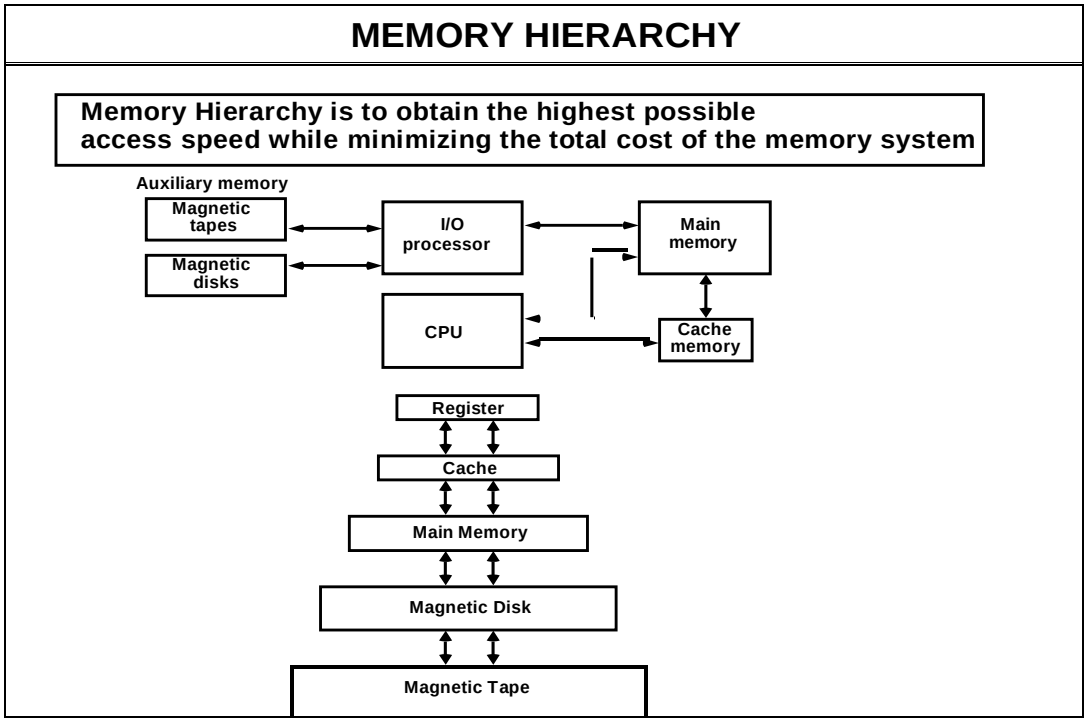
MEMORY ORGANIZATION

Contents: Memory Hierarchy, Main Memory, Auxiliary memory,

Associative Memory, Cache Memory, Virtual Memory

Memory Hierarchy :



memory address map of RAM and ROM.

Main Memory

- The main memory is the central storage unit in a computer system.
- Primary memory holds only those data and instructions on which computer is currently working.
- It has limited capacity and data is lost when power is switched off.
- It is generally made up of semiconductor device.
- These memories are not as fast as registers.
- The data and instruction required to be processed reside in main memory.
- It is divided into two subcategories RAM and ROM.

Memory address map of RAM and ROM

- The designer of a computer system must calculate the amount of memory required for the particular application and assign it to either RAM or ROM.
- The interconnection between memory and processor is then established from knowledge of the size of memory needed and the type of RAM and ROM chips available.
- The addressing of memory can be established by means of a table that specifies the memory address assigned to each chip.

- The table, called a **memory address map**, is a pictorial representation of assigned address space for each chip in the system, shown in table 9.1.

- To demonstrate with a particular example, assume that a computer system needs 512 bytes of RAM and 512 bytes of ROM.

- The RAM and ROM chips to be used are specified in figure 1 and figure 2.

Memory address map of RAM and ROM

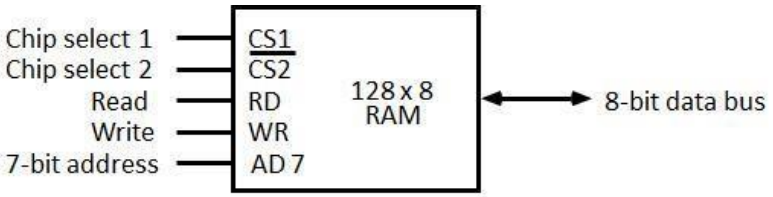


Figure 1: Typical RAM chip

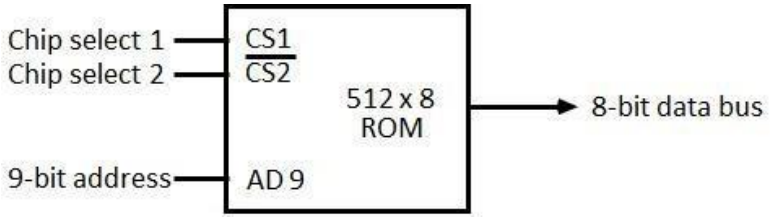


Figure 2: Typical ROM chip

Component	Hexa address	Address bus 10 9 8 7 6 5 4 3 2 1
RAM 1	0000 - 007F	0 0 0 x x x x x x x
RAM 2	0080 - 00FF	0 0 1 x x x x x x x
RAM 3	0100 - 017F	0 1 0 x x x x x x x
RAM 4	0180 - 01FF	0 1 1 x x x x x x x
ROM	0200 - 03FF	1 x x x x x x x x x

- The component column specifies whether a RAM or a ROM chip is used.

- The hexadecimal address column assigns a range of hexadecimal equivalent addresses for each chip.

- The address bus lines are listed in the third column.

- Although there are 16 lines in the address bus, the table shows only 10 lines because the other 6 are not used in this example and are assumed to be zero.

- The small x's under the address bus lines designate those lines that must be connected to the address inputs in each chip.

- The RAM chips have 128 bytes and need seven address lines. The ROM chip has 512 bytes and needs 9 address lines.

- The x's are always assigned to the low-order bus lines: lines 1 through 7 for the RAM and lines 1

through 9 for the ROM.

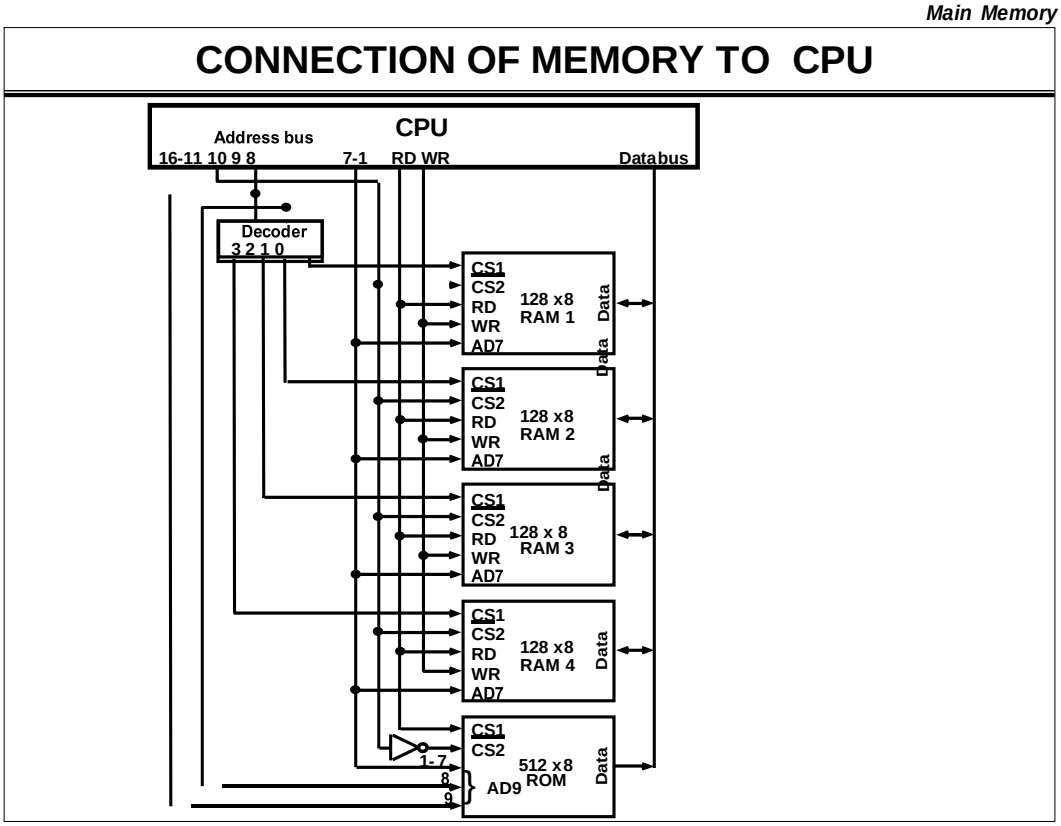
- It is now necessary to distinguish between four RAM chips by assigning to each a different address. For this particular example we choose bus lines 8 and 9 to represent four distinct binary combinations.
- The table clearly shows that the nine low-order bus lines constitute a memory space for RAM equal to $2^9 = 512$ bytes.
- The distinction between a RAM and ROM address is done with another bus line. Here we choose line 10 for this purpose.

When line 10 is 0, the CPU selects a RAM, and when this line is equal to 1, it selects the ROM

Memory connections to CPU :

- RAM and ROM chips are connected to a CPU through the data and address buses

- The low-order lines in the address bus select the byte within the chips and other lines in the address bus select a particular chip through its chip select inputs.



Auxiliary Memory :

- **Magnetic Tape:** Magnetic tapes are used for large computers like mainframe computers where large volume of data is stored for a longer time. In PC also you can use tapes in the form of cassettes. The cost of storing data in tapes is inexpensive. Tapes consist of magnetic materials that store data permanently. It can be 12.5 mm to 25 mm wide plastic film-type and 500 meter to 1200 meter long which is coated with magnetic material. The deck is connected to the central processor and information is fed into or read from the tape through the processor. It's similar to cassette tape recorder.

Magnetic tape is an information storage medium consisting of a magnetisable coating on a thin plastic strip. Nearly all recording tape is of this type, whether used for video with a video cassette recorder, audio storage (reel-to-reel tape, compact audio cassette, digital audio tape (DAT), digital linear tape (DLT) and other formats including 8-track cartridges) or general purpose digital data storage using a computer (specialized tape formats, as well as the above-mentioned compact audio cassette, used with home computers of the 1980s, and DAT, used for backup in workstation installations of the 1990s).

- Magneto-optical and optical tape storage products have been developed using many of the same concepts as magnetic storage, but have achieved little commercial success.

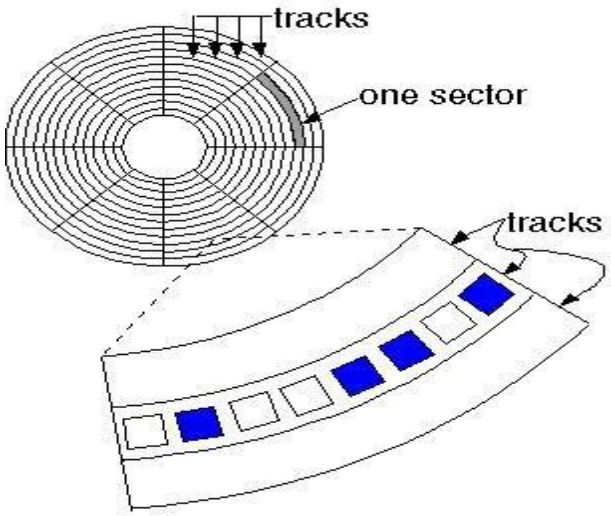
- **Magnetic Disk:** You might have seen the gramophone record, which is circular like a disk and coated with magnetic material. Magnetic disks used in computer are made on the same principle. It rotates with very high speed inside the computer drive. Data is stored on both the surface of the disk. Magnetic disks are most popular for direct access storage device. Each disk consists of a number of invisible concentric circles called tracks. Information is recorded on tracks of a disk surface in the form of tiny magnetic spots. The presence of a magnetic spot represents one bit and its absence represents zero bit. The information stored in a disk can be read many times without affecting the stored data. So the reading operation is non-destructive. But if you want to write a new data, then the existing data is erased from the disk and new data is recorded. For Example-Floppy Disk.

The primary computer storage device. Like tape, it is magnetically recorded and can be re-recorded over and over. Disks are rotating platters with a mechanical arm that moves a read/write head between the outer and inner edges of the platter's surface. It can take as long as one second to find a location on a floppy disk to as little as a couple of milliseconds on a fast hard disk. See hard disk for more details.

The disk surface is divided into concentric tracks (circles within circles). The thinner the tracks, the more storage. The data bits are recorded as tiny magnetic spots on the tracks. The smaller the spot, the more bits per inch and the greater the storage.

Sectors

Tracks are further divided into sectors, which hold a block of data that is read or written at one time; for example, READ SECTOR 782, WRITE SECTOR 5448. In order to update the disk, one or more sectors are read into the computer, changed and written back to disk. The operating system figures out how to fit data into these fixed spaces. Modern disks have more sectors in the outer tracks than the inner ones because the outer radius of the platter is greater than the inner radius



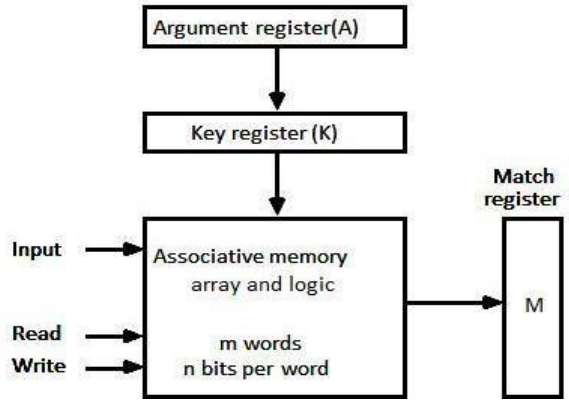
Block diagram of Magnetic Disk

Optical Disk: With every new application and software there is greater demand for memory capacity. It is the necessity to store large volume of data that has led to the development of optical disk storage medium. Optical disks can be divided into the following categories:

1. **Compact Disk/ Read Only Memory (CD-ROM**
2. **Write Once, Read Many (WORM)**
3. **Erasable Optical Disk**

Associative Memory :Content Addressable Memory (CAM).

- The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address.
- A memory unit accessed by content is called an associative memory or content addressable memory (CAM).
- This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.
- The block diagram of an associative memory is shown in figure 9.3.



- It consists of a memory array and logic form words with n bits per word.
- The argument register A and key register K each have n bits, one for each bit of a word.
- The match register M has m bits, one for each memory word.
- Each word in memory is compared in parallel with the content of the argument register.
- The words that match the bits of the argument register set a corresponding bit in the match register.
- After the matching process, those bits in the match register that have been set indicate the fact that their corresponding words have been matched.
- Reading is accomplished by a sequential access to memory for those words whose corresponding bits in the match register have been set.

Hardware Organization

- The key register provides a mask for choosing a particular field or key in the argument word.
- The entire argument is compared with each memory word if the key register contains all 1's.

- Otherwise, only those bits in the argument that have 1st in their corresponding position of the key register are compared.
- Thus the key provides a mask or identifying piece of information which specifies how the reference to memory is made.
- To illustrate with a numerical example, suppose that the argument register A and the key register K have the bit configuration shown below.
- Only the three leftmost bits of A are compared with memory words because K has 1's in these position.

A	101 111100	
K	111 000000	
Word1	100 111100	no match
Word2	101 000001	match

- Word 2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.

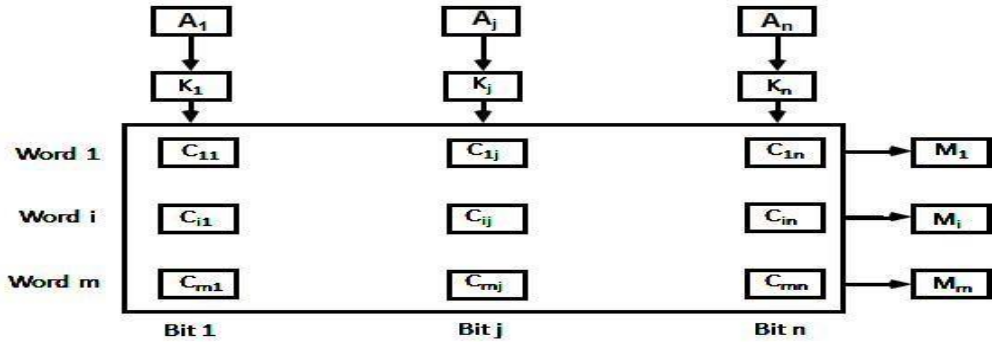


Figure 4: Associative memory of *m* word, n cells per word.

- The relation between the memory array and external registers in an associative memory is shown in figure 4.
- The cells in the array are marked by the letter C with two subscripts.
- The first subscript gives the word number and the second specifies the bit position in the word.
- Thus cell Cij is the cell for bit j in words i.
- A bit Aj in the argument register is compared with all the bits in column j of the array provided that Kj =1.
- This is done for all columns j = 1, 2... n.
- If a match occurs between all the unmasked bits of the argument and the bits in word i, the corresponding bit Mi in the match register is set to 1.
- If one or more unmasked bits of the argument and the word do not match, Mi is cleared to 0.

Cache Memory :

- Cache is a fast small capacity memory that should hold those information which are most likely to be accessed.

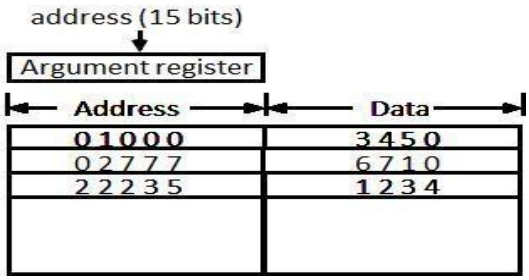
- The basic operation of the cache is, when the CPU needs to access memory, the cache is examined.
- If the word is found in the cache, it is read from the fast memory. If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word.
- The transformation of data from main memory to cache memory is referred to as a **mapping process**.

Associative mapping

- Consider the main memory can store 32K words of 12 bits each.
- The cache is capable of storing 512 of these words at any given time.
- For every word stored in cache, there is a duplicate copy in main memory.
- The CPU communicates with both memories.
- It first sends a 15-bit address to cache. If there is a hit, the CPU accepts the 12-bit data from cache, if there is miss, the CPU reads the word from main memory and the word is then transferred to cache.

Figure 5: Associative mapping cache

(all numbers in octal)



- The associative memory stores both the address and content (data) of the memory word.
- This permits any location in cache to store any word from main memory.
- The figure 9.5 shows three words presently stored in the cache. The address value of 15 bits is shown as a five-digit octal number and its corresponding 12-bit word is shown as a four-digit octal number.
- A CPU address of 15 bits is placed in the argument register and the associative memory is searched for a matching address.
- If the address is found the corresponding 12-bit data is read and sent to CPU.
- If no match occurs, the main memory is accessed for the word.
- The address data pairs then transferred to the associative cache memory.
- If the cache is full, an address data pair must be displaced to make room for a pair that is needed and not presently in the cache.
- This constitutes a first-in first-one (FIFO) replacement policy.

direct mapping in organization of cache memory:

- The CPU address of 15 bits is divided into two fields.
- The nine least significant bits constitute the index field and the remaining six bits from the tag field.
- The figure 6 shows that main memory needs an address that includes both the tag and the index.

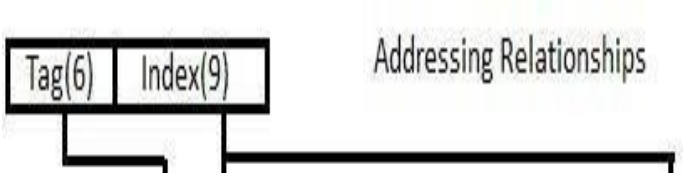


Figure 6: Addressing relationships between main and cache memories

- The number of bits in the index field is equal to the number of address bits required to access the cache memory.
- The internal organization of the words in the cache memory is as shown in figure 7.

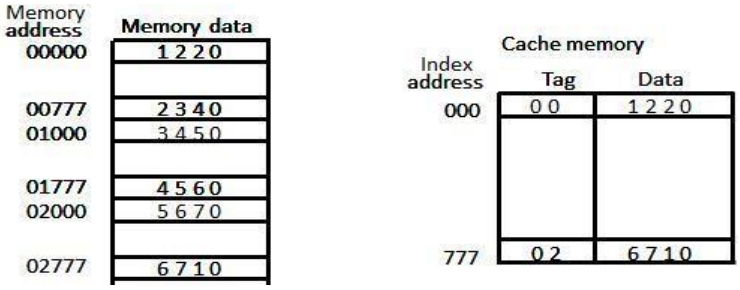


Figure 7: Direct mapping cache organization

- Each word in cache consists of the data word and its associated tag.
- When a new word is first brought into the cache, the tag bits are stored alongside the data bits.
- When the CPU generates a memory request the index field is used for the address to access the cache.
- The tag field of the CPU address is compared with the tag in the word read from the cache.
- If the two tags match, there is a hit and the desired data word is in cache.
- If there is no match, there is a miss and the required word is read from main memory.
- It is then stored in the cache together with the new tag, replacing the previous value.
- The word at address zero is presently stored in the cache (index = 000, tag = 00, data = 1220). Suppose that the CPU now wants to access the word at address 02000.

  The index address is 000, so it is used to access the cache. The two tags are then compared.

  The cache tag is 00 but the address tag is 02, which does not produce a match.

  Therefore, the main memory is accessed and the data word 5670 is transferred to the CPU.

  The cache word at index address 000 is then replaced with a tag of 02 and data of 5670.

  The **disadvantage** of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time.

The comparison logic is done by an associative search of the tags in the set similar to an associative memory search: thus the name "set-associative".

When a miss occurs in a set-associative cache and the set is full, it is necessary to replace one of the tag-data items with a new value.

The most common replacement algorithms used are: random replacement, first-in first-out (FIFO), and least recently used (LRU).

Write-through and Write-back cache write method.

Write Through

- The simplest and most commonly used procedure is to update main memory with every memory write operation.
- The cache memory being updated in parallel if it contains the word at the specified address. This is called the *write-through* method.
- This method has the advantage that main memory always contains the same data as the cache.
- This characteristic is important in systems with direct memory access transfers.
- It ensures that the data residing in main memory are valid at all times so that an I/O device communicating through DMA would receive the most recent updated data.

Write-Back (Copy-Back)

- The second procedure is called the write-back method.
- In this method only the cache location is updated during a write operation.
- The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory.
- The reason for the write-back method is that during the time a word resides in the cache, it may be updated several times.
- However, as long as the word remains in the cache, it does not matter whether the copy in main memory is out of date, since requests from the word are filled from the cache.
- It is only when the word is displaced from the cache that an accurate copy need be rewritten into main memory.

Virtual Memory

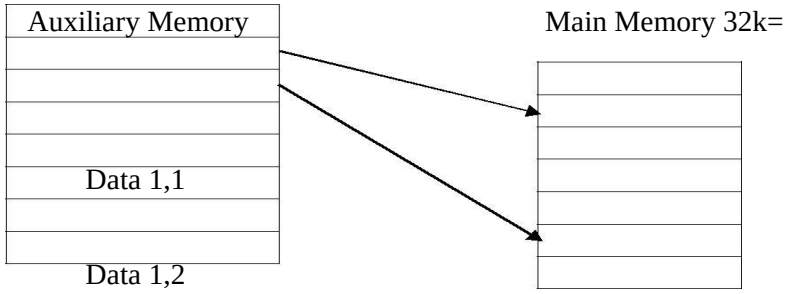
- Virtual memory is used to give programmers the illusion that they have a very large memory at their disposal, even though the computer actually has a relatively small main memory.
- A virtual memory system provides a mechanism for translating program-generated addresses into correct main memory locations.

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.



Address space $1024k=2^{10}$

- 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 $32K = 2^{15}$.
- Suppose that the computer has available auxiliary memory for storing $2^{20} = 1024K$ 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 9.9.
- 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.

Address mapping using pages.

- AThe 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 9.9
- At any given time, up to four pages of address space may reside in main memory in any one of the four blocks.

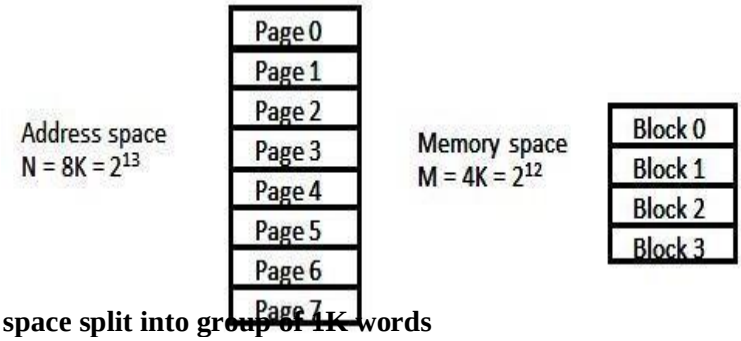


Figure 10 Address and Memory space split into group of 1K words

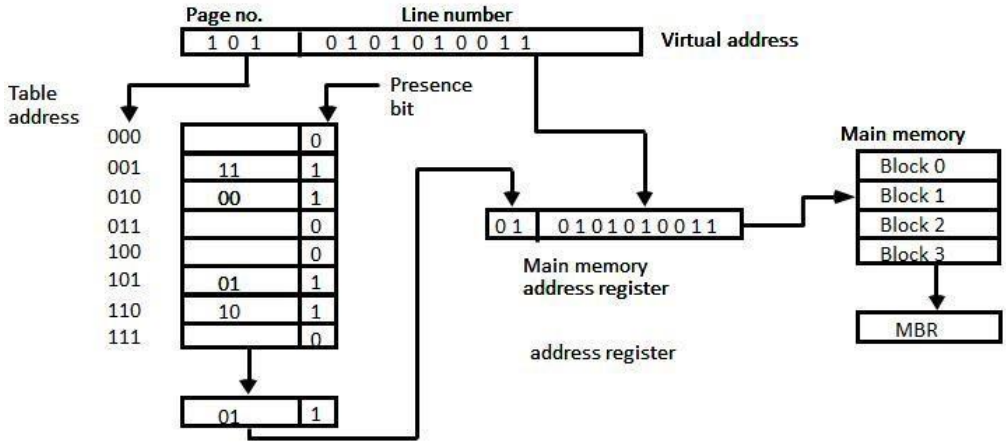


Figure 11: Memory table in paged system

- The organization of the memory mapping table in a paged system is shown in figure 10.
- 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.

Segment

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

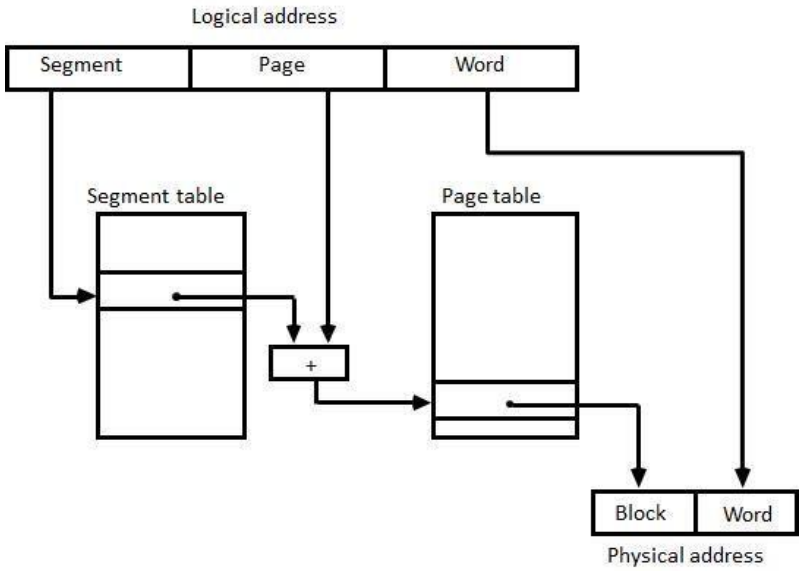
Logical address

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 9.12.

Figure 9 Logical to physical address mapping



- 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 12.
- 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.