

COMPUTER ORGANIZATION&ARCHITECHTURE (COA)

MODULE-3

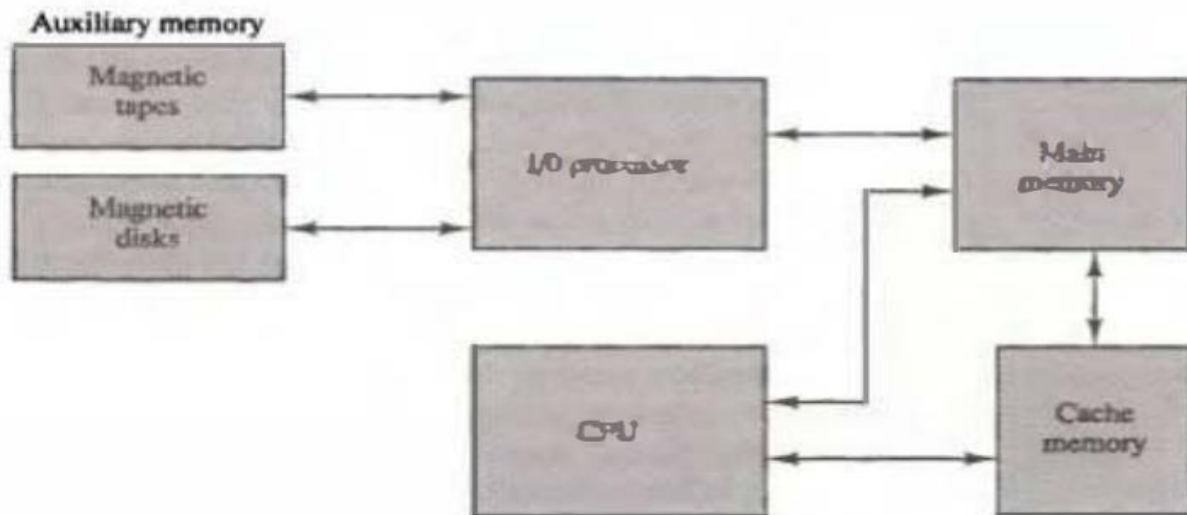
MEMORY ORGANIZATION

MEMORY HIERARCHY

The memory unit is an essential component in any digital computer since It Is needed for store programs and data. A very small computer with a limited application may be able to fulfill its intended task without the need of additional storage capacity. The memory unit that communicates directly with the called CPU is called the **main memory**. Devices that provide backup storage are auxiliary memory. The most common **auxiliary memory** devices used in computer systems are magnetic disks and tapes. They are used for storing system Programs, large data files, and other backup information. Only programs and data currently needed by the processor reside in main memory. The main memory occupies a central position by being able to communicate directly with the CPU and with auxiliary memory devices through an I/O processor. When programs not residing in main memory are needed by the CPU, they are brought in from auxiliary memory. Programs not currently needed in main memory are transferred into auxiliary memory to provide space for currently used programs and data.

A special very-high-speed memory called a **Cache memory** is sometimes used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate. The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor logic.

Figure 12-1 Memory hierarchy in a computer system.



Main Memory

The main memory is the central storage unit in a computer system. It is a relatively large and fast memory used to store programs and data during the computer operation. The principal technology used for the main memory is based on semiconductor integrated circuits. Integrated circuit RAM chips are available in two possible operating modes, static and dynamic. The **static RAM** consists essentially of internal flip-flops that store the binary information. The stored information remains valid as long as power is applied to the unit. The **dynamic RAM** stores the binary information in the form of electric charges that are applied to capacitors. The dynamic RAM offers reduced power consumption and larger storage capacity in a single memory chip. The static RAM is easier to use and has shorter read and write cycles. Most of the main memory in a general-purpose computer is made up of RAM integrated circuit chips, but a portion of the memory may be constructed with ROM chips. Originally, RAM was used to refer to a

random-access memory, but now it is used to designate a read/write memory to distinguish it from a read-only memory, although ROM is also random access. RAM is used for storing the bulk of the programs and data that are subject to change. ROM is used for storing programs that are permanently resident in the computer and for tables of constants that do not change in value once the production of the computer is completed.

The ROM portion of main memory is needed for storing an initial program called a **bootstrap loader**. The bootstrap loader is a program whose function is to start the computer software operating when power is turned on. Since RAM is volatile, its contents are destroyed when power is turned off. The contents of ROM remain unchanged after power is turned off and on again. The startup of a computer consists of turning the power on and starting the execution of an initial program.

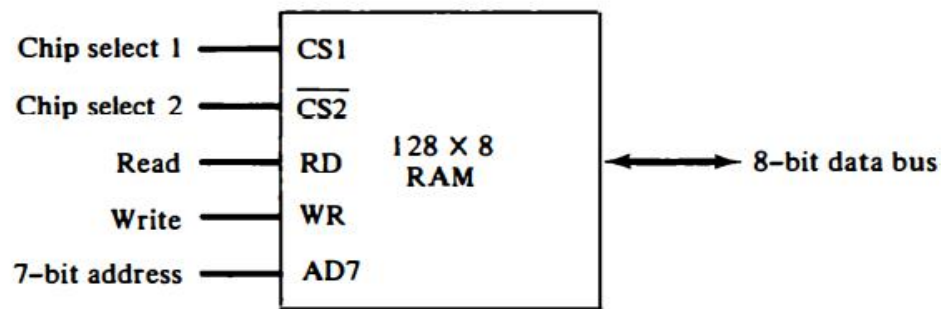
Types of Read Only Memory (ROM)

1. PROM (Programmable read-only memory) – It can be programmed by the user. Once programmed, the data and instructions in it cannot be changed.
2. EPROM (Erasable Programmable read only memory) – It can be reprogrammed. To erase data from it, expose it to ultraviolet light. To reprogram it, erase all the previous data.
3. EEPROM (Electrically erasable programmable read only memory) – The data can be erased by applying an electric field, with no need for ultraviolet light. We can erase only portions of the chip.

RAM and ROM Chips

A **RAM chip** is better suited for communication with the CPU if it has one or more control inputs that select the chip only when needed. Another common feature is a bidirectional data bus that allows the transfer of data either from memory to CPU during a read operation or from CPU to memory during a write operation. A bidirectional bus can be constructed with three-state buffers. A three-state buffer output can be placed in one of three possible states: a signal equivalent to logic 1, a signal equivalent to logic 0, or a highimpedance state. The logic 1 and 0 are normal digital signals. The highimpedance state behaves like an open circuit, which means that the output does not carry a signal and has no logic significance.

The capacity of the memory is 128 words of eight bits (one byte) per word. This requires a 7-bit address and an 8-bit bidirectional data bus. The read and write inputs specifies the memory operation and the two chips select (CS) control inputs are for enabling the chip only when it is selected by the microprocessor. The availability of more than one control input to select the chip facilitates the decoding the address lines when multiple chips are used in the microcomputer. The read and write inputs are sometimes combined into one line labeled R/W. When the chip is selected, the two binary states in this line specify the two operations of read or write.

Figure 12-2 Typical RAM chip.**(a) Block diagram**

CS1	$\overline{\text{CS2}}$	RD	WR	Memory function	State of data bus
0	0	x	x	Inhibit	High-impedance
0	1	x	x	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	x	Read	Output data from RAM
1	1	x	x	Inhibit	High-impedance

(b) Function table

Activate V
Data Edition

A **ROM chip** is organized externally in a similar manner. However, since a ROM can only read, the data bus can only be in an output mode. For the same-size chip, it is possible to have more bits of ROM than of RAM, because the internal binary cells in ROM occupy less space than in RAM. For this reason, the diagram specifies a 512-byte ROM, while the RAM has only 128 bytes. The nine address lines in the ROM chip specify any one of the 512 bytes stored in it. The two chip select inputs must be CS1 = 1 and CS2 = 0 for the unit to operate. Otherwise, the data bus is in a high-impedance state. There is no need for a read or write control because the unit can only read. Thus when the chip is enabled by the two select inputs, the byte selected by the address lines appears on the data bus.

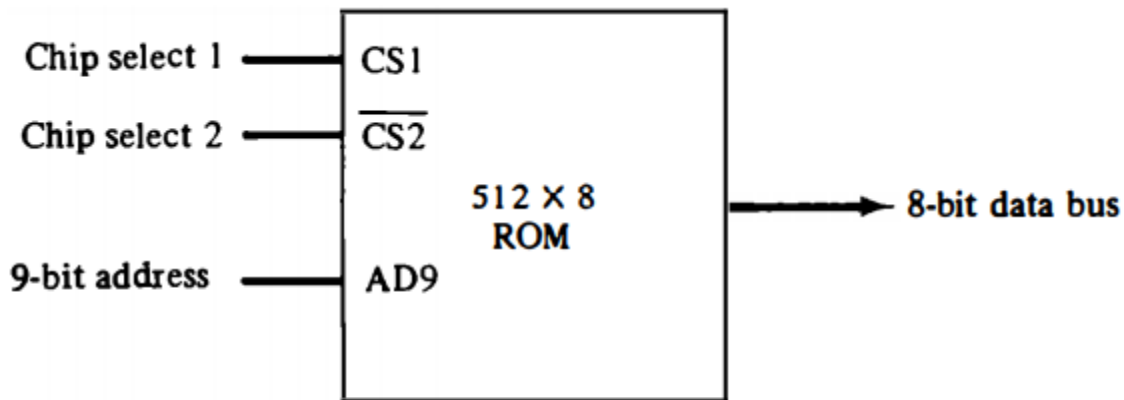


Figure 12-3 Typical ROM chip.

Auxiliary Memory

The most common auxiliary memory devices used in computer systems are magnetic disks and tapes. Other components used, but not as frequently, are magnetic drums, magnetic bubble memory, and optical disks. The important characteristics of any device are its access mode, access time, transfer rate, capacity, and cost. The average time required to reach a storage location in memory and obtain its contents is called the **access time**. In electromechanical devices with moving parts such as disks and tapes, the access time consists of a **seek time** required to position the read-write head to a location and a **transfer time** required to transfer data to or from the device. Because the seek time is usually much longer than the transfer time, auxiliary storage is organized in records or blocks. A **record** is a specified number of characters or words. Reading or writing is always done on entire records. The **transfer rate** is the number of characters or words that the device can transfer per second, after it has been positioned at the beginning of the record. Bits are recorded as magnetic spots on the surface as it passes a stationary mechanism called a **write head**. Stored bits are

detected by a change in magnetic field produced by a recorded spot on the surface as it passes through a **read head**.

Magnetic Disks

A magnetic disk is a circular plate constructed of metal or plastic coated with magnetized material. Often both sides of the disk are used and several disks may be stacked on one spindle with read/write heads available on each surface. All disks rotate together at high speed and are not stopped or started for access purposes. Bits are stored in the magnetized surface in spots along concentric circles called **tracks**. The tracks are commonly divided into sections called **sectors**. In most systems, the minimum quantity of information which can be transferred is a sector.

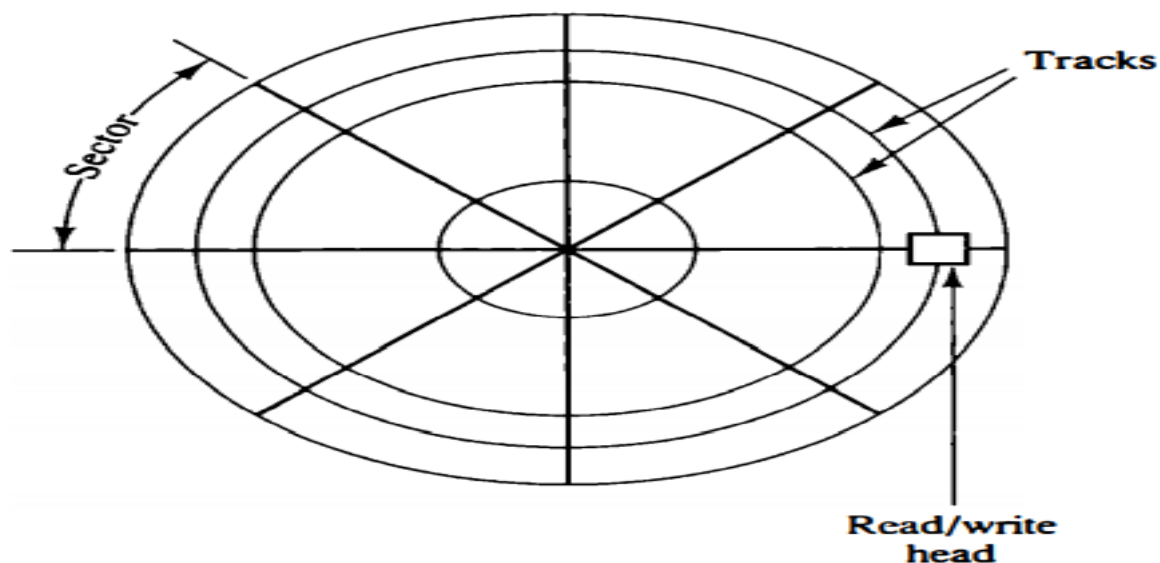


Figure 12-5 Magnetic disk.

A disk system is addressed by address bits that specify the disk number, the disk surface, the sector number and the track within the sector. After the read/write heads are positioned in the specified track, the system has to wait until the rotating disk reaches the specified sector under the read/write head. Information transfer is

very fast once the beginning of a sector has been reached. Disks may have multiple heads and simultaneous transfer of bits from several tracks at the same time.

Disks that are permanently attached to the unit assembly and cannot be removed by the occasional user are called **hard disks**. A disk drive with removable disks is called a **floppy disk**. The disks used with a floppy disk drive are small removable disks made of plastic coated with magnetic recording material. Floppy disks are extensively used in personal computers as a medium for distributing software to computer users.

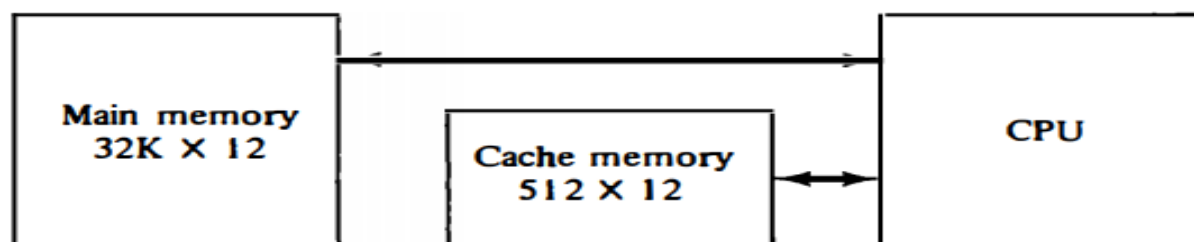
Magnetic Tape

A magnetic tape transport consists of the electrical, mechanical, and electronic components to provide the parts and control mechanism for a magnetic-tape unit. The tape itself is a strip of plastic coated with a magnetic recording medium. Bits are recorded as magnetic spots on the tape along several tracks. Usually, seven or nine bits are recorded simultaneously to form a character together with a parity bit. Read/write heads are mounted one in each track so that data can be recorded and read as a sequence of characters. Magnetic tape units can be stopped, started to move forward or in reverse, or can be rewound. However, they cannot be started or stopped fast enough between individual characters. For this reason, information is recorded in blocks referred to as records.

Cache Memory

Analysis of a large number of typical programs has shown that the references to memory at any given interval of time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of **locality of reference**. If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced, thus reducing the total

execution time of the program. Such a fast small memory is referred to as a cache memory. It is placed between the CPU and main memory. The cache memory access time is less than the access time of main memory by a factor of 5 to 10. The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components. The fundamental idea of cache organization is that by keeping the most frequently accessed instructions and data in the fast cache memory, the average memory access time will approach the access time of the cache. Although the cache is only a small fraction of the size of main memory, a large fraction of memory requests will be found in the fast cache memory because of the locality of reference property of programs. When the CPU needs to access memory, the cache is examined. If the word is found in the cache, it is read from the fast memory. If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word. A block of words containing the one just accessed is then transferred from main memory to cache memory. The block size may vary from one word (the one just accessed) to about 16 words adjacent to the one just accessed. In this manner, some data are transferred to cache so that future references to memory find the required words in the fast cache memory. The performance of cache memory is frequently measured in terms of a quantity called **hit ratio**. When the CPU refers to memory and finds the word in cache, it is said to produce a **hit**. If the word is not found in cache, it is in main memory and it counts as a **miss**. The ratio of the number of hits divided by the total CPU references to memory (hits plus misses) is the **hit ratio**.



The basic characteristic of cache memory is its fast access time. Therefore, very little or no time must be wasted when searching for words in the cache. The transformation of data from main memory to cache memory is referred to as a **mapping process**. Organization of cache memory can be classified into three types of mapping process

1. Associative mapping
2. Direct mapping
3. Set-associative mapping

1. Associative Mapping

The fastest and most flexible cache organization uses an associative memory. The associative memory stores both the address and content (data) of the memory word. This permits any location in cache to store any word from main memory. A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address. If the address is found, the corresponding 12-bit data is read and sent to the CPU. If no match occurs, the main memory is accessed for the word. The address--data pair is then transferred to the associative cache memory. If the cache is full, an address--data pair must be displaced to make room for a pair that is needed and not presently in the cache.

2. Direct Mapping

Associative memories are expensive compared to random-access memories because of the added logic associated with each cell. The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the index field and the remaining six bits form the **tag field**. The number of bits in the index field is equal to the number of address bits required to access the cache memory.

In the general case, there are 2^k words in cache memory and 2^n words in main memory. The n -bit memory address is divided into two fields: k bits for the index field and $n - k$ bits for the tag field. The direct mapping cache organization uses the n -bit address to access the main memory and the k -bit index to access the cache. Each word in cache consists of the data word and its associated tag. When a new word is first brought into the cache, the tag bits are stored alongside the data bits. When the CPU generates a memory request, the index field is used for the address to access the cache. The tag field of the CPU address is compared with the tag in the word read from the cache. If the two tags match, there is a hit and the desired data word is in cache. If there is no match, there is a miss and the required word is read from main memory.

3. Set-Associative Mapping

A third type of cache organization, called set-associative mapping, is an improvement over the directmapping organization in that each word of cache can store two or more words of memory under the same index address. Each data word is stored together with its tag and the number of tag-data items in one word of cache is said to form a set. When the CPU generates a memory request, the index value of the address is used to access the cache. The tag field of the CPU address is then compared with both tags in the cache to determine if a match occurs. The comparison logic is done by an associative search of the tags in the set similar to an associative memory search: thus the name "set-associative." The hit ratio will improve as the set size increases because more words with the same index but different tags can reside in cache. When a miss occurs in a set-associative cache and the set is full, it is necessary to replace one of the tag-data items with a new value. The most common replacement algorithms used are: random replacement, first-in, firstout (FIFO), and least recently used (LRU).

Writing into Cache

An important aspect of cache organization is concerned with memory write requests. When the CPU finds a word in cache during a read operation, the main memory is not involved in the transfer. However, if the operation is a write, there are two ways that the system can proceed. The simplest and most commonly used procedure is to update main memory with every memory write operation, with cache memory being updated in parallel if it contains the word at the specified address. This is called the write-through method. This method has the advantage that main memory always contains the same data as the cache. This characteristic is important in systems with direct memory access transfers. The second procedure is called the write-back method. In this method only the cache location is updated during a write operation. The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory. The reason for the write-back method is that during the time a word resides in the cache, it may be updated several times.

Cache Initialization

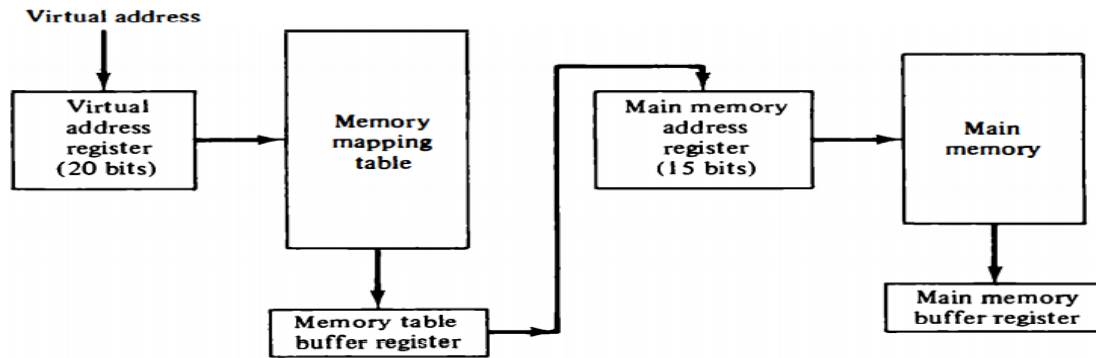
One more aspect of cache organization that must be taken into consideration is the problem of initialization. The cache is initialized when power is applied to the computer or when the main memory is loaded with a complete set of programs from auxiliary memory. After initialization the cache is considered to be empty, but in effect it contains some nonvalid data. It is customary to include with each word in cache a valid bit to indicate whether or not the word contains valid data. The cache is initialized by clearing all the valid bits to 0. The valid bit of a particular cache word is set to 1 the first time this word is loaded from main memory and stays set unless the cache has to be initialized again.

Virtual Memory

Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large Memory space were available, equal to the totality of auxiliary memory. Each address that is referenced by the CPU goes through an address mapping from the so-called virtual address to a physical address in main memory. Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main memory. A virtual memory system provides a mechanism for translating program-generated addresses into correct main memory locations. This is done dynamically, while programs are being executed in the CPU. The translation or mapping is handled automatically by the hardware by means of a mapping table.

Address Space and Memory Space

An address used by a programmer will be called a virtual address, and the set of such addresses the **address space**. An address in main memory is called a location or **physical address**. The set of such locations is called the **memory space**. Thus the address space is the set of addresses generated by programs as they reference instructions and data the memory space consists of the actual main memory locations directly addressable for processing. In most computers the address and memory spaces are identical. The address space is allowed to be larger than the memory space in computers with virtual memory. In a virtual memory system, programmers are told that they have the total address space at their disposal. Moreover, the address field of the instruction code has a sufficient number of bits to specify all virtual addresses.



Address Mapping Using Pages

The table implementation of the address mapping is simplified if the information in the address space and the memory space are each divided into groups of fixed size. The physical memory is broken down into groups of equal size called **blocks**, which may range from 64 to 4096 words each. The term **page** refers to groups of address space of the same size. The programs are also considered to be split into pages. Portions of programs are moved from auxiliary memory to main memory in records equal to the size of a page. The term "page frame" is sometimes used to denote a block.

Page Replacement

A virtual memory system is a combination of hardware and software techniques. The memory management software system handles all the software operations for the efficient utilization of memory space. It must decide

- (1) Which page in main memory ought to be removed to make room for a new page
- (2) When a new page is to be transferred from auxiliary memory to main memory
- (3) Where the page is to be placed in main memory.

The hardware mapping mechanism and the memory management software together constitute the architecture of a virtual memory. When a program starts execution, one or more pages are transferred into main memory and the page table is set to indicate their position. The program is executed from main memory until it attempts to reference a page that is still in auxiliary memory. This condition is called page fault. When a page fault occurs in a virtual memory system, it signifies that the page referenced by the CPU is not in main memory. A new page is then transferred from auxiliary memory to main memory. If main memory is full, it would be necessary to remove a page from a memory block to make room for the new page. The policy for choosing pages to remove is determined from the replacement algorithm that is used. Two of the most common replacement algorithms used are the first-in, first-out (FIFO) and the least recently used (LRU).