UNIT 9: MEMORY AND IO ACCESS

UNIT STRUCTUTRE

9.1 Learning Objective

9.2 Introduction

9.3 Memory Read Write Operations

9.4 Concept of Handshaking

9.5 Polling Techniques

9.5.1 Serial Polling

9.5.2 Hub Polling

9.6 Interrupt driven I/O

9.7 Priority and Daisy Chaining Technique

9.8 Memory Mapping

9.9 Let Us Sum Up

9.10 Further Readings

9.11 Answers to Check Your Progresses

9.12 Possible Questions

9.1 LEARNING OBJECTIVE

After going through this unit, you will be able to-

- Identify the process of memory read and write operations.
- Explain the concept of handshaking in computer memory.
- Describe the four signals involved in the handshaking process.
- Differentiate between serial and hub polling techniques in computer memory.
- Evaluate the efficiency of polling techniques based on the number of memory devices in a system.

9.2 INTRODUCTION

Memory and input/output (I/O) devices are essential components of modern computing systems. Memory devices are used to store and retrieve data and instructions that are required for the operation of a computer, while I/O devices are used to communicate with the external world, including devices such as keyboards, mice, printers, and network interfaces. In this way, memory and I/O

devices work together to enable the processing of data and the interaction of the computer with its environment.

The unit discusses the important concepts related to computer memory, including memory read and write operations, handshaking, and polling techniques. It explains how these concepts are crucial for the proper functioning of computers and enable them to store information, execute programs, and perform complex calculations. The unit also highlights the need for high-speed and efficient data transfer in modern computer systems.

9.3 MEMORY READ WRITE OPERATIONS

Memory read and write operations refer to the process of accessing and manipulating data in computer memory. These operations are critical to the functioning of computers and are used in all types of computing devices, from personal computers to supercomputers.

Memory Read Operations: Memory read operation is the process of retrieving data from computer memory. When a program needs to access data from memory, it sends a read request to the memory controller, which retrieves the data from the specified memory location and transfers it to the program. The read operation can be performed on any type of memory, including RAM, ROM, and cache memory.

The process of memory read operations is initiated by the processor by sending an address to the memory controller. The address specifies the memory location where the data is stored. The memory controller then retrieves the data from that location and sends it back to the processor. The retrieved data is temporarily stored in the processor's register or cache memory, where it can be accessed by the program.

Memory Write Operations: Memory write operation is the process of storing data into computer memory. When a program needs to modify the value of a variable, it sends a write request to the memory controller, which stores the new value at the specified memory location. The write operation can also be performed on any type of memory, including RAM, ROM, and cache memory.

The process of memory write operations is initiated by the processor by sending an address and the new data to the memory controller. The address specifies the memory location where the new data is to be stored. The memory controller

then writes the new data into that location. Once the write operation is complete, the new data is stored in memory, and it can be accessed by other programs.

Memory read and write operations are critical to the functioning of modern computers. They enable programs to access and manipulate data stored in memory, allowing computers to perform complex calculations, store information, and execute programs. The efficiency of memory read and write operations can have a significant impact on the overall performance of a computer system, making it essential for memory controllers to be designed for high-speed and efficient data transfer.

9.4 CONCEPT OF HANDSHAKING

In computer memory, handshaking refers to the process of establishing communication between the memory controller and the memory device. It involves a series of signals that are exchanged between the two devices to ensure that data is transmitted accurately and reliably.

Handshaking is essential in computer memory as it helps to prevent errors that can occur during data transfer. The memory controller and the memory device must be synchronized to ensure that data is transferred at the correct time and in the correct sequence. This is particularly important in modern computer systems where the speed of data transfer is critical for optimal performance.

The handshaking process typically involves a series of signals that are exchanged between the memory controller and the memory device. These signals include:

1. Address Signal - The memory controller sends an address signal to the memory device indicating the location of the data to be read or written.
2. Data Signal - The memory controller sends a data signal to the memory device when data is to be written, or the memory device sends a data signal to the memory controller when data is to be read.
3. Control Signal - The control signal is used to indicate the start and end of a transaction. It signals the memory device that the data transfer is about to begin and when it is complete.

4. Acknowledgment Signal - The memory device sends an acknowledgment signal to the memory controller to indicate that the data has been received correctly.

The handshaking process ensures that the memory controller and the memory device are synchronized and that data is transmitted accurately and reliably. The process also helps to prevent errors that can occur during data transfer, such as data corruption, data loss, or incorrect data sequencing.

In summary, handshaking is a critical concept in computer memory as it helps to ensure accurate and reliable data transfer. By exchanging a series of signals between the memory controller and the memory device, handshaking helps to synchronize the two devices, prevent errors, and improve the overall performance of the computer system.

9.5 POLLING TECHNIQUES

Polling is a technique used in computer memory to manage data transfer between the memory controller and the memory device. It involves the memory controller periodically checking the status of the memory device to determine if it is ready to transfer data. There are two primary polling techniques used in computer memory: Serial polling and Hub polling.

9.5.1 Serial Polling

Serial polling is a polling technique in which the memory controller checks the status of each memory device in a serial manner, one at a time. This technique is used in systems where there is a small number of memory devices that need to be polled. In serial polling, the memory controller sends a polling signal to the first memory device and waits for a response. If the first memory device is ready, it sends an acknowledge signal back to the memory controller, indicating that it is ready to transfer data. The memory controller then proceeds to poll the next memory device and repeats the process until all the memory devices have been polled.

Serial polling is simple to implement and is suitable for systems with a small number of memory devices. However, it can be slow and inefficient in systems with a large number of memory devices, as each device must be polled individually.

9.5.2 Hub Polling

Hub polling is a polling technique in which a hub device is used to manage the polling of multiple memory devices. In this technique, the memory controller sends a polling signal to the hub device, which then polls each memory device in parallel. The hub device collects the responses from each memory device and sends them back to the memory controller.

Hub polling is more efficient than serial polling, as multiple memory devices can be polled simultaneously. It is suitable for systems with a large number of memory devices or systems that require high-speed data transfer.

Polling techniques are used in computer memory to manage data transfer between the memory controller and the memory device. Serial polling is suitable for systems with a small number of memory devices, while hub polling is more efficient and suitable for systems with a large number of memory devices. The choice of polling technique depends on the system requirements, including the number of memory devices, data transfer speed, and system complexity.

CHECK YOUR PROGRESS -1

1. What is memory read operation?

……
……

2. What is the purpose of handshaking in computer memory?

……
……

3. What is hub polling?

……
……

9.6 INTERRUPT DRIVEN I/O

Interrupt-driven I/O is a technique used in computer memory to manage input/output operations. It involves the use of interrupts to notify the processor when data is ready to be transferred between the memory controller and the memory device.

When a memory device has data to transfer, it sends an interrupt request to the processor, which interrupts the current program and switches to an interrupt handler. The interrupt handler then communicates with the memory device and transfers the data to or from the memory device as required.

Interrupt-driven I/O has several advantages over other I/O techniques. Firstly, it is more efficient as it allows the processor to continue executing other instructions while waiting for data to be transferred. This means that the processor can perform other tasks while waiting for I/O operations to complete, improving system performance.

Secondly, interrupt-driven I/O is more flexible as it allows the processor to handle multiple I/O requests simultaneously. This means that multiple memory devices can be serviced at the same time, increasing the overall I/O throughput.

Thirdly, interrupt-driven I/O is more reliable as it ensures that no data is lost or corrupted during the transfer process. The interrupt handler ensures that the data transfer is complete before returning control to the processor, reducing the risk of errors.

However, interrupt-driven I/O also has some drawbacks. It requires additional hardware and software support to manage the interrupt requests and handlers, which can increase system complexity. It also requires careful handling to avoid interrupt storms, where multiple interrupts are generated simultaneously, which can cause system instability.

In summary, interrupt-driven I/O is a technique used in computer memory to manage input/output operations. It uses interrupts to notify the processor when data is ready to be transferred, providing a more efficient, flexible, and reliable I/O solution. However, it also requires additional hardware and software support and careful handling to avoid system instability.

9.7 PRIORITY AND DAISY CHAINING TECHNIQUE

Priority and daisy chaining are two techniques used in computer memory to manage interrupt requests from multiple devices. These techniques help ensure that interrupts are handled in a timely and efficient manner.

Priority Technique: The priority technique is a method of handling interrupt requests based on the priority of the device generating the request. Each device is assigned a priority level, with the highest priority devices being serviced first. When an interrupt request is received, the processor checks the priority level of the device and services the highest priority request first.

The priority technique ensures that the most critical interrupt requests are handled first, improving system responsiveness and reducing the risk of data loss or corruption. However, it may also result in lower priority requests being delayed, which can affect the overall system performance.

Daisy Chaining Technique: The daisy chaining technique is a method of handling interrupt requests in which multiple devices are connected in a chain, with each device passing the interrupt request to the next device in the chain. When an interrupt request is generated by a device, it is sent to the first device in the chain, which checks if the interrupt is for that device. If it is not, the request is passed on to the next device in the chain until it reaches the correct device. Once the correct device is identified, it services the interrupt request and sends an acknowledgement signal back to the previous device in the chain, which then passes on the acknowledgement to the next device in the chain.

The daisy chaining technique ensures that interrupt requests are handled in a specific order, with the first device in the chain being serviced first. It is a simple and efficient method of handling interrupt requests and is commonly used in systems with a large number of devices. However, it can be slow and may result in delays in interrupt handling, particularly if the chain is long.

Priority and daisy chaining techniques are used in computer memory to manage interrupt requests from multiple devices. The priority technique ensures that the most critical interrupt requests are handled first, while the daisy chaining technique is a simple and efficient method of handling interrupt requests in systems with a large number of devices. The choice of technique depends on

the specific system requirements and the number and priority levels of the devices.

CHECK YOUR PROGRESS -2

4. What is interrupt-driven I/O?

……
……

5. What are the advantages of interrupt-driven I/O?

……
……

6. What are priority and daisy chaining techniques used for?

……
……

9.8 LET US SUM UP

- Memory read and write operations refer to the process of accessing and manipulating data in computer memory. These operations are critical to the functioning of computers and are used in all types of computing devices.
- The process of memory read operations is initiated by the processor by sending an address to the memory controller, which retrieves the data from the specified memory location and transfers it to the program. The process of memory write operations is initiated by the processor by sending an address and the new data to the memory controller, which stores the new value at the specified memory location.
- Handshaking refers to the process of establishing communication between the memory controller and the memory device. It involves a series of signals that are exchanged between the two devices to ensure that data is transmitted accurately and reliably.

- Polling is a technique used in computer memory to manage data transfer between the memory controller and the memory device. It involves the memory controller periodically checking the status of the memory device to determine if it is ready to transfer data. There are two primary polling techniques used in computer memory: serial polling and hub polling.

- Interrupt-driven I/O is an alternative to polling, where the memory device interrupts the processor when it is ready to transfer data. This can improve the efficiency of data transfer and reduce the load on the processor.

- Memory mapping is a technique used to address memory in a computer system, where each memory location is assigned a unique address. This can be used to map memory to specific devices or to provide access to memory for certain programs.

- Priority and daisy chaining techniques can be used to manage multiple devices that need to access memory simultaneously, by assigning different priorities to each device and allowing them to access memory in a specific sequence.

9.10 FURTHER READINGS

1. Tanenbaum, A. S., & Woodhull, A. S. (2015). Operating systems: Design and implementation (3rd ed.). Pearson.

2. Hennessy, J. L., & Patterson, D. A. (2017). Computer architecture: A quantitative approach (6th ed.). Morgan Kaufmann.

3. Bryant, R. E., & O'Hallaron, D. R. (2015). Computer systems: A programmer's perspective (3rd ed.). Pearson.

9.11 ANSWERS TO CHECK YOUR PROGRESS

Answer to Q. No 1: Memory read operation is the process of retrieving data from computer memory.

Answer to Q. No 2: Handshaking is essential in computer memory to ensure accurate and reliable data transfer.

Answer to Q. No 3: Hub polling is a polling technique in which a hub device is used to manage the polling of multiple memory devices.

Answer to Q. No 4: Interrupt-driven I/O is a technique used in computer memory to manage input/output operations using interrupts.

Answer to Q. No 5: Interrupt-driven I/O allows for efficient, flexible, and reliable I/O operations, and allows the processor to handle multiple I/O requests simultaneously, improving system performance.

Answer to Q. No 6: Priority and daisy chaining techniques are used to manage interrupt requests from multiple devices in computer memory, ensuring timely and efficient handling of the requests.

9.12 POSSIBLE QUESTIONS

1. Describe the process of memory read operations in detail. What is the role of the processor and the memory controller in this process? How is the retrieved data temporarily stored in the computer system?
2. Explain the concept of handshaking in computer memory. Why is it essential for the memory controller and the memory device to be synchronized during data transfer? What are the primary signals involved in the handshaking process?
3. Compare and contrast the two primary polling techniques used in computer memory: Serial polling and Hub polling. What are the advantages and disadvantages of each technique? In which type of systems is each technique most suitable?
4. How do memory write operations work? What is the role of the processor and the memory controller in this process? What happens after the new data is stored in memory? How are memory write operations critical to the functioning of modern computers?

5. Describe the interrupt-driven I/O technique in computer memory. How does it work, and what are the advantages of using this technique? How does it compare to the polling technique?
6. Explain the priority and daisy chaining technique used in computer memory. What is the purpose of assigning priorities to different devices? How does daisy chaining help to simplify the communication between devices in a computer system?