Input-Output organization and multiprocessor

Peripheral Devices

- Input or output devices that are connected to computer are called peripheral devices.
- These devices are designed to read information into or out of the memory unit upon command from the CPU and are considered to be the part of computer system.
- These devices are also called peripherals.
- For example: Keyboards, display units and printers are common peripheral devices.

There are three types of peripherals:

- Input peripherals : Allows user input, from the outside world to the computer. Example: Keyboard, Mouse etc.
- Output peripherals: Allows information output, from the computer to the outside world. Example: Printer, Monitor etc

- Input-Output peripherals: Allows both input(from outised world to computer) as well as, output(from computer to the outside world). Example: Touch screen etc.

Interfaces

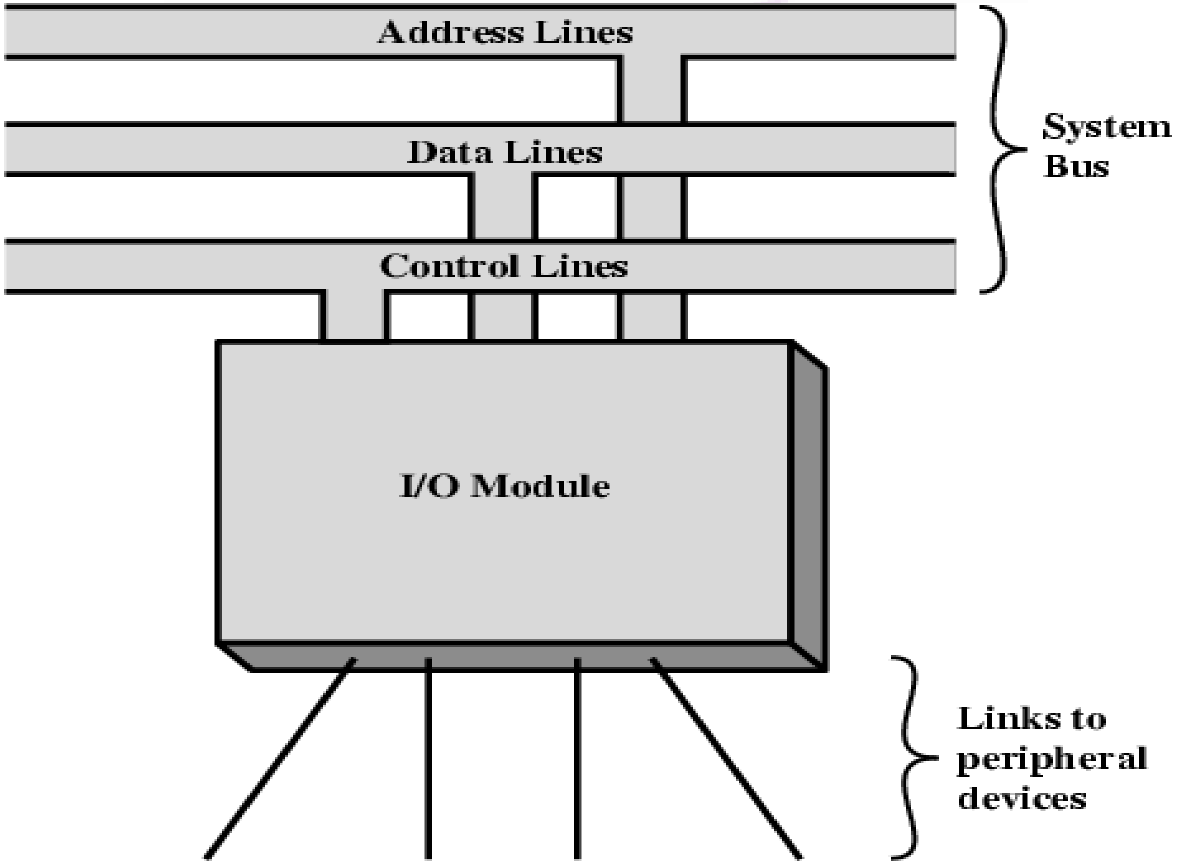
- Interface is a shared boundary between two separate components of the computer system which can be used to attach two or more components to the system for communication purposes.
- There are two types of interface:
  - CPU Interface
  - I/O Interface

Input-Output Interface

- Peripherals connected to a computer need special communication links for interfacing with CPU.
- In computer system, there are special hardware components between the CPU and peripherals to control or manage the input-output transfers.
- These components are called **input-output interface units** because they provide communication links between processor bus and peripherals.
- They provide a method for transferring information between internal system and input-output devices.

I/O Modules

- I/O modules interface creates a link that provides a means of exchanging data between external environment and computer
- The link is used to exchange control status and data between I/O module and the external devices.



- Peripherals are not directly connected to the system bus
- Instead an I/O module is used which contains logic for performing a communication between the peripherals and the system bus.
- The reasons due to which peripherals do not directly connected to the system bus are:
  - There are a wide variety of peripherals with various methods of operation.
  - It would be impractical to incorporate the necessary logic within the processor to control a range of devices.
  - The data transfer rate of peripherals is often much slower than that of the memory or processor.
  - Thus, it is impractical to use high speed system bus to communicate directly with a peripheral and vice versa.
  - Peripherals often use different data format and word length than the computer to which they are connected.

- Thus an I/O module is required which performs two major functions.
  - Interface to the processor and memory via the system bus
  - Interface to one or more peripherals by tailored data links

Functions

- Used to supervise and synchronize all I/O transformation
- **Control & Timing**
  - I/O module includes control and timing to coordinate the flow of traffic between internal resources and external devices.
  - The control of the transfer of data from external devices to processor consists following steps:
    - The processor interrogates the I/O module to check status of the attached device.
    - The I/O module returns the device ststus.

    - If the device is operational and ready to transmit, the processor requests the transfer of data by means of a command to I/O module.
    - The I/O module obtains the unit of data from the external device.
    - The data are transferred from the I/O module to the processor.

- **Processor Communication**
  - I/O module communicates with the processor which involves:
    - Command decoding: I/O module accepts commands from the processor.
    - Data: Data are exchanged between the processor and I/O module over the bus.
    - Status reporting: Peripherals are too slow and it is important to know the status of I/O module.
    - Address recognition: I/O module must recognize one unique address for each peripheral it controls.

- **Device Communication**
  - It involves commands, status information and data.
- **Data Buffering**
  - I/O module must be able to operate at both device and memory speeds.
  - If the I/O device operates at a rate higher than the memory access rate, then the I/O module performs data buffering.
  - If I/O devices rate slower than memory, it buffers data so as not to tie up the memory in slower transfer operation.
- **Error Detection**
  - I/O module is responsible for error detection such as mechanical and electrical malfunction reported by device
  - e.g. paper jam, bad ink track & unintentional changes to the bit pattern and transmission error.

Input-Output interface

- Input-Output interface provides a method for transferring information between internal storage (such as memory and CPU registers) and external I/O devices.
- Peripherals connected to a computer need special communication links for interfacing them with the central processing unit.
- The communication link resolves the following differences between the computer and peripheral devices.
  - – Devices and signals
    - Peripherals - Electromechanical Devices
    - CPU or Memory - Electronic Device
  - – Data Transfer Rate
    - Peripherals - Usually slower
    - CPU or Memory - Usually faster than peripherals Some kinds of Synchronization mechanism may be needed

  - Unit of Information
    - Peripherals - Byte
    - CPU or Memory - Word
  - Operating Modes
    - Peripherals - Autonomous, Asynchronous
    - CPU or Memory – Synchronous
- Interface performs the following
  - Decodes the device address (device code)
  - Decodes the commands (operation)
  - Provides signals for the peripheral controller
  - Synchronizes the data flow and supervises the transfer rate between peripheral and CPU or Memory

- I/O commands that the interface may receive:
  - Control command: issued to activate the peripheral and to inform it what to do.
  - Status command: used to test various status conditions in the interface and the peripheral.
  - Output data: causes the interface to respond by transferring data from the bus into one of its registers
  - Input data: is the opposite of the data output.
- Isolated I/O
  - Separate I/O read/write control lines in addition to memory read/write control lines
  - Separate (isolated) memory and I/O address spaces
  - Distinct input and output instructions

- Memory-mapped I/O
  - A single set of read/write control lines (no distinction between memory and I/O transfer)
  - Memory and I/O addresses share the common address space which reduces memory address range available
  - No specific input or output instruction so the same memory reference instructions can be used for I/O transfers
  - Considerable flexibility in handling I/O operations

Modes of I/O Data Transfer

- Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:
  - Programmed I/O
  - Interrupt Initiated I/O
  - Direct Memory Access

Programmed I/O

- Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program.
- Usually the program controls data transfer to and from CPU and peripheral. Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.
- CPU is in a continuous monitoring of the interface in which it checks the F bit of the status register.
  - If it is set i.e. 1, then the CPU reads the data from data register and sets F bit to zero
  - If it is reset i.e. 0, then the CPU remains monitoring the interface.

Interrupt Initiated I/O

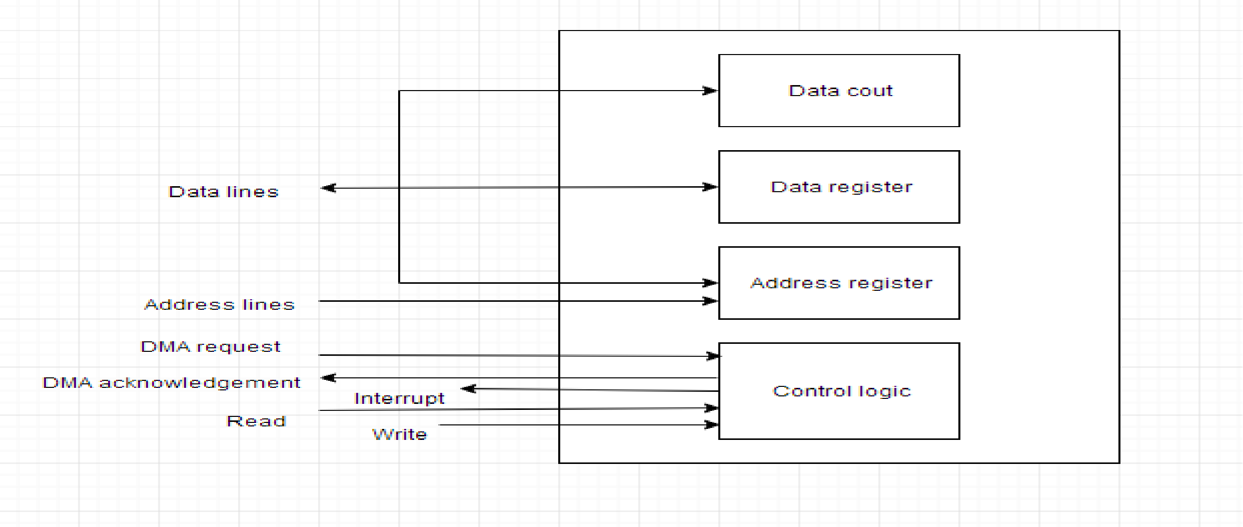
- In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer.

- This is time consuming process because it keeps the processor busy needlessly.
- This problem can be overcome by using **interrupt initiated I/O**.
- In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt.
- After receiving the interrupt signal, the CPU stops the task which it is processing
- And service the I/O transfer and then returns back to its previous processing task.

Direct Memory Access

- Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer.
- This technique is known as **DMA**.

- In this, the interface transfer data to and from the memory through memory bus.
- A DMA controller manages to transfer data between peripherals and memory unit.
- Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc.
- It is also used for intra chip data transfer in multicore processors.
- In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.



Characteristics of Multiprocessor

- **Parallel Computing**
  - This involves the simultaneous application of multiple processors.
  - These processors are developed using a single architecture to execute a common task.
- **Distributed Computing**
  - This involves the usage of a network of processors.
  - Each processor in this network can be considered as a computer in its own right and have the capability to solve a problem.
  - These processors are heterogeneous, and generally, one task is allocated to a single processor.

- **Supercomputing**
  - This involves the usage of the fastest machines to resolve big and computationally complex problems.
  - In the past, supercomputing machines were vector computers but at present, vector or parallel computing is accepted by most people.
- **Pipelining**
  - This is a method wherein a specific task is divided into several subtasks that must be performed in a sequence.
  - The functional units help in performing each subtask.
  - The units are attached serially and all the units work simultaneously.
- **Vector Computing**
  - It involves the usage of vector processors, wherein operations such as ‘multiplication’ are divided into many steps and are then applied to a stream of operands (“vectors”).

- **Systolic**
  - This is similar to pipelining, but units are not arranged in a linear order.
  - The steps in systolic are normally small and more in number and performed in a lockstep manner.
  - This is more frequently applied in special-purpose hardware such as image or signal processors.

Interconnection Structures

- The components that form a multiprocessor system are CPUs, IOPs connected to input- output devices, and a memory unit.
- The interconnection between the components can have different physical configurations, depending on the number of transfer paths that are available
  - Between the processors and memory in a shared memory system
  - Among the processing elements in a loosely coupled system
- There are several physical forms available for establishing an interconnection network.
  - Time-shared common bus
  - Multiport memory
  - Crossbar switch
  - Multistage switching network
  - Hypercube system

Time Shared Common Bus

- A common-bus multiprocessor system consists of a number of processors connected through a common path to a memory unit.
- Disadvantage:
  - Only one processor can communicate with the memory or another processor at any given time.
  - As a consequence, the total overall transfer rate within the system is limited by the speed of the single path
- Part of the local memory may be designed as a cache memory attached to the CPU.

Multiport Memory

- A multiport memory system employs separate buses between each memory module and each CPU.
- The module must have internal control logic to determine which port will have access to memory at any given time.
- Memory access conflicts are resolved by assigning fixed priorities to each memory port.

- Advantage:
  - The high transfer rate can be achieved because of the multiple paths.
- Disadvantage:
  - It requires expensive memory control logic and a large number of cables and connections

Crossbar Switch

- Consists of a number of crosspoints that are placed at intersections between processor buses and memory module paths.
- The small square in each crosspoint is a switch that determines the path from a processor to a memory moudle.
- Advantage:
  - Supports simultaneous transfers from all memory modules
- Disadvantage:
  - The hardware required to implement the switch can become quite large and complex.

Inter Process Communication and Synchronization

- The instruction set of a multiprocessor contains basic instructions that are used to implement communication and synchronization between cooperating processes.
  - Communication refers to the exchange of data between different processes.
  - Synchronization refers to the special case where the data used to communicate between processors is control information.
- Synchronization is needed to enforce the correct sequence of processes and to ensure mutually exclusive access to shared writable data.

- Set of mechanisms for interprocess communication (IPC), including:
  - **Signals**
    - Signals are a lightweight way for processes to send notifications to each other.
    - Signals can be used to indicate that an event has occurred, such as the arrival of a new message or the termination of another process.
  - **Pipes and FIFOs**
    - Pipes and FIFOs are unidirectional channels that can be used to transfer data between processes.
    - Pipes are created by the kernel, while FIFOs are created by user space processes.

- **Sockets**
  - Sockets are bidirectional channels that can be used to transfer data between processes on the same host or on different hosts connected by a network.
  - Sockets are a more general-purpose IPC mechanism than pipes or FIFOs, and they are often used for applications that require reliable and high-performance communication.
- **File locking**
  - File locking allows a process to lock regions of a file in order to prevent other processes from reading or updating the file contents.
  - File locking is often used to implement synchronization between processes that are sharing a file.
- **Message queues**
  - Message queues are a way for processes to exchange messages (packets of data) with each other.

  - Message queues are a more flexible IPC mechanism than pipes or FIFOs, and they can be used to implement a variety of communication patterns.

- **Semaphores**
  - Semaphores are a way for processes to synchronize their actions.
  - Semaphores can be used to implement mutual exclusion, counting semaphores, and barriers.

- **Shared memory**
  - Shared memory is a way for two or more processes to share a piece of memory.
  - When one process changes the contents of the shared memory, all of the other processes can immediately see the changes.
  - Shared memory is a very efficient IPC mechanism, but it can be difficult to use correctly.

- Input or output devices that are connected to computer are called _______________.
  A. Input/Output Subsystem
  B. Peripheral Devices
  C. Interfaces
  D. Interrupt
- How many types of modes of I/O Data Transfer?
  A. 2
  B. 3
  C. 4
  D. 5
- The method which offers higher speeds of I/O transfers is ____________
  A. Interrupts
  B. Memory mapping
  C. Program-controlled I/O
  D. DMA

- In memory-mapped I/O _____________
  A. The I/O devices have a separate address space
  B. The I/O devices and the memory share the same address space
  C. A part of the memory is specifically set aside for the I/O operation
  D. The memory and I/O devices have an associated address space
- The _________ circuit is basically used to extend the processor BUS to connect devices.
  A. Router
  B. Router
  C. Bridge
  D. None of the above
- The registers of the controller are ______
  A. 16 bit
  B. 32 bit
  C. 64 bit
  D. 128 bit

- **The usual BUS structure used to connect the I/O devices is**
  a.Star BUS structure
  b.Multiple BUS structure
  c.Single BUS structure
  d.Node to Node BUS structure
- **The advantage of I/O mapped devices to memory mapped is**
  a. The former offers faster transfer of data
  b. The devices connected using I/O mapping have a bigger buffer space
  c. The devices have to deal with fewer address lines
  d. No advantage as such
-
  To overcome the lag in the operating speeds of the I/O device and the processor we use
  a. Buffer spaces
  b. Status flags
  c. Interrupt signals
  d. Exceptions

- **The method of accessing the I/O devices by repeatedly checking the status flags is**
  a. Program-controlled I/O
  b. Memory-mapped I/O
  c. I/O mapped
  d. None of the above

- **The process where in the processor constantly checks the status flags is called as**

  a. Polling

  b. Inspection

  c. Reviewing

  d. Echoing

- **The interrupt-request line is a part of the**

  a. Data line

  b. Control line

  c. Address line

  d. None of the above

-

  The data transfer rate is given by the formula

  a. Transfer size- transfer time

  b. Transfer size/transfer time

  c. Transfer size+transfer time

  d. Transfer size*transfer time

- Message passing system allows processes to __________
  a) communicate with each other without sharing the same address space
  b) communicate with one another by resorting to shared data
  c) share data
  d) name the recipient or sender of the message
- Which of the following two operations are provided by the IPC facility?
  a) write & delete message
  b) delete & receive message
  c) send & delete message
  d) receive & send message
- In indirect communication between processes P and Q __________
  a) there is another process R to handle and pass on the messages between P and Q
  b) there is another machine between the two processes to help communication
  c) there is a mailbox to help communication between P and Q
  d) none of the mentioned