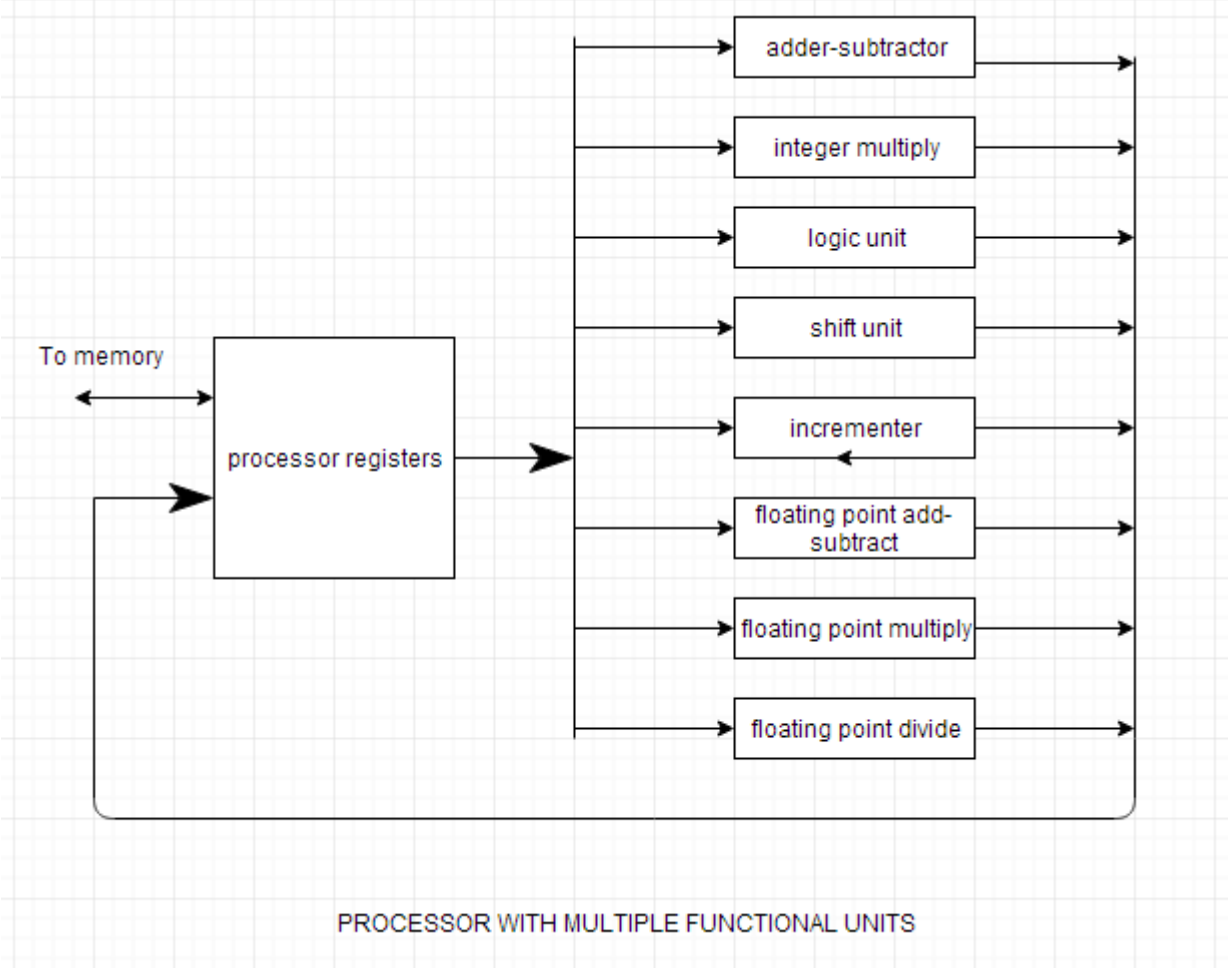
Parallel Processing and Data Transfer Modes in a Computer System

Instead of processing each instruction sequentially, a **parallel processing** system provides concurrent data processing to increase the execution time.

In this the system may have two or more ALU's and should be able to execute two or more instructions at the same time. The purpose of parallel processing is to speed up the computer processing capability and increase its throughput.

NOTE: Throughput is the number of instructions that can be executed in a unit of time.

Parallel processing can be viewed from various levels of complexity. At the lowest level, we distinguish between parallel and serial operations by the type of registers used. At the higher level of complexity, parallel processing can be achieved by using multiple functional units that perform many operations simultaneously.



PROCESSOR WITH MULTIPLE FUNCTIONAL UNITS

Data Transfer Modes of a Computer System

According to the data transfer mode, computer can be divided into 4 major groups:

1. SISD

2. SIMD
3. MISD
4. MIMD

SISD (Single Instruction Stream, Single Data Stream)

It represents the organization of a single computer containing a control unit, processor unit and a memory unit. Instructions are executed sequentially. It can be achieved by pipelining or multiple functional units.

SIMD (Single Instruction Stream, Multiple Data Stream)

It represents an organization that includes multiple processing units under the control of a common control unit. All processors receive the same instruction from control unit but operate on different parts of the data.

They are highly specialized computers. They are basically used for numerical problems that are expressed in the form of vector or matrix. But they are not suitable for other types of computations

---

MISD (Multiple Instruction Stream, Single Data Stream)

It consists of a single computer containing multiple processors connected with multiple control units and a common memory unit. It is capable of processing several instructions over single data stream simultaneously. MISD structure is only of theoretical interest since no practical system has been constructed using this organization.

MIMD (Multiple Instruction Stream, Multiple Data Stream

It represents the organization which is capable of processing several programs at same time. It is the organization of a single computer containing multiple processors connected with multiple control units and a shared memory unit. The shared memory unit contains multiple modules to communicate with all processors simultaneously. Multiprocessors and multicomputer are the examples of MIMD. It fulfills the demand of large scale computations.

Synchronous and Asynchronous data transfer

In synchronous data transfer, the transmitter and the receiver are synchronized with the same clock pulse. In asynchronous data transfer, the transmitter and the receiver do not use a common timing signal. That is the main difference between synchronous and asynchronous data transfer.

Data transferring is the process of sending data from the transmitter (sender) to the receiver. It can be synchronous or asynchronous. Synchronous data transfer uses synchronized clocks to transmit data. In contrast, asynchronous data transfer uses a flow control method of sending start and stop bits with data.

SYNCHRONOUS DATA TRANSFER VERSUS ASYNCHRONOUS DATA TRANSFER

SYNCHRONOUS DATA TRANSFER	ASYNCHRONOUS DATA TRANSFER
A data transfer method that sends a continuous stream of data to the receiver using regular timing signals that ensures both transmitter and receiver are synchronized with each other	A data transfer method that sends data from transmitter to receiver with parity bits (start and stop bits) in uneven intervals
Sender and receiver operate on the same clock frequencies	Sender and receiver operate on different clock frequencies
Faster	Slower
There is no overhead of extra start and stop bits	Uses start and stop bits
There are no gaps between data - data flows as a continuous stream	There can be gaps between data
Uses constant time intervals	Uses random or irregular time intervals
Used in chat rooms and video conferencing	Used in emails

Direct Memory Access (DMA) in Computer Architecture

DMA stands for "Direct Memory Access" and is a method of transferring data from the computer's RAM to another part of the computer without processing it using the CPU. While most data that is input or output from your computer is processed by the CPU, some data does not require processing, or can be processed by another device.

In these situations, DMA can save processing time and is a more efficient way to move data from the computer's memory to other devices.

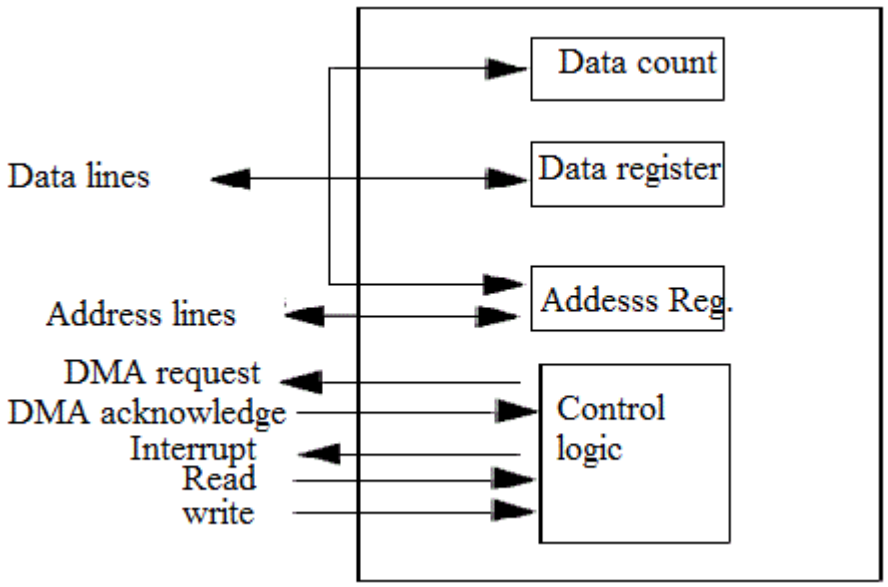
What is a DMA Controller?

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, part of I/O device's interface circuit, 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 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.

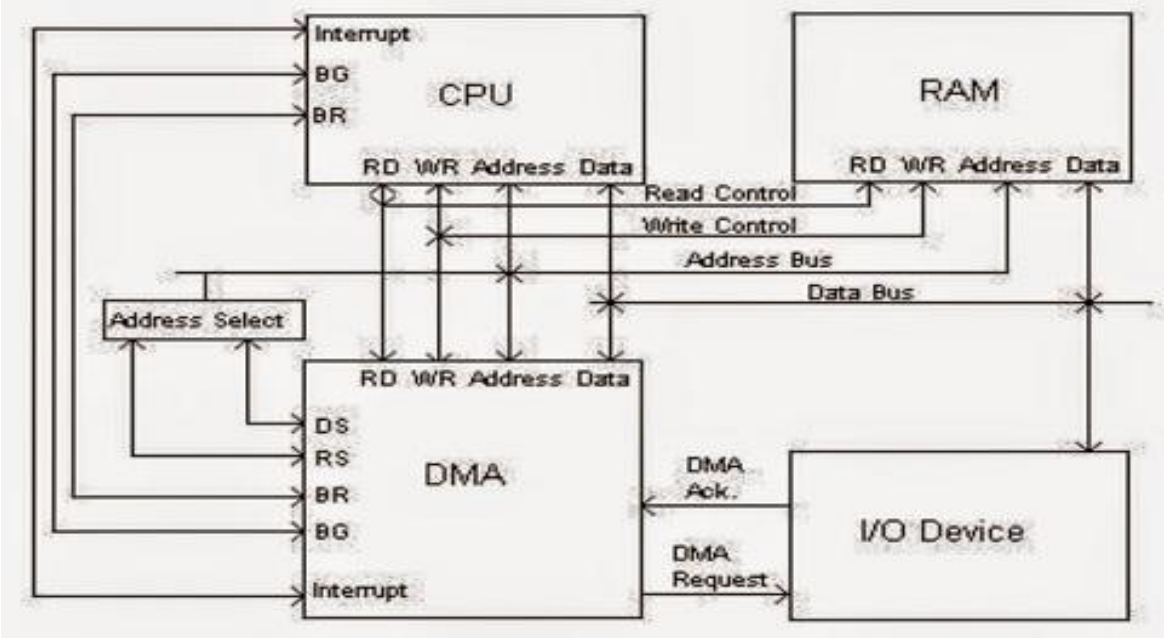


Typical Block Diagram of DMA Controller

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 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.

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) controller is being used in graphics cards, network cards, sound cards etc… DMA is also used for intra-chip transfer in multi-core processors.

Computer Architecture: Input/Output Processor

An input-output processor (IOP) is a processor with direct memory access capability. In this, the computer system is divided into a memory unit and number of processors.

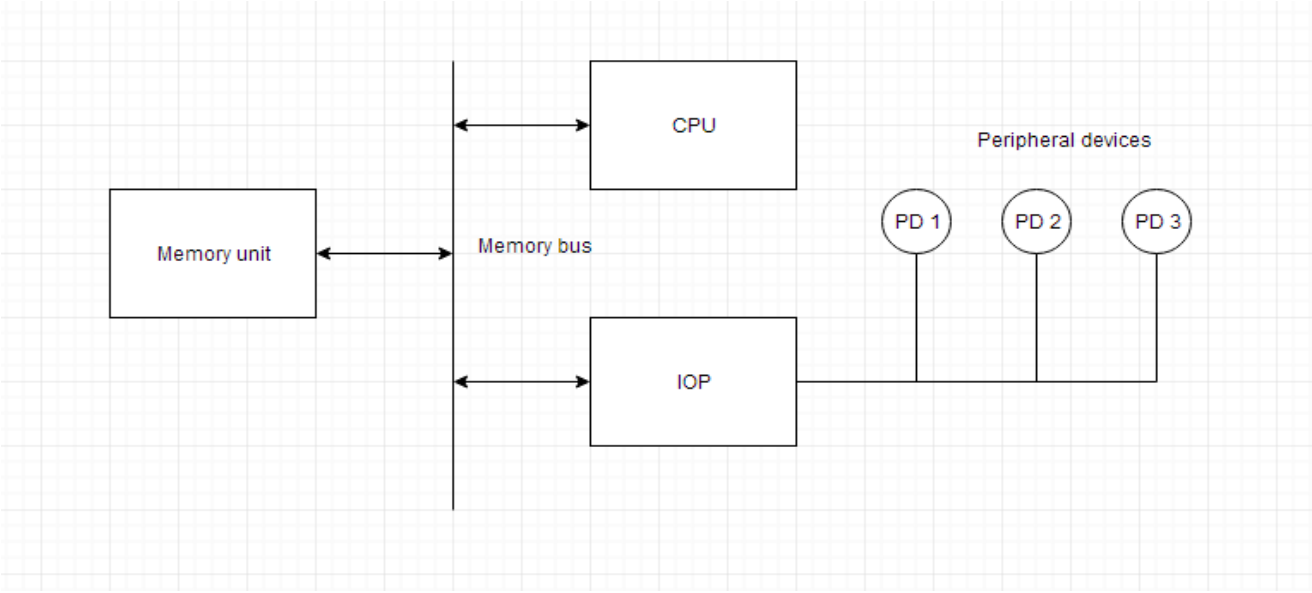
Each IOP controls and manage the input-output tasks. The IOP is similar to CPU except that it handles only the details of I/O processing. The IOP can fetch and execute its own instructions. These IOP instructions are designed to manage I/O transfers only.

Block Diagram Of I/O Processor

Below is a block diagram of a computer along with various I/O Processors. The memory unit occupies the central position and can communicate with each processor.

The CPU processes the data required for solving the computational tasks. The IOP provides a path for transfer of data between peripherals and memory. The CPU assigns the task of initiating the I/O program.

The IOP operates independent from CPU and transfer data between peripherals and memory.



The communication between the IOP and the devices is similar to the program control method of transfer. And the communication with the memory is similar to the direct memory access method.

In large scale computers, each processor is independent of other processors and any processor can initiate the operation.

The CPU can act as master and the IOP act as slave processor. The CPU assigns the task of initiating operations but it is the IOP, who executes the instructions, and not the CPU. CPU instructions provide operations to start an I/O transfer. The IOP asks for CPU through interrupt.

Instructions that are read from memory by an IOP are also called *commands* to distinguish them from instructions that are read by CPU. Commands are prepared by programmers and are stored in memory. Command words make the program for IOP. CPU informs the IOP where to find the commands in memory.

What does *Serial Communication* mean?

Serial communication is a communication technique used in telecommunications wherein data transfer occurs by transmitting data one bit at a time in a sequential order over a computer bus or a communication channel. It is the simplest form of communication between a sender and a receiver. Because of the synchronization difficulties involved in parallel communication, along with cable cost, serial communication is considered best for long-distance communication.

Serial Communication

In contrast to parallel communication, which is half duplex, serial communication is full duplex, i.e., transmission and receipt of signals can occur simultaneously. It is the most popular mode of communication protocol for most instrumentation devices. It is also popular in computer devices, peripheral devices and integrated circuits, which are provided with one or more serial ports, resulting in no additional hardware requirements for serial communication.

There are several advantages with serial communication. As there are fewer conductors in contrast to parallel communication, the cross-talk issue is significantly less. Interconnecting cables are fewer, and there is no need for a serializer/deserializer in any case. The data transfer rate, however, may be low in comparison to parallel communication. Nevertheless, the clock skew problem that often happens between different channels of communication is not an issue with serial communication.

Compared to parallel communication, serial communication has better signal integrity. In addition, serial communication is one of the cheapest modes of communication that can be implemented, and over long-haul communication, it can provide numerous benefits.