**ACCESSING I/O DEVICES**

* Due to the various differences between the CPU and the peripherals, a medium for communication is needed in order to resolve the differences. This medium is known as an Interface.
* This interface resolves issues such as varying operating speeds, availability of CPU or the peripheral, etc.
* Helps in synchronizing peripherals with the CPU.
* Helps in providing control and timing signals.
* Converts digital into analog signals and vice-versa.
* Accessing I/O can be done in three ways:
  + Program I/O where the processor checks the I/O signals again and again which make it time consuming.
  + Interrupt Initiated I/O where every time the peripheral needs something done, an interrupt signal is sent to the processor. The processor finds it much free to perform everything without too much intervention.
  + DMA where the peripherals transact with memory without CPU’s intervention.

**INTERRUPT**

* It is a signal from a peripheral to the processor that it needs some action to be done.
* In hardware interrupts, multiple interrupting devices are connected with the only interrupt request line. The device making an interrupt request has its switch closed and in response to the interrupt request signal, the processor sends an acknowledgement signal to the respective device. This technique is known as Daisy chaining.
* In case of multiple interrupts, following three approaches can be considered:

1. **Polling:** Devices are checked on the basis of priority. The first device encountered with the IRQ bit set is the device to be served. Easy but time inefficient. It is a software method.
2. **Vectored:** Here, the interrupt request is sent along with a special code to enable the processor to identify the device which generated the first signal.
3. **Interrupt Nesting:** Here the devices are organised with priorities. The interrupt generating device with the highest priority is served. //daisy chaining hardware

* Interrupts can be handled in two ways: software and hardware discussed above.
* The programmer should have the facility to have control over the execution of a program. To ensure this, the processors are facilitated with an option for enabling or disabling interrupts.
* It also ensures that the processor doesn’t run into an infinite loop of interrupt requests.
* A bit known as Interrupt Enable (IE) of the status register can either be SET (1) to enable or RESET (0) to disable.

**CONTROLLING DEVICE REQUESTS**

* A mechanism is required to control whether a device can generate an interrupt signal or not.
* The control is usually provided in the form of an interrupt enable bit in the interfacing circuit of the device.

**EXCEPTIONS**

* An event the causes disruption in the normal flow of a program.
* Computers include an error-checking routine in the main memory. If an error is detected, an interrupt is raised. Then it suspends the main program to execute an exception-handling subroutine to recover from the error or informs the user of the error.

**DIRECT MEMORY ACCESS**

* Enables transaction between peripherals and memory without CPU intervention.
* Implemented using an interfacing hardware called DMA controller.
* I/O makes a DMA request, DMA makes a HOLD request to the CPU, CPU provides the base address and the bit count to the DMA and sends an HLDA signal, hence grant permission to access system bus. DMA sends a DMA ACK. signal and proceeds with the required work.

**BUS INTERFACE CIRCUIT**

* Mediator between I/O and the main device.
* The interface circuit may have serial or parallel ports.
* Please further discuss properties associated with serial and parallel ports.

**MEMORY SYSTEM**

* Memory is a component responsible to store and process data and store instructions.

**MAIN MEMORY**

* Faster as compared to secondary memory.
* Volatile usually.
* Integral part of a computer system.

**TYPES OF MEMORY**

* **PRIMARY MEMORY:**
* It is the main memory. It is of two types: