5 INPUT/OUTPUT INTERFACE

5.1 INTRODUCTION

An interface is a point of interaction between a user and a device, system or software application. It's like a bridge that allows users to communicate with and control the technology they're using.

An interface can take different forms, such as a graphical user interface (GUI) or a command-line interface (CLI). A GUI typically includes visual elements like buttons, menus, and icons, making it more user-friendly. On the other hand, a CLI is a text-based commands for interaction. The purpose of an interface is to enable users to interact with technology that is easy to understand and efficient. Interfaces can be found in various devices and applications, from smartphones and computers.

5.2 I/O INTERFACE

An Input/Output interface, is a connection between a computer and external devices. It provides a way for the computer to send and receive data to and from devices such as keyboards, mice, printers, monitors, and storage devices. It is a method and function that includes reading input from the user, writing output on the screen or files, and handling various types of data transfer.

5.2.1 Working of I/O Interface

1. The CPU sends a request to an external device, for a specific operation to be performed (e.g., read data, write data, or control commands).
2. The I/O interface receives the request and translates it into the appropriate signals and protocols required by the specific device.
3. The request is sent to the external device through the appropriate channels (e.g., data buses, control lines, or network connections).
4. The external device performs the requested operation and sends a response back to the CPU through the I/O interface.

5.2.2 Importance of Input-Output Interface

I/O devices are important because they allow us to interact with computers and receive information from them. Input devices like keyboards and mouse, give commands and input data, while output devices like monitors and printers display

or produce the results from the computer. Some importance of I/O interface are given below:-

Communication channel:- I/O devices are important because they enable communication between a computer system and the outside world. They serve as the bridge through which we interact with computers and receive information from them.

Interface between devices:- Input devices, such as keyboards, mice, and touchscreens, allow us to provide commands and input data to the computer. They enable us to type, click, swipe, and perform various actions to control and interact with software applications.

Produce Output:- Output devices, such as monitors, printers, and speakers, display or produce the results or output from the computer. They allow us to see visual information, hear audio, and obtain physical copies of documents or images.

Entertainment purpose:- I/O devices are essential for various tasks and activities. They play a crucial role in everyday computing, from typing documents and browsing the internet to playing games and watching videos. Without input devices, we wouldn't be able to communicate our intentions to the computer.

Expand the capabilities:- I/O devices expand the capabilities of computer systems by allowing them to interact with a wide range of external devices and peripherals. For example, USB drives, external hard drives, and memory cards serve as input and output devices, enabling us to transfer data between computers and store information externally.

In last, I/O devices are important because they facilitate communication, interaction, and information exchange between computers and users. They enable us to input commands, provide data, receive output, and interact with the digital world in meaningful ways.

5.2.3 Requirement of I/O Devices

The requirements of an input-output interface can vary depending on the specific application and devices. However, there are some common requirements that are generally important for an effective interface:

1. **Compatibility:** The interface should be compatible with the devices or systems it is connecting. This includes matching physical connectors, voltage levels, communication protocols, and data formats.
2. **Reliability:** The interface should provide reliable and consistent data transfer between the input and output devices. It should be able to handle

various types of data, such as analog or digital signals, and ensure accurate transmission without errors or data loss.

3. **Speed and Bandwidth:** The interface should support the required speed and bandwidth for the intended data transfer. This includes considering factors like data rate, latency, and throughput to ensure efficient communication between the input and output devices.
4. **Flexibility:** The interface should be flexible enough to accommodate different types of input and output devices. It should be able to handle a range of data formats, signal levels, and device configurations to support diverse applications.
5. **Ease of Use:** The interface should be user-friendly and easy to operate. It should have clear instructions, intuitive controls, and appropriate feedback mechanisms to facilitate smooth interaction between the user and the input-output devices.
6. **Scalability:** The interface should be scalable to accommodate future upgrades or expansions. It should have provisions for adding or removing devices, supporting higher data rates, and adapting to evolving technology standards.

These are some common requirements of an input-output interface.

5.2.4 Components of I/O Interface

The components of an I/O (Input/Output) interface typically include:

1. **Ports:** These are physical connectors on a device that allow for the connection of external devices or cables. Examples of ports include USB ports, HDMI ports, Ethernet ports, and audio jacks.
2. **Controllers:** These are electronic circuits or chips that manage the communication between the device and the connected peripherals. They handle tasks such as data transfer, signal conversion, and protocol interpretation.
3. **Drivers:** These are software components that enable the operating system to communicate with the I/O interface and control the connected devices. Drivers provide the necessary instructions and protocols for the proper functioning of the interface and peripherals.
4. **Buffers:** These are temporary storage areas in the interface that hold data during the transfer between the device and the connected peripherals. Buffers help manage the speed and flow of data to ensure smooth communication.

Scanned with OKEN Scanner

5. **Protocols:** These are sets of rules and standards that define how data is formatted, transmitted, and received between devices. Examples of protocols include USB, Ethernet, HDMI, Bluetooth, and Wi-Fi.
6. **Cables and connectors:** These physical components provide the physical connection between the device and the peripherals. Different types of cables and connectors are used depending on the interface and the devices being connected.

These are some of the main components that work together to facilitate communication and data transfer between devices and peripherals.

5.2.5. I/O Interface Applications

I/O interface is like a bridge that enables devices to understand and work together. I/O interfaces are used in various fields like computers, smartphones, cars, industrial automation, and more. They make our devices more versatile and interconnected. There are few I/O interface applications–

1. In computer hardware, I/O interfaces are used to connect peripherals like keyboards, mice, and printers to a computer system.
2. In the field of audio and video, I/O interfaces enable the connection of devices like speakers, microphones, cameras, and displays to capture and reproduce sound and images.
3. In networking, I/O interfaces are used to connect devices to a network, such as Ethernet ports on routers or network cards in computers.
4. In software development, APIs (Application Programming Interfaces) provide a set of rules and protocols for different software systems to interact with each other, enabling the development of applications that can leverage the functionality of other software services.
5. In user interfaces, I/O interfaces allow users to interact with computer systems through graphical user interfaces (GUIs), touchscreens, or voice commands.
6. In the automotive industry, I/O interfaces are used to connect various sensors that enable communication between different components of a vehicle, such as the engine, brakes etc.
7. In the field of robotics, I/O interfaces play a crucial role in connecting sensors, motors, and other hardware components, allowing robots to perform tasks.

Scanned with OKEN Scanner

8. In the healthcare industry, I/O interfaces are utilized in medical devices to capture and transmit data from sensors, such as heart rate monitors, blood pressure monitors, and glucose meters.

5.3 USES OF I/O INTERFACE

I/O interface applications are used for connecting and communicating between different devices, systems, or components. They enable the exchange of data, signals, and commands, allowing for seamless interaction and integration. Some common uses include:

1. **Connecting input devices:** I/O interfaces allow us to connect keyboards, mice, touchscreens, and other input devices to computers or mobile devices, enabling us to interact with them.
2. **Connecting output devices:** They also facilitate the connection of displays, printers, speakers, and other output devices to computers or other systems, allowing us to see, hear, or receive information.
3. **Data acquisition and control:** I/O interfaces are used in various fields, such as industrial automation, where they connect sensors, actuators, and control systems, enabling the monitoring and control of processes.
4. **Networking:** I/O interfaces are essential for connecting devices to networks, such as Ethernet ports or wireless adapters, enabling communication and data transfer between devices.
5. **External storage:** They allow us to connect external storage devices like hard drives or USB flash drives to computers or other devices, facilitating data transfer and backup.
6. **Audio and video connectivity:** I/O interfaces are used to connect audio and video devices, such as cameras, microphones, speakers, and displays, enabling the capture, recording, playback, and transmission of audio and video signals.

5.3.1 Functions of I/O Interface

The function of the I/O interface is to facilitate communication between a computer system and external devices or users. It allows for the exchange of data and instructions, enabling input from external sources and output to externa destinations.

The I/O interface serves as a bridge between the computer's internal processin and the outside world. It enables users to interact with the computer system b

APIs are commonly used to integrate different systems, access data from external sources, and build new applications on top of existing platforms. They play a crucial role in modern software development and enable the creation of powerful and interconnected applications.

5.6 METHODS OF ASYNCHRONOUS DATA TRANSFER

There are several methods for asynchronous data transfer. Few are given below:-

1. **Message Queues:** This method involves using a message queue system, where data is sent and stored in a queue. The sender can add messages to the queue, and the receiver can retrieve and process them at their own pace.
2. **Publish/Subscribe:** In this method, publishers send messages to a topic or channel, and subscribers who are interested in that topic receive the messages asynchronously.
3. **Asynchronous APIs:** Asynchronous API sends the request and then continues with other tasks without waiting for the response. Once the response is received, the program can handle it asynchronously, often through the use of callback functions or promises. Asynchronous APIs are commonly used in web development.

These are just a few examples, and there are other methods and technologies available depending on the specific use case and programming language you're working with. The choice of method depends on factors such as scalability, reliability, and the requirements of your application.

5.7 SYNCHRONOUS

Synchronous refers to a method where the sender and receiver are synchronized and operate in a coordinated manner. It means that both parties follow a specific timing mechanism and are in lockstep with each other during the data transfer process.

5.7.1 Synchronous Data Transfer

Synchronous data transfer refers to a method of communication where data is transferred in a synchronized and coordinated manner between a sender and receiver. In this method, both the sender and receiver operate on the same clock or timing mechanism, ensuring that data is transmitted and received at predetermined intervals.

5.5 ASYNCHRONOUS

Asynchronous refers to a method of communication or operation where events or tasks can occur independently of each other and in no particular order. In other words, it means that actions can happen at the same time without having to wait for each other to complete.

In an asynchronous process, tasks are initiated and executed independently, and the system doesn't have to pause or wait for a specific task to finish before moving on to the next one.

5.5.1 Asynchronous Data Transfer

Asynchronous data transfer refers to the process of transferring data between systems or devices without requiring them to be synchronized in real-time. It allows for more flexibility and efficiency in data transmission. There are few examples of asynchronous data transfer:-

- ✓ One common example of asynchronous communication is email. When you send an email, you don't have to wait for an immediate response from the recipient. The email is sent, and you can continue with other tasks while waiting for a reply. The recipient can read and respond to the email at their convenience, and you will eventually receive their response.
- ✓ Another example is file uploads or downloads. When you upload a file to a server or download a file from the internet, the transfer happens asynchronously. You don't need to wait for the entire file to be uploaded or downloaded before you can continue using your device or performing other tasks.

Asynchronous data transfer allows for more efficient use of resources and enables systems to handle multiple tasks simultaneously. It's a fundamental concept in modern communication and computing.

5.5.2 Asynchronous API

API stands for Application Programming Interface. It's a set of rules and protocols that allows different software applications to communicate and interact with each other in a non-blocking way. It means that instead of waiting for a response before moving on to the next task, the program can continue executing other tasks while waiting for the API response. This can make the program more efficient and responsive.

Scanned with OKEN Scanner

The process involves the following steps:

1. The CPU sends a command or request to the I/O controller, specifying the desired operation.
2. The I/O controller receives the command and translates it into a format that the external device can understand.
3. The I/O controller sends the command to the appropriate device through the I/O bus.
4. The external device performs the requested operation and sends back any necessary data or status updates.
5. The I/O controller receives the data from the external device and delivers it to the CPU via the I/O bus.

This way, the CPU can interact with various peripherals like keyboards, mouse, printers, storage devices, and more.

Different I/O bus architectures exist, such as PCI, USB, or SATA, each with its own specifications and features. The choice of bus depends on the specific requirements of the system and the devices being used.

5.4.1 Working of I/O Bus

- The I/O bus connecting the computer's internal parts, like the CPU, with external devices, such as keyboards, mice, printers, and more.
- I/O bus is as a highway that allows data and commands to travel between the computer and these external devices. For this connection, we use special hardware components, like I/O controllers or interfaces.
- When you want to interact with an external device, like pressing a key on your keyboard, the CPU sends a command to the I/O controller. The controller then translates that command into a language the device understands and sends it through the I/O bus.
- The device performs the requested action, like displaying the letter you pressed on the screen, and sends the result back to the I/O controller. The controller then passes that information back to the CPU through the I/O bus.

So, I/O bus is all about connecting the computer with external devices so they can communicate and work together smoothly.

Scanned with OKEN Scanner

providing input, such as keyboard input, mouse movements, touch gestures, or voice commands. This input is then processed by the computer system.

Similarly, the I/O interface enables the computer system to produce output that can be understood by external devices or users. This output can take various forms, such as displaying information on a screen, printing documents, playing audio, or sending data over a network connection.

The I/O interface also includes software components, such as device drivers, which allow the operating system to communicate with different hardware devices. These drivers provide a standardized way for the computer system to interact with peripherals, ensuring compatibility and efficient data transfer.

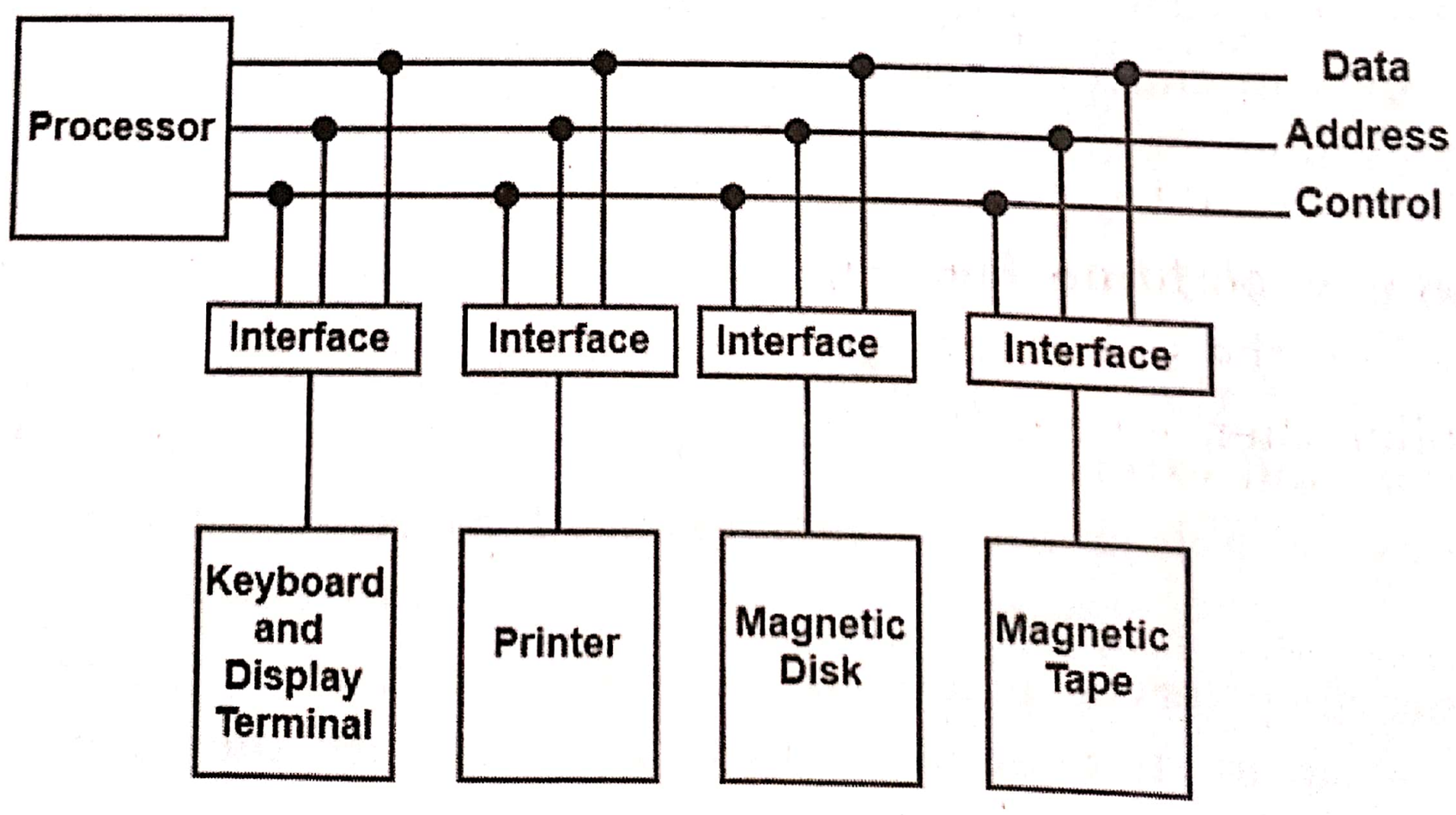
Overall, the function of the I/O interface is to enable seamless communication and interaction between a computer system and the external world. It allows for the exchange of data, commands, and feedback, facilitating the use of various input devices and output devices.

5.4 INTERFACING WITH I/O BUS

When it comes to interfacing with the I/O bus, it's all about establishing a connection between the computer's internal components and the external devices. The I/O bus acts as a pathway for data and instructions to flow between the CPU and the peripherals.

To interface with the I/O bus, the computer system uses specific hardware components, such as I/O controllers or I/O interfaces. These components handle the communication between the CPU and the external devices. I/O Bus consists of data lines, address lines and control lines.

Connection of I/O Bus to I/O Devices



During synchronous data transfer, the sender and receiver maintain a constant and continuous flow of data. The sender waits for acknowledgment from the receiver before sending the next piece of data, ensuring that both parties are in sync. This synchronization helps to prevent data loss or corruption during transmission.

Synchronous data transfer is commonly used in scenarios where precise timing and coordination are necessary, such as in telecommunications, networking protocols, or real-time data processing. It allows for efficient and reliable data transfer between devices or systems.

5.7.2 Characteristics of Asynchronous and Synchronous Data Transfer

The characteristics of asynchronous and synchronous data transfer as follows

Asynchronous data transfer:

- In asynchronous data transfer, data is transmitted in individual units called characters, with each character being preceded by start and stop bits.
- There is no fixed timing or synchronization between the sender and receiver. Each character is sent independently and can have varying intervals between them.
- Asynchronous data transfer is typically used for shorter distances and slower transmission rates.
- It is more flexible and tolerant of variations in timing, making it suitable for scenarios where devices may have different clock speeds or when there is a need for intermittent data transmission.

Synchronous data transfer:

- In synchronous data transfer, data is transmitted in a continuous stream without start and stop bits.
- The sender and receiver are synchronized using a common clock signal or timing mechanism. Data is transmitted and received at predetermined intervals.
- Synchronous data transfer is often used for longer distances and higher transmission rates.
- It is more efficient and reliable for transferring large amounts of data, as the continuous stream of data reduces overhead and allows for faster transfer speeds.

5.7.3 Functions/ Advantages of Asynchronous Data Transfer

Asynchronous data transfer has several functions and advantages:

Scanned with OKEN Scanner

In this method, a strobe signal is transmitted along with the data signal. The strobe signal acts as a timing reference, indicating the valid data bits to the receiver. It is typically a pulse or a signal that transitions from low to high or high to low at specific intervals.

The sender transmits the data bits along with the strobe signal. The receiver uses the strobe signal to sample and interpret the data bits at the correct timing. When the strobe signal transitions, the receiver knows that the corresponding data bit is valid and can be captured.

The strobe control method helps ensure that the sender and receiver are synchronized and that the data is sampled accurately. It allows for precise timing and reliable data transfer, particularly in situations where the sender and receiver may have slightly different clock speeds.

This method is commonly used in parallel data transfer interfaces, such as parallel buses or memory interfaces, where multiple bits are transferred simultaneously.

Following are some simple example to explain the strobe control method.

For example–

Imagine you're playing a game with your friend. You both have a set of cards with numbers written on them. To synchronize your moves, you decide to use a special signal called a "flash."

Here's how it works:

1. You hold up a card with a number on it, let's say "5."
2. As soon as you show the card, you quickly flash a small flashlight to indicate that the number is valid.
3. Your friend sees the flash and knows that the number on the card is ready to be recorded.
4. Your friend quickly writes down the number "5" on their sheet of paper.
5. Once your friend has written down the number, you turn off the flashlight to signal that the data transfer is complete.

By using the flash as a strobe signal, you and your friend can synchronize your actions. The flash indicates when the number is valid and ready to be recorded, ensuring that you both have the same information.

In this example, the card with the number represents the data, and the flash from the flashlight represents the strobe signal. Just like in the strobe control method, the flash helps you and your friend stay in sync during the game.

Scanned with OKEN Scanner

5.8 DIFFERENCE BETWEEN SYNCHRONOUS AND ASYNCHRONOUS TRANSMISSION

Synchronous Transmission: Synchronous transmission is a method where data is transmitted in a continuous stream at a fixed rate. It requires both the sender and receiver to be synchronized or in perfect coordination. This synchronization is achieved by using a clock signal that keeps both ends in sync. The data is divided into frames or blocks of a specific size, and these frames are transmitted one after the other without any gaps or delays. This constant stream of data allows for reliable and efficient communication, especially in applications where timing is critical.

Asynchronous Transmission:- It is a method where data is transmitted in individual units called bytes. Each byte is preceded by a start bit and followed by a stop bit. These start and stop bits serve as markers to indicate the beginning and end of each byte. Unlike synchronous transmission, asynchronous transmission does not require strict synchronization between the sender and receiver. It allows for variations in timing and pauses between transmissions. This flexibility makes it suitable for applications where the timing between data units may vary.

So, synchronous transmission maintains a continuous stream of data at a constant rate and requires synchronization between the sender and receiver. Asynchronous transmission, transmits data in individual units with start and stop bits, allowing for more flexibility in timing and pauses.

5.9 STROBE SIGNAL

A strobe signal is used to indicate the availability or validity of data. The data is transferred when the strobe signal is active or transitions from one state to another. This helps ensure that the data is captured or read at the correct time.

The strobe signal acts as a synchronization mechanism, allowing the receiving component to know when to expect valid data and preventing data corruption or loss.

The specific implementation of the strobe method can vary depending on the system and its requirements, but it is commonly used in scenarios where precise timing and synchronization are crucial for reliable data transfer.

5.9.1 Strobe Control Method

The strobe control method is a technique used in synchronous data transfer to synchronize the sender and receiver by using a separate control signal known as the strobe signal.

Scanned with OKEN Scanner

2. **Continuous Data Stream:** Synchronous data transfer transmits data in a continuous stream without start and stop bits. This eliminates the need for overhead associated with individual character framing, resulting in faster and more efficient data transfer.
3. **Higher Transmission Rates:** Synchronous data transfer is often used for high-speed transmission of large amounts of data. The continuous data stream and synchronized timing allow for faster transfer rates, making it suitable for applications that require rapid data exchange.
4. **Error Detection and Correction:** Synchronous data transfer can incorporate error detection and correction mechanisms such as checksums or cyclic redundancy checks (CRC). These mechanisms help to detect and correct errors during transmission, ensuring data integrity.
5. **Longer Distances:** Synchronous data transfer is commonly used for long-distance communication. It can handle greater distances without significant loss or degradation of the transmitted data.
6. **Reliable Communication:** The synchronized nature of synchronous data transfer ensures reliable communication between the sender and receiver. The continuous flow of data, along with error detection and correction mechanisms, helps to minimize data loss or corruption during transmission.

These functions and advantages make synchronous data transfer a suitable choice for scenarios that require precise timing, high-speed transmission, and reliable communication over longer distances.

5.7.5 Method of Synchronous Data Transfer

One common method of synchronous data transfer is the use of clock signals. In this method, the sender and receiver are synchronized using a shared clock signal. The clock signal acts as a timing reference, ensuring that both devices operate at the same speed and are in sync during data transmission.

During synchronous data transfer, data is transmitted in a continuous stream without start and stop bits. To ensure reliable communication, the sender and receiver must have a predetermined agreement on the clock signal's frequency and timing.

Synchronous data transfer is commonly used in communication protocols such as synchronous serial interfaces and synchronous communication standards like synchronous optical networking (SONET) and synchronous digital hierarchy (SDH).

Scanned with OKEN Scanner

1. **Flexibility:** Asynchronous data transfer allows for more flexibility in terms of timing and data transmission. It doesn't require strict synchronization between the sender and receiver, making it suitable for devices with varying clock speeds or intermittent data transmission.
2. **Start/Stop Bits:** Asynchronous data transfer uses start and stop bits to indicate the beginning and end of each character. This allows for easy identification and separation of individual characters, ensuring accurate data transmission.
3. **Error Detection:** Asynchronous data transfer can include error detection mechanisms such as parity bits or checksums. These help to detect and correct errors during transmission, ensuring data integrity.
4. **Compatibility:** Asynchronous data transfer is widely supported by various communication protocols and interfaces, making it compatible with a wide range of devices and systems.
5. **Lower Overhead:** Asynchronous data transfer has lower overhead compared to synchronous data transfer. It doesn't require continuous synchronization signals or a fixed timing mechanism, resulting in more efficient data transmission.
6. **Simplicity:** Asynchronous data transfer is relatively simple to implement and understand. It doesn't require complex synchronization mechanisms, making it easier to work with, especially in simpler communication systems.

These functions and advantages make asynchronous data transfer a suitable choice for scenarios where flexibility, compatibility, and simplicity are important considerations.

Both asynchronous and synchronous data transfer methods have their own advantages and are used in different scenarios based on factors such as distance, transmission rate, and the need for precise timing and synchronization.

5.7.4 Functions/ Advantages of Synchronous Data Transfer

Synchronous data transfer has several functions and advantages:

1. **Precise Timing:** Synchronous data transfer relies on a common clock signal or timing mechanism to synchronize the sender and receiver. This ensures precise timing and coordination between the two, allowing for accurate and efficient data transmission.

Scanned with OKEN Scanner

5.9.2 Working of Strobe Method

Strobe method is a technique used in certain areas of technology. In networking, the strobe method is a way to determine the distance between two devices. The strobe method works:

1. Device A sends out a signal, often in the form of a light or radio wave, towards Device B.
2. Device B receives the signal and immediately sends a response signal back to Device A.
3. Device A measures the time it takes for the response signal to arrive. By knowing the speed at which the signal travels, Device A can calculate the distance between the two devices.
4. This process can be repeated multiple times to get more accurate measurements and ensure consistency.

The strobe method is commonly used in applications such as radar systems, distance measurement tools, and even in some communication protocols. It allows devices to determine distances quickly and accurately without requiring complex equipment.

5.10 HANDSHAKING

Handshaking refers to the process of establishing a connection or synchronization between two devices or systems. It ensures that both sides are ready to send or receive data. In the context of i/o interfaces, handshaking protocols are used to manage the flow of data between devices, ensuring that data is transmitted accurately and efficiently. It's like a friendly handshake between devices before they start exchanging information.

Example of Handshaking Method

The handshaking method is a way for devices to establish a connection before exchanging data. For example- Imagine you and your friend want to exchange secret messages using a special code. Before you start sending the messages, you need to establish a connection and make sure you both are ready for exchanging the data.

5.10.1 Working of Handshaking Method

1. You and your friend stand facing each other, ready to exchange messages.
2. You extend your hand, and your friend does the same.
3. When your hands meet, you both give a firm handshake.

Scanned with OKEN Scanner

Q.13. Define the Serial transfer.

Ans. Serial transfer is a widely used method for transmitting data in a sequential manner, enabling communication between devices and systems. It allows for efficient and reliable data transmission, especially over long distances.

SHORT ANSWER TYPE QUESTION

1. What is I/O interface?
2. Explain the requirement of I/O devices.
3. Write an essay on I/O interfacing with I/O Bus.
4. What are the components of an I/O interface?
5. Define the methods of asynchronous data transfer.
6. Explain the advantages of synchronous data transfer.
7. Write the difference between synchronous and asynchronous transmission.
8. Explain the working of strobe and handshaking method.
9. Write a note on the asynchronous data transfer.
10. Explain the asynchronous serial transfer.

LONG ANSWER TYPE QUESTION

1. Write the importance of I/O interface.
2. Explain the functions of I/O interface.
3. Explain the working of I/O Bus.
4. Explain the characteristics of Asynchronous and Synchronous Data Transfer.
5. Write the advantages of asynchronous data transfer.

Scanned with OKEN Scanner

and translates it into the appropriate signals .The external device performs the requested operation and sends a response back to the CPU through the I/O interface.

Q.6. What do you know by asynchronous?

Ans. Asynchronous refers to a method of communication or operation where events or tasks can occur independently of each other and in no particular order.

Q.7. Give the example about asynchronous data transfer.

Ans. The example of asynchronous data transfer is as - Suppose you file uploads or downloads a file. When you upload a file to a server or download a file from the internet, the transfer happens asynchronously. You don't need to wait for the entire file to be uploaded or downloaded before you can continue using your device or performing other tasks.

Q.8. Write a note on synchronous.

Ans. Synchronous refers to a method where the sender and receiver are synchronized and operate in a coordinated manner. It means that both parties follow a specific timing mechanism and are in lockstep with each other during the data transfer process.

Q.9. Define the synchronous data transfer.

Ans. Synchronous data transfer refers to a method of communication where data is transferred in a synchronized and coordinated manner between a sender and receiver. In this method, both the sender and receiver operate on the same clock or timing mechanism, ensuring that data is transmitted and received at predetermined intervals.

Q.10. Define Asynchronous API.

Ans. API stands for Application Programming Interface. It's a set of rules and protocols that allows different software applications to communicate and interact with each other in a non-blocking way. APIs are commonly used to integrate different systems, access data from external sources, and build new applications on top of existing platforms.

Q.11. Define the Strobe signal.

Ans. A strobe signal is used to indicate the availability or validity of data. The data is transferred when the strobe signal is active or transitions from one state to another. This helps ensure that the data is captured or read at the correct time.

Q.12. What do you mean by the Handshaking?

Ans. Handshaking refers to the process of establishing a connection or synchronization between two devices or systems. It ensures that both sides are ready to send or receive data.

Scanned with OKEN Scanner

One common example of asynchronous serial transfer is the RS-232 interface, which is used for serial communication between devices like computers and modems. The data is transmitted one bit at a time, with the start and stop bits providing the necessary synchronization.

Overall, asynchronous serial transfer provides a flexible and reliable method of transmitting data, allowing for variations in timing and pauses between transmissions.

EXERCISE

VERY SHORT ANSWER TYPE QUESTIONS

Q.1. Define interface.

Ans. An interface is a point of interaction between a user and a device, system or software application. It's like a bridge that allows users to communicate with and control the technology they're using.

Q.2. What is an I/O interface?

Ans. An Input/Output interface, is a connection between a computer and external devices. It provides a way for the computer to send and receive data to and from devices such as keyboards, mice, printers, monitors, and storage devices.

Q.3. What are the different types of I/O interfaces?

Ans. I/O interfaces are used to connect devices and transfer data between them. There are different types of interfaces

1. USB (Universal Serial Bus)ports are used to connect devices like keyboards, mice etc.
2. HDMI (High-Definition Multimedia Interface) is used to connect devices like TVs, monitors, and projectors.
3. Ethernet allows devices to connect to a local area network (LAN).

Q.4. What is the purpose of an I/O interface?

Ans. The purpose of an interface is to enable users to interact with technology that is easy to understand and efficient. The I/O interface acts as a bridge between the computer and the outside world, allowing for data exchange and interaction.

Q.5. Write the working of I/O interface

Ans. Working of I/O interface is as below:-The CPU sends a request to an external device, for read and write data, or control commands. It receives the request

4. The handshake serves as a signal that you're ready to start exchanging messages.
5. Once the handshake is complete, you can begin sending your secret messages to each other.

The handshaking method acts as a way to synchronize your actions. It ensures that both parties are prepared before initiating the data exchange.

5.11 SERIAL TRANSFER

Serial transfer is a method of sending data where the bits (0s and 1s) are transmitted one after another in a sequential manner over a single communication line. It's like sending a message letter by letter instead of all at once.

In serial transfer, the data is broken down into smaller chunks called "bits." These bits are then sent one by one, starting from the first bit and moving through the communication line. The receiver collects these bits and reconstructs the original data at the other end. Think of it like a conveyor belt where each bit is like a small package moving along. The sender puts the packages (bits) on the belt one by one, and the receiver collects them in the same order they were sent.

Serial transfer is commonly used in various communication systems, such as serial ports, USB connections, and even in network communication protocols like Ethernet. It allows for reliable data transmission over longer distances and with fewer wires or connections.

5.11.1 Asynchronous Serial Transfer

Asynchronous serial transfer is a method of transmitting data where each byte is sent individually, accompanied by start and stop bits. Unlike synchronous transmission, there is no fixed timing or clock signal to synchronize the sender and receiver.

In asynchronous serial transfer, each byte is preceded by a start bit, which is a low voltage level, and followed by one or more stop bits, which are high voltage levels. These start and stop bits serve as markers to indicate the beginning and end of each byte.

The sender and receiver do not need to be perfectly synchronized because the start and stop bits allow for variations in timing and pauses between transmissions. This flexibility makes asynchronous serial transfer suitable for applications where the timing between data units may vary.

Scanned with OKEN Scanner