Basic Concepts of Interfacing

Definition of Interface

- an **interface** is a medium through which two separate components of a computer system exchange information or interact with each other.
- The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these.
- The **interface in a microprocessor** is an integrated circuit that performs the basic functions of the central processing unit. It enables a user to communicate with a computer.

What is the need of interfacing the microprocessor system?

- if you are not using interfacing, then it will be working with less options.
- On interfacing, we can add many features to it like
  - DMA which will give mode types of data transfer,
  - PIC(peripheral *interface* controller) which will provide more number of interrupt handling capacity,
  - PIT(parallel *interface*) for event driven task and many more.
- Interfacing also allow it to connect to other microprocessor and make its computation more easy and fast.

Continued

- **What is the need for peripherals interfacing with a microprocessor?**
  - Microprocessor based system design involves interfacing of the processor with one or more peripheral devices for the purpose of communication with various input and output devices connected to it.
- **Can we get any output directly from a microprocessor without interfacing with peripherals? or Can’t we connect / interface directly the I/O devices to processor ?**
  - No!
  - You can’t get output directly from a microprocessor without interfacing with

Why?

- I/O devices are most of case usually electrical/ mechanical/ electronic devices where processor is an electronic device. Also the data transfer rates of I/O are often slower than the processor and memory. So it is significant that the speed and electrical characteristics of I/O are different from CPU.
- There are a variety of peripherals that exist and may need to be connected to the same system bus. But it may be difficult to incorporate all the peripheral device logic into CPU. This reduces flexibility and creates hindrance in new developments.
- Peripheral often use different data formats and word lengths that used by the CPU Incorporation of I/O module helps to overcome these problems.

Reasons Lead to Use I/O Module

- First, An I/O module is a mediator between the processor and an I/O device/devices.
- Second, It controls the data exchange between the external devices and main memory; or external devices and CPU registers.
- Third, An I/O module provide an interface internal to the computer which connects it to CPU and main memory and an interface external to the computer connecting it to external device or peripheral.
- Fourth, The I/O module should not only communicate the information from CPU to I/O device, but it should also coordinate these two.
- Fifth, In addition since there are speed differences between CPU and I/O devices, the I/O module should have facilities like buffer (storage area) and error detection mechanism.

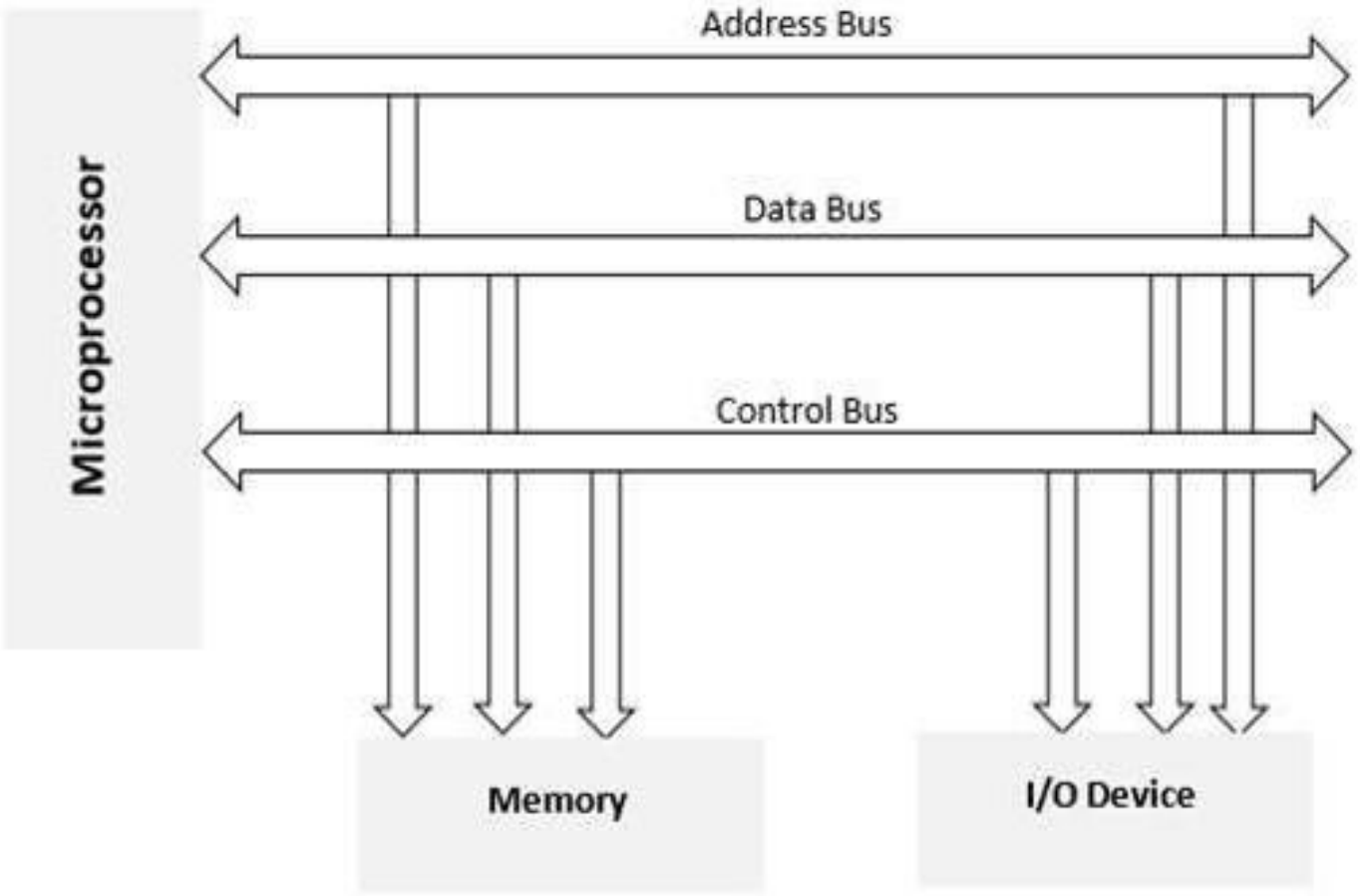
Memory Interfacing

- When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory. For this, both the memory and the microprocessor requires some signals to read from and write to registers.
- The interfacing process includes some key factors to match with the memory requirements and microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.

IO Interfacing

- There are various communication devices like the keyboard, mouse, printer, etc. So, we need to interface the keyboard and other devices with the microprocessor by using latches and buffers. This type of interfacing is known as I/O interfacing.

Block Diagram of Memory and I/O Interfacing



Comparison of Memory Mapped I/O and Peripheral I/O

Characteristics	Memory Mapped I/O	Peripheral I/O
Device Address	16 bit	8 bit
Control signals	MEMR/MEMW	IOR/IOW
Instructions	STA,LDA,STAX,LDAX, MOV M,R ADDM etc.	IN, OUT
Data transfer	Between any register and IO	Only between I/O and the accumulator

Cont.

Characteristics	Memory Mapped I/O	Peripheral I/O
Maximum No. of I/Os possible	64K is shared between I/Os and system memory	256 input devices and 256 output devices
Execution speed	13 T-states(STA,LDA) 7 T-states (MOV M,R)	10 T- states
Hardware Requirements	More hardware is needed to decode 16 bit address	Less Hardware is required to decode 8 bit address
Other features	Arithmetic and logical operations can be directly performed with I/O data	Not Available

Memory Mapped IO and IO Mapped IO in 8086

Memory Mapped IO	IO Mapped IO
❖ Here IO devices treated as Memory	❖ Here IO devices treated as IO.
❖ 20 bits addressing ($A_0 - A_{19}$)	❖ 8 or 16 bits addressing ($A_0 - A_{15}$)
❖ It can address = 2^{20} = 1MB Address	❖ It can address = 2^{16} = 64K Address
❖ Number of devices can be = 2^{20}	❖ Number of devices can be = 2^{16}
❖ Decoding is more complex as total 20 Address lines are used for full decoding.	❖ Decoding is Less complex as total 16 Address lines are used for full decoding.
❖ Decoding is Expensive.	❖ Decoding is cheaper.
❖ Here we use $\overline{MEMR}$ and $\overline{MEMW}$ control signals.	❖ Here we use $\overline{IOR}$ and $\overline{IOW}$ control signals.
❖ IO can be accessed by any memory instructions.	❖ IO can be accessed ONLY by IN and OUT Instructions.
❖ Data transfer happens between any registers and IO.	❖ Data transfer only between AX [AH & AL] and IO.
❖ Works slower due to more gates and circuits.	❖ Works faster due to less delay.

Interfacing with External Memory

- An **external memory interface** is a bus protocol for communication from an integrated circuit, such as a microprocessor, to an external memory device located on a circuit board.
- The memory is referred to as *external* because it is not contained within the internal circuitry of the integrated circuit and thus is externally located on the circuit board.
- The external memory interface enables the processor to interface with third level caches, peripherals, and external memory.
- Some common external memory interfaces include:
  - DDR
  - DDR2
  - GDDR

Data Transfer Schemes

Need for Data Transfer Scheme

- A wide variety of IO devices having wide range of speed and other different characteristics are available
  .
- A slow responding IO device cannot transfer data when microprocessor issues instruction for it as it takes some time to get ready.
- Transfers rates of peripherals is usually slower than the transfer rates of CPU.
- Operating modes of peripheral are different from each other and each must be controlled so as not to disturb the operation of each other peripherals connected to CPU

Types of Data Transfer Scheme

- Different types of data transfer techniques are available which can be broadly divided into two categories:-
  1. MICROPROCESSOR CONTROLLED :-
     HERE data
     transfer is controlled by microprocessor. Microprocessor is primarily responsible for data transfer whether from I/O to the CPU or to the memory or vice versa .
  2. PERIPHERAL/DEVICE CONTROLLED:-
     Here data
     transfer is controlled by I/O device . Data is transferred in between I/O device and memory without the intervention of CPU such a transfer increases rate of transfer and makes the system more efficient

Microprocessor Controlled DTS

- Microprocessor based scheme is further divided into two parts:-
  - PROGRAMMED DATA TRANSFER SCHEME
  - INTERRUPT CONTROL DATA TRANSFER SCHEME

Programmed Data Transfer Scheme

- Programmed data transfer scheme is controlled by the CPU . Data are transferred from an IO device to the CPU or to the memory through CPU or vice versa under the control of programs which are stored in memory. These programs are executed by the CPU when an I/O device is ready to transfer data.
- The program data transfer schemes are employed when small amount of data are to be transferred.

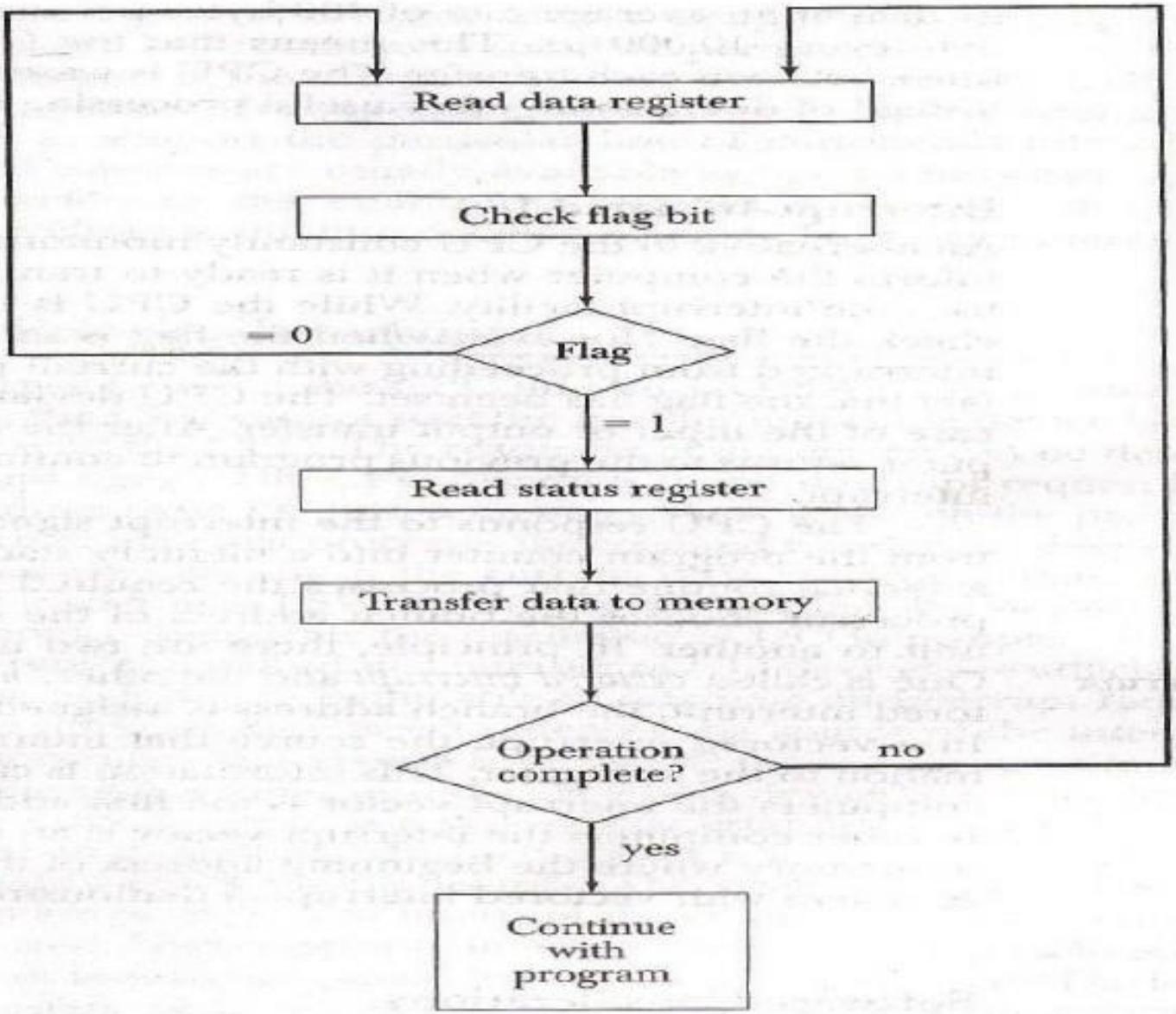


Figure 11-11 Flowchart for CPU program to input data.

- Here also **synchronous and asynchronous mode of transfer** is used.
- **Synchronous Data Transfer :-**
  - Synchronous means ‘at the same time’. The device which sends data and the device which received data are synchronized with the same clock. When the CPU and I/O devices match in speed, Synchronous Data Transfer technique is employed.
  - The data transfer with IO devices is performed by executing IN and OUT instruction. The IN instruction is used to read data from an input device or input port.
    The OUT instruction is used to sends data from CPU to the output device or output port. As the CPU and the IO devices match in speed, the I/O device is ready to transfer data when IN or OUT instruction is executed. The status of the I/O device, whether it is ready or not, is not examined before the data is transferred.

- **Asynchronous mode of transfer :-**
  - Asynchronous means ‘at irregular intervals’. In this method data transfer is not based on predetermined timing pattern. This technique of data transfer is used when the speed of an I/O device does not match the speed of the microprocessor.
  - In this technique the status of the I/O device i.e. whether the device is ready or not, is checked by the microprocessor before the data are transferred. The microprocessor initiates the I/O device to get ready and then continuously checks the status of I/O device till the I/O device becomes ready to transfer data.
    When I/O device becomes ready, the microprocessor executes instruction to transfer data.

- This mode of data transfer is also called **handshaking mode** of data transfer because some signals are exchanged between microprocessor and I/O devices before the actual data transfer takes place. Such signals are called **handshake signals**.

Drawback of Programmed Data Transfer Scheme

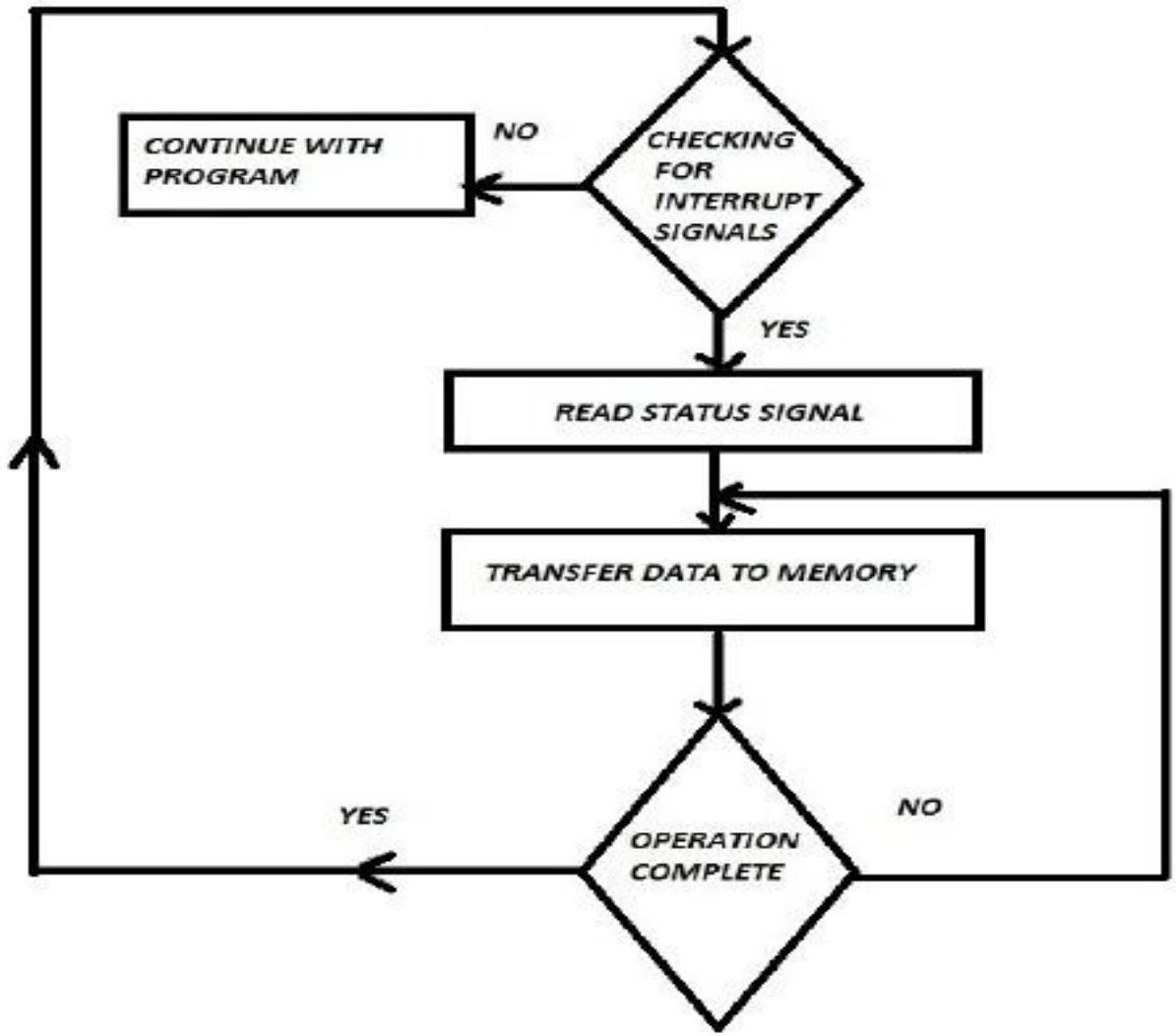
- The microprocessor is too busy.
- The CPU is wasting time while checking the flag instead of doing some useful work.

Interrupt Driven Data Transfer

- The problem with programmed I/O is that CPU has to wait along time for the I/O device to be ready for reception or transmission of data
  .The CPU while waiting, must repeatedly interrogate the status of the I/O device . As a result the level of the performance of the entire system is severely degraded.
- An alternative is ***interrupt driven IO data transfer.***

Transfer Operation

- In this scheme when the I/O device becomes ready to transfer data, it sends a high signal to the microprocessor through a special input line called an **interrupt line**. In other words it interrupts the normal processing sequence of the microprocessor.
- On receiving interrupt the microprocessor completes the current instruction, saves the contents of the program counter on stack first and then attends the I/O devices. It take up a subroutine called **ISS (Interrupt Service Subroutine).** It execute ISS to transfer data from or to the I/O device. Different ISS are to be provided for different IO devices. After completing the data transfer the microprocessor returns back to the main program which it was executing before the interrupt occurred.


CONTINUE WITH PROGRAM
NO
CHECKING FOR INTERRUPT SIGNALS
YES
READ STATUS SIGNAL
TRANSFER DATA TO MEMORY
YES
OPERATION COMPLETE
NO

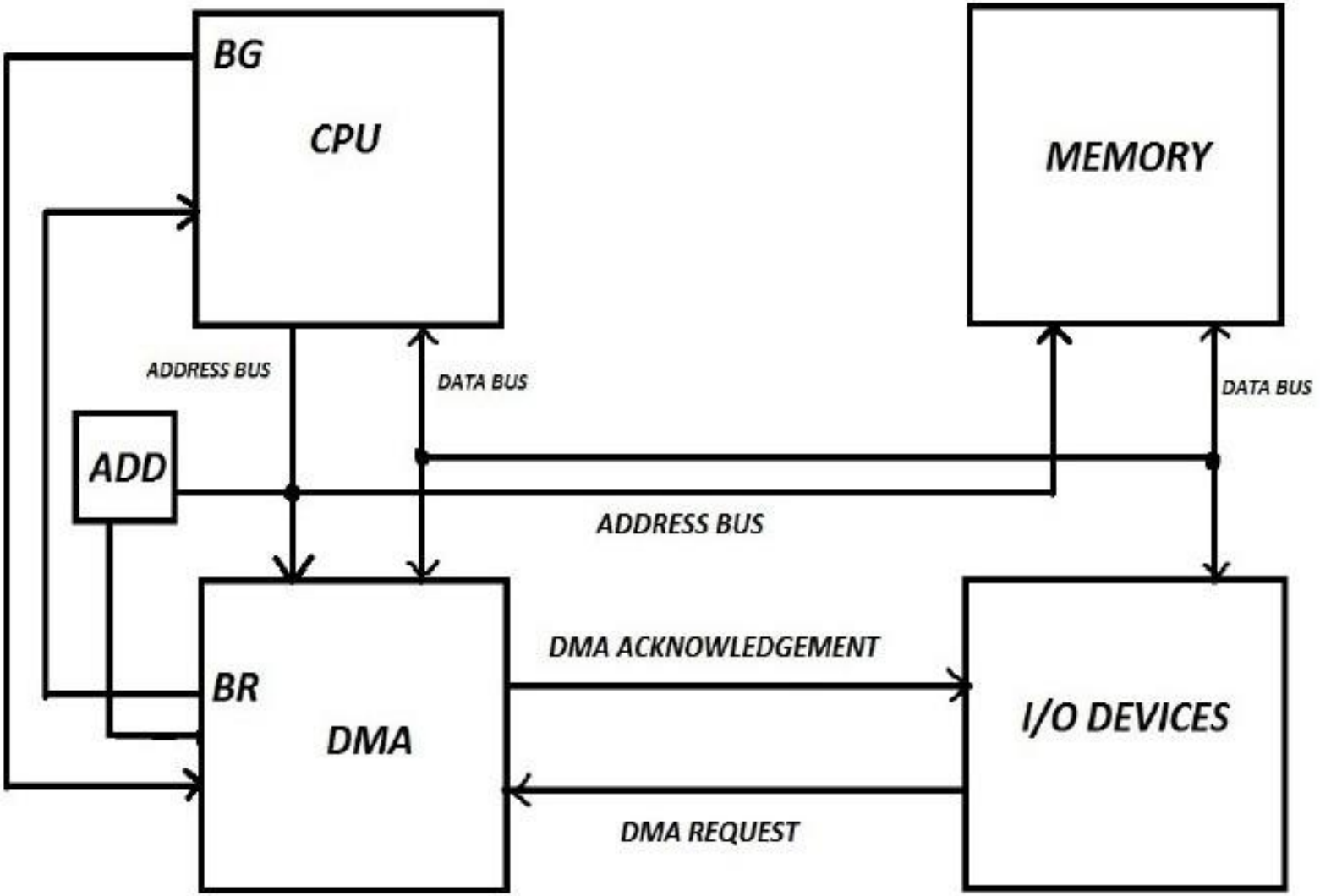
Drawback of Interrupt Driven Data Transfer

- The normal operation of the microprocessor is interrupted.
- It needs to continue monitoring for interrupt signals.

Device Control Data Transfer

- The transfer of data between the mass storage device and a system memory is often limited by the speed of microprocessor. Removing the microprocessor during such a transfer and letting the peripheral manage the transfer to or from memory would improve the speed of transfer and hence will make the system more efficient. This transfer technique is called **DMA Data Transfer**.

- During DMA transfer microprocessor is idle, so it has no longer control on the system buses. A **DMA Controller** takes over the buses and manage the transfer directly between the peripheral and the memory
- It is fastest scheme then Programmed Data Transfer Scheme and the microprocessor regains the control of buses after data transfer


BG
CPU
MEMORY
ADDRESS BUS
DATA BUS
DATA BUS
ADD
ADDRESS BUS
DMA ACKNOWLEDGEMENT
BR
DMA
I/O DEVICES
DMA REQUEST