Chapter 4. Intel 8085/8086/8088

4.1. FEATURES

The features of INTEL 8085 are:

- 8 bit microprocessor (8085 microprocessor can read or write or perform arithmetic and logical operations on 8-bit data at time)
- It is a single chip N-MOS device with 40 pins.
- It has 8 data lines and 16 address lines hence capacity is 2^{16} = 64 kB of memory
- In 8085, the lower 8-bit address bus (A0- A7) and data bus (D0-D7) are multiplexed to reduce the number of external pins. But due to this, external hardware (latch) is required to separate address lines and data lines.
- It works on 5 Volt dc power supply.
- The maximum clock frequency is 3 MHz while minimum frequency is 500 kHz.
- It provides 74 instructions with 5 different addressing modes.
- It generates 8 bit I/O address so it can access 2^8=256 input ports.
- It provides 5 hardware interrupts: TRAP, RST 5.5, RST 6.5, RST 7.5, INTR and 8 software interrupts.
- It provides Accumulator, one flag register, 6 general purpose registers and two special purpose registers (SP, PC).
- It provides serial lines SID, SOD. So serial peripherals can be interfaced with 8085 directly.

4.2. Architecture of 8085

Figure 4.1 shows the architecture of 8085. It consists of various functional block as listed below:

1. Registers

2. Arithmetic and Logic Unit

3. Instruction decoder and machine cycle encoder

4. Address buffer

5. Address/Data buffer

6. Incrementer/Decrementer Address Latch

7. Interrupt Control

8. Serial I/O Control

9. Timing and control circuitry

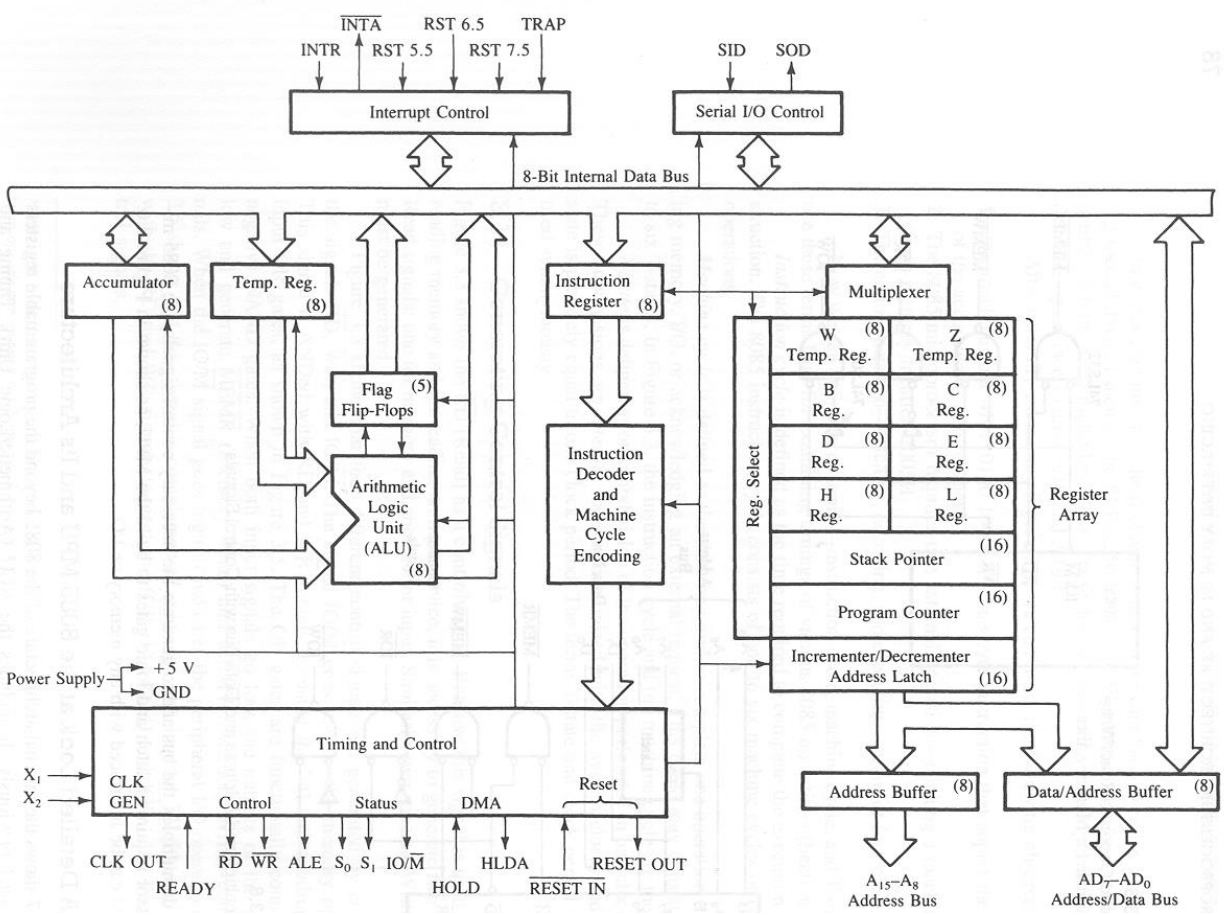


Figure 4-1: Functional block diagram of 8085 microprocessor

1. Registers

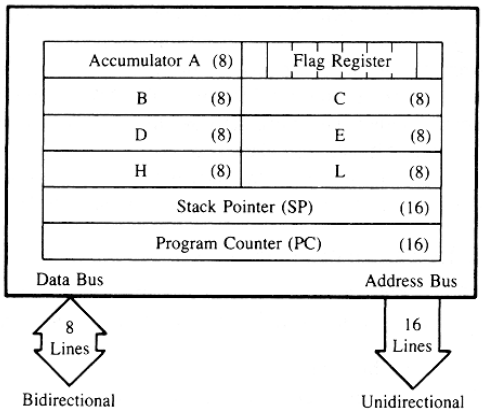


Figure 4-2: Registers of 8085

The Registers are of 8-bit & 16-bit size used for different purposes. It includes six 8-bit general purpose registers- (B, C, D, E, H and L) one accumulator, one flag register and two 16-bit registers (SP and PC). All these registers are assessable to programmer hence they are included in the programmer's model. The remaining registers- temporary, W and Z are not accessible to the programmer; they are used by microprocessor for internal, intermediate operations. The 8085 registers are classified as:

1.1. General Purpose Registers

B, C, D, E, H, and L are 8-bit general purpose registers can be used as a separate 8-bit registers or as 16-bit registers pairs, BC, DE, and HL. When used in register pair mode, the high order byte resides in the first registers (i.e. B when BC is used as a register pair) and low order byte in the second (i.e. in C when BC is used as a register pair).

HL pair also functions as a data pointer or memory pointer.

1.2. Temporary Registers

(a) Temporary register, W & Z – These registers are only used by 8085 and are not available for the programmer.

(b) TMP register, –TMP is also an 8-bit buffer register used to hold data during arithmetic and logical operation.

1.3. Special Purpose Registers:

(a) Register A (Accumulator): This is a special purpose register. All the ALU operations are performed with reference to the contents of Accumulator.

(b) Flag Registers (F – Flag register): This register indicates the status of the ALU operation. This is 8 bit register in which five of the bits carry significant information in the form of flags as shown in fig. 4-3.

The five status flag of 8085 µP: S (sign flag), Z (zero flag), AC (auxiliary carry flag), P (parity flag) & CY (carry flag).The flag bits are affected by the arithmetic & logic operations.

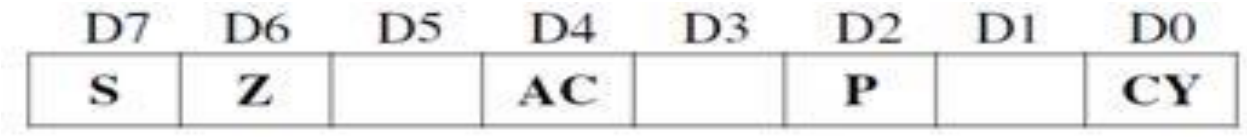


Figure 4-3: Flag registers

S: Sign flag is set when result of an operation is negative.

Z: Zero flag is set when result of an operation is 0.

Ac: Auxiliary carry flag is set when there is a carry out of lower nibble or lower four bits of the operation.

CY: Carry flag is set when there is carry generated by an operation.

P: Parity flag is set when result contains even number of 1's.

Sign Flags-

Used for indicating the sign of the data in the accumulator. The sign flag is set if bit D7 of the accumulator is set after an arithmetic or logic operation

• The sign flag is set if negative (1 – negative)

• The sign flag is reset if positive (0 –positive)

Zero Flag

– Is set if result obtained after an operation is 0

– Is set following an increment or decrement operation of that register

10110011

\+ 01001101

---

1 00000000

Auxiliary Carry Flag

- Is set if there is a carry out of bit 3
- AC hold carry out of the bit number 3 to the bit number 4 resulting from the execution of an arithmetic operation.
- Counting of the bits starts from zero.
- ADD CB and E9

C B = 1100 1011

E 9 = 1110 1001

1011 0100

CY =1

MSB =1 ; S=1

AC =1

Four 1's ; P=1

Result is non zero ; Z=0

Parity Flag- After an ALU operation if the result has an even no of 1's the p-flag is set. Otherwise it is cleared. So, the flag can be used to indicate even parity.

- Is set if parity is even
- Is cleared if parity is odd

Carry flag

- Is set if there is a carry or borrow from arithmetic operation

1011 0101	**1011 0101**
+ 0110 1100	**- 1100 1100**
----------	**----------**
Carry 1 0010 0001	**Borrow 1 1110 1001**

(C) Instruction Register: – It is a 8 bit register that stores the instruction fetched from memory

In a typical processor operation, the processor first fetches the opcode of instruction from memory. The CPU stores this opcode in a register called the instruction register. This opcode is further sent to the instruction decoder.

1.4. Sixteen bit register:

Program Counter (PC) and stack pointer (SP) are two 16-bit registers used to hold memory addresses.

(a) Program Counter (PC):

The function of the PC is to point to the memory address from which the next byte is to be fetched. When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location.

(b) Stack Pointer (SP):

It points to a memory location in R/W memory, called the stack. This is a 16-bit register used to address the top of the stack memory location. The beginning of the stack is defined by loading a 16-bit address in the stack pointer. The PC will automatically update when calling to /returning from Subroutines. The stack is usually accessed in a Last in First out (LIFO) fashion.

2. Arithmetic and Logic Unit (ALU):

The 8085's ALU operation performs arithmetic and logical functions on eight bit variables. The arithmetic unit performs bitwise fundamental arithmetic operations such as addition and subtraction. The logic unit performs logical operations such as complement, AND, OR and EX-OR, as well as rotate and clear.

3. Instruction Decoder:

The processor first fetches the opcode of instruction from memory and stores this opcode in the instruction register. It is then sent to the instruction decoder. The instruction decoder decodes it and accordingly give timing and control signals which control the register, the data buffers, ALU and external peripheral signals depending on the nature of the instruction.

4. Address Buffer:

This is an 8-bit unidirectional buffer. It is used to drive external high order bus (A15-A8). It is also used to tri-state the high order address bus under certain conditions such as reset, hold, halt, and when address lines are not in use.

5. Address/ Data Buffer:

This is 8-bit bi-directional buffer. It is used to drive multiplexed address/data bus. It is also used to tri-state the multiplexed address/data bus under certain conditions such as reset, hold, halt, and when address lines are not in use.

6. Incrementer/Decrementer Address Latch

This is 16-bit register used to increment or decrement the content of program counter or stack pointer as a part of execution of instructions related to them.

7. Interrupt Control

Responsible for controlling interrupt. Receives hardware interrupts and sends the acknowledgement for receiving the interrupt signal and then calls the corresponding ISR (interrupt subroutine). The interrupt control block has five interrupt inputs RST 5.5, RST 6.5, RST 7.5, TRAP and INTR and one acknowledgement signal $\overline{\text{INTA}}$.

8. Serial I/O Control

It is responsible for controlling the serial data. 8085's serial I/O control provides two control signal: SID (Serial Input data) and SOD (Serial Output Data) for serial communication. SID is activated when the data is to be read (receive) serially. For transferring the data serially out, SOD should be active.

9. Timing and Control unit

This unit generates timing and control signals necessary to carry out the instruction, which has been decoded. In reality causes certain connections between blocks of the microprocessor to be opened or closed, so that data goes where it is required, and so that ALU operations occur.

4.3. 8085 Bus Structure

The 8-bit 8085 CPU (or MPU – Micro Processing Unit) communicates with the other units using a 16-bit address bus, an 8-bit data bus and a control bus.

Address Bus

- Consists of 16 address lines: A0 – A15
- Operates in unidirectional mode: The address bits are always sent from the MPU to peripheral devices, not reverse.
- 16 address lines are capable of addressing a total of 2^16 = 65,536 (64k) memory locations.
- Address locations: 0000 (hex) – FFFF (hex)

Data Bus

- Consists of 8 data lines: $D_0 – D_7$
- Operates in bidirectional mode: The data bits are sent from the MPU to peripheral devices, as well as from the peripheral devices to the MPU.
- Data range: 00 (hex) – FF (hex)

Control Bus

Consists of various lines carrying the control signals such as read / write enable, flag bits.

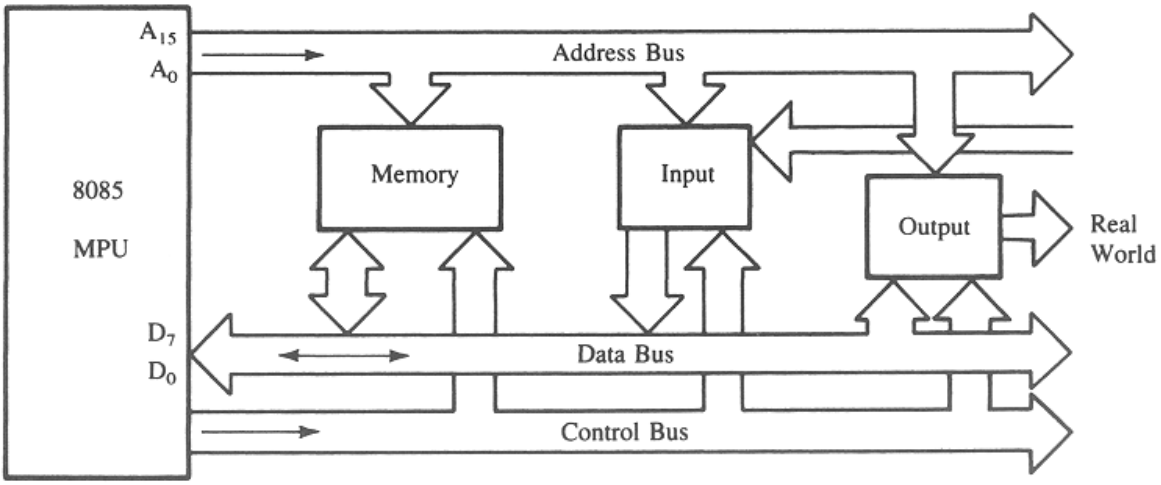


Figure 4-4: 8085 Bus structure

4.4. 8085 Pin Description

Microprocessor 8085 is a 40-pin IC which operate on +5V power supply and 3 MHz frequency. These 40 pins are divided into 6 groups according to their functions which are:

1. Frequency and power supply signal

2. High order address Bus

3. Multiplexed address/data bus

4. Control and status signal

5. Serial IO signals

6. Externally or peripheral initiated signals

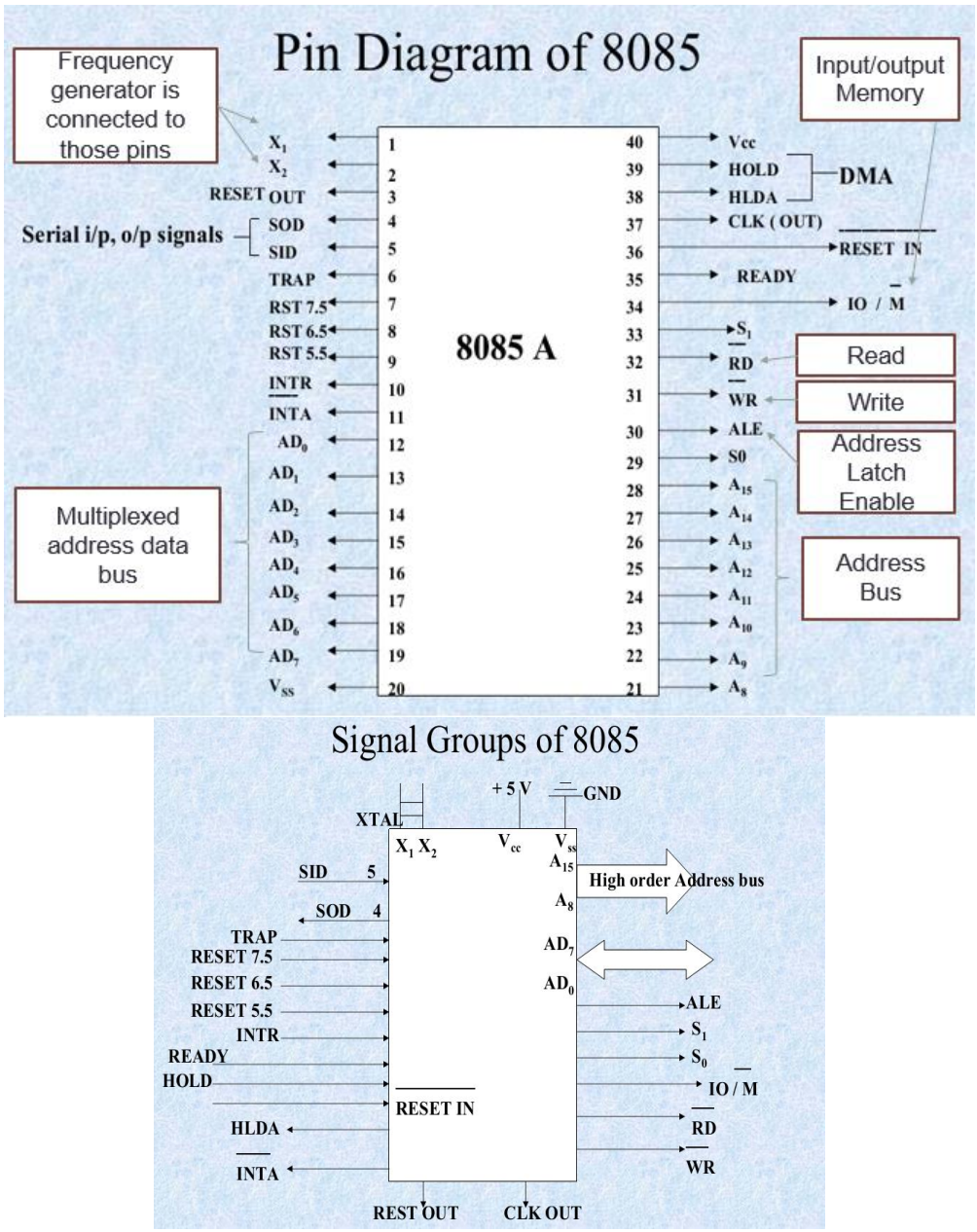


Figure 4-5: Pin configuration of 8085

4.4.1 Group 1: Power Supply and Frequency Signals

VCC: +5V supply

VSS: Ground reference (0V)

X1, X2(Input)

Crystal or RC oscillator circuits are connected to these lines to provide the required frequency to the microprocessor. The input frequency is divided by 2 to give the internal operating frequency

because the frequency is applied to the system through T flip flop which divides the incoming frequency by 2.

CLK OUT

This signal is used as a system clock for other devices. Its frequency is half the oscillator frequency

4.4.2 Group 2: Higher Order Address Bus (Output) (A8-A15)

Instead of having 24 pins for address and data lines, 8085 has only 16 pins. Out of 16 pins 8 pins are used to carry the higher order address and the other 8 pins are multiplexed to carry the address as well as data. This multiplexing is done to keep the number of pins as minimum as possible.

4.4.3. Group 3: Multiplexed Address/Data Bus (AD0-AD7)

The lower 8 lines (AD7-AD0) are often called as multiplexed data lines. The bits AD0 – AD7 are bi-directional and serve as A0 – A7 and D0 – D7 at the same time. During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits. In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes.

4.4.4. Group 4: Control and Status Signal

(a) ALE (Address latch enable) (output): We know AD0 to AD7 lines are multiplexed and the lower half address (A0-A7) is available only during T1 of the machine cycle. This lower half of the address is also necessary during T2 and T3 of machine cycle to access specific location in memory or I/O port. This means that the lower half of an address must be latched in T1 of the machine cycle, so that it is available throughout the machine cycle. The latching of lower half of an address is done by using external latch and ALE signal from 8085.

(b) $\overline{RD}$ and $\overline{WR}$: These signals are basically used to control the data flow between processor and memory or I/O device/port. A low on $\overline{RD}$ indicate that the data must be read from the selected memory location or I/O via data bus. A low on $\overline{WR}$ indicates that the data must be written into the selected memory location or I/O port via data bus.

(c) IO/$\overline{M}$, S0 and S1: IO/$\overline{M}$ indicate whether Input Output operation (IO/$\overline{M}$ =1) or memory operation (IO/$\overline{M}$ =0) is being carried out. S1 and S0 indicate the type of machine cycle in progress.

Table 4-1 shows the status signals of microprocessor 8085.

Table 4-1: 8085 machine cycle status and control signals

Machine Cycle	Status IO/$\overline{M}$	Status S_1	Status S_0	Control Signals
Opcode Fetch	0	1	1	$\overline{RD} = 0$
Memory Read	0	1	0	$\overline{RD} = 0$
Memory Write	0	0	1	$\overline{WR} = 0$
I/O Read	1	1	0	$\overline{RD} = 0$
I/O Write	1	0	1	$\overline{WR} = 0$
Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0$
Halt	Z	0	0	$\overline{RD}$, $\overline{WR}$ = Z and $\overline{INTA}$ = 1
Hold	Z	X	X	
Reset	Z	X	X	

NOTE: Z = Tri-state (high impedance)
X = Unspecified

4.4.5. Group 5: Serial IO signals

(a) SID (Serial I/P Data): This input is used to accept serial data bit by bit from the external device. The data on this line is loaded into accumulator bit 7 whenever a RIM (Read Interrupt Mask) instruction is executed.

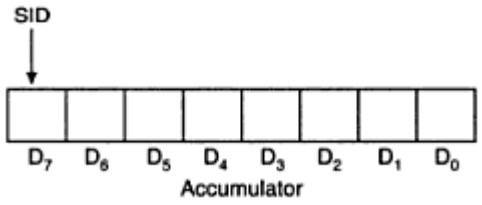


Figure 4-6: SID operation

(b) SOD (Serial O/P Data): This is an output signal which enables the transmission of serial data bit by bit to external device. The output SOD is set or reset as specified by the SIM instruction.

4.3.5. Group 6: Externally or peripheral initiated signals

This group includes all interrupt signals, reset signals, DMA signals and external acknowledgement signals

Interrupt Signals:

The 8085 has five hardware interrupt signals: RST 5.5, RST 6.5, RST 7.5, TRAP and INTR. The microprocessor recognizes interrupt request on these lines at the end of the current instruction execution.

The $\overline{\text{INTA}}$ (Interrupt Acknowledgement) signal is used to indicate that the processor has acknowledged an INTR interrupt.

(a) INTR (Interrupt Request): This is used as general purpose interrupt. The INTR is enabled or disabled by software (maskable)

(b) RST 5.5, RST 6.5, RST 7.5 (Restart Interrupt)

These are maskable interrupts and have low priority than TRAP.

RST 7.5 ~ Highest Priority

RST 6.5

RST 5.5 ~ Lowest Priority

The priority of these interrupts is ordered as shown above. These interrupts have a higher priority than the INTR.

(c) TRAP

TRAP interrupt is a non-maskable interrupt (i.e. it is unaffected by any mask or interrupt enable) and is highest priority interrupt.

(d) $\overline{\text{INTA}}$ (Interrupt Acknowledgement):

This pin is used to acknowledge the interrupt. A low on $\overline{\text{INTA}}$ grants the interrupt request and transfers the program control to the particular ISR (Interrupt service routine).

Reset Signals

(a) $\overline{\text{RESET IN}}$: When the signal on this pin goes low, the program counter is set to zero, the buses are tri-stated, and the microprocessor is reset.

(b) RESET OUT:

This active high signal indicate that processor is being reset. This signal is synchronized to the processor clock and it can be used to reset other devices connected in the system.

DMA Signal

(a) HOLD:

This signal indicate that the peripheral such as DMA (Direct Memory Access) controller is requesting the use of address and data buses. The processor can regain the buses only when HOLD is removed.

(b) HLDA (Hold Acknowledgement)

This active high signal is used to acknowledge HOLD request. Fig.4.7 shows the Hold and hold acknowledgement operation

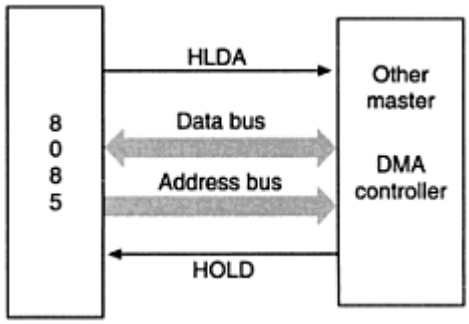


Figure 4-7: HOLD operation

Ready:

This signal is used to synchronize the slower peripherals with microprocessor. If 'Ready' is high during a read or write cycle, it indicate that the memory or peripheral is ready to send or receive data. If 'Ready' is low, the CPU will wait for 'Ready' to go high before completing the read and write cycle

4.5. Addressing modes of 8085

Every instruction of a program has to operate on a data. The method of specifying the data to be operated by the instruction is called Addressing mode. The 8085 has the following 5 different types of addressing.

1. Immediate Addressing
2. Direct Addressing
3. Register Addressing
4. Register Indirect Addressing
5. Implied Addressing

1. Immediate Addressing:

In immediate addressing mode, the data is specified in the instruction itself. The data will be a part of the program instruction.

Example:

MVI B, 3EH : Move the data 3EH given in the instruction to B register;

LXI SP, 2700H : Load stack pointer with 2700H

ADI 06H : Add 06H to the accumulator

2. Direct Addressing:

In this mode of addressing, the address of the data (operand) is specified within the instruction. The data will be in memory or IO location.

Example:

IN 02H : Input the data from input port 02H in accumulator

LDA 4100H : Load the data available in memory location 4100H in to accumulator

STA 2000H : Store the content of the accumulator to memory location 2000H

3. Register Addressing:

In register addressing mode, the instruction specifies the name of the register in which the data is available.

Example:

MOV A, B : Move the content of B register to A register

ANA B : contents of Register B or logically ANDed with contents of register A
ADD C : Add the contents of register C to accumulator

4. Register Indirect Addressing:

In register indirect addressing mode, the instruction specifies the name of the register in which the address of the data is available. Here the data will be in memory and the address will be in the register pair.

Example

MOV A, M : The memory data addressed by H L pair is moved to A register.

SUB M : Subtract the accumulator with the data in memory addressed by H L pair.

LDAX B. : Load accumulator with the data in memory addressed by B C pair

5. Implied Addressing:

In this type of addressing mode, no operand (register or data) is specified in the instruction. The operand is inherent to the instruction.

Example

CMA : Complement the content of accumulator

RAL : Rotate left to the content of accumulator

4.6. Intel 8086/8088 microprocessor

Features of 8086 microprocessor

1. It is 16 bit processor. So that it has 16 bit ALU, 16 bit registers and internal data bus and 16 bit external data bus. It makes faster processing.
2. It has three version based on the frequency of operation:
a) 8086 -> 5MHz
b) 8086-2 ->8MHz
c) 8086-1 ->10 MHz
3.8086 has 20 bit address lines to access memory. Hence it can access 2^20 = 1 MB memory location.
4.8086 has 16-bit address lines to access I/O devices, hence it can access 2^16 = 64K I/O location
5. Pipelining:-8086 uses two stage of pipelining. First is Fetch Stage and the second is Execute Stage.
Fetch stage that prefetch up to 6 bytes of instructions stores them in the queue.
Execute stage that executes these instructions.
Pipelining improves the performance of the processor so that operation is faster.
6. Operates in two modes:-8086 operates in two modes:
a)Minimum Mode: A system with only one microprocessor.
b)Maximum Mode:-A system with multiprocessor.
7.8086 uses memory banks:-The 8086 uses a memory banking system. It means entire data is not stored sequentially in a single memory of 1 MB but memory is divided into two banks of 512KB.
8. Interrupts:-8086 has 256 vectored interrupts.
9. Multiplication and Division:-8086 has a powerful instruction set. So that it supports multiply and Divide operation.

8086 is the first 16-bit microprocessor from INTEL, released in the year 1978. It is a 40 pin DIP chip based on N-channel, depletion load silicon gate technology (HMOS). The term 16 bit means that it's ALU, its internal registers and most of the instructions are designed to work with 16 bit

binary words. 8086 microprocessor has a 16-bit data bus and 20-bit address bus. So, it can address any one of 2^20 =1048576=1 MB memory locations. The 8086 microprocessor can work in two modes of operations .They are Minimum mode and Maximum mode. In the minimum mode of operation the microprocessor do not associate with any co-processors and cannot be used for multiprocessor systems. But in the maximum mode the 8086 can work in multi-processor or co-processor configuration. This minimum or maximum operations are decided by the pin MN/ MX (Active low). When this pin is high 8086 operates in minimum mode otherwise it operates in Maximum mode.

4.7. 8085 VS 8086

8085	**8086**
8 bit microprocessor	16 bit microprocessor
16 bit address bus and 8 bit data bus	20 bit address bus and 16 bit data bus
The clock frequency of 8085 microprocessor is 3MHz	Clock Frequencies of 5,8 &10 MHz
It can access up to 2^16= 64KB of memory	It can access up to 2^20= 1MB of memory
It has 5 flags	It has 9 flags
It does not support pipelining	It support pipelining
It does not have instruction queue	It has instruction queue
It can address 2^8= 256 I/O's.	It can access 2^16= 65,536 I/O's.
It supports only single operating mode	It operates in two modes i.e. minimum and maximum modes
It does not support memory segmentation	It supports memory segmentation
Low cost	High cost

4.8. Architecture of 8086

To improve the performance by implementing the parallel processing concept the CPU of the 8086 is divided into two independent sections. They are Bus Interface Unit (BIU) and Execution Unit (EU). The BIU sends out addresses, fetches instructions, read data from ports and memory and writes data to ports and memory i.e., the BIU handles all transfers data and addresses on the buses required by the execution Unit. Whereas the Execution Unit decodes the instructions and executes those instructions.

The Execution Unit:

The Execution Unit consists of a control system , a 16-bit ALU, 16-bit Flag register and four general purpose registers(AX,BX,CX,DX), pointer registers (SP,BP) and Index registers(SI,DI) of each 16-bits . The control circuitry controls the internal operations .The decoder in the execution unit decodes the instructions fetched from the memory into a series of actions. The ALU can add, subtract, perform operations like logical AND, OR, XOR, increment, decrement, complement, and shifting the binary numbers.

Bus Interface Unit:

The BIU consists of a 6-byte long instruction register called Queue. And four segment registers (ES, CS, SS, DS), one Instruction Pointer (IP) and an adder circuit to calculate the 20bit physical address of a location. This bus interface unit will perform all the external bus operations. They are fetching the instructions from the memory, read/write data from/into memory or port and also supporting the instruction Queue etc. The BIU fetches up to six instruction bytes from the memory and stores these pre-fetched bytes in a first –in first out register set called Queue. When the execution unit is ready for the execution of the instruction ,instead of fetching the byte from the memory ,it reads the byte from the Queue .This will increase the overall speed of microprocessor .Fetching the next instruction while the current instruction executes is called pipelining or parallel processing.

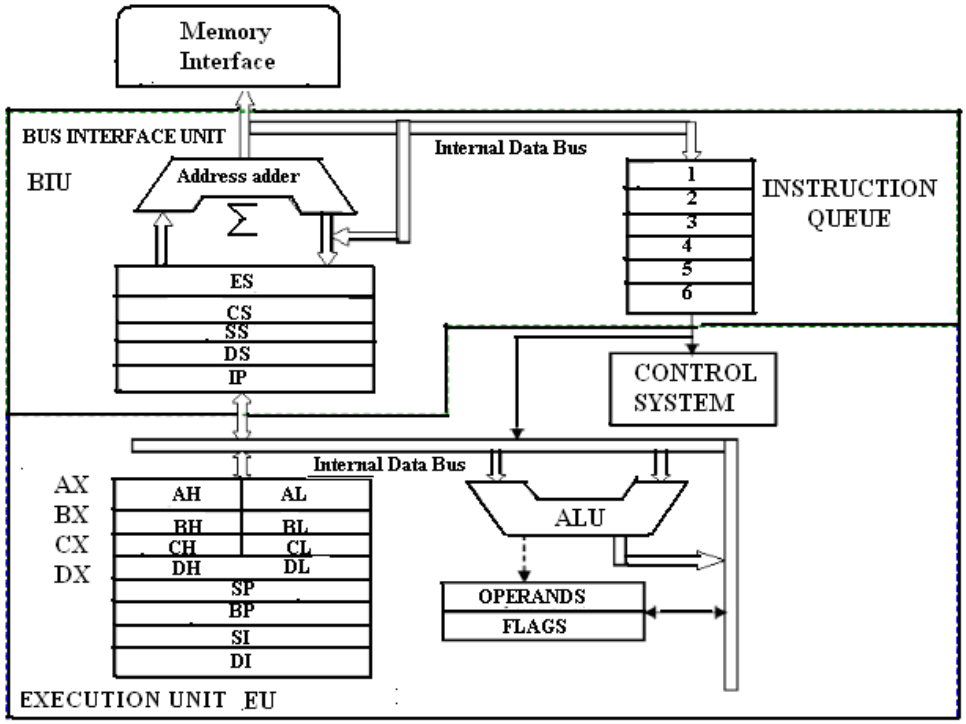


Figure 8-1: Architecture of 8086 Microprocessor

REGISTER ORGANISATION:

The 14 registers of 8086 microprocessor are categorized into four groups. They are general purpose data registers, Pointer & Index registers, Segment registers and Flag register as shown in the table below.

Table 8-1: 8086 Microprocessor Registers.

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	Instruction	16-bit	Instruction Pointer (IP)
6	Flag (PSW)	16-bit	Flag Register

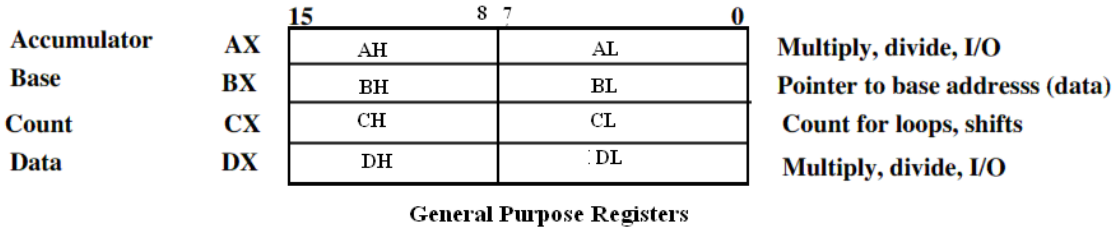
General purpose registers:

There are four 16-bit 4 general purpose registers namely (AH, AL) ;(BH, BL); (CH, CL); (and DH, DL) which are part of Execution unit. The general purpose registers, can be used either 8-bit registers or 16-bit registers. The general purpose registers are either used for holding the data, variables and intermediate results temporarily or for other purpose like counter or for storing offset address for some particular addressing modes etc. These registers can be used individually for storing 16-bit data temporarily .The AL register is also called the accumulator. The pairs of registers can be used together to store 16-bit data words.

It is always advantageous to store the data in these registers because the data can be accessed much more easily as these registers are already in the execution unit. Here L indicates the lower byte and H indicates the higher byte. X indicates the extended register. The general purpose data registers are used for data manipulations. The use of these registers is more dependent on the mode of addressing also.

Index / pointer registers:

The other four registers of EU are referred to as index / pointer registers. They are Stack Pointer register, Base Pointer register, Source Index register and Destination Index registers. The pointer registers contain the offset within a particular segment.



General Purpose Registers

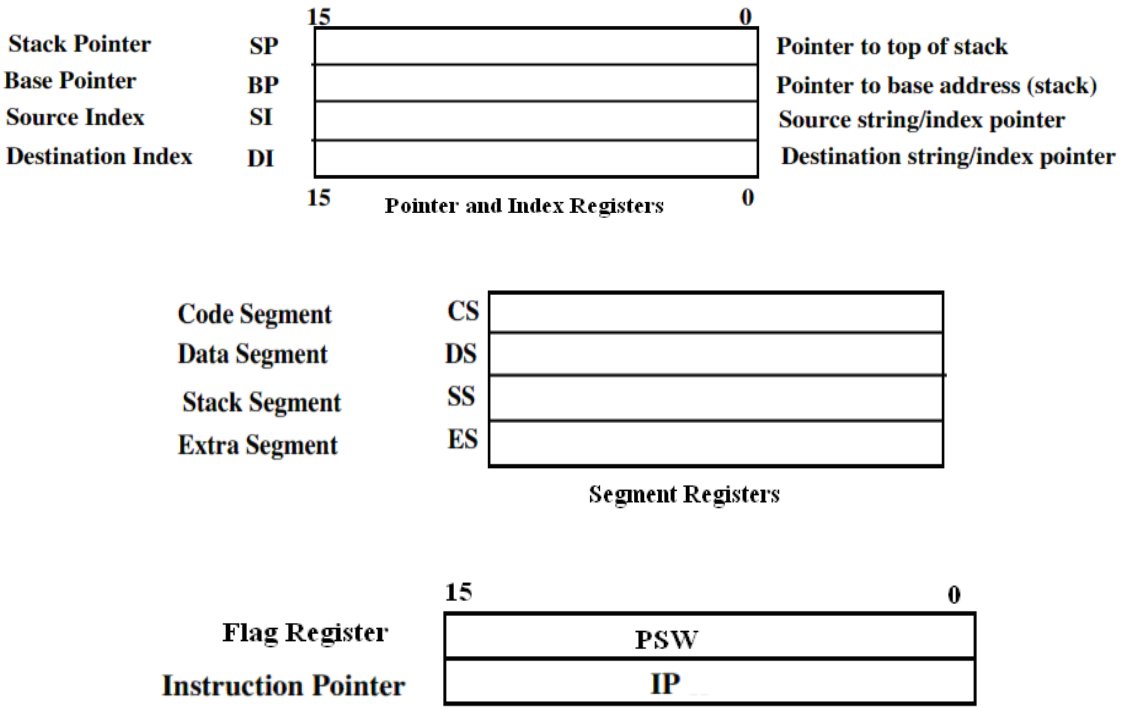


Figure 8-2: Register Organization

The BP & SP registers holds the offsets within the data and stack segments respectively. The Index registers are used as general purpose registers as well as for holding the offset in case of indexed based and relative indexed addressing modes. The source Index register is generally used to store the offset of source data in data segment while the Destination Index register used to store the offset of destination in data or extra segment. These index registers are specifically used in string manipulations.

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

Segment Registers:

There are four 16-bit segment registers namely code segment register(CS),Stack segment register(SS),Data segment register(DS) and Extra segment register(ES).The code segment register is used for addressing the 64kB memory location in the code segment of the memory ,where the code of the executable program is stored. Similarly the DS register points to the data segment of the 64kB memory where the data is stored. The Extra segment register also refers to essentially

another data segment of the memory space. The SS register is useful for addressing stack segment of memory. So, the CS, DS, SS and ES segment registers respectively contains the segment addresses for the code, data, stack and extra segments of the memory.

Instruction Pointer Register

It is a 16-bit register which always points to the next instruction to be executed within the currently executing code segment. So, this register contains the 16-bit offset address pointing to the next instruction code within the 64kB of the code segment area. Its content is automatically incremented as the execution of the next instruction takes place.

Flag Register:

This register is also called status register. It is a 16 bit register which contains six status flags and three control flags. So, only nine bits of the 16 bit register 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.

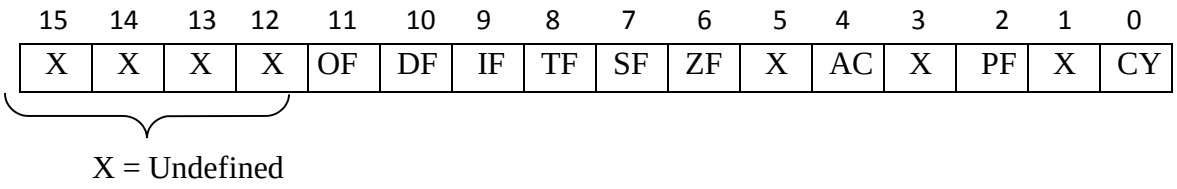


Figure 8-3: Flag Register of 8086

Flags Register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program.

The 8086 flag register as shown in the fig. 8-3. 8086 has 9 active flags and they are divided into two categories:

1. Conditional Flags

2. Control Flags

Conditional Flags

Conditional flags are as follows:

Carry Flag (CY): This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.

Auxiliary Flag (AC): If an operation performed in ALU generates a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), the AC flag is set i.e. carry given by D3 bit to

D4 is AC flag. This is not a general-purpose flag, it is used internally by the Processor to perform Binary to BCD conversion.

Parity Flag (PF): This flag is used to indicate the parity of result. If lower order 8-bits of the result contains even number of 1's, the Parity Flag is set and for odd number of 1's, the Parity flag is reset.

Zero Flag (ZF): It is set; if the result of arithmetic or logical operation is zero else it is reset.

Sign Flag (SF): In sign magnitude format the sign of number is indicated by MSB bit. If the result of operation is negative, sign flag is set.

Over flow Flag (OF)-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.

Control Flags

Control flags are set or reset deliberately to control the operations of the execution unit. Control flags are as follows:

Trap Flag (TF): It is used for single step control. It allows user to execute one instruction of a program at a time for debugging. When trap flag is set, program can be run in single step mode.

Interrupt Flag (IF): It is an interrupt enable/disable flag. If it is set, the maskable interrupt of 8086 is enabled and if it is reset, the interrupt is disabled. It can be set by executing instruction sit and can be cleared by executing CLI instruction.

Direction Flag (DF): It is used in string operation. If it is set, string bytes are accessed from higher memory address to lower memory address. When it is reset, the string bytes are accessed from lower memory address to higher memory address.

Physical address formation:

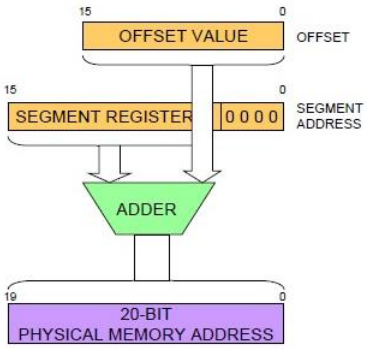
The 8086 addresses a segmented memory. The complete physical address which is 20-bits long is generated using segment and offset registers each of the size 16-bit.The content of a segment register also called as segment address, and content of an offset register also called as offset address. To get total physical address, put the lower nibble 0H to segment address and add offset address. For example: Base address of a segment is 2045H and Offset value is 0345H then,

Logical address is, Base address: Offset---------- > 2045H: 0345H

Physical Address is, *Base address * 10H + offset*

2045H*10H=======> 20450H
20450H
+0345H

20795H ---------- > Physical address



4.9. Addressing mode of 8086

Each instruction performs an operation on the specified data (operand). Hence, an operand must be specified for an instruction to be executed. The operand may reside in the accumulator, in a general purpose register, or in a memory location. The way by which an operand is specified for an instruction is called *addressing mode.*

There are in total eight addressing modes for 8086 to specify an operand. They are:

1. Register Addressing mode: In this mode of addressing, an 8-bit or 16-bit general purpose register contains an operand.

Example:

MOV BX, CX ; move the contents of CX register to BX register.

ADD AL, CH ; add the content of CH to the content of AL

ADD CX, DX ; add the content of DX register to the content of CX

2. Immediate Addressing mode: In the immediate addressing mode, the operand is contained in the instruction itself. The operand (8 or 16 bit data) can be specified as a part of instruction.

Examples:

MOV AL, 58H ; move 58H to AL register

MOV BX, 0354H ; move 0354H to BX register

3. Direct Addressing mode: In this mode of addressing an effective address (or offset) is given in the instruction itself.

Examples:

MOV AL, [5062H] ; move the content of offset address 5062H (20-bit physical address of memory location is calculated using DS and offset value 5062H) to AL

4. Register indirect addressing mode: The operand offset is in the base register, BX or base pointer BP or in an index register (SI or DI) specified in the instruction.

Examples:

ADD CX, [BX] ; add the contents of memory locations addressed by register BX to the register CX.

MOV DX, [SI] ; the content of memory location addressed by SI will move to DL and the content of next memory location will move to DH

5. Based Addressing mode: The operand's offset is the sum of the contents of base register, BX or BP and an 8-bit or 16-bit displacement.

Offset (effective address) = [BX or BP + 8-bit or 16-bit displacement]

Example:

ADD AL, [BX+4] ; the content of [BX+4] will be added to the content of AL, and the result will be placed in AL.

6. Indexed Addressing mode: The operands offset is computed by adding an 8-bit or 16 bit displacement to the contents of indexed register SI or DI.

Offset = [SI or DI + 8-bit or 16-bit displacement]

Examples:

MOV AX, [SI +08]

MOV CX, [SI+1532H]

7. Based Indexed Addressing mode: The operand's offset is computed by adding the contents of a base register to the contents of an index register.

Offset = [BX or BP] + [SI or DI]

BX is used as a baser register for data segment. BP is used as a base register for stack.

Examples:

MOV AX, [BX+SI]

ADD CX, [BX+SI]

8. Based Indexed with Displacement: The operand's offset is computed by adding a base register's contents, an index register's contents and an 8-bit or 16-bit displacement.

Offset = [BX or BP] + [SI or DI] + Displacement

Examples:

MOV AX, [BX+SI +4]

ADD CX, [BX+SI+1523H]

4.10. Intel 8088

- Intel 8088 microprocessor was released in 1979, or one year after the Intel 8086 CPU. Both processors have the same architecture, and the only difference of the 8088 CPU from the 8086 is the external data bus width - it was reduced from 16 bits to 8 bits.
- Address bus – 20 bit
- Data bus -8 bit
- The 8088 CPU uses two consecutive bus cycles to read or write 16 bit data instead of one bus cycle for the 8086, which makes the 8088 processor to run slower
- 16-bit registers, 16-bit internal data bus and 20-bit address bus, which allows the processor address up to 1 MB of memory
- The 8088 uses the same segmented memory addressing as the 8086.
- The clock frequency of 8088 is 5 MHz and that of 8088-2 is 8 MHz
- 40 pin IC and operate at 5V dc supply
- Register set, instructions and addressing modes are same as those of 8086.
- The instruction queue length of 8088 microprocessor is of 4 byte.
- 8088 has only 8 data lines and hence it can use 8 bit I/O devices which are cheaper compared to 16-bit I/O devices.

4.11. 8086 VS 8088

The Intel 8088 has the same ALU, the same registers and the same instruction set as 8086. The 8088 has also 20-bit address bus so it can address any one of 2^20 or 1048576 locations. The 8088 however, has an 8-bit data bus, so it can read data from or write data to memory and ports only 8-bits at time. To read 16-bit word from two successive memory locations, the 8088 always perform to two read operations. The Intel 8088 is used as the CPU in the original IBM personal computer, the IBM PC/XT and several compatible personal computers.

8086 Microprocessor	**8088 Microprocessor**
8086 has 16-bit data lines.	8088 has 8-bit data lines.
8086 is available in three clock speed 5 MHz, 8 MHz and 10 MHz	Whereas 8088 is available in two clock speed 5 MHz and 8 MHz
The memory space of 8086 is organized as two 512KB banks.	The memory space of 8088 is implemented as single 1M*8 Memory bank.

8086 has 6-bit instruction queue.	8088 has4-bit instruction queue.
The 8086 has BHE (Bank high enable)	The 8088 has SSO status signal.
The 8086 can read or write 8-bit or 16-bit data at a time.	The 8088 can read/write 8-bit data at a time.
The I/O voltages level for 8086 is measured at 2.5 mA.	The I/O voltages level for 8086 is measured at 2 mA.
The 8086 draws maximum supply current of 360mA.	The 8088 draws maximum supply current of 340mA.

4.12. Architecture of Intel 8088 microprocessor