UNIT 4

I/O System

Peripherals:

The input and output devices that are attached to the computer are called peripheral devices.

Eg: Keyboard, printer, display unit.

- Peripherals that provide auxiliary storage to the system are magnetic disks and tapes.
- Printers provide a permanent record on paper of computer output data or text.
  The 3 types of printers:
  1) Daisy wheel
  2) Dot matrix
  3) Laser
- Magnetic tapes are used for storing files of data.
  Access – Sequential
  Slowest and cheapest method.
  Tapes can be removed when not in use.

- Magnetic disks are used for bulk storage of programs and data.
  Access – by moving the read/write mechanism to a track in the magnetized surface.

  ASCII –American Standard Code for Information Interchange
  It is a 7 bit code. Most computers manipulate an 8 bit quantity as a single unit called a byte. ASCII characters are often stored one per byte.

Input Output Interface

It provides a method for transferring information between internal storage and external I/O devices. Peripherals need special communication links for interfacing with CPU. The purpose of the communication link is to resolve the differences between the central computer and each peripheral.

The major differences are:

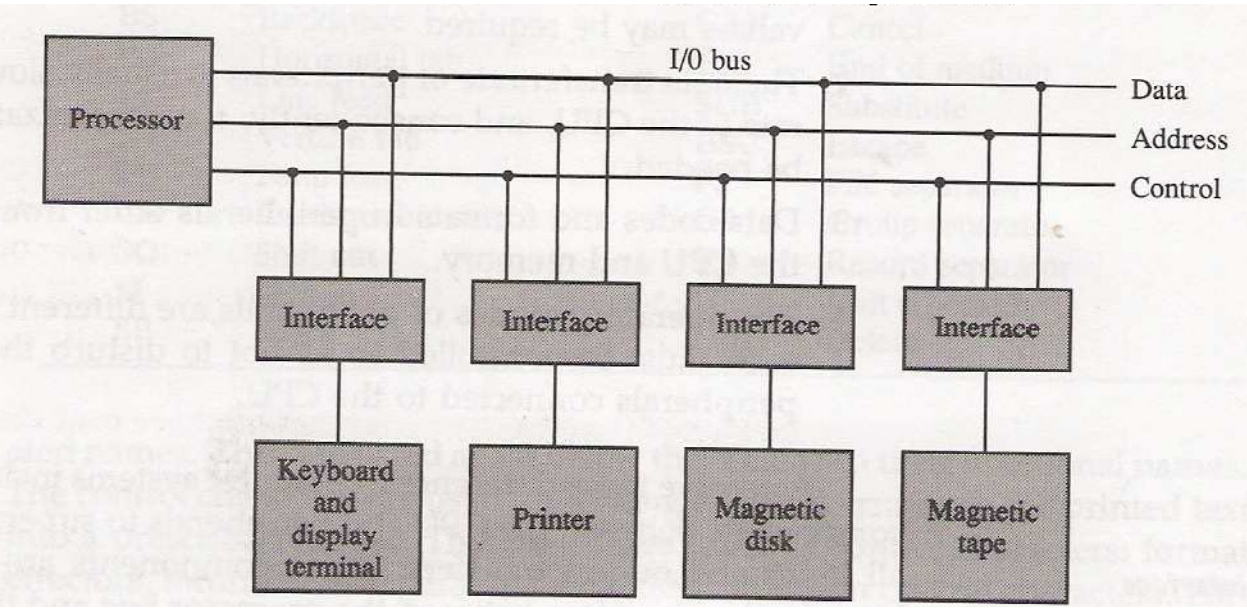
1. Peripherals are electromagnetic and electromechanical devices, whereas CPU and memory are electronic devices.
2. The data transfer rate of peripherals is slower than that of the CPU.

3. The operating modes of peripherals are different from each other.
4. Data codes and formats in peripherals differ from the word format of CPU and memory.

To resolve the differences, special hardware components are included between CPU and peripherals to supervise and synchronize all the input and output transfers called interface units.

I/O Bus and Interface Modules

The communication between the processor and several peripherals through the I/O Bus is shown in the following figure.



I/O Command:

The function code is referred to as an I/O command. It is an instruction that is executed in the interface and its attached peripheral unit.

An interface may receive four types of commands. They are control, status, data input and data output

Control command: It is issued to activate the peripheral and to inform it what to do.

Status command: It is used to test various status conditions in the interface and the peripheral.

Data input command: The interface receives an item of data from the peripheral and places it in its buffer register.

Data output command: The command causes the interface to respond by transferring data from the bus into one of its registers.

I/O versus memory bus

There are 3 ways that computer buses can be used to communicate with memory and I/O.

1. Use two separate buses, one for memory and the other for I/O.
2. Use one common bus for both memory and I/O but have separate control lines for each.
3. Use one common bus for memory and I/O with common control lines.

Isolated versus Memory mapped I/O

Isolated I/O :

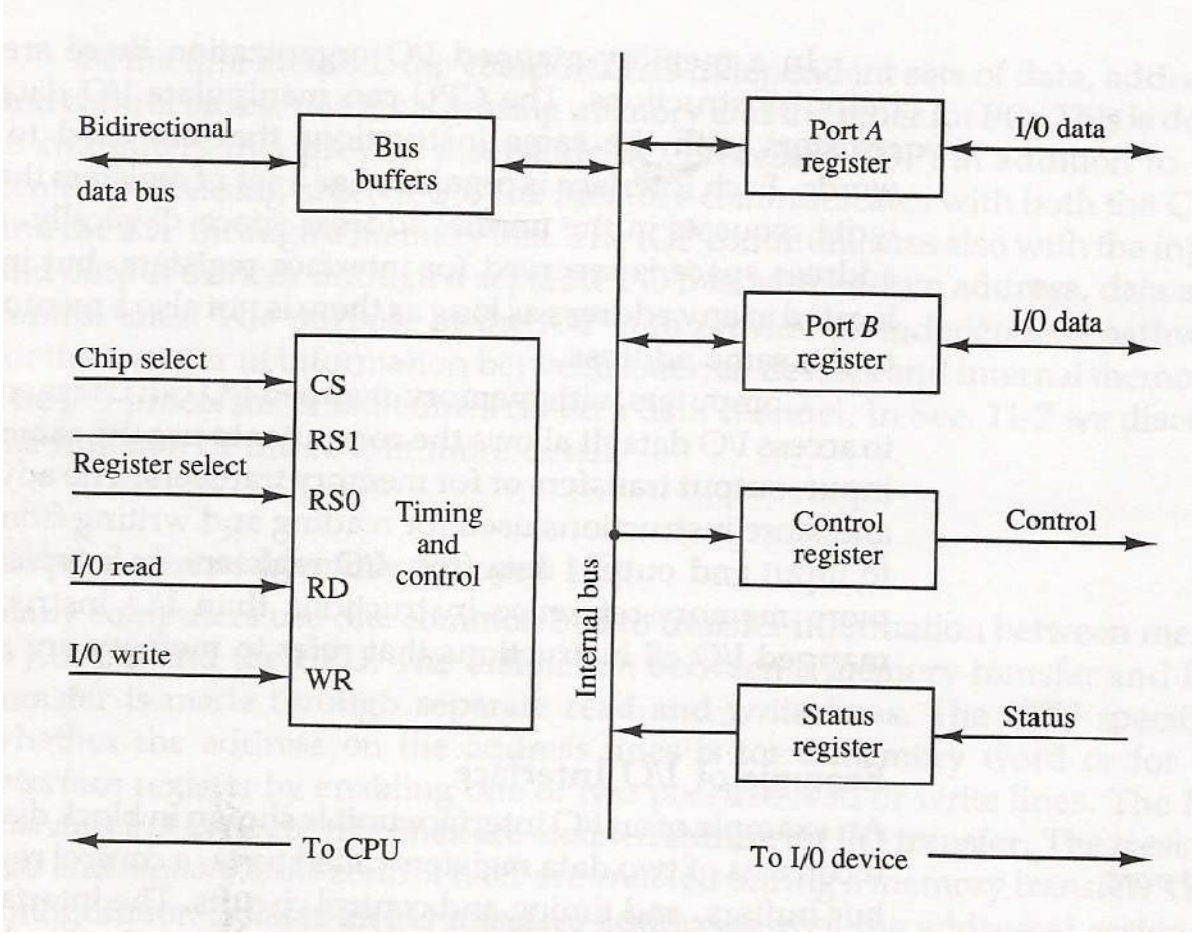
1) The CPU has distinct input and output instructions.
2) Isolates memory and I/O addresses.
3) Each has its own address space.

Memory mapped I/O

1) No specific input output instructions.
2) Use memory related instuctions for accessing data.
3) Do not distinguish between memory and I/O addresses.
4) Memory and I/O share the same set of addresses.

Example of I/O interface :

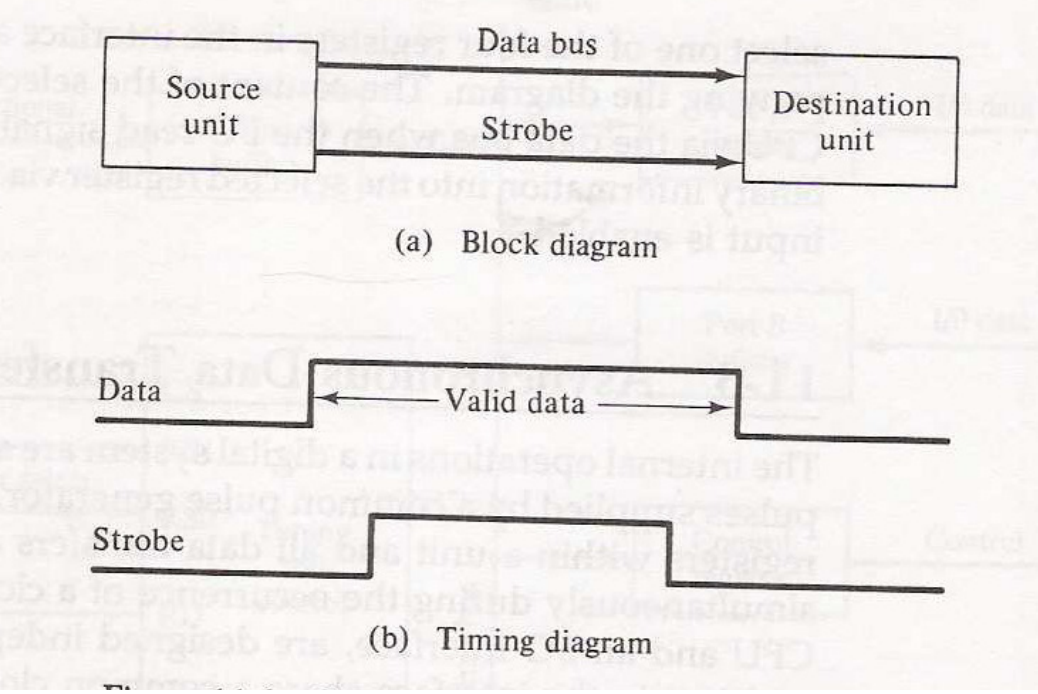
An example of an I/O interface unit is shown in the following block diagram.



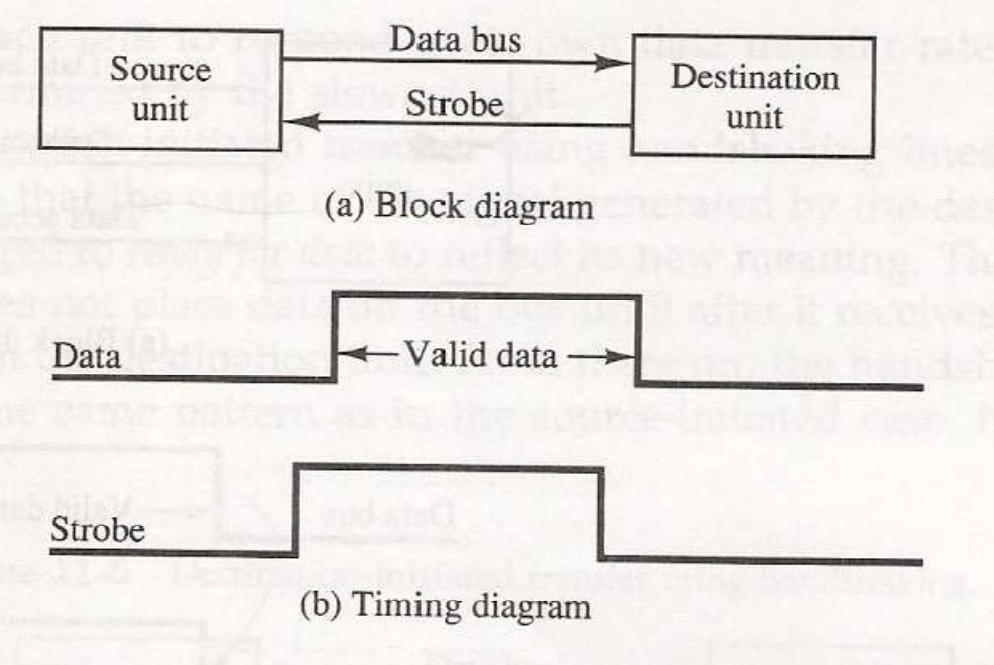
CS	RS1	RS0	Register selected
0	×	×	None: data bus in high–impedance
1	0	0	Port *A* register
1	0	1	Port *B* register
1	1	0	Control register
1	1	1	Status register

Asynchronous data transfer :

- Control signals are transmitted between the two communicating units to indicate the time at which data is being transmitted.
- A strobe pulse is supplied by one of the units to indicate to the other unit when the transfer has to occur.
- The unit receiving the data item responds with another control signal to acknowledge receipt of the data.
- This type of agreement between two independent units is referred to as handshaking.



Source initiated strobe for data transfer



Destination initiated strobe for data transfer

Disadvantage of strobe method:

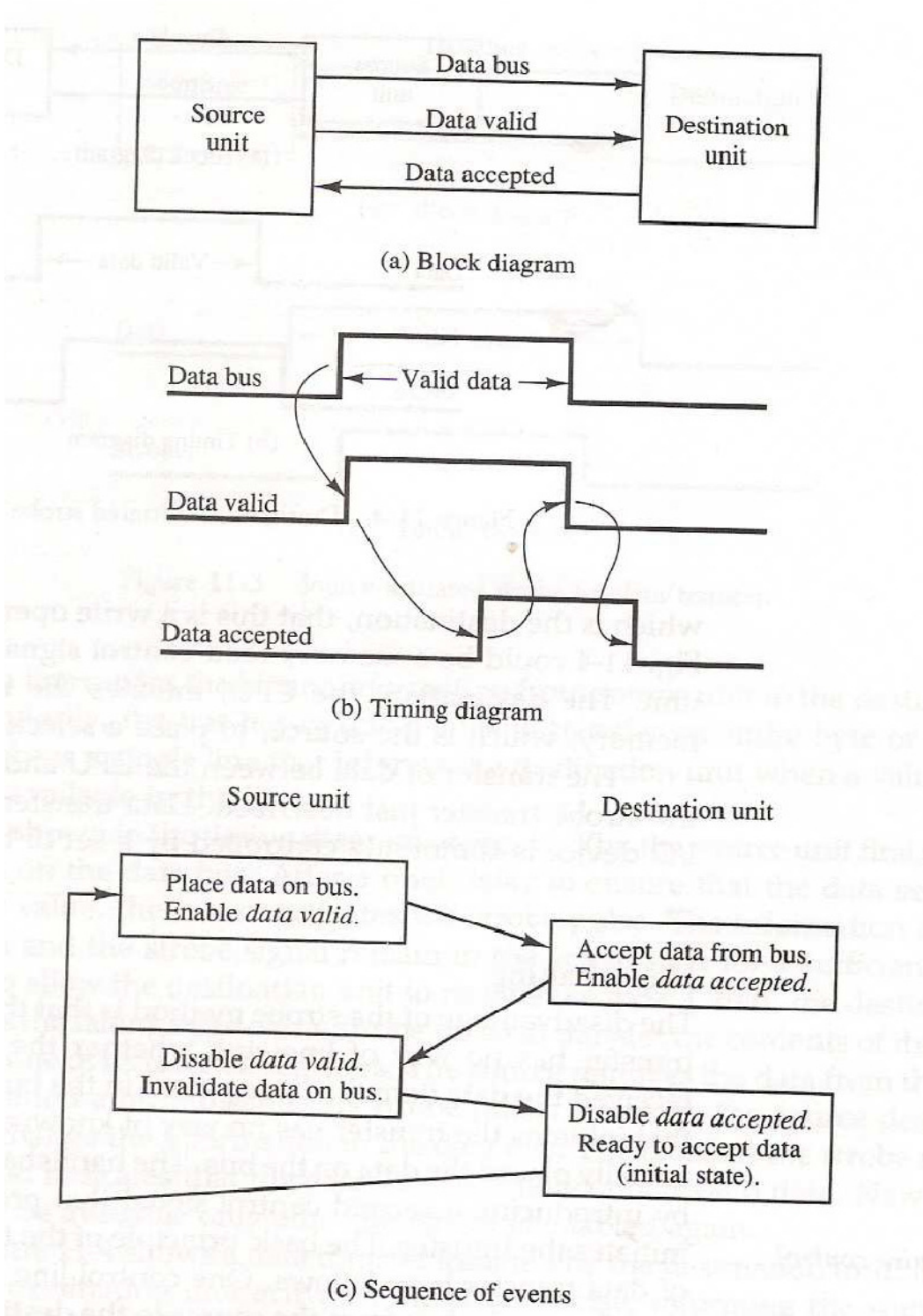
The source/ destination unit that initiates the transfer has no way of knowing whether the destination/source unit has actually received the data item that was placed in the bus.

Handshaking:

The two handshaking lines are data valid, which is generated by the source unit and data accepted, generated by the destination unit.

1) The source unit initiates the transfer by placing the data on the bus and enabling its data valid signal.
2) The data accepted signal is activated by the destination unit after it accepts the data from the bus.
3) The source unit then disables its data valid signal, which invalidates the data on the bus.
4) The destination unit then disables its data accepted signal and the system goes into initial state.
5) The source does not send the next data item until after the destination unit shows its readiness to accept new data by disabling its data accepted signal,

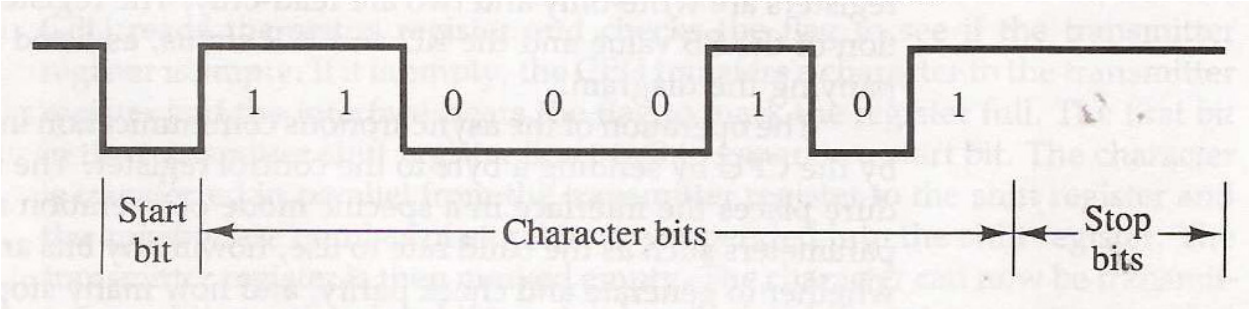
This scheme allows arbitrary delays from one state to the next and permits each unit to respond at its own data transfer rate. The rate of transfer is determined by the slowest unit.



Source initiated transfer using handshaking

Asynchronous serial transfer

- Serial transmission may be synchronous or asynchronous.
- In serial synchronous transmission, two units share a common clock frequency and the bits are transmitted at a rate dictated by the clock pulses.
- A serial asynchronous data transmission technique used in many interactive terminals employs special bits that are inserted at both ends of the character code.
- Each character consists of three fields : a start bit, the character bits and stop bits.
- The transmitter rests at the 1- state when no characters are transmitted.
- The first bit, called the start bit is always a 0 and is used to indicate the beginning of a character,
- The last bit called the stop bit is always a 1.



Asynchronous serial transmission

UART:

The integrated circuits that are specially designed to provide the interface between computer and similar interactive terminals is called an asynchronous communication interface or a universal asynchronous receiver – transmitter (UART)

Baud rate

The rate at which serial information is transmitted and is equivalent to the data transfer in bits per second.

FIFO buffer

- A first in first out (FIFO) buffer is a memory unit that stores information in such a manner that the item first in is the item first out.
- A FIFO buffer comes with separate input and output terminals.

- It can input data and output data at two different rates and the output data are always in the same order in which the data entered the buffer.
- Useful in some applications when data are transferred asynchronously.

Modes of transfer

Data transfer to and from peripherals may be handled in one of the three modes

1. Programmed I/O
2. Interrupt initiated I/O
3. Direct memory access

Programmed I/O

- Each data item is initiated by an instruction in the program.
- The transfer is to and from a CPU register and peripheral.
- Transferring data under program control requires constant monitoring of the peripheral by the CPU.
- The CPU stays in a program loop until the I/O unit indicates that it is ready for data transfer.
- Time consuming process since it keeps the processor busy needlessly.

Interrupt initiated I/O

- The interface monitors the device.
- When the device is ready for data transfer, the interface generates an interrupt request to the computer.
- Upon detecting the external interrupt signal, the CPU momentarily stops the task it is processing, branches to a service program to process the I/O transfer and then returns to the task it was originally performing.

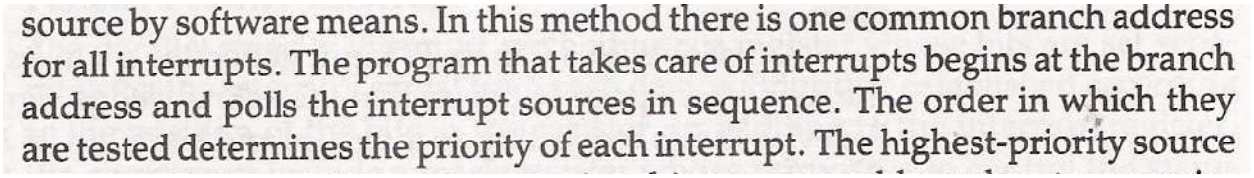
DMA

- The interface transfers data into and out of the memory unit through the memory bus.
- The CPU initiates the transfer by supplying the interface with the starting address and the number of words needed to be transferred and then proceeds to execute other tasks.
- When the transfer is made, the DMA requests memory cycles through the memory bus.

- When the request is granted by the memory controller, the DMA transfers the data directly into memory.
- The CPU delays its memory access operation to allow the direct memory I/O transfer.

Priority Interrupt

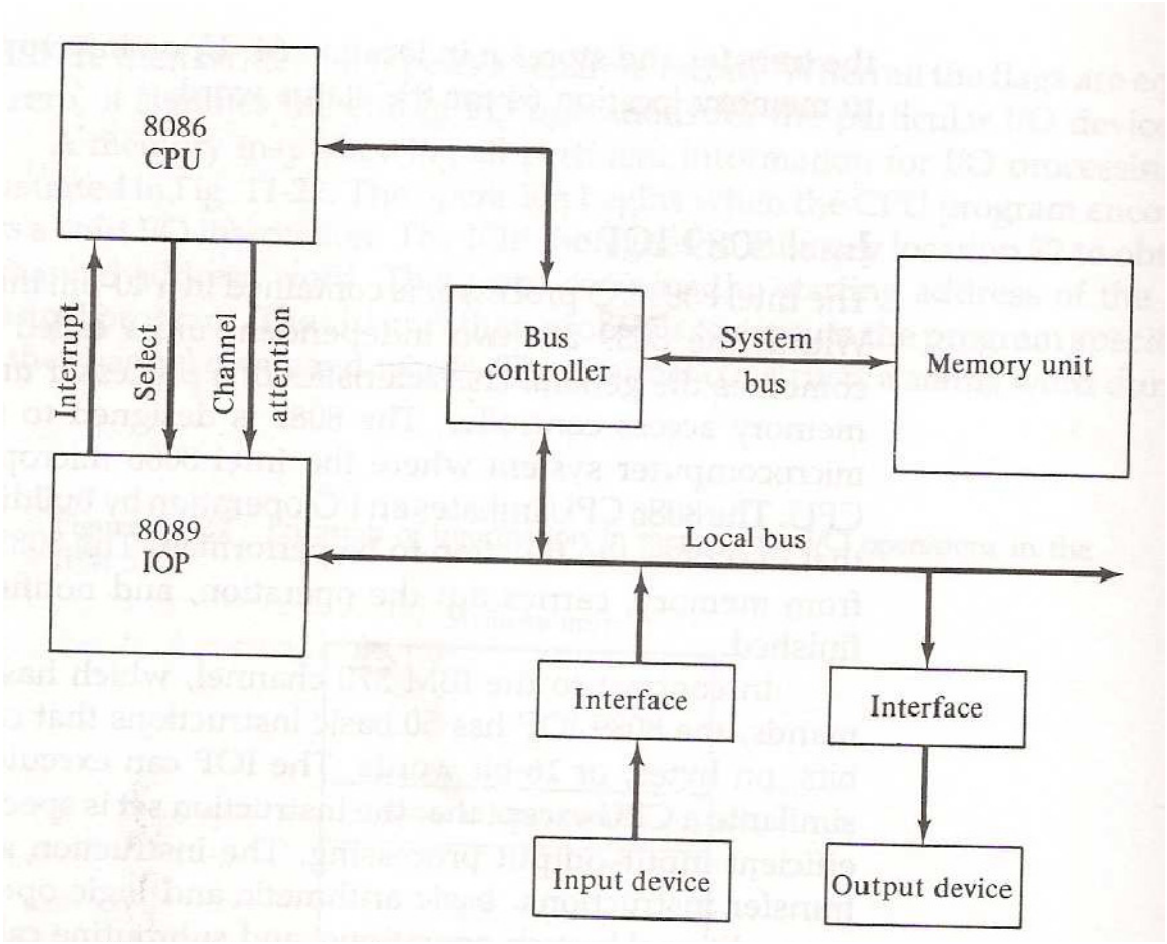
- A priority interrupt is a system that establishes a priority over the various sources to determine which condition is serviced first when two or more requests arrive simultaneously.
- Establishing the priority of simultaneous interrupts can be done by software or hardware.
- A polling procedure is used to identify the highest priority source by software means.
- In this method, there is one common branch address for all interrupts.
- The program that takes care of interrupts begins at the branch address and polls the interrupt sources in sequence.
- The order in which they are tested deter
-


source by software means. In this method there is one common branch address for all interrupts. The program that takes care of interrupts begins at the branch address and polls the interrupt sources in sequence. The order in which they are tested determines the priority of each interrupt. The highest-priority source

IOP

- The Intel 8089 I/O processor is contained in a 40 pin integrated circuit package.
- There are two independent units called channels.

- Reads the message from memory, carries out the operation and notifies the CPU when it has finished.
- Contains 50 basic instructions that can operate on individual bits, on bytes or 16 bit words.
- It can execute programs in a manner similar to a CPU except that the instruction set is specifically chosen to provide efficient input-output processing.
- It provides efficient data transfer between any two components attached to the system bus, such as I/O to memory, memory to memory or I/O to I/O.
- In the Intel 8086/8089 microcomputer system, the 8086 fuctions as the CPU and the 8089 as the IOP.
- The two units share a common memory through a bus controller connected to a system bus, called a "multibus" by Intel.
- The IOP uses a local bus to communicate with various interface units connected to I/O devices.
- The CPU communicates with the IOP by enabling the channel attention line.
- The CPU uses the select line to select one of the two channels in IOP.
- The IOP gets the attention of the CPU by sending an interrupt request.
- The CPU and IOP communicates with each other by writing messages for one another in system memory.
- The CPU prepares the message area and signals the IOP by enabling the channel attention line.
- The IOP reads the message, performs the required I/O functions and executes the appropriate channel program .
- When the channel has completed its program, it issues an interrupt request to the CPU.
- The communication scheme consists of program sections called blocks.
- Each block contains control and parameter information as well as an address pointer to its successor block.
- The address of the control block is passed to each IOP channel during initialization.
- The busy flag indicates whether the IOP is busy or ready to perform a new I/O operation.
- The CCW (Channel command word) is specified by the CPU to indicate the type of operation required from the IOP.



Intel 8086/8089 microcomputer block diagram

Direct Memory Access (DMA)

Direct memory access (**DMA**) is a feature of computer systems that allows certain hardware subsystems to access main system memory (RAM) independently of the central processing unit (CPU).

Without DMA, when the CPU is using programmed input/output, it is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work. With DMA, the CPU first initiates the transfer, then it does other operations while the transfer is in progress, and it finally receives an interrupt from the DMA controller when the operation is done. This feature is useful at any time that the CPU cannot keep up with the rate of data transfer, or when the CPU needs to perform useful work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards. DMA is also used for intra-chip data transfer in multi-core

processors. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without DMA channels. Similarly, a processing element inside a multi-core processor can transfer data to and from its local memory without occupying its processor time, allowing computation and data transfer to proceed in parallel.

DMA can also be used for "memory to memory" copying or moving of data within memory. DMA can offload expensive memory operations, such as large copies or scatter-gather operations, from the CPU to a dedicated DMA engine. An implementation example is the I/O Acceleration Technology.

Types of modes:

Burst mode

An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system bus by the CPU, it transfers all bytes of data in the data block before releasing control of the system buses back to the CPU, but renders the CPU inactive for relatively long periods of time. The mode is also called "Block Transfer Mode". It is also used to stop unnecessary data.

Cycle stealing mode

The *cycle stealing mode* is used in systems in which the CPU should not be disabled for the length of time needed for burst transfer modes. In the cycle stealing mode, the DMA controller obtains access to the system bus the same way as in burst mode, using BR (Bus Request) and BG (Bus Grant) signals, which are the two signals controlling the interface between the CPU and the DMA controller. However, in cycle stealing mode, after one byte of data transfer, the control of the system bus is deasserted to the CPU via BG. It is then continually requested again via BR, transferring one byte of data per request, until the entire block of data has been transferred. By continually obtaining and releasing the control of the system bus, the DMA controller essentially interleaves instruction and data transfers. The CPU processes an instruction, then the DMA controller transfers one data value, and so on. On the one hand, the data block is not transferred as quickly in cycle stealing mode as in burst mode, but on the other hand the CPU is not idled for as long as in burst mode. Cycle stealing mode is useful for controllers that monitor data in real time.

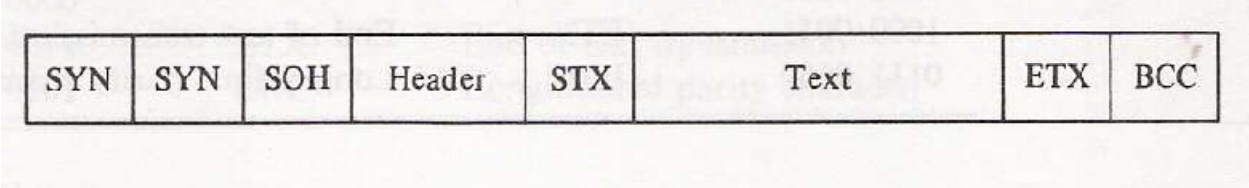
Transparent mode

The *transparent mode* takes the most time to transfer a block of data, yet it is also the most efficient mode in terms of overall system performance. The DMA controller only transfers data when the CPU is performing operations that do not use the system buses. It is the primary advantage of the transparent mode that the CPU never stops executing its programs and the DMA transfer is free in terms of time. The disadvantage of the transparent mode is that the hardware needs to determine when the CPU is not using the system buses, which can be complex.

Serial Communication

Data transmission between two points occurs in three different modes

1) Simplex – This line carries information in one direction only.
   Ex: radio and TV broadcasting
2) Half-duplex – This transmission system is capable of transmitting in both directions , but data can be transmitted in only one direction at a time.
   Ex: Modem
3) Full-duplex – This transmission system can send and receive data in both directions simultaneously.
   Ex: a two wire circuit

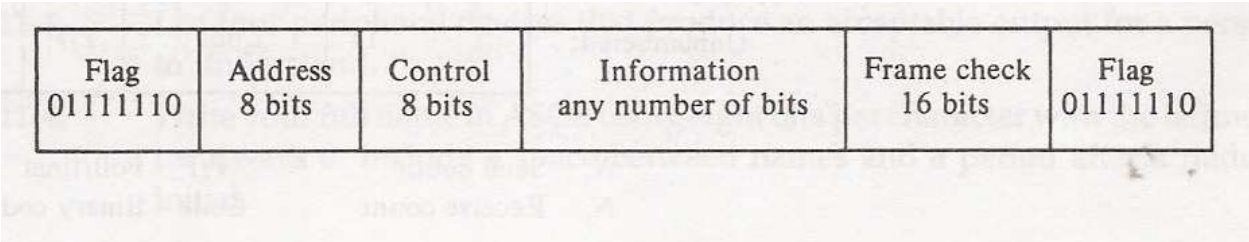


Typical message format for character oriented protocol

- There are a number of control characters used for message formation.
- Each character, including the control characters, is transmitted serially as an 8- bit binary code which consists of the 7 bit ASCII code plus an odd parity bit in the eighth most significant position.
- The two SYN characters are used to synchronize transmitter and receiver.
- The heading starts with the SOH character and continues with two characters that specify the address of the terminal.

Bit oriented protocol

- A frame starts with the 8 bit flag 01111110 followed by an address and control sequence.
- The frame check field is a CRC (cyclic redundancy check) sequence used for detecting errors in transmission.
- The ending flag indicates to the receiving station that the 16 bits just received constitute the CRC bits.
- The ending frame can be followed by another frame, another flag or a sequence of consecutive 1's.
- A frame must have a minimum of 32 bits between flags to accomodate the address, control and frame check fields.



Frame format for bit oriented protocol