UNIT 8: INTRODUCTION TO CACHE AND VIRTUAL MEMORY

UNIT STRUCTURE

8.1 Learning Objective
8.2 Introduction
8.3 Memory Hierarchy
8.4 Semiconductor memories
8.5 Internal organization of typical RAM and ROM Memory
  8.5.1 Internal Organisation of RAM
  8.5.2 Internal Organisation of ROM
8.6 Switches
8.7 Cache memory
8.8 Mapping functions
8.9 Virtual memory
8.10 Locality of reference
8.11 Paging
8.12 Cache Coherence Problem
8.13 Let Us Sum Up
8.14 Answers to Check Your Progress
8.15 Further Reading

8.1 LEARNING OBJECTIVE

After going through this unit a learner will be able to

- Identify the different levels of the memory hierarchy and describe their characteristics.
- Identify and explain the different types of semiconductor memories.
- Describe the internal organization of a typical RAM and ROM memory.
- Identify the types of cache memory
- Explain the concept of virtual memory and its advantages.
- Describe the two types of locality of reference and their importance in computer performance optimization.
- Define paging and explain how it is used to manage memory allocation in modern operating systems.

8.2 INTRODUCTION

Cache and virtual memory are two essential components of modern computer architecture. Cache memory is a small and fast type of memory that stores frequently used data, allowing the CPU to access it quickly. Virtual memory, on the other hand, is a technique that allows a computer to use more memory than it physically has by temporarily transferring data from RAM to a hard disk. In this introduction, we will provide an overview of cache and virtual memory, their benefits, and how they work together to improve the performance of a computer system.

In this unit we will first introduce you to the type of memories and about semiconductor memories. Then we will discuss about organisation of RAM and ROM, switches and cache memory. At the end of the unit you will be introduced to the concept of mapping function, virtual memory, locality of reference, paging and cache coherence problem

8.3 MEMORY HIERARCHY

Memory Hierarchy describes the order in which a computer system's storage devices are used to store data and instructions for the CPU to carry out. It is created to achieve an equilibrium between capacity, cost, and speed. Based on the access time, capacity, and cost of each level, the memory hierarchy is divided into different categories. Registers, cache memory, main memory, and secondary storage are some of these categories.

- • **Registers:** The CPU stores data and instructions in registers, the smallest and fastest type of memory. The processor has immediate access to them because they are situated on the processor chip. Registers are used to store both the instructions that the processor needs to execute and the data that it is currently processing, such as operands and addresses. Among all the memory types in the hierarchy, registers have the least amount of storage space.
- **Cache Memory:** Cache memory is a small amount of fast memory that is located between the processor and main memory. It is designed to

reduce the average access time of the processor to data stored in main memory. Cache memory holds the most frequently used data and instructions. When the processor needs data, it first checks the cache memory. If the data is not found in cache memory, it is fetched from the main memory and stored in cache memory for future use.

Cache memory is divided into two types: Level 1 (L1) and Level 2 (L2). L1 cache memory is the closest to the processor, while L2 cache memory is located farther away from the processor. L1 cache memory has a smaller capacity and is faster than L2 cache memory.

- **Main Memory:** Main memory, also known as Random Access Memory (RAM), is the memory used by the computer to store data and instructions that are currently being used by the processor. It is a volatile memory, which means that its contents are lost when the power is turned off. Main memory is slower than cache memory but has a larger capacity.
- **Secondary Storage:** Secondary storage refers to the long-term storage devices used by the computer, such as hard disk drives, solid-state drives, and optical drives. Secondary storage is used to store data and instructions that are not currently being used by the processor. Unlike main memory, secondary storage is non-volatile, which means that its contents are retained even when the power is turned off. Secondary storage has the largest capacity among all the memory types in the hierarchy but is slower than main memory.

In summary, the memory hierarchy is designed to provide a balance between speed, cost, and capacity. The processor uses registers and cache memory to store data and instructions that are currently being processed, while main memory and secondary storage are used to store data and instructions that are not currently being used. The hierarchy allows the processor to access data quickly by first checking the smallest and fastest memory types before accessing larger and slower ones.

8.4 SEMICONDUCTOR MEMORIES

Semiconductor memories are a type of electronic memory that uses semiconductor technology to store and retrieve digital information. They are widely used in modern computing devices such as smartphones, laptops, and

servers due to their high-speed access and low power consumption. Semiconductor memories can be categorized into two types: volatile and non-volatile.

1. Volatile Semiconductor Memories: Volatile semiconductor memories lose their data when the power is turned off. The two most common types of volatile semiconductor memories are Static Random Access Memory (SRAM) and Dynamic Random Access Memory (DRAM).

- SRAM: SRAM is a type of volatile semiconductor memory that uses flip-flop circuits to store data. It is faster and consumes less power than DRAM. SRAM is used for cache memory in processors and other high-speed memory applications.

- DRAM: DRAM is a type of volatile semiconductor memory that uses capacitors to store data. It is slower and consumes more power than SRAM. DRAM is used as main memory in computers and other devices that require large amounts of memory.

2. Non-Volatile Semiconductor Memories: Non-volatile semiconductor memories retain their data even when the power is turned off. The most common types of non-volatile semiconductor memories are Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Erasable Programmable Read-Only Memory (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), and Flash Memory.

- ROM: ROM is a type of non-volatile semiconductor memory that is pre-programmed with data during manufacturing. It is used to store permanent data such as firmware, boot loaders, and operating system.

- PROM: PROM is a type of non-volatile semiconductor memory that can be programmed once after manufacturing. It is used in applications where the data needs to be programmed only once, such as calibration data, product identification, and serial numbers.

- EPROM: EPROM is a type of non-volatile semiconductor memory that can be erased and reprogrammed multiple times. It is used in applications where the data needs to be changed frequently, such as firmware updates and configuration data.

- EEPROM: EEPROM is a type of non-volatile semiconductor memory that can be erased and reprogrammed electrically. It is used in applications where the data needs to be changed frequently, but the erasing and reprogramming process needs to be fast and easy.
- Flash Memory: Flash memory is a type of non-volatile semiconductor memory that is widely used in portable devices such as smartphones, digital cameras, and USB drives. It can be erased and reprogrammed in blocks, making it faster and more efficient than EEPROM.

In summary, semiconductor memories are a type of electronic memory that uses semiconductor technology to store and retrieve digital information. They can be categorized into two types: volatile and non-volatile. Volatile semiconductor memories lose their data when the power is turned off, while non-volatile semiconductor memories retain their data even when the power is turned off. Semiconductor memories are widely used in modern computing devices due to their high-speed access and low power consumption.

8.5 INTERNAL ORGANIZATION OF TYPICAL RAM AND ROM MEMORY

Internal organization of typical RAM and ROM memory is an important aspect of computer memory architecture. RAM or Random-Access Memory is a type of volatile semiconductor memory that allows data to be read or written in any order, making it ideal for temporary storage of data during program execution. The internal organization of a typical RAM includes memory cells, word lines, bit lines, and address decoders that work together to provide high-speed access to data. On the other hand, a Read-Only Memory or ROM is a type of non-volatile semiconductor memory that is pre-programmed with data during manufacturing. The internal organization of a typical ROM includes memory cells, address decoders, and word lines that work together to provide permanent storage of data.

8.5.1 Internal Organisation of RAM

Random Access Memory (RAM) is a type of volatile semiconductor memory that is commonly used as the main memory in computers and other digital devices. The internal organization of a typical RAM can vary

depending on the specific type and manufacturer, but there are some common elements that most RAMs share.

1. Memory Cells: The basic unit of a RAM is a memory cell, which is responsible for storing one bit of information. A memory cell typically consists of a transistor and a capacitor. The transistor acts as a switch, allowing the capacitor to either hold a charge (representing a binary 1) or discharge (representing a binary 0).
2. Address Decoder: In order to access a specific memory location in a RAM, the device needs to send an address signal to the memory. The address decoder is responsible for decoding this signal and selecting the appropriate memory cells for reading or writing.
3. Sense Amplifiers: After a memory cell has been accessed, the sense amplifiers are responsible for amplifying the signal from the memory cell to a level that can be read by the device's input/output (I/O) system.
4. Control Logic: The control logic in a RAM is responsible for managing the various operations that the memory can perform, such as reading, writing, refreshing, and activating the different memory cells.
5. Data Input/Output: The data input/output (I/O) lines of a RAM are responsible for transferring data between the memory and the device's processor or other I/O devices. These lines are typically arranged in groups of eight (for byte-level access) or sixteen (for word-level access).
6. Refresh Circuitry: RAM is a volatile memory, which means that the data it stores will be lost if the power supply is turned off. To prevent this, a refresh circuitry is included in the RAM that continuously refreshes the data stored in the memory cells, typically every few milliseconds.

In summary, the internal organization of a typical RAM includes memory cells, an address decoder, sense amplifiers, control logic, data input/output, and refresh circuitry. The specific implementation of each of these components may vary depending on the type and manufacturer of the RAM.

8.5.2 Internal Organisation of ROM

A Read-Only Memory (ROM) is a type of non-volatile semiconductor memory that is pre-programmed with data during manufacturing. The internal organization of a typical ROM is designed to store data permanently, and it is different from other types of semiconductor memories such as Random-Access Memory (RAM) or Flash Memory. Here are the key components of a typical ROM:

1. Address Decoder: The Address Decoder is responsible for decoding the address signals received from the memory controller and selecting the appropriate memory cell or word line. It uses a combination of logic gates and transistors to generate the select signals.

2. Word Lines: Word Lines are a set of conductive lines that connect the memory cells in a ROM. Each word line represents a unique memory address, and it is activated by the Address Decoder when the memory controller sends the appropriate address signal.

3. Memory Cells: Memory Cells are the fundamental building blocks of a ROM. They are responsible for storing the data permanently, and they are typically composed of a transistor and a fuse. The state of the memory cell is determined by the presence or absence of a fuse.

4. Data Output Circuit: The Data Output Circuit is responsible for amplifying the signal from the selected memory cell and sending it to the memory controller as a digital output. It uses a combination of amplifiers and logic gates to ensure that the signal is stable and accurate.

CHECK YOUR PROGRESS -1

1. What is the memory hierarchy?

……

……

……

2. What are the two types of semiconductor memories?

……

……

……

3. What is the difference between SRAM and DRAM?

……

……

……

The internal organization of a typical ROM is designed to provide fast and reliable access to pre-programmed data. The Address Decoder selects the appropriate word line based on the memory controller's address signal, and the Memory Cells store the data permanently using fuses or other non-volatile elements. The Data Output Circuit ensures that the digital output signal is stable and accurate, providing reliable data access to the memory controller.

8.6 SWITCHES

Switches play an important role in the operation of electronic memory devices such as RAM and ROM. They are used to control the flow of electricity through the circuits, allowing data to be stored, retrieved, and processed. There are two main types of switches used in electronic memory devices: transistors and fuses.

1. Transistors: Transistors are electronic devices that can act as switches or amplifiers. They are commonly used in RAM to store and retrieve data. In a RAM cell, a transistor acts as a switch that controls the flow of electricity between the memory cell and the bit line. When the transistor is turned on, electricity flows from the bit line into the memory

cell, and the data stored in the cell can be read. When the transistor is turned off, the data is retained in the memory cell until it is overwritten.

2. Fuses: Fuses are electrical components that are designed to burn out when a certain amount of current flows through them. They are commonly used in ROM to store permanent data. In a ROM cell, a fuse acts as a switch that controls the presence or absence of a current path between the memory cell and the word line. When the fuse is intact, electricity flows from the word line into the memory cell, and the data stored in the cell can be read. When the fuse is burned out, the memory cell is permanently disabled, and the corresponding data cannot be read.

3. Address Decoders: Address decoders are electronic circuits that are used to select a specific memory cell or word line in the memory array. In RAM, the address decoder receives the memory controller's address signals and activates the appropriate word line and bit line. In ROM, the address decoder receives the memory controller's address signals and activates the appropriate word line, enabling the corresponding fuse to be read or written.

In summary, switches are an important component of electronic memory devices such as RAM and ROM. Transistors are used in RAM to control the flow of electricity between the memory cell and the bit line, while fuses are used in ROM to control the presence or absence of a current path between the memory cell and the word line. Address decoders are used in both RAM and ROM to select a specific memory cell or word line in the memory array. By using switches and address decoders, electronic memory devices can provide fast and reliable access to data, making them an essential component of modern computing systems.

8.7 CACHE MEMORY

Cache memory is a small and fast type of computer memory that is used to temporarily store frequently accessed data or instructions. It is an essential component of modern computer architecture, providing faster access to data and reducing the overall latency of computer systems. In this article, we will discuss cache memory in detail, including its types, organization, and benefits.

Types of Cache Memory: There are mainly two types of cache memory: Level 1 (L1) and Level 2 (L2). L1 cache memory is a small, high-speed memory that is located inside the CPU, and L2 cache memory is a larger, slower memory that is located outside the CPU.

L1 Cache Memory: L1 cache memory is the primary cache memory in a computer system, and it is typically the fastest and smallest memory in the system. It is integrated directly into the CPU, allowing for faster access to data and instructions. The L1 cache memory is divided into two parts: instruction cache and data cache. The instruction cache stores frequently used instructions, and the data cache stores frequently accessed data.

L2 Cache Memory: L2 cache memory is a larger and slower cache memory that is located outside the CPU. It is typically located on the motherboard or integrated into the CPU package. L2 cache memory is used to store data that is not frequently accessed, or data that is too large to fit in the L1 cache memory. It is usually larger than L1 cache memory, but slower in terms of access time.

Organization of Cache Memory: Cache memory is organized into lines or blocks, each containing multiple bytes of data or instructions. When the CPU needs to access data or instructions, it first checks the L1 cache memory. If the required data or instruction is not present in the L1 cache, the CPU checks the L2 cache memory. If the data or instruction is still not found, the CPU accesses the main memory. When data or instructions are read from the main memory, they are copied into the cache memory to improve the overall performance of the system.

Benefits of Cache Memory: The use of cache memory provides several benefits, including:

1. Faster Access to Data: Cache memory provides faster access to frequently accessed data or instructions, reducing the overall latency of the system.
2. Reduced Memory Accesses: By storing frequently accessed data or instructions in the cache memory, the number of accesses to the main memory is reduced, improving the overall performance of the system.
3. Improved System Performance: Cache memory improves the overall performance of the system by reducing the latency of data accesses and reducing the number of accesses to the main memory.

In summary, cache memory is a small and fast type of computer memory that is used to temporarily store frequently accessed data or instructions. It is an essential component of modern computer architecture, providing faster access to data and reducing the overall latency of computer systems. By using cache memory, computer systems can achieve higher performance and better responsiveness, making it an important component of modern computing.

8.8 MAPPING FUNCTION

Mapping functions are an essential aspect of computer memory management. They determine how the data is stored and accessed in memory, and they affect the performance of the entire system. In this article, we will discuss the different types of mapping functions used in computer memory, including direct mapping, fully associative mapping, and set-associative mapping.

1. Direct Mapping: Direct mapping is the simplest type of mapping function used in computer memory. In direct mapping, each block of memory is mapped to a unique location in the cache memory. The mapping is done using a simple formula that calculates the index of the cache location based on the memory address of the data.

Direct mapping is efficient in terms of hardware implementation, but it can lead to conflicts in the cache memory. Conflicts occur when two blocks of memory are mapped to the same cache location, causing one block to be evicted from the cache memory. Direct mapping is typically used in L1 cache memory, where the cache size is small, and conflicts are less likely to occur.

2. Fully Associative Mapping: Fully associative mapping is the opposite of direct mapping, where each block of memory can be stored in any cache location. In fully associative mapping, the cache memory is searched for a particular block of memory, regardless of its location.

Fully associative mapping provides the highest hit rate, as there is no limit to where a particular block of memory can be stored in the cache memory. However, it requires more hardware resources and is more expensive to implement than direct mapping.

Fully associative mapping is typically used in L2 cache memory, where the cache size is larger, and conflicts are more likely to occur.

3. Set-Associative Mapping: Set-associative mapping is a compromise between direct mapping and fully associative mapping. In set-associative mapping, the cache memory is divided into sets, and each set contains a small number of cache locations. Each block of memory is mapped to one of the cache locations in the set.

Set-associative mapping reduces the likelihood of conflicts in the cache memory, as each set has its own set of cache locations. It is less expensive to implement than fully associative mapping, but more flexible than direct mapping.

Set-associative mapping is typically used in L3 cache memory, where the cache size is even larger than L2 cache memory, and conflicts are more likely to occur.

In summary, mapping functions are an essential aspect of computer memory management. They determine how the data is stored and accessed in memory, and they affect the performance of the entire system. There are three types of mapping functions used in computer memory: direct mapping, fully associative mapping, and set-associative mapping. Each has its advantages and disadvantages, and they are typically used in different levels of cache memory.

8.9 VIRTUAL MEMORY

Virtual memory is a technique used by modern operating systems to increase the amount of memory available to running programs. It provides a way for programs to use more memory than is physically available in a computer's RAM by temporarily transferring data to and from the hard disk. This allows programs to run as if they have more memory than they actually do.

The concept of virtual memory is based on the idea of paging. In paging, the computer's RAM is divided into fixed-size blocks called pages. When a program requests memory, the operating system assigns it one or more pages. If the requested page is not currently in RAM, the operating system retrieves it from the hard disk and stores it in an available page. The operating system keeps track of which pages are in RAM and which are on the hard disk. When a program accesses a page that is not currently in RAM, a page fault occurs, and the operating system retrieves the page from the hard disk and puts it into an available page in RAM.

The main advantage of virtual memory is that it allows programs to use more memory than is physically available in the computer's RAM. This can be particularly useful for large programs that require a lot of memory to run. Virtual memory also makes it possible to run multiple programs simultaneously, since each program can be assigned its own set of pages in RAM.

Another advantage of virtual memory is that it allows the operating system to protect memory from unauthorized access. Each program is assigned its own set of pages in RAM, and the operating system can prevent one program from accessing the pages assigned to another program. This can help prevent bugs and malicious programs from causing damage to the computer.

However, virtual memory also has some disadvantages. The main disadvantage is that accessing data stored in virtual memory is slower than accessing data stored in RAM. This is because data must be transferred between RAM and the hard disk, which is much slower than accessing data stored in RAM directly. This can cause programs to run more slowly when they are accessing data stored in virtual memory.

Overall, virtual memory is a useful technique that allows programs to use more memory than is physically available in the computer's RAM. While it has some disadvantages, the benefits of virtual memory generally outweigh the drawbacks, and it is an important component of modern operating systems.

CHECK YOUR PROGRESS -2

4. What are the two main types of switches used in electronic memory devices?
5. What is cache memory and what are its benefits?
6. What are the three types of mapping functions used in computer memory?

8.10 LOCALITY OF REFERENCE

Locality of reference is a fundamental principle in computer science that refers to the tendency of programs to access the same memory locations repeatedly, rather than accessing random locations. It is a property of memory access patterns that can be observed in many types of programs, from simple algorithms to complex applications.

The concept of locality of reference is important for understanding how computer memory is used and how performance can be optimized. There are two main types of locality of reference: temporal and spatial locality.

Temporal locality refers to the tendency of programs to access the same memory locations repeatedly over a short period of time. For example, a program that processes data in a loop may access the same data multiple times during each iteration of the loop. By keeping this data in the cache or in registers, the program can avoid the overhead of repeatedly accessing it from main memory.

Spatial locality refers to the tendency of programs to access memory locations that are near each other in memory. For example, a program that processes an array of data may access adjacent elements of the array sequentially. By keeping these elements in the cache or in registers, the program can take advantage of the faster access times and reduce the number of cache misses.

There are several techniques that can be used to take advantage of the principle of locality of reference. Caching is one such technique that is widely used in computer systems. Caches are small, fast memory devices that store recently accessed data and instructions. By keeping frequently accessed data and instructions in the cache, the computer can reduce the number of memory accesses and improve performance.

Another technique that can be used to take advantage of locality of reference is prefetching. Prefetching involves predicting which memory locations are likely to be accessed in the near future and fetching them into the cache before they are actually needed. This can reduce the latency of memory accesses and improve performance.

In summary, locality of reference is a fundamental principle that plays a critical role in computer system design and performance optimization. By understanding and exploiting the temporal and spatial locality of memory

access patterns, computer systems can achieve better performance and efficiency.

8.11 PAGING

Paging is a memory management technique used by modern operating systems to manage the allocation of physical memory to running programs. It allows the operating system to divide the physical memory into small fixed-size blocks called pages, and to allocate pages to running programs as needed. Paging is used in combination with virtual memory to increase the amount of memory available to programs.

In paging, the physical memory is divided into fixed-size blocks called frames, and the virtual memory is divided into fixed-size blocks called pages. Each program is allocated a set of pages in virtual memory, and these pages are mapped to frames in physical memory by the operating system. The mapping between virtual pages and physical frames is maintained in a page table, which is used by the operating system to translate memory addresses used by the program into physical memory addresses.

When a program accesses a page that is not currently in physical memory, a page fault occurs. The operating system retrieves the required page from disk and stores it in a free frame in physical memory. If there are no free frames, the operating system selects a frame to evict, typically using a page replacement algorithm, to make space for the required page. The evicted page is written to disk if it has been modified since it was last read from disk.

Paging has several advantages over other memory management techniques. First, it allows the operating system to allocate physical memory to running programs on demand, rather than requiring that all memory be allocated when the program is started. This allows programs to use more memory than is physically available, and can improve the performance of the system by reducing the amount of swapping required.

Second, paging allows the operating system to protect memory from unauthorized access. Each program is allocated its own set of pages in virtual memory, and the operating system can prevent one program from accessing the pages allocated to another program. This can help prevent bugs and malicious programs from causing damage to the system.

Finally, paging allows the operating system to optimize memory usage by using a page replacement algorithm to select the most appropriate page to evict when physical memory is full. There are many page replacement algorithms that can be used, including least recently used (LRU), least frequently used (LFU), and random replacement. The choice of algorithm can have a significant impact on the performance of the system.

However, paging also has some disadvantages. First, the overhead of managing the page table and handling page faults can reduce system performance. Second, the need to transfer pages between physical memory and disk can result in slower access times compared to using physical memory alone. Finally, the use of paging can result in fragmentation of physical memory, which can reduce the amount of available memory.

In summary, paging is a memory management technique used by modern operating systems to manage the allocation of physical memory to running programs. It allows programs to use more memory than is physically available, provides memory protection, and optimizes memory usage. However, it also has some overhead and can result in slower access times and fragmentation of physical memory.

8.12 CACHE COHERENCE PROBLEM

The cache coherence problem is a fundamental challenge in computer architecture that arises when multiple processors or cores in a system have their own private caches, which can lead to inconsistencies in the shared memory data between them. This problem arises due to the fact that each processor or core may have its own copy of a particular data block, and if any processor updates its copy of the data, the other processors may not be aware of the update.

In a multiprocessor system, the caches are typically organized in a hierarchy, with each processor having its own private cache, which is closer to the processor than the main memory. This cache hierarchy is used to improve the performance of the system by reducing the latency of memory accesses. However, this also leads to the cache coherence problem.

There are several techniques that can be used to maintain cache coherence in a multiprocessor system. The most common technique is called the snooping protocol. In this protocol, each cache controller monitors the bus to determine

whether any other processor is accessing a shared memory location. If a processor writes to a shared memory location, the cache controller broadcasts an invalidation message to all other processors that may have a copy of the data block. This forces them to invalidate their copy of the data block and to fetch the latest copy from main memory or from the processor that made the update.

Another technique that can be used to maintain cache coherence is called the directory protocol. In this protocol, each cache controller maintains a directory that keeps track of which processors have a copy of each data block. When a processor updates a data block, it sends a message to the directory, which updates the status of the data block in the directory and sends a message to all other processors that have a copy of the data block to invalidate their copy. When a processor needs to read a data block, it sends a message to the directory, which determines which processor has the latest copy of the data block and sends it back to the requesting processor.

Cache coherence is important because it ensures that all processors in a multiprocessor system have a consistent view of the shared memory data. Inconsistencies in the shared memory data can lead to incorrect program behaviour, such as data corruption or program crashes. Therefore, it is essential that the cache coherence problem be addressed in any multiprocessor system.

In summary, the cache coherence problem is a fundamental challenge in computer architecture that arises when multiple processors or cores in a system have their own private caches. Techniques such as the snooping protocol and the directory protocol can be used to maintain cache coherence and ensure that all processors in a multiprocessor system have a consistent view of the shared memory data.

CHECK YOUR PROGRESS- 3

7. What is locality of reference?

……
……
……

8. What is paging?

.....
.....
.....

9. What is the cache coherence problem?

……
……
……

8.13 LET US SUM UP

- The memory hierarchy in computer systems is designed to achieve a balance between speed, capacity, and cost. It consists of four levels, namely registers, cache memory, main memory, and secondary storage.
- Semiconductor memories are a type of electronic memory widely used in modern computing devices such as smartphones, laptops, and servers. They can be categorized into volatile and non-volatile types. Volatile semiconductor memories lose their data when power is turned off, while non-volatile semiconductor memories retain their data even when the power is turned off.
- The most common types of semiconductor memories include SRAM, DRAM, ROM, PROM, EPROM, EEPROM, and Flash Memory. They have different characteristics in terms of speed, power consumption, and reprogrammability, which make them suitable for different applications in computer systems and other electronic devices.

- RAM and ROM are two types of semiconductor memory used in computers and other digital devices. RAM is volatile and allows data to be read or written in any order, while ROM is non-volatile and pre-programmed with data during manufacturing.
- The internal organization of a typical RAM includes memory cells, an address decoder, sense amplifiers, control logic, data input/output, and refresh circuitry. The specific implementation of each of these components may vary depending on the type and manufacturer of the RAM. The internal organization of a typical ROM includes address decoders, word lines, memory cells, and data output circuitry.
- Switches play an important role in the operation of electronic memory devices such as RAM and ROM. Transistors and fuses are the two main types of switches used in electronic memory devices, with transistors used in RAM to store and retrieve data and fuses used in ROM to permanently store pre-programmed data.
- Cache memory is a small and fast type of computer memory that stores frequently accessed data or instructions, providing faster access to data and reducing the overall latency of computer systems. It is organized into lines or blocks, each containing multiple bytes of data or instructions.
- Mapping functions are an essential aspect of computer memory management that determine how data is stored and accessed in memory. The three types of mapping functions are direct mapping, fully associative mapping, and set-associative mapping. Each has its advantages and disadvantages, and they are typically used in different levels of cache memory.
- Virtual memory is a technique used by modern operating systems to increase the amount of usable memory available to a computer system. It uses a combination of RAM and hard drive space to simulate additional RAM, allowing programs to run even if they require more memory than physically available. Virtual memory can improve system performance but may also lead to performance issues if not managed properly.

- Locality of reference is a principle in computer science that refers to the tendency of programs to access the same memory locations repeatedly, which can be exploited to improve computer performance through techniques such as caching and prefetching.
- Paging is a memory management technique used by modern operating systems that allows the allocation of physical memory to running programs on demand, increases the amount of memory available to programs, and protects memory from unauthorized access. However, it also has some overhead and can result in slower access times and fragmentation of physical memory.
- The cache coherence problem arises when multiple processors or cores in a system have their own private caches, which can lead to inconsistencies in the shared memory data between them. Techniques such as snooping, directory-based coherence, and cache-invalidation protocols are used to solve this problem.

8.14 ANSWERS TO CHECK YOUR PROGRESS

Answer to Q. No 1: The organization of a computer's storage devices based on speed, capacity, and cost

Answer to Q. No 2: Volatile and non-volatile.

Answer to Q, No 3: SRAM is faster and consumes less power than DRAM, but DRAM has a larger capacity and is used as main memory in computers.

Answer to Q. No 4: Transistors and fuses.

Answer to Q. No 5: Cache memory is a small and fast type of memory used to temporarily store frequently accessed data or instructions, providing faster access to data and reducing the overall latency of computer systems.

Answer to Q. No 6: Direct mapping, fully associative mapping, and set-associative mapping.

Answer to Q. No 7: Locality of reference is a principle in computer science that refers to the tendency of programs to access the same memory locations repeatedly.

Answer to Q. No 8: Paging is a memory management technique used by modern operating systems to manage the allocation of physical memory to running programs.

Answer to Q. No 9: The cache coherence problem is a fundamental challenge in computer architecture that arises when multiple processors or cores in a system have their own private caches, which can lead to inconsistencies in the shared memory data between them.

8.15 FURTHER READINGS

1. K. Hwang and F.A. Briggs. (2018). Computer Architecture and Parallel Processing. Tata McGraw-Hill Education.
2. V. Heuring and H. Jordan. (2017). Computer Systems Design and Architecture. Pearson Education India.

8.16 POSSIBLE QUESTIONS

1. Explain the concept of the memory hierarchy and its significance in modern computing devices. How does the memory hierarchy achieve a balance between speed, cost, and capacity?
2. Describe the two types of semiconductor memories and their respective features.
3. Explain the role of switches in electronic memory devices such as RAM and ROM.
4. What is cache memory, and how does it work to improve computer performance?
5. Explain in detail how locality of reference is used in computer systems to improve performance. Give examples of techniques that take advantage of temporal and spatial locality, and describe how they work.
6. Explain in detail how a cache works in a computer system.