Contents

[1. Data Transfer Techniques 1](#_Toc149420457)

[1.1. Working of DMA 2](#_Toc149420458)

[2. DMA modes 3](#_Toc149420459)

[2.1. Burst Transfer Mode: - 3](#_Toc149420460)

[2.2. Single Mode or Cycle Steal Mode 3](#_Toc149420461)

[2.3. Scatter-Gather Mode 4](#_Toc149420462)

[2.4. Transparent Mode 4](#_Toc149420463)

# ****Data Transfer Techniques****

Data transfer between CPU and the I/O devices may be done in any of the three possible ways:

1. Programmed I/O.
2. Interrupt- initiated I/O.
3. Direct memory access (DMA).
4. **Programmed I/O:**

It is due to the result of the I/O instructions that are written in the computer program. Each data item transfer is initiated by an instruction in the program. Usually, the transfer is from a CPU register and memory. In this case, it requires constant monitoring by the CPU of the peripheral devices.

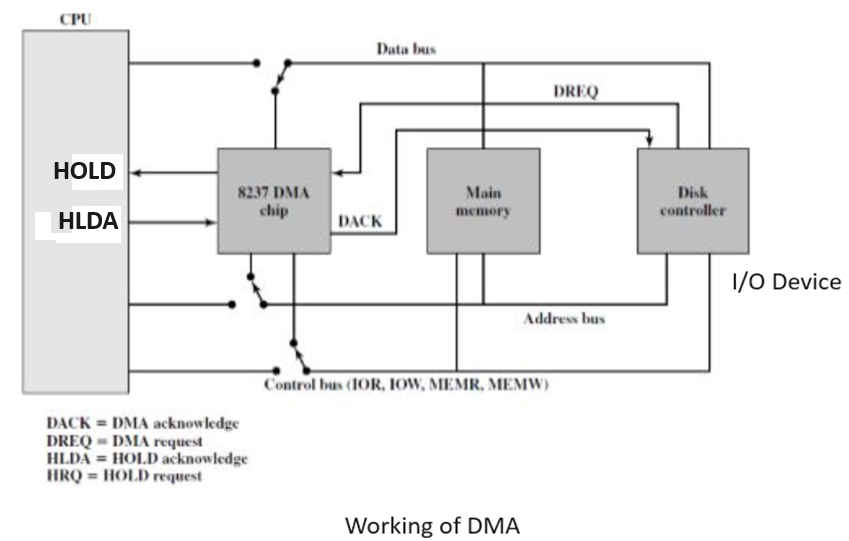
1. **Interrupt Initiated I/O: (Interrupt Driven I/O)**
2. **Direct memory access (DMA)**

Without DMA, when the CPU is using [programmed input/output](https://en.wikipedia.org/wiki/Programmed_input/output), it is typically fully occupied for the entire duration of the read or writes operation, and is thus unavailable to perform other work.

With DMA, the CPU first initiates the transfer, and then it does other operations while the transfer is in progress, and it finally receives an [interrupt](https://en.wikipedia.org/wiki/Interrupt) from the DMA controller (DMAC) 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 work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including [disk drive](https://en.wikipedia.org/wiki/Disk_storage) controllers, [graphics cards](https://en.wikipedia.org/wiki/Video_card), [network cards](https://en.wikipedia.org/wiki/Network_interface_controller) and [sound cards](https://en.wikipedia.org/wiki/Sound_card).

## ****Working of DMA****



The processor has two pins HOLD and HLDA which are used for DMA operation.

* Disk controller or I/O device sends DMA request (DREQ) to DMA controller, telling that it needs DMA transfer and then DMA controller sends a request by making Bus Request (BR) control line i.e. HOLD line high.
* CPU then shares two basic information with DMAC before the Data transfer which are: Starting address (memory address starting from where data transfer should be performed) and Data Count (no of bytes or words to be transferred).
* CPU completes the execution of current machine cycle and it takes few clocks and sends HLDA signal to the DMA controller.
* After receiving HLDA from CPU, the DMA controller sends DMA ACK (DACK) to I/O device and takes control over system bus and transfers data directly between memory and I/O without the involvement of the CPU. During DMA operation, the processor is free to perform next job which does not need system bus.
* At the end of data transfer, the DMA controller terminates the request by sending a low signal to HOLD pin and MP regains control of system bus by making HLDA low.

# ****DMA modes****

1. **Burst Transfer Mode**
2. **Single Mode or Cycle Steal Mode**
3. **Scatter-Gather Mode**
4. **Transparent Mode**

## ****Burst Transfer Mode: -****

* In 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, DMA controller transfers all bytes of data in the data block before releasing control of the system buses back to the CPU, but this way it keeps the CPU inactive for relatively long periods of time. The mode is also called "Block Transfer Mode".
* Processor is disconnected from system bus during DMA transfer, while transferring N number of bytes.
* DMA sends HOLD signal to processor to request system bus and waits for HLDA signal.
* After receiving HLDA signal, DMA gains control of system bus and transfers one byte. After transferring one byte, it increments memory address, decrements counter (which is initialized with no. of bytes to transfer i.e. N) and transfers next byte.
* In this way, it transfers all data bytes between memory and I/O devices. After transferring all data bytes, the DMA controller disables HOLD signal & enters into slave mode.

## ****Single Mode or**** Cycle Steal ****Mode****

* The [cycle stealing](https://en.wikipedia.org/wiki/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.
* In this mode only one byte is transferred at a time. This is slower than burst DMA.
* DMA sends HOLD signal to processor and waits for HLDA signal on receiving HLDA signal, it gains control of system bus and executes only one DMA cycle.
* After transfer one byte, it disables HOLD signal and enters into slave mode.
* Processor gains control of system bus and executes next machine cycle. If count is not zero and data is available then the DMA controller sends HOLD signal to the processor and transfer next byte of data block.

## ****Scatter-Gather Mode****

* In computing, **scatter/gather I/O**, also known as **vectored I/O**, is a method of input and output by which a single procedure call sequentially reads data from multiple buffers and writes it to a single data stream (gather), or reads data from a data stream and writes it to multiple buffers (scatter), as defined in a vector of buffers. *Scatter/gather* refers to the process of gathering data from, or scattering data into, the given set of buffers.
* Scatter-Gather DMA-Controller: This type of Direct Memory Access controller is used in systems where data needs to be transferred to or from multiple non-contiguous memory locations.

## ****Transparent Mode****

* 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. In transparent mode, the DMA controller transfers data only when the CPU is performing operations that do not use the system buses.
* The primary advantage of transparent mode is that the CPU never stops executing its programs and the DMA transfer is free in terms of time, while the disadvantage is that the hardware needs to determine when the CPU is not using the system buses, which can be complex.
* This is also called "Hidden DMA data transfer mode".