

Direct Memory Access (DMA)

Direct Memory Access (DMA) transfers the block of data between the *memory* and *peripheral devices* of the system, **without the participation** of the **processor**. The unit that controls the activity of accessing memory directly is called a **DMA controller**.

The processor **relinquishes the system bus** for a few clock cycles. So, the DMA controller can accomplish the task of data transfer via the system bus. In this section, we will study in brief about DMA, DMA controller, registers, advantages and disadvantages. So let us start.

Content: Direct Memory Access in Computer Architecture

1. [What is DMA and Why it is used?](#)
2. [DMA Controller's Working](#)
3. [DMA Block Diagram](#)
4. [Advantages and Disadvantages](#)
5. [Key Takeaways](#)

What is DMA and Why it is used?

Direct memory access (DMA) is a **mode of data transfer** between the memory and I/O devices. This happens **without the involvement** of the processor. We have two other methods of data transfer, **programmed I/O** and **Interrupt driven I/O**. Let's revise each and get acknowledge with their drawbacks.

In **programmed I/O**, the processor keeps on scanning whether any device is ready for data transfer. If an I/O device is ready, the processor **fully dedicates** itself in transferring the data between I/O and memory. It transfers data at a **high rate, but it can't get involved in any other activity** during data transfer. This is the major **drawback** of programmed I/O.

In **Interrupt driven I/O**, whenever the device is ready for data transfer, then it raises an **interrupt to processor**. Processor completes executing its ongoing instruction and saves its current state. It then switches to data transfer which causes a **delay**. Here, the processor doesn't keep scanning

for peripherals ready for data transfer. But, it is **fully involved** in the data transfer process. So, it is also not an effective way of data transfer.

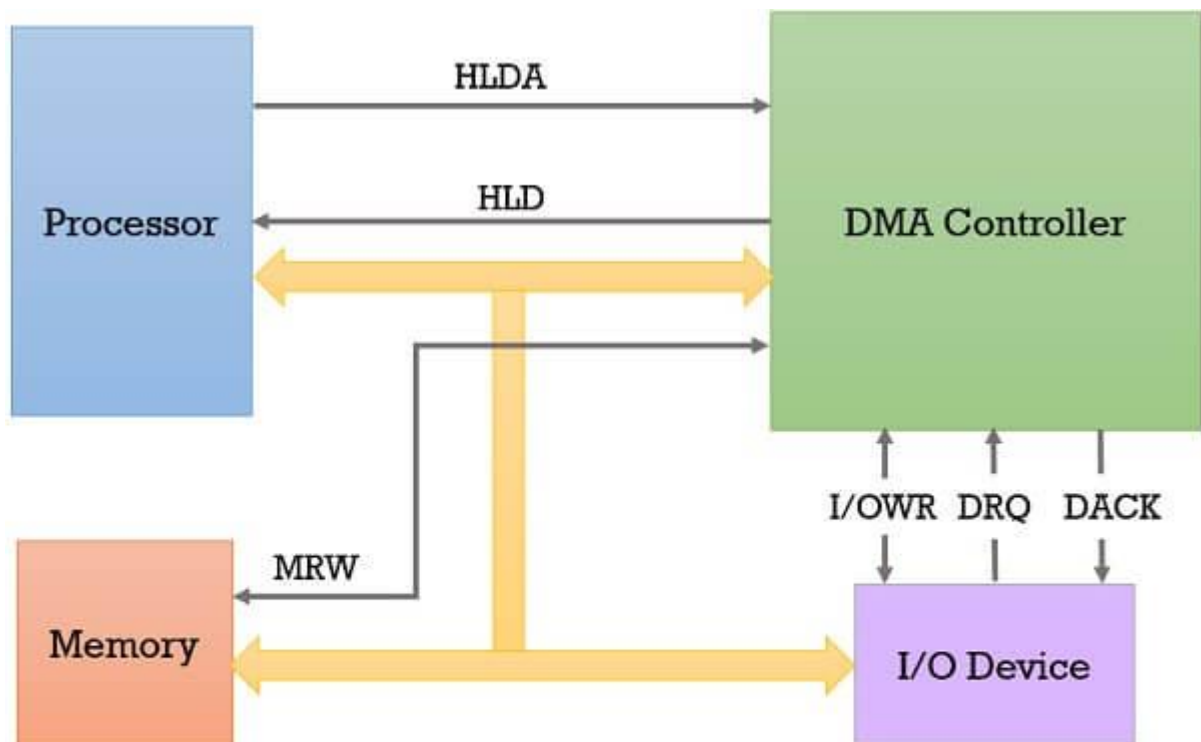
The above two modes of data transfer are not useful for transferring a large block of data. But, the DMA controller completes this task at a faster rate and is also effective for transfer of large data block.

The DMA controller transfers the data in three modes:

1. **Burst Mode:** Here, once the DMA controller gains the charge of the system bus, then it releases the system bus only after **completion** of data transfer. Till then the CPU has to wait for the system buses.
2. **Cycle Stealing Mode:** In this mode, the DMA controller **forces** the CPU to stop its operation and **relinquish the control over the bus** for a **short term** to DMA controller. After the **transfer of every byte**, the DMA controller **releases** the **bus** and then again requests for the system bus. In this way, the DMA controller steals the clock cycle for transferring every byte.
3. **Transparent Mode:** Here, the DMA controller takes the charge of system bus only if the **processor does not require the system bus**.

Direct Memory Access Controller & it's Working

DMA controller is a **hardware unit** that allows I/O devices to access memory directly without the participation of the processor. Here, we will discuss the working of the DMA controller. Below we have the diagram of DMA controller that explains its working:



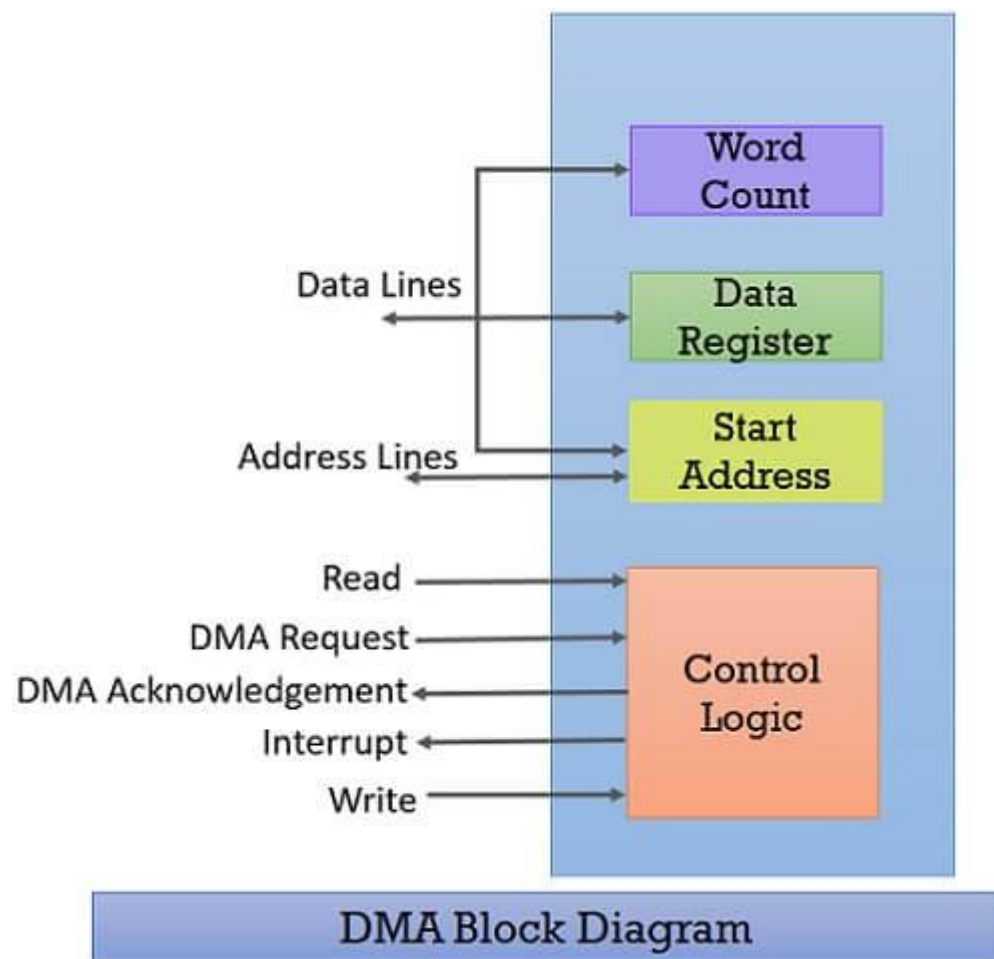
DMA Controller Data Transfer

1. Whenever an I/O device wants to transfer the data to or from memory, it sends the DMA request (**DRQ**) to the DMA controller. DMA controller accepts this DRQ and asks the CPU to hold for a few clock cycles by sending it the Hold request (**HLD**).
2. CPU receives the Hold request (HLD) from DMA controller and relinquishes the bus and sends the Hold acknowledgement (**HLDA**) to DMA controller.
3. After receiving the Hold acknowledgement (HLDA), DMA controller acknowledges I/O device (**DACK**) that the data transfer can be performed and DMA controller takes the charge of the system bus and transfers the data to or from memory.
4. When the data transfer is accomplished, the DMA raise an **interrupt** to let know the processor that the task of data transfer is finished and the processor can take control over the bus again and start processing where it has left.

Now the DMA controller can be a separate unit that is shared by various I/O devices, or it can also be a part of the I/O device interface.

Direct Memory Access Diagram

After exploring the working of DMA controller, let us discuss the block diagram of the DMA controller. Below we have a block diagram of DMA controller.



Whenever a processor is requested to read or write a block of data, i.e. transfer a block of data, it instructs the DMA controller by sending the following information.

1. The first information is whether the data has to be read from memory or the data has to be written to the memory. It passes this information via **read or write control lines** that is between the processor and DMA controllers **control logic unit**.
2. The processor also provides the **starting address** of/ for the data block in the memory, from where the data block in memory has to be read or where the data block has to be written in memory. DMA controller stores this in its **address register**. It is also called the **starting address register**.
3. The processor also sends the **word count**, i.e. how many words are to be read or written. It stores this information in the **data count** or the **word count** register.

4. The most important is the **address of I/O device** that wants to read or write data. This information is stored in the **data register**.

Direct Memory Access Advantages and Disadvantages

Advantages:

1. Transferring the data without the involvement of the processor will **speed up** the read-write task.
2. DMA **reduces the clock cycle** requires to read or write a block of data.
3. Implementing DMA also **reduces the overhead** of the processor.

Disadvantages

1. As it is a hardware unit, it would **cost** to implement a DMA controller in the system.
2. Cache **coherence** problem can occur while using DMA controller.

Key Takeaways

- DMA is an abbreviation of **direct memory access**.
- DMA is a **method of data transfer** between main memory and peripheral devices.
- The hardware unit that controls the DMA transfer is a **DMA controller**.
- DMA controller transfers the data to and from memory **without the participation of the processor**.
- The processor provides the **start address** and the **word count** of the data block which is transferred to or from memory to the DMA controller and **frees the bus for DMA controller** to transfer the block of data.
- DMA controller transfers the data block at the **faster rate** as data is directly accessed by I/O devices and is not required to pass through the processor which save the clock cycles.
- DMA controller transfers the block of data to and from memory in three modes **burst mode**, **cycle steal mode** and **transparent mode**.
- DMA can be configured in various ways it can be a part of individual I/O devices, or all the peripherals attached to the system may share the same DMA controller.

Thus the DMA controller is a convenient mode of data transfer. It is preferred over the programmed I/O and Interrupt-driven I/O mode of data transfer.