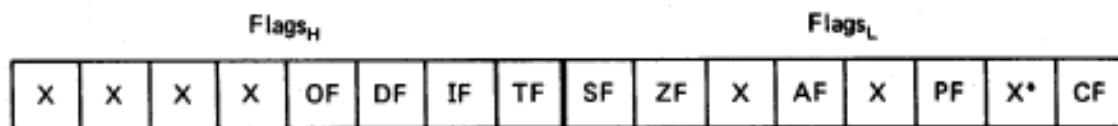
3. **Source Index (SI)** and **Destination Index (DI)** -These two 16-bit registers are used to hold the offset address for DS and ES in case of string manipulation instruction.

- SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.
- DI is used for indexed, based indexed and register indirect addressing, as well as a destination data addresses in string manipulation instructions.

4. Instruction Pointer (IP) - It is a 16-bit register. It acts as a program counter and is used to hold the offset address for CS.

5. Flag Register: 16-bit flag register which indicates some condition produced by execution of an instruction or controls certain operations of EU.

- Flags are modified automatically by CPU after mathematical operations.
- It has 9 flags divided into 2 categories: Conditional flags and control flags.
- Conditional flags are carry flag, zero flag, sign flag, auxiliary flag, parity flag and overflow flag.
- Control flags are trap flag, interrupt flag and direction flag.



*Bits marked X are undefined.

MEMORY SEGMENTATION:

Reason for Segmented Memory:

- 8086 has a 20-bit address bus. So it can address a maximum of 1MB of memory and each memory location is addressed by a 20 bit address.
- To hold a 20-bit address there must be a 20-bit address register available within processor but 8086 only has 16-bit registers. So 20-bit address can't be stored inside the 16-bit register. To avoid this problem segmented memory is used in 8086.

Total 1MB memory can be divided into some equal size segments each of having capacity 64KB. So max no of segments is 16. (1mb/64 kb=16) 8086 can work with only four 64KB segments at a time within this 1MB range.

Each location in a particular segment can be expressed by two addresses.

i) Segment Address (16 bit): It refers the starting address of a segment and it is fixed for whole of the segment.

ii) Offset or Displacement Address (16 bit): It refers the individual location in that segment and it is varied location wise.

By using these two addresses the 20 bit physical address can be calculated as below:

Physical address (20 bit) = [Segment Address (16 bit) * 10] H + Offset Address (16 bit)

According to this formula segment address is multiplied by 10 and is added to offset. This is equivalent to shifting of segment register content towards left 4 times so that four zero are

added to right side (MSB) of the segment address and added with the offset address to get the physical address which is 20 bit.

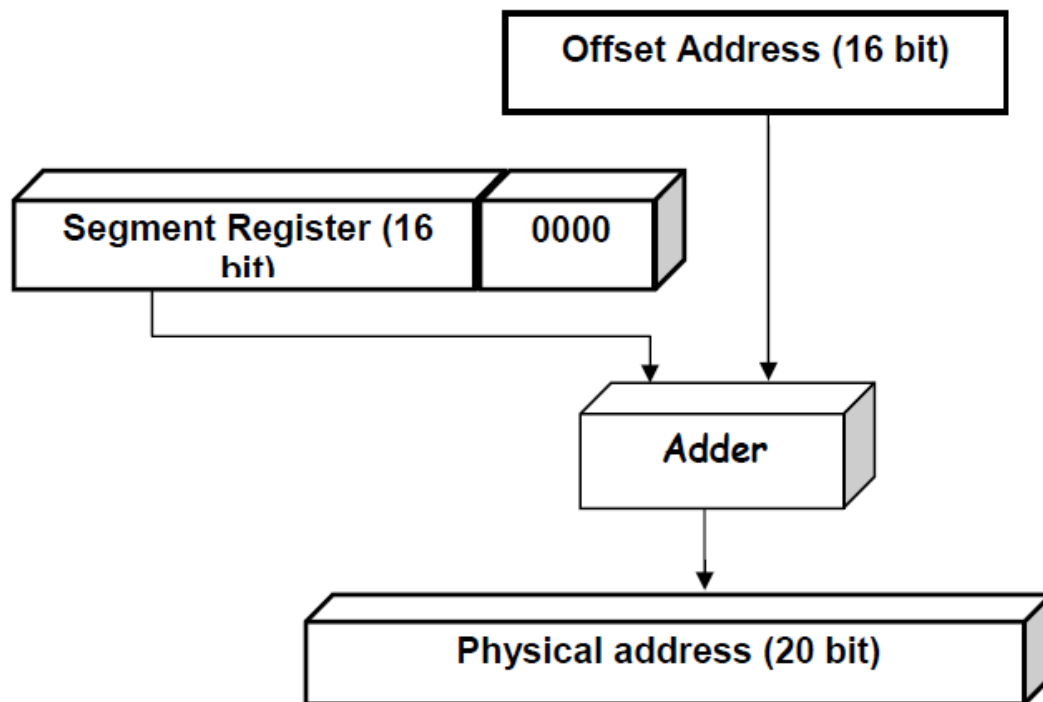


Fig: Physical address calculation

EX:-

Given Segment Address=3578H, Offset Address =6676H

$$\begin{aligned}
 \text{So Physical address} &= [\text{Segment Address} * 10]_{\text{H}} + \text{Offset Address} \\
 &= [3578 * 10]_{\text{H}} + 6676\text{H} \\
 &= 35780 + 6676 \\
 &= 3\text{BDF}6\text{H}
 \end{aligned}$$

Types of Segments

There are four types of memory segments

- Code segment (CS)
- Data segment (DS)
- Stack segment (SS)
- Extra segment (ES)

Code segment (CS): This segment is used to store code/program instructions.

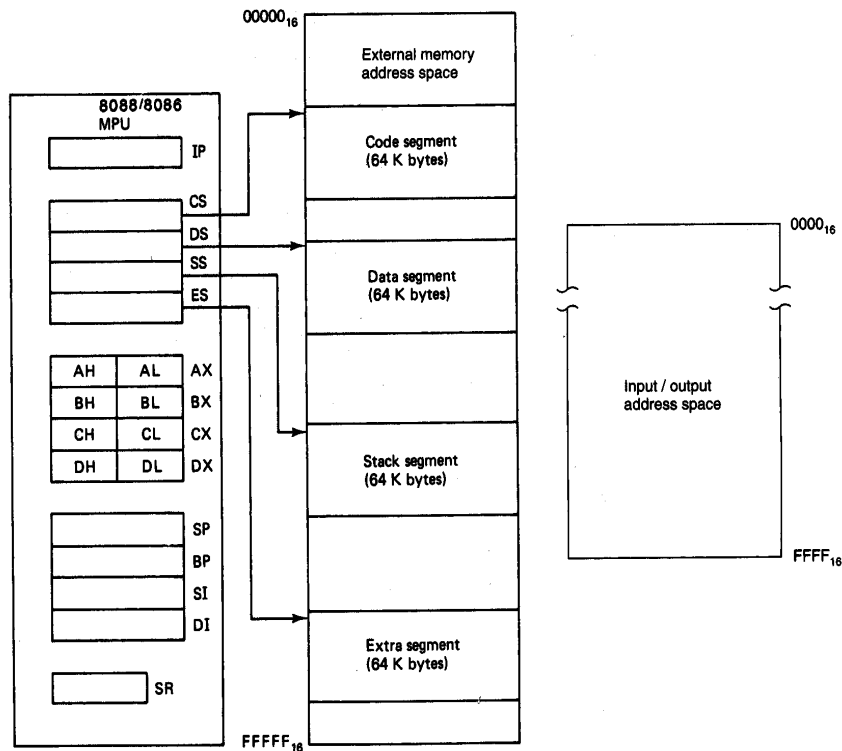
Data and Extra segment (DS&ES)- This segment is used to store data used in the program.

Stack segment (SS): This segment is used to store the stack contents.

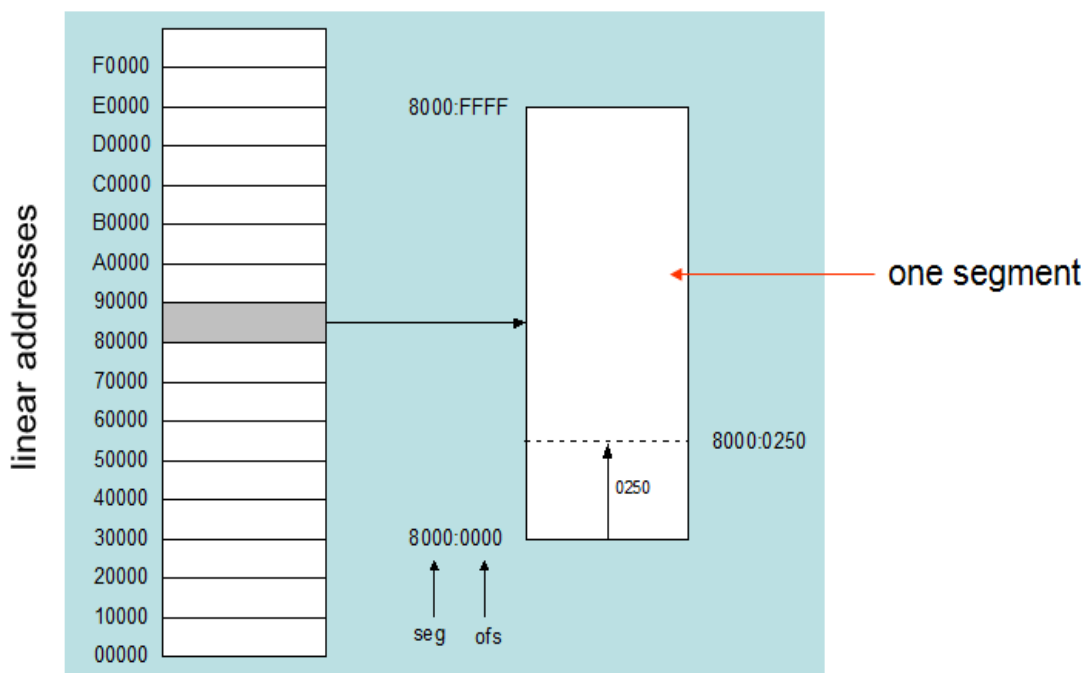
Types of Segments Registers

To hold the upper 16-bits of the starting address for each of the segments segment registers:

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



Offset may be content of IP, BX, SI, DI, SP, BP or an immediate 16-bit value based on addressing mode.

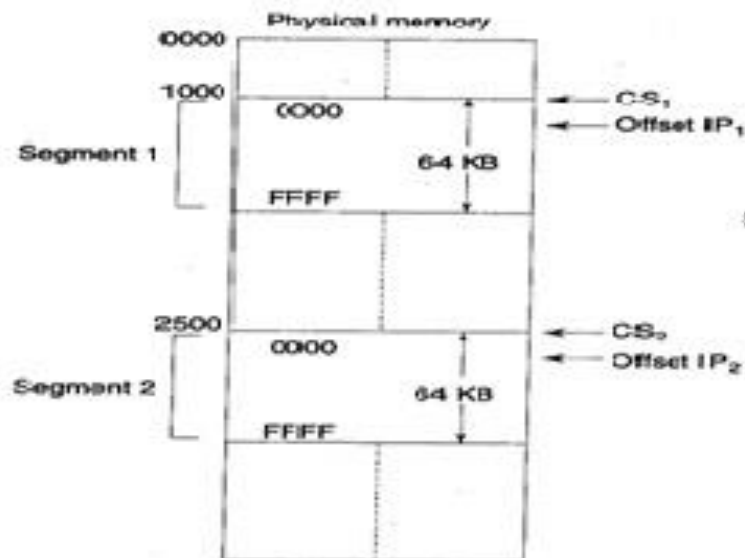


The address of segments may be assigned as 0000H to F000H respectively.

The offset address values are from 0000H to FFFFH.

The physical addresses range from 00000H to FFFFFH.

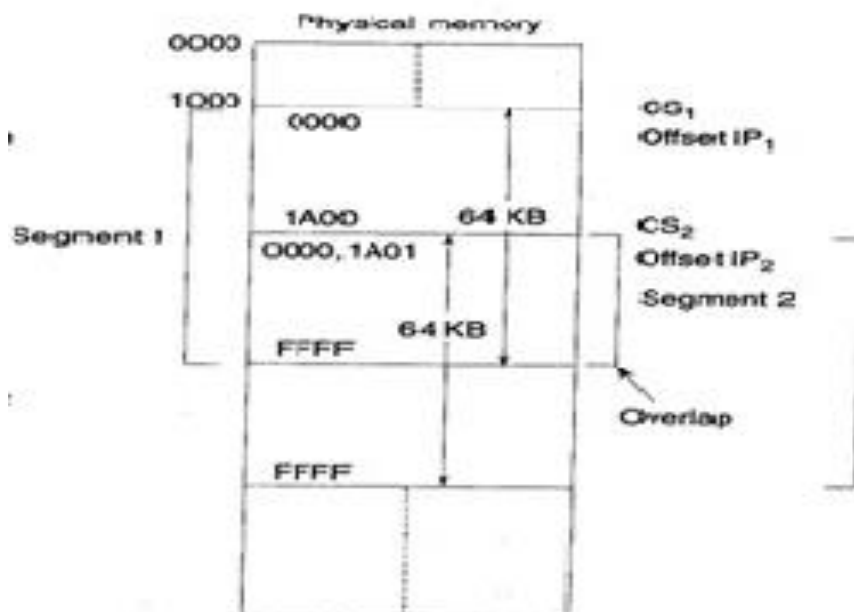
In this case, the segments are called Non-overlapping segments.



In some cases, the segments may be overlapping.

For example, a segment starts at a particular address and its maximum size can be 64KB.

If another segment starts before this 64KB locations first segment, two segments are said to be overlapping.



Advantages of segmented memory:

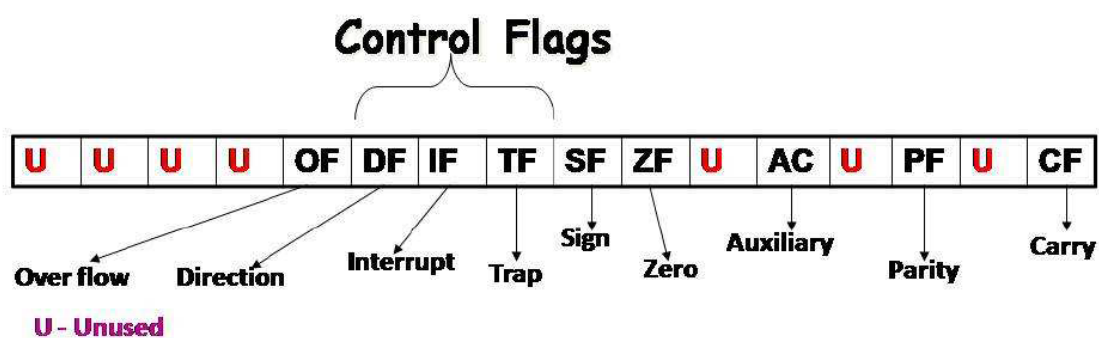
1. Allows placing of code, data and stack portions of same program in different parts of memory for data and code protection.
2. Permits a program and/or data to be placed in different areas of memory each time program is executed. Provision of relocation is done.
3. In overlapped area, physical address = $CS_1 + IP_1 = CS_2 + IP_2$.
+ indicates procedure for calculating physical address.

FLAG REGISTER:

16-bit flag register which indicates some condition produced by execution of an instruction or controls certain operations of EU.

- Flags are modified automatically by CPU after mathematical operations.
- It has 9 flags divided into 2 categories: Conditional flags and control flags.
- Conditional flags are carry flag, zero flag, sign flag, auxiliary flag, parity flag and overflow flag.
- Control flags are trap flag, interrupt flag and direction flag.

Flag Register



A flag is a 16-bit register containing 9 one-bit flags.

i. Overflow Flag (OF)-This flag is set if an overflow occurs. i.e. if the result of a signed operation is large enough to be accommodated in a destination register.

ii. Direction Flag (DF) – 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-decrementing mode.

iii. Interrupt-enable Flag (IF) – If this flag is set, the maskable interrupts are recognized by the CPU. Otherwise they are ignored. Setting this bit enables maskable interrupts.

iv. Single-step Flag (TRAP) (TF) – If this flag is set, the processor enters the single step execution mode. In other words, a trap interrupt is generated after execution of each instruction. The processor executes the current instruction and the control is transferred to the Trap interrupt service routine.

v. Sign Flag (SF) – This flag is set when the result of any computation is negative. For signed computations, the sign flag equals the MSB of the result.

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

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

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

ix. Carry Flag (CF) – This flag is set when there is a carry out of MSB in case of addition or a borrow in case of subtraction. For example. When two numbers are added, a carry may be generated out of the most significant bit position. The carry flag, in this case, will be set to '1'. In case, no carry is generated, it will be '0'.

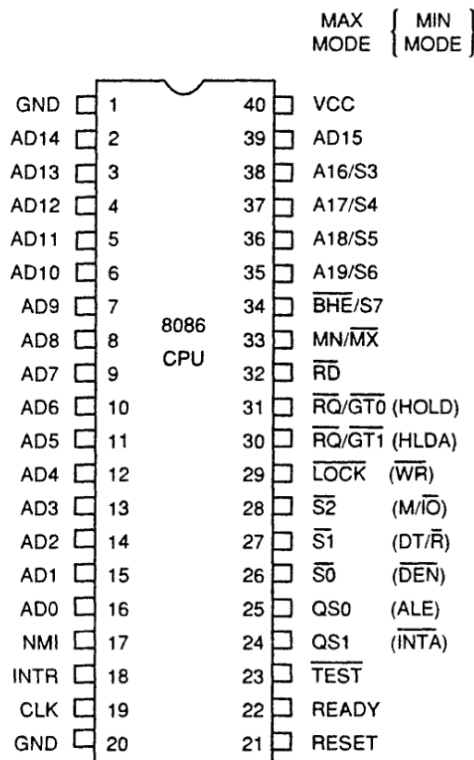
PIN CONFIGURATION:

8086 is a 16-bit processor available in 3 clock rates, in 40-pin plastic package.

8086 operates in single processor or multiprocessor configurations to achieve high performance.

8086 signals can be categorized into 3 groups.

1. Signals having common functions in minimum and maximum modes.
2. Signals which have special functions for minimum mode.
3. Signals having special functions for maximum mode.



1. Signals having common functions in minimum and maximum modes.

AD15-AD0:

Multiplexed address and data lines.

Address will be on the lines during T1 state and data will be available during T2, T3, Tw and T4.

T1, T2, T3, T4 are clock states of machine cycle and Tw is wait state.

A19/S6- A16/S3:

Multiplexed address and status lines.

During T1, these are most significant address lines for memory operations.

During I/O operations, these lines are low.

During memory or I/O operations, status information is available on these lines for T2, T3, Tw and T4.

S5 indicates the status of interrupt enable flag.

S4, S3 together indicate which segment register is presently used for memory access.

S4	S3	Indications
0	0	Alternate data
0	1	Stack segment
1	0	Code segment or none
1	1	Data segment

Status line S6 is always low.

Address bits are separated from status bits using latches controlled by ALE signal.

BHE/S7:

BHE is used to indicate the transfer of data over higher order data bus D15-D8.

BHE	A0	Indications
0	0	Whole word
0	1	Upper byte to/from odd address
1	0	Lower byte to/from even address.
1	1	None

- It goes low for data transfers over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals.
- BHE is low during T1 for read, write cycles whenever a byte is to be transferred on the higher byte of data bus.
- S7 is not currently used.

RD:

- Read signal, when low indicates peripherals that processor is performing memory or I/O read operation.
- RD is active low during T2, T3, Tw of any read cycle.

Ready:

- Acknowledge from slow devices or memory that they have completed the data transfer and ready for the next transfer.
- It is active high.

INTR: Interrupt Request

- A level triggered input. It is sampled during last clock cycle of any instruction to determine the availability of request.
- If any interrupt request is pending, the processor enters the interrupt acknowledge cycle.
- It is active high.

TEST:

- This input is examined by a wait instruction. If this goes low, execution will continue, otherwise processor remains in idle state.

NMI: Non-maskable interrupt.

- This is an edge triggered input which causes a type 2 interrupt.
- Not maskable internally by software. A transition from low to high initiates the interrupt response at the end of current instruction.

RESET: Causes the processor to terminate the current activity and start executing from FFFF0H. It is active high and must be active for at least 4 clock cycles.

CLK: Clock input

- Provides basic timing for processor operation and bus control activity.
- The clock signal is an asymmetric square wave of 33% duty cycle.
- Range of clock frequency for different versions of 8086 is 5-10 MHz.

VCC: +5V power supply for operation of internal circuit.

GND: Ground

MN/MX: Logic level at this pin indicates whether the processor is to operate in minimum or maximum mode.

2. Signals which have special functions for minimum mode.**M/I/O:** Memory/I/O

- This is a status line.
- If it is low, it indicates an I/O operation
- If it is high, it is a memory operation
- Becomes active in previous T4 and remains active till final T4 of current cycle.

INTA: Interrupt Acknowledgement

- It means the processor has accepted the interrupt.
- It is active low during T2, T3, Tw of each Interrupt Acknowledgement cycle.

ALE: Address Latch Enable

- Indicates the availability of valid address on address/data lines.
- It is active high.

DT/R: Data transmit/Receive

- This output signal is used to decide the direction of data flow through transceivers.
- When processor is sending out data, this pin is high
- When processor is receiving data, this is low

DEN: Data Enable

- Indicates the availability of valid data over address/data lines.
- It is used to enable transceivers to separate data from multiplexed address/data signal.
- It is active from middle of T2 to middle of T4.

HOLD, HLDA:

- When HOLD line goes high, it indicates to the processor that another device is requesting bus access.
- Processor after receiving HLOD request, issue HLDA signal in the middle of next clock cycle after completing the current bus cycle.

3. Signals having special functions for maximum mode**S2, S1, S0:** Status lines

- Indicate the type operation being carried out by the processor.
- These become active during T4 of previous cycle and remain active during T1, T2 of current bus cycle.

S2	S1	S0	Indication
0	0	0	Interrupt Ack
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	HALT
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive

LOCK:

- This output pin indicates that other processors in the system will be prevented from gaining the system bus while lock signal is low.
- The lock signal is activated by lock prefix instruction and remains active until completion of next instruction.

QS1, QS0: Queue Status

- These lines give information about status of code in pre-fetch queue.
- These are active during clock cycle after which queue operation is performed.

QS1	QS0	Indications
0	0	NOP
0	1	First byte of opcode from queue
1	0	Empty queue
1	1	Subsequent byte from queue

RQ/GT0, RQ/GT1: Request/Grant- These pins are used by other local bus masters in maximum mode to force the processor to release the bus at the end of processor's current bus cycle.

PHYSICAL MEMORY ORGANIZATION:

- In 8086, 1 MB memory is physically organized as an odd bank and an even bank each of 512 KB addressed in parallel by the processor.
- Byte data with an even address is transferred on D7-D0 and byte data with an even address is transferred on D15-D8 lines.
- The processor provides two enable signals BHE, A0 for selection of either even or odd or both the banks.

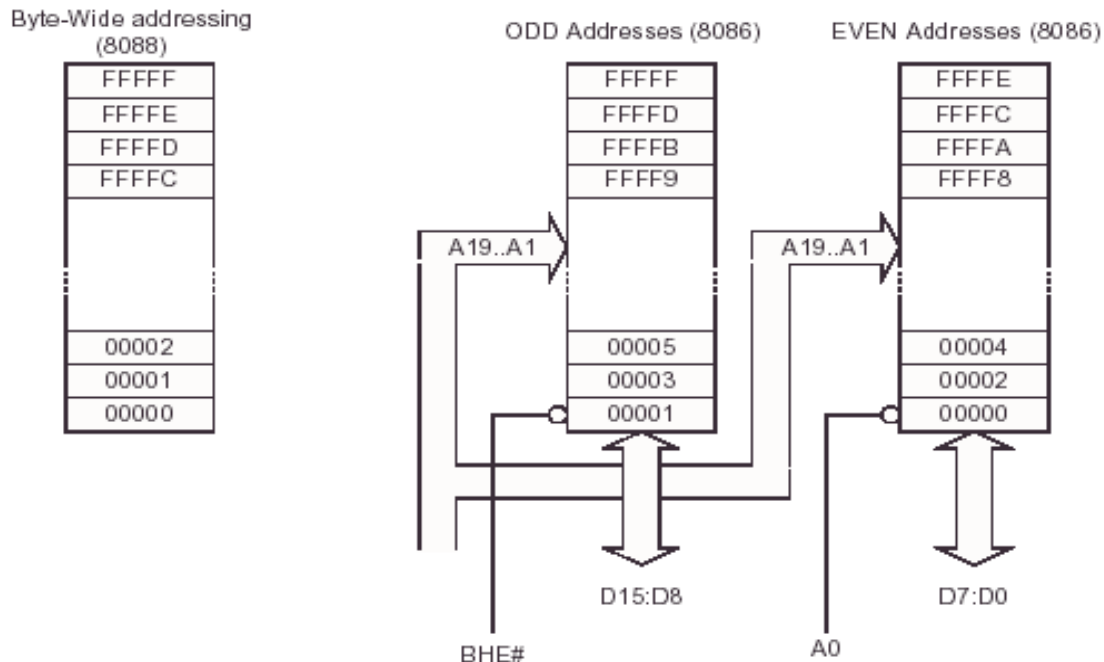
BHE	A0	Indications
0	0	Whole word
0	1	Upper byte to/from odd address
1	0	Lower byte to/from even address.
1	1	None

The instruction stream is fetched from memory as words and is addressed internally by the processor.

If processor fetches a word from memory, there are different possibilities,

1. Both bytes may be data operands
 2. Both bytes may contain opcode bits
 3. One of the bytes may be opcode and other may be data
- All the above possibilities are taken care of by internal decoder circuit which further derives signals to timing and control unit. The timing and control unit then derives all the signals required for execution of instruction.
 - BIU requires one or two memory cycles based on whether the starting byte is located at an even address or odd address. It is always better to locate the word data at an even address.
 - To read or write a complete word from or to memory, only one read or write cycle is required if it is located at an even address. If the word is located at an odd address, the first read or write cycle is required for accessing the lower byte while second read or write cycle is required for accessing upper byte.
 - 8086 is a 16-bit processor and hence can access two bytes of data in one memory or I/O read or write operation. But commercially available memory chips are only one byte size i.e they can store only one byte in a memory location. Therefore, to store 16-bit data, two successive memory locations are used. Both the bytes will be read or write in a single machine cycle.
 - 8086 memory starts at 00000H and ends at FFFFFH. As it is a 16-bit processor, it is expected to access 16-bit data to or from 8-bit memory chips in parallel. Bits D7-D0 of a 16-bit data will be transferred over D7-D0 of 16-bit data bus to or from 8-bit memory 2.
 - Bit D15-D8 of 16-bit data will be transferred over D15-D8 of 16-bit data bus to or from 8-bit memory 1. Therefore, to achieve 16-bit data transfer using 8-bit memories in parallel, the map of complete system memory address will be divided into two memory banks.
 - The lower byte of 16-bit data is stored at the first address of the map 00000H and is to be transferred over D7-D0 of processor bus, so, 00000H must be in 8-bit memory 2. Higher byte of 16-bit data is stored in next address 00001H and is to be transferred over D15-D8 of processor bus, so address 00001H must be in 8-bit memory 1.

- Similarly, for next 16-bit data stored in memory, the lower byte will be stored at 00002H in 8-bit memory 2 and higher byte at 00003H and must be in 8-bit memory 1.
- Therefore, if complete memory map of 8086 is filled with 16-bit data, all the lower bytes will be stored in 8-bit memory bank 2 and all the higher bytes will be stored in 8-bit memory bank 1.



Features of some advanced Processors - 80286 Microprocessor

Salient Features of 80286

- The 80286 is the first member of the family of advanced microprocessors with memory management and protection abilities.
- It is enhanced version of 8086 microprocessor
- Designed for multiuser and multitasking environments.
- It is 16-bit microprocessor, consists of 16 data lines.
- It has 24-bit address bus
- It was invented in February 1982 by Intel.
- Further in 1985, Intel produced upgraded version of 80286 which was a 32-bit microprocessor.
- It has non-multiplexed address and data bus that reduces operational speed.
- It has 4 processing units – Bus unit, Instruction unit, address unit and execution unit.
- The 80286 CPU, with its 24-bit address bus is able to address 16 Mbytes of physical memory and support virtual memory upto 1GB.
- Various versions of 80286 are available that runs on 12.5 MHz , 10 MHz and 8 MHz clock frequencies.
- 80286 is upwardly compatible with 8086 in terms of instruction set.
- 80286 has two operating modes namely real address mode and virtual address mode.
- In real address mode, the 80286 can address up to 1Mb of physical memory address like 8086.

- In virtual address mode, it can address up to 16 Mb of physical memory address space and 1 Gb of virtual memory address space.
- The instruction set of 80286 includes the instructions of 8086 and 80186.
- 80286 has some extra instructions to support operating system and memory management.
- In real address mode, the 80286 is object code compatible with 8086.
- In protected virtual address mode, it is source code compatible with 8086. The performance of 80286 is 5-6 times faster than the standard 8086.

Need for Memory Management The part of main memory in which the operating system and other system programs are stored is not accessible to the users. In view of this, an appropriate management of the memory system is required to ensure the smooth execution of the running process and also to ensure their protection. The memory management which is an important task of the operating system is supported by a hardware unit called memory management unit. Swapping in of the Program Fetching of the application program from the secondary memory and placing it in the physical memory for execution by the CPU.

Swapping out of the executable Program Saving a portion of the program or important results required for further execution back to the secondary memory to make the program memory free for further execution of another required portion of the program.

Concept of Virtual Memory Large application programs requiring memory much more than the physically available 16 Mbytes of memory, may be executed by dividing it into smaller segments. Thus, for the user, there exists a very large logical memory space which is not actually available. Thus, there exists a virtual memory which does not exist physically in a system. This complete process of virtual memory management is taken care of by the 80286 CPU and the supporting operating system.

80386 Microprocessor

The Intel386 DX Microprocessor is an entry-level 32-bit microprocessor designed for single-user applications and operating systems such as MS-DOS and Windows. The 32-bit registers and data paths support 32-bit addresses and data types. The processor addresses up to four gigabytes of physical memory and 64 terabytes (2^{46}) of virtual memory. The integrated memory management and protection architecture includes address translation registers, multitasking hardware and a protection mechanism to support operating systems. Instruction pipelining, on-chip address translation, ensure short average instruction execution times and maximum system throughput.

The Intel386 DX CPU offers new testability and debugging features. Testability features include a self-test and direct access to the page translation cache. Four new breakpoint registers provide breakpoint traps on code execution or data accesses, for powerful debugging of even ROM-based systems.

Object-code compatibility with all 8086 family members (8086, 8088, 80186, 80188, 80286) means the Intel386 DX offers immediate access to the world's largest microprocessor software base.

Salient Features of 80386

- ❖ Flexible 32-Bit Microprocessor
 - 8, 16, 32-Bit Data Types
 - 8 General Purpose 32-Bit Registers
- ❖ Very Large Address Space
 - 4 Gigabyte Physical
 - 64 Terabyte Virtual
 - 4 Gigabyte Maximum Segment Size
- ❖ Integrated Memory Management Unit
 - Virtual Memory Support
 - Optional On-Chip Paging
 - 4 Levels of Protection
 - Fully Compatible with 80286
- ❖ Object Code Compatible with All 8086 Family Microprocessors
- ❖ Virtual 8086 Mode Allows Running of 8086 Software in a Protected and Paged System
- ❖ Hardware Debugging Support
- ❖ Optimized for System Performance
 - Pipelined Instruction Execution
 - On-Chip Address Translation Caches
 - 20, 25 and 33 MHz Clock
 - 40, 50 and 66 Megabytes/Sec Bus Bandwidth
- ❖ Numerics Support via Intel387™ DX Math Coprocessor
- ❖ Complete System Development Support
 - Software: C, PL/M, Assembler
 - System Generation Tools
 - Debuggers: PSCOPE, ICETM-386
- ❖ High Speed CHMOS IV Technology
- ❖ 132 Pin Grid Array Package
- ❖ 132 Pin Plastic Quad Flat Package

PENTIUM MICROPROCESSOR

The Pentium family of processors originated from the 80486 microprocessor. The term "Pentium processor" refers to a family of microprocessors that share a common architecture and instruction set. The first Pentium processors were introduced in 1993. It runs at a clock frequency of either 60 or 66 MHz and has 3.1 million transistors.

Some of the features of Pentium architecture are:

- ❖ Complex Instruction Set Computer (CISC) architecture with Reduced Instruction Set Computer (RISC) performance.
- ❖ 64-Bit data bus and 36-bit address bus
- ❖ Upward code compatibility.
- ❖ Pentium processor uses Superscalar architecture and hence can issue multiple instructions per cycle.
- ❖ Multiple Instruction Issue (MII) capability.

- ❖ Pentium processor executes instructions in five stages. This staging, or pipelining, allows the processor to overlap multiple instructions so that it takes less time to execute two instructions in a row.
- ❖ The Pentium processor fetches the branch target instruction before it executes the branch instruction.
- ❖ The Pentium processor has two separate 8-kilobyte (KB) caches on chip, one for instructions and one for data. It allows the Pentium processor to fetch data and instructions from the cache simultaneously.
- ❖ When data is modified, only the data in the cache is changed. Memory data is changed only when the Pentium processor replaces the modified data in the cache with a different set of data
- ❖ The Pentium processor has been optimized to run critical instructions in fewer clock cycles than the 80486 processor.

The Pentium processor has two primary operating modes -

1. **Protected Mode** - In this mode all instructions and architectural features are available, providing the highest performance and capability. This is the recommended mode that all new applications and operating systems should target.
2. **Real-Address Mode** - This mode provides the programming environment of the Intel 8086 processor, with a few extensions. Reset initialization places the processor in real mode where, with a single instruction, it can switch to protected mode

The Pentium's basic integer pipeline is five stages long, with the stages broken down as follows:

1. **Pre-fetch/Fetch** : Instructions are fetched from the instruction cache and aligned in pre-fetch buffers for decoding.
2. **Decode1** : Instructions are decoded into the Pentium's internal instruction format. Branch prediction also takes place at this stage.
3. **Decode2** : Same as above, and microcode ROM kicks in here, if necessary. Also, address computations take place at this stage.
4. **Execute** : The integer hardware executes the instruction.
5. **Write-back** : The results of the computation are written back to the register file.

Floating Point Unit :

There are 8 general-purpose 80-bit Floating point registers. Floating point unit has 8 stages of pipelining. First five are similar to integer unit. Since the possibility of error is more in Floating Point unit (FPU) than in integer unit, additional error checking stage is there in FPU.