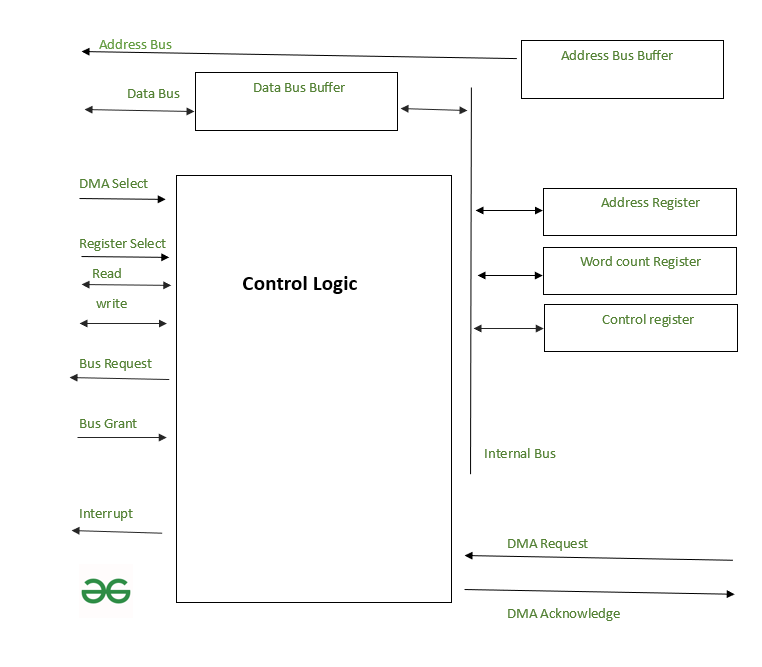
**Direct Memory Access**

Direct Memory Access (DMA) is a technique used in computers and other electronic devices to allow peripherals (like hard drives, network cards, and sound cards) to communicate directly with the main memory (RAM) without involving the CPU. This process speeds up data transfer and frees up the CPU to perform other tasks, improving overall system performance.

* The peripheral device sends a request to the DMA controller to initiate a data transfer.
* The DMA controller takes control of the system’s memory bus and accesses memory directly, either reading data from it or writing data to it.
* After the transfer is complete, the DMA controller signals the CPU that the task is finished, and the CPU can continue with other tasks.

**Working of DMA Transfer**

Below diagram represents a Direct Memory Access (DMA) controller and its components in a typical DMA system.

Working of DMA System

**DMA Controller Components**

1. **Control Logic:**The Control Logic is the central component that manages the overall DMA operation. It processes control signals and directs data transfers between the peripherals and memory. It receives commands from other components and determines how and when data should be moved.
2. **DMA Select and DMA Request:**DMA Select is used by the DMA controller to select the appropriate data transfer request. DMA Request is initiated by a peripheral device when it needs to perform a data transfer. The request tells the DMA controller that the device is ready to either read or write data.
3. **DMA Acknowledge:**The DMA Acknowledge signal is sent back from the control logic to the peripheral device to confirm that the DMA operation has been initiated and the device can proceed with the data transfer.
4. **Bus Request and Bus Grant:**Bus Request is generated by the DMA controller when it needs access to the system's bus for data transfer. The Bus Grant signal is sent from the CPU or the system’s bus controller to give the DMA controller permission to use the bus for transferring data.
5. **Address Bus and Data Bus:**The Address Bus and Data Bus are used to transfer data and memory addresses between the DMA controller, memory, and peripherals. The Data Bus Buffer temporarily holds data being transferred, while the Address Bus Buffer holds memory addresses.
6. **Registers:**
   * **Address Register**: This stores the memory address where data will be written or read from.
   * **Word Count Register**: This keeps track of the number of words or units of data that need to be transferred.
   * **Control Register**: This contains control information, including the direction of data transfer (read or write), and any other control signals necessary to manage the DMA operation.
7. **Internal Bus**: The Internal Bus connects all the components inside the DMA controller, allowing them to communicate and pass data efficiently.
8. **Interrupt:** The Interrupt signal is used to inform the CPU once the DMA operation is completed. After the data has been transferred, the DMA controller sends an interrupt to notify the CPU, so the CPU can resume processing or handle other tasks.

**Working:**

* The **DMA controller** facilitates the transfer of data between memory and peripherals without involving the CPU for each individual data operation, as mentioned in the article.
* The **DMA Select** and **DMA Request** initiate the process when a peripheral wants to transfer data, similar to how DMA allows peripherals to operate independently of the CPU.
* **Address Bus** and **Data Bus** handle the flow of data and memory addresses during the transfer, improving system efficiency by bypassing the CPU.
* The **Registers** (Address Register, Word Count Register, Control Register) store the necessary information to control the transfer, as described in the article, where DMA controls the movement of data between the device and memory.
* The **Interrupt** is triggered once the transfer is completed, similar to how the CPU is notified in the article that DMA operations have been finished.

**Types of DMA**

There are several types of DMA, each with its own way of transferring data:

1. **Burst Mode DMA**:
   * In this mode, the DMA controller takes control of the memory bus and transfers a block of data in one go.
   * The CPU is temporarily locked out of memory access while the DMA controller completes the data transfer.
2. **Cycle Stealing DMA**:
   * In this mode, the DMA controller transfers one data item at a time but allows the CPU to access memory between each transfer.
   * This allows the CPU and DMA controller to share the memory bus and work more collaboratively.
3. **Block Mode DMA**:
   * The DMA controller transfers a block of data without interruption, but it uses a more organized approach than burst mode, allowing for more efficient transfers.
   * The CPU is locked out of memory access during the transfer.
4. **Demand Mode DMA**:
   * In this mode, the DMA controller transfers data only when the CPU is not using the memory bus, essentially waiting for an idle time to perform the transfer.

**Direct Memory Access (DMA) Controller**

In modern computer systems, transferring data between input/output devices and memory can be a slow process if the CPU is required to manage every step. To address this, a Direct Memory Access (DMA) Controller is utilized.

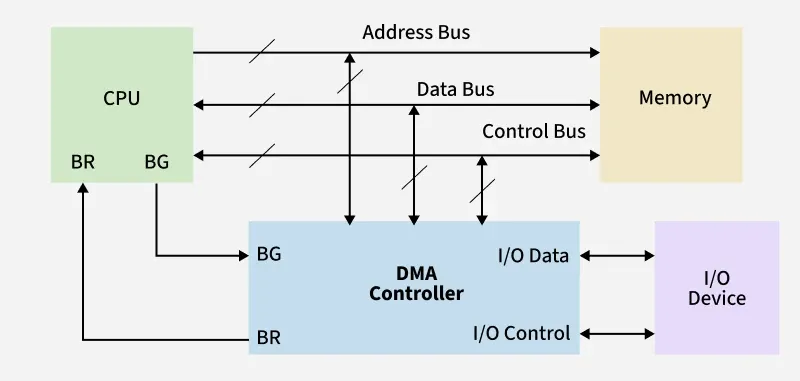
A Direct Memory Access (DMA) Controller solves this by allowing I/O devices to transfer data directly to memory, reducing CPU involvement. This increases system efficiency and speeds up data transfers, freeing the CPU to focus on other tasks. DMA controller needs the same old circuits of an interface to communicate with the CPU and Input/Output devices.

**DMA Controller**

[Direct Memory Access (DMA)](https://www.geeksforgeeks.org/computer-organization-architecture/direct-memory-access-with-dma-controller-8257-8237/)uses hardware for accessing the memory. This hardware is called a DMA Controller. It has the work of transferring the data between input, output devices and main memory with very less interaction with the processor. The Direct Memory Access Controller is a control unit, which has the work of transferring data.

**DMA Controller in Computer Architecture**

DMA Controller is a type of control unit that works as an interface for the data bus and the I/O Devices. As mentioned, DMA Controller has the work of transferring the data without the intervention of the processors, processors can control the data transfer. DMA Controller also contains an address unit, which generates the address and selects an I/O device for the transfer of data. Below is the block diagram of the DMA Controller.



Block Diagram of DMA Controller

**Types of Direct Memory Access (DMA)**

There are four popular types of DMA.

* **Single-Ended DMA:** In this type, the DMA controller is connected only to one device (usually either the memory or the I/O device), and it directly controls data transfer.
* **Dual-Ended DMA:**The DMA controller is connected to both the source and the destination, typically memory and an I/O device.
* **Arbitrated-Ended DMA:**In systems with multiple DMA devices or masters, arbitration is needed to decide which device gets control of the bus. It is more advanced than Dual-Ended DMA.
* **Interleaved DMA:**Interleaved DMA are those DMA that read from one memory address and write from another memory address.

**Working of DMA Controller**

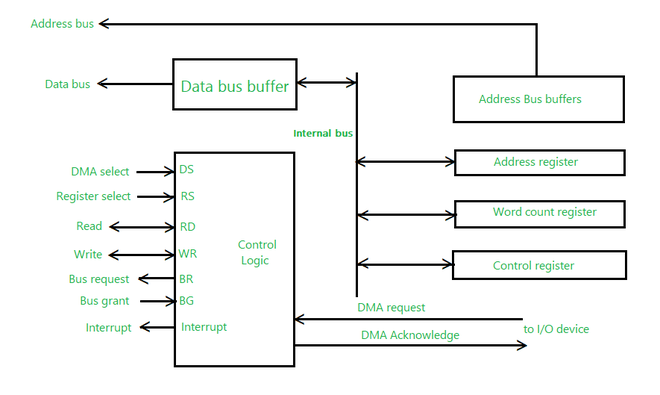
The [DMA controller registers](https://www.geeksforgeeks.org/computer-organization-architecture/internal-registers-of-dma-controller/) have three registers as follows.

* **Address register:**It contains the address to specify the desired location in memory.
* **Word count register:**It contains the number of words to be transferred.
* **Control register:**It specifies the transfer mode.

All registers in the DMA appear to the [CPU](https://www.geeksforgeeks.org/computer-organization-architecture/difference-between-cpu-and-gpu/) as I/O interface registers. Therefore, the CPU can both read and write into the DMA registers under program control via the data bus.

The figure below shows the block diagram of the DMA controller. The unit communicates with the CPU through the data bus and control lines. Through the use of the address bus and allowing the DMA and RS register to select inputs, the register within the DMA is chosen by the CPU.

RD and WR are two-way inputs. When BG (bus grant) input is 0, the CPU can communicate with DMA registers. When BG (bus grant) input is 1, the CPU has relinquished the buses and DMA can communicate directly with the memory.

Working Diagram of DMA Controller

The CPU initializes the DMA by sending the given information through the [data bus](https://www.geeksforgeeks.org/computer-organization-architecture/introduction-of-alu-and-data-path/).

* The starting address of the memory block where the data is available (to read) or where data are to be stored (to write).
* It also sends word count which is the number of words in the memory block to be read or written.
* Control to define the mode of transfer such as read or write.
* A control to begin the DMA transfer

**Modes of Data Transfer in DMA**

There are 3 [modes of data transfer](https://www.geeksforgeeks.org/computer-organization-architecture/modes-of-dma-transfer/) in DMA that are described below.

**Burst Mode**

* In Burst Mode, the DMA controller takes full control of the system bus and transfers the entire block of data in one go.
* The bus is not handed back to the CPU until the entire data transfer is complete.
* This mode is efficient for large data transfers but can delay CPU operations.

**Transparent Mode**

* In Transparent Mode, the DMA controller transfers data only when the CPU is not using the system bus.
* It effectively sneaks in transfers during idle CPU cycles, ensuring the CPU is never interrupted.
* Best when CPU performance is critical and some delay in data transfer is acceptable.

**Cycle Stealing Mode**

* In Cycle Stealing Mode, the DMA controller transfers one byte (or word) at a time and then releases control of the bus back to the CPU.
* This mode generates frequent bus requests but allows the CPU to execute instructions in between DMA transfers.
* Useful when the DMA task is important but should not entirely block the CPU, such as in audio or video streaming.

**Advantages of DMA Controller**

* DMA speeds up memory operations and data transfers by allowing peripherals to communicate directly with memory, bypassing the CPU.
* During data transfer, the CPU is not involved, which significantly reduces its workload.
* Operates efficiently, requiring very few clock cycles to complete data transfers.
* It distributes workload very appropriately.
* By offloading data movement tasks from the CPU, DMA distributes the overall system workload more effectively

**Disadvantages of DMA Controller**

* Costly operation due to the need for additional hardware and control logic.
* Suffers from [Cache-Coherence Problems](https://www.geeksforgeeks.org/operating-systems/cache-coherence/).
* It increases the overall cost of the system.
* It increases the complexity of the software.