**Direct Memory Access (DMA) in Computer Architecture**

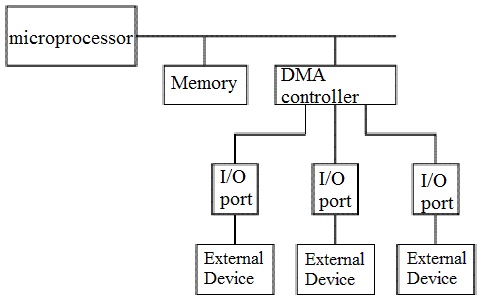
For the execution of a computer program, it requires the synchronous working of more than one component of a computer. For example, [Processors](https://www.elprocus.com/evolution-of-microprocessor-with-applications) – providing necessary control information, addresses…etc, buses – to transfer information and data to and from memory to I/O devices…etc. The interesting factor of the system would be the way it handles the transfer of information among processor, memory and I/O devices. Usually, processors control all the process of transferring data, right from initiating the transfer to the storage of data at the destination. This adds load on the processor and most of the time it stays in the ideal state, thus decreasing the efficiency of the system. To speed up the transfer of data between I/O devices and memory, DMA controller acts as station master. DMA controller transfers data with minimal intervention of the processor.

**What is a DMA Controller?**

The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller. DMA [controller is a control unit](https://www.elprocus.com/arm7-based-lpc2148-microcontroller-pin-configuration/), part of I/O device’s[interface circuit](https://www.elprocus.com/led-interfacing-with-8051-microcontroller/), which can transfer blocks of data between I/O devices and main memory with minimal intervention from the processor.

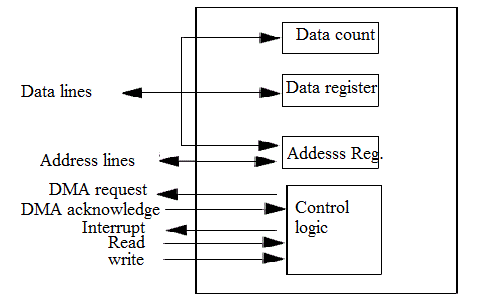
**DMA Controller Diagram in Computer Architecture**

DMA controller provides an interface between the bus and the input-output devices. Although it transfers data without intervention of processor, it is controlled by the processor. The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data .i.e. from I/O devices to the memory or from main memory to I/O devices. More than one external device can be connected to the DMA controller.



DMA in Computer Architecture

DMA controller contains an address unit, for generating addresses and selecting I/O device for transfer. It also contains the control unit and data count for keeping counts of the number of blocks transferred and indicating the direction of transfer of data. When the transfer is completed, DMA informs the processor by raising an interrupt. The typical block diagram of the DMA controller is shown in the figure below.

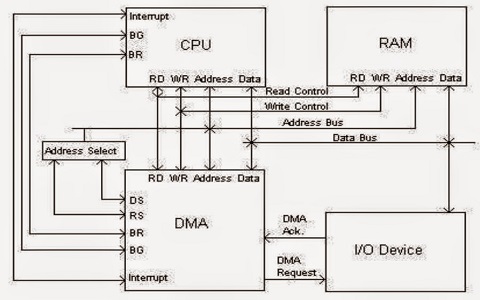


Typical Block Diagram of DMA Controller

**Working of DMA Controller**

DMA controller has to share the bus with the processor to make the data transfer. The device that holds the bus at a given time is called bus master. When a transfer from I/O device to the memory or vice verse has to be made, the processor stops the execution of the current program, increments [the program](https://www.elprocus.com/8086-assembly-language-programs-explanation/) counter, moves data over stack then sends a DMA select signal to DMA controller over the address bus.

If the DMA controller is free, it requests the control of bus from the processor by raising the bus request signal. Processor grants the bus to the controller by raising the bus grant signal, now DMA controller is the bus master. The processor initiates the DMA controller by sending the memory addresses, number of blocks of data to be transferred and direction of data transfer. After assigning the data transfer task to the DMA controller, instead of waiting ideally till completion of data transfer, the processor resumes the execution of the program after retrieving instructions from the stack.



Transfer Of Data in Computer By DMA Controller

DMA controller now has the full control of buses and can interact directly with memory and I/O devices independent of CPU. It makes the data transfer according to the control instructions received by the processor. After completion of data transfer, it disables the bus request signal and CPU disables the bus grant signal thereby moving control of buses to the CPU.

When an I/O device wants to initiate the transfer then it sends a DMA request signal to the DMA controller, for which the controller acknowledges if it is free. Then the controller requests the processor for the bus, raising the bus request signal. After receiving the bus grant signal it transfers the data from the device. For n channeled DMA controller n number of external devices can be connected.

The DMA transfers the data in three modes which include the following.

a) **Burst Mode**: In this mode DMA handover the buses to CPU only after completion of whole data transfer. Meanwhile, if the CPU requires the bus it has to stay ideal and wait for data transfer.

b) **Cycle Stealing Mode**: In this mode, DMA gives control of buses to CPU after transfer of every byte. It continuously issues a request for bus control, makes the transfer of one byte and returns the bus. By this CPU doesn’t have to wait for a long time if it needs a bus for higher priority task.

c) **Transparent Mode:** Here, DMA transfers data only when CPU is executing the instruction which does not require the use of buses.

**8237 DMA Controller**

* 8237 has 4 I/O channels along with the flexibility of increasing the number of channels.
* Each channel can be programmed individually and has a 64k address and data capability.
* The timing control block, Program command control block, Priority Encoder Block are the three main blocks of 8237A.
* The internal timing and external control signals are driven by the timing control block.
* Various commands given by the microprocessor to the DMA are decoded by program command control block.Which channel has to be given the highest priority is decided by the priority [encoder block](https://www.elprocus.com/encoders-and-decoders/) 8237A has 27 internal registers.

8237A operates in two cycles- Ideal cycle and active cycle, where each cycle contains 7 separate states composed of one clock period each.

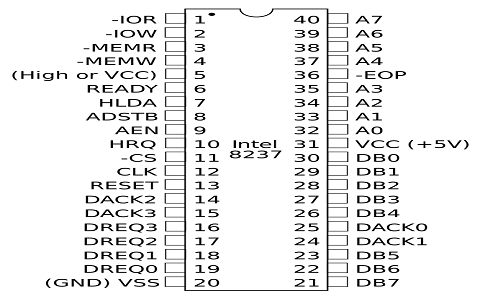
S0- The first state, where the controller has requested for the bus and waiting for the acknowledgment from the processor.

S1, S2, S3, S4 are called the working states of the 8237A where the actual transfer of data takes place. If more time is needed for transfer wait states SW are added between these states.

For memory –to- memory transfer read-from-memory and write-to-memory transfers have to be made. Eight states are required for single transfer. The first four states with subscripts S11, S12, S13, S14 does the read-from-memory transfer and the next four S21, S22, S23, S24 are for write-to-memory transfer.

DMA goes into the ideal state when no channel is requesting service and perform SI state. SI is an inactive state where the DMA is inactive until it receives a request. In this state, DMA is in program condition where the processor can program the DMA.

When DMA is in the ideal state and gets no further channel requests, it outputs an HRQ signal to the processor and enters into Active state where it can start the transfer of data either by burst mode, cycle stealing mode or transparent mode.



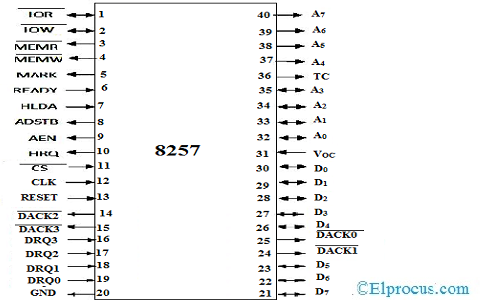
8237 Pin Diagram

**8257 DMA Controller**

When paired with single Intel 8212 I/O port device, the 8257 DMA controller forms a complete 4 channel **DMA controller**. Upon receiving a transfer request the 8257 controller-

* Acquires the control over system bus from the processor.
* The peripheral connected to the highest priority channel is acknowledged.
* The least significant bits of the memory address are moved over the address lines A0-A7 of the system bus.
* The most significant [8 bits](https://www.elprocus.com/8085-microprocessor-pin-diagram-and-its-description) of the memory address are driven to 8212 I/O port through data lines.
* Generates the appropriate controls signals for the transfer of data between peripherals and addressed memory locations.
* When the specified number of bytes are transferred, the controller informs the CPU end of transfer by activating the terminal count ( TC) output.

For each channel 8257 contains two [16-bit registers](https://www.elprocus.com/know-about-types-of-registers-in-8051-microcontroller)– 1) DMA address register and 2) Terminal count register, which should be initialized before a channel is enabled. The address of first memory location to be accessed is loaded in the DMA address register. The lower order 14 bits of the value loaded in the terminal count register indicates the number of DMA cycles minus one before the activation of Terminal count output. Type of operation for a channel is indicated by the most significant two bits of the Terminal count register.



8257 Pin Diagram

**Advantages and Disadvantages of DMA Controller**

The advantages and disadvantages of DMA controller include the following.

**Advantages**

* DMA speedups the memory operations by bypassing the involvement of the CPU.
* The work overload on the CPU decreases.
* For each transfer, only a few numbers of clock cycles are required

**Disadvantages**

* Cache coherence problem can be seen when DMA is used for data transfer.
* Increases the price of the system.

DMA ([Direct Memory Access](https://en.wikipedia.org/wiki/Direct_memory_access)) controller is being used in graphics cards, network cards, sound cards etc… DMA is also used for intra-chip transfer in multi-core processors. Operating in one of its three modes, DMA can considerably reduce the load of the processor. In which of the modes of DMA have you worked with? Which of the mode you consider is more effective?