Programming model of 8086: The programming model of a processor deals with internal registers, status and control flags, number of address lines ,number of data lines and the input/output port addresses which are needed by the programmer to write the programs.

How can a 20-bit address be obtained, if there are only 16-bit registers? However, the largest register is only 16 bits (64k); so physical addresses have to be calculated.

These calculations are done in hardware within the microprocessor. The 16-bit contents of segment register gives the starting/ base address of particular segment. To address a specific memory location within a segment we need an offset address. The offset address is also 16-bit wide and it is provided by one of theassociated pointer or index register.

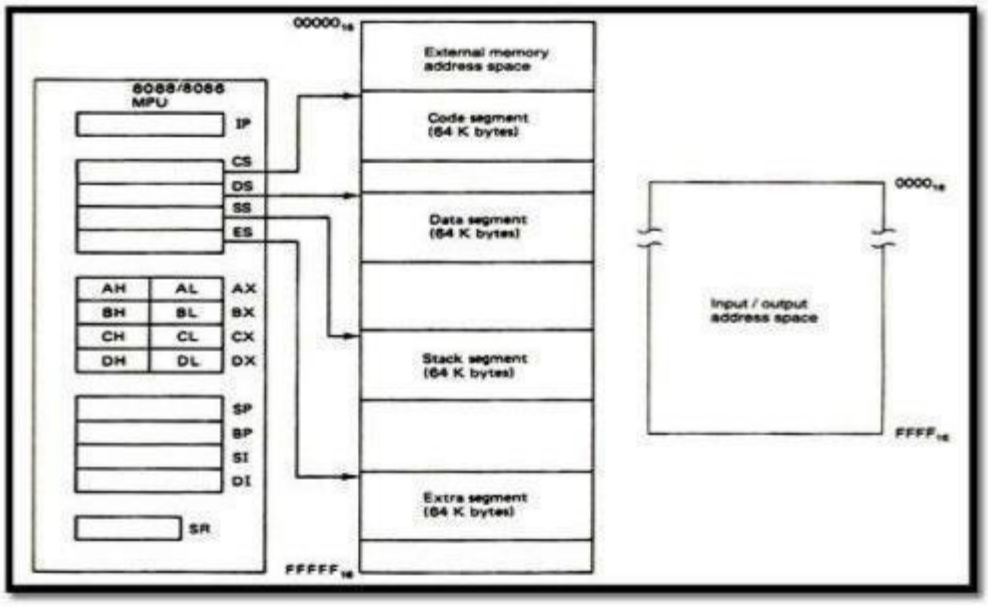
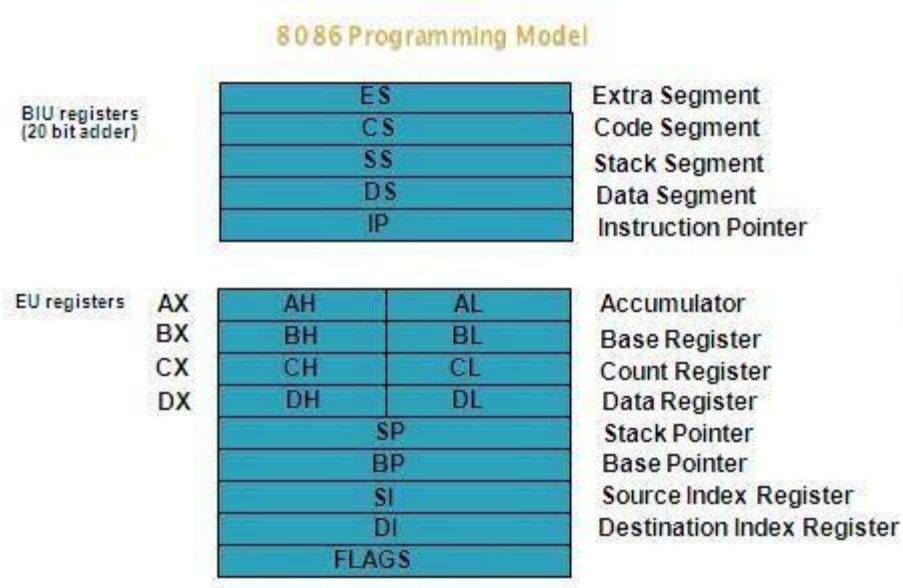


Figure: Software model of 8086 microprocessor

To be able to program a microprocessor, one does not need to know all of its hardware architectural features. What is important to the programmer is being aware of the various registers within the device and to understand their purpose, functions, operating capabilities, andlimitations.

The above figure illustrates the software architecture of the 8086 microprocessor. From this diagram, we see that it includes fourteen l6-bit internal registers: the instruction pointer (IP), four data registers (AX, BX, CX, and DX), two pointer registers (BP and SP), two index registers (SI and DI), four segment registers (CS, DS, SS, and ES) and status register (SR), with nine of its bits implemented as status and control flags.



The point to note is that the beginning segment address must begin at an address divisible by 16.Also note that the four segments need not be defined separately. It is allowable for all four segments to completely overlap (CS = DS = ES =SS).

REGISTER ORGANISATION:

A register is a very small amount of fast memory that is built in the CPU (or Processor) in order to speed up the operation. Register is very fast and efficient than the other memories like RAM, ROM, external memory etc,. That‟s why the registers occupiedthe top position in memory hierarchy model.

The 8086 microprocessor has a total of fourteen registers that are accessible to the programmer. All these registers are 16-bit in size. The registers of 8086 are categorized into 5 different groups.

a) Generalregisters

b) Indexregisters

c) Segmentregisters

d) Pointerregisters

e) StatusRegister

S.No	Type	Register width	Name of the Registers
1	General purpose Registers(4)	16-bit	AX,BX,CX,DX
		8-bit	AL,AH,BL,BH,CL,CH,DL,DH
2	Pointer Registers	16-bit	Stack Pointer(SP) Base Pointer(BP)
3	Index Registers	16-bit	Source Index(SI) Destination Index(DI)
4	Segment Registers	16-bit	Code Segment(CS) Data Segment(DS) Stack Segment(SS) Extra Segment(ES)
5	Flag (PSW)	16-bit	Flag Register

a) General purpose Registers:

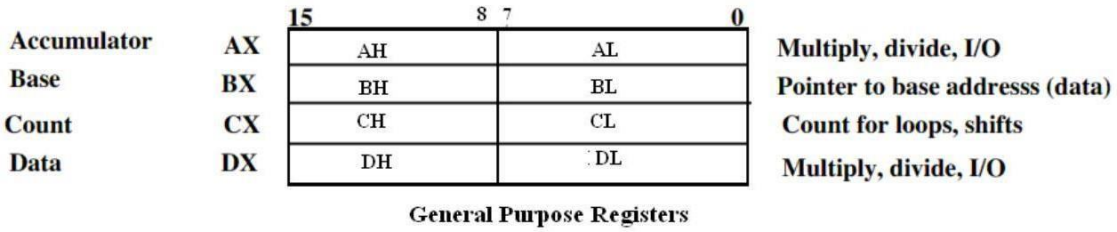
All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. These all general registers can be used as either 8-bit or 16-bit registers. The general registers are:

AX (Accumulator): AX is used as 16-bit accumulator. The lower 8-bits of AX are designated to use as AL and higher 8-bits as AH. AL can be used as an 8-bit accumulator for 8-bit operation.

This Accumulator used in arithmetic, logic and data transfer operations. For manipulation and division operations, one of the numbers must be placed in AX or AL.

BX (Base Register): BX is a 16 bit register, but BL indicates the lower 8-bits of BX and BH indicates the higher 8-bits of BX. The register BX is used as address register to form physical address in case of certain addressing modes (ex: indexed and register indirect).

CX (Count Register): The register CX is used default counter in case of string and loop instructions. Count register can also be used as a counter in string manipulation and shift/rotate instruction.

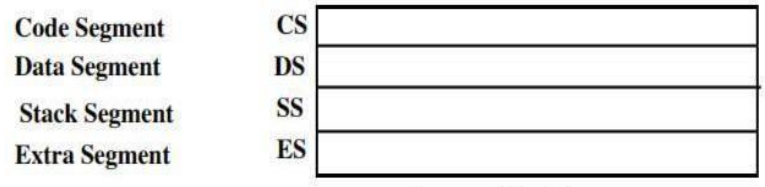


General Purpose Registers

DX (Data Register): DX register is a general purpose register which may be used as an implicit operand or destination in case of a few instructions. Data register can also be used as a port number in I/Ooperations.

Segment Register:

The 8086 architecture uses the concept of segmented memory. 8086 can able to access a memory capacity of up to 1 megabyte. This 1 megabyte of memory is divided into 16 logical segments. Each segment contains 64 Kbytes ofmemory.



Segment Registers

This memory segmentation concept will discuss later in this document. There are four segment registers to access this 1 megabyte of memory. The segment registers of 8086 are:

CS(Code Segment): Code segment (CS) is a 16-bit register that is used for addressing memory location in the code segment of the memory (64Kb), where the executable program is stored. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions.

Stack segment (SS): Stack Segment (SS) is a 16-bit register that used for addressing stack segment of the memory (64kb) where stack data is stored. SS register can be changed directly using POPinstruction.

Data segment (DS): Data Segment (DS) is a 16-bit register that points the data segment of the memory (64kb) where the program data is stored. DS register can be changed directly using POP and LDS instructions.

Extra segment (ES): Extra Segment (ES) is a 16-bit register that also points the data segment of the memory (64kb) where the program data is stored. ES register can be changed directly using POP and LES instructions.

b) IndexRegisters:

The index registers can be used for arithmetic operations but their use is usually concerned with the memory addressing modes of the 8086 microprocessor (indexed, base indexed and relative base indexed addressing modes).

The index registers are particularly useful for string manipulation.

SI (Source Index):

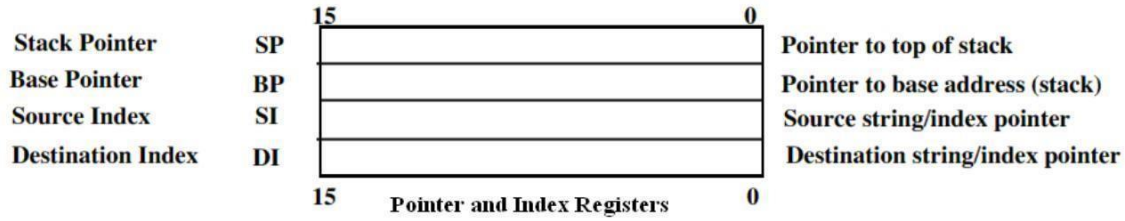
SI is a 16-bit register. This register is used to store the offset of source data in data segment. In other words the Source Index Register is used to point the memory locations in the data segment.

DI (Destination Index):

DI is a 16-bit register. This is destination index register performs the same function as SI. There is a class of instructions called string operations that use DI to access the memory locations in Data or Extra Segment.

c) PointerRegisters:

Pointer Registers contains the offset of data(variables, labels) and instructions from their base segments (default segments).8086 microprocessor contains three pointer registers.



SP (Stack Pointer): Stack Pointer register points the program stack that means SP stores the base address of the Stack Segment.

BP (Base Pointer): Base Pointer register also points the same stack segment. Unlike SP, we can use BP to access data in the other segments also.

IP (Instruction Pointer):

The Instruction Pointer is a register that holds the address of the next instruction to be fetched from memory. It contains the offset of the next word of instruction code instead of its actualaddress

d) Status Register or FlagRegister:

The status register also called as flag register. The 8086 flag register contents indicate the results of computation in the ALU. It also contains some flag bits to control the CPU operations.

It is a 16 bit register which contains six status flags and three control flags. So, only nine bits of the 16 bitregister are defined and the remaining seven bits are undefined. Normally this status flag bits indicate the status of the ALU after the arithmetic or logical operations. Each bit of the status register is a flip/flop. The Flag register contains Carry flag, Parity flag, Auxiliary flag Zero flag, Sign flag ,Trap flag, Interrupt flag, Direction

flag and overflow flag as shown in the diagram. The CF,PF,AF,ZF,SF,OF are the status flags and the TF,IF and CF are the control flags.

X	X	X	X	**OF**	**DF**	**IF**	**TF**	**SF**	**ZF**	**X**	**AF**	**X**	**PF**	**X**	**CF**

CF- Carry Flag: This flag is set, when there is a carry out of MSB in case of addition or a borrow in case ofsubtraction.

PF - Parity Flag : This flag is set to 1, if the lower byte of the result contains even number of 1"s else (for odd number of 1s) set to zero.

AF- Auxilary Carry Flag: This is set, if there is a carry from the lowest nibble, i.e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

ZF- Zero Flag: This flag is set, if the result of the computation or comparison performed by the previous instruction is zero

SF- Sign Flag : This flag is set, when the result of any computation is negative

TF - Tarp Flag: If this flag is set, the processor enters the single step execution mode.

IF- Interrupt Flag: If this flag is set, the maskable interrupt INTR of 8086 is enabled and if it is zero ,the interrupt is disabled. It can be set by using the STI instruction and can be cleared by executing CLIinstruction.

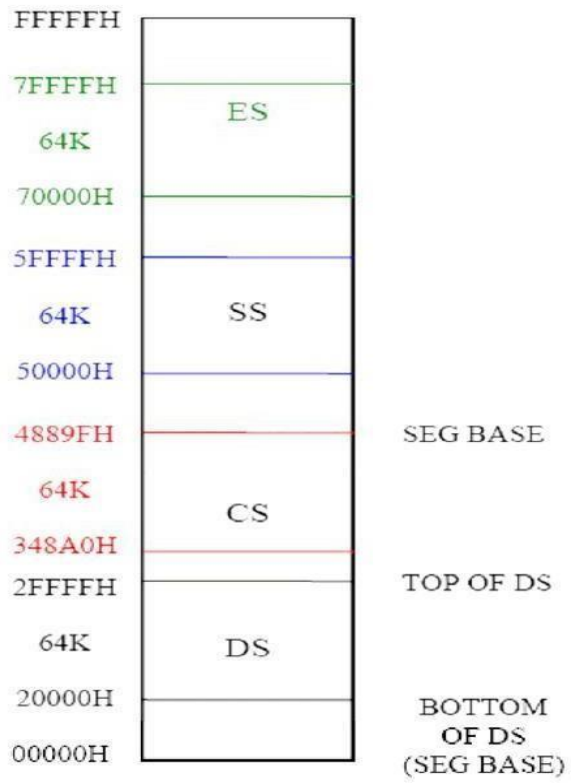
DF- Direction Flag: This is used by string manipulation instructions. If this flag bit is „0", the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

OF- Over flow Flag: This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of

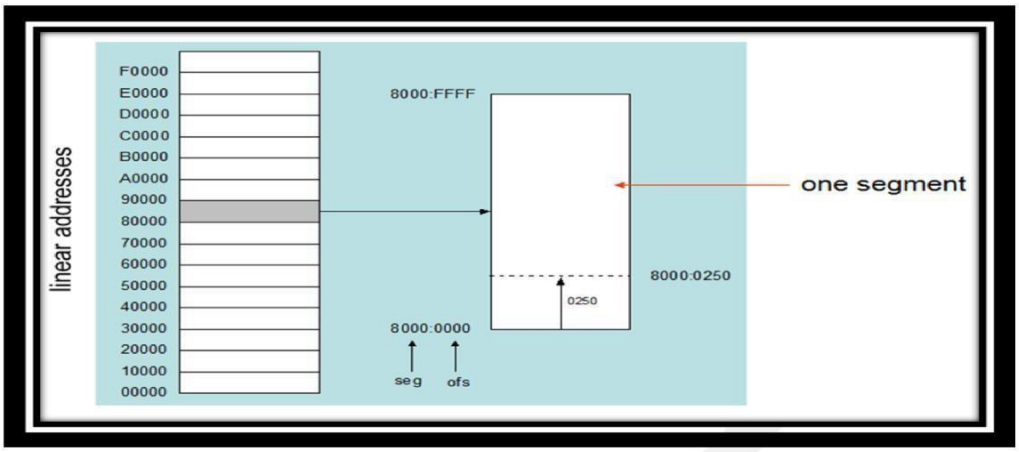
more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

MEMORY SEGMENTATION:

- It is the process in which the main memory of computer is divided into different segments and each segment has its own baseaddress.



- Segmentation is used to increase the execution speed of computer system sothat processor can able to fetch and execute the data from memory easily andfastly.
- The size of address bus of 8086 is 20 and is able to address 1 Mbytes ofphysical memory.
- The compete 1 Mbytes memory can be divided into 16 segments, each of 64Kbytes 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*.



Types of Segmentation:

1. **OverlappingSegment:**

- A segment starts at a particular address and its maximum size can go up to 64 Kbytes. But if another segment starts along this 64 Kbytes location of the first segment, the two segments are said to be overlappingsegment.
- The area of memory from the start of the second segment to the possibleend of the first segment is called as overlappedsegment
- **Non Overlapped Segment:**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 Non- overlapping segment.

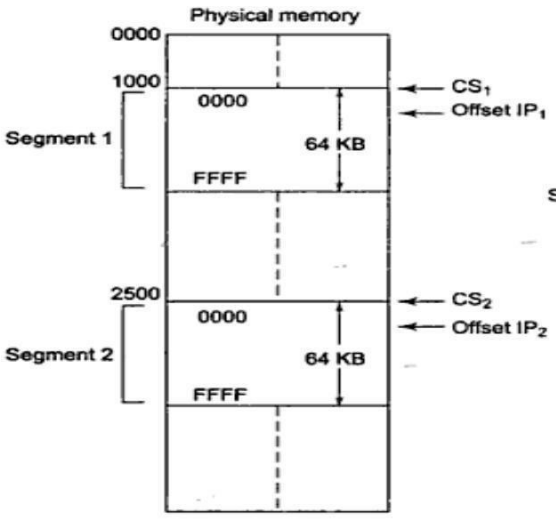


Fig. 1.3(a) *Non-overlapping Segments*

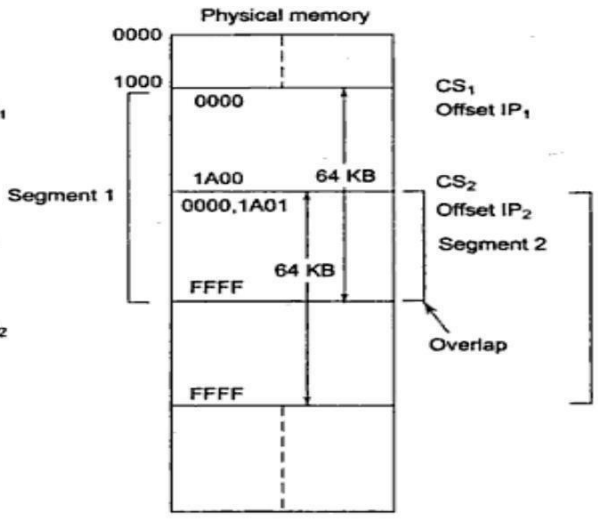


Fig. 1.3(b) *Overlapping Segments*

Advantages of Segmented memory:

- Allows the memory capacity to be 1MB although the actual addresses to behandled are of 16 bitsize.
- Allows the placing of code, data and stack portions of the same program indifferent parts (segments) of the memory, for data and code protection.
- Permits a program and/or its data to be put into different areas of memory eachtime program is executed, i.e. provision for relocation may bedone.
- The segment registers are used to allow the instruction, data or stack portion of a program to be more than 64Kbytes long. The above can be achieved by using more than one code, data or stacksegments.

Addressing Modes of 8086:

Addressing mode indicates a way of locating data or operands. Depending up on the data type used in the instruction and the memory addressing modes, any instruction may belong to one or more addressing modes or same instruction may not belong to any of the addressing modes.

The addressing mode describes the types of operands and the way they are accessed for executing an instruction. According to the flow of instruction execution, theinstructions may be categorizedas

1. **Sequential control flow instructionsand**
2. **Control transferinstructions.**

Sequential control flow instructions are the instructions in which after execution of current instruction, control will be transferred to the next instruction appearing immediately after it (in the sequence) in the program. For example the arithmetic, logic, data transfer and processor control instructions are Sequential control flow instructions.

The control transfer instructions on the other hand transfer control to some predefined address or the address somehow specified in the instruction, after their execution. For example INT, CALL, RET & JUMP instructions fall under this category.

The addressing modes for Sequential and control flow instructions are explained as follows.

1. Immediate addressing mode: In this type of addressing, immediate data is a part of instruction, and appears in the form of successive byte orbytes.

In the above example, 0005H is the immediate data. The immediate data may be 8- bit or 16-bit in size.

2. Direct addressing mode: In the direct addressing mode, a 16-bit memoryaddress (offset) directly specified in the instruction as a part ofit.

3. Register addressing mode: In the register addressing mode, the data is stored in a register and it is referred using the particular register. All the registers, except IP, may be used in thismode.

4. Register indirect addressing mode: Sometimes, the address of the memory location which contains data or operands is determined in an indirect way, using the offset registers. The mode of addressing is known as register indirectmode.

In this addressing mode, the offset address of data is in either BX or SI or DI Register. The default segment is either DS or ES.

5. Indexed addressing mode: In this addressing mode, offset of the operand is stored one of the index registers. DS & ES are the default segments for index registers SI & DI respectively.

6. Register relative addressing mode: In this addressing mode, the data is available at an effective address formed by adding an 8-bit or 16-bit displacement with the content of any one of the register BX, BP, SI & DI in the default (either in DS & ES)segment.

7. Based indexed addressing mode: The effective address of data is formed in this addressing mode, by adding content of a base register (any one of BX or BP) tothe

content of an index register (any one of SI or DI). The default segment register may be ES or DS.

Example: MOV AX, [BX][SI]

8. Relative based indexed: The effective address is formed by adding an 8 or 16-bit displacement with the sum of contents of any of the base registers (BX or BP) and any one of the index registers, in a defaultsegment.

Example: MOV AX, 50H [BX] [SI]

For the control transfer instructions, the addressing modes depend upon whether the destination location is within the same segment or in a different one. It also depends upon the method of passing the destination address to the processor. Basically, there are two addressing modes for the control transfer instructions, viz. intersegment and intrasegment addressing modes.

If the location to which the control is to be transferred lies in a different segment other than the current one, the mode is called intersegment mode. If the destination location lies in the same segment, the mode is called intrasegment mode.

Addressing Modes for control transfer instructions:

9. Intersegment

a) Intersegment direct b) Intersegmentindirect

10Intrasegment

a) Intrasegment direct b) Intrasegmentindirect

9. (a) Intersegment direct: In this mode, the address to which the control is to be transferred is in a different segment. This addressing mode provides a means of branching from one code segment to another code segment. Here, the CS and IP of the destination address are specified directly in theinstruction.

9. (b) Intersegment indirect: In this mode, the address to which the control is to be transferred lies in a different segment and it is passed to the instruction indirectly, i.e. contents of a memory block containing four bytes, i.e. IP(LSB), IP(MSB), CS(LSB) and

CS(MSB) sequentially. The starting address of the memory block may be referred using any of the addressing modes, except immediate mode. **Example:** JMP [2000H].

Jump to an address in the other segment specified at effective address 2000H in DS.

10.(a) Intrasegment direct mode: In this mode, the address to which the control is to be transferred lies in the same segment in which the control transfers instruction lies and appears directly in the instruction as an immediate displacement value. In this addressing mode, the displacement is computed relative to the content of the instructionpointer. The effective address to which the control will be transferred is given by the sum of 8 or 16 bit displacement and current content of IP. In case of jump instruction, if the signed displacement (d) is of 8-bits (i.e. -128<d<+127), it as short jump and if it is of 16 bits (i.e. -32768<d<+32767), it is termed as long jump. **Example:** JMP SHORT LABEL.

10.(b) Intrasegment indirect mode: In this mode, the displacement to which the control is to be transferred is in the same segment in which the control transfer instruction lies, but it is passed to the instruction directly. Here, the branch address is found as the content of a register or a memory location.This addressing mode may be used in unconditional branchinstructions.

Example: JMP [BX]; Jump to effective address stored in BX.

Instruction set of 8086

The Instruction set of 8086 microprocessor is classified into 8, they are:-

- **Data transfer instructions**
- **Arithmetic instructions**
- **Logical instructions**
- **Shift / rotate instructions**
- **Flag manipulation instructions**
- **Program control transfer instructions**
- **Machine Control Instructions**
- **String instructions**