

MP & MC

Doc	Topic
2	Difference between cores vs threads Define: hyper threading, multithreading, single-core cpu, multi-core cpu
3	Intel 8259 pic: features, block diagram, advantage-disadvantage
4	Intel 8259 Pic : Functional Description, Interrupt Sequence, Cascading The 8259a
5	Microprocessor: Definition, function, block diagram, working procedure, Microcomputer, organization of microcomputer based system, difference between microprocessor and microcomputer, 8086: Feature, addressing mode, instruction set
6	-Draw and explain the block diagram of the programmable peripheral interface (8255A). -Explain 8255A I/O Operating Modes -Explain BSR Mode of the programmable peripheral interface (8255A) with necessary diagrams. -Explain 8255A Control Word and Control Register with necessary diagram. -What is the need of the programmable interrupt controller (8259A)? -Draw and explain the block diagram of 8259A -features of 8259 -8259 Internal Block Diagram -Explain I/O interfacing Methods -Comparison of Memory-Mapped I/O and Peripheral Mapped I/O
7	-Draw and explain 8086 Logical Block diagram, -Explain 8086 Registers -Draw the format of a Flag register of an 8086 microprocessor -Explain Segmentation in 8086 -How to calculate physical address from segment address -How is a 20-bit physical address obtained if data bus is of 16-bit -Explain 8086 pin diagram -Describe the architecture of the 80286 with a neat block diagram. -Explain Register Organization of 80286 -80286 Flag Register, Machine Status Word (MSW) Register -Real Mode, Protected Virtual Address Mode (PVAM) -Differentiate between the real mode and protected mode of an 80286 microprocessor. -Page Translation Mechanism in 80386 -Features of 80386 -Explain the architecture of the 80386 with a neat block diagram. -Register organization of 80386 -Register organization of 80386 -Explain features of 80486, -Features of Pentium Processor
8	-DMA controller 8257/8237: feature, operation, Modes of DMAC, advantage-disadvantage -Define paging, How paging works
9	- 8086 Interrupts, types and difference

10	All topics
12	-feature and functional block diagram of 8051 microcontroller -execution of an interrupt -memory organization:
13	PLC: features, -plc programming languages, define ladder diagram (ld) with of ladder symbol/ element, -logical function: and, or

Solutions:

2.1 Difference between cores vs threads

Define: hyper threading, multithreading, single-core cpu, multi-core cpu

The Difference between Cores vs Threads

The main difference between cores and threads is that a core is an individual physical processing unit, while threads are virtual sequences of instructions.

The performance of a computer depends on the number of cores AND the threading technique. For example, a computer with a quad-core CPU will benefit from multithreading as it utilizes several cores. Meanwhile, a hyperthreading technique can further increase the number of threads that can be active by splitting a single core into two virtual cores, allowing them to run multiple threads.

The trade-off to such strength is that it often comes with a cost, consumes more power, and may only sometimes result in an overall improvement in performance. It's critical to have comprehensive knowledge not only about the technical specifications of the CPUs you're considering but also about how your organization will be using them.

Difference between Core vs. Threads

Parameters	Core	Threads
Definition	CPU cores mean the actual hardware component.	Threads refer to the virtual component which manages the tasks.

Process	The CPU is fed tasks from a thread. Therefore, it only accesses the second thread when the information sent by the first thread is not reliable.	There are many different variations of how the CPU can interact with multiple threads.
Implementation	Achieved through interleaving operation	Performed through using multiple CPU'S
Benefit	Increase the amount of work accomplished at a time.	Improve throughput, computational speed-up.
Make use of	Core uses content switching	Uses multiple CPUs for operating numerous processes.
Processing units required	Requires only signal process units.	Requires multiple processing units.
Example	Running multiple applications at the same time.	Running a web crawler on a cluster.

Hyperthreading

Hyperthreading further increases the performance of multi-core processors by allowing them to execute two threads concurrently. The process works by sharing the resources of each core between two threads. That way, both can be active at the same time while accessing the same cache memory, registers, and execution units.

Multithreading

Multithreading is a process during which a single processor executes multiple threads simultaneously. This allows the processor to divide tasks into separate threads and run them in parallel, thereby increasing the utilization of available system resources and improving performance.

Why It Pays to Understand CPU Cores and Threads

Understanding the difference between cores and threads can help you make informed decisions about how to maximize performance. Let's start with a few key concepts:

Cores are physical processing units.

Threads are virtual sequences of instructions given to a CPU.

Multithreading allows for better utilization of available system resources by dividing tasks into separate threads and running them in parallel.

Hyperthreading further increases performance by allowing processors to execute two threads concurrently.

Single-Core CPU

Single-core CPUs are cheaper than multi-core CPUs, and they consume less power. This makes them great options for laptops, tablets, and other mobile devices. They also work well if the tasks you need to complete are relatively simple or don't require too much multitasking. On the other hand, they will lack the performance of a multi-core CPU.

Multi-Core CPU

A multi-core CPU is ideal for multitasking and running applications that require high levels of performance or processing large datasets. This type of processor can divide tasks among the cores, allowing each to handle its own piece. A multi-core CPU will require more energy and supporting hardware to support its power.

3. Intel 8259 pic: features, block diagram, advantage-disadvantage

8259 PIC Microcontroller:

Intel 8259 is a Programmable Interrupt Controller (PIC). There are 5 hardware interrupts and 2 hardware interrupts in Intel 8085 and Intel 8086 microprocessors respectively. But by connecting Intel 8259 with these microprocessors, we can increase their interrupt handling capability. Intel 8259 combines the multi-interrupt input sources into a single interrupt output. Interfacing of single PIC provides 8 interrupts inputs from IR0-IR7. For example, Interfacing of 8085 and 8259 increases the interrupt handling capability of 8085 microprocessor from 5 to 8 interrupt levels.

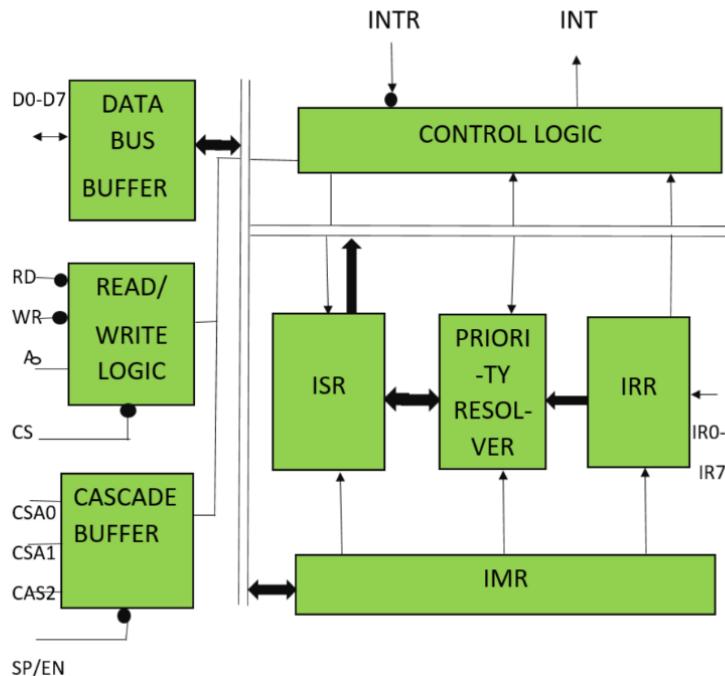
Features of Intel 8259 PIC are as follows:

1. Intel 8259 is designed for Intel 8085 and Intel 8086 microprocessors.
2. It can be programmed either in level triggered or in edge triggered interrupt level.
3. We can mask individual bits of interrupt request register.
4. We can increase interrupt handling capability upto 64 interrupt level by cascading further 8259 PICs.

5. Clock cycle is not required.

1.

Block Diagram of 8259 PIC microprocessor –



The Block Diagram consists of 8 blocks which are – Data Bus Buffer, Read/Write Logic, Cascade Buffer Comparator, Control Logic, Priority Resolver and 3 registers- ISR, IRR, IMR.

1. Data bus buffer – This Block is used as a mediator between 8259 and 8085/8086 microprocessor by acting as a buffer. It takes the control word from the 8085 (let say) microprocessor and transfers it to the control logic of the 8259 microprocessor. After selection of Interrupt by 8259 microprocessor (based on priority of the interrupt), it transfer the opcode of the selected Interrupt and address of the Interrupt service subroutine to the other connected microprocessor. The data bus buffer consists of 8 bits represented as D0-D7 in the block diagram. Thus, shows that a maximum of 8 bits of data can be transferred at a time.

2. Read/Write logic – This block works only when the value of pin CS is low (as this pin is active low). This block is responsible for the flow of data depending upon the inputs of RD and WR. These two pins are

active low pins used for read and write operations.

3. Control logic – It is the center of the PIC and controls the functioning of every block. It has pin INTR which is connected with other microprocessor for taking interrupt request and pin INT for giving the output. If 8259 is enabled, and the other microprocessor Interrupt flag is high then this causes the value of the output INT pin high and in this way 8259 responds to the request made by another microprocessor.

4. Interrupt request register (IRR) – It stores all the interrupt level which are requesting for Interrupt services.

5. Interrupt service register (ISR) – It stores the interrupt level which are currently being executed.

6. Interrupt mask register (IMR) – It stores the interrupt level which have to be masked by storing the masking bits of the interrupt level.

7. Priority resolver – It examines all the three registers and set the priority of interrupts and according to the priority of the interrupts, the interrupt with highest priority is set in the ISR register. Also, it reset the interrupt level which has already been serviced in IRR.

8. Cascade buffer – To increase the Interrupt handling capability, we can further cascade more number of pins by using cascade buffer. So, during increment of interrupt capability, CSA lines are used to control multiple interrupt structure. SP/EN (Slave program/Enable buffer) pin is when set to high, works in master mode else in slave mode. In Non Buffered mode, SP/EN pin is used to specify whether 8259 work as master or slave and in Buffered mode,

SP/EN pin is used as an output to enable data bus.

Advantages:

Interrupt Management: The 8259 PIC is designed to handle interrupts efficiently and effectively, allowing for faster and more reliable processing of interrupts in a system.

Flexibility: The 8259 PIC is programmable, meaning that it can be customized to suit the specific needs of a given system, including the number and type of interrupts that need to be managed.

Compatibility: The 8259 PIC is compatible with a wide range of microprocessors, making it a popular choice for managing interrupts in many different systems.

Multiple Interrupt Inputs: The 8259 PIC can manage up to 8 interrupt inputs, allowing for the management of complex systems with multiple devices.

Ease of Use: The 8259 PIC includes simple interface pins and registers, making it relatively easy to use and program.

Disadvantages:

Cost: While the 8259 PIC is relatively affordable, it does add cost to a system, particularly if multiple PICs are required.

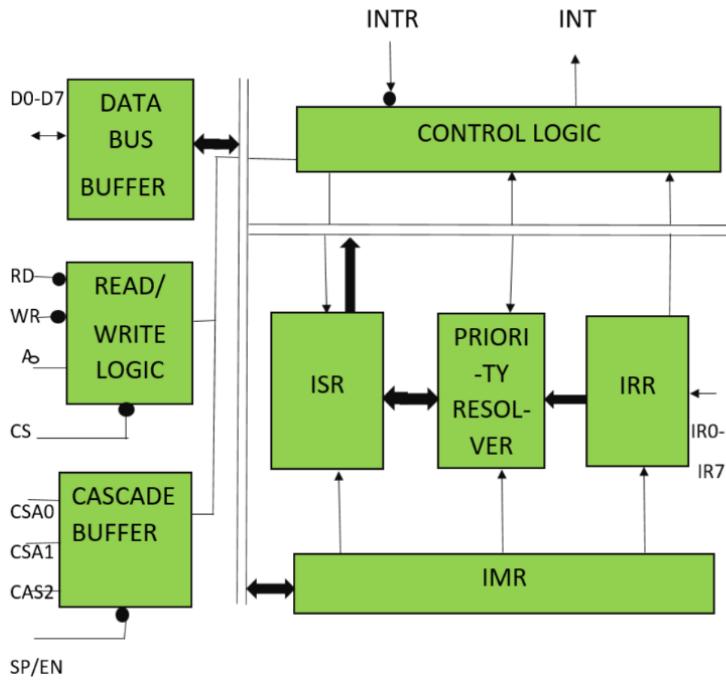
Limited Number of Interrupts: The 8259 PIC can manage up to 8 interrupt inputs, which may be insufficient for some applications.

Complex Programming: Although the interface pins and registers of the 8259 PIC are relatively simple, programming the 8259 can be complex, requiring careful attention to interrupt prioritization and other parameters.

Limited Functionality: While the 8259 PIC is a useful peripheral for interrupt management, it does not include more advanced features, such as DMA (direct memory access) or advanced error correction.

4. Intel 8259 Pic : Functional Description, Interrupt Sequence, Cascading The 8259a

Block Diagram of 8259 PIC microprocessor –



The Block Diagram consists of 8 blocks which are – Data Bus Buffer, Read/Write Logic, Cascade Buffer Comparator, Control Logic, Priority Resolver and 3 registers- ISR, IRR, IMR.

1. Data bus buffer – This Block is used as a mediator between 8259 and 8085/8086 microprocessor by acting as a buffer. It takes the control word from the 8085 (let say) microprocessor and transfers it to the control logic of the 8259 microprocessor. After selection of Interrupt by 8259 microprocessor (based on priority of the interrupt), it transfer the opcode of the selected Interrupt and address of the Interrupt service subroutine to the other connected microprocessor. The data bus buffer consists of 8 bits represented as D0-D7 in the block diagram. Thus, shows that a maximum of 8 bits of data can be transferred at a time.

2. Read/Write logic – This block works only when the value of pin CS is low (as this pin is active low). This block is responsible for the flow of data depending upon the inputs of RD and WR. These two pins are active low pins used for read and write operations.

3. Control logic – It is the center of the PIC and controls the functioning of every block. It has pin INTR which is connected with other microprocessor for taking interrupt request and pin INT for giving the output. If 8259 is enabled, and the other microprocessor Interrupt flag is high then this causes the value of the output INT pin high and in this

way 8259 responds to the request made by another microprocessor.

4. Interrupt request register (IRR) – It stores all the interrupt level which are requesting for Interrupt services.

5. Interrupt service register (ISR) – It stores the interrupt level which are currently being executed.

6. Interrupt mask register (IMR) – It stores the interrupt level which have to be masked by storing the masking bits of the interrupt level.

7. Priority resolver – It examines all the three registers and set the priority of interrupts and according to the priority of the interrupts, the interrupt with highest priority is set in the ISR register. Also, it reset the interrupt level which has already been serviced in IRR.

8. Cascade buffer – To increase the Interrupt handling capability, we can further cascade more number of pins by using cascade buffer. So, during increment of interrupt capability, CSA lines are used to control multiple interrupt structure. SP/EN (Slave program/Enable buffer) pin is when set to high, works in master mode else in slave mode. In Non Buffered mode, SP/EN pin is used to specify whether 8259 work as master or slave and in Buffered mode,

SP/EN (Slave program/Enable buffer) pin is when set to high, works in master mode else in slave mode. In Non Buffered mode, SP/EN pin is used to specify whether 8259 work as master or slave and in Buffered mode, SP/EN pin is used as an output to enable data bus.

Interrupt Sequence of 8259A:

The main feature of 8259A is its programmability and the interrupt routine addressing capability. The addressing capability allows direct or indirect jumping to specific ISR based on the interrupt number and the interrupting device. The normal sequence of events during an interrupt depends on the type of the CPU used and are given below:

 
UCC5617DWP Data
UCC5617DWP Bus Terminator Price & Stock

Ad findchips.com

[Visit Site](#)

1. After a bit in the IRR is set to '1' it is compared with the corresponding mask bit in IMR. If the mask bit is 0, the request is passed to the priority resolver, but if it is 1, the request is blocked.
2. When an interrupt is input to the priority resolver its priority is examined and. If according to the current state of the priority resolver the interrupt is to be sent to the CPU, the INT line is activated.
3. Assuming that the IF flag in the CPU was set to 1, the CPU will enter its interrupt sequence at the completion of the current instruction and return two negative pulses over the INTA' line
4. Upon receiving the 1st INTA' pulse, the IRR latches are disabled, so that the IRR will ignore further signals on the IR7-IR0 lines. This state is maintained until the end of the 2nd INTA' pulse.
5. Also the 1st INTA' pulse will cause the appropriate ISR bits to be set and the corresponding IRR bit to be cleared.
6. The second INTA' pulse cause the current content of ICW2 to be placed on D7-D0, and the CPU uses this byte as the interrupt type.
7. Now if the automatic end of interrupt (AE0I) bit ICW4 is 1, at the end of the second INTA' pulse the ISR bit that was set by 1st INTA' pulse is cleared; otherwise, the ISR bit is not cleared until the proper end of interrupt (EOI) command is sent to OCW2

Cascade buffer – To increase the Interrupt handling capability, we can further cascade more number of pins by using a cascade buffer. So, during increment of interrupt capability, CSA lines are used to control multiple interrupt structure. SP/EN (Slave program/Enable buffer) pin is high, works in master mode else in slave mode. In Non Buffered mode, SP/EN pin is used to specify whether 8259 work as master or slave and in Buffered mode,

5.

**-Microprocessor: Definition, function, block diagram, working procedure,
-Microcomputer, organization of microcomputer based system, difference between microprocessor and microcomputer,**

-8086: Feature, addressing mode, instruction set

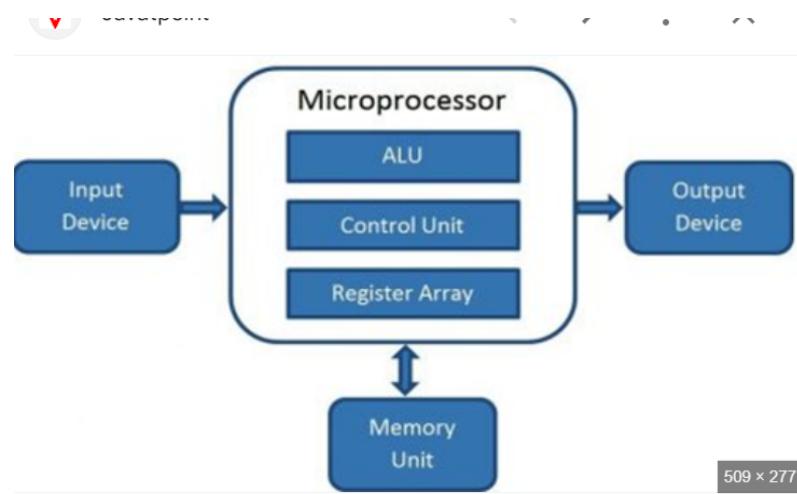
-Microprocessor: A microprocessor is a computer processor where the data processing logic and control is included on a single integrated circuit (IC), or a small number of ICs. The microprocessor contains the arithmetic, logic, and control circuitry required to perform the functions of a computer's central processing unit (CPU).

Functions of Microprocessor

Among various functions of microprocessor some are as follows

1. Controlling all other parts of the machine and sending timing signals.
2. Transferring data between memory and I/O devices
3. Fetching data and instructions from memory
4. Decoding instruction
5. Performing arithmetical and logical operations
6. Executing programs stored in memory
7. Performing communication among the I/O devices etc.

Block diagram of microprocessor:



Working of Microprocessor

The microprocessor follows a sequence to execute the instruction: Fetch, Decode, and then Execute.

Initially, the instructions are stored in the storage memory of the computer in sequential order. The microprocessor fetches those instructions from the stored area (memory), then decodes it and executes those instructions till STOP instruction is met.

Then, it sends the result in binary form to the output port. Between these processes, the register stores the temporary data and ALU (Arithmetic and Logic Unit) performs the computing functions.

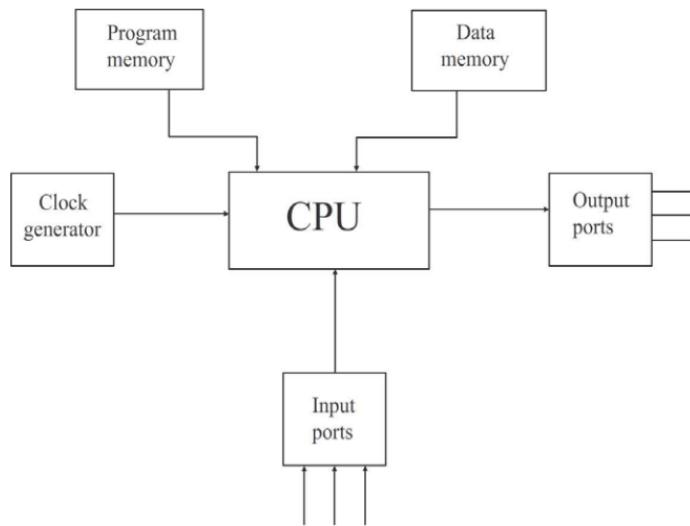
Microcomputer:

A microcomputer is a computer with a central processing unit (CPU) as a microprocessor. Designed for individual use, a microcomputer is smaller than a mainframe or a minicomputer.

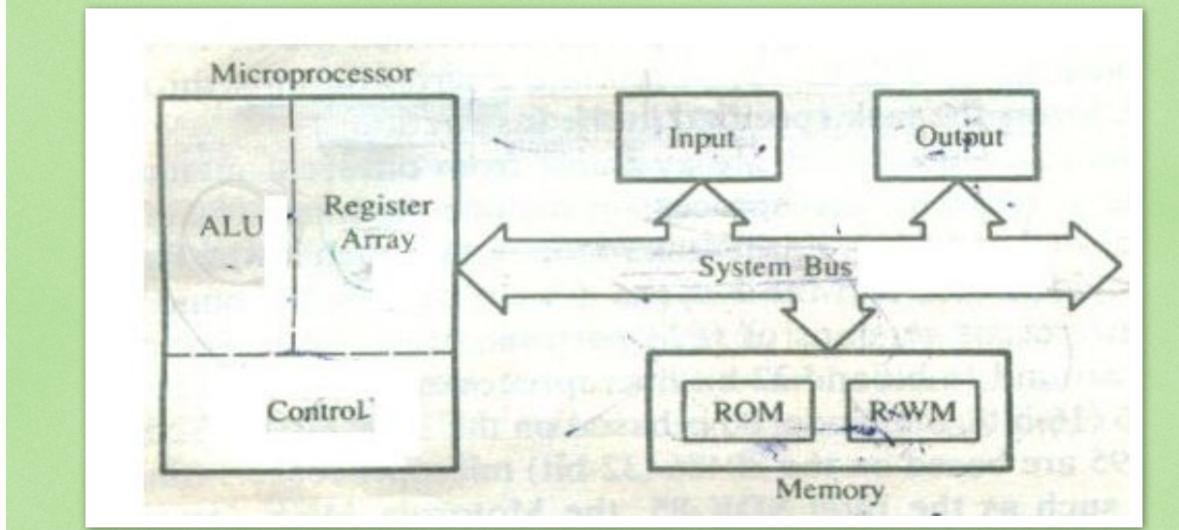
The basic components of a microcomputer are:

- 1) CPU
- 2) Program memory
- 3) Data memory
- 4) Output ports
- 5) Input ports
- 6) Clock generator.

These components are shown in figure below:



Basic Organization of Microcomputer



1. MP

i. ALU

- This unit executes all arithmetic and logical operations as specified by instruction set; and produces output.
- The results of addition, subtraction, and logical operations (AND, OR, XOR) are stored in the registers or in memory unit or sent to output unit.

ii. Register unit

- Consists of various registers.
- Used for temporary storage of data during execution of data.

iii. CU

- Controls the operations of different instructions.
- Provides necessary timing and control signals to all the operations in the MP and peripherals including memory.

2. Memory

- Stores binary information such as instruction and data, and provide these information to MP when required.
- To execute programs, the MP reads data and instructions from memory and performs the computing operations.

3. System bus

- The system bus is a communication path between MP and peripherals.
- It is used to carry data, address and control signals.
- It consists:

Data bus: carries data

Address bus: carries address

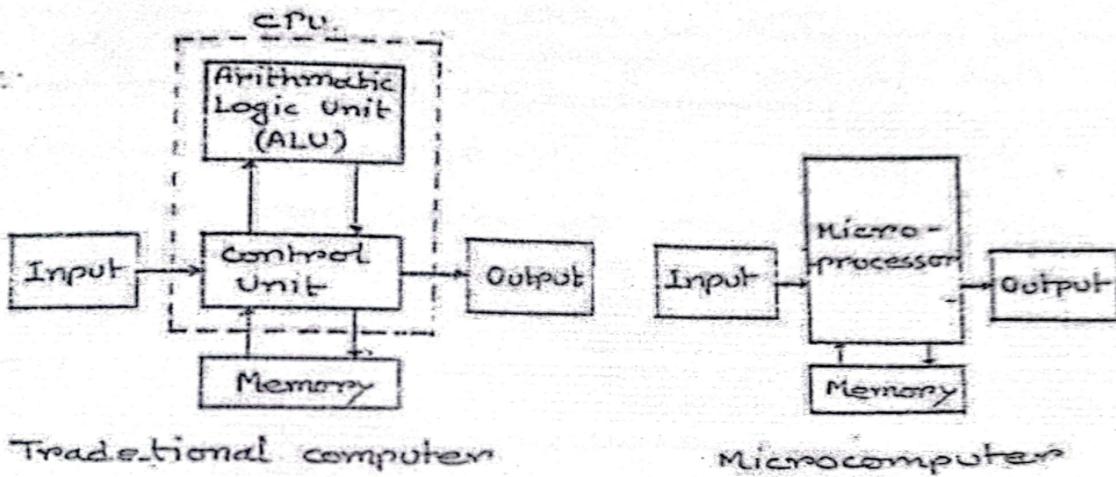
Control bus: carries control signals

4. I/O bus

- Input unit is used to input instruction or data to the MP externally.
- Output unit is used to carry out the information from the MP unit.

Basic differences between a Microcomputer and a traditional computer:

1. The block diagram of traditional computer and microcomputer:



2. The CPU of a microcomputer possesses only microprocessor but the CPU of a traditional computer possesses not only microprocessor but also some additional device.

3. Microcomputer is a version of computer but traditional computer is the latest version of the microcomputer.

4. For traditional computer cost is less,
for microcomputer cost is high.

5. The size of a traditional computer
larger than that of a microcomputer.

~~Basic difference between microprocessor &~~
~~microcomputer:~~

1. Microprocessor is a part of a
microcomputer but on the other hand,
microcomputer is a complete computer simi-
to any other computer.

Comparisons of Microcontroller and Microprocessor:

Microcontroller	Microprocessor
A microprocessor is a general purpose device which is called a CPU	A microcontroller is a dedicated chip which is also called single chip computer.
A microprocessor do not contain on chip I/O Ports, Timers, Memories etc.	A microcontroller includes RAM, ROM, serial and parallel interface, timers, interrupt circuitry (in addition to CPU) in a single chip.
Microprocessors are most commonly used as the CPU in microcomputer systems.	Microcontrollers are used in small, minimum component designs performing control-oriented applications.

Microprocessor instructions are mainly nibble or byte addressable.	Microcontroller instructions are both bit addressable as well as byte addressable.
Microprocessor based system design is complex and expensive	Microcontroller based system design is rather simple and cost effective
The Instruction set of microprocessor is complex with large number of instructions.	The instruction set of a Microcontroller is very simple with less number of instructions. For, ex: PIC microcontrollers have only 35 instructions.
A microprocessor has zero status flag.	A microcontroller has no zero flag.

Difference between microprocessor and microcomputer:chok akare karo kache likha thakle amake dio

Features of 8086

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 prefetch 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.

Microprocessor - 8086 Addressing Modes

The different ways in which a source operand is denoted in an instruction is known as **addressing modes**. There are 8 different addressing modes in 8086 programming –

Immediate addressing mode

The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.

Example

```
MOV CX, 4929 H, ADD AX, 2387 H, MOV AL, FFH
```

Register addressing mode

It means that the register is the source of an operand for an instruction.

Example

```
MOV CX, AX ; copies the contents of the 16-bit AX register into  
           ; the 16-bit CX register),
```

```
ADD BX, AX
```

Direct addressing mode

The addressing mode in which the effective address of the memory location is written directly in the instruction.

Example

```
MOV AX, [1592H], MOV AL, [0300H]
```

Register indirect addressing mode

This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI.

Example

```
MOV AX, [BX] ; Suppose the register BX contains 4895H, then the contents  
           ; 4895H are moved to AX
```

```
ADD CX, {BX}
```

Based addressing mode

In this addressing mode, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement.

Example

MOV DX, [BX+04], ADD CL, [BX+08]

Indexed addressing mode

In this addressing mode, the operands offset address is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.

Example

MOV BX, [SI+16], ADD AL, [DI+16]

Based-index addressing mode

In this addressing mode, the offset address of the operand is computed by summing the base register to the contents of an Index register.

Example

ADD CX, [AX+SI], MOV AX, [AX+DI]

Based indexed with displacement mode

In this addressing mode, the operands offset is computed by adding the base register contents. An Index registers contents and 8 or 16-bit displacement.

Example

MOV AX, [BX+DI+08], ADD CX, [BX+SI+16]

Microprocessor - 8086 Instruction Sets

The 8086 microprocessor supports 8 types of instructions –

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Bit Manipulation Instructions
4. String Instructions
5. Program Execution Transfer Instructions (Branch & Loop Instructions)
6. Processor Control Instructions
7. Iteration Control Instructions
8. Interrupt Instructions

Data Transfer Instructions: These instructions are used to transfer the data from the source operand to the destination operand.

**MOV,
PPUSH ,POP,PUSHA ,POPA ,XCHG ,XLAT**

Arithmetic Instructions: These instructions are used to perform arithmetic operations like addition, subtraction, multiplication, division, etc.

Bit Manipulation Instructions: These instructions are used to perform operations where data bits are involved, i.e. operations like logical, shift, etc.

String Instructions: String is a group of bytes/words and their memory is always allocated in a sequential order.

Program Execution Transfer Instructions (Branch and Loop Instructions): These instructions are used to transfer/branch the instructions during an execution.

Processor Control Instructions: These instructions are used to control the processor action by setting/resetting the flag values.

Iteration Control Instructions: These instructions are used to execute the given instructions for number of times.

Interrupt Instructions: These instructions are used to call the interrupt during program execution.

6. Draw and explain the block diagram of the programmable peripheral interface (8255A).

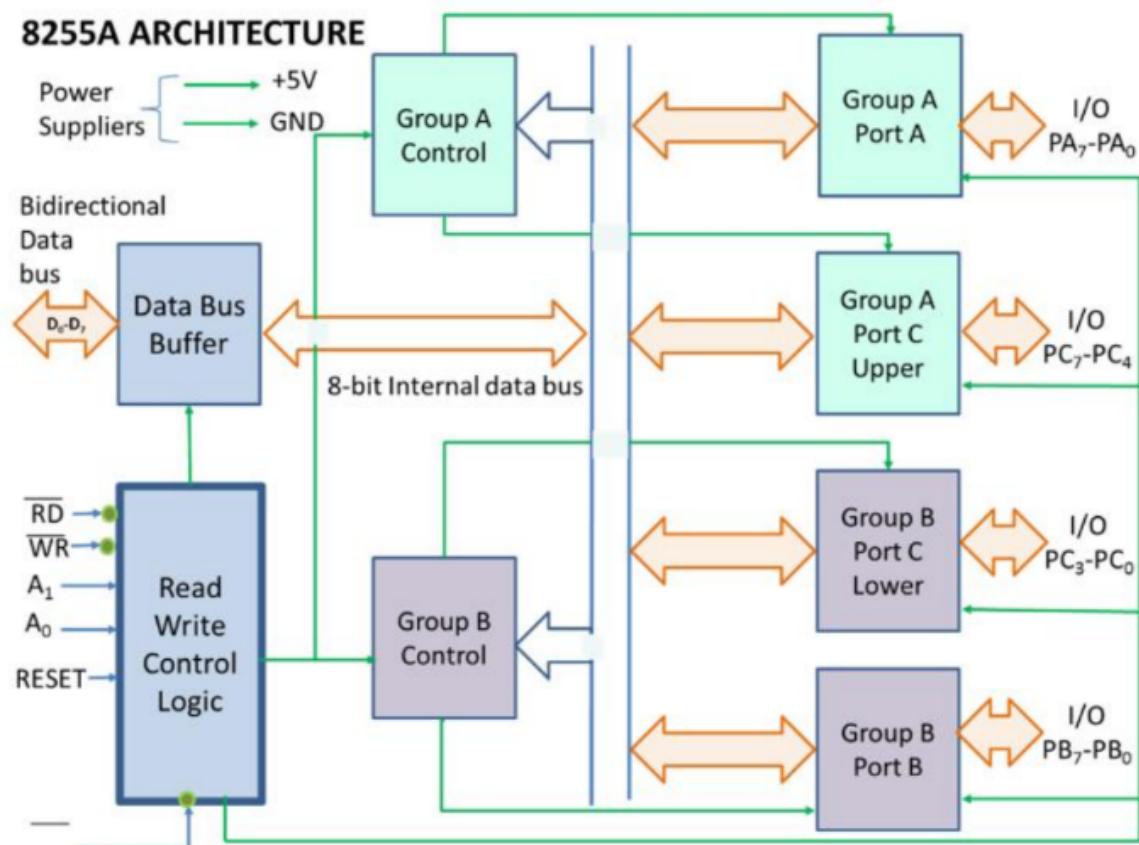


Figure: 8255A Architecture

Figure. 8255A Architecture

Read Write Control Logic

RD (READ)	This is an active low signal that enables Read operation. When signal is low MPU reads data from selected I/O port of 8255A
WR (WRITE)	This is an active low signal that enables Write operation. When signal is low MPU writes data into selected I/O port or control register
RESET	This is an active high signal, used to reset the device. That means clear control registers
CS	<p>This is Active Low signal.</p> <p>When it is low, then data is transfer from 8085</p> <p>CS signal is the master Chip Select.</p> <p>A₀ and A₁ specify one of the I/O ports or control register</p>

CS	A1	A0	Selected
0	0	0	POR TA
0	0	1	POR TB
0	1	0	POR TC
0	1	1	Control Register
1	X	X	8255A is not selected

Data Bus Buffer

- This three-state bi-directional 8-bit buffer is used to interface the 8255 to the system data bus.
- Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU.
- Control words and status information are also transferred through the data bus buffer.

Group A and Group B Controls

- The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255.
- The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255.
- Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Ports A, B, and C

- The 8255 contains three 8-bit ports (A, B, and C).
- All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255.
- Port A One 8-bit data output latch/buffer and one 8-bit data input latch.
- Both "pull-up" and "pull-down" bus-hold devices are present on Port A.
- Port B One 8-bit data input/output latch/buffer and one 8-bit data input buffer.
- Port C One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control.
- Each 4-bit port contains a 4-bit latch and it can be used for the control signal output and status signal inputs in conjunction with ports A and B.

2. Explain 8255A I/O Operating Modes

Ans. 8255A has three different I/O operating modes:

1. Mode 0
2. Mode 1
3. Mode 2

Mode 0

- Simple I/O for port A,B and C
- In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports.
- Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched.
- Ports do not have handshake or interrupt capability.

Mode 1: Input or Output with Handshake

- Handshake signal are exchanged between MPU and peripheral prior to data transfer.
- In this mode, Port A and B is used as 8-bit I/O ports.
- Mode 1 is a handshake Mode whereby ports A and/or B use bits from port C as handshake signals.
- In the handshake mode, two types of I/O data transfer can be implemented: status check and interrupt.
- Port A uses upper 3 signals of Port C: PC3, PC4, PC5
- Port B uses lower 3 signals of Port C : PC0, PC1, PC2
- PC6 and PC7 are general purpose I/O pins

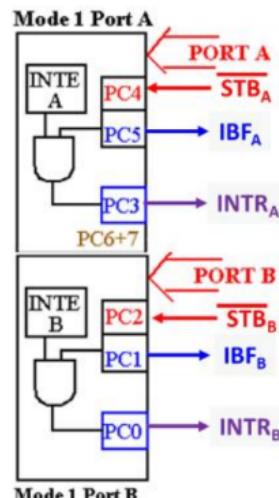


Figure: Mode1 Input Handshake

STB (Strobe Input):

- This active low signal is generated by a peripheral device to indicate that, it has transmitted a byte of data. The 8255A, in response to **STB**, generates **IBF** and **INTR**.

IBF (Input Buffer Full)

This signal is acknowledged by 8255A to indicate that the input latch has received the data byte. It will get reset when the MPU reads the data.

INTR(Interrupt Request)

This is an output signal that may be used to interrupt the MPU. This signal is generated if STB, IBF and INTE (internal flip-flop) are all at logic 1. It will get reset by the falling edge of RD

INTE(Interrupt Enable)

- This signal is an internal flip-flop, used to enable or disable the generation of INTR signal.
- The interrupt enable signal is neither an input nor an output; it is an internal bit programmed via the PC4 (port A) or PC2 (port B) bits.

Mode 2

- In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1.
- Port A uses five signals from Port C as handshake signals for data transfer.
- The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.

Handshake for port B.

3. Explain BSR Mode of the programmable peripheral interface (8255A) with necessary diagrams.

Ans.

- These are two basic modes of operation of 8255. I/O mode and Bit Set-Reset mode (BSR).
- In I/O mode, the 8255 ports work as programmable I/O ports, while in BSR mode only port C (PC0-PC7) can be used to set or reset its individual port bits.
- Under the I/O mode of operation, further there are three modes of operation of 8255, so as to support different types of applications, mode 0, mode 1 and mode 2.

8255A: BSR(Bit Set/Reset) Mode

- In this mode any of the 8-bits of port C can be set or reset depending on D₀ of the control word.
- The bit to be set or reset is selected by bit select flags D₃, D₂ and D₁ of the CWR (Control Word Register).
- BSR Control Word affects one bit at a time
- It does not affect the I/O mode

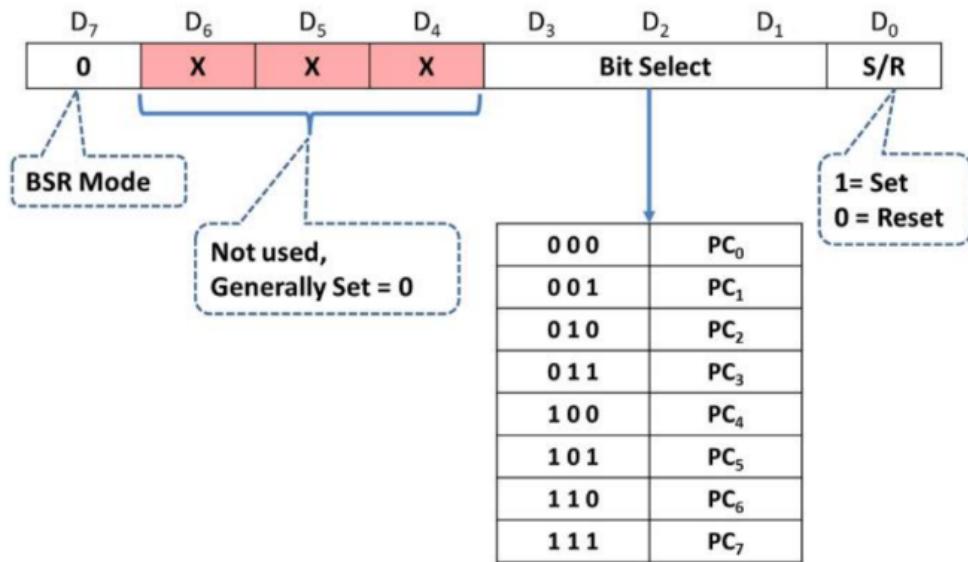


Figure: BSR Mode Control Word

4. Explain 8255A Control Word and Control Register with necessary diagram.

Ans. Control Register

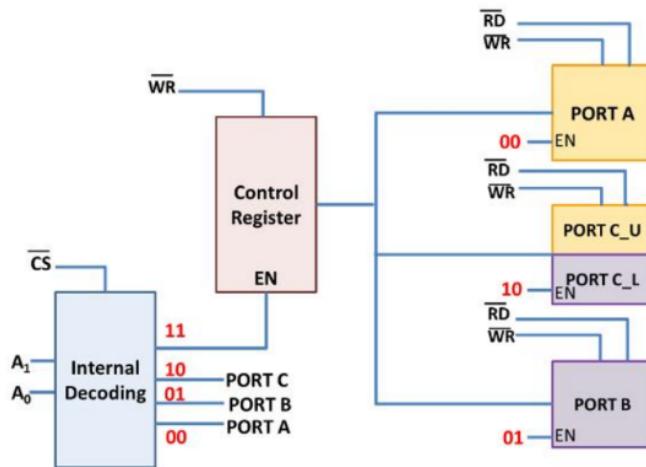


Figure: Control Register 8255A

Control Word: Content of Control register is known as Control Word.

- Control word specify an I/O function for each port this register can be.

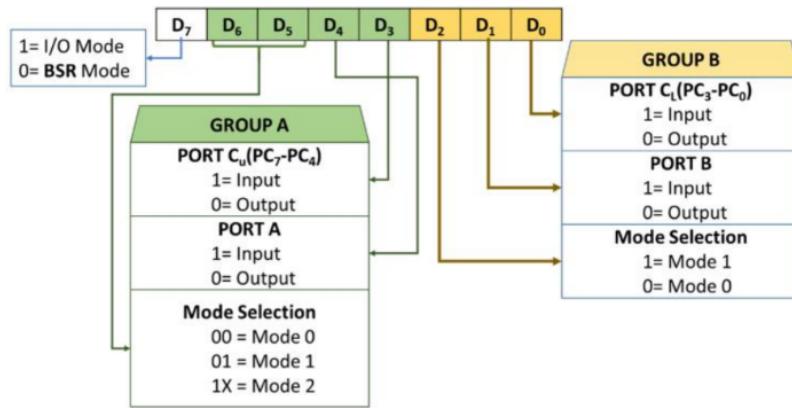


Figure:8255A Control Word

- Accessed to write a control word when A0 and A1 are at logic1, the register is not accessible for a read operation.
- Bit D7 of the control register either specifies the I/O function or the bit Set/Reset function, as classified in figure 1.
- If bit D7=0, bits D6-D0 determine I/O function in various mode, as shown in figure 4.
- If bit D7=0 port C operates in the bit Set/Reset (BSR) mode.
- The BSR control word does not affect the function of port A and B.

5. What is the need of the programmable interrupt controller (8259A)?

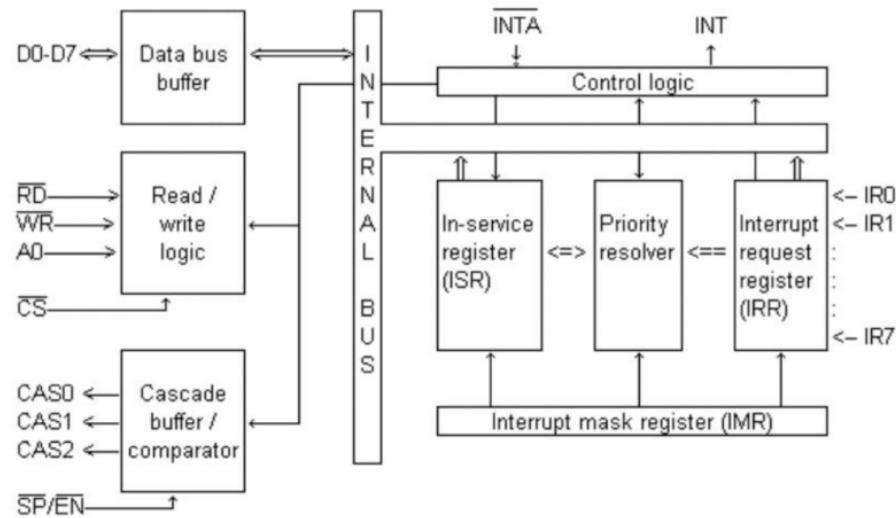
Draw and explain the block diagram of 8259A.

- Ans.**
- The Intel 8259 is a Programmable Interrupt Controller (PIC) designed for use with the 8085 and 8086 microprocessors.
 - The 8259 can be used for applications that use more than five numbers of interrupts from multiple sources.

The main features of 8259 are listed below

- Manage eight levels of interrupts.
- Eight interrupts are spaced at the interval of four or eight locations.
- Resolve eight levels of priority in fully nested mode, automatic rotation mode or specific rotation mode.
- Mask each interrupt individually.
- Read the status of pending interrupt, in-service interrupt, and masked interrupt.
- Accept either the level triggered or edge triggered interrupt

8259 Internal Block Diagram



Read/Write Logic

- It is typical R/W logic.
- When address line A0 is at logic 0, the controller is selected to write a command word or read status.
- The Chip Select logic and A0 determine the port address of controller.

Read/Write Logic

- It is typical R/W logic.
- When address line A0 is at logic 0, the controller is selected to write a command word or read status.
- The Chip Select logic and A0 determine the port address of controller.

Control Logic

- It has two pins: INT as output and INTA as input.
- The INT is connected to INTR pin of MPU

Interrupt Registers and Priority Resolver

1. Interrupt Request Register (IRR)
2. Interrupt In-Service Register (ISR)
3. Priority Resolver
4. Interrupt Mask Register (IMR)

Interrupt Request Register (IRR) and Interrupt In-Service Register (ISR)

- Interrupt input lines are handled by two registers in cascade – IRR and ISR
- IRR is used to store all interrupt which are requesting service.
- ISR is used to store all interrupts which are being serviced.

Priority Resolver

- This logic block determines the priorities of the bit set in IRR.
- IR₀ is having highest priority, IR₇ is having lowest priority

Interrupt Mask Register

- It stores bits which mask the interrupt lines to be masked
- IMR operates on the IRR.
- Masking of high priority input will not affect the interrupt request lines.

Cascade Buffer / Comparator

This block is used to expand the number of interrupt levels by cascading two or more 8259As.

8. Explain I/O interfacing Methods

Ans. There are two method of interfacing memory or I/O devices with the microprocessor are as follows:

- 1) I/O mapped I/O
- 2) Memory mapped I/O

1) I/O MAPPED I/O

- In this technique, I/O device is treated as an I/O device and memory as memory. Each I/O device uses eight address lines.
- If eight address lines are used to interface to generate the address of the I/O port, then 256 Input/output devices can be interfaced with the microprocessor.
- The 8085 microprocessor has 16 bit address bus, so we can either use lower order address lines (A₀ – A₇) or higher order address lines(A₈ – A₁₅) to address I/O devices. We used lower order address bus & address available on A₀ – A₇ will be copied on the address lines A₈ – A₁₅.
- In I/O mapped I/O, the complete 64 Kbytes of memory can be used to address memory locations separately as the address space is not shared with I/O devices.
- In this interface type, the data transfer is possible between accumulator (A) and I/O devices only. Arithmetic and logical operation are not possible directly.
- As 8 bit device address used, Address decoding is simple so less hardware is required.
- The separate control signals are used to access I/O devices and memory such as IOR, IOW for I/O port and MEMR, MEMW for memory hence memory location are protected from the I/O access.

2) MEMORY MAPPED I/O

- In this technique, I/O devices are treated as memory and memory as memory, hence the address of the I/O devices are as same as that of memory i.e. 16 bit for 8085 microprocessor.
- So, the address space of the memory i.e. 64 Kbytes will be shared by the I/O devices as well as by memory. All 16 address lines i.e. A0-A15 is used to address memory locations as well as I/O devices.
- The control signals MEMR and MEMW are used to access memory devices as well as I/O devices.

Comparison of Memory-Mapped I/O and Peripheral Mapped I/O

No	Characteristics	Memory mapped I/O	I/O mapped I/O
1	Device Address	16 bit	8 Bit
2	Control signals for inputs	MEMR & MEMW	IOR & IOW
3	Instruction Available	All memory related instruction : LDA; STA; LDAX; STAX; MOV M,R; ADD M; SUB M	IN and OUT instructions only
4	Data Transfer	Between any register and I/O devices.	Between I/O device and Accumulator only.
5	Maximum Numbers of I/Os Possible	Memory Map (64K) is shared between I/Os and System memory.	I/O Mapped is independent of memory map; 256 Input and 256 output devices can be connected.
6	Execution Speed	13 T-State (LDA, STA, ..) 7 T-State (MOV M,R)	10 T-State
7	Hardware Requirement	More hardware is needed to decode 16 bit address	Less hardware is needed to decode 8 bit address
8	Other Feature	Arithmetic and logical operations are directly performed with I/O devices.	Not available

7.

8086

1. Draw and explain 8086 Logical Block diagram

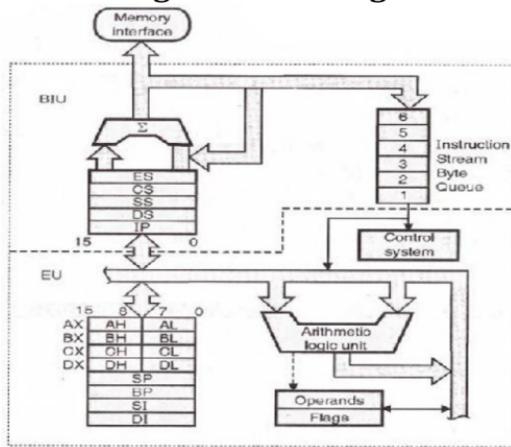


Figure: 8086 Architecture

- In 8086 CPU is divided into two independent functional parts BIU and EU.
- Dividing the work between these two units' speeds up the processing.

BIU (Bus Interface Unit)

Components of BIU

- Instruction queue
 - It holds the instruction bytes of the next instruction to be executed by EU
- Segment Registers
 - Four 16-bit register that provides powerful memory management mechanism
- ES (extra segment), CS (code segment), SS (stack segment), DS (data segment).
 - The size of each register is 64kb.
- Instruction pointer (IP)
- Instruction pointer (IP)
 - Register that holds 16-bit address or offset of next code byte within code segment
- Address Generation and bus control
 - Generation of 20-bit physical address

Task carried out by BIU

- Fetch instruction from memory
- Read/ Write instruction from / to the memory
- Input/ Output (I/O) of data from / to peripheral ports
- Write the data to memory.
- Address generation for memory reference
- Queuing of instruction (The instruction bytes are transferred to the instruction queue)
- Thus, BIU handles all transfer of data and address on the buses for Execution unit.
- BIU works in synchronous with machine cycles

➤ **EU (Execution Unit)**

Components of EU

- ALU (Arithmetic logic Unit)
Contains 16-bit ALU, that performs add, subtract, increment, decrement, compliment, shift binary numbers, AND, OR, XOR etc.
- CU (Control Unit)
Directs internal operation
- Flag Register
16-bit flag register
EU contains 9 active flags
- General Purpose Registers (GPR)
EU has 4 general purpose 16-bit register
i.e. AX, BX, CX, DX
each register is the combination of two 8-bit register
AH, AL, BH, BL, CH, CL, DH, DL where 'L' means Lower byte and 'H' means higher byte.
- Index Register
16-bit Register is SI (source index) and DI (destination index).
Both the register are used for string related operation and for moving block of memory from one location to the other.
- Pointers
16-bit Register.
i.e. SP (stack pointer), BP (base pointer)
BP : is used when we need to pass parameter through stack
SP: It always points to the top of the stack. Used for sequential access of stack segment.
- Decoder (instruction decoder)
Translates the instruction fetched from into series of action which EU carries out

Task carried out by EU

- Decodes the instruction
- It executes instructions (executes decoded instructions)
- Tells BIU from where to fetch the instruction
- Decodes instruction (decode the fetched instruction)
- EU takes care of performing operation on the data
- EU is also known as execution heart of the processor

2. Explain 8086 Registers

Ans. The 8086 microprocessor has a total of fourteen registers that are accessible to the programmer as follows:-

General Purpose Register

AX: - Accumulator register consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX.

- AX works as an intermediate register in memory and I/O operation.

- Accumulator is used for the instruction such as MUL and DIV.

BX: - Base register consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX.

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

CX: - Count register consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. Count register can be used in Loop, shift/rotate instructions and as a counter in string manipulation.

DX: - Data register can be used together with AX register to execute MUL and DIV instruction.

- Data register can be used as a port number in I/O operations.

Segment Register

Types of Segment registers are as follows:-

1. Code Segment (CS): The CS register is used for addressing a memory location in the Code Segment of the memory, where the executable program is stored.
2. Data Segment (DS): The DS contains most data used by program. Data are accessed in the Data Segment by an offset address or the content of other register that holds the offset address.
3. Stack Segment (SS): SS defined the area of memory used for the stack.
4. Extra Segment (ES): ES is additional data segment that is used by some of the string to hold the destination data

Pointer Registers

The pointers IP, BP, SP usually contain offsets within the code, data and stack segments respectively.

1. Stack Pointer (SP): SP is a 16-bit register pointing to program stack in stack segment.
2. Base Pointer (BP): 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.
3. Instruction Pointer (IP): IP is a 16-bit register pointing to next instruction to be executed.

Index registers

The Index Registers are as follows:-

1. Source Index (SI): SI is a 16-bit register used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.
2. Destination Index (DI): DI is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data addresses in string manipulation instructions.

Flag Registers

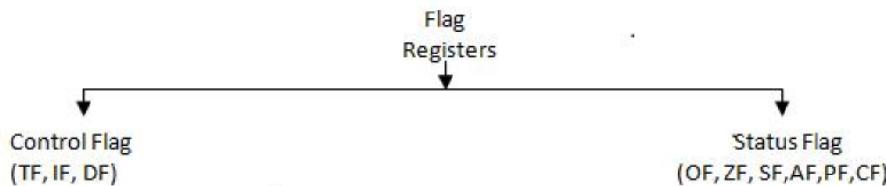
The 16-bit flag register of 8086 contains 9 active flags (six conditional & 3 control flags), other 7 flags are undefined.

3. Draw the format of a Flag register of an 8086 microprocessor.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

Figure:8086 Flag Register

- The 16-bit flag register of 8086 contains 9 active flags (six conditional & 3 control flags), other 7 flags are undefined.



- Status Flags: It indicates certain condition that arises during the execution. They are controlled by the processor.
- Control Flags: It controls certain operations of the processor. They are deliberately set/ reset by the user.

the user.

Control Flags

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

1. 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.

2. 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.

3. 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.

Status Flag

1. **Carry Flag (CF):** This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.
2. **Auxiliary Flag (AF):** 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 AF flag is set i.e. carry given by D3 bit to D4 is AF flag. This is not a general-purpose flag, it is used internally by the processor to perform Binary to BCD conversion.
3. **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.
4. **Zero Flag (ZF):** It is set; if the result of arithmetic or logical operation is zero else it is reset.
5. **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.
6. **Overflow Flag (OF):** It occurs when signed numbers are added or subtracted. OF=1 indicates that the result has exceeded the capacity of machine.

8.-DMA controller 8257/8237: feature, operation, Modes of DMAC, advantage-disadvantage

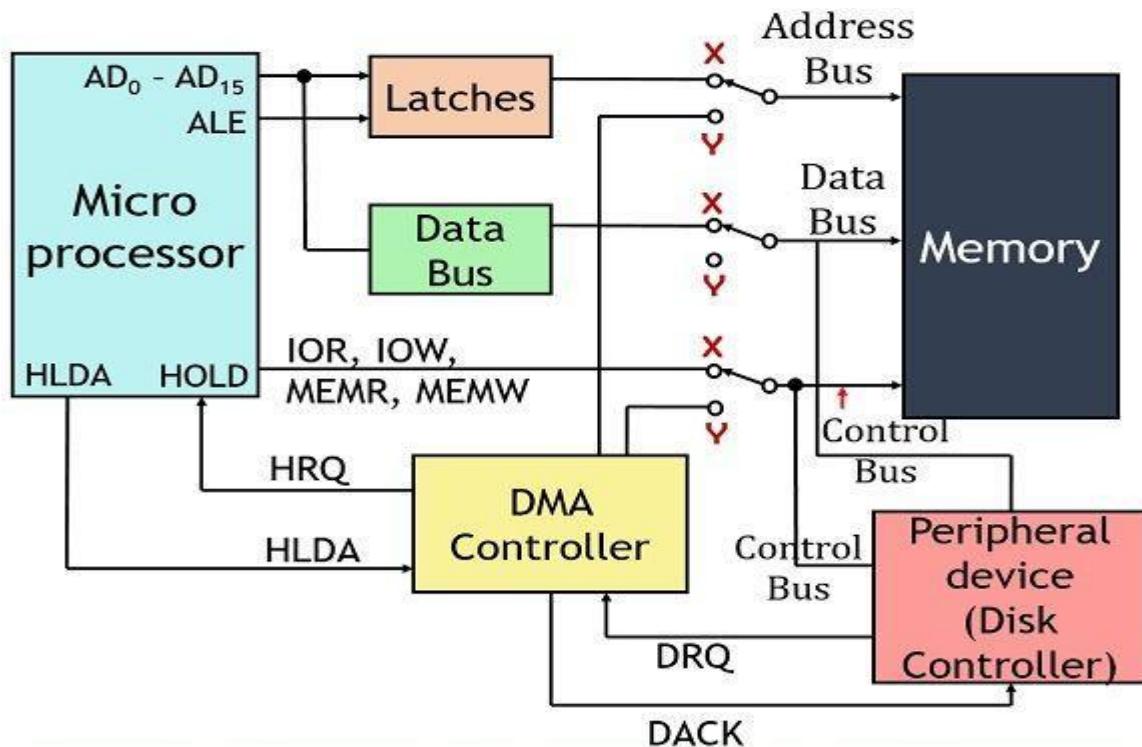
Direct memory access with DMA controller 8257/8237

Suppose any device which is connected to input-output port wants to transfer data to memory, first of all it will send input-output port address and control signal, input-output read to input-output port, then it will send memory address and memory write signal to memory where data has to be transferred. In normal input-output technique the processor becomes busy in checking whether any input-output operation is completed or not for next input-output operation, therefore this technique is slow.

This problem of slow data transfer between input-output port and memory or between two memory is avoided by implementing Direct Memory Access (DMA) technique. This is faster as the microprocessor/computer is bypassed and the control of address bus and data bus is given to the DMA controller.

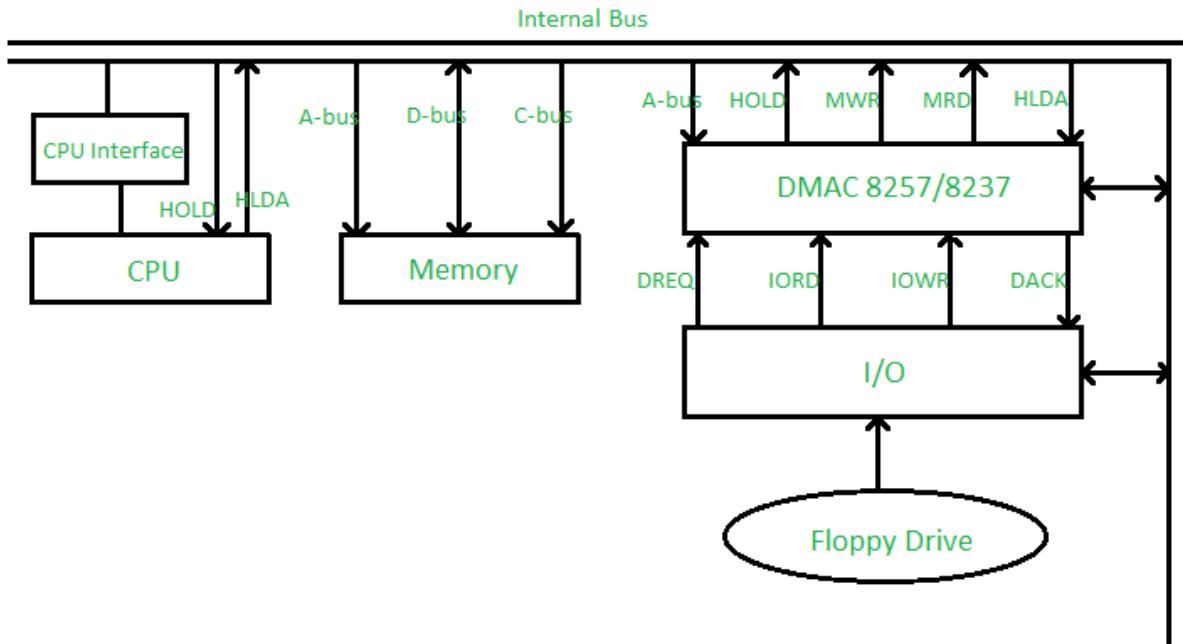
- HOLD – hold signal

- HLDA – hold acknowledgment
- DREQ – DMA request
- DACK – DMA acknowledgment



Operation of DMA Controller in a Microprocessor System

Electronics Desk

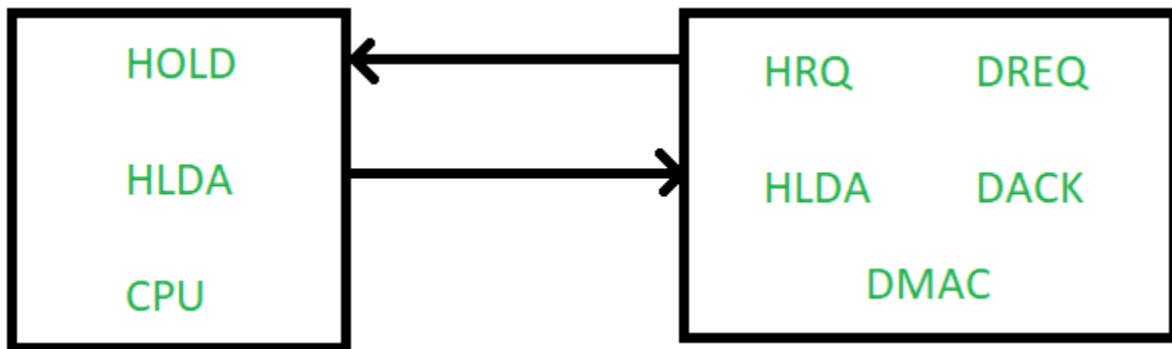


Suppose a floppy drive that is connected at input-output port wants to transfer data to memory, the following steps are performed:

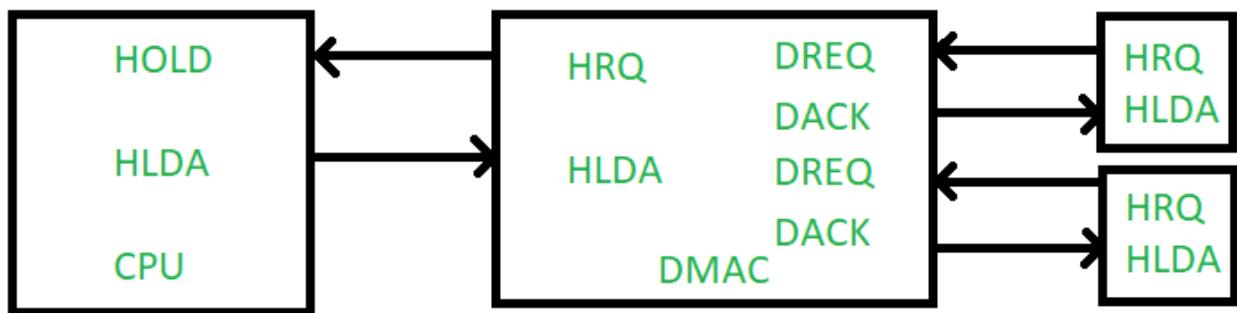
- **Step-1:** First of all the floppy drive will send a DMA request (DREQ) to the DMAC, it means the floppy drive wants its DMA service.
- **Step-2:** Now the DMAC will send a HOLD signal to the CPU.
- **Step-3:** After accepting the DMA service request from the DMAC, the CPU will send hold acknowledgment (HLDA) to the DMAC, it means the microprocessor has released control of the address bus the data bus to DMAC and the microprocessor/computer is bypassed during DMA service.
- **Step-4:** Now the DMAC will send one acknowledgement (DACL) to the floppy drive which is connected at the input-output port. It means the DMAC tells the floppy drive be ready for its DMA service.
- **Step-5:** Now with the help of input-output read and memory write signal the data is transferred from the floppy drive to the memory.

Modes of DMAC:

1. Single Mode – In this only one channel is used, means only a single DMA is connected to the bus system.



2. Cascade Mode – In this multiple channels are used, we can further cascade more number of DMAs.



Advantages:

Improved performance: DMA improves system performance by freeing up the CPU to perform other tasks while data is being transferred between memory and I/O devices. This allows for faster and more efficient data transfer.

Reduced CPU overhead: With DMA, the CPU is not required to be involved in data transfer, which reduces the CPU overhead and allows it to focus on other tasks. This is particularly useful in real-time systems where low latency and fast response times are important.

Support for high-bandwidth devices: DMA can support high-bandwidth devices such as graphics cards and network interfaces that require fast data transfer rates.

Efficient use of system resources: DMA allows multiple devices to access

memory simultaneously, which makes more efficient use of system resources.

Disadvantages:

Complexity: DMA requires specialized hardware and software to function, which can add to the complexity of a system. This can make it difficult to implement and troubleshoot.

Security risks: DMA can be a security risk if not properly configured or secured. Hackers can exploit vulnerabilities in DMA to gain unauthorized access to a computer system or steal data.

Limited control: Since the CPU is not involved in data transfer with DMA, it has limited control over the transfer process. This can lead to data corruption or errors if the transfer process is not properly managed.

Resource conflicts: DMA can lead to resource conflicts if multiple devices attempt to access memory simultaneously. This can cause system instability and performance issues if not properly managed

-Define paging, How paging works

Paging: Paging is a memory management scheme that eliminates the need for a contiguous allocation of physical memory. The process of retrieving processes in the form of pages from the secondary storage into the main memory is known as paging. The basic purpose of paging is to separate each procedure into pages. Additionally, frames will be used to split the main memory. This scheme permits the physical address space of a process to be non – contiguous

How does paging work?

Paging works by dividing the physical memory and the disk into equal-sized blocks called frames and sectors, respectively. Each frame and sector can hold one page of

data. When an application or a process requests a page of data, the operating system checks the page table to see if the page is in physical memory or on disk. If the page is in physical memory, the operating system accesses it directly. If the page is on disk, the operating system triggers a page fault and invokes the page fault handler. The page fault handler selects a frame in

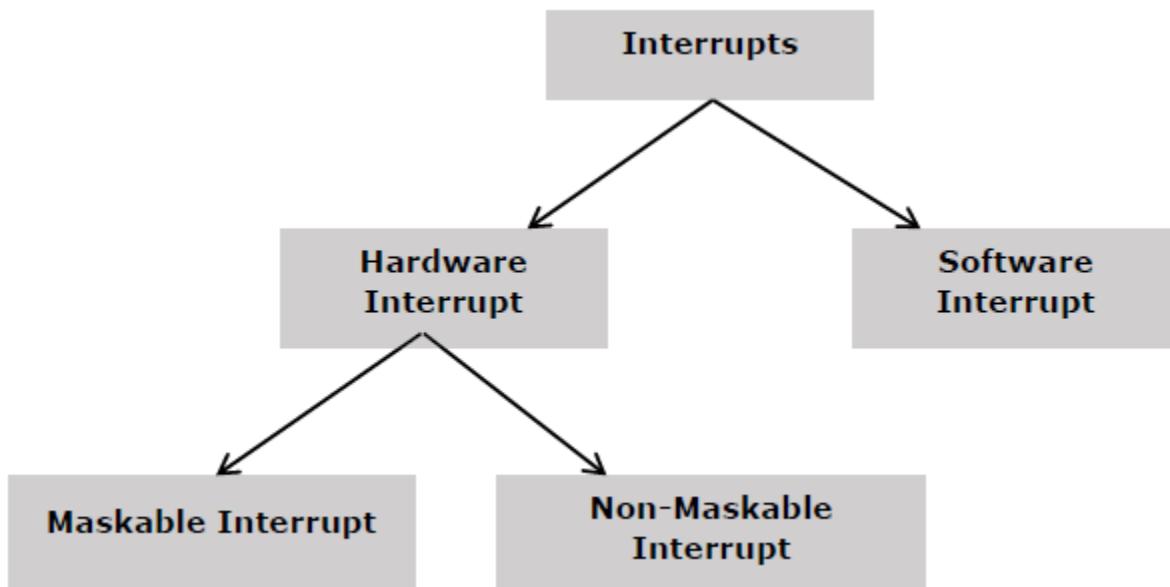
physical memory to replace, either by using a free frame or by evicting an existing page using a replacement algorithm. The page fault handler then copies the requested page from disk to the selected frame, updates the page table, and resumes the execution of the application or process.

9.- 8086 Interrupts, types and difference

Microprocessor - 8086 Interrupts

Interrupt is the method of creating a temporary halt during program execution and allows peripheral devices to access the microprocessor. The microprocessor responds to that interrupt with an **ISR** (Interrupt Service Routine), which is a short program to instruct the microprocessor on how to handle the interrupt.

The following image shows the types of interrupts we have in a 8086 microprocessor –



Hardware Interrupts

Hardware interrupt is caused by any peripheral device by sending a signal through a specified pin to the microprocessor.

The 8086 has two hardware interrupt pins, i.e. NMI and INTR. NMI is a non-maskable interrupt and INTR is a maskable interrupt having lower priority. One more interrupt pin associated is INTA called interrupt acknowledge.

NMI

It is a single non-maskable interrupt pin (NMI) having higher priority than the maskable interrupt request pin (INTR) and it is of type 2 interrupt.

When this interrupt is activated, these actions take place –

- Completes the current instruction that is in progress.
- Pushes the Flag register values on to the stack.
- Pushes the CS (code segment) value and IP (instruction pointer) value of the return address on to the stack.
- IP is loaded from the contents of the word location 00008H.
- CS is loaded from the contents of the next word location 0000AH.
- Interrupt flag and trap flag are reset to 0.

INTR

The INTR is a maskable interrupt because the microprocessor will be interrupted only if interrupts are enabled using set interrupt flag instruction. It should not be enabled using clear interrupt Flag instruction.

The INTR interrupt is activated by an I/O port. If the interrupt is enabled and NMI is disabled, then the microprocessor first completes the current execution and sends '0' on INTA pin twice. The first '0' means INTA informs the external device to get ready and during the second '0' the microprocessor receives the 8 bit, say X, from the programmable interrupt controller.

These actions are taken by the microprocessor –

- First completes the current instruction.
- Activates INTA output and receives the interrupt type, say X.
- Flag register value, CS value of the return address and IP value of the return address are pushed on to the stack.
- IP value is loaded from the contents of word location $X \times 4$
- CS is loaded from the contents of the next word location.
- Interrupt flag and trap flag is reset to 0

Software Interrupts

Some instructions are inserted at the desired position into the program to create interrupts. These interrupt instructions can be used to test the working of various interrupt handlers. It includes –

INT- Interrupt instruction with type number

It is 2-byte instruction. First byte provides the op-code and the second byte provides the interrupt type number. There are 256 interrupt types under this group.

Its execution includes the following steps –

- Flag register value is pushed on to the stack.
- CS value of the return address and IP value of the return address are pushed on to the stack.
- IP is loaded from the contents of the word location ‘type number’ $\times 4$
- CS is loaded from the contents of the next word location.
- Interrupt Flag and Trap Flag are reset to 0

The starting address for type0 interrupt is 000000H, for type1 interrupt is 00004H similarly for type2 is 00008H andso on. The first five pointers are dedicated interrupt pointers. i.e. –

- **TYPE 0** interrupt represents division by zero situation.
- **TYPE 1** interrupt represents single-step execution during the debugging of a program.
- **TYPE 2** interrupt represents non-maskable NMI interrupt.
- **TYPE 3** interrupt represents break-point interrupt.
- **TYPE 4** interrupt represents overflow interrupt.

The interrupts from Type 5 to Type 31 are reserved for other advanced microprocessors, and interrupts from 32 to Type 255 are available for hardware and software interrupts.

INT 3-Break Point Interrupt Instruction

It is a 1-byte instruction having op-code is CCH. These instructions are inserted into the program so that when the processor reaches there, then it stops the normal execution of program and follows the break-point procedure.

Its execution includes the following steps –

- Flag register value is pushed on to the stack.
- CS value of the return address and IP value of the return address are pushed on to the stack.
- IP is loaded from the contents of the word location $3 \times 4 = 0000\text{CH}$
- CS is loaded from the contents of the next word location.
- Interrupt Flag and Trap Flag are reset to 0

INTO - Interrupt on overflow instruction

It is a 1-byte instruction and their mnemonic **INTO**. The op-code for this instruction is CEH. As the name suggests it is a conditional interrupt instruction, i.e. it is active only when the overflow flag is set to 1 and branches to the interrupt handler whose interrupt type number is 4. If the overflow flag is reset then, the execution continues to the next instruction.

Its execution includes the following steps –

- Flag register values are pushed on to the stack.
- CS value of the return address and IP value of the return address are pushed on to the stack.
- IP is loaded from the contents of word location $4 \times 4 = 00010H$
- CS is loaded from the contents of the next word location.
- Interrupt flag and Trap flag are reset to 0

HARDWARE INTERRUPT VERSUS SOFTWARE INTERRUPT

HARDWARE INTERRUPT	SOFTWARE INTERRUPT
An interrupt that is generated from an external device	A type of interrupt that is caused by an instruction in the program
Generated by external devices	Generated by executing instructions
Asynchronized events	Synchronized events
Do not increment the program counter	Increase the program counter
Do not get a higher priority	Get a higher priority

Visit www.PEDIAA.com

Difference between Hardware and Software Interrupts

S.No.	Software Interrupt	Hardware Interrupt
1	Asynchronous event	Synchronous event
2	Requested by executing instruction	Requested by external device on pin
3	Cannot be ignored/masked	Can be masked except for Trap
4	Highest priority	Priority is lower than s/w interrupt
5	No effect on interrupt control logic	Affects interrupt control logic
6	Not used for interfacing peripherals	Used for interfacing peripherals
7	Does not execute any interrupt ACK cycle/idle machine cycle	Executes either interrupt ACK cycle/idle machine cycle

10.All features

Features of 80186 Microprocessor:

In 1982, the 80186 microprocessor was developed by Intel. This is an improved 8086 with several common functions built in blocks such as clock generator, system controller, interrupt controller, DMA controller, and timer/counter. This processor has 8 new instructions and executes instructions faster than the 8086. Just like the 8086 processor, it has a 16-bit external data bus. It is also available with an 8-bit external data bus, and then the processor name is 80188 microprocessor. The initial clock frequency of the 80186 and 80188 is about 6 MHz. Generally, these processors are used as embedded processors and also used as the CPU of personal computers.

The second generation of the 80186 family, such as the 80C186/C188 processors, have been developed by Intel in 1987. The 80186 was redesigned as a static, stand-alone module known as the 80C186 Modular Core and its pin configuration is compatible with the 80186 family. The high-performance CHMOS III process allowed the 80C186 to operate at twice the clock rate of the NMOS 80186, but this processor consumes less than one-fourth the power.

The 80C186 Modular Core family was further developed and the 80C186XL processor was developed in 1991. The 80C186XL/C188XL is a higher performance and lower power replacement for the 80C186/C188.

The 80186 and 80188 processor series are generally intended for embedded systems such as modems, public and private PBX switching systems, cellular phones, etc. The architecture of 80186 can also be found in many real-time environments such as robotics, automation industry, measurement control systems, sensors and test equipment's, fax machines, copiers, printers and

medical equipment. Therefore, Intel 80186 processor family, 80186, 80C186XL, 80C186EA/EB/EC, have been accepted in a wide range of applications.

Features of 80286 Microprocessor:

The Intel 80286 was introduced in early 1982. This is also known as iAPX 286 and it is an x86 I6-bit microprocessor with 134,000 transistors. It was the first Intel processor that could run all the software written for its predecessor. It was widely used in IBM PC compatible computers such as IBM PC/AT during the mid 1984 to early 1990s.

Initially, 80286 was released with 6 MHz and 8 MHz, it was subsequently scaled up to 12.5 MHz. The 80286 had an average speed of about 0.21 instructions per clock. The 6 MHz model usually operated at 0.9 MIPS, the 10 MHz model at 1.5 MIPS, and the 12 MHz model at 1.8 MIPS.

The 80286's performance is more than twice that of its predecessors, i.e., Intel 8086 and Intel 8088 per clock cycle. The 80286 processors have a 24-bit address bus. Therefore, it is able to address up to 16 MB of RAM, whereas the 8086 could directly access up to 1 MB. The 80286 CPU was designed to run multitasking applications, digital communications, real-time process control systems, and multi-user systems.

This processor is the first x86 processor, which can be used to operate in protected mode. The protected mode enabled up to 16 MB of memory to be addressed by the on-chip linear memory management unit (MMU) with 1 GB logical address space. The memory management unit is able to provide some degree of protection from applications writing outside their allocated memory zones. But the 80286 could not revert to the 8086 compatible real mode without resetting the processor.

80286 is a high-performance 16-bit microprocessor with on-chip memory management and protection capabilities. Actually, this processor has been designed for a multi-user as well as a multitasking system. Usually, the 80286 processor is booted in real mode, and thereafter it works in protected mode by software command. But it is not possible to switch the 80286 from protected mode to real mode. To shift from protected mode to real mode, 80286 microprocessors must be reset. The 80286 with 8 MHz clock provides up to 6 times higher than the 5 MHz 8086. There is no on-chip clock generator circuit in 80286. Therefore, an external 82284 chip is required to generate the external clock. The 80286 has a single CLK pin for single-phase clock input. Usually, the external clock is divided by 2 internally to generate the internal clock. The 82284 provides the 80286 RESET and READY signals.

The 80286 operates in two different modes such as real mode and protected mode. The real mode is used for compatibility with existing 8086/8088 software base, and the protected mode is used for enhanced system level features such as memory management, multitasking, and protection. The 80286 is the first advanced microprocessor with memory management and protection abilities.

Features of 80386 Microprocessor:

The concepts of memory management, privilege and protection was introduced with 80286. The 16-bit word length of 80286 provides limitations on its operating speed. But for advanced applications, technology demanded high-speed machines with more powerful instruction sets incorporating all the features of 80286. Subsequently, a CPU with a 32-bit word size and higher operating frequency and high speed of operation, has been developed to overcome all the limitations of 80286. The new processor is called the 80386 processor. This is the

third-generation processor and is introduced by Intel in 1985. The Features of 80386 Microprocessor are as follows:

- The 80386 is a 32-bit microprocessor that can support 8-bit, 16-bit and 32-bit operands. It has 32-bits registers, 32-bits internal and external data bus, and 32-bit address bus.
- Due to its 32-bit address bus, the 80386 can address up to 4GB of physical memory. The physical memory of this processor is organized in terms of segments of 4 GB size at maximum.
- The 80386 CPU is able to support 16k number of segments and the total virtual memory space is $4 \text{ giga bytes} \times 16\text{k} = 64 \text{ terabytes}$.
- Another Features of 80386 Microprocessor has a 16-byte prefetch queue.
- It is manufactured by Intel using 0.8-micron CHMOS technology.
- It is available with 275k transistors in a 132-Pin PGA package.
- It operates at clock speeds of 16 MHz to 33 MHz.
- This Features of 80386 Microprocessor has memory management unit with a segmentation unit and a paging Unit.
- It operates in real, protected and virtual real mode. The protected mode of 80386 is fully compatible with 80286.
- The 80386 instruction set is upward compatible with all its predecessors.
- The 80386 can run 8086 applications under a protected mode in its virtual 8086 mode of operation.
- The 80386 processor supports Intel 80387 numeric data processor.

Features of 80486 Microprocessor:

Due to the increasing demand for more sophisticated processing capability in advanced applications, the 80387 numeric data processors became compulsory for processors. Subsequently, the designers developed a new processor after incorporating the floating-point unit inside the CPU itself. The Intel 80486 is the first processor with an in-built 80387 floating-point unit and it is developed in 1989 using CHMOS IV technology.

The features of the 80486 processor are given below:

- It has complete 32-bit architecture which can support 8-bit, 16-bit and 32-bit data types.
- 8 KB unified level 1 cache for code and data has been added to the CPU. In advanced versions of the 80486 processor, the size of level 1 cache has been increased to 16 KB.
- The 80486 is packaged in a 168-pin grid array package. The 25 MHz, 33 MHz, 50 MHz and 100 MHz (DX-4) versions of 80486 are available in the market.
- Execution time of instructions is significantly reduced. Load, store and arithmetic instructions are executed in just one cycle when data already exists in the cache.
- Intel 80486 operates at much faster bus transfers.
- This processor retains all complex instruction sets of 80386, and more pipelining has been introduced to improve performance in speed.
- Floating-point unit is integrated with 80486 processor. Hence the delay in communications between the CPU and FPU has been eliminated and all floating-point instructions are executed within very few CPU cycles.
- For fast execution of complex instructions, the 80486 has a five-stage pipeline. Two out of the five stages are used for decoding the complex instructions and the other three stages are used for execution.

- Clock-doubling and clock-tripling technology has been incorporated in faster versions of Intel 80486 CPU. These advanced i486 processors can operate in existing motherboards with 20-33 MHz bus frequency, while running internally at two or three times of bus frequency.
- Power management and System Management Mode (SMM) of 80486 became a standard feature of the processor.

The different variations of 80486 processors are manufactured, but two most common versions are 80486DX with integrated FPU and 80486SX without integrated FPU. The Intel 80486 microprocessor was developed for speeds up to 100 MHz. AMD486 produced at 120 and 133 MHz versions of the 80486, and also manufactured in small quantities the 150 MHz and possibly 166 MHz versions.

Difference

8086 Microprocessor	80186 Microprocessor
Intel 8086 was developed in 1978 and it has about 3000 different instructions for programming.	Intel 80186 was developed in 1982 and it is the improved version of 8086. in the sense that it has faster instruction execution time. includes a few more instructions as compared to 8086.
DMA channels, programmable interrupt controller, programmable timers, programmable chip select logic are not incorporated in the 8086 processor.	High-speed DMA channels, programmable interrupt controller, programmable timers, programmable chip select logic are incorporated in the 8086 processor.
8086 has an operating frequency of 5 to 10 MHz.	80186 has an operating frequency of 8 MHz to 10 MHz.
The operation codes 63H and 64H are not available in 8086.	80186 accepts the operation code 63H and 64H and these codes will present an interrupt of byte-6.
8086 has arithmetic and logic instructions and numbers of instructions are more than 3000.	80186 is compatible with all existing instructions of 8086 and it has ten new instructions.
Slow performance with respect to 80186. Power consumption is more than 80186.	80186 is high-performance processor, with two times the performance of the standard 8086. The power consumption is less than 8086.
8086 is a 40-pin IC and available in plastic leaded chip carrier (PLCC), ceramic leads chip carrier (LCC) and pin grid array (PGA) packages.	80186 is a 68-pin IC and available in plastic leaded chip carrier (PLCC), ceramic leads chip carrier (LCC) and pin grid array (PGA) packages.
PUSH and POP instructions are available in 8086.	PUSH, POP and PUSH immediate instructions are available in 80186.
IMUL instruction is available for signed multiplication and it's format is IMUL BL	IMUL instruction is available for signed multiplication and it's format is IMUL BX, CX, 22.
8086 has no high-level instructions.	80186 has three high-level instructions such as BOUND, LEAVE and ENTER.

For 8086 to execute a rotate or shift instruction, it will need more than 1000 cycles to complete, and, therefore, there is a long delay if an interrupt of highest priority is requested.	In order to execute a rotate or shift instruction, the number of bits to shift is the count specified in the instruction modulo 32. This limits the number of shifts to 31. It requires less execution time compared to 8086.
In 8086, the <u>LOCK signal</u> can be initiated by a lock instruction prefix and is maintained until the end of the next instruction.	The LOCK signal in 80186 will not be activated until the locked instruction starts its operand reference bus cycles.

80186 microprocessor	80286 microprocessor
Intel 80186 was developed in 1982 and it is the improved version of 8086, in the sense that it has faster instruction execution time, includes a few more instructions as compared to 8086.	Intel 80286 was developed in 1983 by Intel and it is the improved version of 80186, in the sense that it has faster instruction execution time, includes a few more instructions as compared to 80186.
80186 has an operating frequency of 8 MHz to 10 MHz	80286 has an operating frequency of 8 MHz to 12.5 MHz.
80186 has no memory management capability.	80286 has memory management capability that maps 230(1GB) of virtual address.
The 80186 has 20-bit address lines and is able to access $2^{20} = 1\text{MB}$ memory.	The 80286 has 24-bit address lines and can able to access $2^{24} = 16\text{ MB}$ memory.
80186 operates in real addressing mode	80286 operates in real mode as well as <u>protected virtual address mode</u> .
80186 is compatible with all existing instructions of 8086 and it has ten new instructions.	80286 is compatible with all existing instructions of 80186 and it has some new instructions
80186 is two times the performance of the standard 8086. The power consumption is less than 8086.	80286 is six times the performance of the standard 8086. The power consumption is less than 8086.
CAP is not available in the 80186 microprocessor.	80286 has a new pin CAP. An external capacitor is connected to the CAP <u>pin</u> for filtering the bias voltage.

The difference between 80386 and 80486 Microprocessor is given in table below

80386 microprocessor	80486 microprocessor
Intel 80386 was developed in 1985 using CHMOS III technology and it is improved version of 80286.	Intel 80486 was developed by Intel in 1989 using CHMOS IV technology and it is improved version of 80386.
It is available with 275K transistors in a 132-pin PGA package.	The 80486 is available with 1200K transistors packaged in a 168-pin grid array package.
80386 operated at clock speed of 16 MHz to 33 MHz.	The 25 MHz, 33 MHz, 50 MHz and 100 MHz (DX-4) versions of 80486 are available in the market.
On-Chip cache is not available in 80386.	8 KB unified level-1 cache for code and data is available to the CPU. In advanced versions of the

	80486 processor, the size of level 1 cache has been increased to 16 KB.
4.0 MIPS at 25 MHz. Most of the instructions require 2 CLK for execution.	15.0 MIPS at 25 MHz. Most of the instructions require 1 CLK for execution.
The 80386 processor supports Intel 80387 numeric data processor. There is no on-chip numeric data processor.	Numeric data processor (floating point unit) is integrated with the 80486 processor. Therefore the delay in communications between the CPU and FPU has been eliminated and all floating point instructions are executed with in very few CPU <u>cycles</u> .
The 80386 processor has a 16-byte prefetch queue.	80486 processor has a 32-byte prefetch queue.
The 80386 processor has no multiprocessing support capability.	The 80386 processor has multiprocessing support capability.
The 80386 has less power management capability compared to 80486.	The 80486 has 2-3 times more power management capability compared to 80386
There is no RISC feature in 80386.	RISC feature is incorporated in 80486.
The alignment check (AC) flag does not exist in 80386.	The alignment Check (AC) flag exists in 80486.

8086 Microprocessor

Following are the features of 8086 microprocessor:

- Data Bus Width: 16
- Addressed Memory Size of: 1M

80286 Microprocessor

Following are the features of 80286 microprocessor:

- Data bus width: 16
- Addressed Memory size of : 16M
- Clock speed is higher and hence some instructions are executed in as little as 250ns.
- As shown in the figure, 80286 does not have multiplexed address/data bus lines.

80386 Microprocessor

Following are the features of 80386 microprocessor:

- 32-bit data bus and 32-bit memory address
- It addresses up to 4G bytes of memory.
- 80386SX: address 16M bytes of memory.
- 80386SL: address 32M bytes of memory.
- 80386SLC: address 32M bytes of memory. It contain internal cache memory which allowed it to process data at higher rates

80486 Microprocessor

Following are the features of 80486 microprocessor:

- Data Bus Width: 32 bit
- Address bus : 32 bit
- Memory Size: 4G +16K cache
- The 80486 architecture has been ungraded such that half of its instructions are executed in 1 clock cycle instead of two clock cycles.
- It has 80386 like microprocessor and 80387 like numeric coprocessor.

Following table compares 80286 vs 80386 vs 80486 and mentions difference between them.

Specifications	80286	80386	80486
CPU Speed	6 to 25 MHz	12 to 40 MHz	16 to 100 MHz
Cores	1	1	1
RAM	16MB	4GB	4GB
Functional Units	4	6	9
Pipeline stages	3	3	5
Cache off chip	0	YES (Support)	YES (Support)
Cache on chip	0	0	8 KB
Transistors	134,000	275,000	>1000000

12.-feature and functional block diagram of 8051 microcontroller

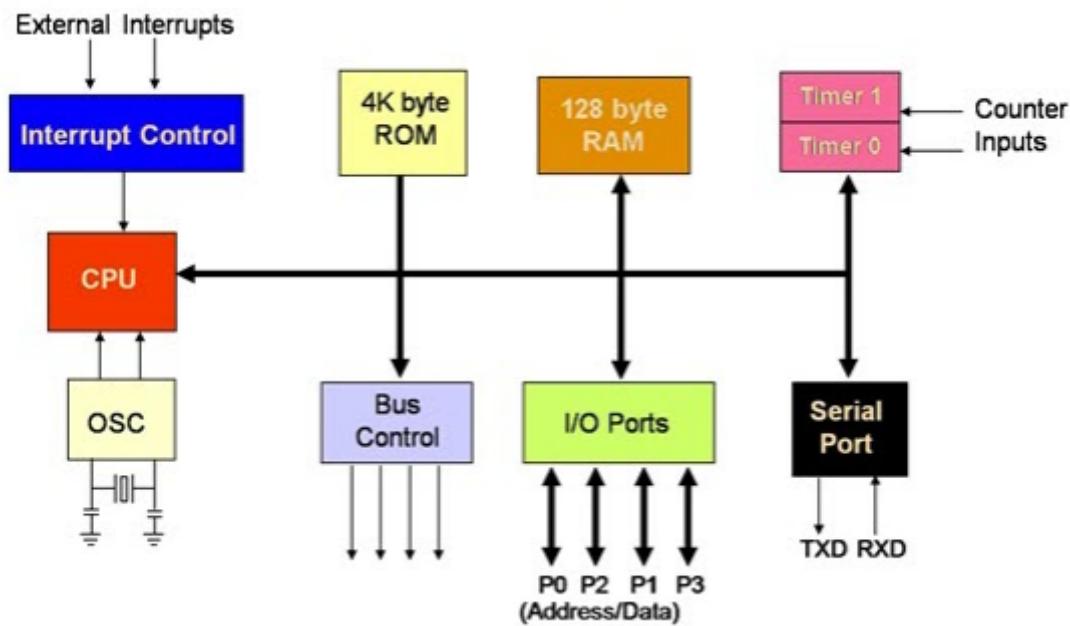
-execution of an interrupt

-memory organization:

Feature of 8051:

- The 8051 is an 8-bit microcontroller with 8 bit data bus and 16-bit address bus.
- The 16 bit address bus can address a 64K(2¹⁶) byte code memory space and a separate 64K byte of data memory space.
- The 8051 has 4K on-chip read only code memory and 128 bytes of internal Random AccessMemory(RAM)
- Besides internal RAM, the 8051 has various Special Function Registers (SFR) such as the Accumulator, the B register, and many other control registers.
- 34 8-bit general purpose registers in total.
- The ALU performs one 8-bit operation at a time.
- Two 16 bit /Counter timers
- 3 internal interrupts (one serial), 2 external interrupts.
- Four 8-bit I/O ports
- Some 8051 chips come with UART for serial communication and ADC for analog to digital Conversion

Block diagram of 8051:



Basic components present internally inside 8051 Microcontroller architecture are:

CPU (Central Processing Unit): CPU acts as a mind of any processing machine. It synchronizes and manages all processes that are carried out in a microcontroller. User has no power to control the functioning of the CPU. It interprets the program stored in ROM and carries out from storage and then performs its projected duty. CPUs manage the different types of registers available in 8051 microcontrollers.

Interrupts: Interrupts is a subroutine call that is given by the microcontroller when some other program with high priority is requested for acquiring the system buses then interrupts occur in the current running program.

Interrupts provide a method to postpone or delay the current process, perform a sub-routine task and then restart the standard program again.

Types of interrupt in 8051 Microcontroller:

Let's see the five sources of interrupts in 8051 Microcontroller:

- Timer 0 overflow interrupt - TF0
- Timer 1 overflow interrupt - TF1
- External hardware interrupt - INT0
- External hardware interrupt - INT1
- Serial communication interrupt - RI/TI

Memory: For operation Microcontroller required a program. This program guides the microcontroller to perform the specific tasks. This program installed in the microcontroller required some on chip memory for the storage of the program.

Microcontroller also required memory for storage of data and operands for a short duration. In microcontroller 8051 there is code or program memory of 4 KB that is it has 4 KB ROM and it also comprises data memory (RAM) of 128 bytes.

Bus : Bus is a group of wires which is used as a communication canal or acts as a means of data transfer. The different bus configuration includes 8, 16 or more cables. Therefore, a bus can bear 8 bits, 16 bits all together.

Types of buses in 8051 Microcontroller:

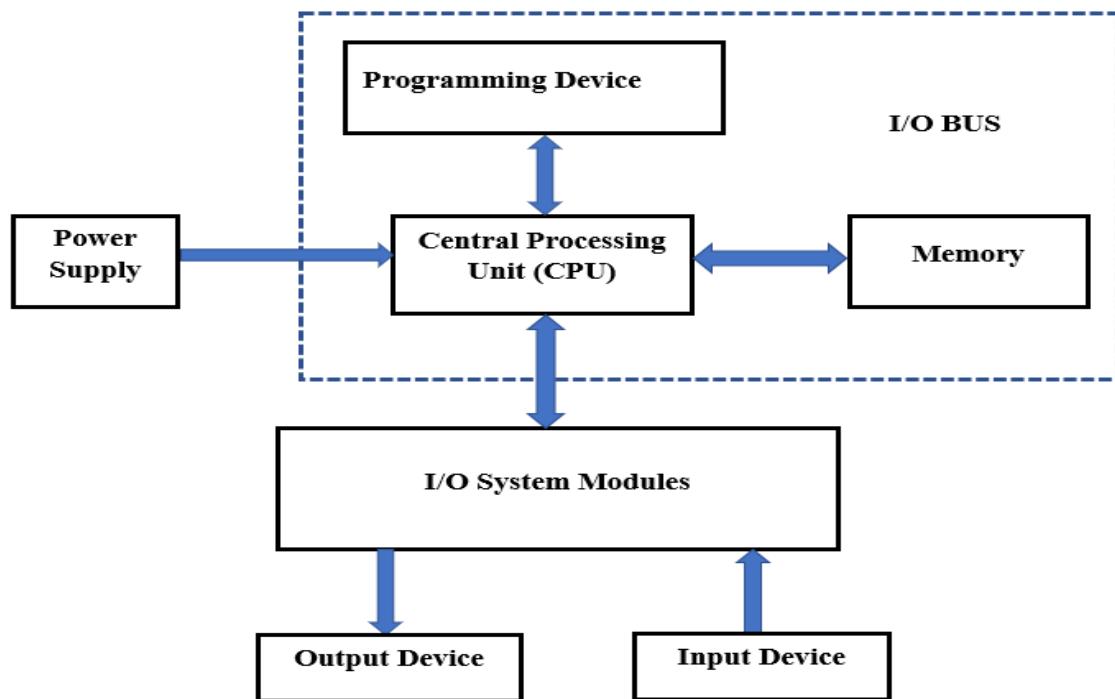
Let's see the two types of bus used in 8051 microcontroller:

- **Address Bus:** 8051 microcontrollers consist of a 16 bit address bus. It is generally be used for transferring the data from Central Processing Unit to Memory
- **Data bus:** 8051 microcontroller consists of 8 bits data bus. It is generally used for transferring the data from one peripheral position to other peripherals.

Oscillator: As the microcontroller is a digital circuit therefore it needs a timer for their operation. To perform timer operation inside a microcontroller it required an externally connected or on-chip oscillator. Microcontroller is used inside an embedded system for managing the function of devices. Therefore, 8051 uses the two 16 bit counters and timers. For the operation of these timers and counters the oscillator is used inside the microcontroller.

**13.PLC: features, plc programming languages,
-define ladder diagram (ld) with of ladder symbol/ element,
-logical function: and, or**

A PLC (programmable logic controller) is a digital computer used for industrial automation to automate different electro-mechanical processes. It was introduced to eliminate issues such as high power consumption that arose from the use of relays to control manufacturing processes. It consists of a programmed microprocessor whose program is written on a computer and later downloaded via a cable to the PLC. The program is stored in a non-volatile PLC memory.



//How does a PLC work?

The programmable logic controller receives information from connected input devices and sensors, processes the received data, and triggers required outputs as per its pre-programmed parameters. Based on its inputs and outputs, a PLC can easily monitor and record runtime data like operating temperature, machine productivity, generation of alarms when a machine fails, automatic start and stop processes and more. This means that PLCs are robust and flexible manufacturing process control solutions that are adaptable to most applications.

//PLC hardware

// PLC hardware components include:

- CPU: checks the PLC regularly to prevent errors and performs functions like arithmetic operations and logic operations.
- Memory: system ROM permanently stores fixed data used by the CPU while RAM stores the input and output device information, timer values, counters, and other internal devices.
- O/P section: this section gives output control over devices like pumps, solenoids, lights, and motors.

- **I/O section:** an input section that tracks on field devices like switches and sensors.
- **Power supply:** though most PLCs work at 24 VDC or 220VAC, some have isolated power supplies.
- **Programming device:** is used to feed the program into the processor's memory.

PLC key features

Key features of a programmable logic controller include:

- **I/O:** The CPU retains and processes data while the input and output modules connect the PLC to the machinery. I/O modules provide the CPU with information and trigger specified results. I/O modules can be analog or digital. Note that I/O can be mix-matched to achieve the right configuration for an application.
- **Communications:** Apart from input and output devices, PLCs must connect with other system types. For instance, a user may need to export application data recorded by the PLC to a SCADA (supervisory control and data acquisition) system designed to monitor several connected devices. A PLC provides different communication protocols and ports to facilitate communication between the PLC and the other systems.
- **HMI:** Users require a HMI (human machine interface) to interact with a PLC. The operator interfaces can be large

touchscreen panels or simple displays that allow users to input and review PLC information in real-time.

PLCs will continue to grow in prominence due to the current Industry 4.0 and the industrial internet of things hype. These movements require programmable logic controllers to communicate via web browsers, connect to the cloud via MQTT and to databases via SQL. As a result, PLCs will become an increasingly important part of modern machine auto

PLC Programming Languages

EN 61131-3 defines five PLC programming languages:

1 Ladder Diagram (LD): graphic language derived from circuit diagram of directly wired relay controls.

2 Function Block Diagram (FBD): functions & function blocks are represented graphically and interconnected into networks.

3 Instruction List (IL): textual assembler-type language consisting of an operator and an operand.

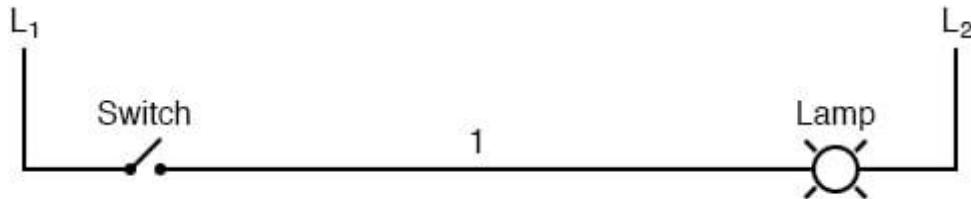
4 Structured Text (ST): high level language based on Pascal.

5 Sequential Function Chart (SFC): a language resource for the structuring of sequence-oriented control programs.

-define ladder diagram (ld) with ladder symbol/ element

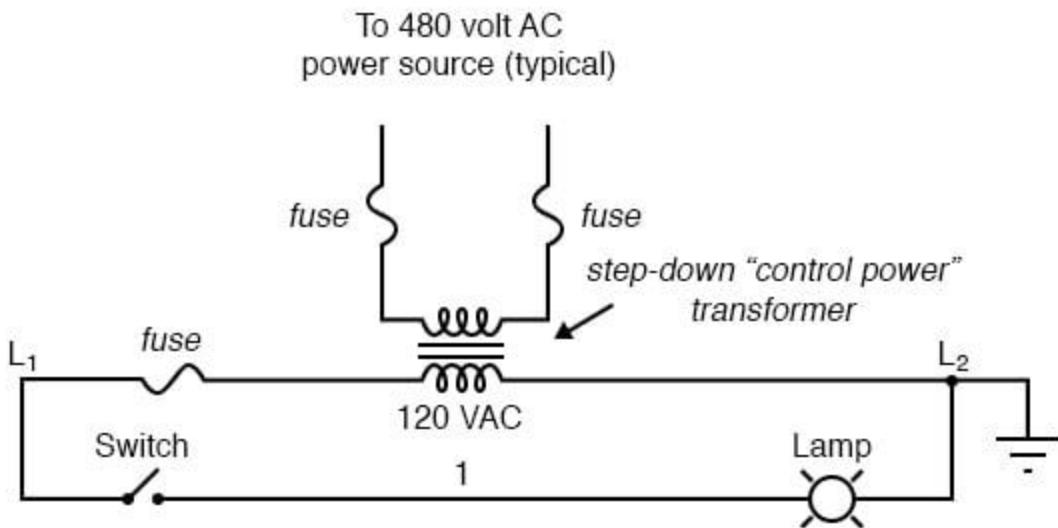
Ladder: Ladder diagrams are specialized schematics commonly used to document industrial control logic systems. They are called “ladder” diagrams because they resemble a ladder, with two vertical rails (supply power) and as many “rungs” (horizontal lines) as there are control circuits to represent.

If we wanted to draw a simple ladder diagram showing a lamp that is controlled by a hand switch, it would look like this:



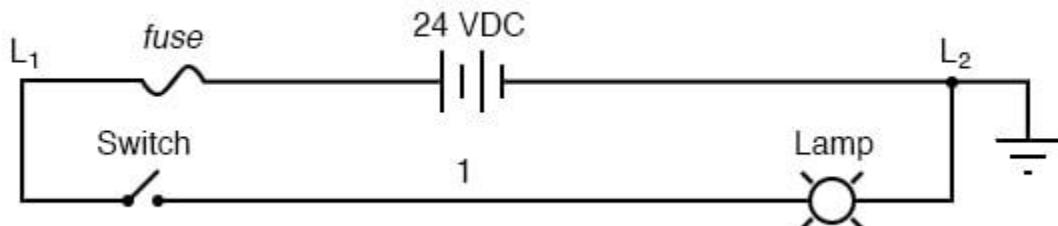
The “L₁” and “L₂” designations refer to the two poles of a 120 VAC supply unless otherwise noted. L₁ is the “hot” conductor, and L₂ is the grounded (“neutral”) conductor.

These designations have nothing to do with **inductors**, just to make things confusing. The actual transformer or generator supplying power to this circuit is omitted for simplicity. In reality, the circuit looks something like this:



Typically in industrial relay logic circuits, but not always, the operating voltage for the switch contacts and relay coils will be 120 volts AC.

Lower voltage AC and even DC systems are sometimes built and documented according to "ladder" diagrams:



So long as the switch contacts and **relay coils are all adequately rated, it really doesn't matter what level of **voltage** is chosen for the system to operate with.**

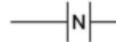
BUET PDF:
Ladder Diagram (LD)

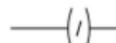
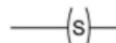
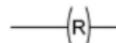
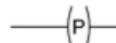
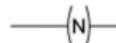
The use of ladder programming involves writing a program in a manner to draw a switching circuit. The ladder diagram consists of two vertical lines representing the power rails, and circuits are connected as horizontal lines.

Advantages of Ladder Language -

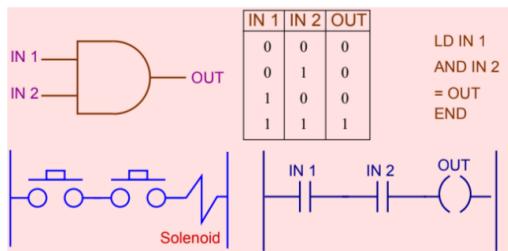
- It is readily understood and maintained.
- It provides a graphic display of program flow.
- Programming is fast.
- Generates more readable programs for sequence control.
-

Elements of Ladder Diagram

Contacts	
	Normally open contact
	Normally closed contact
	Edge contact, positive edge
	Edge contact, negative edge

Coils	
	Coil
	Negating coil
	Setting coil
	Resetting coil
	Edge coil, positive edge
	Edge coil, negative edge

Logical Function: AND



x020.eps

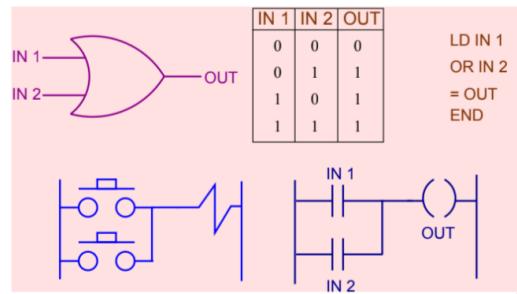


Prof. Dr. M. Zahurul Haq (BUET)

PLC

ME 475 31 / 44

Logical Function: OR



x021.eps



Prof. Dr. M. Zahurul Haq (BUET)

PLC

ME 475 32 / 44