UNIT-IV

8086 CPU Pin Diagram: Special functions of general purpose registers, Segment register, concept of pipelining, 8086 Flag register, Addressing modes of 8086.

Introduction to processor:

A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer.

The term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor.

The **processor** (**CPU**, for Central Processing Unit) is the computer's brain. It allows the processing of numeric data, meaning information entered in binary form, and the execution of instructions stored in memory.

Evolution of Microprocessor:

A microprocessor is used as the CPU in a microcomputer. There are now many different microprocessors available.

Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions. Most Micro Processor are single- chip devices.

Microprocessor is a backbone of computer system. which is called CPU

Microprocessor speed depends on the processing speed depends on DATA BUS WIDTH.

A common way of categorizing microprocessors is by the no. of bits that their ALU can Work with at a time

The address bus is unidirectional because the address information is always given by the Micro Processor to address a memory location of an input / output devices.

The data bus is Bi-directional because the same bus is used for transfer of data between Micro Processor and memory or input / output devices in both the direction.

It has limitations on the size of data. Most Microprocessor does not support floating- point operations.

Microprocessor contain ROM chip because it contain instructions to execute data.

What is the primary & secondary storage device? - In primary storage device the Storage capacity is limited. It has a volatile memory. In secondary storage device the storage capacity is larger. It is a nonvolatile memory.

Primary devices are: RAM (Read / Write memory, High Speed, Volatile Memory) / ROM (Read only memory, Low Speed, Non Voliate Memory)

Secondary devices are: Floppy disc / Hard disk

Compiler:

Compiler is used to translate the high-level language program into machine code at a time. It doesn't require special instruction to store in a memory, it stores automatically. The Execution time is less compared to Interpreter.

1.4-bit Microprocessor:

The first **microprocessor** (Intel 4004) was invented in 1971. It was a 4-bit calculation device

with a speed of 108 kHz. Since then, microprocessor power has grown exponentially. So what exactly are these little pieces of silicone that run our computers(" Common Operating Machine Particularly Used For Trade Education And Research ")

- It has 3200 PMOS transistors.
- It is a 4-bit device used in calculator.

2.8-Bit microprocessor:

In 1972, Intel came out with the 8008 which is 8-bit.

In 1974, Intel announced the 8080 followed by 8085 is a 8-bit processor Because 8085 processor has 8 bit ALU (Arithmetic Logic Review). Similarly 8086 processor has 16 bit ALU. This had a larger instruction set then 8080. used NMOS transistors, so it operated much faster than the 8008.

The 8080 is referred to as a "Second generation Microprocessor"

Limitations of 8 Bit microprocessor:

- Low speed of execution
- Low memory addressing capability
- Limited number of general purpose registers
- Less power full instruction set

Examples for 4/ 8 / 16 / 32 bit Microprocessors:

4-Bit processor – 4004/4040

8-bit Processor - 8085 / Z80 / 6800

c) 16-bit Processor - 8086 / 68000 / Z8000

d) 32-bit Processor - 80386 / 80486

What are 1st / 2nd / 3rd / 4th generation processor?

1. The processor made of PMOS technology is called 1st generation processor, and it is made up of 4 bits
2. The processor made of NMOS technology is called 2nd generation processor, and
3. it is made up of 8 bits
4. The processor made of CMOS technology is called 3rd generation processor, and it is made up of 16 bits
5. The processor made of HCMOS technology is called 4th generation processor,
6. and it is made up of 32 bits (**HCMOS** : High-density n- type Complementary Metal Oxide Silicon field effect transistor)

8086 OVERVIEW

8086 Microprocessor is an enhanced version of 8085Microprocessor that was designed by Intel in 1976. It is a 16-bit Microprocessor having 20 address lines and16 data lines that 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, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for system having multiple processors and Minimum mode is suitable for system having a single processor.

The most prominent features of a 8086 microprocessor are as follows −

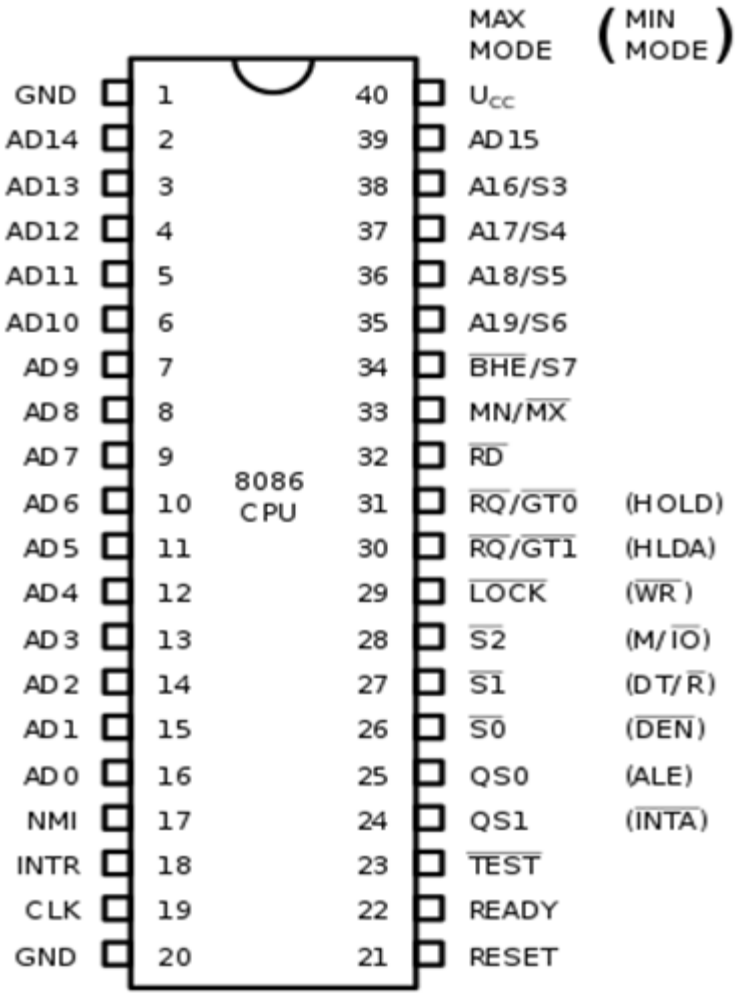
- It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
- It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It is available in 3 versions based on the frequency of operation −
  - 8086 → 5MHz
  - 8086-2 → 8MHz
  - (c)8086-1 → 10 MHz
- It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.
- Fetch stage can pre fetch up to 6 bytes of instructions and stores them in the queue.
- Execute stage executes these instructions.
- It has 256 vectored interrupts.
- It consists of 29,000 transistors.

Difference between 8085 and 8086 Microprocessor

085 Microprocessor	086 Microprocessor
is an 8-bit microprocessor.	is a 16-bit microprocessor.
has a 16-bit address line.	has a 20-bit address line.
has a 8-bit data bus.	has a 16-bit data bus.
he memory capacity is 64 KB.	he memory capacity is 1 MB.
he Clock speed of this icroprocessor is 3 MHz.	he Clock speed of this microprocessor varies between 5, 8 nd 10 MHz for different versions.
has five flags.	has nine flags.
085 microprocessor does not support emory segmentation.	086 microprocessor supports memory segmentation.
does not support pipelining.	supports pipelining.
is accumulator based processor.	is general purpose register based processor.
has no minimum or maximum mode.	has minimum and maximum modes.

ı 8085, only one processor is used.	ı 8086, more than one processor is used. An additional xternal processor can also be employed.
contains less number of transistors ɔmpare to 8086 microprocessor. It ɔntains about 6500 transistor.	contains more number of transistors compare to 8085 ıicroprocessor. It contains about 29000 in size.
he cost of 8085 is low.	he cost of 8086 is high.

8086 CPU Pin Diagram



Power supply and frequency signals

It uses 5V DC supply at V_{CC} pin 40, and uses ground at V_{SS} pin 1 and 20 for its operation.

Clock signal

Clock signal is provided through Pin-19. It provides timing to the processor for operations. Its frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.

Address/data bus

AD0-AD15. These are 16 address/data bus. AD0-AD7 carries low order byte data and AD8AD15 carries higher order byte data. During the first clock cycle, it carries 16-bit address and after that it carries 16-bit data.

Address/status bus

A16-A19/S3-S6. These are the 4 address/status buses. During the first clock cycle, it carries 4-bit address and later it carries status signals.

S7/BHE

BHE stands for Bus High Enable. It is available at pin 34 and used to indicate the transfer of data using data bus D8-D15. This signal is low during the first clock cycle, thereafter it is active.

Read(*RD*)

It is available at pin 32 and is used to read signal for Read operation.

Ready

It is available at pin 22. It is an acknowledgement signal from I/O devices that data is transferred. It is an active high signal. When it is high, it indicates that the device is ready to transfer data. When it is low, it indicates wait state.

RESET

It is available at pin 21 and is used to restart the execution. It causes the processor to immediately terminate its present activity. This signal is active high for the first 4 clock cycles to RESET the microprocessor.

INTR

It is available at pin 18. It is an interrupt request signal, which is sampled during the last clock cycle of each instruction to determine if the processor considered this as an interrupt or not.

NMI

It stands for non-maskable interrupt and is available at pin 17. It is an edge triggered input, which causes an interrupt request to the microprocessor.

TEST

This signal is like wait state and is available at pin 23. When this signal is high, then the processor has to wait for IDLE state, else the execution continues.

MN/*MX*

It stands for Minimum/Maximum and is available at pin 33. It indicates what mode the processor is to operate in; when it is high, it works in the minimum mode and vice-aversa.

INTA

It is an interrupt acknowledgement signal and id available at pin 24. When the microprocessor receives this signal, it acknowledges the interrupt.

ALE

It stands for address enable latch and is available at pin 25. A positive pulse is generated each time the processor begins any operation. This signal indicates the availability of a valid address on the address/data lines.

DEN

It stands for Data Enable and is available at pin 26. It is used to enable Transreceiver 8286. The transreceiver is a device used to separate data from the address/data bus.

DT/R

It stands for Data Transmit/Receive signal and is available at pin 27. It decides the direction of data flow through the transreceiver. When it is high, data is transmitted out and vice-a-versa.

M/IO

This signal is used to distinguish between memory and I/O operations. When it is high, it indicates I/O operation and when it is low indicates the memory operation. It is available at pin 28.

WR

It stands for write signal and is available at pin 29. It is used to write the data into the memory or the output device depending on the status of M/IO signal.

HLDA

It stands for Hold Acknowledgement signal and is available at pin 30. This signal acknowledges the HOLD signal.

HOLD

This signal indicates to the processor that external devices are requesting to access the address/data buses. It is available at pin 31.

QS_1 and QS_0

These are queue status signals and are available at pin 24 and 25. These signals provide the status of instruction queue. Their conditions are shown in the following table −

QS_0	QS_1	Status
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty the queue
1	1	Subsequent byte from the queue

S_0, S_1, S_2:These are the status signals that provide the status of operation, which is used by the Bus Controller 8288 to generate memory & I/O control signals. These are available at pin 26, 27, and 28. Following is the table showing their status −

S_2	S_1	S_0	Status
0	0	0	Interrupt acknowledgement
0	0	1	I/O Read

0	1	0	I/O Write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive

LOCK

When this signal is active, it indicates to the other processors not to ask the CPU to leave the system bus. It is activated using the LOCK prefix on any instruction and is available at pin 29.

RQ/GT_1 and RQ/GT_0

These are the Request/Grant signals used by the other processors requesting the CPU to release the system bus. When the signal is received by CPU, then it sends acknowledgment. RQ/GT_0 has a higher priority than RQ/GT_1.

Operating Modes of 8086

There are two operating modes of operation for Intel 8086, namely the **minimum mode** and the **maximum mode**.

When only one 8086 CPU is to be used in a microprocessor system, the 8086 is used in the **Minimum mode** of operation.

In a multiprocessor system 8086 operates in the **Maximum mode**.

Pin Description for Minimum Mode

In this minimum mode of operation, the pin MN/$\overline{MX}$ is connected to 5V D.C. supply i.e. MN/$\overline{MX}$ = VCC.

The description about the pins from 24 to 31 for the minimum mode is as follows:

$\overline{INTA}$ (Output): Pin number 24 interrupts acknowledgement. On receiving interrupt signal, the processor issues an interrupt acknowledgment signal. It is active LOW.

ALE (Output): Pin no. 25. Address latch enable. It goes HIGH during T1. The microprocessor 8086 sends this signal to latch the address into the Intel 8282/8283 latch.

$\overline{DEN}$ (Output): Pin no. 26. Data Enable. When Intel 8287/8286 octal bus transceiver is used this signal. It is active LOW.

DT/R (output): Pin No. 27 data Transmit/Receives. When Intel 8287/8286 octal bus transceiver is used this signal controls the direction of data flow through the transceiver. When it is HIGH, data is sent out. When it is LOW, data is received.
M/IO (Output): Pin no. 28, Memory or I/O access. When this signal is HIGH, the CPU wants to access memory. When this signal is LOW, the CPU wants to access I/O device.
WR (Output): Pin no. 29, Write. When this signal is LOW, the CPU performs memory or I/O write operation.
HLDA (Output): Pin no. 30, Hold Acknowledgment. It is sent by the processor when it receives HOLD signal. It is active HIGH signal. When HOLD is removed HLDA goes LOW.
HOLD (Input): Pin no. 31, Hold. When another device in microcomputer system wants to use the address and data bus, it sends HOLD request to CPU through this pin. It is an active HIGH signal.

Pin Description for Maximum Mode

In the maximum mode of operation, the pin MN/¯MX is made LOW. It is grounded. The description about the pins from 24 to 31 is as follows:
QS1, QS0 (Output): Pin numbers 24, 25, Instruction Queue Status. Logics are given below:

QS1	QS0	Operation
0 0	00	No operation
0 0	11	1st byte of opcode from queue.
11	00	Empty the queue
11	11	Subsequent byte from queue

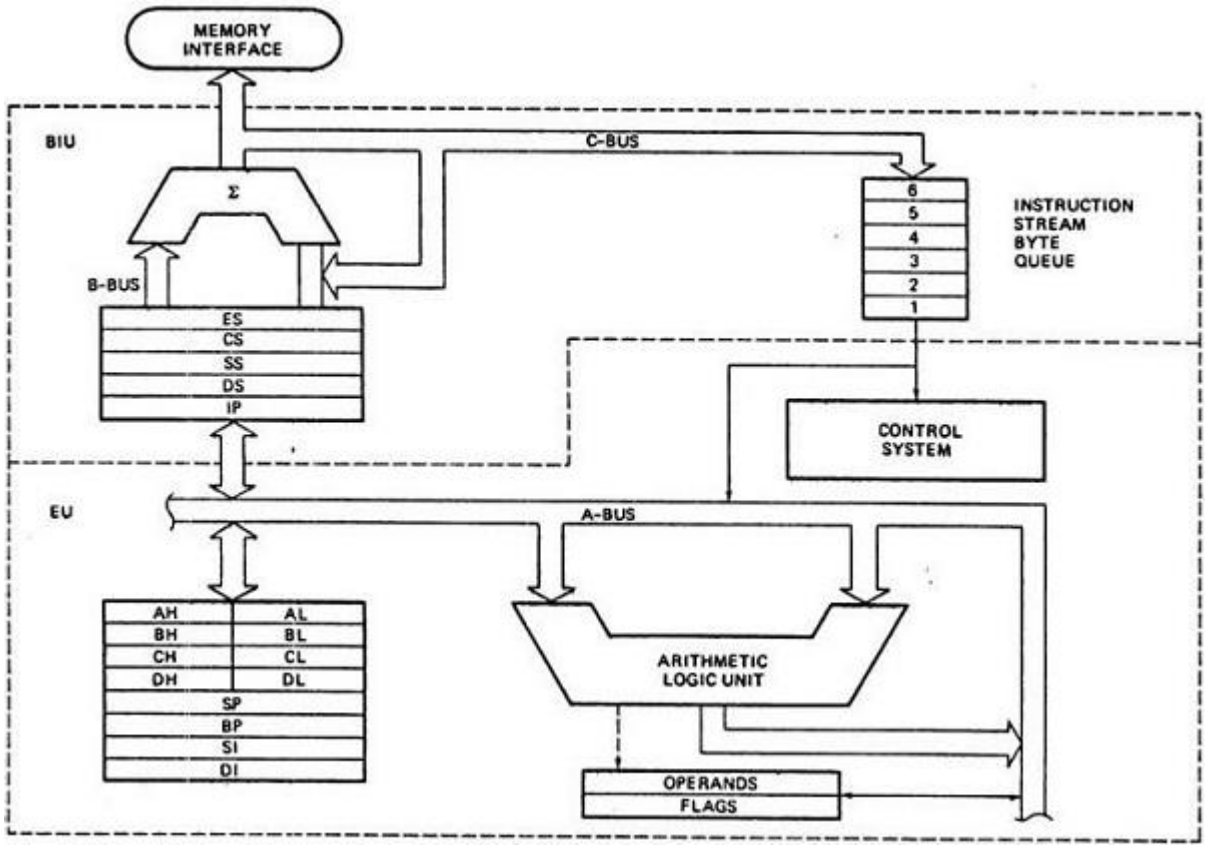
S0, S1, S2 (Output): Pin numbers 26, 27, 28 Status Signals. These signals are connected to the bus controller of Intel 8288. This bus controller generates memory and I/O access control signals. Logics for status signal are given below:

S2	S1	S0	Operation
00	00	00	Interrupt acknowledgement

00	00	11	Read data from I/O port
00	11	00	Write data from I/O port
00	11	11	Halt
11	00	00	Opcode fetch
11	00	11	Memory read
11	11	00	Memory write
1	1	1	Passive state

LOCK (Output): Pin no. 29. It is an active LOW signal. When this signal is LOW, all interrupts are masked and no HOLD request is granted. In a multiprocessor system all other processors are informed through this signal that they should not ask the CPU for relinquishing the bus control. RG/GT1, RQ/GT0 (Bidirectional): Pin numbers 30, 31, Local Bus Priority Control. Other processors ask the CPU by these lines to release the local bus.
In the maximum mode of operation signals WR, ALE, DEN, DT/R etc. are not available directly from the processor. These signals are available from the controller 8288.

Architecture of 8086



8086 Microprocessor is divided into two functional units, i.e., **EU** (Execution Unit) and **BIU** (Bus Interface Unit).

EU (Execution Unit)

Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions. Its function is to control operations on data using the instruction decoder & ALU. EU has no direct connection with system buses as shown in the above figure, it performs operations over data through BIU.
Let us now discuss the functional parts of 8086 microprocessors.

ALU

It handles all arithmetic and logical operations, like +, −, ×, /, OR, AND, NOT operations.

Flag Register

It is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups − Conditional Flags and Control Flags.

Conditional Flags

It represents the result of the last arithmetic or logical instruction executed. Following is the list of conditional flags −

- **Carry flag −** This flag indicates an overflow condition for arithmetic operations.
- **Auxiliary flag −** When an operation is performed at ALU, it results in a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), then this flag is set, i.e. carry given by D3 bit to D4 is AF flag. The processor uses this flag to perform binary to BCD conversion.
- **Parity flag −** This flag is used to indicate the parity of the result, i.e. when the lower order 8-bits of the result contains even number of 1's, then the Parity Flag is set. For odd number of 1's, the Parity Flag is reset.
- **Zero flag −** This flag is set to 1 when the result of arithmetic or logical operation is zero else it is set to 0.
- **Sign flag −** This flag holds the sign of the result, i.e. when the result of the operation is negative, then the sign flag is set to 1 else set to 0.
- **Overflow flag −** This flag represents the result when the system capacity is exceeded.

Control Flags

Control flags controls the operations of the execution unit. Following is the list of control flags −

- **Trap flag −** It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in a single step mode.
- **Interrupt flag −** It is an interrupt enable/disable flag, i.e. used to allow/prohibit the interruption of a program. It is set to 1 for interrupt enabled condition and set to 0 for interrupt disabled condition.
- **Direction flag −** It is used in string operation. As the name suggests when it is set then string bytes are accessed from the higher memory address to the lower memory address and vice-a-versa.

General purpose register

There are 8 general purpose registers, i.e., AH, AL, BH, BL, CH, CL, DH, and DL. These registers can be used individually to store 8-bit data and can be used in pairs to store 16bit data. The valid register pairs are AH and AL, BH and BL, CH and CL, and DH and DL. It is referred to the AX, BX, CX, and DX respectively.

- **AX register −** It is also known as accumulator register. It is used to store operands for arithmetic operations.
- **BX register −** It is used as a base register. It is used to store the starting base address of the memory area within the data segment.
- **CX register −** It is referred to as counter. It is used in loop instruction to store the loop counter.
- **DX register −** This register is used to hold I/O port address for I/O instruction.

Stack pointer register

It is a 16-bit register, which holds the address from the start of the segment to the memory location, where a word was most recently stored on the stack.

BIU (Bus Interface Unit)

BIU takes care of all data and addresses transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as

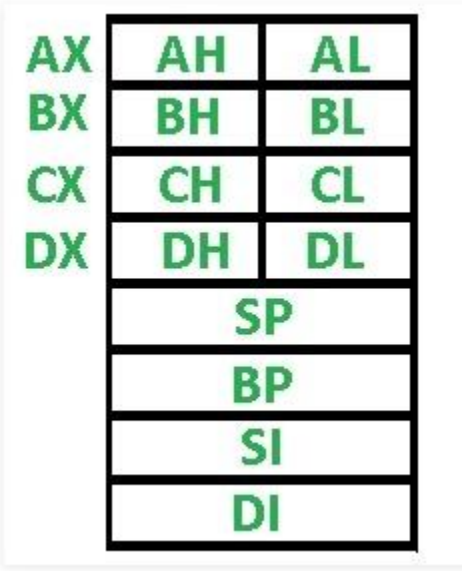
writing data to the ports and the memory. EU has no direction connection with System Buses so this is possible with the BIU. EU and BIU are connected with the Internal Bus.

It has the following functional parts −

- **Instruction queue** − BIU contains the instruction queue. BIU gets up to 6 bytes of next instructions and stores them in the instruction queue. When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called **pipelining**.
- **Segment register** − BIU has 4 segment buses, i.e. CS, DS, SS& ES. It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations. It also contains 1 pointer register IP, which holds the address of the next instruction to executed by the EU.
  - **CS** − It stands for Code Segment. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.
  - **DS** − It stands for Data Segment. It consists of data used by the program and is accessed in the data segment by an offset address or the content of other register that holds the offset address.
  - **SS** − It stands for Stack Segment. It handles memory to store data and addresses during execution.
  - **ES** − It stands for Extra Segment. ES is additional data segment, which is used by the string to hold the extra destination data.
- **Instruction pointer** − It is a 16-bit register used to hold the address of the next instruction to be executed.

Special functions of general purpose registers

he general purpose registers are used to store temporary data in the time of different operations in microprocessor. 8086 has eight general purpose registers.



The description of these general purpose registers

Register	Function
AX	This is the accumulator. It is 16-bit registers, but it is divided into two 8-bit registers. These registers are AH and AL. AX generally used for arithmetic or logical instructions, but it is not mandatory in 8086.
BX	BX is another register pair consisting of BH and BL. This register is used to store the offset values.
CX	CX is generally used as control register. It has two parts CH and CL. For different looping and counting purposes these are used.
DX	DX is data register. The two parts are DH and DL. This register can be used in Multiplication, Input/output addressing etc.
SP	This is the stack pointer. The stack pointer points the top most element of the stack. For empty stack SP will be at position FFFEH.
BP	BP is another 16-bit register. This is base pointer register. This register is primary used in accessing the parameters passed by the stack. It's offset address relatives to stack segment.
SI	This is Source Index register. This is used to point the source in some string related operations. Its offset is relative to data segment.
DI	This is destination index register. This is used to point destination in some string related operations. Its offset is relative to extra segment.

Segmentation is the process in which the main memory of the computer is divided into different segments and each segment has its own base address. It is basically used to enhance the speed of execution of the computer system, so that processor is able to fetch and execute the data from the memory easily and fast.

Need for Segmentation –

The Bus Interface Unit (BIU) contains four 16 bit special purpose registers (mentioned below) called as Segment Registers.

Code segment register (CS): is used for addressing memory location in the code segment of the memory, where the executable program is stored.

Data segment register (DS): points to the data segment of the memory where the data is stored.

Extra Segment Register (ES): also refers to a segment in the memory which is another data

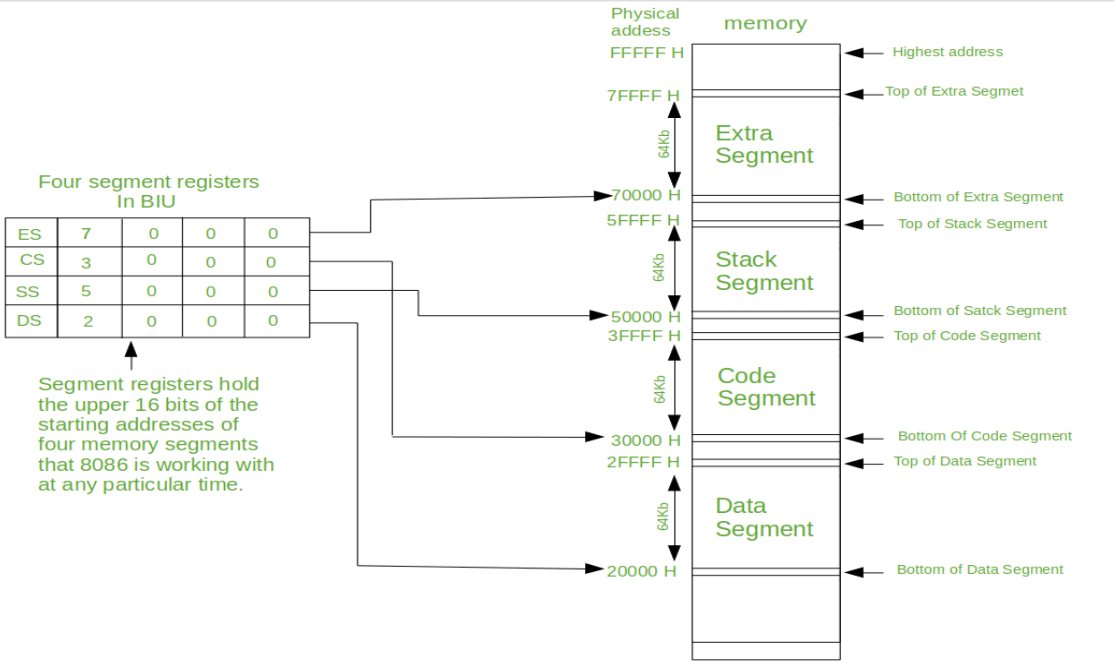
segment in the memory.

Stack Segment Register (SS): is used for addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data.

The number of address lines in 8086 is 20, 8086 BIU will send 20bit address, so as to access one of the 1MB memory locations. The four segment registers actually contain the upper 16 bits of the starting addresses of the four memory segments of 64 KB each with which the 8086 is working at that instant of time. A segment is a logical unit of memory that may be up to 64 kilobytes long. Each segment is made up of contiguous memory locations. It is independent, separately addressable unit. Starting address will always be changing. It will not be fixed.

Note that the 8086 does not work the whole 1MB memory at any given time. However it works only with four 64KB segments within the whole 1MB memor

Bellow is the one way of positioning four 64 kilobyte segments within the 1M byte memory space of an 8086.



Types Of Segmentation –

Overlapping Segment – A segment starts at a particular address and its maximum size can go up to 64kilobytes. But if another segment starts along this 64kilobytes location of the first segment, then the two are said to be *Overlapping Segment*.

Non-Overlapped Segment – A segment starts at a particular address and its maximum size can go up to 64kilobytes. But if another segment starts before this 64kilobytes location of the first segment, then the two segments are said to be *Non-Overlapped Segment*.

Advantages of the Segmentation The main advantages of segmentation are as follows:

It provides a powerful memory management mechanism.

Data related or stack related operations can be performed in different segments.

Code related operation can be done in separate code segments.

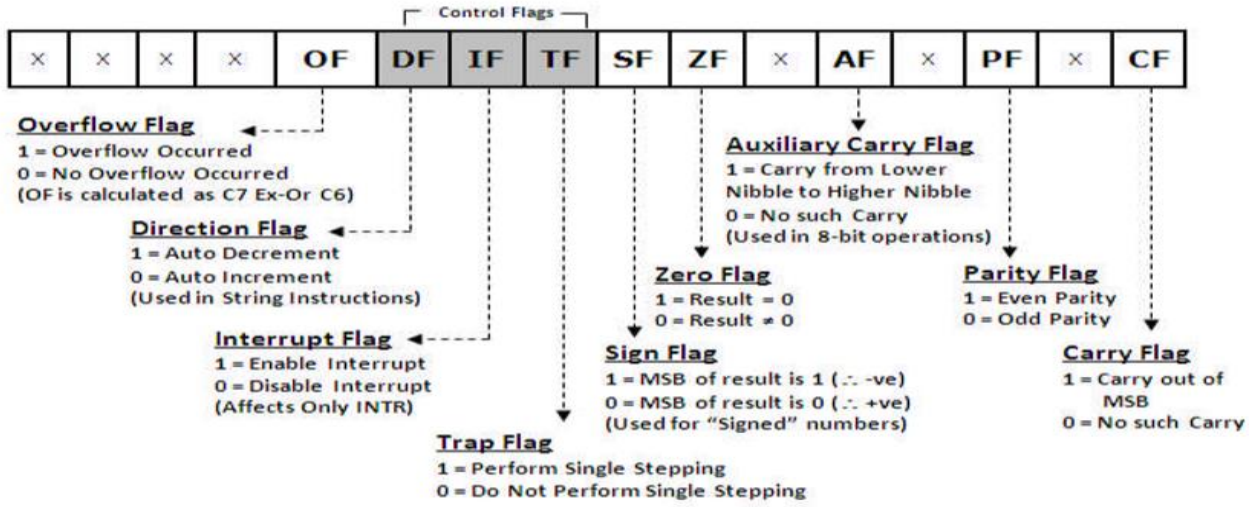
It allows to processes to easily share data.
It allows to extend the address ability of the processor, i.e. segmentation allows the use of 16 bit registers to give an addressing capability of 1 Megabytes. Without segmentation, it would require 20 bit registers.
It is possible to enhance the memory size of code data or stack segments beyond 64 KB by allotting more than one segment for each area.

concept of pipelining

The process of fetching the next instruction when the present instruction is being executed is called as pipelining. Pipelining has become possible due to the use of queue.BIU (Bus Interfacing Unit) fills in the queue until the entire queue is full.BIU restarts filling in the queue when at least two locations of queue are vacant. Advantages of pipelining: The execution unit always reads the next instruction byte from the queue in BIU. This is faster than sending out an address to the memory and waiting for the next instruction byte to come. In short pipelining eliminates the waiting time of EU and speeds up the processing. -The 8086 BIU will not initiate a fetch unless and until there are two empty bytes in its queue. 8086 BIU normally obtains two instruction bytes per fetch.

8086 Flag register

The Flag register is a Special Purpose Register. Depending upon the value of result after any arithmetic and logical operation the flag bits become set (1) or reset (0)



We can divide the flag bits into two sections. The Status Flags, and the Control Flags.

Status Flags

In 8086 there are 6 different flags which are set or reset after 8-bit or 16-bit operations. These flags and their functions are listed below.

Flag Bit	Function

Flag Bit	Function
S	After any operation if the MSB is 1, then it indicates that the number is negative. And this flag is set to 1
Z	If the total register is zero, then only the Z flag is set
AC	When some arithmetic operations generates carry after the lower half and sends it to upper half, the AC will be 1
P	This is even parity flag. When result has even number of 1, it will be set to 1, otherwise 0 for odd number of 1s
CY	This is carry bit. If some operations are generating carry after the operation this flag is set to 1
O	The overflow flag is set to 1 when the result of a signed operation is too large to fit.

Control Flags

In 8086 there are 3 different flags which are used to enable or disable some basic operations of the microprocessor. These flags and their functions are listed below.

Flag Bit	Function
D	This is directional flag. This is used in string related operations. D = 1, then the string will be accessed from higher memory address to lower memory address, and if D = 0, it will do the reverse.
I	This is interrupt flag. If I = 1, then MPU will recognize the interrupts from peripherals. For I = 0, the interrupts will be ignored
T	This trap flag is used for on-chip debugging. When T = 1, it will work in a single step mode. After each instruction, one internal interrupt is generated. It helps to execute some program instruction by instruction.

Addressing Modes.

The way of specifying data to be operated by an instruction is known as **addressing modes**. This specifies that the given data is an immediate data or an address. It also specifies whether the given operand is register or register pair.

Types of addressing modes:

Register mode – In this type of addressing mode both the operands are registers.

Example:

```
MOV AX, BX
XOR AX, DX
ADD AL, BL
```

Immediate mode – In this type of addressing mode the source operand is a 8 bit or 16 bit data. Destination operand can never be immediate data.

Example:

```
MOV AX, 2000
MOV CL, 0A
ADD AL, 45
AND AX, 0000
```

Displacement or direct mode – In this type of addressing mode the effective address is directly given in the instruction as displacement.

Example:

```
MOV AX, [DISP]
MOV AX, [0500]
```

Register indirect mode – In this addressing mode the effective address is in SI, DI or BX.

Example:

```
MOV AX, [DI]
ADD AL, [BX]
MOV AX, [SI]
```

Based indexed mode – In this the effective address is sum of base register and index register.

```
Base register: BX, BP
Index register: SI, DI
```

The physical memory address is calculated according to the base register.

Example:

```
MOV AL, [BP+SI]
MOV AX, [BX+DI]
```

Indexed mode – In this type of addressing mode the effective address is sum of index register and displacement.

Example:

```
MOV AX, [SI+2000]
MOV AL, [DI+3000]
```

Based mode – In this the effective address is the sum of base register and displacement.

Example:

```
MOV AL, [BP+ 0100]
```

Based indexed displacement mode – In this type of addressing mode the effective address is the sum of index register, base register and displacement.

Example:

```
MOV AL, [SI+BP+2000]
```

String mode – This addressing mode is related to string instructions. In this the value of SI and DI are auto incremented and decremented depending upon the value of directional flag.

Example:

```
MOVS B
MOVS W
```

Input/Output mode – This addressing mode is related with input output operations.

Example:

```
IN A, 45
OUT A, 50
```

Relative mode –

In this the effective address is calculated with reference to instruction pointer.

Example:

```
JNZ 8 bit address
IP=IP+8 bit address
```