I/O Management and Disk Scheduling

Agenda

- **I/O Devices**
- **Organization of the I/O Function**
- **Operating System Design Issues**
- **I/O Buffering**
- **Disk Scheduling**
- **RAID**
- **Disk Cache**
- **Linux I/O**

Categories of I/O Devices

External devices that engage in I/O with computer systems can be roughly grouped into three categories:

- Human readable
  - Used to communicate with the user
  - Printers
  - Video display terminals
    - Display
    - Keyboard
    - Mouse

- Machine readable
  - Used to communicate with electronic equipment
  - Disk and tape drives
  - Sensors
  - Controllers
  - Actuators

- Communication
  - Used to communicate with remote devices
  - Digital line drivers
  - Modems

Differences in I/O Devices

- Data rate
  - May be differences of several orders of magnitude between the data transfer rates

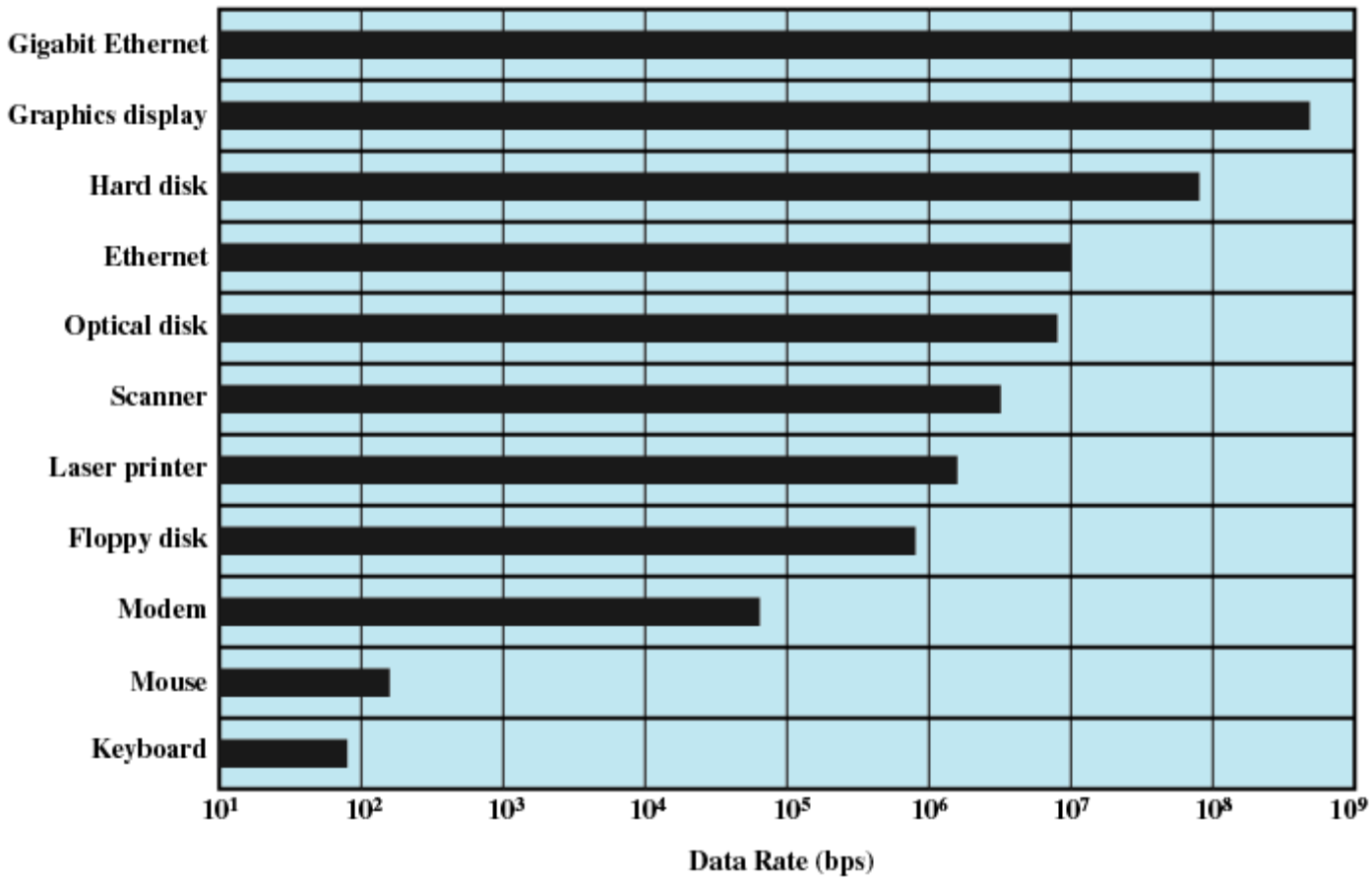


Figure 11.1 Typical I/O Device Data Rates

Differences in I/O Devices

- Application
  - Disk used to store files requires file management software
  - Disk used to store virtual memory pages needs special hardware and software to support it
  - Terminal used by system administrator may have a higher priority

Differences in I/O Devices

- Complexity of control
- Unit of transfer
  - Data may be transferred as a stream of bytes for a terminal or in larger blocks for a disk
- Data representation
  - Encoding schemes
- Error conditions
  - Devices respond to errors differently

Performing I/O

- Programmed I/O
  - Process is busy-waiting for the operation to complete
- Interrupt-driven I/O
  - I/O command is issued
  - Processor continues executing instructions
  - I/O module sends an interrupt when done

Performing I/O

- Direct Memory Access (DMA)
  - DMA module controls exchange of data between main memory and the I/O device
  - Processor interrupted only after entire block has been transferred

Relationship Among Techniques

Table 11.1 I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Evolution of the I/O Function

- Processor directly controls a peripheral device
- Controller or I/O module is added
  - Processor uses programmed I/O without interrupts
  - Processor does not need to handle details of external devices

Evolution of the I/O Function

- Controller or I/O module with interrupts
  - Processor does not spend time waiting for an I/O operation to be performed
- Direct Memory Access
  - Blocks of data are moved into memory without involving the processor
  - Processor involved at beginning and end only

Evolution of the I/O Function

- I/O module is a separate processor
- I/O processor
  - I/O module has its own local memory
  - Its a computer in its own right

Direct Memory Access

- Processor delegates I/O operation to the DMA module
- DMA module transfers data directly to or form memory
- When complete DMA module sends an interrupt signal to the processor

DMA

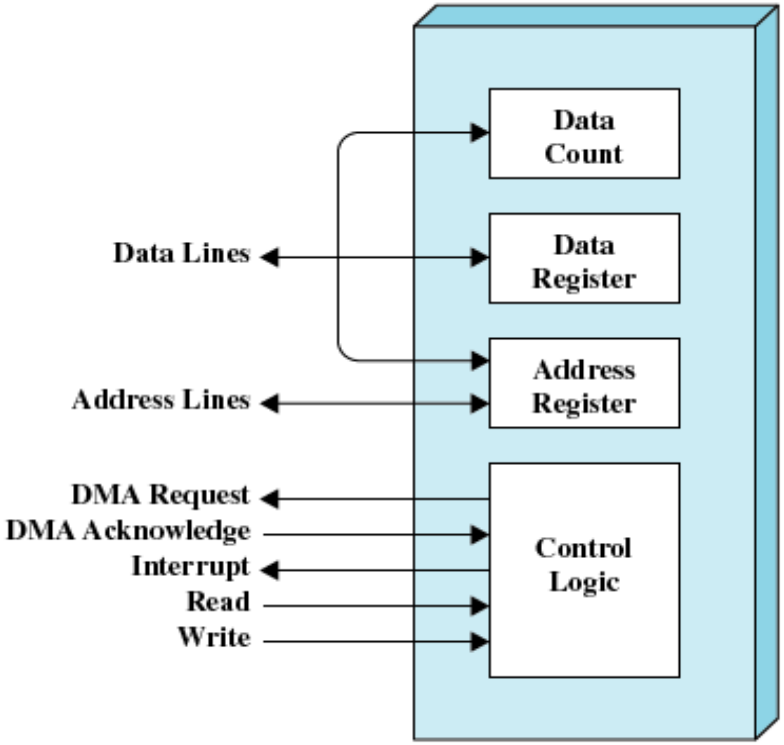


Figure 11.2 Typical DMA Block Diagram

•The DMA technique works as follows.

•When the processor wishes to read or write a block of data, it issues a command to the DMA module by sending to the DMA module the following information:

• Whether a read or write is requested, using the read or write control line between the processor and the DMA module

- The address of the I/O device involved, communicated on the data lines.

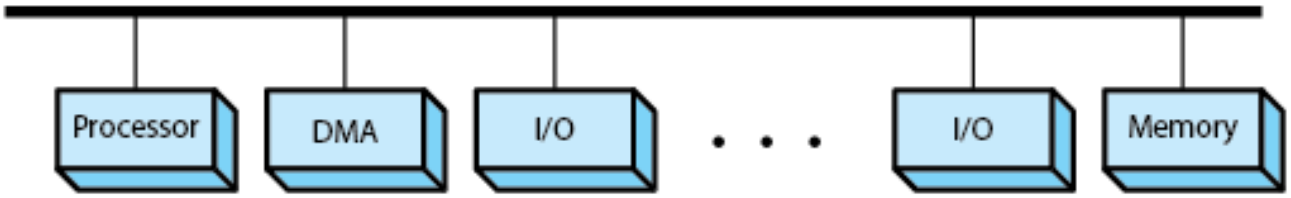
- The starting location in memory to read from or write to, communicated on the data lines and stored by the DMA module in its address register.

- The number of words to be read or written, again communicated via the data lines and stored in the data count register

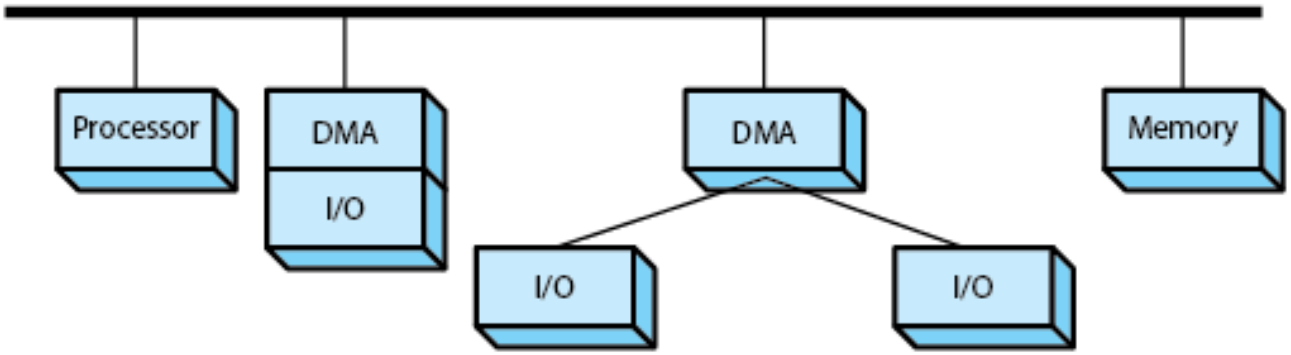
- The processor then continues with other work.
- It has delegated this I/O operation to the DMA module.
- The DMA module transfers the entire block of data, one word at a time, directly to or from memory, without going through the processor.
- When the transfer is complete, the DMA module sends an interrupt signal to the processor.
- Thus, the processor is involved only at the beginning and end of the transfer.

DMA Configurations

DMA mechanism can be configured in a variety of ways.

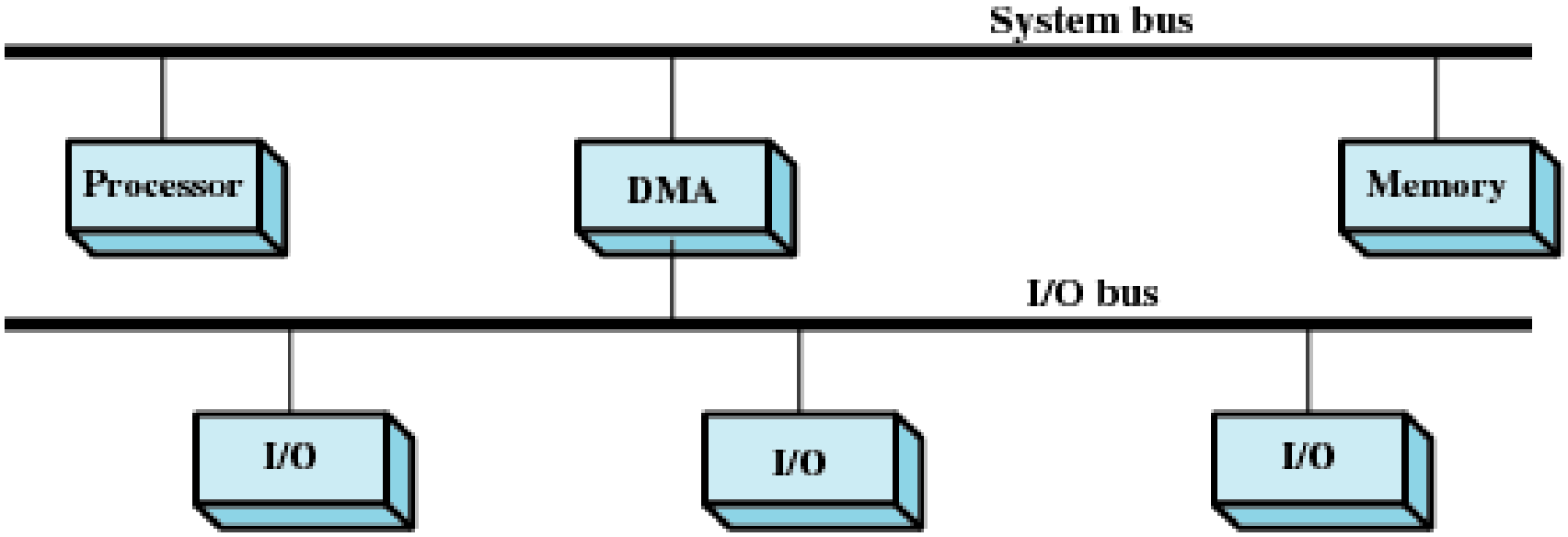


(a) Single-bus, detached DMA



(b) Single-bus, Integrated DMA-I/O

DMA Configurations



(c) I/O bus

Figure 11.3 Alternative DMA Configurations

Operating System Design Issues

- Efficiency
  - Most I/O devices extremely slow compared to main memory
  - Use of multiprogramming allows for some processes to be waiting on I/O while another process executes
  - I/O cannot keep up with processor speed
  - Swapping is used to bring in additional Ready processes which is an I/O operation

Operating System Design Issues

- Generality
  - Desirable to handle all I/O devices in a uniform manner
  - Hide most of the details of device I/O in lower-level routines so that processes and upper levels see devices in general terms such as read, write, open, close, lock, unlock

Logical Structure of the I/O Function

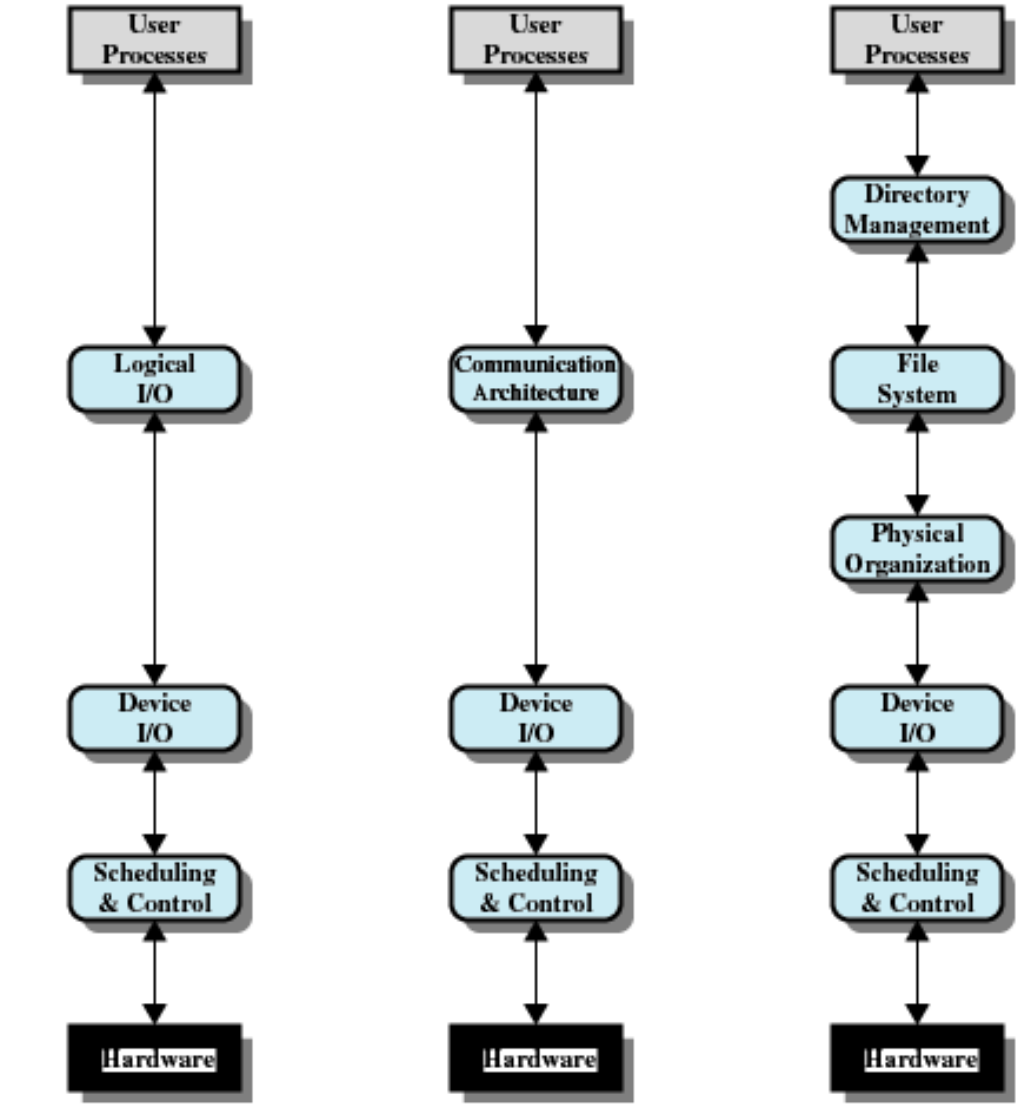


Figure 11.4 A Model of I/O Organization

The details of the organization will depend on the type of device and the application. The three most important logical structures are presented in the pervious figure.
Of course, a particular operating system may not conform exactly to these structures.

However, the general principles are valid, and most operating systems approach I/O in approximately this way.

Let us consider the simplest case first, that of a local peripheral device that communicates in a simple fashion, such as a stream of bytes or records (Figure 11.4a).

The following layers are involved:

- **Logical I/O:**

The logical I/O module deals with the device as a logical resource and is not concerned with the details of actually controlling the device.

The logical I/O module is concerned with managing general I/O functions on behalf of user processes, allowing them to deal with the device in terms of a device identifier and simple commands such as open, close, read, and write.

Device I/O:

The requested operations and data (buffered characters, records, etc.) are converted into appropriate sequences of I/O instructions, channel commands, and controller orders. Buffering techniques may be used to improve utilization.

Scheduling and control:

The actual queueing and scheduling of I/O operations occurs at this layer, as well as the control of the operations. Thus, interrupts are handled at this layer and I/O status is collected and reported. This is the layer of software that actually interacts with the I/O module and hence the device hardware.

For a communications device, the I/O structure (Figure 11.4b) looks much the same as that just described.
The principal difference is that the logical I/O module is replaced by a communications architecture, which may itself consist of a number of layers.
An example is TCP/IP.

Figure 11.4c shows a representative structure for managing I/O on a secondary storage device that supports a file system.
The three layers not previously discussed are as follows:

•**Directory management:** At this layer, symbolic file names are converted to identifiers that either reference the file directly or indirectly through a file descriptor or index table. This layer is also concerned with user operations that affect the directory of files, such as add, delete, and reorganize.

•**File system:** This layer deals with the logical structure of files and with the operations that can be specified by users, such as open, close, read, and write. Access rights are also managed at this layer.

•**Physical organization:** Just as virtual memory addresses must be converted into physical main memory addresses, taking into account the segmentation and paging structure, logical references to files and records must be converted to physical secondary storage addresses, taking into account the physical track and sector structure of the secondary storage device.

Allocation of secondary storage pace and main storage buffers is generally treated at this layer as well.

I/O Buffering

- Reasons for buffering
  - Processes must wait for I/O to complete before proceeding
  - Certain pages must remain in main memory during I/O

I/O Buffering

- Block-oriented
  - Information is stored in fixed sized blocks
  - Transfers are made a block at a time
  - Used for disks and tapes
- Stream-oriented
  - Transfer information as a stream of bytes
  - Used for terminals, printers, communication ports, mouse and other pointing devices, and most other devices that are not secondary storage

Single Buffer

- Operating system assigns a buffer in main memory for an I/O request
- Block-oriented
  - Input transfers made to buffer
  - Block moved to user space when needed
  - Another block is moved into the buffer
    - Read ahead

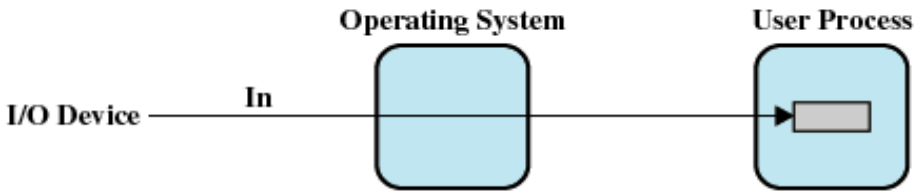
Single Buffer

- Block-oriented
  - User process can process one block of data while next block is read in
  - Swapping can occur since input is taking place in system memory, not user memory
  - Operating system keeps track of assignment of system buffers to user processes

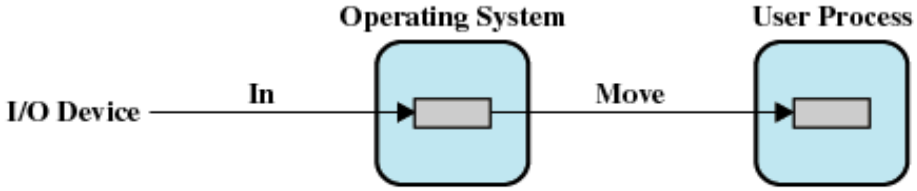
Single Buffer

- Stream-oriented
  - Used a line at time
  - User input from a terminal is one line at a time with carriage return signaling the end of the line
  - Output to the terminal is one line at a time

I/O Buffering



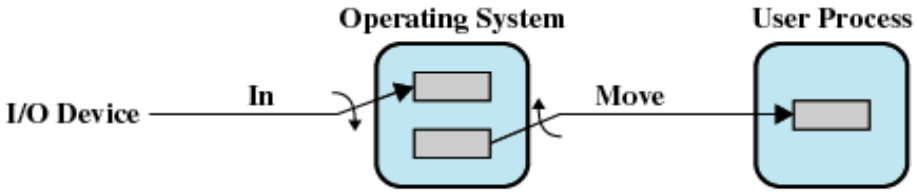
(a) No buffering



(b) Single buffering

Double Buffer

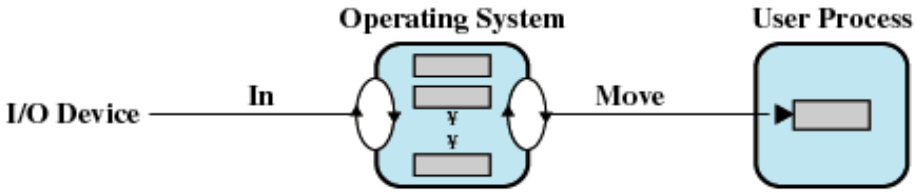
- Use two system buffers instead of one
- A process can transfer data to or from one buffer while the operating system empties or fills the other buffer



(c) Double buffering

Circular Buffer

- More than two buffers are used
- Each individual buffer is one unit in a circular buffer
- Used when I/O operation must keep up with process



(d) Circular buffering

Disk Performance Parameters

- To read or write, the disk head must be positioned at the desired track and at the beginning of the desired sector
- Seek time
  - Time it takes to position the head at the desired track
- Rotational delay or rotational latency
  - Time its takes for the beginning of the sector to reach the head

Timing of a Disk I/O Transfer

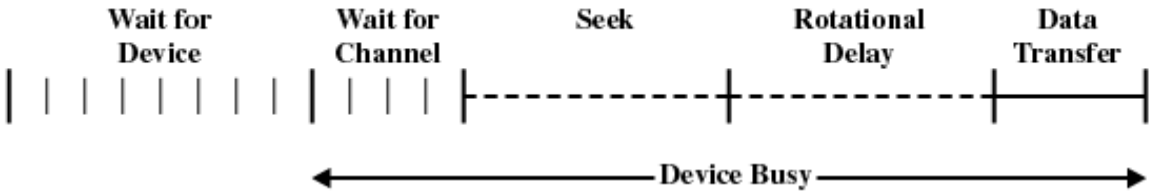


Figure 11.6 Timing of a Disk I/O Transfer

Disk Performance Parameters

- Access time
  - Sum of seek time and rotational delay
  - The time it takes to get in position to read or write
- Data transfer occurs as the sector moves under the head