I/O FUNDAMENTALS

Introduction

- The I/O subsystem provides an efficient mode of communication between CPU and outside environment
- Devices that are under the direct control of computer are said to be connected on-line
- Input or output devices attached to the computer are also referred as peripherals.
- I/O interface or I/O Module provides a method for transferring information between internal storage and external i/o devices.

IO interface/IO module

Resolves the *differences* between the computer and peripheral devices

-Design

Peripherals	- Electromechanical Devices
CPU or Memory	- Electronic Device

- Data Transfer Rate

Peripherals	- Usually slower
CPU or Memory	- Usually faster than peripherals

Some kinds of Synchronization mechanism may be needed

- Unit of Information

Peripherals	- Byte
CPU or Memory	- Word

- Operating Modes

Peripherals	- Asynchronous
CPU or Memory	- Synchronous

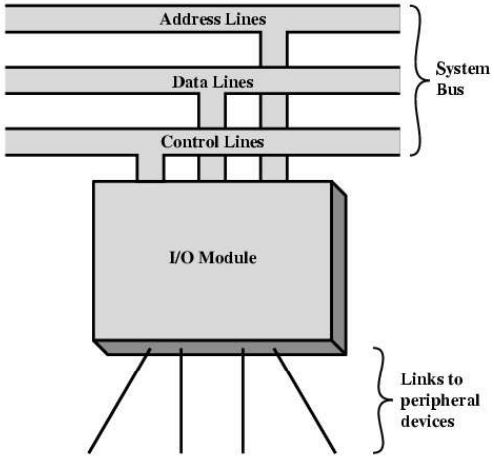
Input/Output Problems

- Wide variety of peripherals
  - Delivering different amounts of data
  - At different speeds
  - In different formats
- All slower than CPU and RAM
- Need I/O modules

Input/Output Module

- Interface to CPU and Memory
- Interface to one or more peripherals

Generic Model of I/O Module



I/O Module Function

- Control & Timing
- CPU Communication
- Device Communication:- BUSY, READY signal
- Data Buffering
- Error Detection

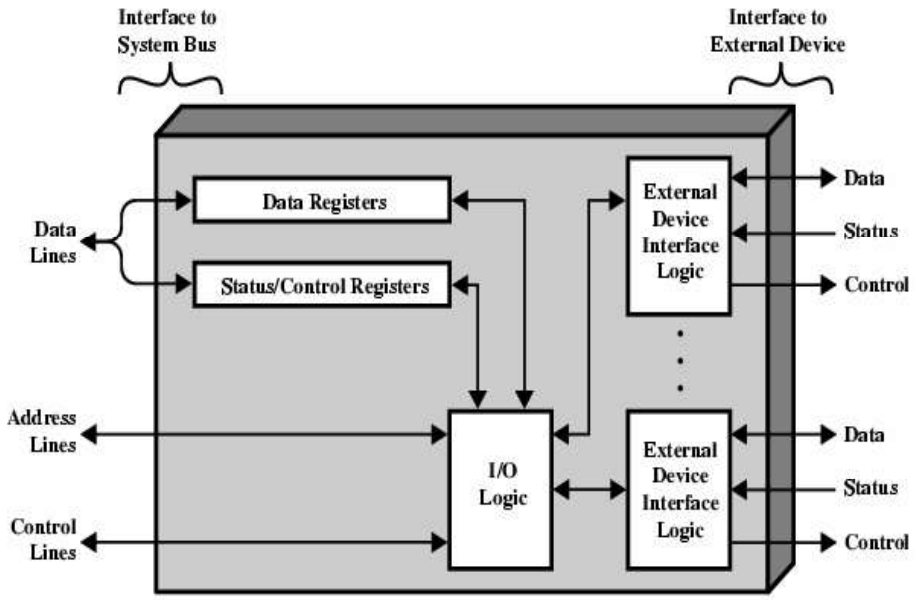
I/O Steps

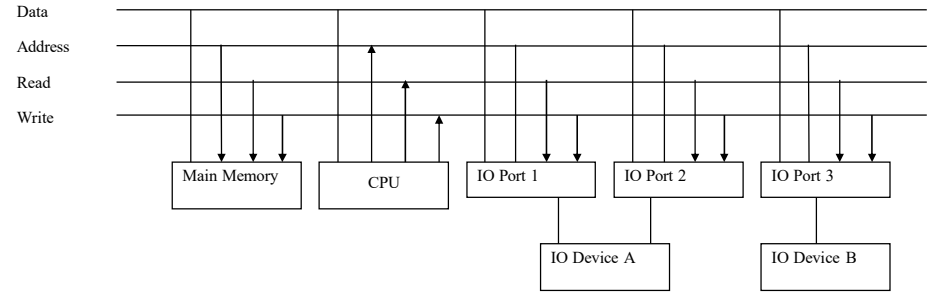
- Steps needed to transfer data to or from external device to

CPU:

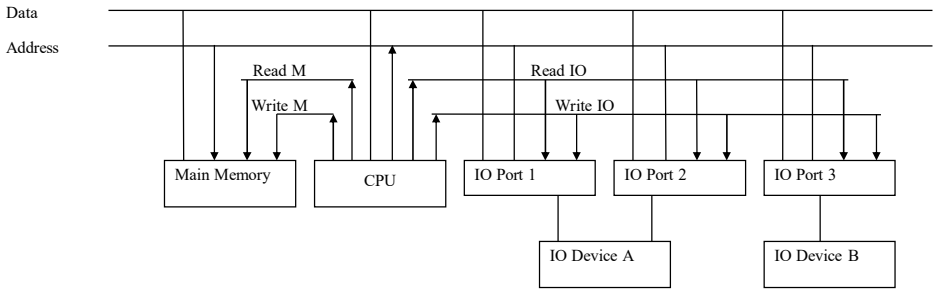
1. CPU checks I/O module device status
2. I/O module returns status
3. If ready, CPU requests data transfer
4. I/O module gets data from device
5. I/O module transfers data to CPU

I/O Module Diagram





Memory-Mapped I/O



I/O Mapped I/O

I/O Bus and Memory Bus

- *MEMORY BUS* **is for information transfers between CPU and the Main Memory.**
- *I/O BUS* **is for information transfers between CPU and I/O devices through their I/O interface.**
- **Many computers use a common single bus system for both memory and I/O interface units**

  - Use one common bus but separate control lines for each function

  - Use one common bus with common control lines for both functions

Isolated 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
- When CPU fetches and decodes the opcode of an I/O instruction, it places the address into the common address lines. Also enables read/write control lines => the address in the address lines is for interface register and not for a memory word.

Memory Mapped I/O

- A single set of read/write control lines
- Memory and I/O addresses share the common address space .
  - Reduces memory address range available.
- No specific input or output instruction
- The same memory reference instructions can be used for I/O transfers
- When the bus sees certain addresses, it knows they are not memory addresses, but are addresses for accessing I/O devices.

-

Asynchronous Data Transfer

Two Asynchronous Data Transfer Methods:

Strobe pulse :-

- A strobe pulse is supplied by one unit to indicate the other unit when the transfer has to occur

Handshaking:-

- A control signal is accompanied with each data being transmitted to indicate the presence of data
- The receiving unit responds with another control signal to acknowledge receipt of the data

Asynchronous Data Transfer

Synchronous - All devices derive the timing information from common clock line.

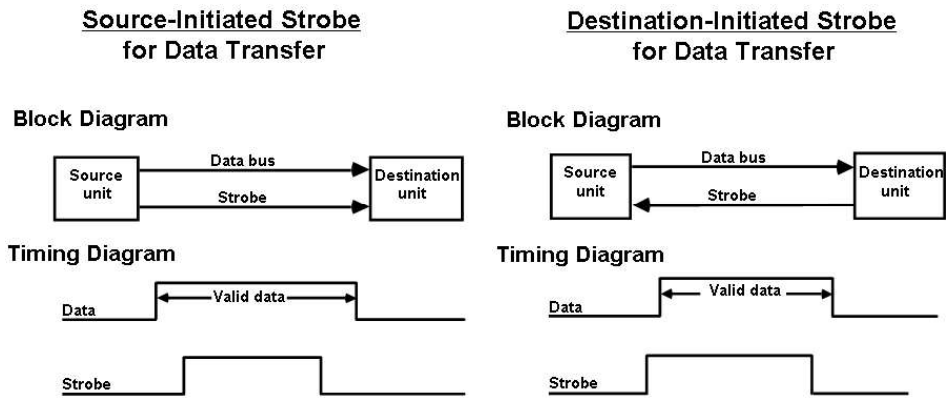
Asynchronous - No common clock

Asynchronous Data Transfer:-

Asynchronous data transfer between two independent units requires that *control signals* be transmitted between the communicating units *to indicate the time at which data is being transmitted*

Strobe Control

- **The strobe may be activated by either the source or the destination unit**
- **Data bus carries the binary information from source to destination**



Source initiated strobe

- It could be a memory write control signal from the CPU to a memory unit.
- Source is the CPU, places a word on the data bus and informs the memory unit which is destination, that this is a write operation.
- Disadvantage – no way of knowing whether the destination unit has actually received data.

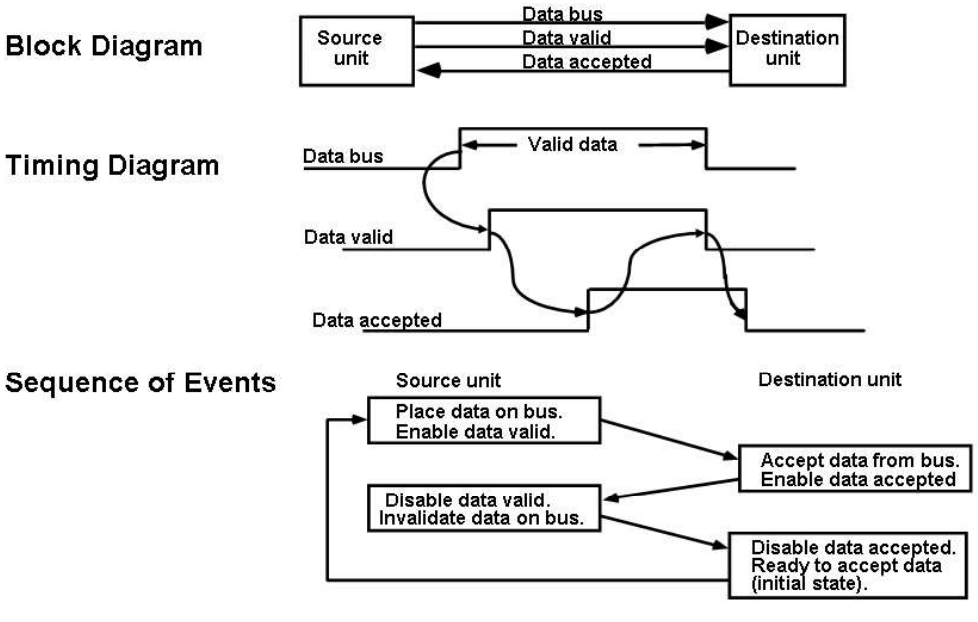
Destination initiated strobe

- It could be a memory read control signal.
- CPU, the destination initiates the read operation to inform the memory which is the source to place a selected word into the data bus.
- Disadvantage - no way of knowing whether the source has actually placed the data on the bus

Hand Shaking

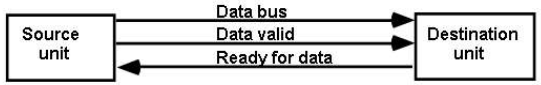
- The handshake method introduces a second control signal to provide a *reply* to the unit that initiates the transfer.
- 2 types
  - Source initiated transfer
  - Destination initiated transfer
- Provides high degree of flexibility and reliability
- Incompletion of data transfer can be detected by means of a timeout mechanism
- The timeout signal can be used to interrupt the processor and hence execute a service routine that takes appropriate error recovery action.

SOURCE-INITIATED TRANSFER USING HANDSHAKE

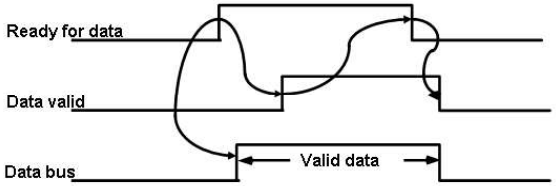


DESTINATION-INITIATED TRANSFER USING HANDSHAKE

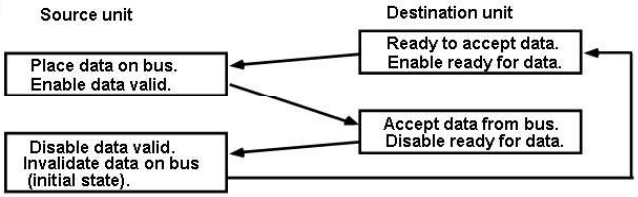
Block Diagram



Timing Diagram



Sequence of Events



- Strobe pulse: Data transfer between CPU and Interface
- Handshaking signals:- Data transfer between IO interface and peripheral device.