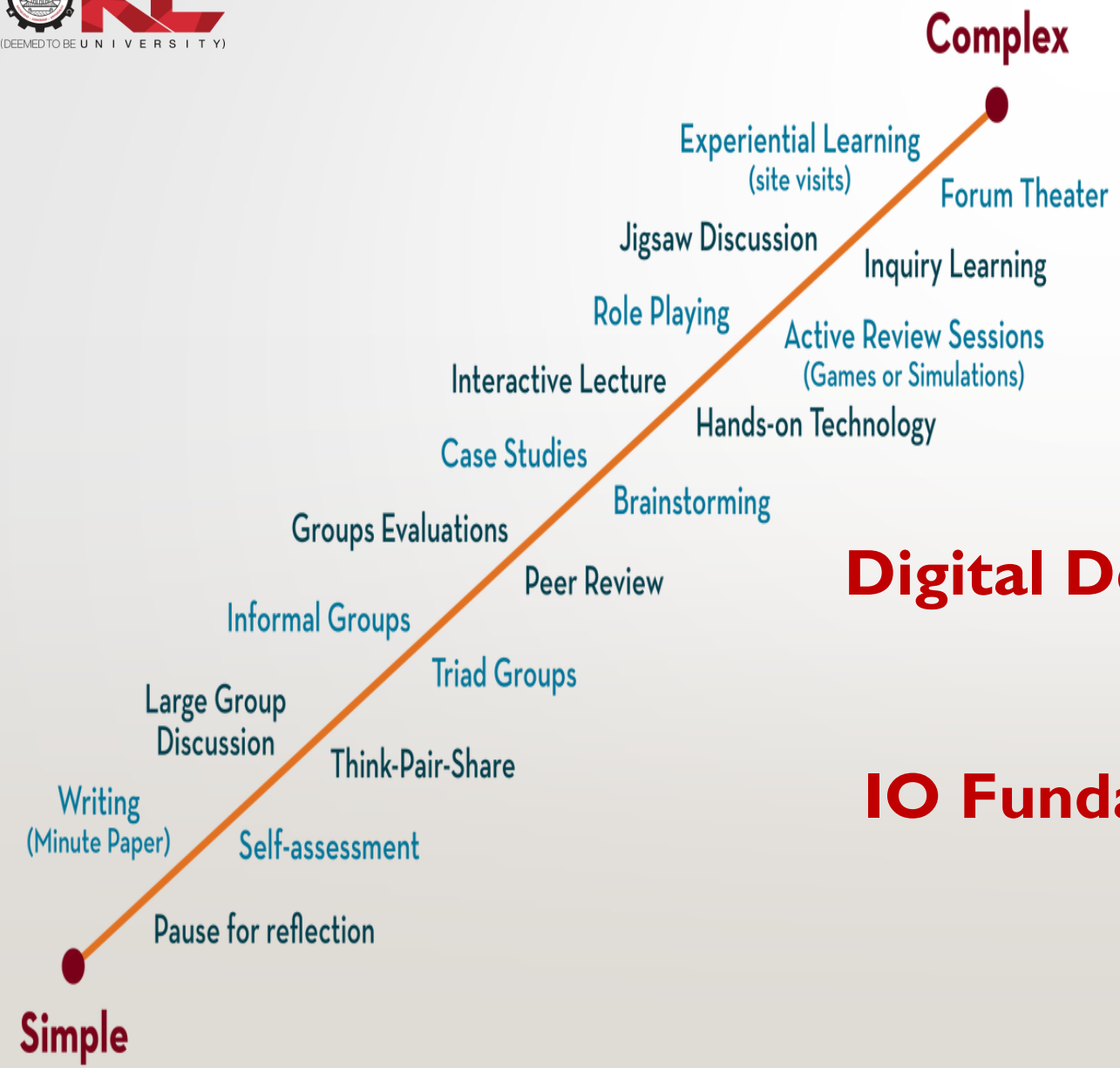


Department of BES-II

Digital Design and Computer Architecture
23EC1202

Topic:

IO Fundamentals: Handshaking, Buffering

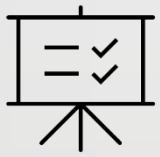
Session No: 38

(DEEMED TO BE UNIVERSITY)

AIM OF THE SESSION

To familiarize students with the basic concept of I/O communication occurring between CPU and peripherals attached to it.

INSTRUCTIONAL OBJECTIVES



This Session is designed to:

1. Demonstrate the fundamentals of Input Output communications
2. Describe the asynchronous data transfer mechanism: Strobing, handshaking
3. List out the advantages and disadvantages of buffering
4. Describe the I/O Buses and Interfaces modules

LEARNING OUTCOMES

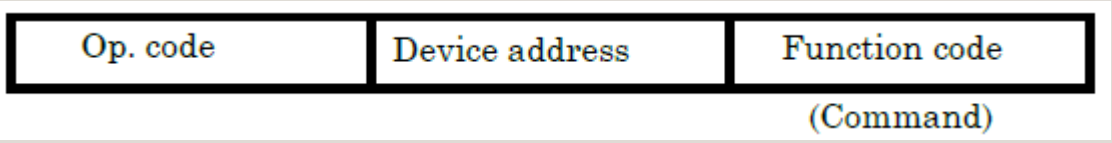
At the end of this session, you should be able to:

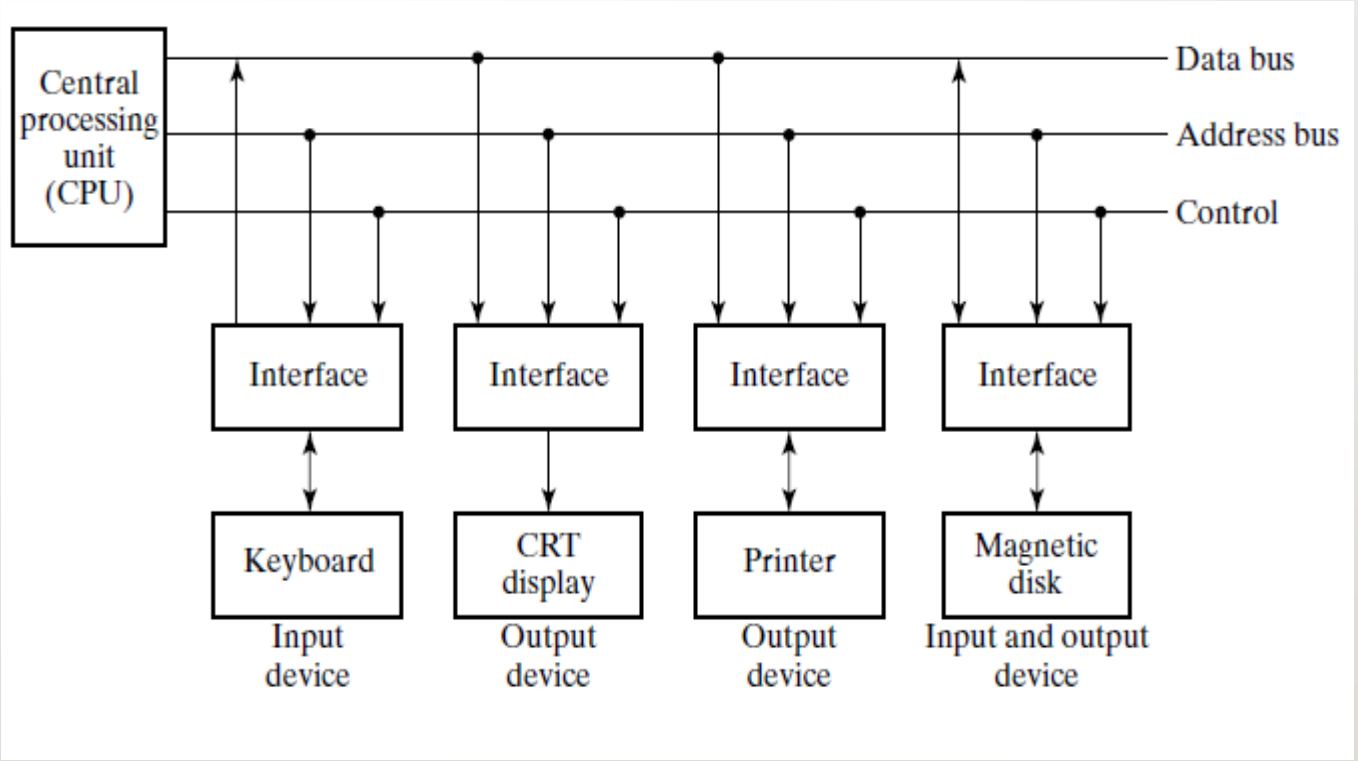
1. Define Strobing, handshaking, and buffering
2. Describe fundamentals of Input Output communications
3. Summarize the working of Strobing, handshaking, and buffering

I/O Bus and Interface Modules

- Each IO peripheral has an interface module associated with it interface with do following:
  - Decodes the device address (device code)
  - Decodes the commands (operation)
  - Provides signals for the peripheral controller
  - Synchronizes the data flow and supervises the transfer rate between peripheral and CPU or Memory

Typical I/O instruction





Data Transfer

Synchronous and Asynchronous Operations

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.*

- One way to achieving this is by means of a STROBE pulse method.
- Other way is HANDSHAKING method.

Note: In general case we consider the transmitting unit as the source and receiving unit as the destination.

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

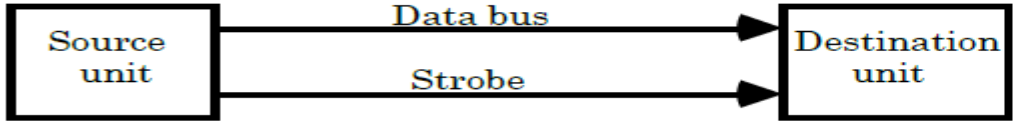
- A control signal is accompanied with each data being transmitted to indicate the presence of data.
- The receiving unit responds with another control signal to acknowledge receipt of the data.

Strobe Control

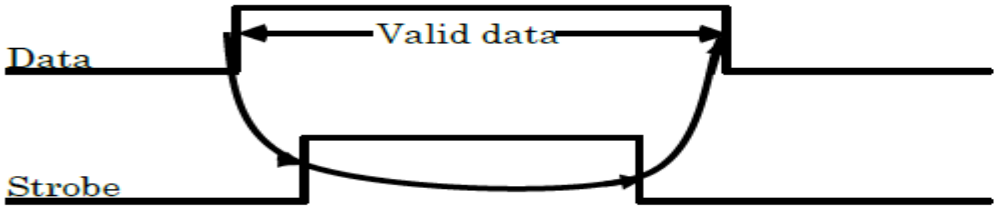
- Employs a single control line (STROBE) and a data bus.
- The strobe may be activated by either the source or the destination unit.

Source-Initiated Strobe for Data Transfer

Block Diagram

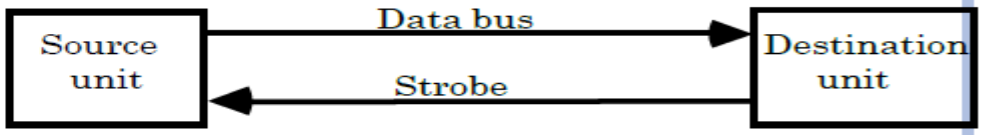


Timing Diagram

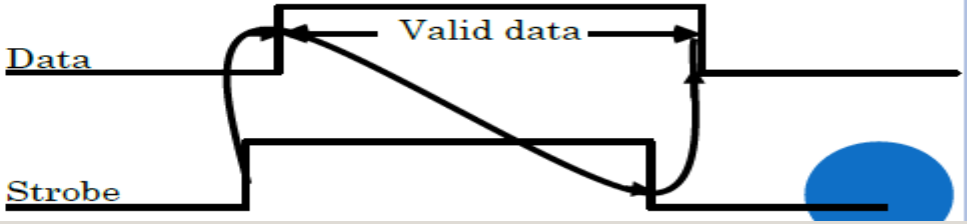


Destination-Initiated Strobe for Data Transfer

Block Diagram



Timing Diagram



Handshaking

In strobe methods:

- **Source-Initiated** The source unit that initiates the transfer has no way of knowing whether the destination unit has actually received data

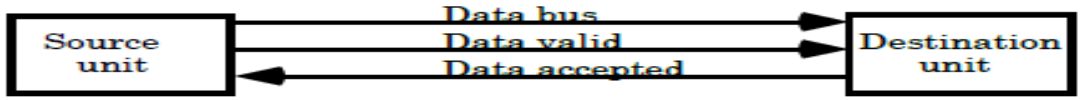
- **Destination-Initiated** The destination unit that initiates the transfer no way of knowing whether the source has actually placed the data on the bus

To solve this problem, the HANDSHAKE method introduces a second control signal to provide a reply to the unit that initiates the transfer

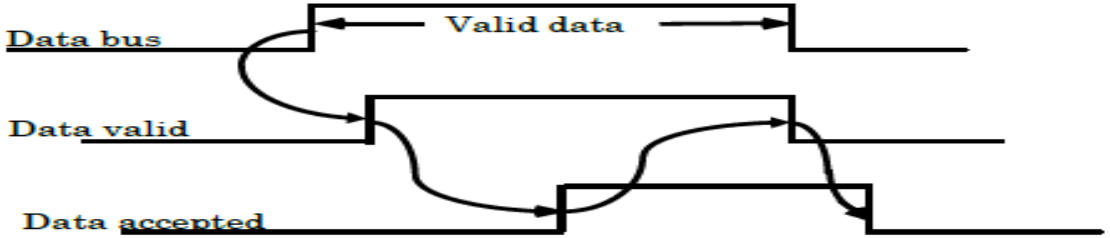
Source-Initiated Transfer Using Handshaking

- Allows arbitrary delays from one state to the next
- Permits each unit to respond at its own data transfer rate
- The rate of transfer is determined by the slower unit

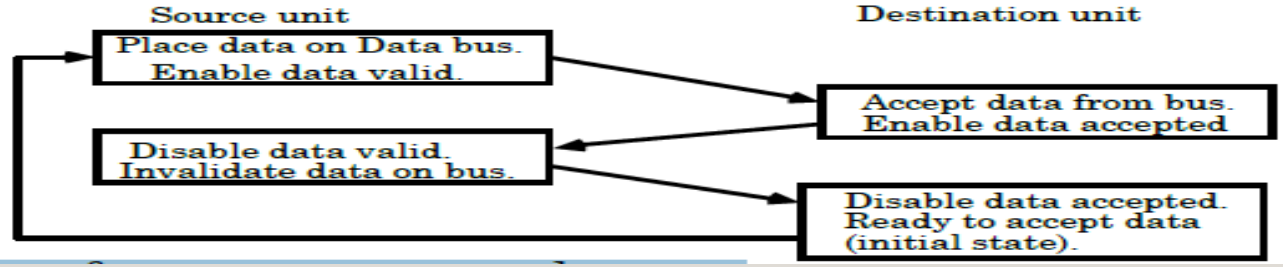
Block Diagram



Timing Diagram

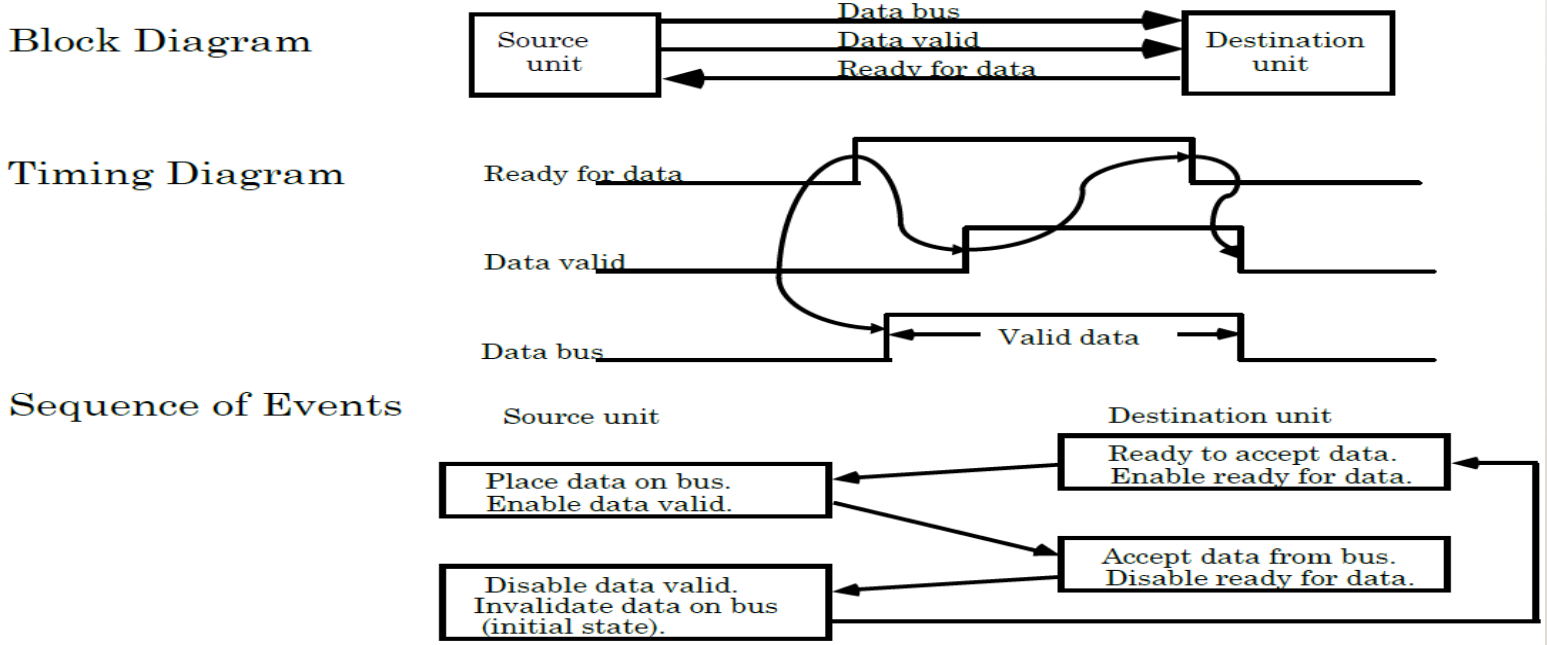


Sequence of Events



Destination-initiated Transfer Using Handshaking

- Handshaking provides a high degree of flexibility and reliability because the successful completion of a data transfer relies on active participation by both units
- If one unit is faulty, data transfer will not be completed -> Can be detected by means of a timeout mechanism, which produces a alarm if data transfer is not completed in time.



What is Buffering in OS?

- In operating systems, **buffering** is a technique which is used to enhance the performance of I/O operations of the system.
- Basically, buffering in operating system is a method of storing data in a buffer or cache temporarily, this buffered data then can be accessed more quickly as compared to the original source of the data.
- In a computer system, data is stored on several devices like hard discs, magnetic tapes, optical discs and network devices.
- In the case, when a process requires to read or write data from one of these storage devices, it has to wait while the device retrieves or stores the data.
- This waiting time could be very high, especially for those devices which are slow or have a high latency.
- This problem can be addressed by buffering. Buffering provides a temporary storage area, called **buffer.**
- Buffer can store data before it is sent to or retrieved from the storage device.
- When the buffer is fully occupied, then data is sent to the storage device in a batch, this will reduce the number of access operations required and hence improves the performance of the system.

Reason for Buffering in OS?

- The following three are the major reasons for buffering in operating systems −

  - Buffering creates a synchronization between two devices having different processing speed. For example, if a hard disc (supplier of data) has high speed and a printer (accepter of data) has low speed, then buffering is required.
  - Buffering is also required in cases where two devices have different data block sizes.
  - Buffering is also required to support copy semantics for application I/O operations.

Types of Buffering

Single Buffering

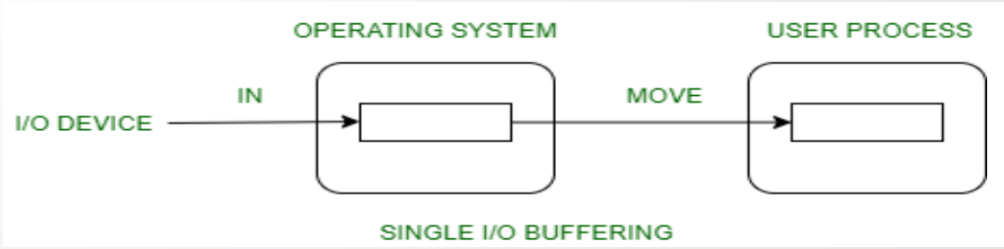
- It is the simplest buffering that operating system can support.
- In the case of single buffering, when a process issues an I/O request, the operating system assigns a buffer (or cache) in the system potion of the main memory to the operation.
- Then, the input transfers are made to the buffer and are moved to the user space when needed.

Double Buffering

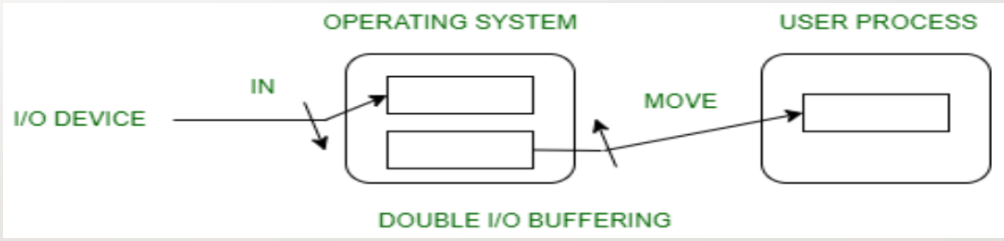
- Double buffering is an extended variant of single buffering.
- In this type buffering, a process can transfer data to or from one buffer while the operating system removes or fills the other.
- Therefore, double buffering has two system buffers instead of one.

Circular Buffering

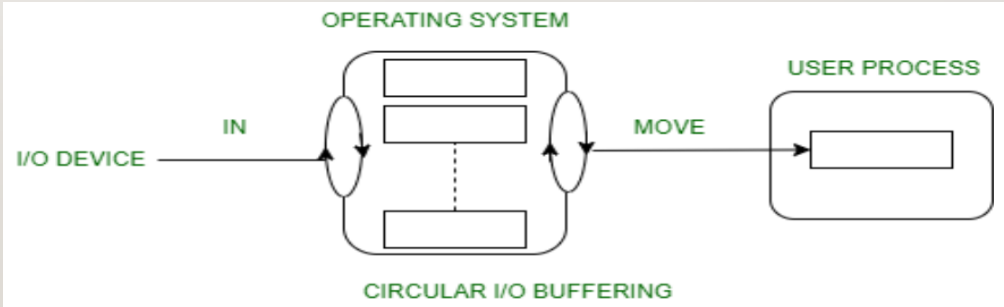
- When more than two buffers are used, then it is called circular buffering.
- It is used to solve the issues associated with the double buffering technique.
- Sometimes, the double buffering becomes insufficient, when the process performs rapid bursts of I/O. In the circular buffer, each individual buffer acts a unit.



SINGLE I/O BUFFERING



DOUBLE I/O BUFFERING



CIRCULAR I/O BUFFERING

Buffering

Advantages of Buffering

- Buffering reduces the number of I/O operations required to access data.
- Buffering reduces the amount of time for that processes have to wait for the data.
- Buffering improves the performance of I/O operations as it allows data to be read or written in large blocks instead of 1 byte or 1 character at a time.
- Buffering can improve the overall performance of the system by reducing the number of system calls and context switches required for I/O operations.

Disadvantages of Buffering

- Buffers of large sizes consume significant amount of memory that can degrade the system performance.
- Buffering may cause a delay between the time data is read or written and the time it is processed by the application.
- Buffering may also impact the real-time system performance and hence, can cause synchronization issues.

SELF-ASSESSMENT QUESTIONS

1. In handshaking protocols, what does "flow control" refer to?

(a) The direction of data flow between the CPU and the I/O device

(b) Controlling the rate of data transmission

(c) The physical connection between the CPU and the I/O device

(d) Ensuring error-free data transfer during I/O operations

2. What is the purpose of a "ready" signal in handshaking?

(a) To indicate that the I/O device is ready to receive data

(b) To signal the CPU to initiate an I/O operation

(c) To indicate the completion of an I/O operation

(d) To allocate memory for the I/O device

SELF-ASSESSMENT QUESTIONS

3. Which handshaking method is suitable for real-time applications where timing is critical?

(a) Asynchronous handshaking

(b) Synchronous handshaking

(c) Simplex handshaking

(d) Full-duplex handshaking

4. In full-duplex handshaking, how are data transmission and reception handled between the CPU and the I/O device?

(a) Only one direction at a time

(b) Simultaneously in both directions

(c) Alternating directions at regular intervals

(d) Data transmission is handled by the CPU, and data reception is handled by the I/O device

SELF-ASSESSMENT QUESTIONS

5. What is buffering in the context of I/O operations?

(a) Storing data in a temporary memory area

(b) Controlling the flow of data between the CPU and I/O devices

(c) Handshaking signals between the CPU and I/O devices

(d) Allocating memory dynamically for I/O operations

6. What is the primary purpose of using a buffer in I/O operations?

(a) To permanently store data

(b) To speed up data transfer between the CPU and I/O devices

(c) To control interrupts during I/O operations

(d) To allocate memory for the I/O device

SELF-ASSESSMENT QUESTIONS

7. Which of the following is an advantage of buffering in I/O operations?

(a) Increased latency
(b) Reduced response time
(c) Lower data transfer rates
(d) Increased CPU utilization

8. What is a drawback of using a small-sized buffer in I/O operations?

(a) Increased data transfer rates
(b) Higher memory requirements
(c) Lower latency
(d) Increased chance of overflow or underflow

TERMINAL QUESTIONS

Short answer questions:

1. Summarize various Asynchronous Data Transfer methods.
2. List potential reasons for buffering in I/O operations.
3. List the various data transfer methods in IO communication.

Long answer questions:

1. Elaborate role and architecture of I/O buses and interface modules with diagram.
2. Analyze buffering with its types in the context of I/O operations.
3. Analyze handshaking with its types in the context of I/O communication.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books:

1. Computer Organization by Carl Hamacher, Zvonko Vranesic and Saftwat Zaky.
2. Computer System Architecture by M. Morris Mano
3. Computer Organization and Architecture by William Stallings

Sites and Web links:

1. https://www.tutorialspoint.com/what-is-handshaking

THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – Digital Design & Computer Architecture