UNIT-3

Memory System Design

Memory system design: Semiconductor memory technologies, memory organization.

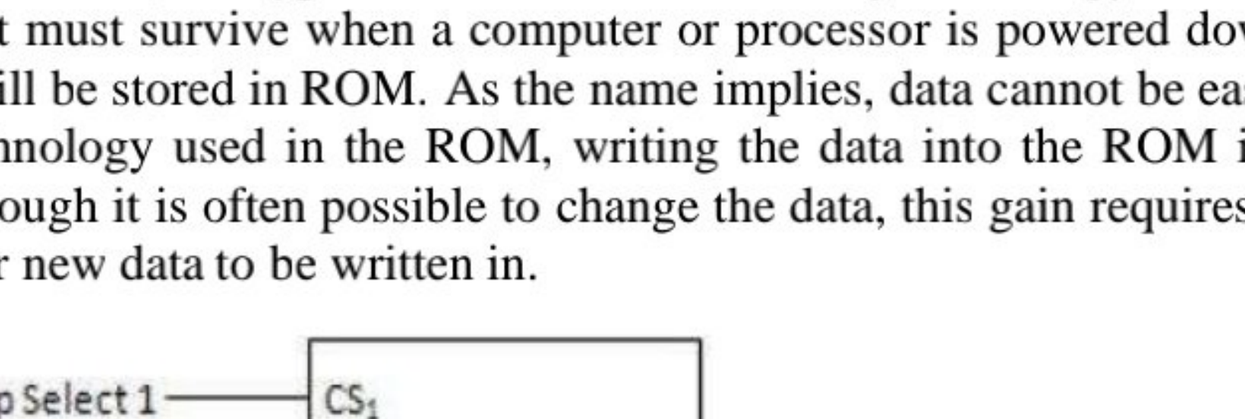
Memory organization: Memory interleaving, concept of hierarchical memory organization, Cache memory, mapping functions, Replacement algorithms, write policies, Virtual Memory Management

Semiconductor Memory Technologies:

Semiconductor random-access memories (RAMs) are available in a wide range of speeds. Their cycle times range from 100 ns to less than 10 ns. Semiconductor memory is used in any electronics assembly that uses computer processing technology. The use of semiconductor memory has grown, and the size of these memory cards has increased as the need for larger and larger amounts of storage is needed.

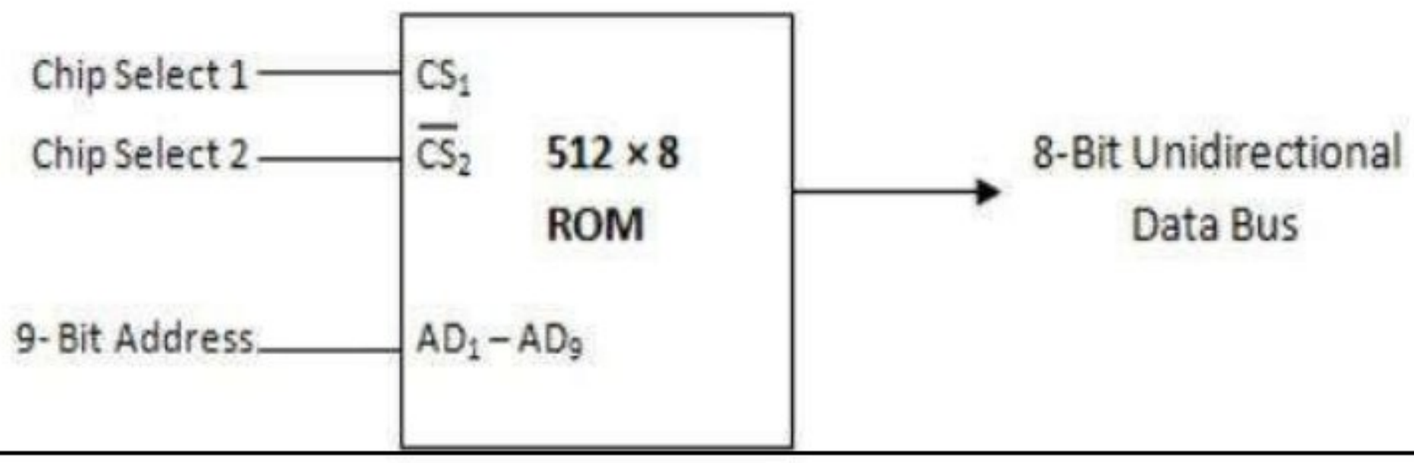
There are two main types or categories that can be used for semiconductor technology.

RAM - Random Access Memory: As the names suggest, the RAM or random access memory is a form of semiconductor memory technology that is used for reading and writing data in any order - in other words as it is required by the processor. It is used for such applications as the computer or processor memory where variables and other stored and are required on a random basis. Data is stored and read many times to and from this type of memory.



Block Diagram Representing 128 x 8 RAM
(Random Access Memory)

ROM - Read Only Memory: A ROM is a form of semiconductor memory technology used where the data is written once and then not changed. In view of this it is used where data needs to be stored permanently, even when the power is removed - many memory technologies lose the data once the power is removed. As a result, this type of semiconductor memory technology is widely used for storing programs and data that must survive when a computer or processor is powered down. For example the BIOS of a computer will be stored in ROM. As the name implies, data cannot be easily written to ROM. Depending on the technology used in the ROM, writing the data into the ROM initially may require special hardware. Although it is often possible to change the data, this gain requires special hardware to erase the data ready for new data to be written in.



The different memory types or memory technologies are detailed below:

DRAM: Dynamic RAM is a form of random access memory. DRAM uses a capacitor to store each bit of data, and the level of charge on each capacitor determines whether that bit is a logical 1 or 0. However these capacitors do not hold their charge indefinitely, and therefore the data needs to be refreshed periodically. As a result of this dynamic refreshing it gains its name of being a dynamic RAM. DRAM is the form of semiconductor memory that is often used in equipment including personal computers and workstations where it forms the main RAM for the computer.

EEPROM: This is an Electrically Erasable Programmable Read Only Memory. Data can be written to it and it can be erased using an electrical voltage. This is typically applied to an erase pin on the chip. Like other types of PROM, EEPROM retains the contents of the memory even when the power is turned off. Also like other types of ROM, EEPROM is not as fast as RAM.

EPROM: This is an Erasable Programmable Read Only Memory. This form of semiconductor memory can be programmed and then erased at a later time. This is normally achieved by exposing the silicon to ultraviolet light. To enable this to happen there is a circular window in the package of the EPROM to enable the light to reach the silicon of the chip. When the PROM is in use, this window is normally covered by a label, especially when the data may need to be preserved for an extended period. The PROM stores its data as a charge on a capacitor. There is a charge storage capacitor for each cell and this can be read repeatedly as required. However it is found that after many years the charge may leak away and the data may be lost. Nevertheless, this type of semiconductor memory used to be widely used in applications where a form of ROM was required, but where the data needed to be changed periodically, as in a development environment, or where quantities were low.

FLASH MEMORY: Flash memory may be considered as a development of EEPROM technology. Data can be written to it and it can be erased, although only in blocks, but data can be read on an individual cell basis. To erase and re-programme areas of the chip, programming voltages at levels that are available within electronic equipment are used. It is also non-volatile, and this makes it particularly useful. As a result Flash memory is widely used in many applications including memory cards for digital cameras, mobile phones, computer memory sticks and many other applications.

F-RAM: Ferroelectric RAM is a random-access memory technology that has many similarities to the standard DRAM technology. The major difference is that it incorporates a ferroelectric layer instead of the more usual dielectric layer and this provides its non-volatile capability. As it offers a non-volatile capability, F-RAM is a direct competitor to Flash.

MRAM: This is Magneto-resistive RAM, or Magnetic RAM. It is a non-volatile RAM memory technology that uses magnetic charges to store data instead of electric charges. Unlike technologies including DRAM, which require a constant flow of electricity to maintain the integrity of the data, MRAM retains data even when the power is removed. An additional advantage is that it only requires low power for active operation. As a result this technology could become a major player in the electronics industry now that production processes have been developed to enable it to be produced.

P-RAM / PCM: This type of semiconductor memory is known as Phase change Random Access Memory, P-RAM or just Phase Change memory, PCM. It is based around a phenomenon where a form of chalcogenide glass changes is state or phase between an amorphous state (high resistance) and a polycrystalline state (low resistance). It is possible to detect the state of an individual cell and hence use this for data storage. Currently this type of memory has not been widely commercialized, but it is expected to be a competitor for flash memory.

PROM: This stands for Programmable Read Only Memory. It is a semiconductor memory which can only have data written to it once - the data written to it is permanent. These memories are bought in a blank format and they are programmed using a special PROM programmer. Typically a PROM will consist of an array of fuseable links some of which are "blown" during the programming process to provide the required data pattern.

SDRAM: Synchronous DRAM. This form of semiconductor memory can run at faster speeds than conventional DRAM. It is synchronised to the clock of the processor and is capable of keeping two sets of memory addresses open simultaneously. By transferring data alternately from one set of addresses, and then the other, SDRAM cuts down on the delays associated with non-synchronous RAM, which must close one address bank before opening the next.

SRAM: Static Random Access Memory. This form of semiconductor memory gains its name from the fact that, unlike DRAM, the data does not need to be refreshed dynamically. It is able to support faster read and write times than DRAM (typically 10 ns against 60 ns for DRAM), and in addition its cycle time is much shorter because it does not need to pause between accesses. However it consumes more power, is less dense and more expensive than DRAM. As a result of this it is normally used for caches, while DRAM is used as the main semiconductor memory technology.

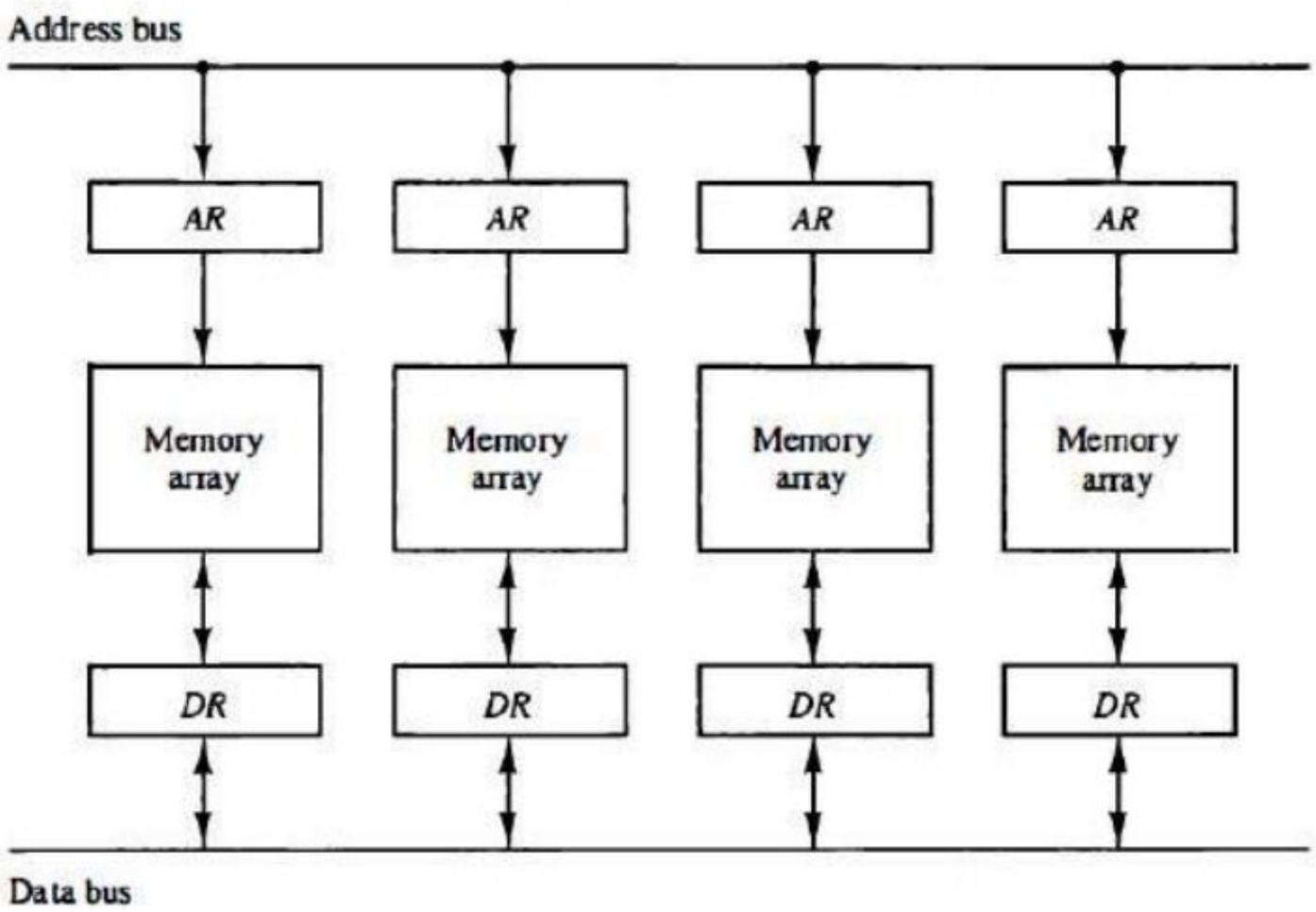
MEMORY ORGANIZATION

Memory Interleaving:

Pipeline and vector processors often require simultaneous access to memory from two or more sources. An instruction pipeline may require the fetching of an instruction and an operand at the same time from two different segments.

Similarly, an arithmetic pipeline usually requires two or more operands to enter the pipeline at the same time. Instead of using two memory buses for simultaneous access, the memory can be partitioned into a number of modules connected to a common memory address and data buses. A memory module is a memory array together with its own address and data registers. Figure 9-13 shows a memory unit with four modules. Each memory array has its own address register AR and data register DR.

Figure 9-13 Multiple module memory organization.



The address registers receive information from a common address bus and the data registers communicate with a bidirectional data bus. The two least significant bits of the address can be used to distinguish between the four modules. The modular system permits one module to initiate a memory access while other modules are in the process of reading or writing a word and each module can honor a memory request independent of the state of the other modules.

The advantage of a modular memory is that it allows the use of a technique called interleaving. In an interleaved memory, different sets of addresses are assigned to different memory modules. For example, in a two-module memory system, the even addresses may be in one module and the odd addresses in the other.

Concept of Hierarchical Memory Organization

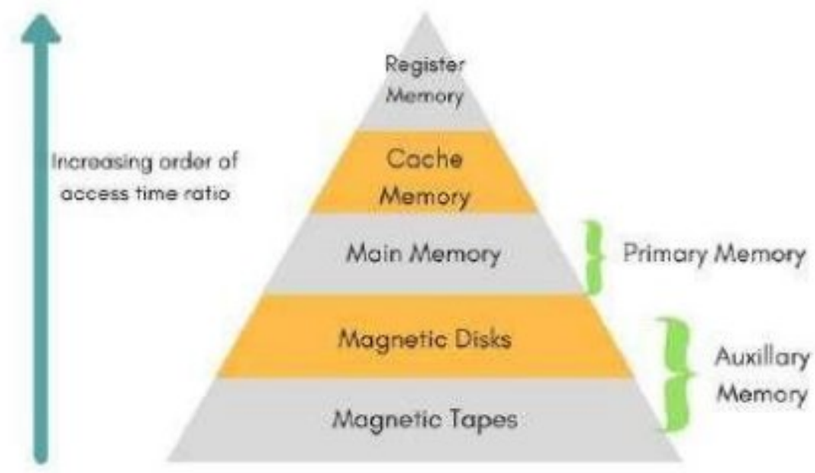
This Memory Hierarchy Design is divided into 2 main types:

External Memory or Secondary Memory

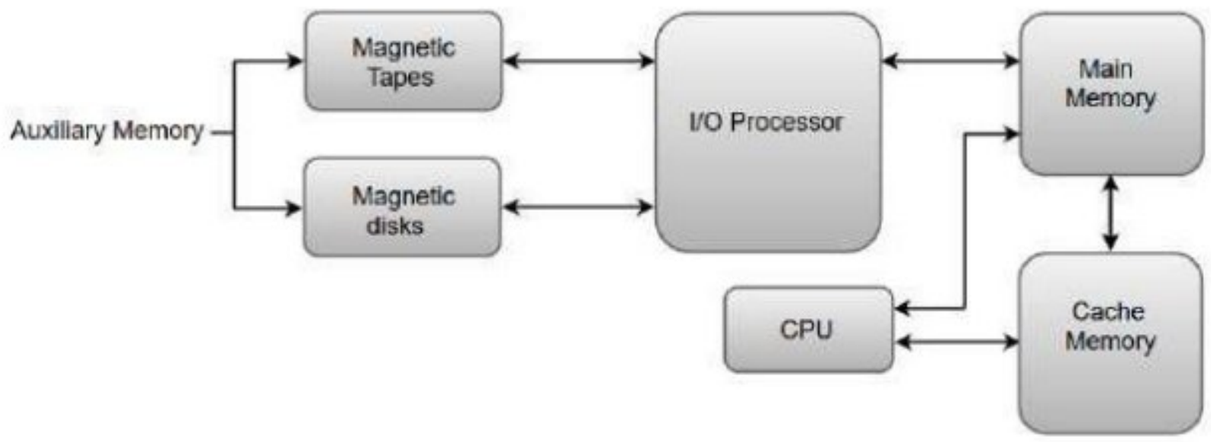
Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

Internal Memory or Primary Memory

Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.



Memory Hierarchy in a Computer System:



Characteristics of Memory Hierarchy

Capacity:

It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.

Access Time:

It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.

Performance:

Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time. This results in lower performance of the system and thus, enhancement was required. This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.

Cost per bit:

As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory.

Cache Memories:

The cache is a small and very fast memory, interposed between the processor and the main memory. Its purpose is to make the main memory appear to the processor to be much faster than it actually is. The effectiveness of this approach is based on a property of computer programs called locality of reference.

Analysis of programs shows that most of their execution time is spent in 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.

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 cache's *replacement algorithm*.

Cache Hits

The processor does not need to know explicitly about the existence of the cache. It simply issues Read andWrite 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 orWrite operation is performed on the appropriate cache location. In this case, a *read* or *write hit* is said to have occurred.

Cache Misses

A Read operation for a word that is not in the cache constitutes a *Read miss*. It causes the block of words containing the requested word to be copied from the main memory into the cache.

Cache Mapping:

There are three different types of mapping used for the purpose of cache memory which are as follows: Direct mapping, Associative mapping, and Set-Associative mapping. These are explained as following below.

Direct mapping

The simplest way to determine cache locations in which to store memory blocks is the *direct-mapping* technique. In this technique, block *j* of the main memory maps onto block *j* modulo 128 of the cache, as depicted in Figure 8.16. Thus, whenever one of the main memory blocks 0, 128, 256, . . . is loaded into the cache, it is stored in cache block 0. Blocks 1, 129, 257, . . . are stored in cache block 1, and so on. Since more than one memory block is mapped onto a given cache block position, contention may arise for that position even when the cache is not full.

For example, instructions of a program may start in block 1 and continue in block 129, possibly after a branch. As this program is executed, both of these blocks must be transferred to the block-1 position in the cache. Contention is resolved by allowing the new block to overwrite the currently resident block.

With direct mapping, the replacement algorithm is trivial. Placement of a block in the cache is determined by its memory address. The memory address can be divided into three fields, as shown in Figure 8.16. The low-order 4 bits select one of 16 words in a block.

When a new block enters the cache, the 7-bit cache block field determines the cache position in which this block must be stored. If they match, then the desired word is in that block of the cache. If there is no match, then the block containing the required word must first be read from the main memory and loaded into the cache.

The direct-mapping technique is easy to implement, but it is not very flexible.

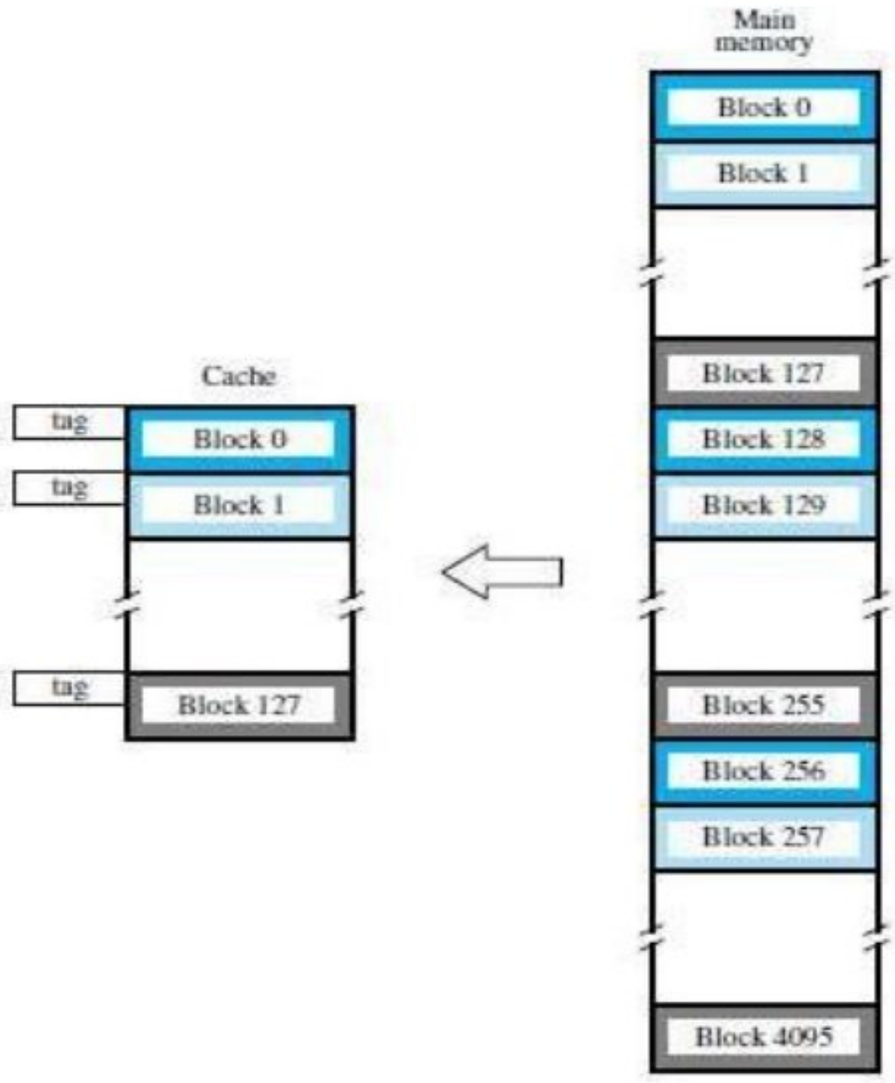


Figure 8.16 Direct-mapped cache.

Associative Mapping

In Associative mapping method, in which a main memory block can be placed into any cache block position. In this case, 12 tag bits are required to identify a memory block when it is resident in the cache. The tag bits of an address received from the processor are compared to the tag bits of each block of the cache to see if the desired block is present. This is called the *associative-mapping* technique.

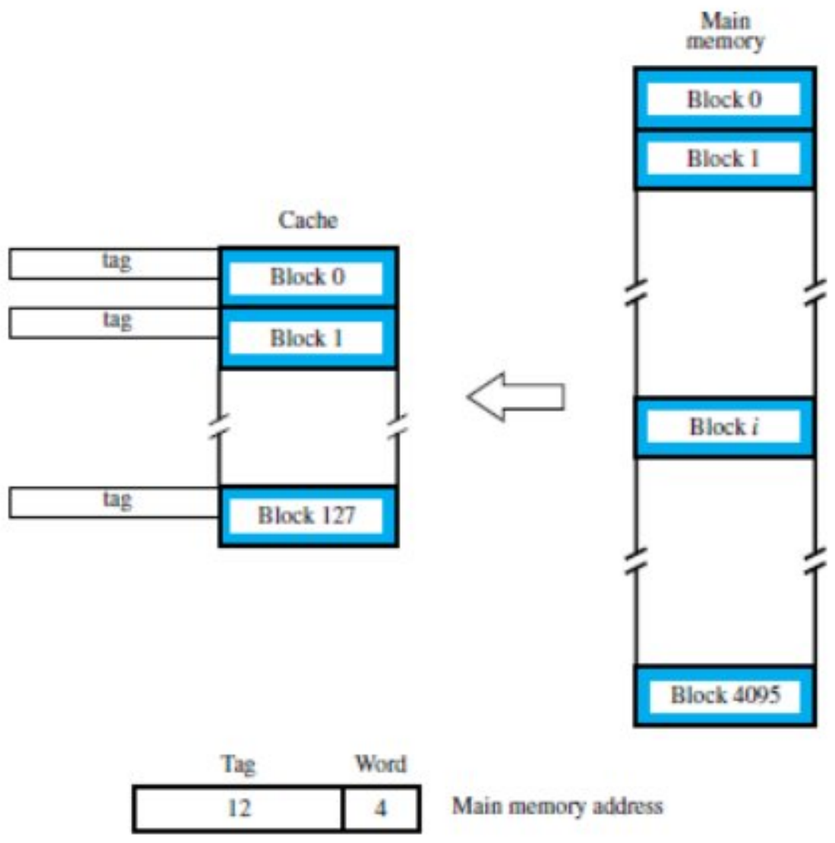


Figure 8.17 Associative-mapped cache.

It gives complete freedom in choosing the cache location in which to place the memory block, resulting in a more efficient use of the space in the cache. When a new block is brought into the cache, it replaces (ejects) an existing block only if the cache is full. In this case, we need an algorithm to select the block to be replaced.

To avoid a long delay, the tags must be searched in parallel. A search of this kind is called an *associative search*.

Set-Associative Mapping

Another approach is to use a combination of the direct- and associative-mapping techniques. The blocks of the cache are grouped into sets, and the mapping allows a block of the main memory to reside in any block of a specific set. Hence, the contention problem of the direct method is eased by having a few choices for block placement.

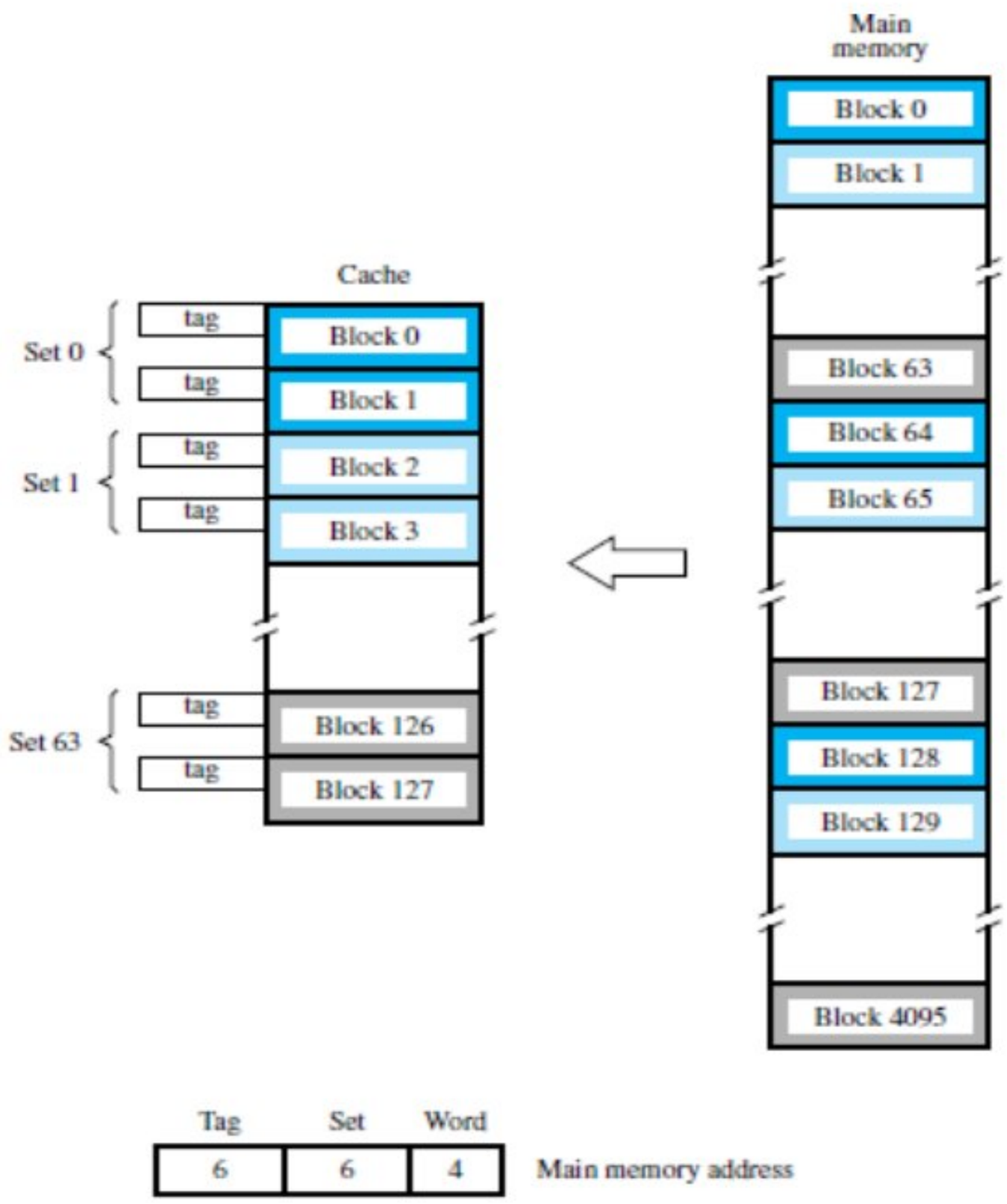


Figure 8.18 Set-associative-mapped cache with two blocks per set.

At the same time, the hardware cost is reduced by decreasing the size of the associative search. An example of this *set-associative-mapping* technique is shown in Figure 8.18 for a cache with two blocks per set. In this case, memory blocks 0, 64, 128, . . . , 4032 map into cache set 0, and they can occupy either of the two block positions within this set.

Having 64 sets means that the 6-bit set field of the address determines which set of the cache might contain the desired block. The tag field of the address must then be associatively compared to the tags of the two blocks of the set to check if the desired block is present. This two-way associative search is simple to implement.

The number of blocks per set is a parameter that can be selected to suit the requirements of a particular computer. For the main memory and cache sizes in Figure 8.18, four blocks per set can be accommodated by a 5-bit set field, eight blocks per set by a 4-bit set field, and so on. The extreme condition of 128 blocks per set requires no set bits and corresponds to the fully-associative technique, with 12 tag bits. The other extreme of one block per set is the direct-mapping.

Replacement Algorithms

In a direct-mapped cache, the position of each block is predetermined by its address; hence, the replacement strategy is trivial. In associative and set-associative caches there exists some flexibility.

When a new block is to be brought into the cache and all the positions that it may occupy are full, thecache controller must decide which of the old blocks to overwrite.

This is an important issue, because the decision can be a strong determining factor in system performance. In general, the objective is to keep blocks in the cache that are likely to be referenced in the near future. But, it is not easy to determine which blocks are about to be referenced.

The property of locality of reference in programs gives a clue to a reasonable strategy. Becauseprogram execution usually stays in localized areas for reasonable periods of time, there is a high probability that the blocks that have been referenced recently will be referenced again soon. Therefore, when a block is to be overwritten, it is sensible to overwrite the one that has gone the longest time without being referenced. This block is called the *least recently used* (LRU) block, and the technique iscalled the *LRU replacement algorithm*.

The LRU algorithm has been used extensively. Although it performs well for many access patterns, it can lead to poor performance in some cases.

Write Policies

The write operation is proceeding in 2 ways.

- Write-through protocol
- Write-back protocol

Write-through protocol:

Here the cache location and the main memory locations are updated simultaneously.

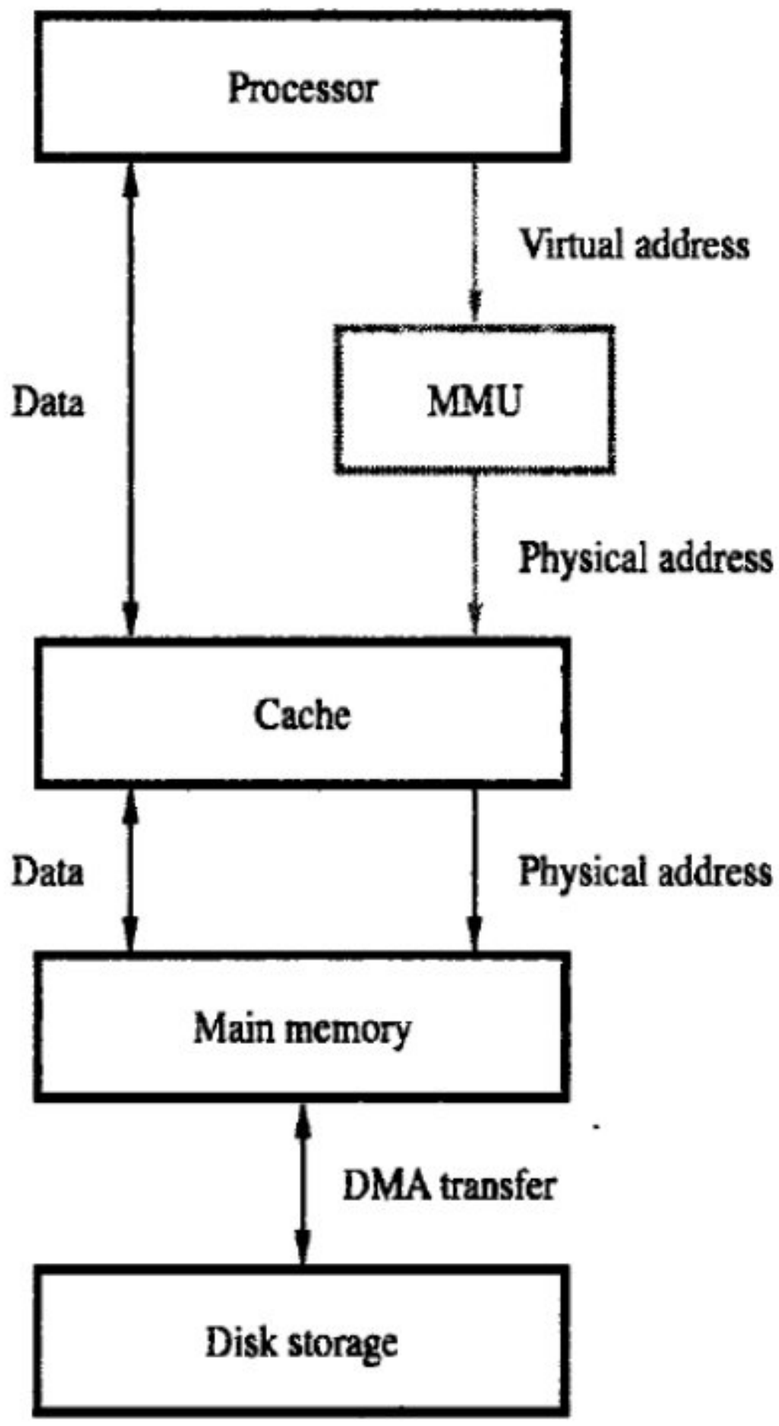
Write-back protocol:

- This technique is to update only the cache location and to mark it as withassociated flag bit called dirty/modified bit.
- The word in the main memory will be updated later, when the block containing thismarked word is to be removed from the cache to make room for a new block.
- To overcome the read miss Load –through / Early restart protocol is used.

Virtual Memory Management /Paging:

In most modern computer systems, the physical main memory is not as large as the address space spanned by an address issued by the processor. For example, a processor that issues 32-bit addresses has an addressable space of 4G bytes. The size of the main memory in a typical computer ranges from a few hundred megabytes to 1G bytes. When a program does not completely fit into the main memory, the parts of it not currently being executed are stored on secondary storage devices, such as magnetic disks. Of course, all parts of a program that are eventually executed are first brought into the

Fig: Virtual Memory Organization



main memory. When a new segment of a program is to be moved into a full memory, it must replace another segment already in the memory. In modern computers, the operating system moves programs and data automatically between the main memory and secondary storage. Thus, the application programmer does not need to be aware of limitations imposed by the available main memory.

Techniques that automatically move program and data blocks into the physical main memory when they are required for execution are called *virtual-memory* techniques. Programs, and hence the processor, reference an instruction and data space that is independent of the available physical main memory space. The binary addresses that the processor issues for either instructions or data are called *virtual* or *logical addresses*. These addresses are translated into physical addresses by a combination of hardware and software components. If a virtual address refers to a part of the program or data space that is currently in the physical memory, then the contents of the appropriate location in the main memory are accessed immediately. On the other hand, if the referenced address is not in the main memory, its contents must be brought into a suitable location in the memory before they can be used.

Figure 5.26 shows a typical organization that implements virtual memory. A special hardware unit, called the *Memory Management Unit* (MMU), translates virtual

addresses into physical addresses. When the desired data (or instructions) are in the main memory, these data are fetched as described in our presentation of the cache mechanism. If the data are not in the main memory, the MMU causes the operating system to bring the data into the memory from the disk. Transfer of data between the disk and the main memory is performed using the DMA scheme discussed in

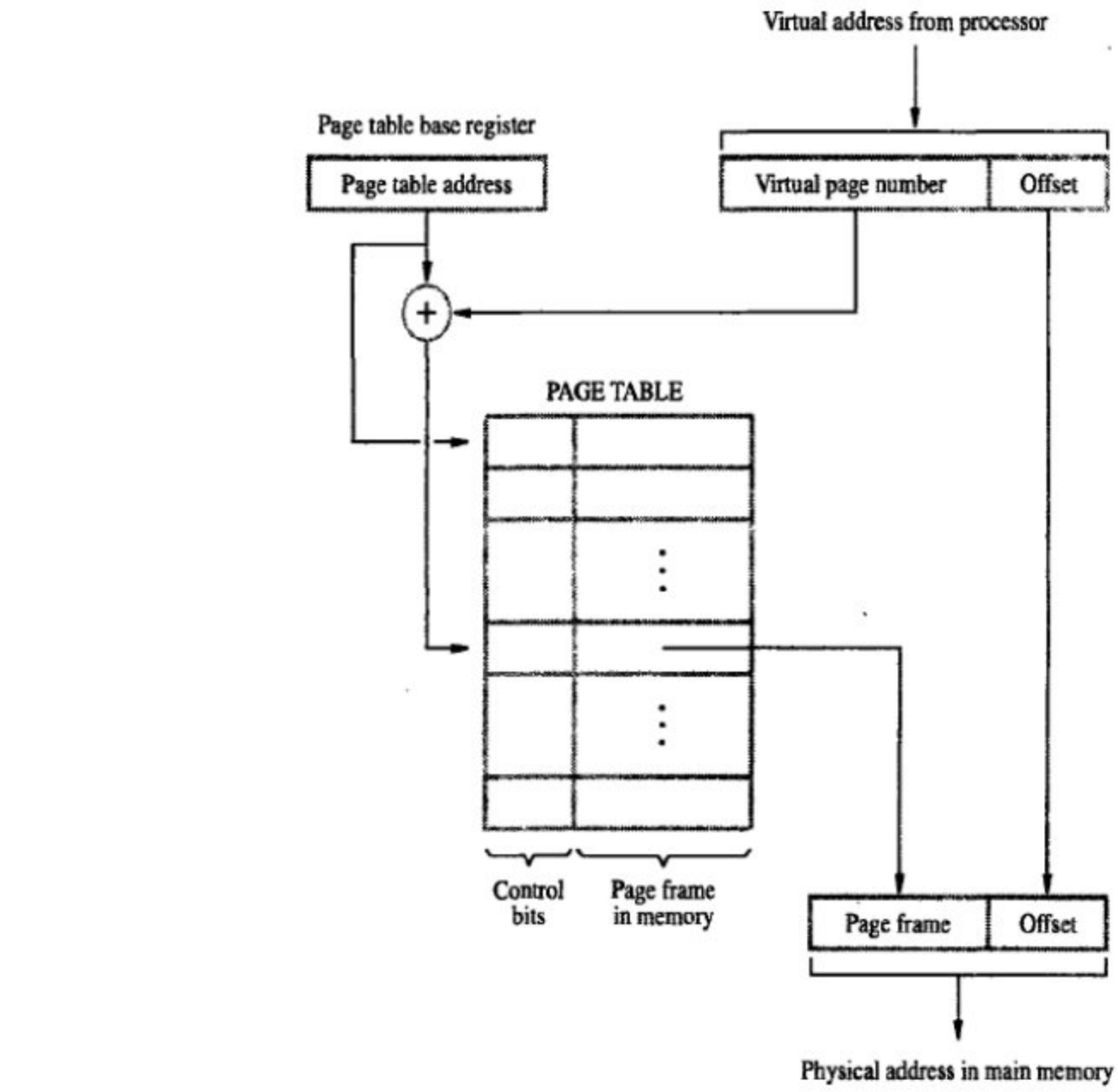


Fig: Virtual Memory Address Translation

Internal Memory Organization:

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

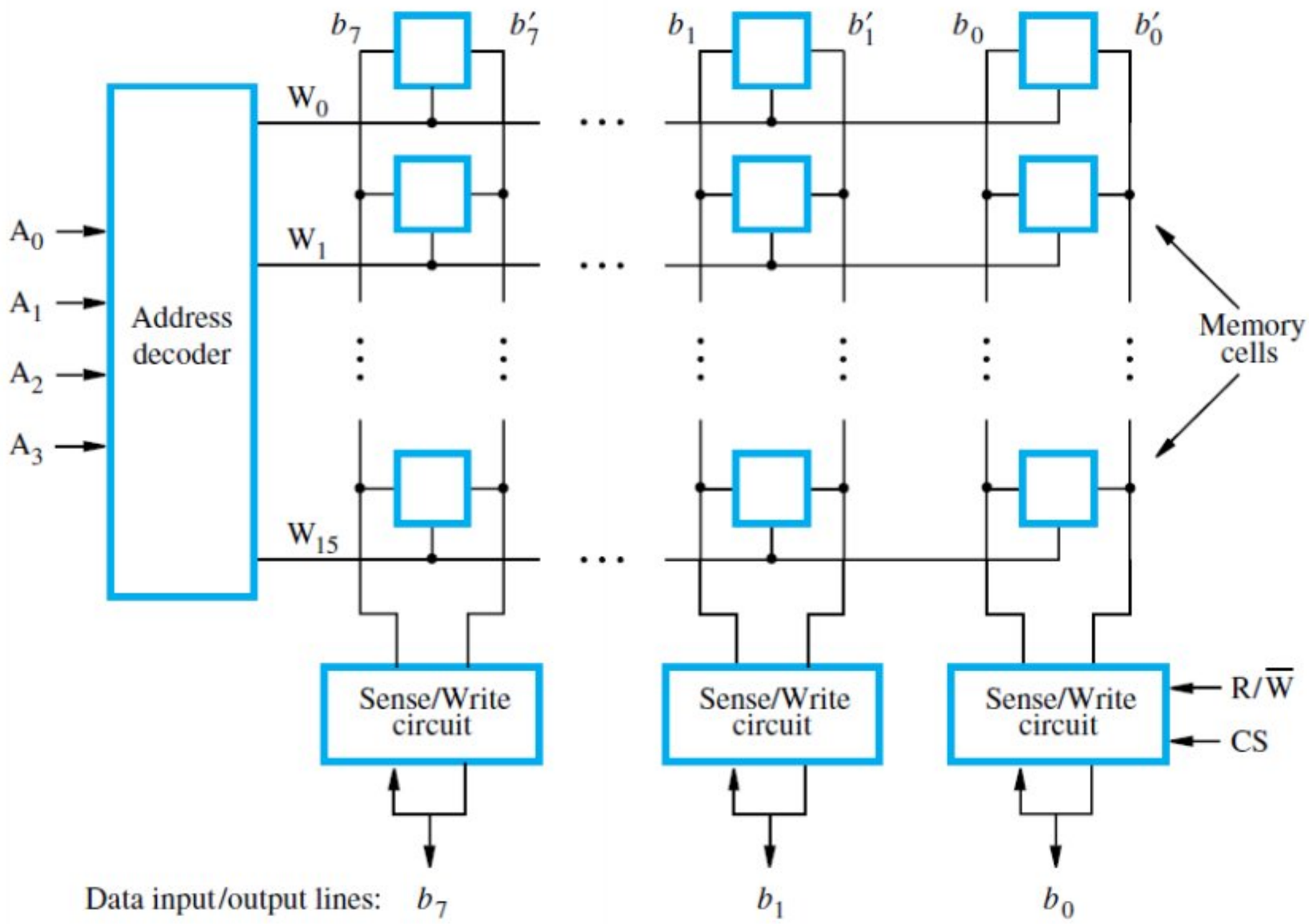


Fig: Organization of bit cells in a memory chip.

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**, and the Sense/Writecircuits 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** andplace this information on the output data lines. **During a Write operation, the Sense/Write circuits receive input data and store them in the cells of the selected word**.Above figureis an example of a very small memory circuit consisting of 16 words of 8 bitseach. This is referred to as a 16 × 8 organization. The data input and the data output of eachSense/Write circuit are connected to a single bidirectional data line that can be connectedto the data lines of a computer.Two control lines, R/W and CS, are provided. The R/W(Read/Write) input specifies the required operation, and the CS (Chip Select) input selectsa given chip in a multichip memory system.**The above memory circuit stores 128 bits and requires 14 external connections for address, data, and control lines.** It also needs two lines for power supply and groundconnections.

Consider now a slightly larger memory circuit, one that has 1K (1024) memorycells. This **circuit can be organized as a 128 × 8 memory, requiring a total of 19 external connections**. Alternatively, the same number of cells can be organized into a 1K×1 format.In this case, a 10-bit address is needed, but there is only one data line, resulting in 15 externalconnections. Figure below shows such an organization. The required 10-bit address is dividedinto two groups of 5 bits each to form the row and column addresses for the cell array. Arow address selects a row of 32 cells, all of which are accessed in parallel. But, only oneof these cells is connected to the external data line, based on the column address.For example, a 1Giga-bitchip may have a 256M × 4 organization, in which case a 28-bit address is needed and 4bits are transferred to or from the chip.

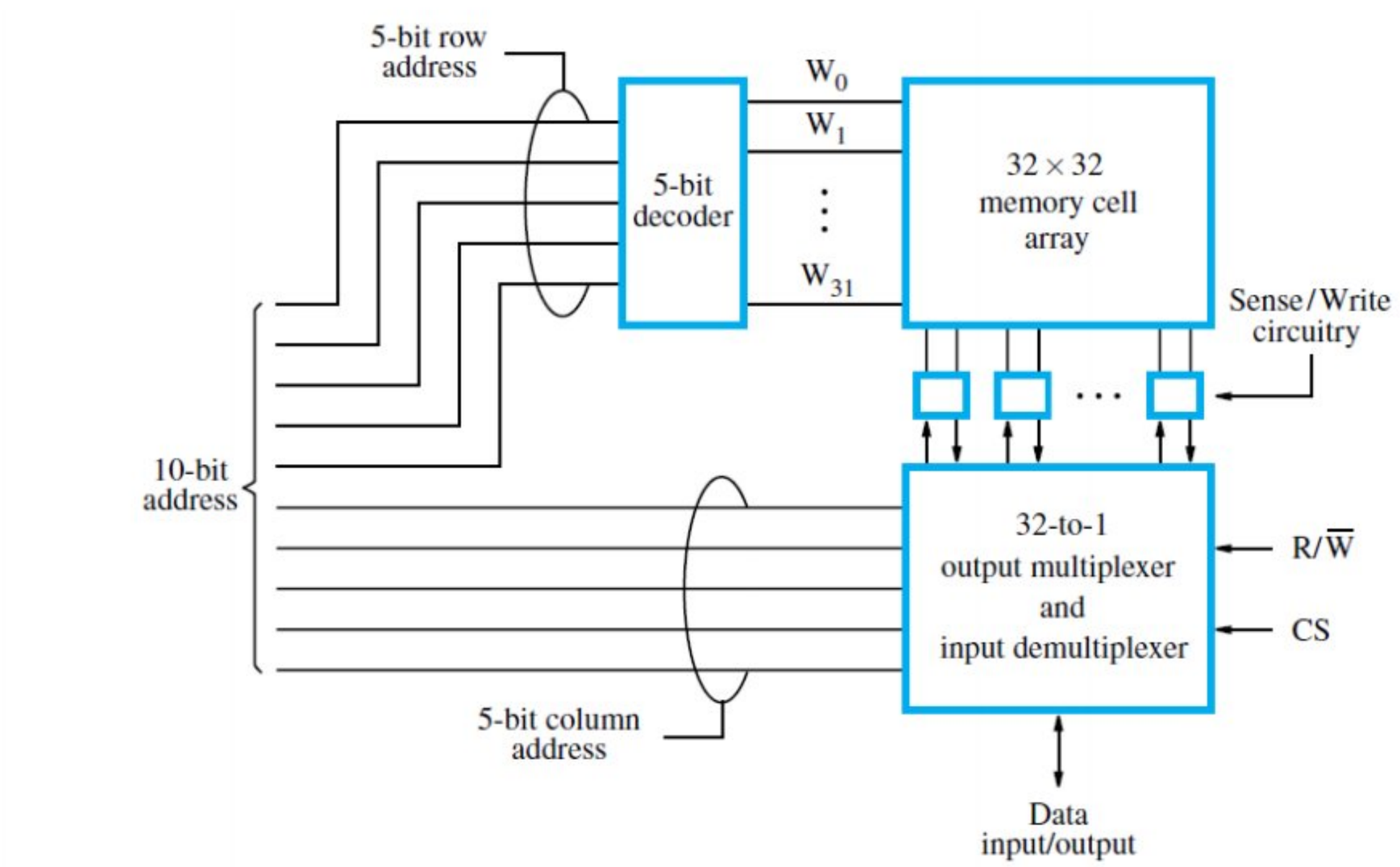


Fig: Organization of a 1K × 1 memory chip.

Memory Hierarchy

- The total memory capacity of a computer can be visualized as being a hierarchy of components. The memory hierarchy system consists of all storage devices employed in a computer system

from the slow but high-capacity secondary memory to a relatively faster main memory, to an even smaller and faster cache memory accessible to the high-speed processing logic.

- The purpose of any memory device is to store programs and data. Several types of memory devices are used in the computer forming a Memory Hierarchy. Each plays a specific role contributing to the speed, cost effectiveness, portability etc.

Main Memory

The memory unit that communicates directly with the CPU is called the main memory.It comprises of RAM and ROM, both are Semi-Conductor memories. (chip memories). ROM is non-volatile.It is used is storing permanent information like the BIOS program.It is typically of 2 MB - 4 MB in size.RAM is writable and hence is used for day-to-day operations.Every file that we access from secondary memory, is first loaded into RAM.The main memory occupies
a central position by being able to communicate directly with the CPU andwith auxiliary memory devices through an UO processor. To provide large amount of working space RAM is typically 4 GB - 8 GB.

Secondary Memory (Auxiliary Memory):

Devices that provide backup storage arecalled auxiliary memory. The most common auxiliary memory devices used incomputer systems are magnetic disks and tapes. They are used for storingsystemprograms, large data files, and other backup information. Only pro-grams and data currently needed by the processor reside in main memory.When programs notresiding in main memory are needed by the CPU, they are brought in fromauxiliary memory.The main purpose of Secondary Memory is to increase the storage capacity, at low cost.Its biggest component is the Hard Disk.It is writeable as well as non-volatile. Typical size of a HD is 1 TB.Disk memories are much slower than chip memories but are also much cheaper.

Cache Memory:

The cache memory is employed in computersystems to compensate for the speed differential between main memory accesstime and processor logic. CPU logic is usually faster than main memory accesstime, with the result that processing speed is limited primarily by the speedof main memory. A technique used to compensate for the mismatch in operating speeds is to employ an extremely fast, small cache between the CPU andmain memory whose access time Is dose to processor logic dock cycle time.It is the fastest form of memory as it uses SRAM (Static RAM).The Main Memory uses DRAM (Dynamic RAM).SRAM uses flip-flops and hence is much faster than DRAM which uses capacitors.But SRAM is also very expensive as compared to DRAM.Hence only the current portion of the file we need to access is copied from Main Memory (DRAM)to Cache memory (SRAM), to be directly accessed by the processor.This gives maximum performance and yet keeps the cost low.While the VO processor manages data transfers between auxiliary memoryand main memory, the cache organization is concerned with the transferof information between main memory and CPU.Typical size of Cache is around 2 MB – 8MB. If code and data are in the same cache then it is unified cache else its called split cache.Depending upon the location of cache, it is of three types: L1, L2 and L3.L1 cache is present inside the processor and is a split cache typically 4-8 KB.L2 is present on the same die as the processor and is a unified cache typically 1 MB.L3 is present outside the processor. It is also unified and is typically of 2-8 MB.

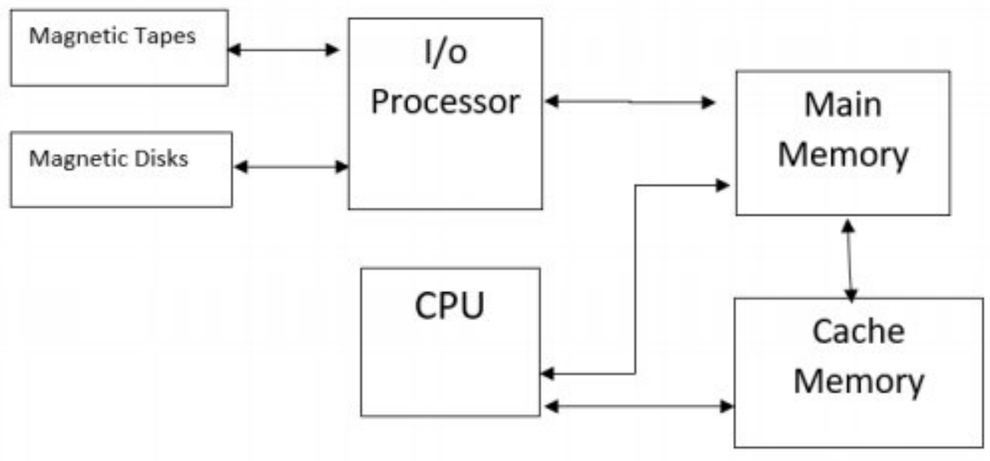


Fig: Memory Hierarchy in a computer system

The reason for having twoor three levels of memory hierarchy is economics. As the storage capacity ofthe memory increases, the cost per bit for storing binary information decreasesand the access time of the memory becomes longer. The auxiliary memory hasa large storage capacity, is relatively inexpensive, but has low access speedcompared to main memory. The cache memory is very small, relatively expensive,and has very high access speed. Thus as the memory access speedincreases, so does its relative cost. The overall goal of using a memory hierarchyis to obtain the highest-possible average access speed while minimizing thetotal cost of the entire memory system.

Memory Interleaving

Pipeline and vector processors often require simultaneous access to memory from two or more sources. An instruction pipeline may require the fetching of an instruction and an operand at the same time from two different segments. Similarly, an arithmetic pipeline usually requires two or more operands to enter the pipeline at the same time. Instead of using two memory buses for simultaneous access, the memory can be partitioned into a number of modules connected to a common memory address and data buses. A memory module is a memory array together with its own address and data registers. Figure below shows a memory unit with four modules.

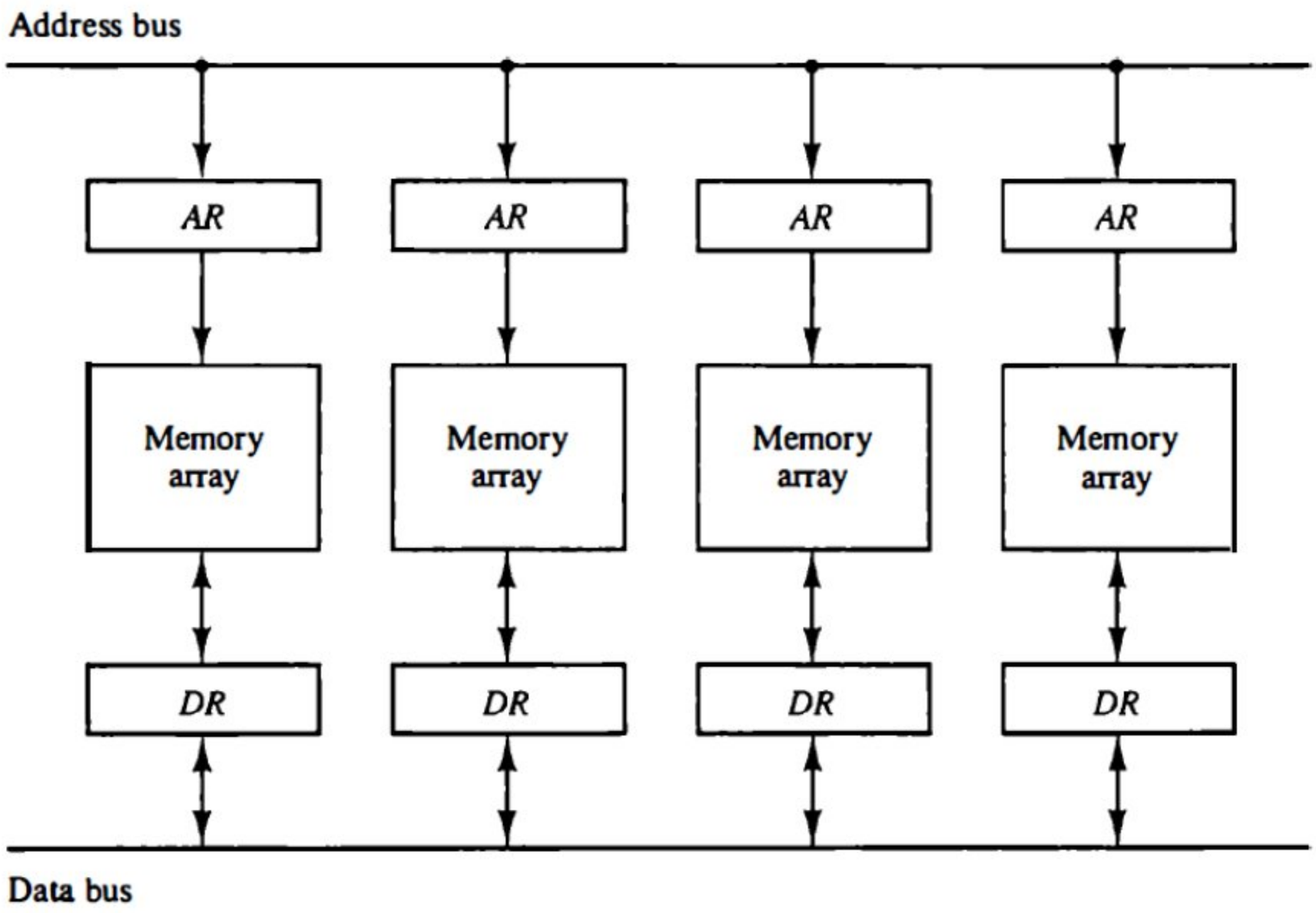


Fig: Multiple module memory organization.

Each memory array has its ownaddress register AR and data register DR . The address registers receive informationfrom a common address bus and the data registers communicate witha bidirectional data bus. The two least significant bits of the address can be usedto distinguish between the four modules. The modular system permits onemodule to initiate a memory access while other modules are in the process ofreading or writing a word and each module can honor a memory requestindependent of the state of the other modules.The advantage of a modular memory is that it allows the use of a techniquecalled interleaving. In an interleaved memory, different sets of addressesare assigned to different memory modules. For example, in a two-modulememory system, the even addresses may be in one module and the oddaddresses in the other. When the number of modules is a power of 2, the leastsignificant bits of the address select a memory module and the remaining bitsdesignate the specific location to be accessed within the selected module.A modular memory is useful in systems with pipeline and vector processing.A vector processor that uses an n-way interleaved memory can fetch noperands from n different modules. By staggering the memory access, theeffective memory cycle time can be reduced by

a factor close to the number ofmodules. A CPU with instruction pipeline can take advantage of multiplememory modules so that each segment in the pipeline can access memoryindependent of memory access from other segments.

Cache Memory:

The cache is a small and very fast memory, interposed between the processor and the mainmemory. Its purpose is to make the main memory appear to the processor to be muchfaster than it actually is.

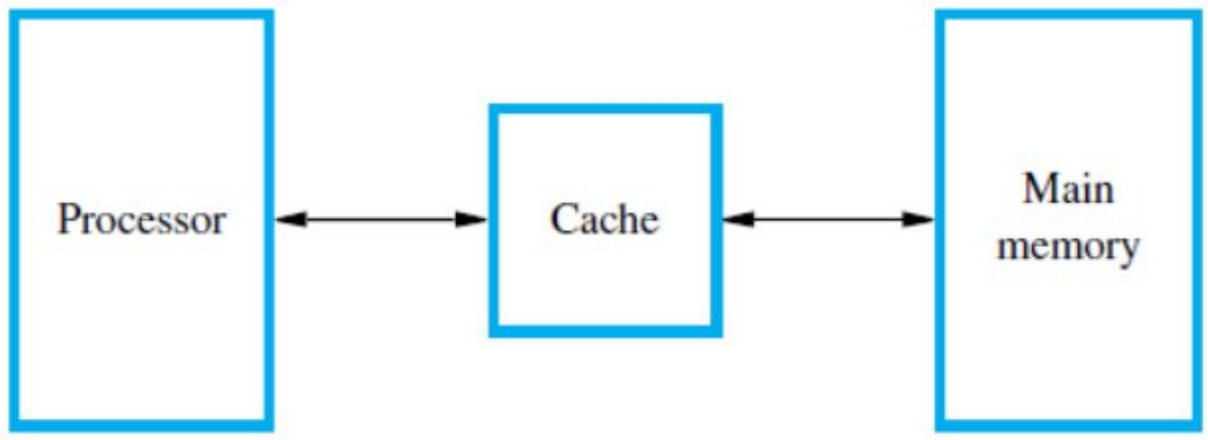


Fig: Use of Cache Memory

operation of a cache memory is very simple. The memory control circuitry is designed to take advantage of the property of locality of reference. Temporal locality suggests that whenever an information item, instruction or data, is first needed, this item should be brought into the cache, because it is likely to be needed again soon. Spatial locality suggests that instead of fetching just one item from the main memory to the cache, it is useful to fetch several items that are located at adjacent addresses as well. The termcache 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 a cache line.

When the processor issues a Read request, the contents of a block of memory words containing the location specified are transferred into the cache. 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 cache's replacement algorithm.

Cache Hits:

The processor does not need to know explicitly about the existence of the cache. It simply issues Read andWrite 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. The main memory is not involved when there is a cache hit in a Read operation. For a Write operation, the system can proceed in one of two ways. In the first technique, called the write-through protocol, both the cache location and the main memory location are updated. The second technique is to update only the cache location and to mark the block containing it 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 removed from the cache to make room fora new block. This technique is known as the write-back, or copy-back, protocol.

The write-through protocol is simpler than the write-back protocol, but it results in unnecessary Write operations in the main memory when a given cache word is updated several times during its cache residency. The write-back protocol also involves unnecessary Write operations, because all words of the block are eventually written back, even if only a single word has been changed while the block was in the cache. The write-back protocol is used most often, to take advantage of the high speed with which data blocks can be transferred to memory chips.

Cache Misses

A Read operation for a word that is not in the cache constitutes a Read miss. It causes the block of words containing the requested word to be 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 time somewhat, at the expense of more complex circuitry.

When a Write miss occurs in a computer that uses the write-through protocol, the information is written directly into the main memory. For 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.