Unit-5
Memory Organization

By- Yadvir Kaur

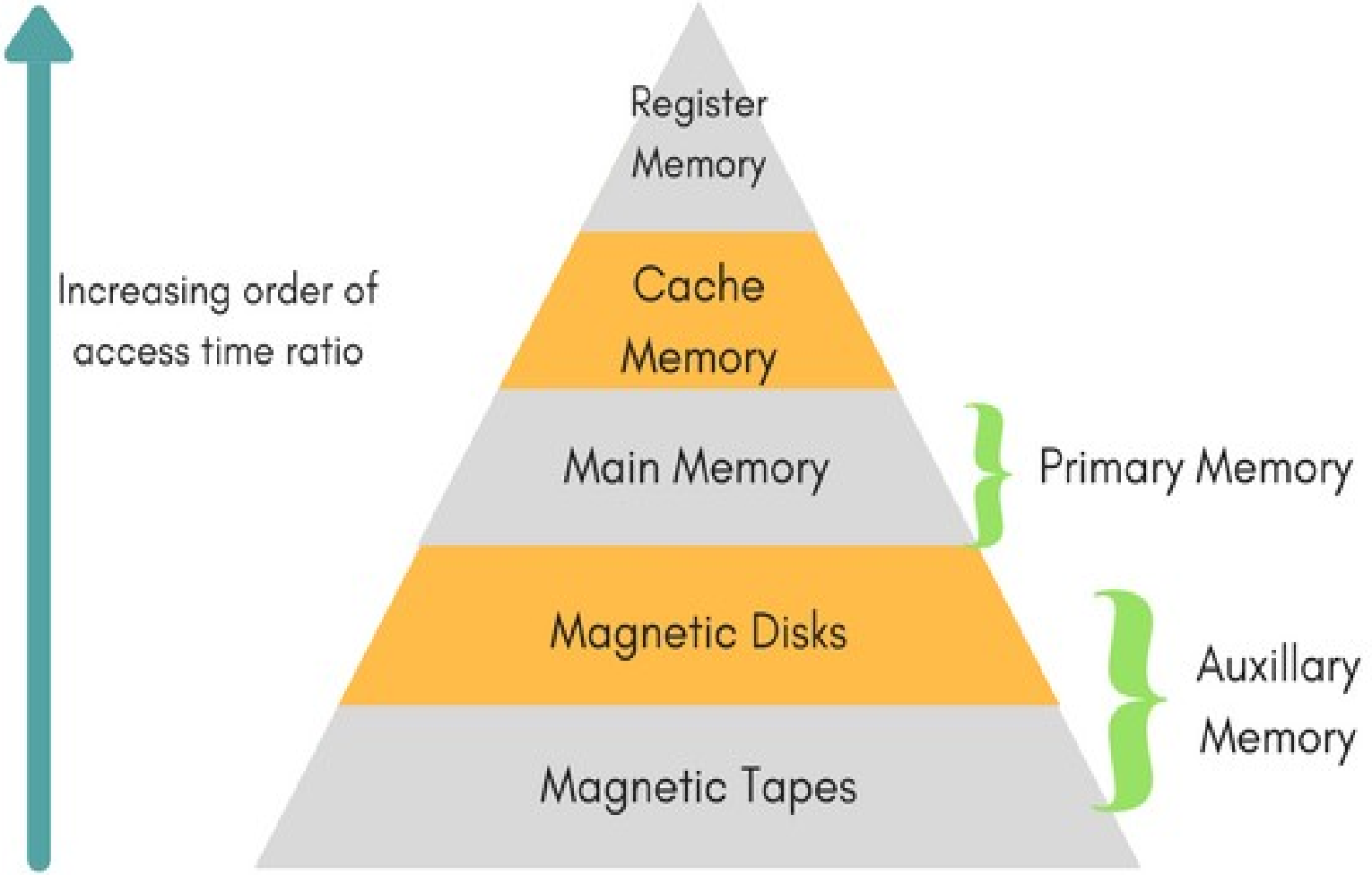
Contents

- Memory Hierarchy
- Main Memory
- Auxiliary Memory
- Associative Memory
- Cache Memory
- Virtual Memory

Memory Hierarchy

- The memory unit is an essential component in any digital computer since it is needed for storing programs and data.
- There is just not enough space in one memory unit to accommodate all the programs used in a typical computer.
- Not all accumulated information is needed by the CPU at the same time.
- Therefore, it is more economical to use low-cost storage devices to serve as a backup for storing the information that is not currently used by CPU.
- **Main memory:** The memory unit that directly communicate with CPU.
- **Auxiliary memory**: devices that provide backup storage. For example: magnetic disks and tapes. They are used for storing system programs, large data files, and other backup information.
- Only programs and data currently needed by the processor reside in main memory. All other information Is stored in auxiliary memory and transferred to main memory when needed.

- The memory hierarchy system consists of all storage devices employed in a computer system from the slow but high-capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory.



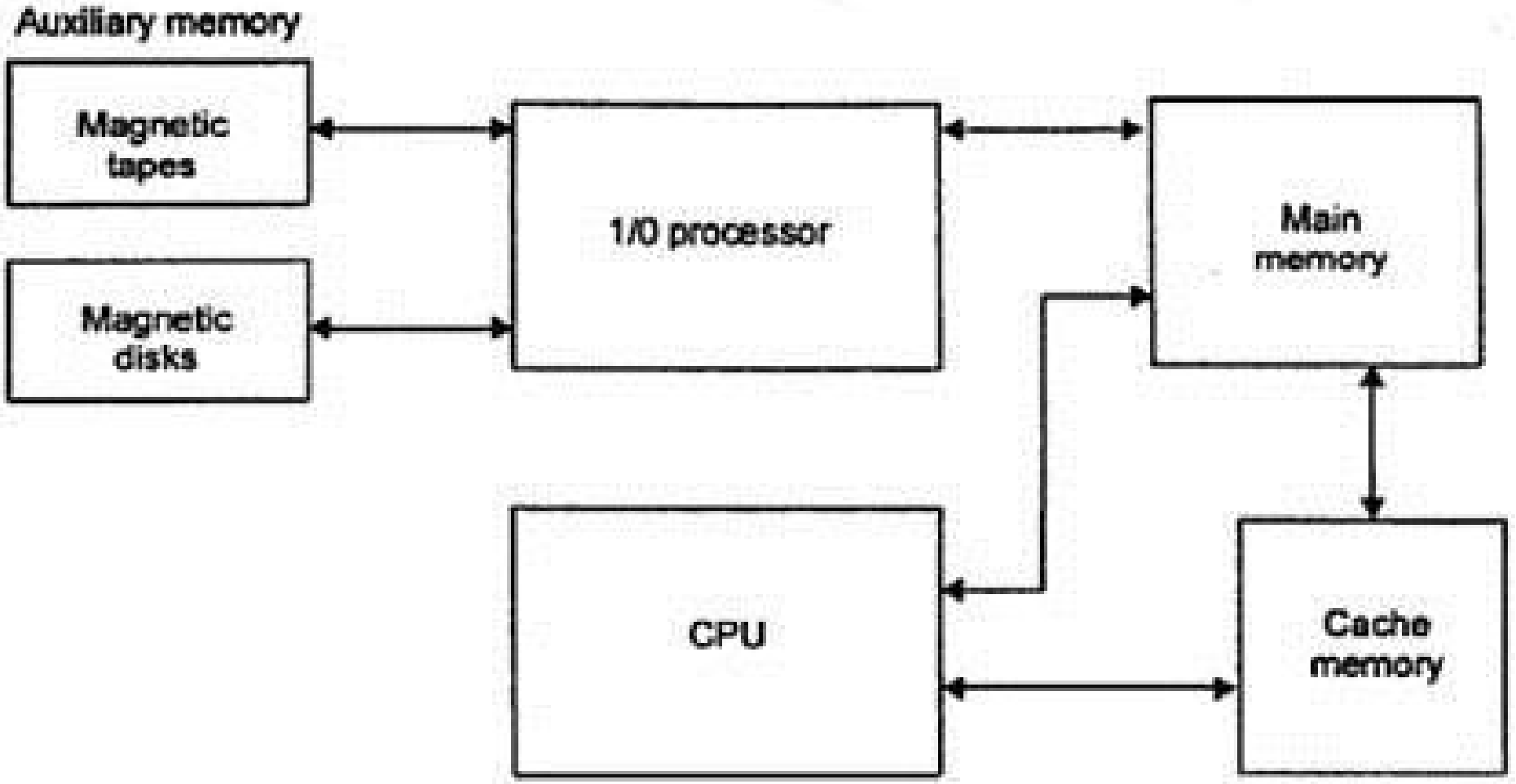


Fig 12.1 Memory Hierarchy in a computer system

- The **main memory** occupies a central position by being able to communicate directly with the CPU and with auxiliary memory devices through an I/O processor.
- When programs not residing in main memory are needed by the CPU, they are brought in from auxiliary memory. Programs not currently needed in main memory are transferred into auxiliary memory to provide space for currently used programs and data.
- The part of the computer system that supervises the flow of information between auxiliary memory and main memory is called the **memory management system.**
- A special very-high-speed memory called **cache** is used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate.
- **CPU logic is usually faster than main memory access time. The cache memory is employed in computer systems to compensate for the speed differential between main memory access time and processor(CPU) logic.**
- **The cache is used for storing segments of programs currently being executed in the CPU and temporary data frequently needed in the present calculations.**

- While the I/O processor manages data transfers between auxiliary memory and main memory, the cache organization is concerned with the transfer of information between main memory and CPU.
- Thus each is involved with a different level in the memory hierarchy system.
- The reason for having two or three levels of memory hierarchy is economics.
- **As the storage capacity of the memory increases, the cost per bit for storing binary information decreases and the access time of the memory becomes longer.**
- The auxiliary memory has a large storage capacity, is relatively inexpensive, but has low access speed compared to main memory.
- The cache memory is very small, relatively expensive, and has very high access speed.
- **Thus, the overall goal of using a memory hierarchy is to obtain the highest-possible average access speed while minimizing the total cost of the entire memory system.**

Main Memory

The main memory is the central storage unit in a computer system.

Most of the main memory in a general purpose computer is made up of RAM integrated circuits chips, but a portion of the memory may be constructed with ROM chips

RAM– Random Access memory: Integated RAM are available in two possible operating modes, S*tatic and Dynamic.*

ROM– Read Only memory.

RAM (Random Access memory)

RAM is used for storing bulk of programs and data that is subject to change.

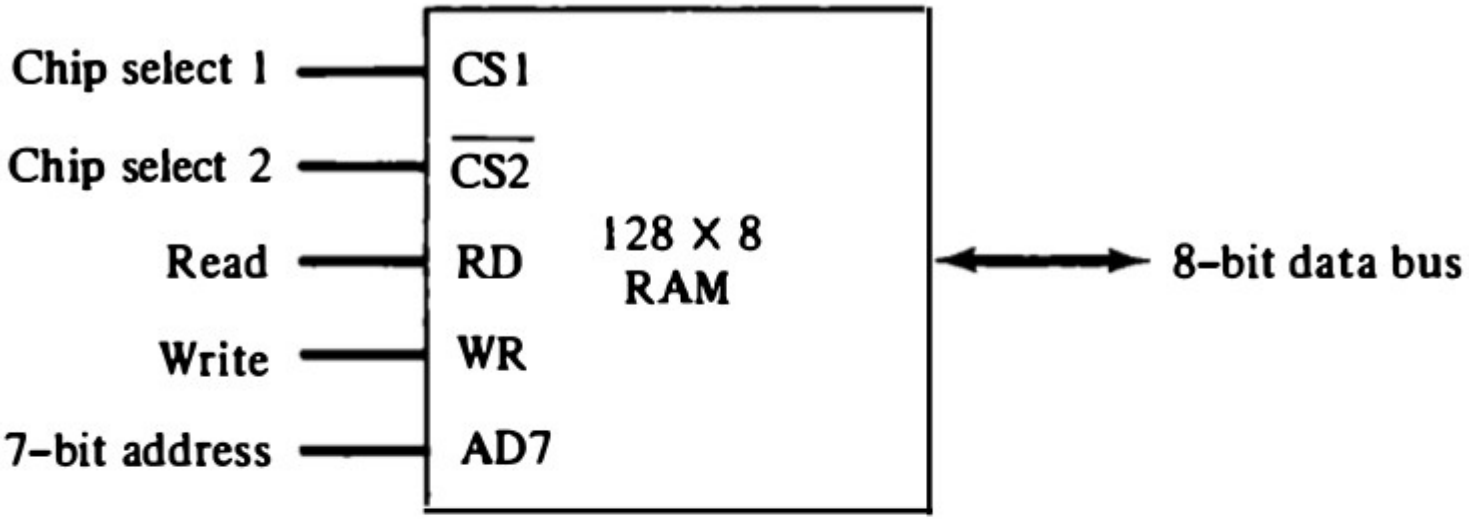
Static RAM (SRAM) :

- Each cell stores bit with a six-transistor circuit.
- Retains value indefinitely, as long as it is kept powered.
- Faster and more expensive than DRAM.
- **SRAMs are used for implementing the cache memories.**

Dynamic RAM (DRAM) :

- Each cell stores bit with a capacitor and transistor.
- Value must be refreshed every 10-100 ms.
- Slower and cheaper than SRAM. Has reduced power consumption, and a large storage capacity.
- **DRAMs are used for implementing the main memory**

Figure 12-2 Typical RAM chip.



(a) Block diagram

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

(b) Function table

- The block diagram of a RAM chip is shown in Fig. 12-2.
- **Bidirectional data bus that allows the transfer of data either from memory to CPU during a read operation, or from CPU to memory during a write operation.**
- The capacity of the memory is 128 words of eight bits (one byte) per word. This requires a 7-bit address and an 8-bit bidirectional data bus.
- The read and write inputs specify the memory operation and the two chips select (CS) control inputs are for enabling the chip only when it is selected by the microprocessor.
- The function table listed in Fig. 12-2(b) specifies the operation of the RAM chip.
- The unit is in operation only when CS1 = 1 and CS2= 0. The bar on tor of the second select variable indicates that this input is enabled when it is = 0. If the chip select inputs are not enabled, or if they are enabled but the read or write inputs are not enabled, the memory is inhibited and its data bus is in a high-impedance state.
- **When CS1 = 1 and CS2 = 0, the memory can be placed in a write or read mode.**
- When the WR input is enabled, the memory stores a byte from the data bus into a location specified by the address input lines. When the RD input is enabled, the content of the selected byte is placed into the data bus.

ROM (Read Only memory)

- **It is non-volatile memory, which retains the data even when power is removed from this memory. Programs and data that can not be altered are stored in ROM.**
- ROM is used for storing programs that are **PERMANENTLY** resident in the computer and for tables of constants that do not change in value once the production of the computer is completed.
- The ROM portion of main memory is needed for storing an initial program called **bootstrap loader.** The bootstrap loader is a program whose function is to start the computer operating system when power is turned on.

- Since the ROM can only READ, the data bus can only be in output mode.
- No need of READ and WRITE control.
- The two chip select inputs must be CS1 = 1 and CS2 = 0 for the unit to operate. Thus when the chip is enabled by the two select inputs, the byte selected by the address lines appears on the data bus.

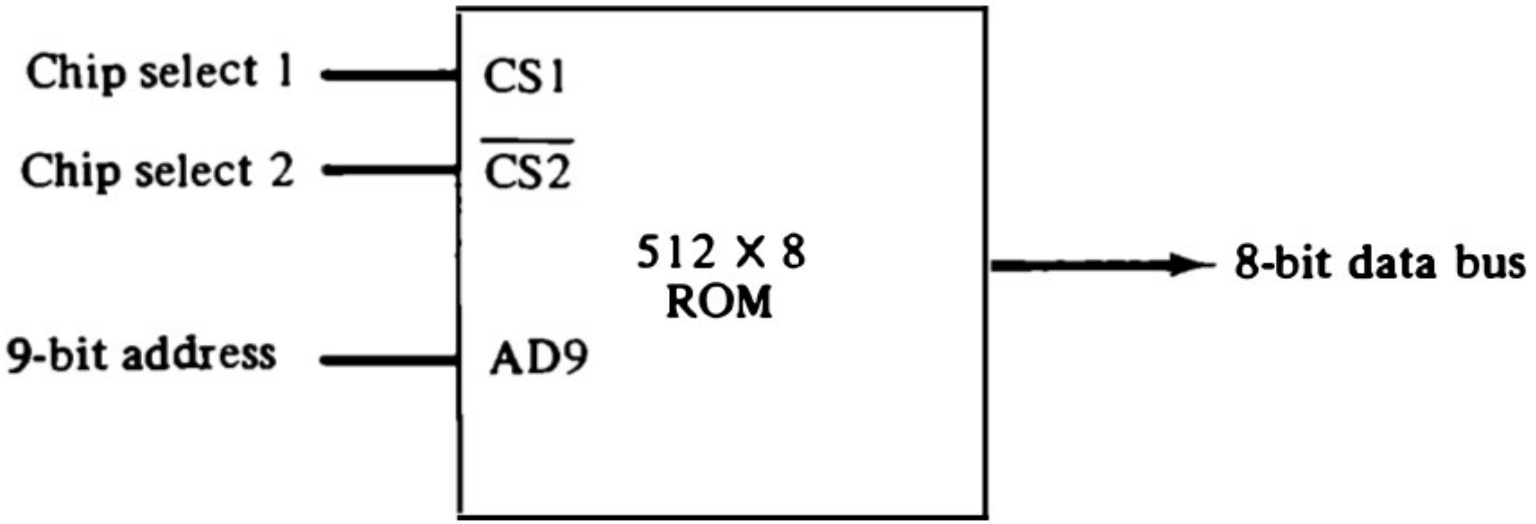


Figure 12-3 Typical ROM chip.

Auxiliary Memory

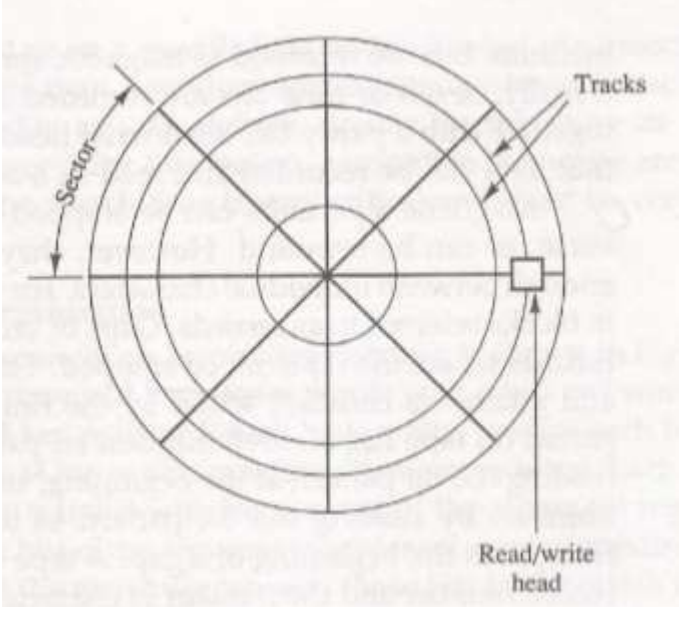
- Also called as Secondary Memory, used to store large chunks of data at a lesser cost per byte than a primary memory for backup.
- It does not lose the data when the device is powered down—it is **non-volatile.**
- It is not directly accessible by the CPU, they are accessed via the input/output channels.
- The most common form of auxiliary memory devices used in consumer systems is flash memory, optical discs, and magnetic disks, magnetic tapes.

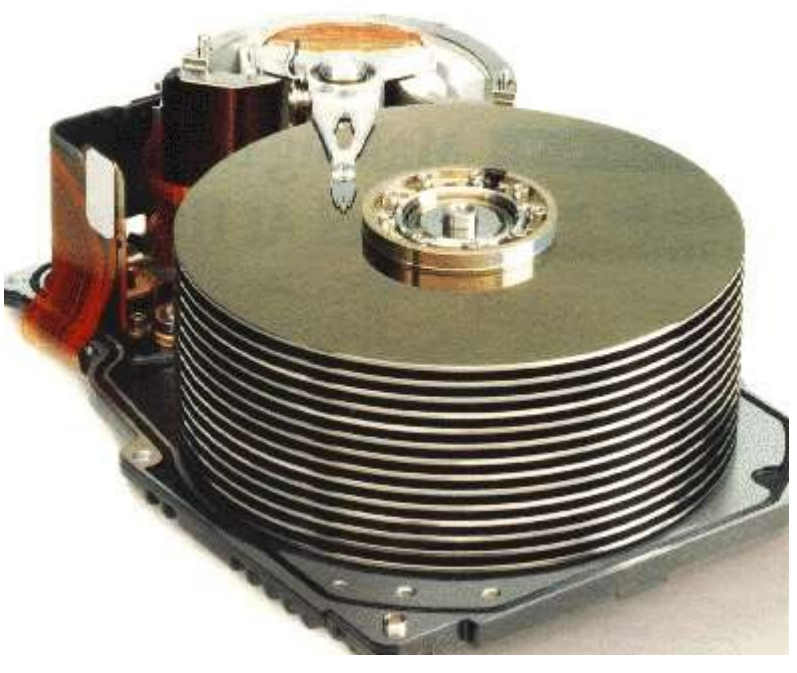
Types of Auxiliary Memory

- **Flash memory**: An electronic non-volatile computer storage device that **can be electrically erased and reprogrammed**, and works without any moving parts. Examples of this are **USB flash drives and solid state drives.**
- **Optical disc:** Its a storage medium from which data is read and to which it is written by lasers. **There are three basic types of optical disks: CD-ROM (read-only), WORM (write-once read many) & EO (erasable optical disks).**
- **Magnetic tapes:** A magnetic tape consists of electric, mechanical and electronic components to provide the parts and control mechanism for a magnetic tape unit. **The tape itself is a strip of plastic coated with a magnetic recording medium.** Bits are recorded as magnetic spots on tape along several tracks called RECORDS. Each record on tape has an identification bit pattern at the beg. and the end. **R/W heads are mounted in each track so that data can be recorded and read as a sequence of characters. Can be stopped , started to move forward, or in reverse, or can be rewound, but cannot be stopped fast enough between individual characters.**

Types of Auxiliary Memory

- **Magnetic Disk:** A magnetic disk is a **circular plate constructed of metal or plastic coated with magnetized material. Both sides of the disk are used** and several disks may be stacked on one spindle with read/write heads available on each surface. **Bits are stored in magnetized surface in spots along concentric circles called tracks. Tracks are commonly divided into sections called sectors. Disk that are permanently attached and cannot removed by occasional user are called hard disks.**





Associative Memory

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.**
- **Write operation:** When a word is written in in an associative memory, no address is given. The memory is capable of finding an unused location to store the word.
- **Read operation:** When a word is to be read from an associative memory, the contents of the word, or a part of the word is specified. The memory locates all the words which match the specified content and marks them for reading.
- An associative memory is **more expensive than a random access memory** because each cell must have storage capability as well as logic circuits for matching its content with an external argument. For this reason, associative memories are **used in applications where the search time is very critical** and must be very short.

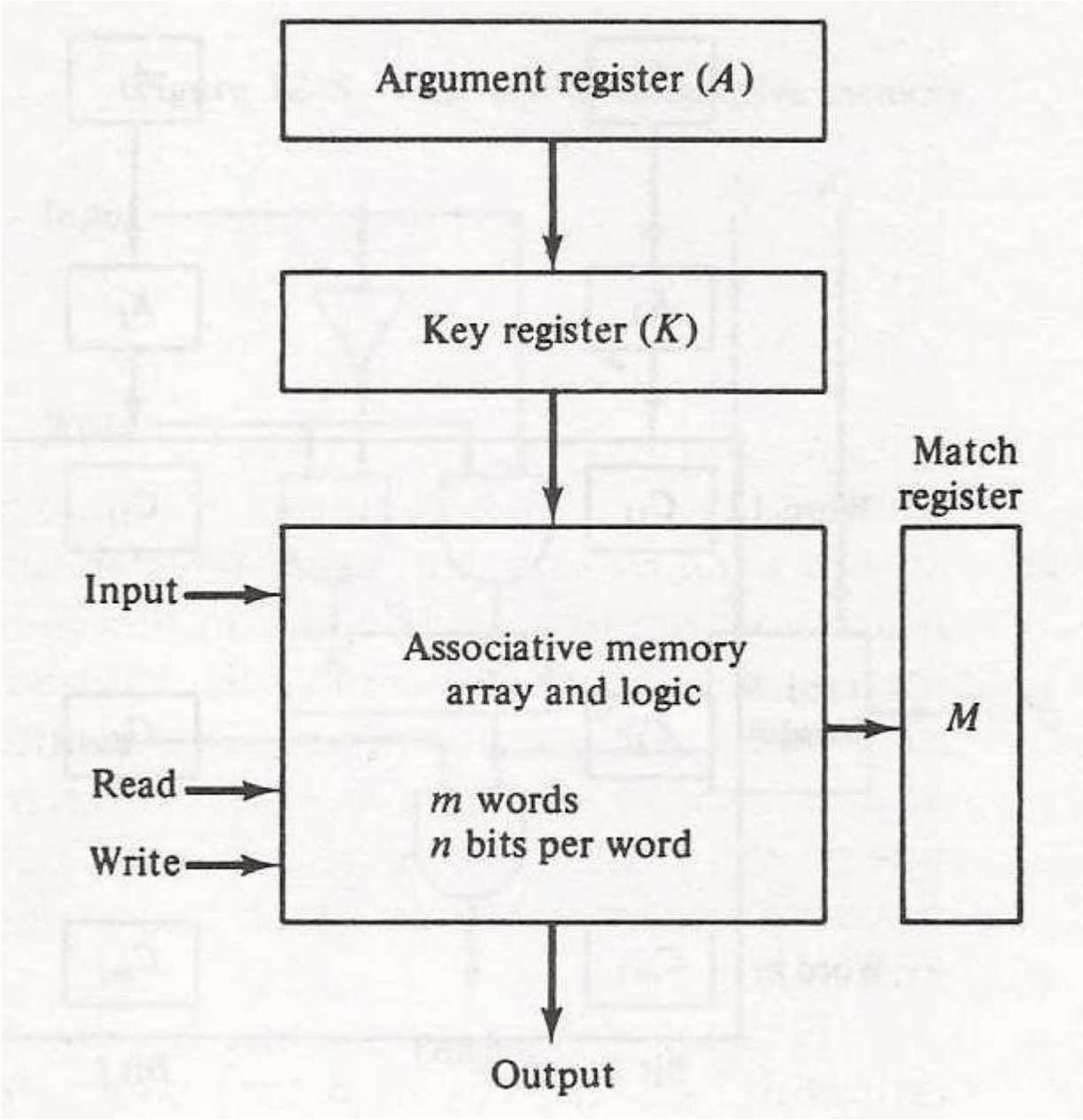
Hardware Organization : Block diagram of an associative memory

Argument register(A): It contains the word to be searched. It has n bits(one for each bit of the word).

Key Register(K): It provides mask for choosing a particular field or key in the argument word. It also has n Bits.

Associative memory array: It contains the words which are to be compared with the argument word.

Match Register(M): It has m bits, one bit corresponding to each word in the memory array . 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.



Matching process in associative memory

The entire argument is compared with each memory word if the key register contains all l' s. Otherwise, only those bits in the argument that have l's 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.

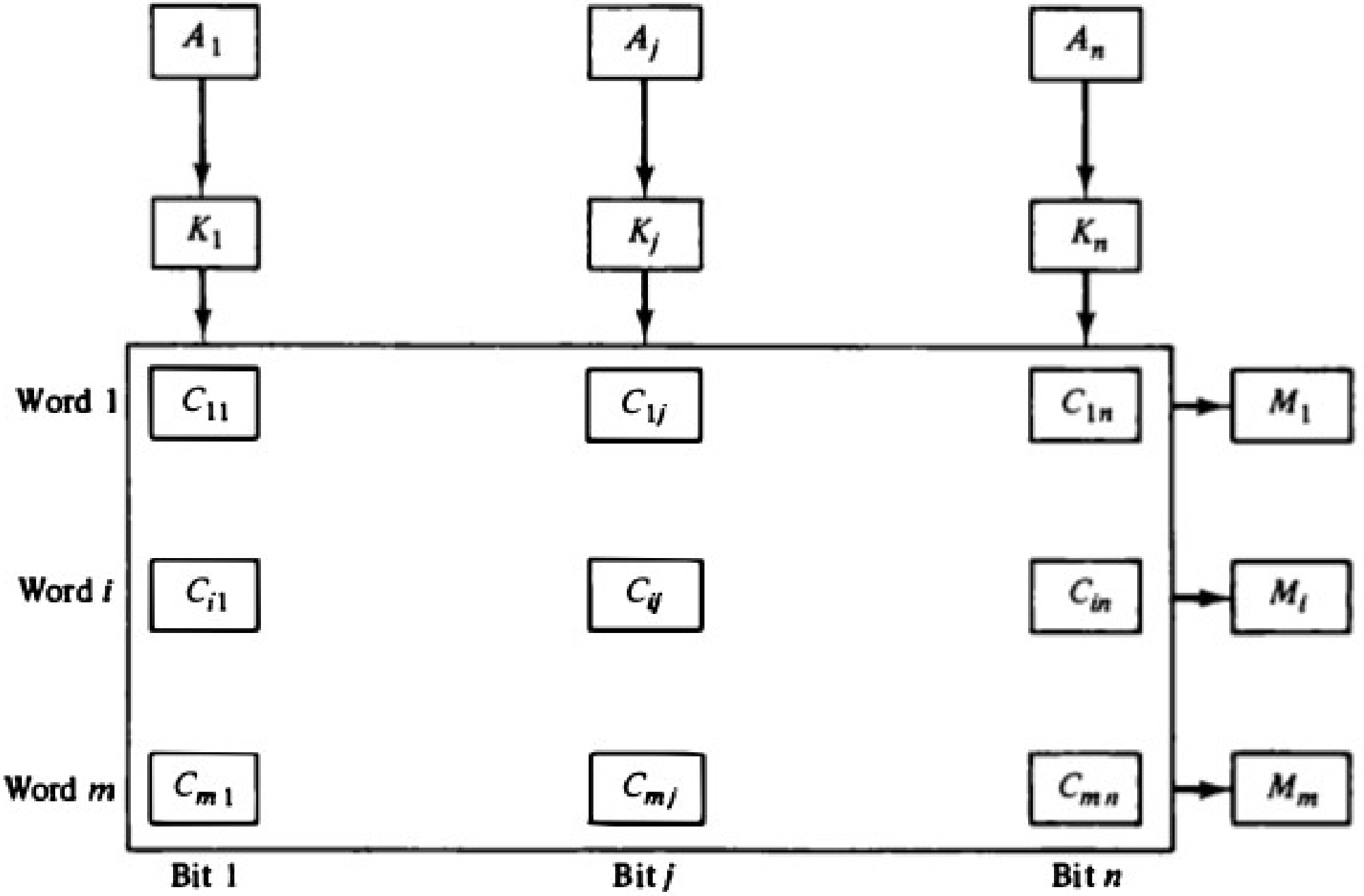
Only the three leftmost bits of A are compared with memory words because K has 1's in these positions

A	101 111100	
K	111 000000	
Word 1	100 111100	no match
Word 2	101 000001	match

The relation between the memory array and external registers in an associative memory is shown in Fig. 12-7.

- The cells in the array are marked by the letter C with two subcripts. The first subscript gives the word number and the second specifies the bit position in the word.
- A bit Ai in the argument register is compared with all the bits in column j of the array provided that Ki = 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 M1 in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, M1 is cleared to 0.

Figure 12-7 Associative memory of m word, n cells per word.



Cache Memory

- If the active portions of the program and data are placed in a fast small memory, the **average memory access time** can be reduced, thus reducing the **total execution time** of the program.
- Such a fast small memory is referred to as a cache memory. **It is placed between the CPU and main memory.**
- **The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components. The most frequently accessed instructions and data are kept in the fast cache memory.**

The basic operation of the cache is as follows:

- When the CPU needs to access memory, the cache is examined.
- If the word is found in the cache, it is read from the fast memory.
- If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word.
- A block of words containing the one just accessed is then transferred from main memory to cache memory.
- If the cache is full, then a block equivalent to the size of the used word is replaced according to the replacement algorithm being used.

- **When the CPU refers to memory and finds the word in cache**, it is said to produce a main memory and **it counts as a hit** .
- **If the word is not found in cache, it is in miss**(it is in the main memory) .
- **The performance of cache memory is frequently measured in terms of a quantity called hit ratio .**

Hit ratio = hit / (hit+miss)

Mapping process

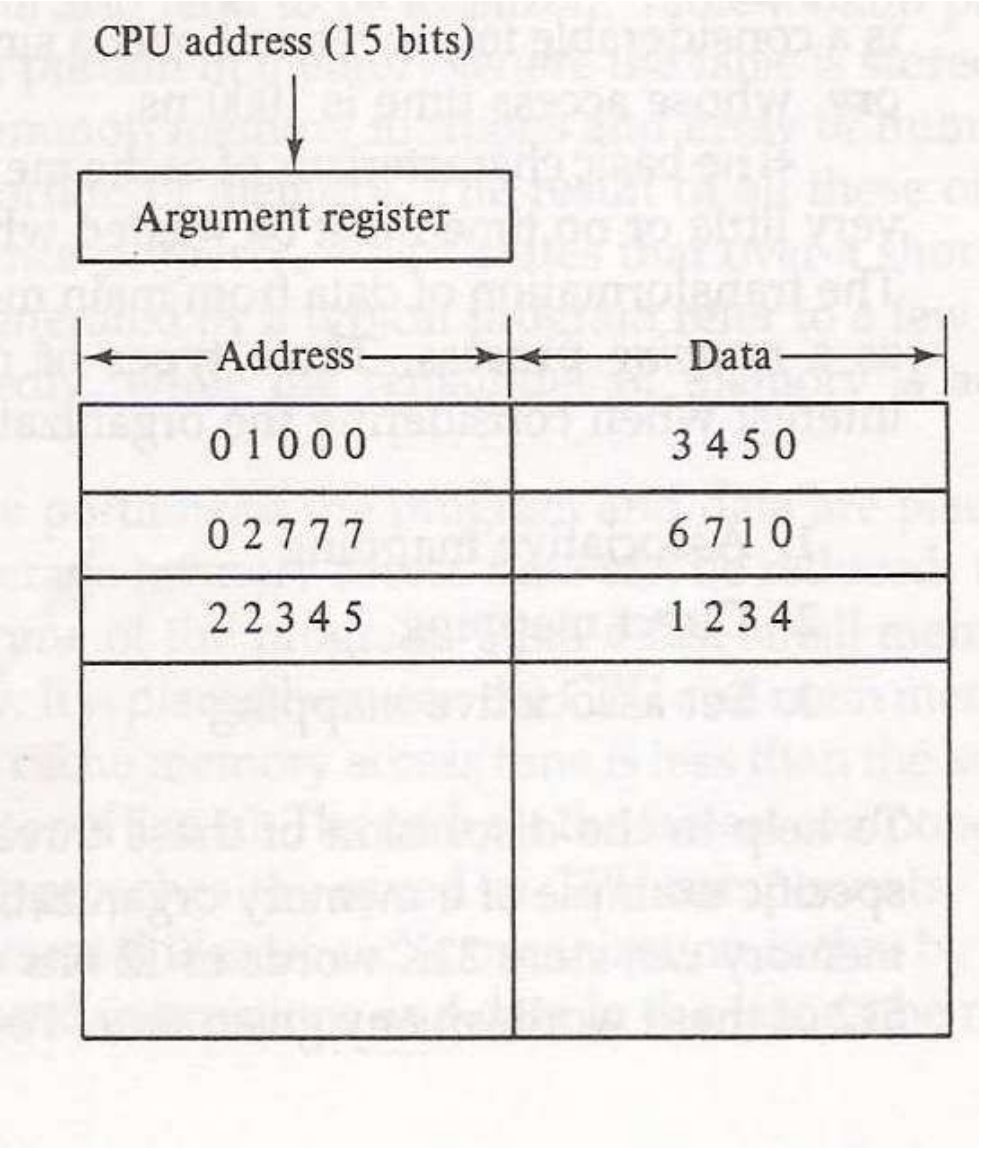
The transformation of data from main memory to cache memory is referred to as a mapping process, there are three types of mapping:

- **Associative mapping**
- **Direct mapping**
- **Set-associative mapping**

Associative Mapping

- **The fastest and most flexible cache organization uses an associative memory.**
- **The associative memory stores both the address and content (data) of the memory word.**

- This permits any location in cache to store any word from main memory.
- 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 the CPU. If no match occurs, the main memory is accessed for the word.**
- **Disadvantage:** Associative memories are expensive compared to random-access memories because of the added logic associated with each cell.

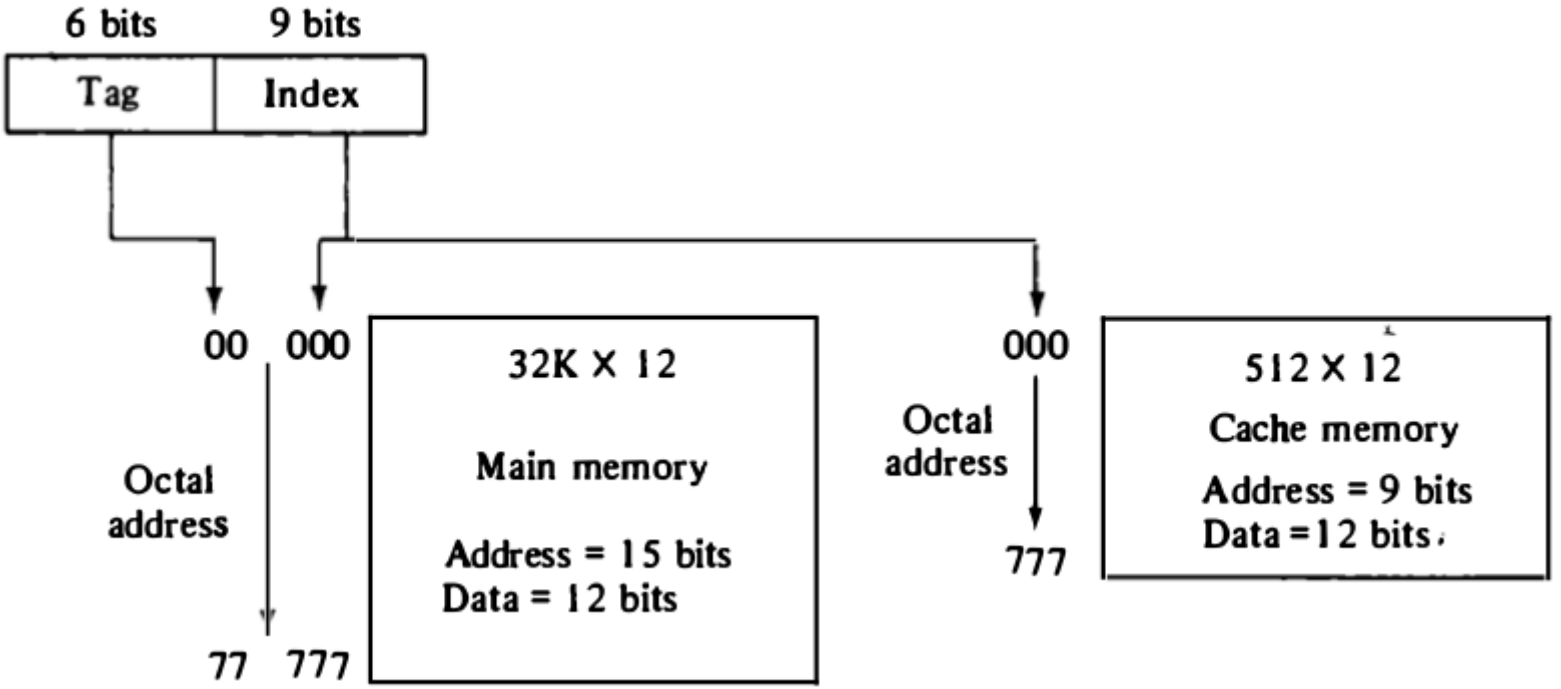


Direct Mapping

The possibility of using a random-access memory for the cache is investigated in Fig. 12-12.

- **The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the index field and the remaining six bits form the tag field.**
- The figure shows that main memory needs an address that includes both the tag and the index bits. **The number of bits in the index field is equal to the number of address bits required to access the cache memory.**

Figure 12-12 Addressing relationships between main and cache memories.



- **In the general case, there are 2^k words in cache memory and 2^n words in main memory (in the figure k=9, n=15).**
- **The n-bit memory address is divided into two fields: k bits for the index field and n - k bits for the tag field.**
- The direct mapping cache organization uses the n-bit address to access the main memory and the k-bit index to access the cache.
- The internal organization of the words in the cache memory is as shown in Fig. 12-13(b).
- **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 disadvantage of direct mapping** is that the hit ratio can drop considerably if two or more words whose addresses have the same index but different tags are accessed repeatedly. However, this possibility is minimized by the fact that such words are relatively far apart in the address range.
- To see **how the direct-mapping organization operates**, **consider the numerical example shown in Fig. 12-13. 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.**

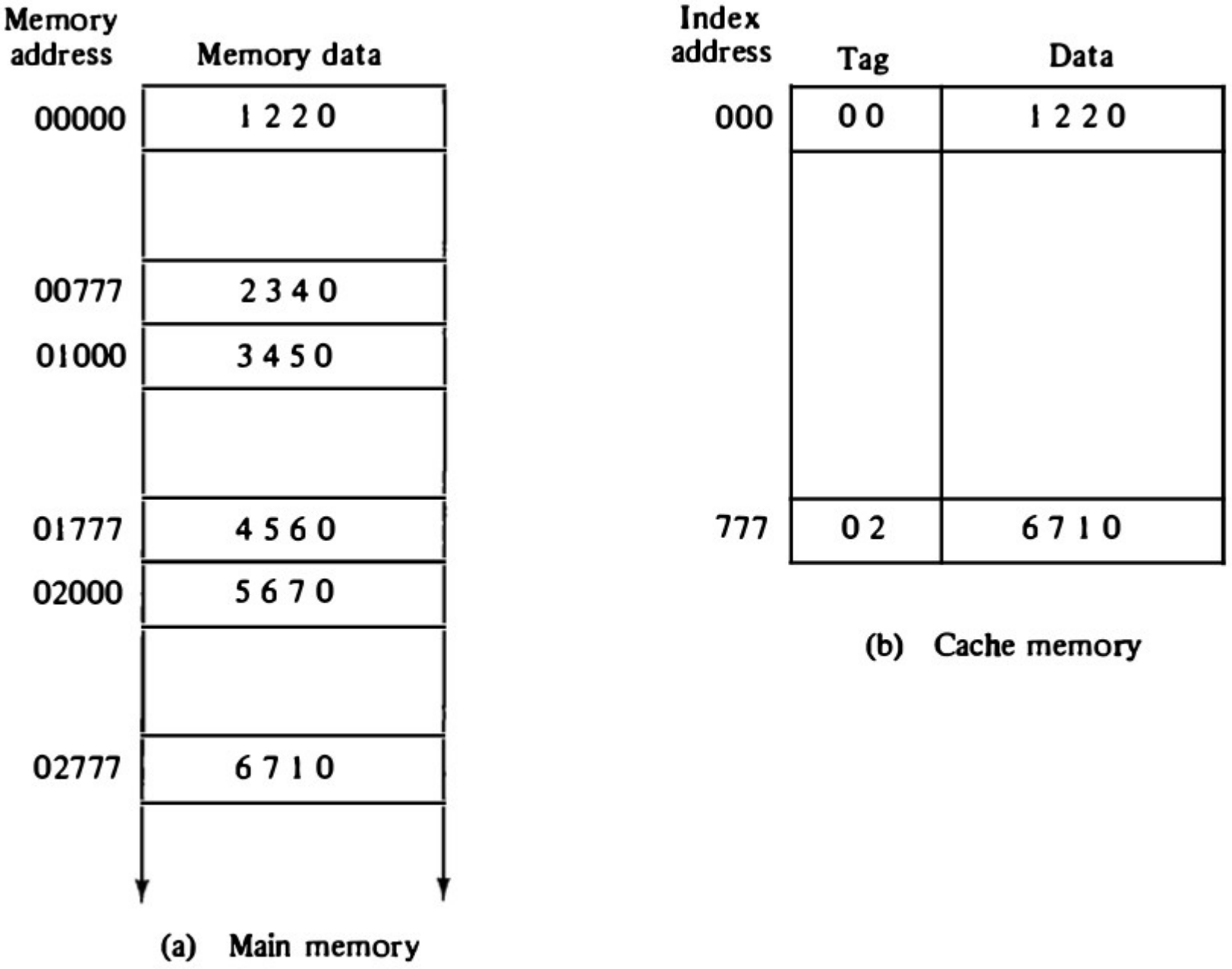


Figure 12-13 Direct mapping cache organization.

Set-Associative Mapping

- **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.**
- **set-associative mapping, is an improvement over the direct mapping organization in that each word of cache can store two or more words of memory under the same index address.**
- Each data word is stored together with its tag and the number of tag-data items in one word of cache is said to form a set.
- An example of a **set-associative cache organization for a set size of two is shown in Fig. 12-15**. **Each index address refers to two data words and their associated tags. Each tag requires six bits and each data word has 12 bits, so the word length is 2(6 + 12) = 36 bits.**

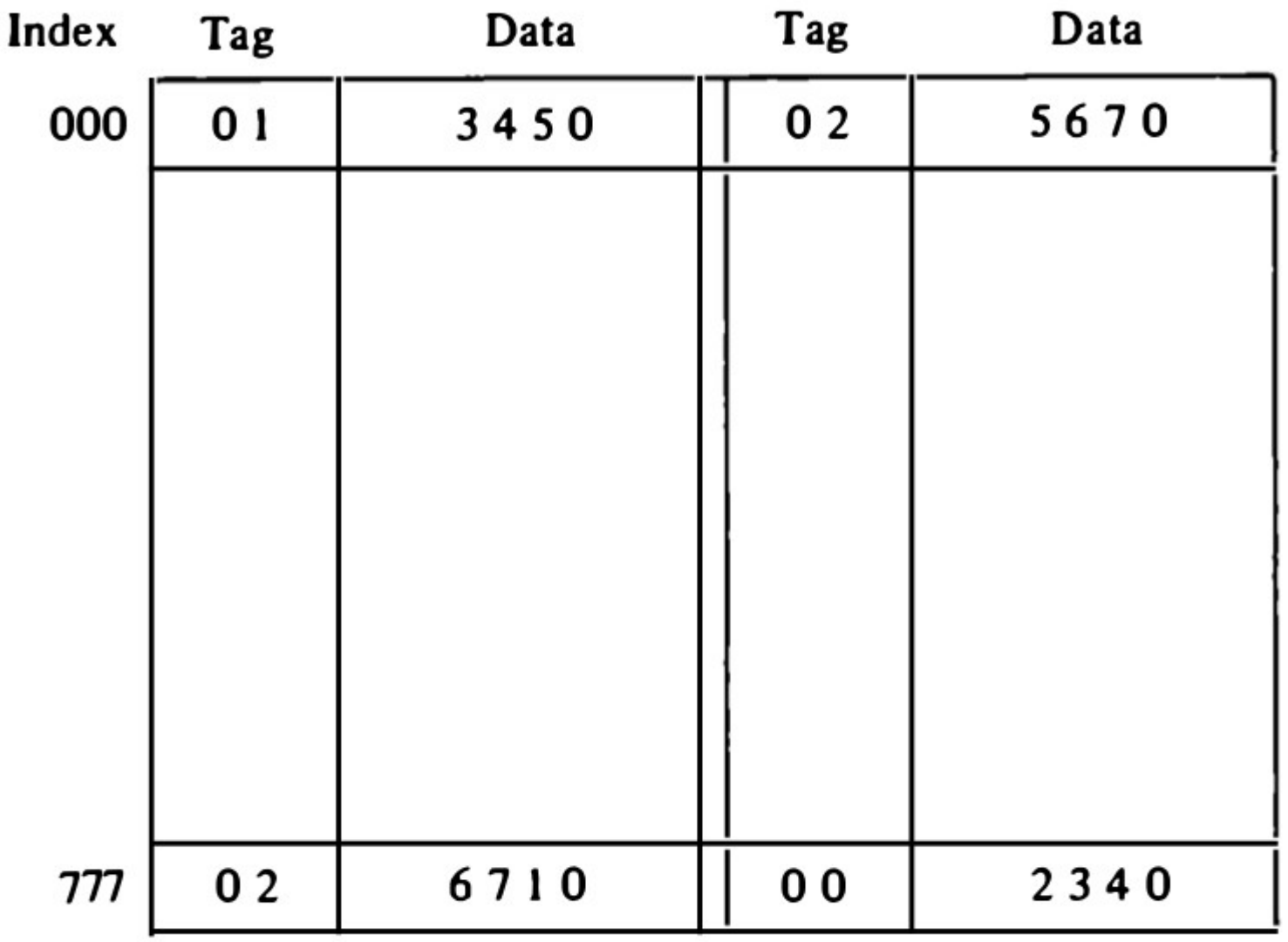


Figure 12-15 Two-way set-associative mapping cache.

Virtual memory

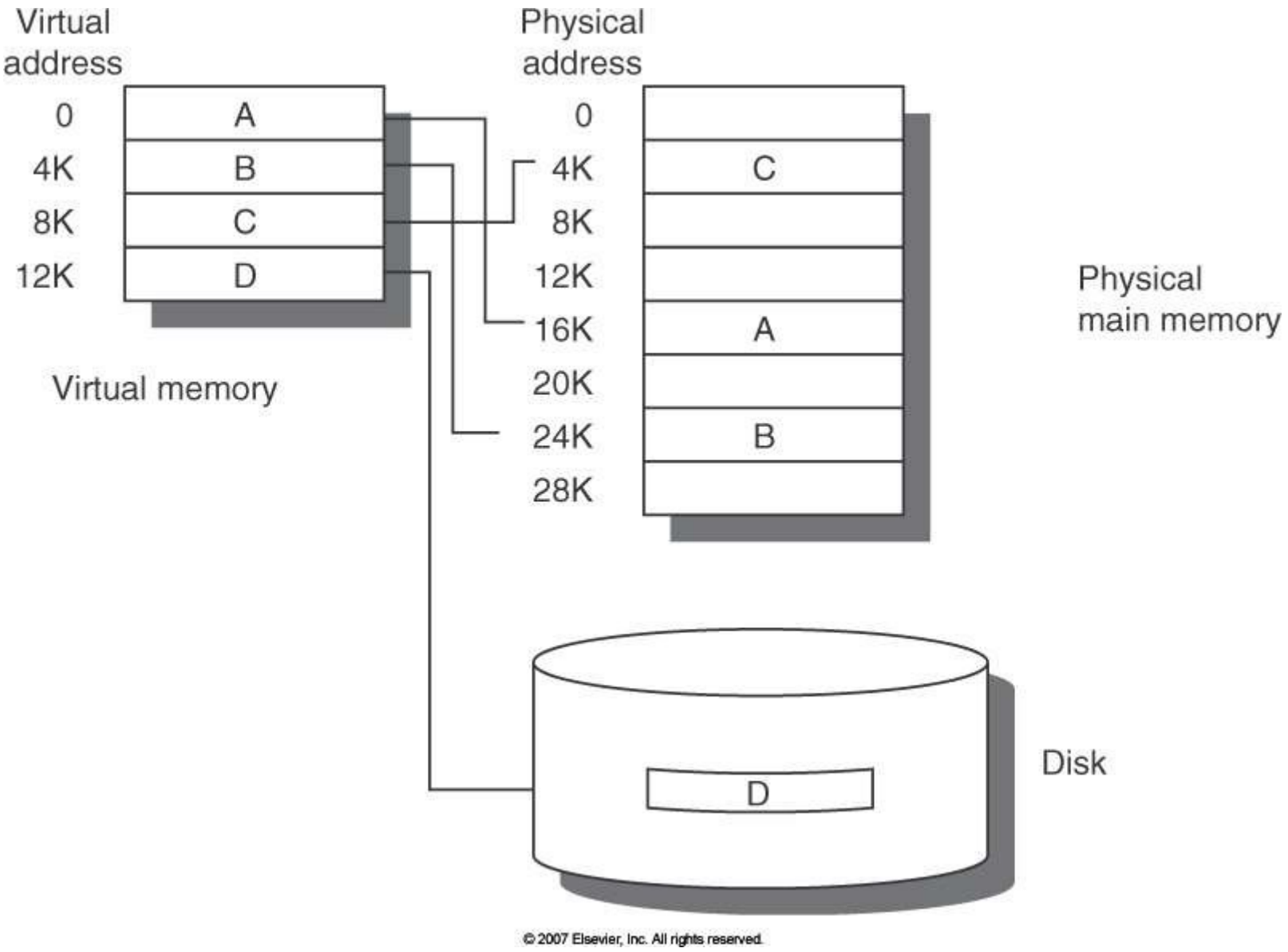
- Virtual memory is a common part of operating system on desktop computers.
- **The term Virtual Memory refers to something which appears to be present but actually is not.**
- **This technique allows users to use more memory for a program than the real memory of a computer.**
- Virtual Memory is a imaginary memory which we assume or use, when we have a material that exceeds our memory at that time.

Advantages:

- Allows Processes whose aggregate memory requirement is greater than the amount of physical memory, as infrequently used pages can reside on the disk.
- Virtual memory allows speed gain when only a particular segment of the program is required for the execution of the program.
- This concept is very helpful in implementing multiprogramming environment.

Disadvantages: Applications run rather slower when they are using virtual memory. It takes more time to switch between applications. Reduces system stability.

Virtual memory



Practice Questions

1) Define Virtual Memory and its need.

2) What is the need of cache memory?

3) Explain memory hierarchy in a computer system?

4) Illustrate the memory hierarchy in order of their features with their comparative analysis.

5) A digital computer has a memory unit of 64K x 16 and a cache memory of 1K words. The cache uses direct mapping. How many bits are there in the tag, index fields of the address format?

6) Explain about the Cache miss.

7) Explain different types of cache mapping.

Solution

Q5)

Main memory has 64K = 64 x 1024 = 2^6 x 2^10 = 2^16 words

Cache memory has 1K = 1024 = 2^10 words

Index and Tag together make main memory address. Index part addresses cache memory.

In this case to address main memory we need 16 bits (2^16) and to address cache memory we need 10 bits (2^10).

So **Index is 10 bits wide and Tag is 6 bits wide** (16 - 10 = 6).