UNIT IV

Memory System

Basic concepts, Semiconductor RAMs, ROMs, Speed, size and cost, Cache memories, Performance consideration, Virtual memory, Memory Management requirements, Secondary storage

4.1 MEMORY SYSTEM

4.1.1 Basic Concepts

Q1. Explain the basic concepts of a memory system.

Ans :

(Imp.)

Computer should have a large memory to facilitate execution of programs that are large and deal with huge amounts of data. The memory should be fast, large, and inexpensive. Unfortunately, it is impossible to meet all three of these requirements simultaneously. Increased speed and size are achieved at increased cost. To solve this problem, much work has gone into developing clever structures that improve the apparent speed and size of the memory, yet keep the cost reasonable.

The maximum size of the memory that can be used in any computer is determined by the addressing scheme. For example, a 16-bit computer that generates 16-bit addresses is capable of addressing up to 216 = 64K (65536) memory locations. Similarly, machines whose instructions generate 32-bit addresses can utilize a memory that contains up to 232 = 4G (giga) memory locations, whereas machines with 40-bit addresses can access up to 240 = 1 T (tera) locations. The number of locations represents the size of the address space of the computer.

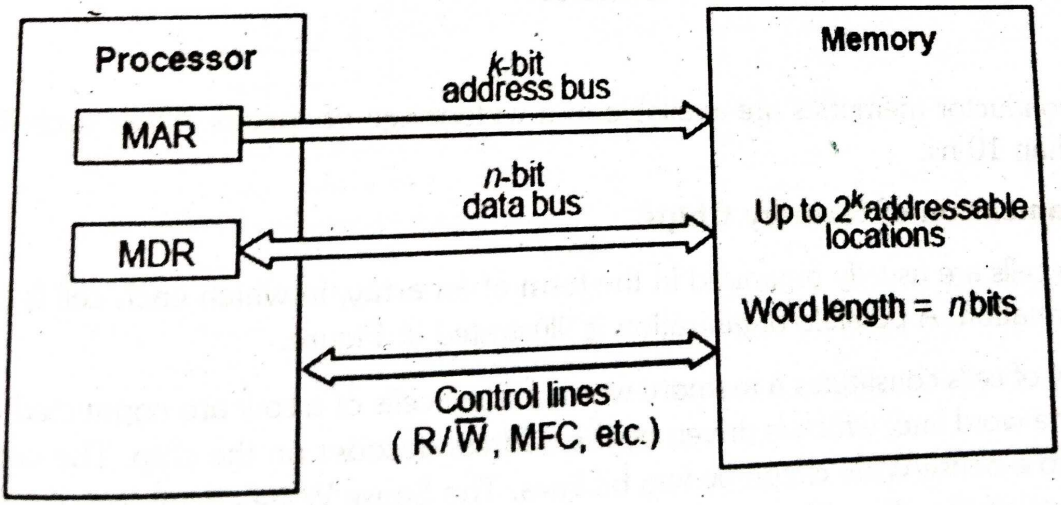


Fig.: Connection of the memory to the processor

From the system standpoint, we can view the memory unit as a black box. Data transfer between the memory and the processor takes place through the use of two processor registers, usually called MAR

(memory address register) and MDR (memory data register), If MAR is k bits long and MDR is n bits long, then the memory unit may contain up to 2 k addressable locations. During a memory cycle, n bits of data are transferred between the memory and the processor. This transfer takes place over the processor bus, which has k address lines and n data lines. The bus also includes the con1rollines Read / Write (R / W) and Memory Function Completed (MFC) for coordinating data transfers. Other control lines may be added to indicate the number of bytes to be transferred. The connection between the processor and the memory is shown schematically in Figure.

The processor reads data from the memory by loading the address of the required memory location into the MAR register and setting the R / W line to 1. The memory responds by placing the data from the addressed location onto the data lines, and confirms this action by asserting the MFC signal. Upon receipt of the MFC signal, the processor loads the data on the data lines into the MDR register. The processor writes data into a memory location by loading the address of this location into MAR and loading the data into MDR. It indicates that a write operation is involved by setting the R/ W line to 0. If read or write operations involve consecutive address locations in the main memory, then a "block transfer" operation can be performed in which the only address sent to the memory is the one that identifies the first location.

The time between the Read and the MFC signals is referred to as the memory access time. The memory cycle time is the minimum time delay required between the initiations of two successive memory operations, If any location can be accessed for a Read or Write operation some fixed amount of time that is independent of the location's address in a memory unit is called random-access memory (RAM). One way to reduce the memory access time is to use a cache memory. This is a small, fast memory that is inserted between the larger, slower main memory and the processor. It holds the currently active segments of a program and their data.

Virtual memory is used to increase the apparent size of the physical memory. Data are addressed in a virtual address space that can be as large as the addressing capability of the processor. But at any given time, only the active portion of this space is mapped onto locations in the physical memory. The remaining virtual addresses are mapped onto the bulk: storage devices used, which are usually magnetic disks. The virtual address space is mapped onto the physical memory where data are actually stored. The mapping function is implemented by a special memory control circuit, often called the memory management unit.

4.1.2 Semiconductor RAM's

Q2. Discuss Semiconductor RAM Memories. **(Imp.)**

Ans :

Semiconductor memories are available in a wide range of speeds. Their cycle times range from l00ns to less than 10 ns.

Internal Organization of Memory Chips

Memory cells are usually organized in the form of an array, in which each cell is capable of storing one bit of information. A possible organization is illustrated in Figure.

Each row of cells constitutes a memory word, and all cells of a row are connected to a common line referred to as the word line, which is driven by the address decoder on the chip. The cells in each column are connected to a Sense/Write circuit by two bit lines. The Sense/Write circuits are connected to the data input/output lines of the chip. During a Read operation, these circuits sense, or read, the information stored in the cells selected by a word line and transmit this information to the output data lines. During a Write operation, the Sense/Write circuits receive input information and store it in the cells of the selected word.

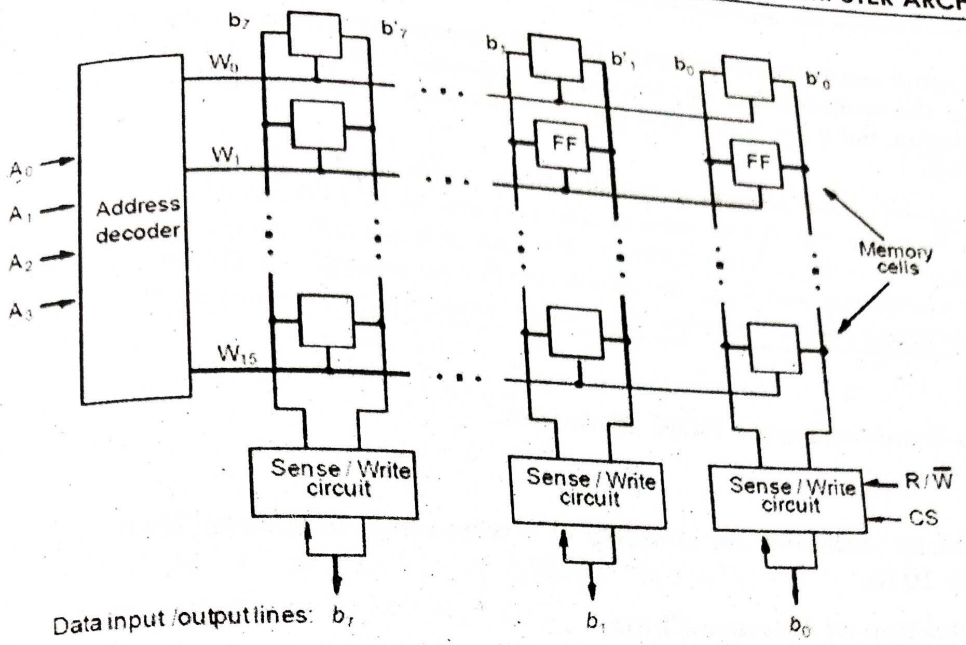


Fig. : Organization of bit cell in a memory chip

Figure 2 is an example of a very small memory chip consisting of 16 words of 8 bits each. This is referred to as a 16 x 8 organization. The data input and the data output of each Sense/Write circuit are connected to a single bidirectional data line that, can be connected to the data bus of a computer. Two control lines, R/W and CS, are provided in addition to address and data lines. The R/W (Read/Write) input specifies the required operation, and the CS (Chip Select) input selects a given chip in a multichip memory system.

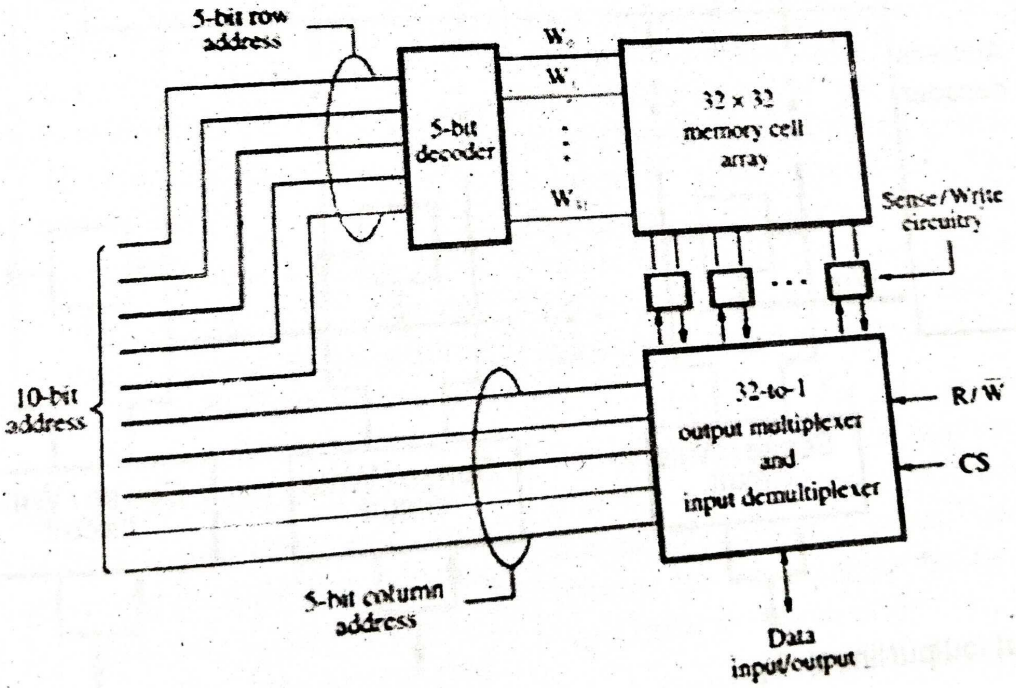


Fig. : Memory circuit

The memory circuit in Figure stores 128 bits and requires 14 external connections for address data, and control lines. Of course, it also needs two lines for power supply and ground connection Consider now a slightly larger memory circuit, one that has 1 K (1024) memory cells.

This circuit can be organized as a 128 x 8 memory, requiring a total of 19 external connections. Alternatively, the same number of cells can be organized into a 1 K x 1 format. In this case, a 100bit address is needed, but there is only one data line, resulting in 15 external connections. Figure shows such an organization.

The required 100bit address is divided into two groups of 5 bits each to form the row and column addresses for the cell array. A row address selects a row of 32 cells, all of which are accessed in parallel. However, according to the column address, only one of these cells is connected to the external data line by the output multiplexer and input demultiplexer. For an example, a 4M-bit chip may have a 512K x 8 organization, in which case 19 address and 8 data input/output pins are needed.

4.1.3 ROMS

Q3. Discuss Semiconductor ROM Memories.

Ans :

Semiconductor memories are available in a wide range of speeds. Their cycle times range from 100ns to less than 10 ns.

Internal Organization of Memory Chips

Memory cells are usually organized in the form of an array, in which each cell is capable of storing one bit of information. A possible organization is illustrated in Figure.

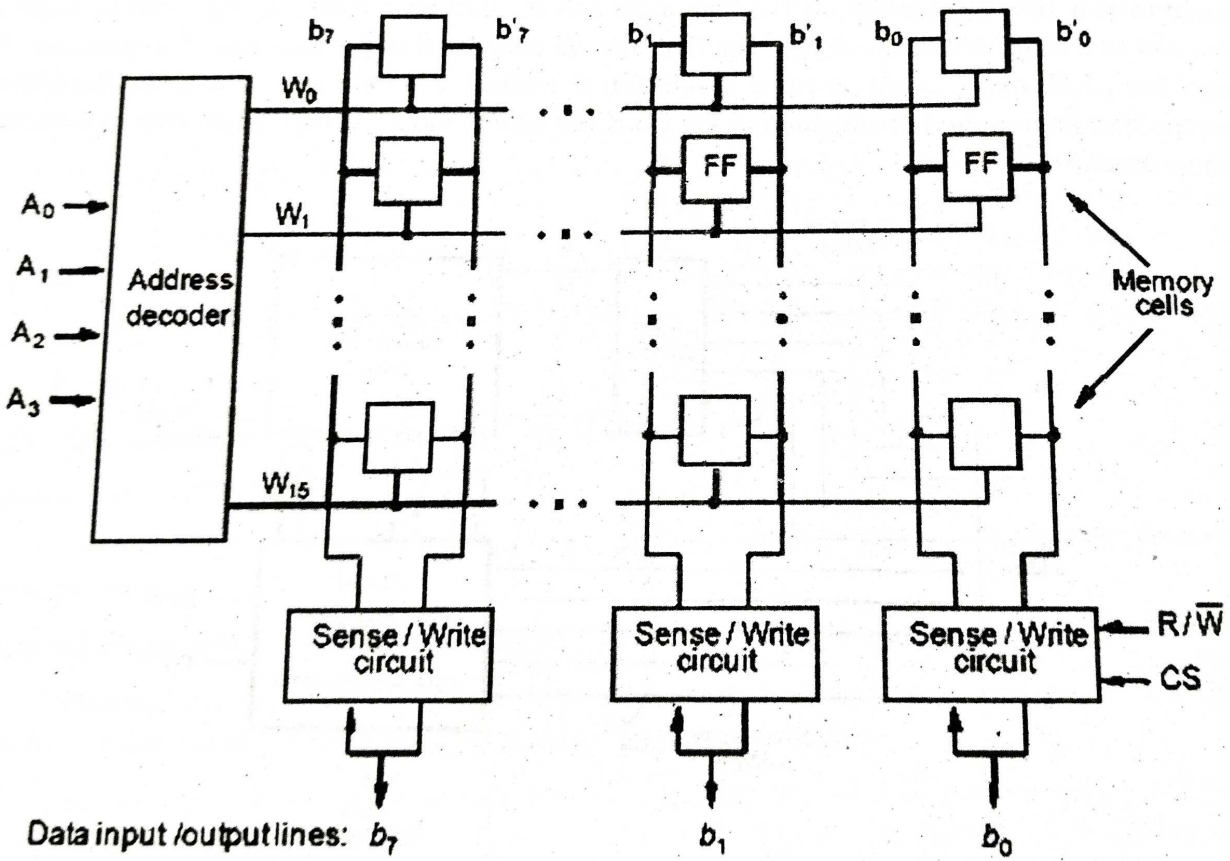


Fig.: Organization of bit cell in a memory chip

Each row of cells constitutes a memory word, and all cells of a row are connected to a common line ed to as the word line, which is driven by the address decoder on the chip. The cells in each column onnected to a Sense/Write circuit by two bit lines. The Sense/Write circuits are connected to the data

input/output lines of the chip. During a Read operation, these circuits sense, or read, the information stored in the cells selected by a word line and transmit this information to the output data lines. During a Write operation, the Sense/Write circuits receive input information and store it in the cells of the selected word.

Figure is an example of a very small memory chip consisting of 16 words of 8 bits each. This is referred to as a 16 x 8 organization. The data input and the data output of each Sense/Write circuit are connected to a single bidirectional data line that, can be connected to the data bus of a computer. Two control lines, R/W and CS, are provided in addition to address and data lines. The R/W (Read/ Write) input specifies the required operation, and the CS (Chip Select) input selects a given chip in a multichip memory system.

The memory circuit in Figure stores 128 bits and requires 14 external connections for address, data, and control lines. Of course, it also needs two lines for power supply and ground connections. Consider now a slightly larger memory circuit, one that has l K (1024) memory cells.

This circuit can be organized as a 128 x 8 memory, requiring a total of 19 external connections. Alternatively, the same number of cells can be organized into a l K x 1 format. In this case, a 100bit address is needed, but there is only one data line, resulting in 15 external connections. Figure 4.3 shows such an organization.

The required 100bit address is divided into two groups of 5 bits each to form the row and column addresses for the cell array. A row address selects a row of 32 cells, all of which are accessed in parallel. However, according to the column address, only one of these cells is connected to the external data line by the output multiplexer and input demultiplexer. For an example, a 4M-bit chip may have a 512K x 8 organization, in which case 19 address and 8 data input/output pins are needed.

The important advantage of EPROM chips is that their contents can be erased and reprogrammed. Erasure requires dissipating the charges trapped in the transistors of memory cells; this can be done by exposing the chip to ultraviolet light. For this reason, EPROM chips are mounted in packages that have transparent windows.

EEPROM

A significant disadvantage of EPROMs is that a chip must be physically removed from the circuit for reprogramming and that its entire contents are erased by the ultraviolet light. It is possible to implement another version of erasable PROMs that can be both programmed and erased electrically. Such chips, called EEPROMs, do not have to be removed for erasure. Moreover, it is possible to erase the cell contents selectively. The only disadvantage of EEPROMs is that different voltages are needed for erasing, writing, and reading the stored data.

Flash Memories

An approach similar to EEPROM technology has more recently given rise to flash memory devices. A flash cell is based on a single transistor controlled by trapped charge, just like an EEPROM cell. While similar in some respects, there are also substantial differences between flash and EEPROM devices. In EEPROM it is possible to read and write the contents of a single cell. In a flash device it is possible to read the contents of a single cell, but it is only possible to write an entire block of cells. Prior to writing, the previous contents of the block are erased. Flash devices have greater density, which leads to higher capacity and a lower cost per bit. They require a single power supply voltage, and consume less power in their operation. The low power consumption of flash memory makes it attractive for use in portable equipment that is battery driven. typical applications include hand-held computers, cell phones, digital cameras, and MP3 music players.

Single flash chips do not provide sufficient storage capacity for the applications mentioned above. Larger memory modules consisting of a number of chips are needed. There are two popular choices for the implementation of such modules: flash cards and flash drives.

Flash Cards

One way of constructing a larger module is to mount flash chips on a small card. Such flash cards have a standard interface that makes them usable in a variety of products. A card is simply

plugged into a conveniently accessible slot. Flash cards come in a variety of memory sizes. Typical sizes are 8, 32, 64 MB and so on.

Flash Drive

Larger flash memory modules have been developed to replace, hard disk drives. These flash drives are designed to fully emulate the bard disks, to the point that they can be fitted into standard disk drive bays. The fact that flash drives are solid state electronic devices that have no movable parts provides some important advantages. They have shorter seek and access times, which results in faster response. They have lower power consumption, which makes them attractive for battery driven applications, and they are also insensitive to vibration. Another type of ROM chip allows the stored data to be erased and new data to be loaded. The disadvantages of flash drives are less storage capacity, higher cost per bit and it will become weak after it has been written several times.

4.1.4 Speed Size And Cost

Q4. Write about speed size and cost of memory.

Ans :

An ideal memory would be fast, large, and inexpensive. a very fast memory can be implemented if SRAM chips are used. But these chips are expensive because their basic cells have six transistors, which preclude packing a very large number of cells onto a single chip. Thus, for cost reasons, it is impractical to build a large memory using SRAM chips. The alternative is to use Dynamic RAM chips, which have much simpler basic cells and thus are much less expensive. But such memories are significantly slower.

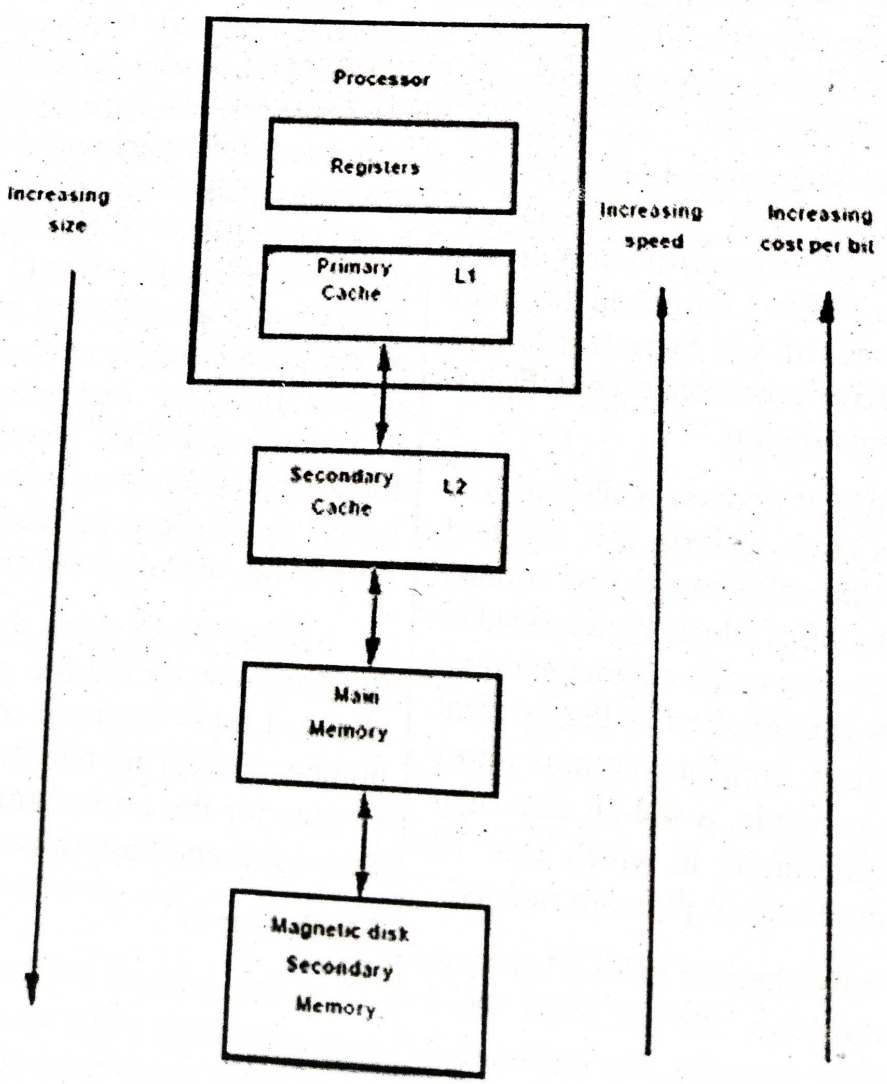


Fig. 4: Memory hierarchy

Although dynamic memory units in the range of hundreds of megabytes can be implemented at a reasonable cost, the affordable size is still small compared to the demands of large programs with voluminous data. A solution is provided by using secondary storage, mainly magnetic disks, to implement large memory spaces. Very large disks are available at a reasonable price, and they are used extensively in computer systems. However, they are much slower than the semiconductor memory units. So A huge amount of cost-effective storage can be provided by magnetic disks. A large, yet affordable, main memory can be built with dynamic RAM technology. This leaves SRAMs to be used in smaller units where speed is of the essence, such as in cache memories.

All of these different types of memory units are employed effectively in a computer. The entire computer memory can be viewed as the hierarchy depicted in Figure. The fastest access is to data held in processor registers. Therefore, if we consider the registers to be part of the memory hierarchy, then the processor registers are at the top in terms of the speed of access. Of course, the registers provide only a minuscule portion of the required memory.

At the next level of the hierarchy is a relatively small amount of memory that can be implemented directly on the processor chip. This memory, called a processor cache, holds copies of instructions and data stored in a much larger memory that is provided externally. There are often two levels of caches. A primary cache is always located on the processor chip. This cache is small because it competes for space on the processor chip, which must implement many other functions.

The primary cache is referred to as level (L1) cache. A larger, secondary cache is placed between the primary cache and the rest of the memory. It is referred to as level 2 (L2) cache. It is usually implemented using SRAM chips. It is possible to have both LI and L2 caches on the processor chip.

The next level in the hierarchy is called the main memory. This rather large memory is implemented using dynamic memory components, typically in the form of SIMMs, DIMMs, or RIMMs. The main memory is much larger but significantly slower than the cache memory. In a typical computer, the access time for the main memory is about ten times longer than the access time for the L 1 cache.

Disk devices provide a huge amount of inexpensive storage. They are very slow compared to the semiconductor devices used to implement the main memory. A hard disk drive (HDD; also hard drive, hard disk, magnetic disk or disk drive) is a device for storing and retrieving digital information, primarily computer data. It consists of one or more rigid (hence "hard") rapidly rotating discs (often referred to as platters), coated with magnetic material and with magnetic heads arranged to write data to the surfaces and read it from them. During program execution, the speed of memory access is of utmost importance. The key to managing the operation of the hierarchical memory system in Figure 4. is to bring the instructions and data that will be used in the near future as close to the processor as possible. This can be done by using the hardware mechanisms.

4.2 Cache Memories

Q5. Explain about cache memory and use of cache memory.

Ans : **(Imp.)**

Cache Memories

In most of the computer system, the speed of the main memory is very low than the speed of modem processors. For good performance, it is important to devise a scheme that reduces the time needed to access the necessary information. Since the speed of the main memory unit is limited by electronic and packaging constraints, the solution must be sought in a different architectural arrangement an efficient solution is to use a fast cache memory which essentially makes the main memory appear to the processor to be faster.

The effectiveness of the cache mechanism is based on a property of computer programs called locality of reference. Analysis of programs shows that most of their execution time is spent on routines in which many instructions are executed repeatedly. These instructions may constitute a simple loop, nested loops, or a few procedures that repeatedly call each other. Many instructions in localized areas of the program are executed repeatedly during

some time period, and the remainder of the program is accessed relatively infrequently. This is referred to as locality of reference. It manifests itself in two ways: temporal and spatial. The temporal means that a recently executed instruction is likely to be executed again very soon. The spatial aspect means that instructions in close proximity to a recently executed instruction (with respect to the instructions' addresses) are also likely to be executed soon. Block refers to a set of contiguous address locations of some size. Another term that is often used to refer to a cache block is cache line.

Consider the simple arrangement in Figure. When a Read request is received from the processor, the contents of a block of memory words containing the location specified are transferred into the cache one word at a time. Subsequently, when the program references any of the locations in this block, the desired contents are read directly from the cache. Usually, the cache memory can store a reasonable number of blocks at any given time, but this number is small compared to the total number of blocks in the main memory. The correspondence between the main memory blocks and those in the cache is specified by a mapping function. When the cache is full and a memory word (instruction or data) that is not in the cache is referenced, the cache control hardware must decide which block should be removed to create space for the new block that contains the referenced word. The collection of rules for making this decision constitutes the replacement algorithm.

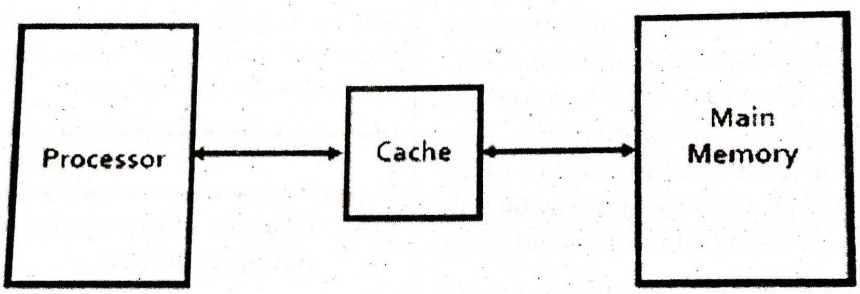


Fig. : Use of cache memory

The processor does not need to know explicitly about the existence of the cache. It simply issues Read and Write requests using addresses that refer to locations in the memory. The cache control circuitry determines whether the requested word currently exists in the cache. If it does, the Read or Write operation is performed on the appropriate cache location. In this case, a read or write hit is said to have occurred. In a Read operation, the main memory is not involved. For a Write operation, the system can proceed in two ways. In the first technique, called the writethrough protocol, the cache location and the main memory location are updated simultaneously. The second technique is to update only the cache location and to mark it as updated with an associated flag bit, often called the dirty or modified bit. The main memory location of the word is updated later, when the block containing this marked word is to be removed from the cache to make room for a new block. This technique is known as the write- back, or copy-back, protocol. The write-through protocol is simpler, but it results in unnecessary Write operations in the main memory when a given cache word is updated several times during its cache residency. Note that the write-back protocol may also result in unnecessary Write operations because when a cache block is written back to the memory all words of the block are written back, even if only a single word has been changed while the block was in the cache.

When the addressed word in a Read operation is not in the cache, a read miss occurs. The block of words that contains the requested word is copied from the main memory into the cache. After the entire block is loaded into the cache, the particular word requested is forwarded to the processor. Alternatively, this word may be sent to the processor as soon as it is read from the main memory. The latter approach,

which is called load-through, or early restart, reduces the processor's waiting period somewhat, but at the expense of more complex circuitry. During a Write operation, if the addressed word is not in the cache, a write miss occurs. Then, if the write-through protocol is used, the information is written directly into the main memory. In the case of the write-back protocol, the block containing the addressed word is first brought into the cache, and then the desired word in the cache is overwritten with the new information.

4.2.1 Performance Consideration

Q6. Explain, how to improve the performance of cache memory. **(Imp.)**

Ans :

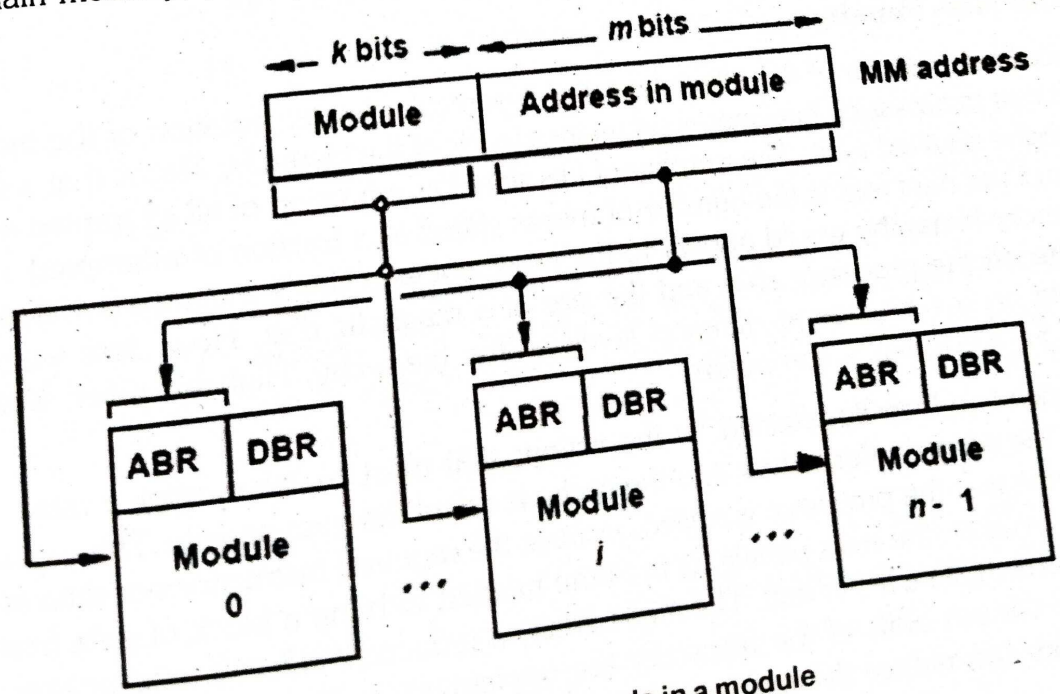
Improving Cache Performance

Two key factors in the commercial success of a computer are performance and cost; the best possible performance at the lowest cost is the objective. The challenge in considering design alternatives is to improve the performance without increasing the cost. A common measure of success is the price/performance ratio. Performance depends on how fast machine instructions can be brought into the processor for execution and how fast they can be executed.

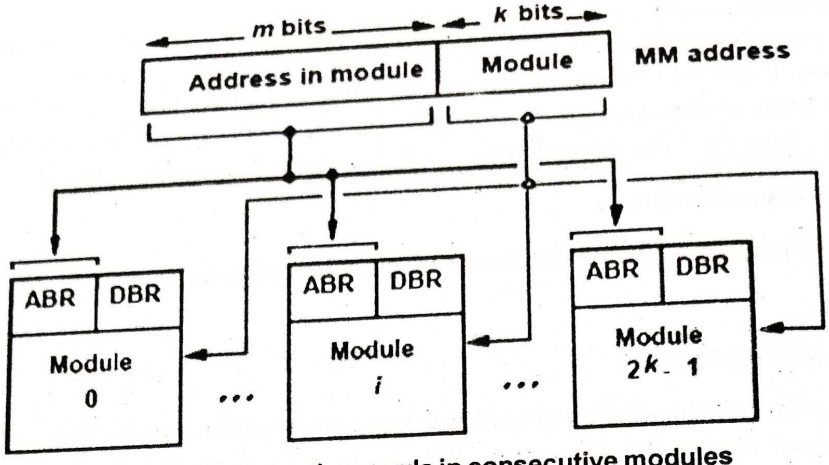
The memory hierarchy is used for the best price/performance ratio. The main purpose of this hierarchy is to create a memory I that the processor sees as having a short access time and a large capacity. Each level of the hierarchy plays an important role. The speed and efficiency of data transfer between various levels of the hierarchy are also of great significance. It is beneficial if transfers to and from the faster units can done at a rate equal to that of the faster unit. This is not possible if both the slow and the fast units are accessed in the same manner, but it can be achieved when parallelism is used in the organization of the slower unit. An effective way to introduce parallelism is to use an interleaved organization.

Interleaving

If the main memory of a computer is structured as a collection of physically separate modules, each with its own address buffer register (ABR) and data buffer register (DBR), memory access operations may proceed in more than one module at the same time. Thus, the aggregate rate of transmission of words to and from the main memory system can be increased.



(a) Consecutive words in a module



(b) Consecutive words in consecutive modules

Fig.: Addressing multiple-module memory systems

How individual addresses are distributed over the modules is critical in determining the average number of modules that can be kept busy as computations proceed. Two methods of address layout are indicated in Figure (b). In the first case, the memory address generated by the processor is decoded as shown in Figure (a). The high- order k bits name one of n modules, and the low-order m bits name a particular word in that module. When consecutive locations are accessed, as happens when a block of data is transferred to a cache, only one module is involved. At the same time, however, devices with direct memory access (DMA) ability may be accessing information in other memory modules.

The second and more effective way to address the modules is shown in Figure 6b. It is called memory interleaving. The low-order k bits of the memory address select a module, and the high-order m bits name a location within that module. In this way, consecutive addresses are located in successive modules. Thus, any component of the system that generates requests for access to consecutive memory locations can keep several modules busy at anyone time. This results in both faster accesses to a block of data and higher average utilization of the memory system as a whole. To implement the interleaved structure, there must be 2k modules; otherwise, there will be gaps of nonexistent locations in the memory address space.

Hit Rate And Miss Penalty

An excellent indicator of the effectiveness of a particular implementation of the memory hierarchy is the success rate in accessing information at various levels of the hierarchy. Recall that a successful access to data in a cache is called a hit. The number of hits stated as a fraction of all attempted accesses is called the hit rate, and the miss rate is the number of misses stated as a fraction of attempted accesses. Ideally, the entire memory hierarchy would appear to the processor as a single memory unit that has the access time of a cache on the processor chip and the size of a magnetic disk. How close we get to this ideal depends largely on the hit rate at different levels of the hierarchy. High hit rates, well over 0.9, are essential for high-performance computers.

Performance is adversely affected by the actions that must be taken after a miss. The extra time needed to bring the desired information into the cache is called the miss penalty. This penalty is ultimately reflected in the time that the processor is stalled because the required instructions or data are not available for execution. In general, the miss penalty is the time needed to bring a block of data from a slower unit in the memory hierarchy to a faster unit. The miss penalty is reduced if efficient mechanisms for transferring data between the various units of the hierarchy are implemented. The previous section shows how an interleaved memory can reduce the miss penalty substantially.

Caches on The Processor Chip

From the speed point of view, the optimal place for a cache is on the processor chip. Unfortunately, space on the processor chip is needed for many other functions; this limits the size of the cache that can be accommodated. All high-performance processor chips include some form of a cache. Some manufacturers have chosen to implement two separate caches, one for instructions and another for data. A combined cache for instructions and data is likely to have a somewhat better hit rate because it offers greater flexibility in mapping new information into the cache.

In high-performance processors two levels of caches are normally used. The L1 cache(s) is on the processor chip. The L2 cache, which is much larger, may be implemented externally using SRAM chips. But, a somewhat smaller L2 cache may also be implemented on the processor chip. If both L1 and L2 caches are used, the L1 cache should be designed to allow very fast access by the processor because its access time will have a large effect on the clock rate of the processor. A cache cannot be accessed at the same speed as a register file because the cache is much bigger and, hence, more complex. A practical way to speed up access to the cache is to access more than one word simultaneously and then let the processor use them one at a time.

The L2 cache can be slower, but it should be much larger to ensure a high hit rate. Its speed is less critical because it only affects the miss penalty of the L 1 cache. A workstation computer may include an Ll cache with the capacity of tens of kilobytes and an L2 cache of several megabytes. Including an L2 cache further reduces the impact of the main memory speed on the performance of a computer. The average access time experienced by the processor in a system with two levels of caches is

have = h1 C1 (1 - h1) h2 C2 (1 - h1) (1 - h2) M

where

h1 is the hit rate in the Ll cache.

h2 is the hit rate in the L2 cache.

C1 is the time to access information in the Ll cache.

C2 is the time to access information in the L2 cache.

M is the time to access information in the main memory.

The number of misses in the L2 cache, given by the term (1 - h1) (1 - h2), should be low. If both h1 and h 2 are in the 90 percent range, then the number of misses will be less than 1 percent of the processor's memory accesses.

4.2.2 Virtual Memory

Q7. What do you mean by address space and memory space in virtual memory? Also explain the relation between address space and memory space in virtual memory. (Imp.)

Ans :

Virtual 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.

Address Space

- An address used by a programmer will be called a virtual address, and the set of such addresses is known as address space.

Memory Space

- An address in main memory is called a location or physical address. The set of such locations is called the memory space.

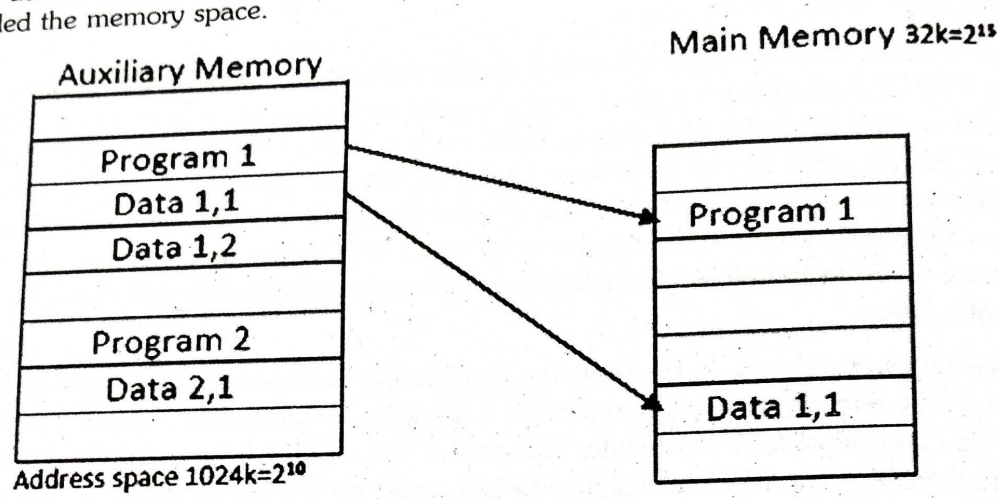


Fig.: Relation between address and memory space in a virtual memory system

- As an illustration, consider a computer with a main-memory capacity of 32K words (K = 1024). Fifteen bits are needed to specify a physical address in memory since 32K = 215.
- Suppose that the computer has available auxiliary memory for storing 220 = 1024K words.
- Thus auxiliary memory has a capacity for storing information equivalent to the capacity of 32 main memories.
- Denoting the address space by N and the memory space by M, we then have for this example N = 1024K and M = 32K.
- In a multiprogramming computer system, programs and data are transferred to and from auxiliary memory and main memory based on demands imposed by the CPU.
- Suppose that program 1 is currently being executed in the CPU. Program 1 and a portion of its associated data are moved from auxiliary memory into main memory as shown in figure.
- Portions of programs and data need not be in contiguous locations in memory since information is being moved in and out, and empty spaces may be available in scattered locations in memory.
- In our example, the address field of an instruction code will consist of 20 bits but physical memory addresses must be specified with only 15 bits.
- Thus CPU will reference instructions and data with a 20-bit address, but the information at this address must be taken from physical memory because access to auxiliary storage for individual words will be too long.

Q8. Explain address mapping using pages.

Ans : **(Imp.)**

- 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.
- Consider a computer with an address space of 8K and a memory space of 4K.
- If we split each into groups of 1K words we obtain eight pages and four blocks as shown in figure.
- At any given time, up to four pages of address space may reside in main memory in any one of the four blocks.

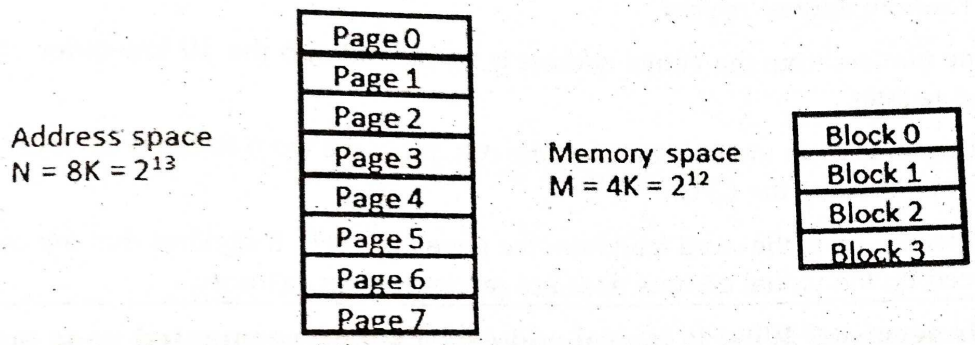


Fig. 8: Address and memory space split into group of 1K words

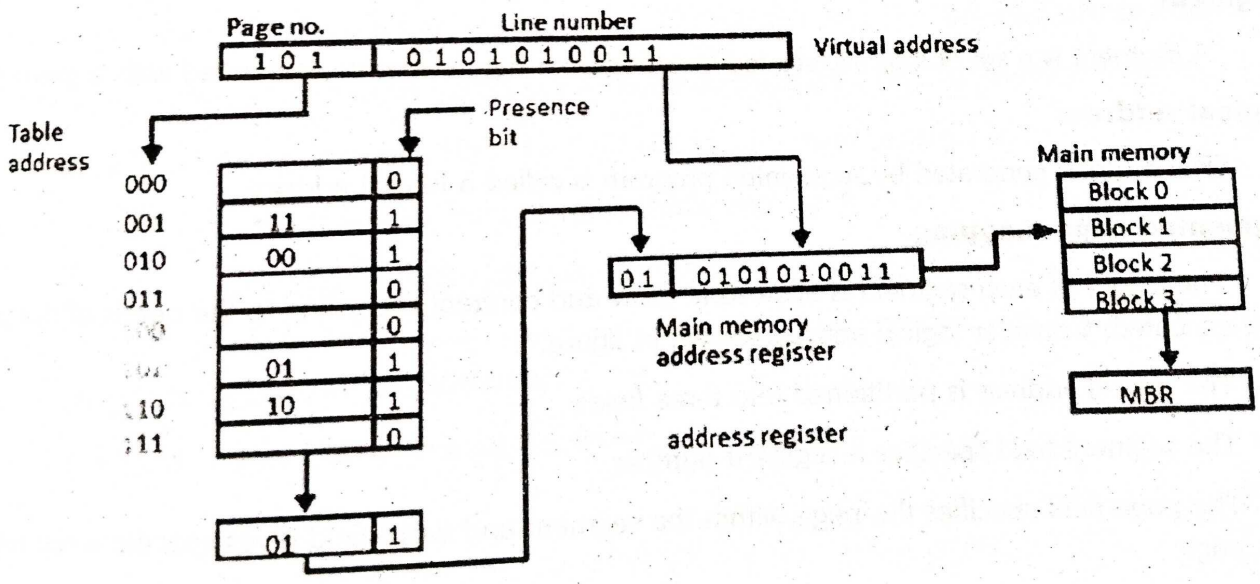


Fig.: Memory table in paged system

- The organization of the memory mapping table in a paged system is shown in figure8
- The memory-page table consists of eight words, one for each page.
- The address in the page table denotes the page number and the content of the word give the block number where that page is stored in main memory.

- The table shows that pages 1, 2, 5, and 6 are now available in main memory in blocks 3, 0, 1, and 2, respectively.
- A presence bit in each location indicates whether the page has been transferred from auxiliary memory into main memory.
- A 0 in the presence bit indicates that this page is not available in main memory.
- The CPU references a word in memory with a virtual address of 13 bits.
- The three high-order bits of the virtual address specify a page number and also an address for the memory-page table.
- The content of the word in the memory page table at the page number address is read out into the memory table buffer register.
- If the presence bit is a 1, the block number thus read is transferred to the two high-order bits of the main memory address register.
- The line number from the virtual address is transferred into the 10 low-order bits of the memory address register.
- A read signal to main memory transfers the content of the word to the main memory buffer register ready to be used by the CPU.
- If the presence bit in the word read from the page table is 0, it signifies that the content of the word referenced by the virtual address does not reside in main memory.

Q9. What is segment? What is logical address? Explain segmented page mapping. **(Imp.)**

Ans :

Segment

A segment is a set of logically related instructions or data elements associated with a given name.

Logical address

The address generated by segmented program is called a logical address.

Segmented page mapping

The length of each segment is allowed to grow and contract according to the needs of the program being executed. Consider logical address shown in figure.

- The logical address is partitioned into three fields.
- The segment field specifies a segment number.
- The page field specifies the page within the segment and word field gives specific word within the page.
- A page field of k bits can specify up to 2k pages.
- A segment number may be associated with just one page or with as many as 2k pages.
- Thus the length of a segment would vary according to the number of pages that are assigned to it.
- The mapping of the logical address into a physical address is done by means of two tables, as shown in figure.

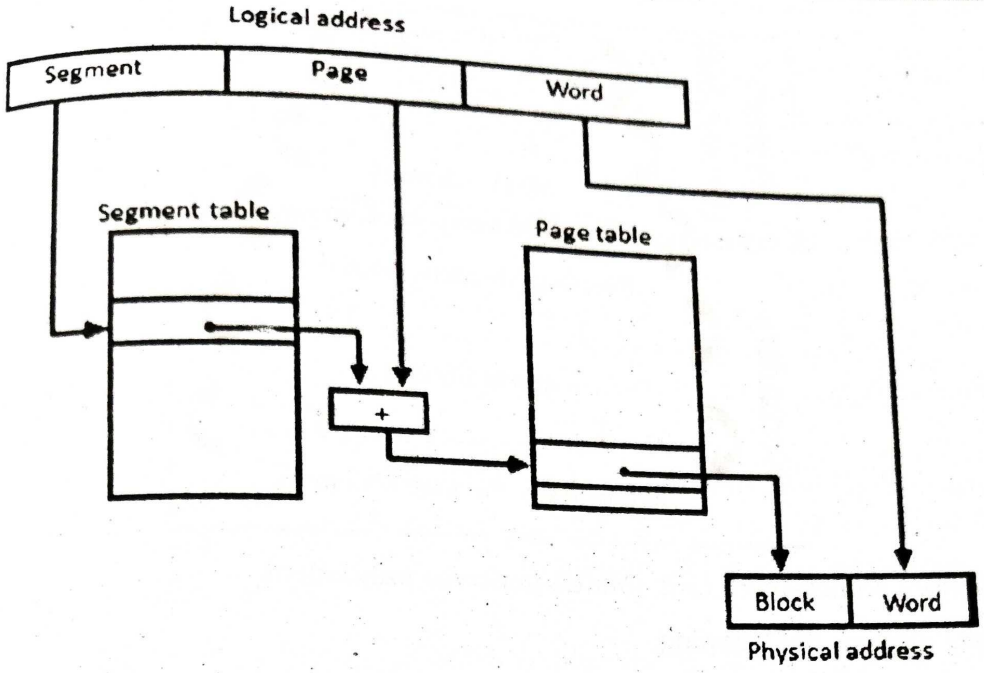


Fig.: Logical to physical address mapping

- The segment number of the logical address specifies the address for the segment table.
- The entry in the segment table is a pointer address for a page table base.
- The page table base is added to the page number given in the logical address.
- The sum produces a pointer address to an entry in the page table.
- The concatenation of the block field with the word field produces the final physical mapped address.
- The two mapping tables may be stored in two separate small memories or in main memory.
- In either case, memory reference from the CPU will require three accesses to memory:
- One from the segment table, one from the page table and the third from main memory.
- This would slow the system significantly when compared to a conventional system that requires only one reference to memory.

4.2.3 Memory Management Requirements

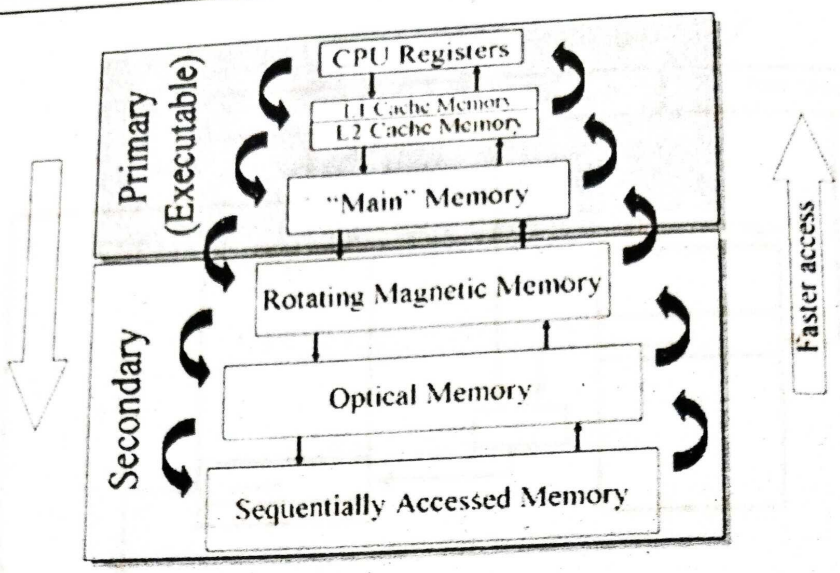
Q10. Write about memory management hardware. **(Imp.)**

Ans :

Memory Hierarchy

Computers have several different types of memory. This memory is often viewed as a hierarchy as shown below.

Our main concern here will be the computer's main or RAM memory. The cache memory is important because it boost's the speed of accessing memory, but it is managed entirely by the hardware. The rotating magnetic memory or disk memory is used by the Virtual Memory Management.



The *Memory Management Unit* (MMU) performs translations.

The MMU contains the following:

- The table walk unit, which contains logic that reads the translation tables from memory.
- Translation Lookaside Buffers (TLBs), which cache recently used translations.

All memory addresses that are issued by software are virtual. These memory addresses are passed to the MMU, which checks the TLBs for a recently used cached translation. If the MMU does not find a recently cached translation, the table walk unit reads the appropriate table entry, or entries, from memory, as shown here:

A virtual address must be translated to a physical address before a memory access can take place (because we must know which physical memory location we are accessing). This need for translation also applies to cached data, because on Armv6 and later processors, the data caches store data using the physical address (addresses that are physically tagged). Therefore, the address must be translated before a cache lookup can complete.

Note:

Architecture is a behavioural specification. The caches must behave as if they are physically tagged. An implementation might do something different, as long as this is not software-visible.

Table entry

The translation tables work by dividing the virtual address space into equal-sized blocks and by providing one entry in the table per block.

Entry 0 in the table provides the mapping for block 0, entry 1 provides the mapping for block 1, and so on. Each entry contains the address of a corresponding block of physical memory and the attributes to use when accessing the physical address.

Table lookup

A table lookup occurs when a translation takes place. When a translation happens, the virtual address that is issued by the software is split in two, as shown in this diagram:

This diagram shows a single-level lookup.

The upper-order bits, which are labelled 'Which entry' in the diagram, tell you which block entry to look in and they are used as an index into the table. This entry block contains the physical address for the virtual address.

The lower-order bits, which are labelled 'Offset in block' in the diagram, are an offset within that block and are not changed by the translation.

Multilevel translation

In a single-level lookup, the virtual address space is split into equal-sized blocks. In practice, a hierarchy of tables is used.

The first table (Level 1 table) divides the virtual address space into large blocks. Each entry in this table can point to an equal-sized block of physical memory or it can point to another table which subdivides the block into smaller blocks. We call this type of table a 'multilevel table'. Here we can see an example of a multilevel table that has three levels:

In Armv 8-A, the maximum number of levels is four, and the levels are numbered 0 to 3. This multilevel approach allows both larger blocks and smaller blocks to be described. The characteristics of large and small blocks are as follows:

- Large blocks require fewer levels of reads to translate than small blocks. Plus, large blocks are more efficient to cache in the TLBs.
- Small blocks give software fine-grain control over memory allocation. However, small blocks are less efficient to cache in the TLBs. Caching is less efficient because small blocks require multiple reads through the levels to translate.

To manage this trade-off, an OS must balance the efficiency of using large mappings against the flexibility of using smaller mappings for optimum performance.

4.2.4 Secondary Storage

Q11. What is auxiliary/secondary storage? Write about them.

Ans :

Auxiliary storage also known as auxiliary memory or secondary storage, is the memory that supplements the main storage. This is a longterm, nonvolatile memory. The term nonvolatile means that stores and retains the programs and data even after the computer is switched off. Unlike RAM which loses the contents when the computer is turned off, and ROM, to which it is not possible to add anything new, auxiliary storage devices allow the computer to record information semi permanently, so it can be read later by the same computer or by another computer. Auxiliary storage devices are also useful in transferring data or programs from one computer to another.

They also function as backup devices which allow to backup the valuable information. So even if by some accident the computer crashes and the stored data is unrecoverable, we can restore it from the backups. The most common types of auxiliary storage devices are magnetic tapes, magnetic disks, floppy disks, hard disks, etc.There are two types of auxiliary storage devices.

This classification is based on the type of data access:

1. sequential
2. random.

Based on the type of access, they are called sequentialâ □'access media or randomâ □'media. In the case of sequentialâ □ 'access media, the data stored in the media can only be read in sequence and to get to a particular point on the media, we have to go through all the preceding points. Magnetic tapes are examples of sequential'access media. In contrast, disks are randomâ □'access also called direct'access media because a disk drive can access any point at random without passing through intervening points. Other examples of direct access media are floppy diskettes, optical disks, zip disks, etc.

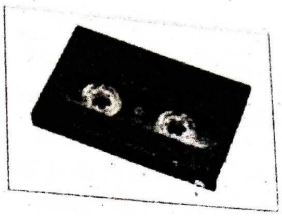
Magnetic Tape

Magnetic tape is a magnetically coated strip of plastic on which data can be encoded. Tapes for computers are similar to the tapes used to store music. Some computers, in fact, enable us to use normal cassette tapes. Storing data on tapes is considerably cheaper than storing data on disks. Tapes also have large storage capacities, ranging from a few hundred kilobytes to several gigabytes. Accessing data on tapes, however, is much slower than accessing data on disks. Because tapes are so slow, they are generally used only for longâ □'term storage and backup. Data to be used regularly is almost always kept on a disk. Tapes are also used for transporting large amounts of data. Tapes come in a variety of sizes and formats as given in Table. Tapes are sometimes called streamers or streaming tapes.

Helical-scan Cartridge:

It's a type of magnetic tape that uses the same technology as VCR tapes. The term helical scan usually refers to 8â□'mm tapes, although 4â □'mm tapes (called DAT tapes) use the same technology. The 8â□'mm helicalâ□'scan tapes have data capacities from 2.5GB to 5 GB.

Type	Capacity	Description
Halt Inch	60MB-400MB	Halt-inch tapes come both as 9 track reels and as cartridges. These tapes are relatively cheap, but require expensive tape drives
Quarter Inch	40MB - 5GB	Quarter Inch Cartridges (QIC tapes) are relatively inexpensive and support fast data transfer rates, QIC mini cartridges are even less expensive, but their data capacities are smaller and their transfer rates are lower.
8-mm Helical scan	1GB-5GB	8 min helical-scan cartridges use the same technology as VCR tapes and have the great capacity. But they require expensive tape drives and have relatively slow data transfer rates.
4-mm DAT	2GB - 24GB	DAT (Digital Audio Tape) cartridges have the greatest capacity but they require expensive tape drives and have relatively slow data transfer Rates.



DAT Cartridge

This is a type of magnetic tape that uses an ingenious scheme called helical scan to record data, as shown in Fig. 4.2.1. A DAT cartridge is slightly larger than a credit card and contains a magnetic tape that can hold from 2 to 24 gigabytes of data. It can support data transfer rates of about 2 MBPS (Million Bytes Per Second). Like other types of tapes, DATs are sequentialâ□'access media. The most common format for DAT cartridges is DDS (Digital Data Storage) which is the industry standard for digital audio tape (DAT) formats. The latest format, DDSâ□'3, specifies tapes that can hold 24 GB (the equivalent of over 40 CD ROMs) and support data transfer rates of 2 MBPS.

Short Question and Answers

1. Memory system.

Ans :

Computer should have a large memory to facilitate execution of programs that are large and deal with huge amounts of data. The memory should be fast, large, and inexpensive. Unfortunately, it is impossible to meet all three of these requirements simultaneously. Increased speed and size are achieved at increased cost. To solve this problem, much work has gone into developing clever structures that improve the apparent speed and size of the memory, yet keep the cost reasonable.

The maximum size of the memory that can be used in any computer is determined by the addressing scheme. For example, a 16-bit computer that generates 16-bit addresses is capable of addressing up to 216 = 64K (65536) memory locations. Similarly, machines whose instructions generate 32-bit addresses can utilize a memory that contains up to 232 = 4G (giga) memory locations, whereas machines with 40-bit addresses can access up to 240 = 1 T (tera) locations. The number of locations represents the size of the address space of the computer.

2. Speed size and cost of memory.

Ans :

An ideal memory would be fast, large, and inexpensive. a very fast memory can be implemented if SRAM chips are used. But these chips are expensive because their basic cells have six transistors, which preclude packing a very large number of cells onto a single chip. Thus, for cost reasons, it is impractical to build a large memory using SRAM chips. The alternative is to use Dynamic RAM chips, which have much simpler basic cells and thus are much less expensive. But such memories are significantly slower.

3. Cache Memories

Ans :

In most of the computer system, the speed of the main memory is very low than the speed of modem processors. For good performance, it is important to devise a scheme that reduces the time needed to access the necessary information. Since the speed of the main memory unit is limited by electronic and packaging constraints, the solution must be sought in a different architectural arrangement an efficient solution is to use a fast cache memory which essentially makes the main memory appear to the processor to be faster.

The effectiveness of the cache mechanism is based on a property of computer programs called locality of reference. Analysis of programs shows that most of their execution time is spent on routines in which many instructions are executed repeatedly. These instructions may constitute a simple loop, nested loops, or a few procedures that repeatedly call each other. Many instructions in localized areas of the program are executed repeatedly during some time period, and the remainder of the program is accessed relatively infrequently. This is referred to as locality of reference. It manifests itself in two ways: temporal and spatial. The temporal means that a recently executed instruction is likely to be executed again very soon. The spatial aspect means that instructions in close proximity to a recently executed instruction (with respect to the instructions' addresses) are also likely to be executed soon. Block refers to a set of contiguous address locations of some size. Another term that is often used to refer to a cache block is cache line.

4. Virtual Memory

Ans :

- 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.

Address Space

- An address used by a programmer will be called a virtual address, and the set of such addresses is known as address space.

Address mapping using pages.

5. *Ans :*

- 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.
- Consider a computer with an address space of 8K and a memory space of 4K.
- If we split each into groups of 1K words we obtain eight pages and four blocks as shown in figure.
- At any given time, up to four pages of address space may reside in main memory in any one of the four blocks.

6. Segmented page mapping

Ans :

The length of each segment is allowed to grow and contract according to the needs of the program being executed. Consider logical address shown in figure.

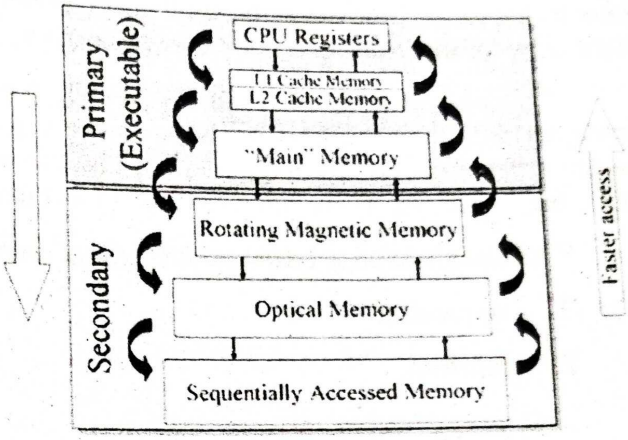
- The logical address is partitioned into three fields.
- The segment field specifies a segment number.
- The page field specifies the page within the segment and word field gives specific word within the page.
- A page field of k bits can specify up to 2k pages.
- A segment number may be associated with just one page or with as many as 2k pages.
- Thus the length of a segment would vary according to the number of pages that are assigned to it.

7. Memory Hierarchy

Ans :

Computers have several different types of memory. This memory is often viewed as a hierarchy as shown below.

Our main concern here will be the computer's main or RAM memory. The cache memory is important because it boost's the speed of accessing memory, but it is managed entirely by the hardware. The rotating magnetic memory or disk memory is used by the Virtual Memory Management.



The *Memory Management Unit* (MMU) performs translations.

The MMU contains the following:

- The table walk unit, which contains logic that reads the translation tables from memory.
- Translation Lookaside Buffers (TLBs), which cache recently used translations.

All memory addresses that are issued by software are virtual. These memory addresses are passed to the MMU, which checks the TLBs for a recently used cached translation. If the MMU does not find a recently cached translation, the table walk unit reads the appropriate table entry, or entries, from memory, as shown here:

A virtual address must be translated to a physical address before a memory access can take place (because we must know which physical memory location we are accessing). This need for translation also applies to cached data, because on Armv6 and later processors, the data caches store data using the physical address (addresses that are physically tagged). Therefore, the address must be translated before a cache lookup can complete.

8. Secondary storage.

Ans :

Auxiliary storage also known as auxiliary memory or secondary storage, is the memory that supplements the main storage. This is a longterm, nonvolatile memory. The term nonvolatile means that stores and retains the programs and data even after the computer is switched off. Unlike RAM which loses the contents when the computer is turned off, and ROM, to which it is not possible to add anything new, auxiliary storage devices allow the computer to record information semi permanently, so it can be read later by the same computer or by another computer. Auxiliary storage devices are also useful in transferring data cr programs from one computer to another.

They also function as backup devices which allow to backup the valuable information. So even if by some accident the computer crashes and the stored data is unrecoverable, we can restore it from the backups. The most common types of auxiliary storage devices are magnetic tapes, magnetic disks, floppy disks, hard disks, etc.There are two types of auxiliary storage devices.

This classification is based on the type of data access:

1. sequential
2. random.

Choose the Correct Answers

1. In the Principle of locality, there is a justification of the use of: [b]
   (a) DMA (b) Cache memory
   (c) Threads (d) Interrupts

2. Which of these memories would have the lowest access time in a system: [c]
   (a) Main Memory (b) Magnetic Disk
   (c) Registers (d) Cache

3. With the help of _____ we reduce the memory access time: [b]
   (a) SDRAM (b) Cache
   (c) Heaps (d) Higher capacity RAMs

4. The address in the main memory is known as: [b]
   (a) Logical address (b) Physical address
   (c) Memory address (d) None of the above

5. Which of the following memory unit communicates directly with the CPU? [b]
   (a) Auxiliary memory (b) Main memory
   (c) Secondary memory (d) None of the above

6. In which of the following term the performance of cache memory is measured? [b]
   (a) Chat ratio (b) Hit ratio
   (c) Copy ratio (d) Data ratio

7. The block transfer capability of the DRAM is called _____ [c]
   (a) Burst mode (b) Block mode
   (c) Fast page mode (d) Fast frame mode

8. The less space consideration as lead to the development of _____ (for large memories). [d]
   (a) SIMM's (b) DIMS's
   (c) SRAM's (d) Both SIMM's and DIMS's

9. The smallest unit of memory that the CPU can read or write is _____. [c]
   (a) Word (b) Mode
   (c) Cell d Field

10. The configuration, in which no difference between memory and I/O devices is seen by the CPU, is referred to as _____. [c]
   (a) Memory unit (b) Memory-mapped I/O
   (c) Memory address register (d) Memory unit

Fill in the Blanks

1. What does EEPROM stands for _____.
2. _____occurs when a program accesses a page which is not present in main memory.
3. During a write operation if the required block is not present in the cache then _____ occurs.
4. The minimum time delay between two successive memory read operations is _____
5. The SDRAM performs operation on the _____
6. The SRAM's are basically used as _____
7. The configuration, in which no difference between memory and I/O devices is seen by the CPU, is referred to as _____.
8. The equation of average memory access time = Hit time + _____ x _____.
9. PROM stands for _____
10. For the synchronization of the read head, we make use of a _____

Answers

1. Electrically Erasable and Programmable Read-Only Memory
2. Page fault
3. Write Miss
4. Cycle time
5. Rising edge of the clock
6. Cache's
7. Memory-mapped I/O
8. Miss rate, Miss penalty
9. Programmable Read Only Memory
10. Clock