

## **Memory mapped I/O and Isolated I/O**

The microprocessor obviously is not used as a single unit. It needs to be linked with memory, extra peripherals, or I/O devices. This linking can also be called as ***Interfacing***. An interface acts as a communication channel between the processor and the externally interfaced device.

I/O interfacing is the link between the processor or CPU and the various I/O peripherals such as the keyboard, printer, mouse, etc. Such type of interfacing is referred to as I/O Interfacing.

The interfacing of the I/O devices can be done in two ways:

1. Memory-Mapped I/O Interfacing
2. Standard I/O Mapped I/O Interfacing (also known as isolated I/O interfacing).

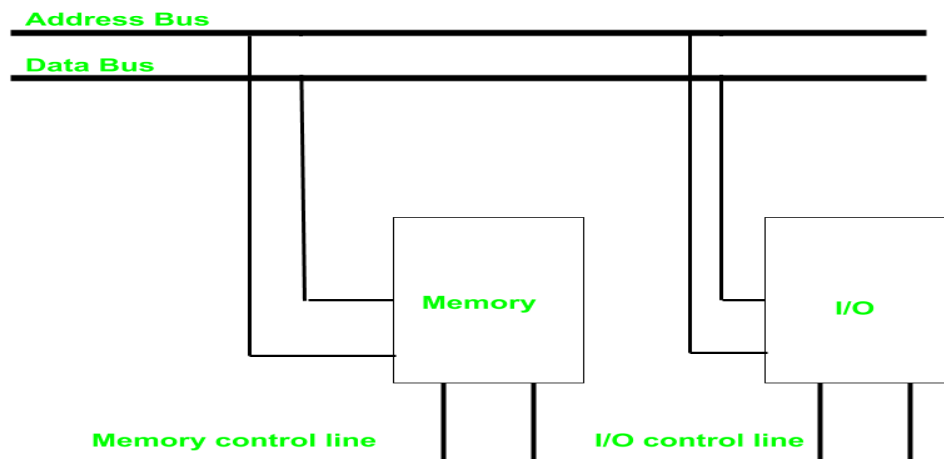
As a CPU needs to communicate with the various memory and input-output devices (I/O) as we know data between the processor and these devices flow with the help of the system bus. There are three ways in which system bus can be allotted to them :

1. Separate set of address, control and data bus to I/O and memory.
2. Have common bus (data and address) for I/O and memory but separate control lines.
3. Have common bus (data, address, and control) for I/O and memory.

In first case it is simple because both have different set of address space and instruction but require more buses.

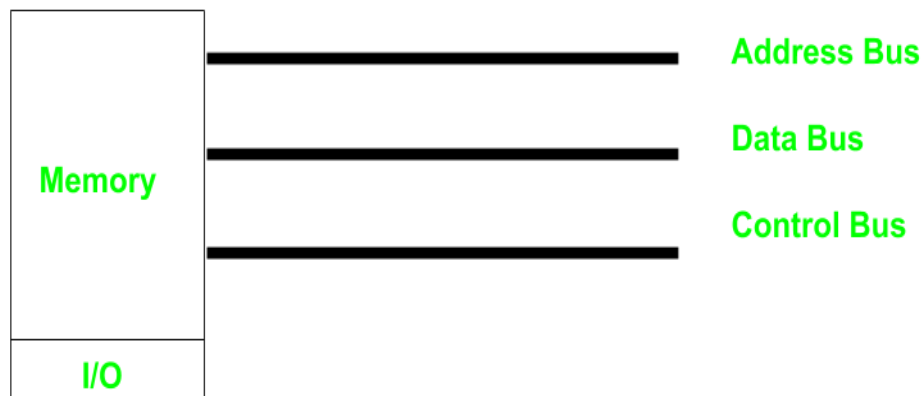
### **Isolated I/O –**

Then we have Isolated I/O in which we Have common bus(data and address) for I/O and memory but separate read and write control lines for I/O. So when CPU decode instruction then if data is for I/O then it places the address on the address line and set I/O read or write control line on due to which data transfer occurs between CPU and I/O. As the address space of memory and I/O is isolated and the name is so. The address for I/O here is called ports. Here we have different read-write instruction for both I/O and memory.



### Memory Mapped I/O –

In this case every bus is common due to which the same set of instructions work for memory and I/O. Hence we manipulate I/O same as memory and both have same address space, due to which addressing capability of memory becomes less because some part is occupied by the I/O.



## Differences between memory mapped I/O and isolated I/O –

Isolated I/O	Memory Mapped I/O
Memory and I/O have separate address space	Both have same address space
All address can be used by the memory	Due to addition of I/O addressable memory become less for memory
Separate instruction control read and write operation in I/O and Memory	Same instructions can control both I/O and Memory
In this I/O address are called ports.	Normal memory address are for both
More efficient due to separate buses	Lesser efficient
Larger in size due to more buses	Smaller in size
It is complex due to separate logic is used to control both.	Simpler logic is used as I/O is also treated as memory only.

## **Data Transfer Techniques**

```
graph TD; A[Data Transfer Techniques] --> B[Parallel Data Transfer]; A --> C[Serial Data Transfer]; B --> B1[1. Programmed Data Transfer]; B --> B2[2. Interrupt Driven Data Transfer]; B --> B3[3. Device or DMA Data Transfer]; B1 --> B1a[(a) Synchronous Data Transfer]; B1 --> B1b[(b) Asynchronous Data Transfer]; B3 --> B3a[(a) Burst or Block Transfer DMA]; B3 --> B3b[(b) Cycle steal or single byte transfer DMA]; B3 --> B3c[(c) Transparent or hidden DMA]; C --> C1[1. Synchronous Data Transfer]; C --> C2[2. Asynchronous Data Transfer];
```

### **Parallel Data Transfer**

1. Programmed Data Transfer
  - (a) Synchronous Data Transfer
  - (b) Asynchronous Data Transfer
2. Interrupt Driven Data Transfer
3. Device or DMA Data Transfer
  - (a) Burst or Block Transfer DMA
  - (b) Cycle steal or single byte transfer DMA
  - (c) Transparent or hidden DMA

### **Serial Data Transfer**

1. Synchronous Data Transfer
2. Asynchronous Data Transfer