

UNIT-5

INPUT/OUTPUT

Peripheral Devices:-

- The input-output subsystem of a computer, referred to as I/O, provides an efficient mode of communication between central system and the outside environment.
- A computer serves no useful purpose without the ability to receive information from an outside source and to transmit results in a meaningful form.
- The most familiar means of entering information into a computer is through a typewriter like keyboard that allows a person to enter alphanumeric info. directly. Everytime a key is depressed, the terminal sends a binary coded character to the computer.
 - Quite slow process
 - To use a computer efficiently, a large amount of programs and data must be prepared in advance and transmitted into a storage medium such as magnetic tapes or disks. The information in the disk is then transferred into computer memory at a rapid rate.
 - Similarly, results are also transferred into a high-speed storage and later into a printer.

- Devices that are used under the direct control of the computer are said to be connected on-line.
- Designed to read information into or out of the memory unit upon command from the CPU and are considered to be part of the total computer system.
- Input or output devices attached to the computer are also called peripherals. eg:- keyboards, display units and printers.
- Peripherals that provide auxiliary storage for the system are magnetic disks and tapes.
- Peripherals are electromechanical and electromagnetic devices of some complexity.

Input-Output Interface:

- Input-Output interface provides a method for transferring information between internal storage and external I/O devices.
- Peripherals connected to a computer need special communication links for interfacing them with the central processing unit. To resolve the differences that exist between the central computer and each peripheral.
- The major differences are -
 - 1) A conversion of signal values may be required.
(Peripherals - electromechanical and electromagnetic devices, CPU and memory - electronic devices).
 - 2) The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and

consequently, a synchronization mechanism may be needed.

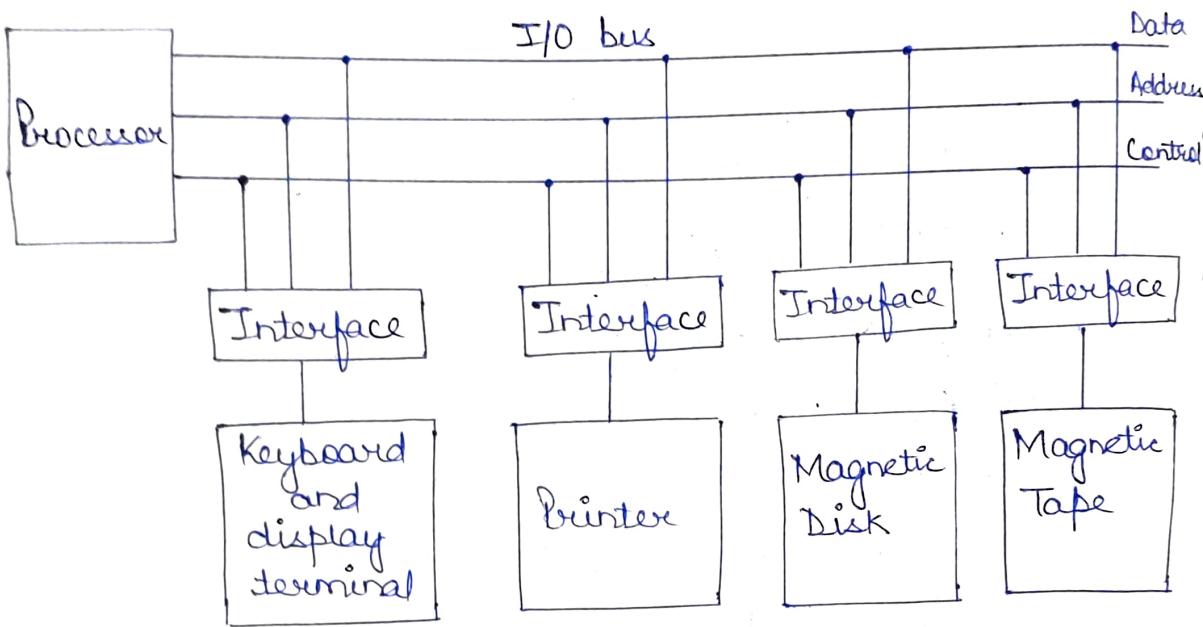
- 3) Data codes and formats in peripherals differ from the word format in the CPU and memory.
 - 4) The operating modes of peripherals are different from each other and each must be controlled so as not to disturb the operation of other peripherals connected to the CPU.
- To resolve these differences, computer systems include special hardware components (called interface units) between the CPU and peripherals to supervise and synchronize all inputs and output transfers.
- They interface between the processor bus and the peripheral device.
- In addition, each device may have its own controller that supervises the operations of the particular mechanism in the peripherals.

I/O Bus and Interface Modules :-

- Each interface decodes the address and control received from the I/O bus, interprets them for the peripherals and provides signals for the peripheral controller.
- It also synchronizes the data flow and supervises the transfer between peripheral and processor.
- Each peripheral has its own controller that operates the particular electromechanical device.
- . For example - the printer controller controls the paper motion, the paper timing and the selection of printing

characters.

- A controller may be housed separately or may be physically integrated with the peripherals.



Connection of I/O bus to input-output devices

- The processor places a device address on the address lines.
- Each interface contains an address decoder that monitors the address lines.
- When the interface detects its own address, it activates the path between the bus lines and the device that it controls.
- All peripherals whose address does not correspond to the address in the bus are disabled by their interface.

I/O Command :-

- At the same time that the address is made available in the address lines, the processor provides a function code in the control lines.

- The interface selected responds to the function code and proceeds to execute it.
- Few types of commands -
 - Control
 - Status
 - Data Output
 - Data Input

① Control Command :-

- A control command is issued to activate the peripheral and to inform it what to do.

② Status :-

- A status command is used to test various status conditions in the interface and the peripheral.

③ Output data :-

- It causes the interface to respond by transferring data from the bus into one of its registers.
- Consider an example with a tape unit - the computer starts the tape moving by issuing a control command. The processor then monitors the status of the tape by means of a status command. When the tape is in the correct position, the processor issues a data output command. The interface responds to the address and command and transfers the information from the data lines in the bus to its buffer register. The interface then communicates with the tape controller and sends the data to be stored on tape.

④ Input data :-

- The interface receives an item of data from the peripheral and places it in its buffer register.

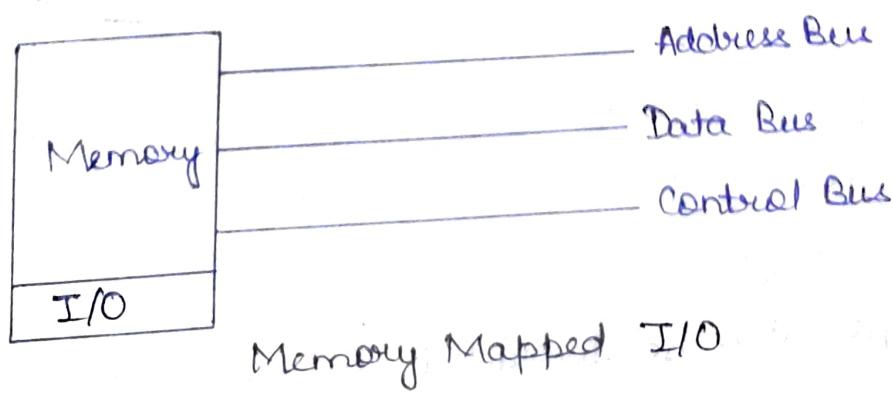
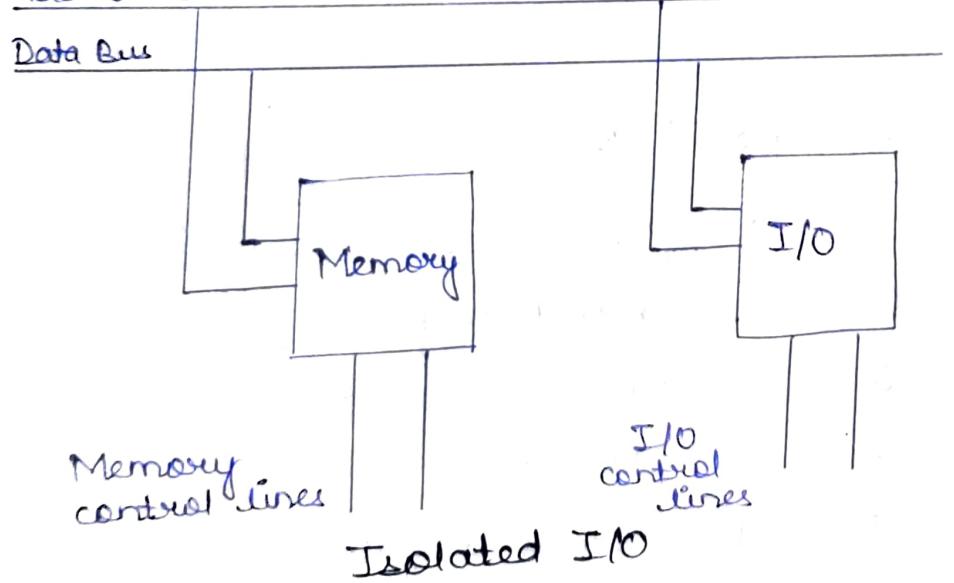
- The processor checks if data are available by means of a status command and then issues a data input command.
- The interface places the data on the data lines, where they are accepted by the processor.

I/O versus Memory Bus :-

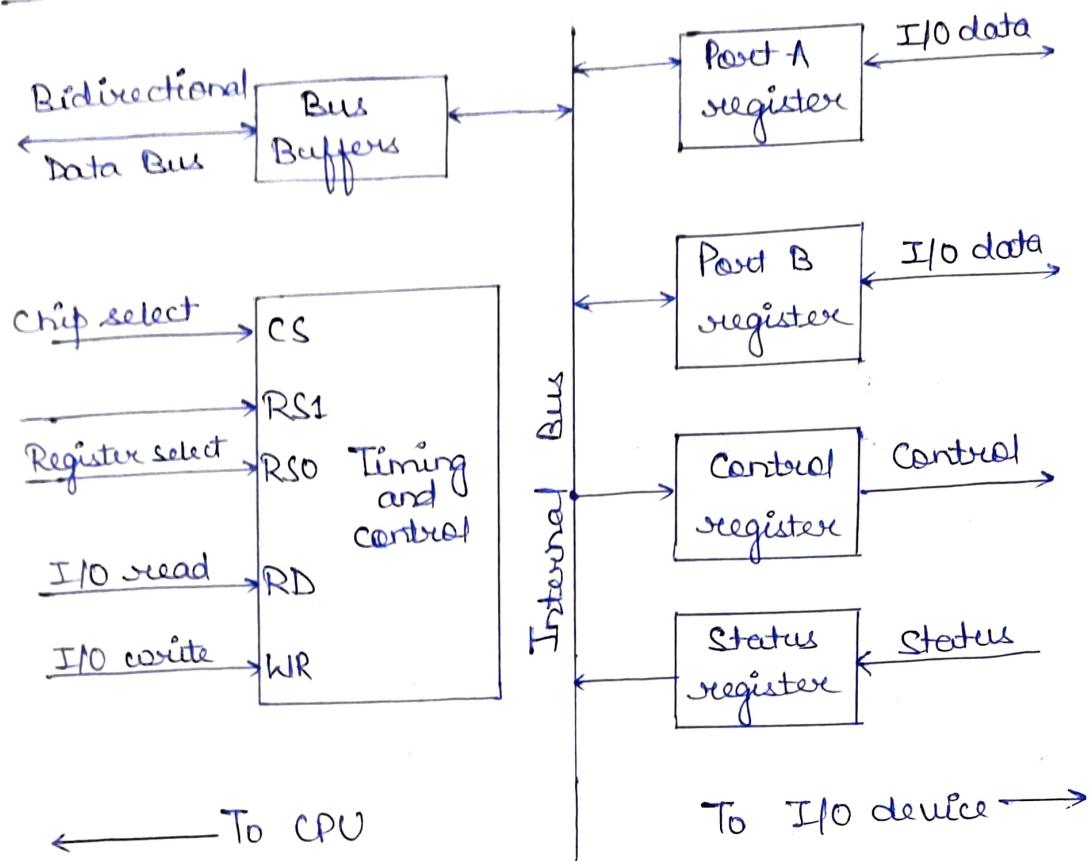
- In addition to communicating with I/O, the processor must communicate with the memory unit.
- 3 such ways:-
 - i) Use 2 separate busy buses, one for memory and other for I/O.
 - ii) Use one common bus for both memory and I/O but have separate control lines for each.
 - iii) Use one common bus for memory and I/O with common control lines.
- IOP :-
 - In the first method, the computer has independent sets of data, address and control buses, one for accessing memory and other for I/O.
 - This is done in computers that provide a separate I/O processor (IOP) in addition to the central processing unit (CPU).
 - The I/O processor is sometimes called a data channel.

- Isolated versus Memory-Mapped I/O
- Isolated I/O :-
- Separate I/O read/write control lines in addition to memory read/write control lines.
- Separate (isolated) memory and I/O address spaces.
- Distinct input and output instructions.
- Memory-mapped I/O :-
- A single set of read/write control lines (no distinction between memory and I/O transfer).
- Memory and I/O addresses share the common address space, which reduces memory address range available.

Address Bus



Example of I/O Interface :-



CS	RS1	RS0	Register Selected
0	x	x	None: data bus in high-impedance
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

INTERRUPTS :-

- Interrupts is a process that allows the processor to suspend its current execution and respond to external/internal requests.
- Interruptions can be provided to the processor by 2 ways:-

- i) program written in computer memory.
- ii) through external peripherals.

There are following types of Interrupts :-

- i) Hardware v/s software interrupt
- ii) Maskable v/s Non-maskable.
- iii) Vectored v/s Non-vectorized

1) Hardware and software interrupts

- When interrupts are through an external hardware by sending a signal to a particular pin of the processor, they are called hardware interrupt.

The software interrupts are initiated by programs, while executing a program, a software interrupt instruction is encountered the processor control is given back to the system.

2) Maskable and Non-maskable interrupts

- Instructions which require immediate action by the processor are called non-maskable interrupts or enable interrupts.

Some instructions may not be responded by the processor immediately or even processor may reject it, are classified under maskable interrupts.

3) Vectored and Non-vectorized interrupts

- When interrupts are provided or given, the execution of one program stops. In vectorized interrupt, the branch address is assigned to a fixed location in memory. eg. 8085 processors.

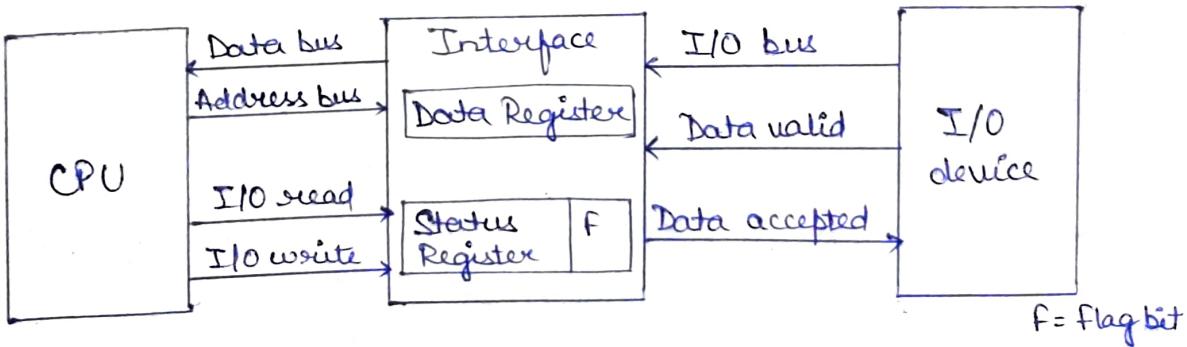
In non-vectorized interrupts the source that interrupt supplies the branch information to the processor.

MODES OF DATA TRANSFER :-

- Data transfer between the central computer to I/O devices may be handled in three of modes:-
 - 1) Programmed I/O
 - 2) Interrupt Initiated I/O
 - 3) Direct Memory Access (DMA)

Programmed I/O Transfer :-

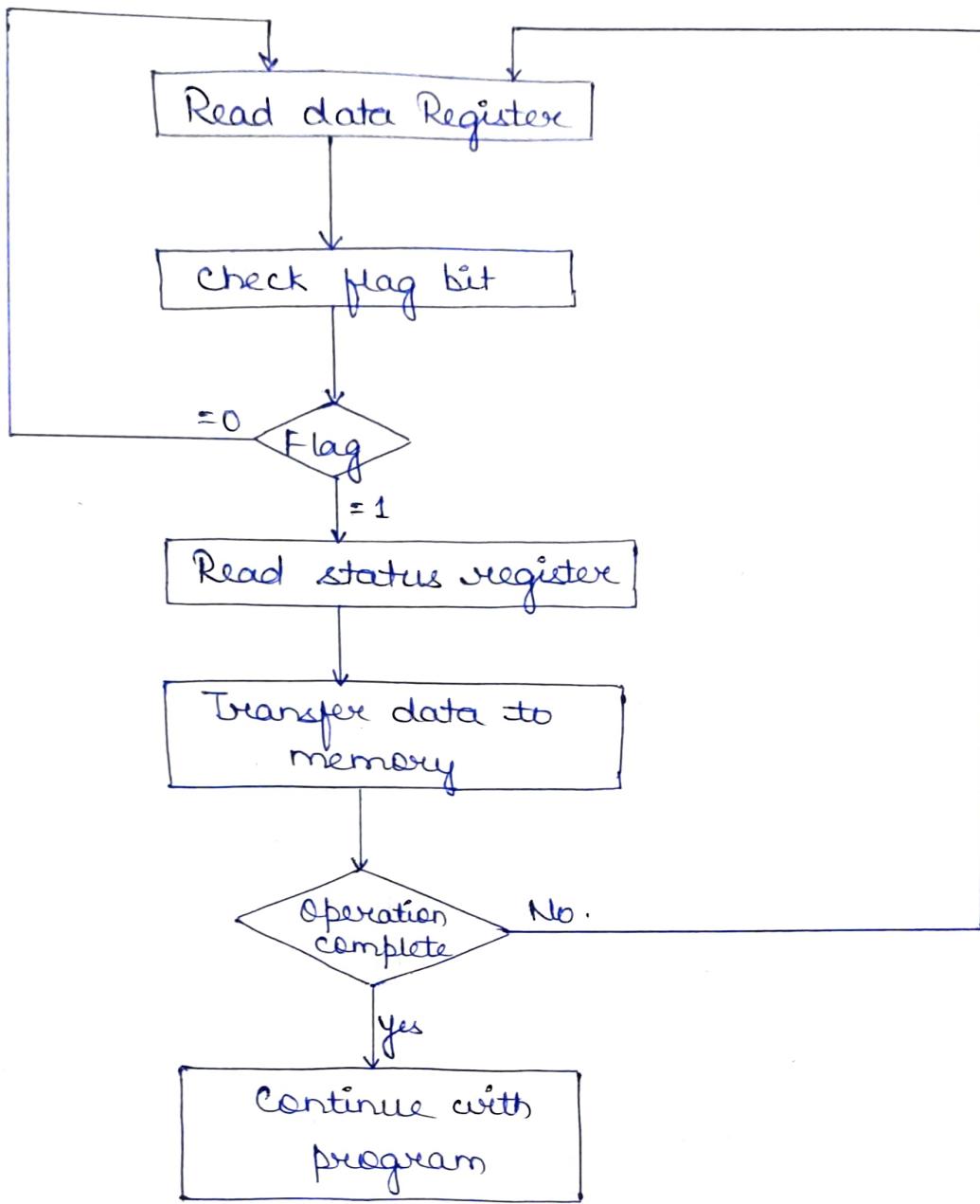
- The programmed I/O operations are result of I/O instⁿ written in the computer program. Each data item transfer is initiated by an instruction in the program. Usually, the transfer is to or from CPU register and peripheral. Other instructions are needed to transfer the data to and from CPU and memory.
- In the programmed I/O method, the CPU stays in the program loop until the I/O unit indicate that it is ready for data transfer. This is time-consuming process since it keeps the processor busy needlessly. It can be avoided by using an interrupt facility.



Data transfer from I/O device to CPU.

Example of Programmed I/O :-

- In this method, the I/O device does not have direct access to memory. Transfer from I/O device to memory requires the execution of several instructions by the CPU, including an input instruction to transfer the data from the device to CPU and store instruction to transfer the data from CPU to memory.
- The I/O device transfers bytes of data one at a time, as they are available. When a byte of data is available the device places it in the I/O bus and enables its data valid line.
- The interface accepts the byte into its data register and enables the data accepted line. The interface sets the flag bit f. The device then disable the data valid line but it will not transfer another byte until the data accepted line is disabled by the interface.
- If the flag bit is equal to 1, CPU reads the data from data register. The flag bit is then cleared to 0 by either CPU or interface, depending on how the interface circuits are designed. Once the flag is cleared, the interface disables the data accepted line and the device can transfer the next data byte.



Flowchart for CPU program to input data.

Interrupt-Initiated I/O Transfer :-

- The problem with programmed I/O is that the processor has to wait a long time for I/O module of concern to be ready for either reception or transmission of data. The processor, while waiting, must repeatedly interrogate the status of the I/O module. As a result, the level of the performance of the entire system is severely degraded.

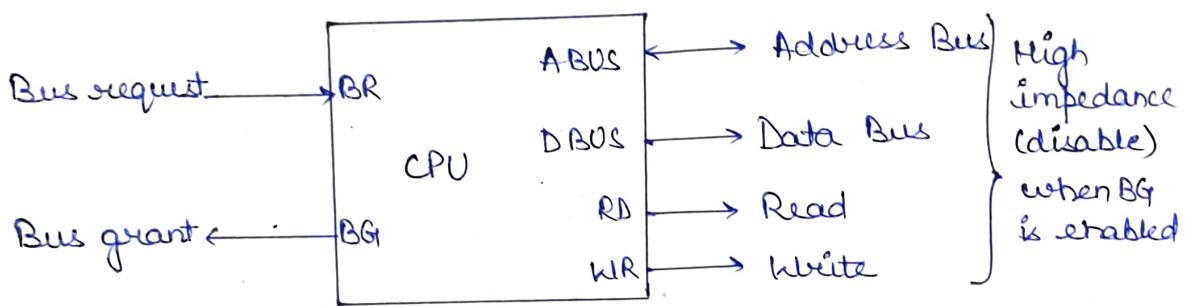
- An alternative is for the processor to issue an I/O command to a module and then go on to do some other useful work.
- The I/O module will then interrupt the processor to request service when it is ready to exchange data with the processor.
- The processor then executes the data transfer, as before, and then resumes its former processing.

Interrupt Processing :-

- A device driver initiates an I/O request on behalf of a process.
- The device driver signals the I/O controller for the proper device, which initiates the requested I/O.
- The device signals the I/O controller that is ready to retrieve input, the output is complete or that an error has been generated.
- The CPU receives the interrupt signal on the interrupt-request line and transfers control over the interrupt handler routine (also known as an interrupt service routine or ISR, which is special block of code associated with a specific interrupt condition).
- The interrupt handler determines the cause of the interrupt performs the necessary processing and executes a "return from" interrupt instruction.
- The CPU returns to the execution state prior to the interrupt being signaled.
- The CPU continues processing until the cycle begins again.

Direct Memory Access (DMA) :-

- The transfer of data between a fast storage device such as magnetic disk and memory is often limited by the speed of the CPU. Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This transfer technique is called Direct Memory Access (DMA).
- During DMA transfer, the CPU is ~~ideal~~ idle and has no control of the memory buses.
- A DMA controller takes over the buses to manage the transfer directly between the I/O device and memory.

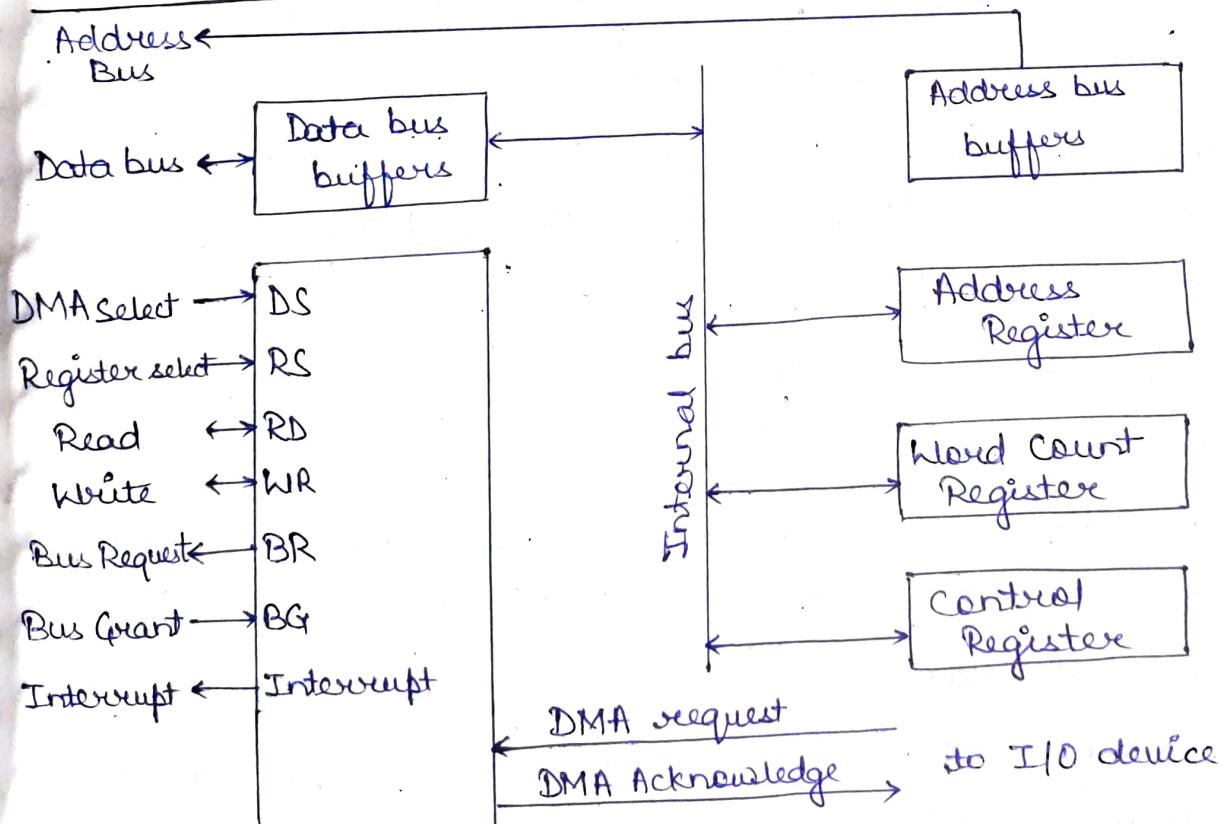


CPU bus signals for DMA transfer.

- The bus request (BR) input is used by the DMA controller to request the CPU for allocation of buses. When this input is active, the CPU terminates the execution of the current instruction and places the address bus, the data bus and the read and write lines into a high-impedance state.
- The CPU activates the bus grant (BG) output to inform the external DMA that the buses are in high-impedance state. The DMA that originated the bus request can now take control of the buses to conduct memory transfers without processor intervention.

- When the DMA terminates the transfer, it disables the bus request line. The CPU disables the bus grant, takes control of the buses and reverts to its normal operation.
- In DMA, burst transfer, a block sequence consisting of a number of memory words is transferred in a continuous burst while the DMA controller is master of the memory buses. This mode of transfer is needed for fast devices such as magnetic disks, where data transmission cannot be stopped or slowed down until an entire block is transferred.
- Cycle stealing allows the DMA controller to transfer one data word at a time, after which it must return control of the buses to the CPU. The CPU merely delays its operation for one memory cycle to allow the direct memory I/O transfer to "steal" one memory cycle.

DMA Controller :-



Block diagram of DMA controller

- The DMA controller needs the usual circuits of an interface to communicate with the CPU and I/O device. In addition, it needs an address register, a word count register and a set of address lines.
- The address register and address lines are used for direct communication with the memory. The word count register specifies the number of words that must be transferred.
- The address register contains an address to specify the desired location in memory. The address register is incremented after each word that is transferred to memory.
- The word count register holds the number of words to be transferred. This register is decremented by one after each word transfer and internally tested for zero.
- The control register specifies the mode of the transfer.
- All registers in the DMA appear to the CPU as I/O interface registers. Thus the CPU can read from or write into the DMA registers under program control via the data bus.

DMA Transfer :-

- The CPU communicates with the DMA through the address and data buses as with any interface unit. The DMA has its own address, which activates DS and RS lines. The CPU initializes the DMA through the data bus. Once DMA receives the start control command, it can start the transfer between the peripheral device and the memory.