UNIT III

Memory Organization

**MEMORY HIERARCHY**

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 accessible to the high speed processing logic.

* + - **Main Memory**: memory unit that communicates directly with the CPU (RAM)
    - **Auxiliary Memory**: device that provide backup storage (Disk Drives)
    - **Cache Memory**: special very-high-speed memory to increase the processing speed (Cache RAM)

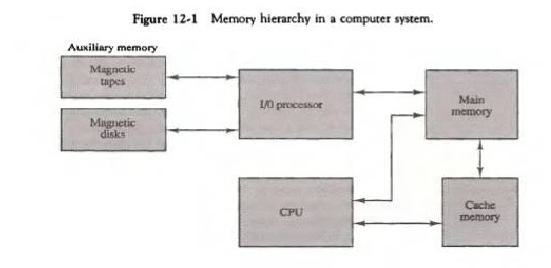


Figure12- 1 illustrates the components in a typical memory hierarchy. At the bottom of the hierarchy are the relatively slow magnetic tapes used to store removable files. Next are the Magnetic disks used as backup storage. The main memory occupies a central position by being able to communicate directly with CPU and with auxiliary memory devices through an I/O process. Program not currently needed in main memory are transferred into auxiliary memory to provide space for currently used programs and data.

The cache memory is used for storing segments of programs currently being executed in the CPU. The I/O processor manages data transfer between auxiliary memory and main memory. 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. The CPU has direct access to both cache and main memory but not to auxiliary memory.

In the Computer System Design, Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time. The Memory Hierarchy was developed based on a program behavior known as locality of references.The figure below clearly demonstrates the different levels of memory hierarchy :



This Memory Hierarchy Design is divided into 2 main types:

1. **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.

2. **Internal Memory or Primary Memory –**Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

We can infer the following characteristics of Memory Hierarchy Design from above figure:

1. **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.

2. **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.

3. **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.

4. **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.

**RAM and ROM architecture.**

**1) Read-only memory, or ROM,** is a form of data storage in computers and other electronic devices that cannot be easily altered or reprogrammed. RAM is referred to as volatile memory and is lost when the power is turned off whereas ROM in non-volatile and the contents are retained even after the power is switched off.

**Types of ROM: Semiconductor-Based**

Classic mask-programmed ROM chips are integrated circuits that physically encode the data to be stored, and thus it is impossible to change their contents after fabrication. Other types of non-volatile solid-state memory permit some degree of modification:

• **Programmable read-only memory (PROM),** or one-time programmable ROM (OTP), can be written to or programmed via a special device called a PROM programmer. Typically, this device uses high voltages to permanently destroy or create internal links (fuses or antifuses) within the chip. Consequently, a PROM can only be programmed once.

• **Erasable programmable read-only memory (EPROM)** can be erased by exposure to strong ultraviolet light (typically for 10 minutes or longer), then rewritten with a process that again needs higher than usual voltage applied. Repeated exposure to UV light will eventually wear out an EPROM, but the endurance of most EPROM chips exceeds 1000 cycles of erasing and reprogramming. EPROM chip packages can often be identified by the prominent quartz "window" which allows UV light to enter. After programming, the window is typically covered with a label to prevent accidental erasure. Some EPROM chips are factory-erased before they are packaged, and include no window; these are effectively PROM.

• **Electrically erasable programmable read-only memory (EEPROM)** is based on a similar semiconductor structure to EPROM, but allows its entire contents (or selected banks) to be electrically erased, then rewritten electrically, so that they need not be removed from the computer (whether general-purpose or an embedded computer in a camera, MP3 player, etc.). Writing or flashing an EEPROM is much slower (milliseconds per bit) than reading from a ROM or writing to a RAM (nanoseconds in both cases).

• **Electrically alterable read-only memory (EAROM)** is a type of EEPROM that can be modified one bit at a time. Writing is a very slow process and again needs higher voltage (usually around 12 V) than is used for read access. EAROMs are intended for applications that require infrequent and only partial rewriting. EAROM may be used as non-volatile storage for critical system setup information; in many applications, EAROM has been supplanted by CMOS RAM supplied by mains power and backed-up with a lithium battery.

• **Flash memory (or simply flash)** is a modern type of EEPROM invented in 1984. Flash memory can be erased and rewritten faster than ordinary EEPROM, and newer designs feature very high endurance (exceeding 1,000,000 cycles). Modern NAND flash makes efficient use of silicon chip area, resulting in individual ICs with a capacity as high as 32 GB as of 2007; this feature, along with its endurance and physical durability, has allowed NAND flash to replace magnetic in some applications (such as USB flash drives). Flash memory is sometimes called flash ROM or flash EEPROM when used as a replacement for older ROM types, but not in applications that take advantage of its ability to be modified quickly and frequently.

**2) Random-access memory, or RAM,** is a form of data storage that can be accessed randomly at any time, in any order and from any physical location in contrast to other storage devices, such as hard drives, where the physical location

of the data determines the time taken to retrieve it. RAM is measured in megabytes and the speed is measured in nanoseconds and RAM chips can read data faster than ROM.

**Types of RAM:** The two widely used forms of modern RAM are **static RAM** (SRAM) and **dynamic RAM** (DRAM). In SRAM, a bit of data is stored using the state of a six transistor memory cell. This form of RAM is more expensive to produce, but is generally faster and requires less dynamic power than DRAM. In modern computers, SRAM is often used as cache memory for the CPU. DRAM stores a bit of data using a transistor and capacitor pair, which together comprise a DRAM cell. The capacitor holds a high or low charge (1 or 0, respectively), and the transistor acts as a switch that lets the control circuitry on the chip read the capacitor's state of charge or change it. As this form of memory is less expensive to produce than static RAM, it is the predominant form of computer memory used in modern computers. The figure below shows DRAM & SRAM resp.

Both static and dynamic RAM are considered volatile, as their state is lost or reset when power is removed from the system. By contrast, read-only memory (ROM) stores data by permanently enabling or disabling selected transistors, such that the memory cannot be altered. Writeable variants of ROM (such as EEPROM and flash memory) share properties of both ROM and RAM, enabling data to persist without power and to be updated without requiring special equipment. These persistent forms of semiconductor ROM include USB flash drives, memory cards for cameras and portable devices, and solid-state drives. ECC memory (which can be either SRAM or DRAM) includes special circuitry to detect and/or correct random faults (memory errors) in the stored data, using parity bits or error correction codes.

**Difference between Static Ram And Dynamic Ram**



**MAIN MEMORY**

Main memory is the central storage unit in a computer system. It is a relatively large and fast memory used to store programs and data during the computer operation. The principal technology used for the main memory is based on semi conductor integrated circuits. Integrated circuits RAM chips are available in two possible operating modes, static and dynamic.

* Static RAM – Consists of internal flip flops that store the binary information.
* Dynamic RAM – Stores the binary information in the form of electric charges that are applied to capacitors.

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

* Read Only Memory –Store 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.

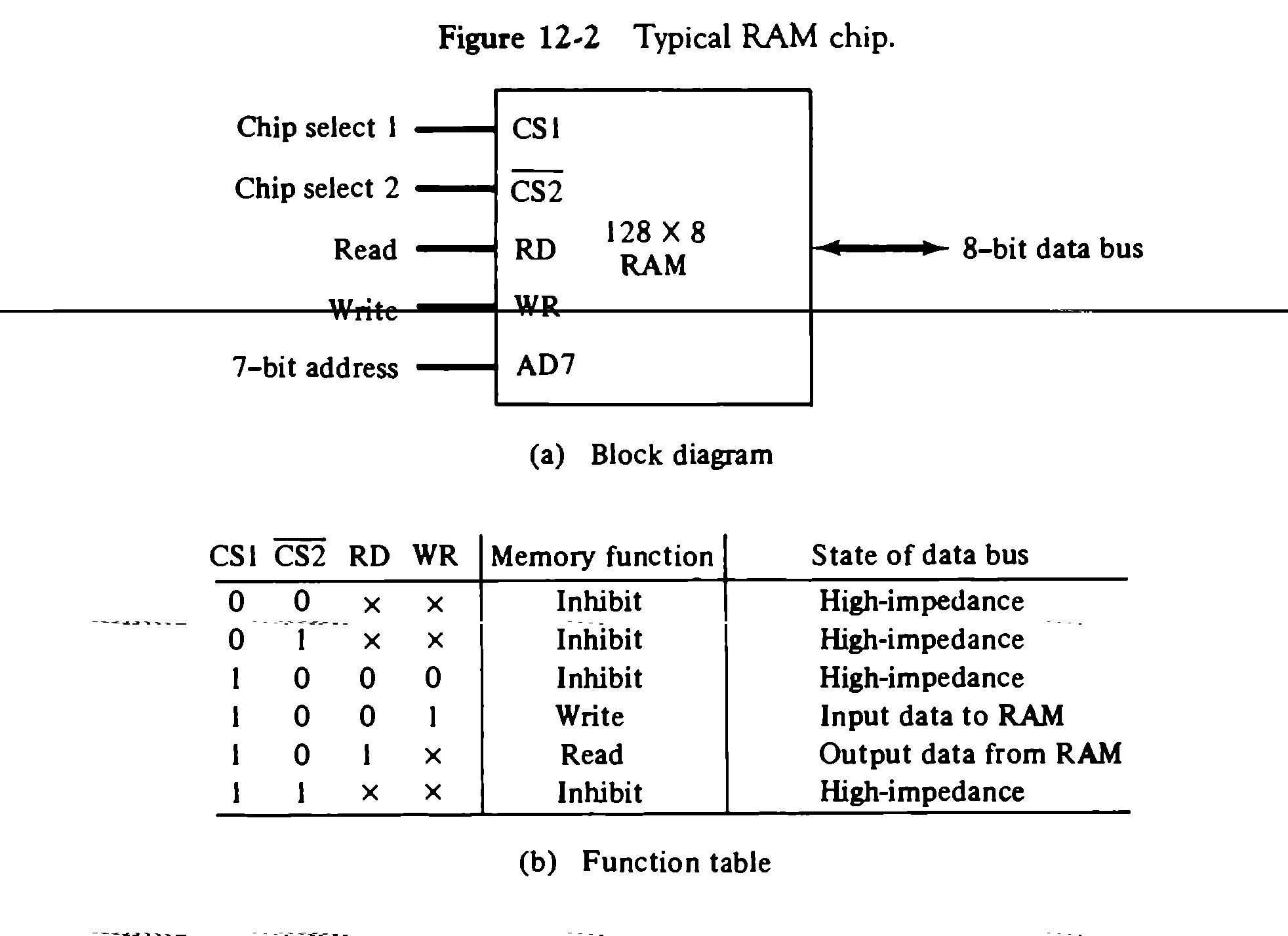
loader.

The ROM portion of main memory is needed for storing an initial program called a Bootstrap

* Boot strap loader –function is start the computer software operating when power is turned on.

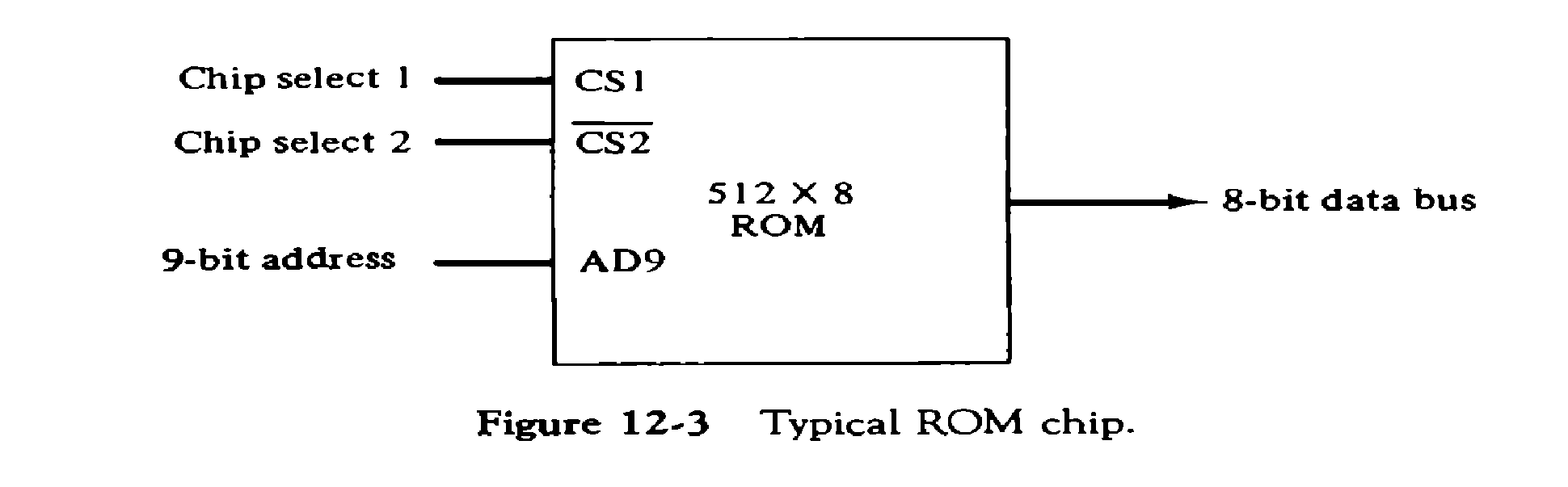
# RAM and ROM CHIP

* + RAM chip –utilizes bidirectional data bus with three state buffers to perform communication with CPU



The block diagram of a RAM Chip is shown in Fig.12-2. The capacity of 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 enabling the chip only when it is selected by the microprocessor. The read and write inputs are sometimes combined into one line labelled R/W.

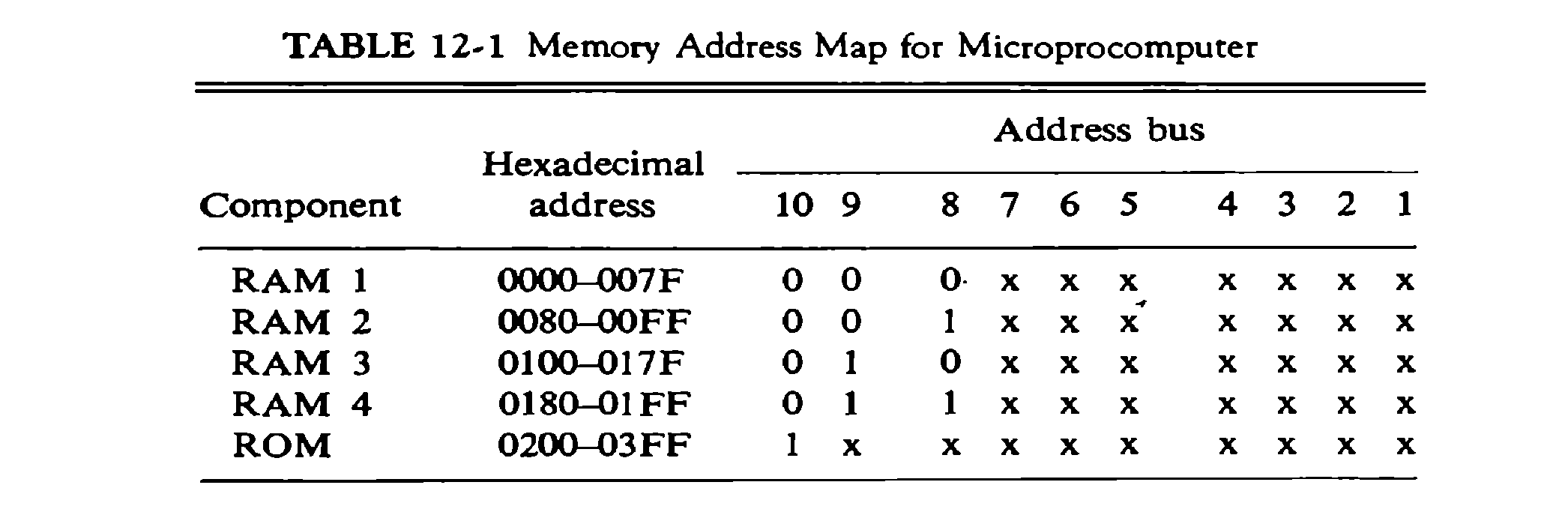
The function table listed in Fig.12-2(b) specifies the operation of the RAM chip. The unit is in operation only when CS1=1 and CS2=0.The bar on top of the second select variable indicates that this input is enabled when it is equal to 0. If the chip select inputs are not enabled, or if they are enabled 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. The RD and WR signals control the memory operation as well as the bus buffers associated with the bidirectional data bus.



A ROM chip is organized externally in a similar manner. However, since a ROM can only read, the data bus can only be in an output mode. The block diagram of a ROM chip is shown in fig.12-3. The nine address lines in the ROM chip specify any one of the512 bytes stored in it. The two chip select inputs must be CS1=1 and CS2=0 for the unit to operate. Otherwise, the da~~ta bu~~s is in a high-impedance state.

# Memory Address Map

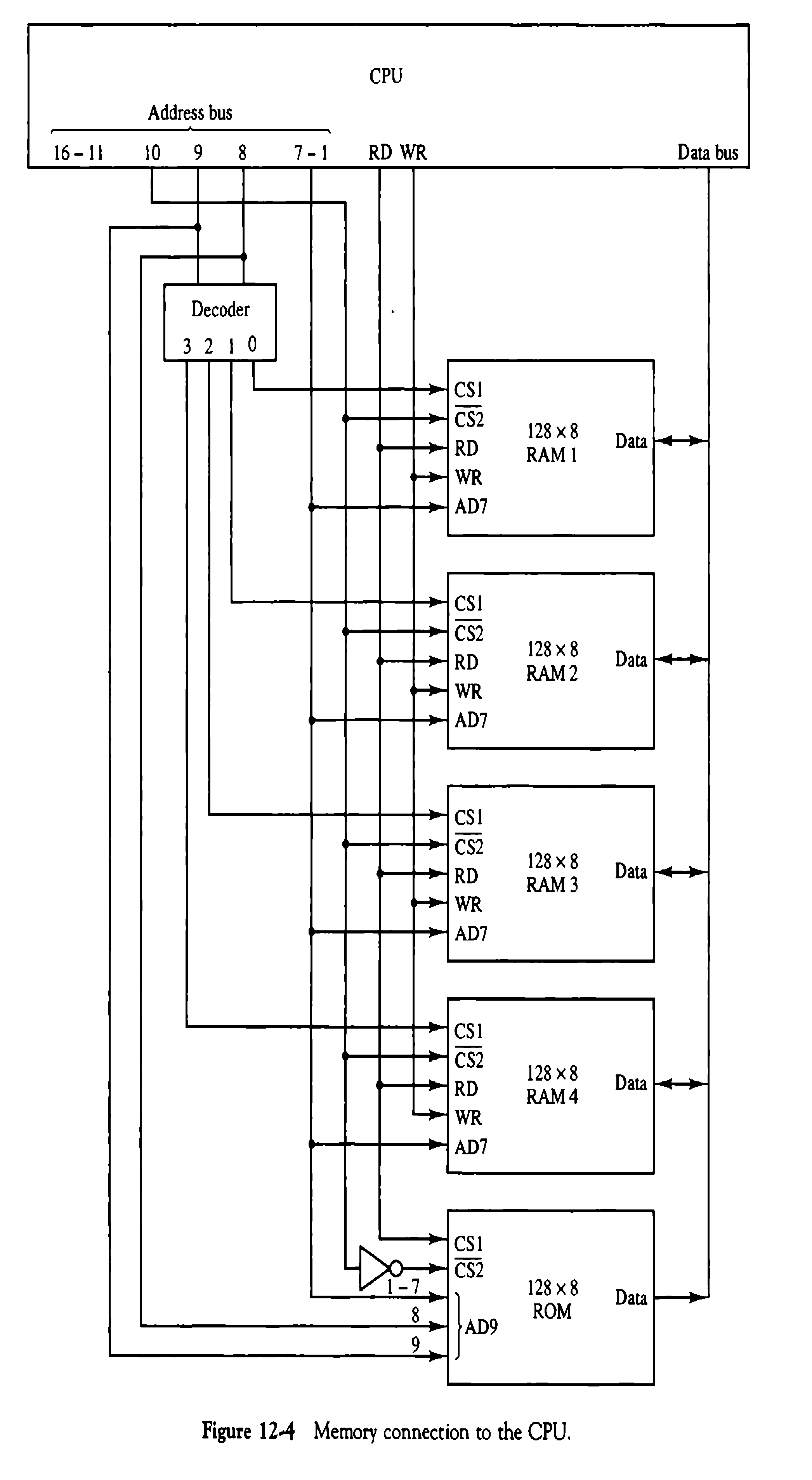
The interconnection between memory and processor is then established from knowledge of the size of memory needed and the type of RAM and ROM chips available. The addressing of memory can be established by means of a table that specify the memory address assigned to each chip. The table called Memory address map, is a pictorial representation of assigned address space for each chip in the system.



The memory address map for this configuration is shown in table 12-1. 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. The RAM chips have 128 bytes and need seven address lines. The ROM chip has 512 bytes and needs 9 address lines.

# Memory Connection 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.



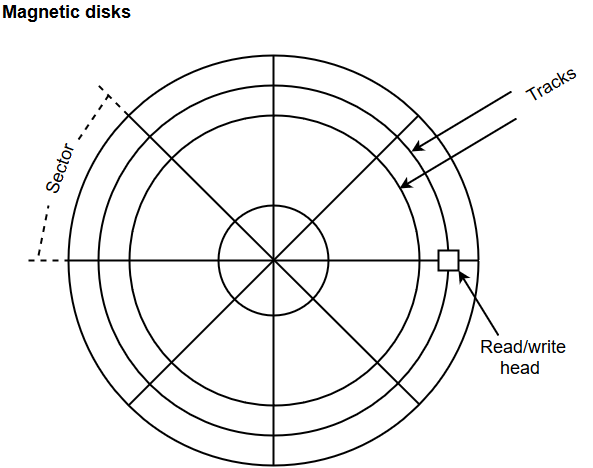
The connection of memory chips to the CPU is shown in Fig.12-4. This configuration gives a memory capacity of 512 bytes of RAM and 512 bytes of ROM. Each RAM receives the seven low-order bits of the address bus to select one of 128 possible bytes. The particular RAM chip selected is determined from lines 8 and 9 in the address bus. This is done through a 2 X 4 decoder whose outputs go to the CS1 inputs in each RAM chip. Thus, when address lines 8 and 9 are equal to 00, the first RAM chip is selected. When 01, the second RAM chip is select, and so on. The RD and WR outputs from the microprocessor are applied to the inputs of each RAM chip. The selection between RAM and ROM is achieved through bus line 10. The RAMs are selected when the bit in this line is 0, and the ROM when the bit is 1. Address bus lines 1 to 9 are applied to the input address of ROM without going through the decoder. The data bus of the ROM has only an output capability, whereas the data bus connected to the RAMs can transfer information in both directions.

An Auxiliary memory is known as the lowest-cost, highest-capacity and slowest-access storage in a computer system. It is where programs and data are kept for long-term storage or when not in immediate use. The most common examples of auxiliary memories are magnetic tapes and magnetic disks.

## Magnetic Disks

A magnetic disk is a type of memory constructed using a circular plate of metal or plastic coated with magnetized materials. Usually, both sides of the disks are used to carry out read/write operations. However, several disks may be stacked on one spindle with read/write head available on each surface.

The following image shows the structural representation for a magnetic disk.



* The memory bits are stored in the magnetized surface in spots along the concentric circles called tracks.
* The concentric circles (tracks) are commonly divided into sections called sectors.

## Magnetic Tape

Magnetic tape is a storage medium that allows data archiving, collection, and backup for different kinds of data. The magnetic tape is constructed using a plastic strip coated with a magnetic recording medium.

The bits are recorded as magnetic spots on the tape along several tracks. Usually, seven or nine bits are recorded simultaneously to form a character together with a parity bit.

Magnetic tape units can be halted, started to move forward or in reverse, or can be rewound. However, they cannot be started or stopped fast enough between individual characters. For this reason, information is recorded in blocks referred to as records.

## [Optical discs](https://www.britannica.com/technology/optical-disc)

Another form of largely read-only memory is the optical [compact disc](https://www.britannica.com/technology/compact-disc), developed from [videodisc](https://www.britannica.com/technology/videodisc) [technology](https://www.britannica.com/technology/technology) during the early 1980s. Data are recorded as tiny pits in a single spiral track on plastic discs that range from 3 to 12 inches (7.6 to 30 cm) in diameter, though a diameter of 4.8 inches (12 cm) is most common. The pits are produced by a laser or by a stamping [machine](https://www.britannica.com/technology/machine) and are read by a low-power laser and a photocell that generates an electrical signal from the varying light reflected from the pattern of pits. Optical discs are removable and have a far greater memory capacity than diskettes; the largest ones can store many gigabytes of information.

A common optical disc is the [CD-ROM](https://www.britannica.com/technology/CD-ROM) (compact disc read-only memory). It holds about 700 megabytes of data, recorded with an error-correcting code that can correct bursts of errors caused by dust or imperfections. CD-ROMs are used to distribute [software](https://www.britannica.com/technology/software), encyclopaedias, and multimedia text with audio and images. CD-R (CD-recordable), or [WORM](https://www.britannica.com/technology/WORM-computer-science) (write-once read-many), is a variation of CD-ROM on which a user may record information but not subsequently change it. CD-RW (CD-rewritable) disks can be re-recorded. [DVDs](https://www.britannica.com/technology/compact-disc/Analog-versus-digital-sound#ref92853) (digital video, or [versatile](https://www.britannica.com/dictionary/versatile), discs), developed for recording movies, store data more densely than does CD-ROM, with more powerful error correction. Though the same size as CDs, DVDs typically hold 5 to 17 gigabytes—several hours of video or several million text pages.

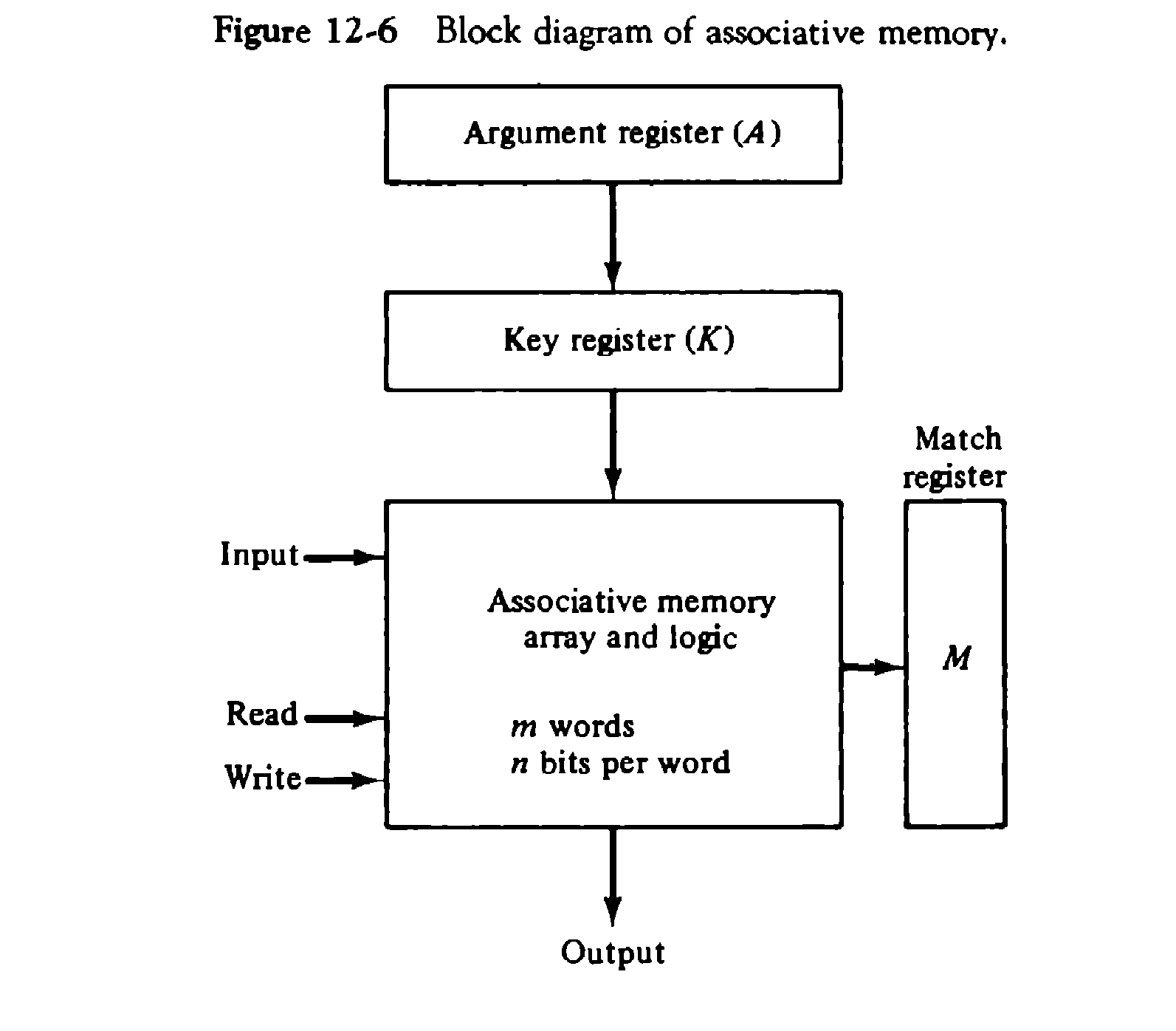
**ASSOCIATIVE MEMORY**

An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location.Associative memory is often referred to as **Content Addressable Memory (CAM)**.

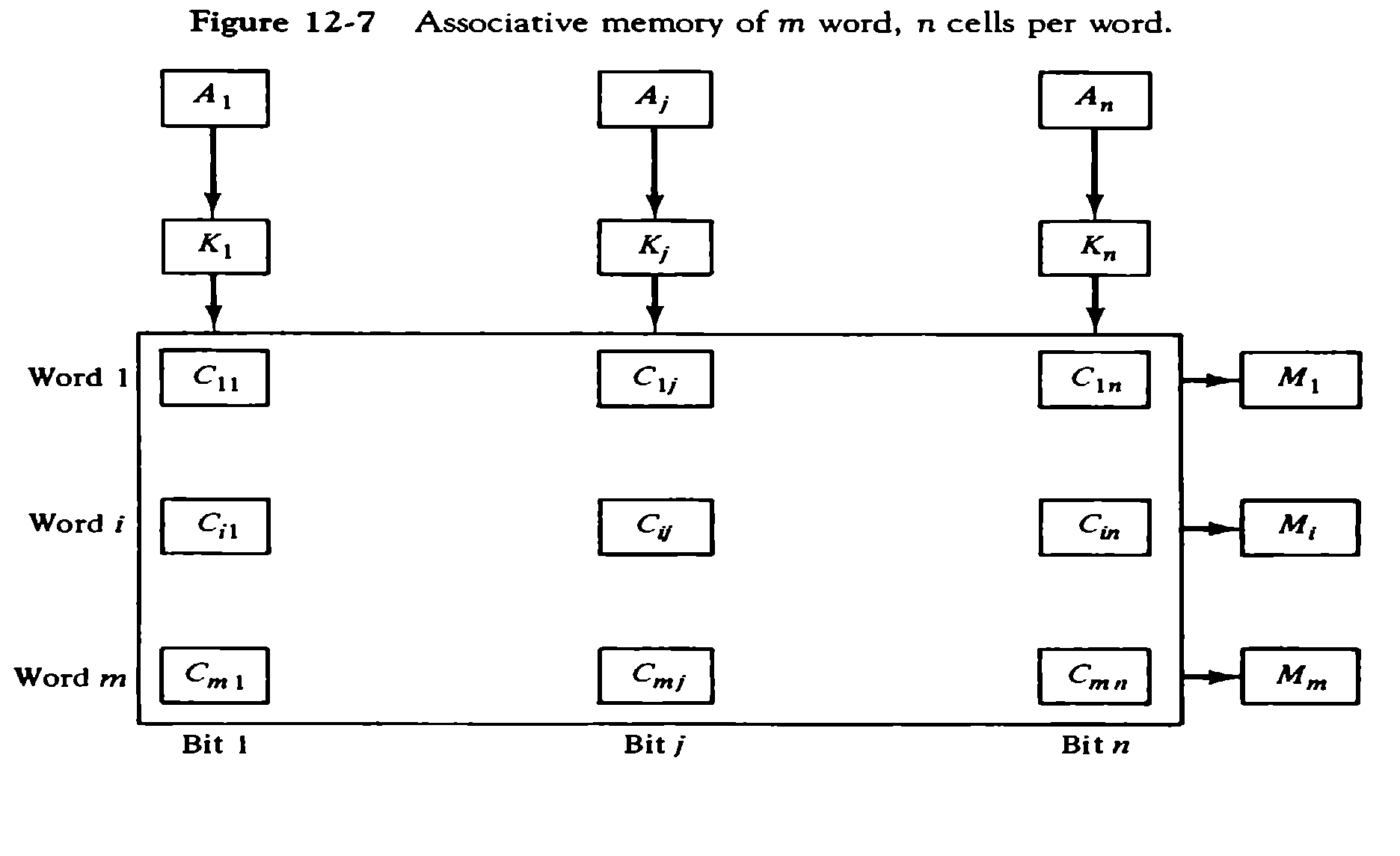
The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit accessed by content is called an associative memory or content addressable memory (CAM).

* + - CAM is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location
    - Associative memory is more expensive than a RAM because each cell must have storage capability as well as logic circuits
    - Argument register –holds an external argument for content matching
    - Key register –mask for choosing a particular field or key in the argument word

# Hardware Organization



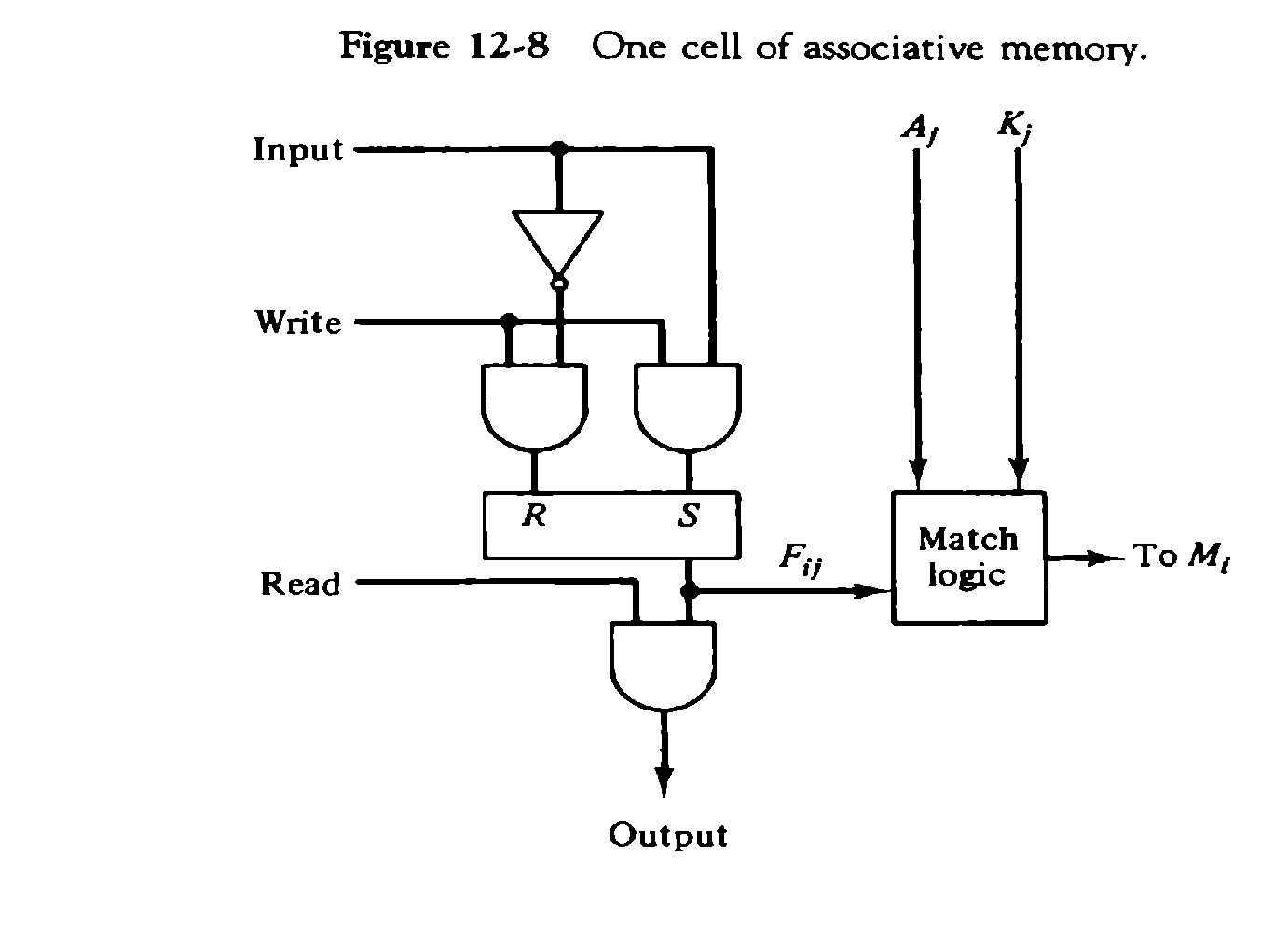
It consists of a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word. The match register M has m bits, one for each memory word. Each word in memory is compared in parallel with the content of the argument register. 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 bindicate 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.



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 subscripts. The first subscript gives the word number and second specifies the bit position in the word. Thus cell Cij is the cell for bit j in word

1. 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.



It consists of flip-flop storage element Fij and the circuits for reading, writing, and matching the cell. The input bit is transferred into the storage cell during a write operation. The bit stored is read out during a read operation. The match logic compares the content of the storage cell with corresponding unmasked bit of the argument and provides an output for the decision logic that sets the bit in Mi.

# Match Logic

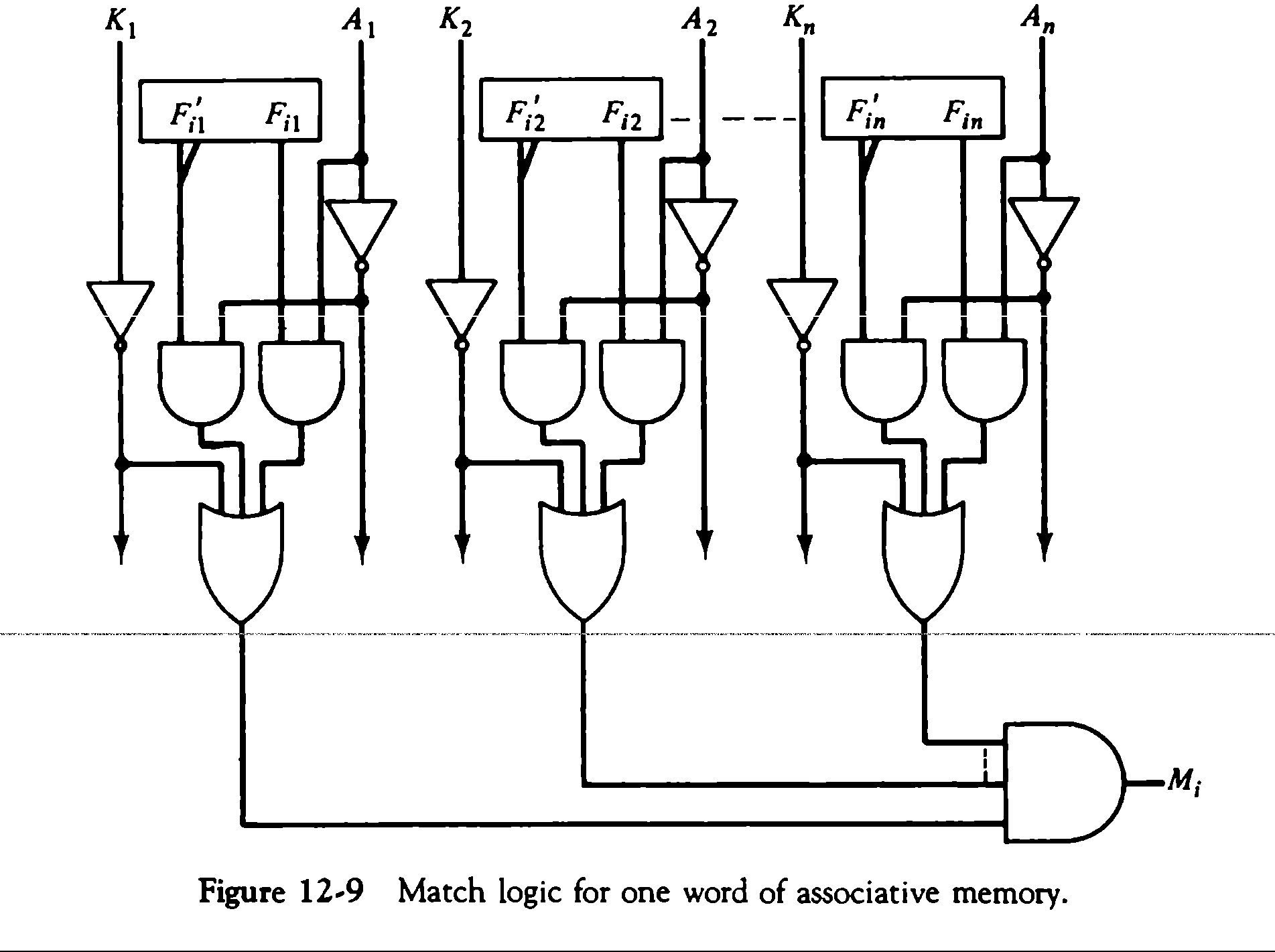
The match logic for each word can be derived from the comparison algorithm for two binary numbers. First, neglect the key bits and compare the argument in A with the bits stored in the cells of the words.

Word i is equal to the argument in A if Aj=F ijfor j=1,2,…..,n. Two bits are equal if they are both 1 or both 0. The equality of two bits can be expressed logically by the Boolean function

xj=Aj Fij + Aj ‘Fij ‘

where xj = 1 if the pair of bits in position j are equal;otherwise , xj =0. For a word i is equal to the argument in A we must have all xj variables equal to 1. This is the condition for setting the corresponding match bit Mi to 1. The Boolean function for this condition is

Mi = x1 x2 x3…… xn



Each cell requires two AND gate and one OR gate. The inverters for A and K are needed once for each column and are used for all bits in the column. The output of all OR gates in the cells of the same word go to the input of a common AND gate to generate the match signal for Mi. Mi will be logic 1 if a match occurs and 0 if no match occurs.

# Read and Write operation Read Operation

* If more than one word in memory matches the unmasked argument field , all the matched words will have 1’s in the corresponding bit position of the match register
* In read operation all matched words are read in sequence by applying a read signal to each word line whose corresponding Mi bit is a logic 1
* In applications where no two identical items are stored in the memory , only one word may match

, in which case we can use Mi output directly as a read signal for the corresponding word

# Write Operation

Can take two different forms

* 1. Entire memory may be loaded with new information
  2. Unwanted words to be deleted and new words to be inserted

1. Entire memory : writing can be done by addressing each location in sequence – This makes it random access memory for writing and content addressable memory for reading – number of lines needed for decoding is d Where m = 2 d , m is number of words.
2. Unwanted words to be deleted and new words to be inserted :
   * Tag register is used which has as many bits as there are words in memory
   * For every active ( valid ) word in memory , the corresponding bit in tag register is set to 1
   * When word is deleted the corresponding tag bit is reset to 0
   * The word is stored in the memory by scanning the tag register until the first 0 bit is encountered After storing the word the bit is set to 1.
   1. **CACHE MEMORY**

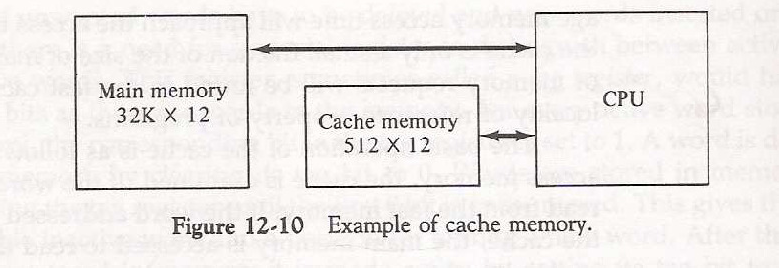
* Effectiveness of cache mechanism is based on a property of computer programs called **“locality of reference”**
* The references to memory at any given time interval tend to be confined within a localized areas
* Analysis of programs shows that most of their execution time is spent on routines in which instructions are executed repeatedly These instructions may be – loops, nested loops , or few procedures that call each other
* Many instructions in localized areas of program are executed
* repeatedly during some time period and reminder of the program is accessed infrequently This property is called “Locality of Reference”.

# Locality of Reference

Locality of reference is manifested in two ways :

1. Temporal- means that a recently executed instruction is likely to be executed again very soon.
   * The information which will be used in near future is likely to be in use already( e.g. reuse of information in loops)
2. Spatial- means that instructions in close proximity to a recently executed instruction are also likely to be executed soon
   * If a word is accessed, adjacent (near) words are likely to be accessed soon ( e.g. related data items (arrays) are usually stored together; instructions are executed sequentially)
3. If active segments of a program can be placed in afast (cache) memory , then total execution time can be reduced significantly
4. Temporal Locality of Reference suggests whenever an information (instruction or data) is needed first , this item should be brought in to cache
5. Spatial aspect of Locality of Reference suggests that instead of bringing just one item from the main memory to the cache ,it is wise to bring several items that reside at adjacent addresses as well ( ie a block of information )

# Principles of cache



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 , there is a duplicate copy in main memory. The Cpu communicates with both memories. It first sends a 15 bit address to cahache. If there is a hit, the CPU accepts the 12 bit data from cache. If there is a miss, the CPU reads the word from main memory and the word is then transferred to cache.

* When a read request is received from CPU,contents of a block of memory words containing the location specified are transferred in to cache
* When the program references any of the locations in this block , the contents are read from the cache Number of blocks in cache is smaller than number of blocks in main memory
* Correspondence between main memory blocks and those in the cache is specified by a mapping function
* Assume cache is full and memory word not in cache is referenced
* Control hardware decides which block from cache is to be removed to create space for new block containing referenced word from memory
* Collection of rules for making this decision is called **“Replacement algorithm ” Read/ Write operations on cache**

# Cache Hit Operation

* + CPU issues Read/Write requests using addresses that refer to locations in main memory
  + Cache control circuitry determines whether requested word currently exists in cache
  + If it does, Read/Write operation is performed on the appropriate location in cache **(Read/Write Hit )**

# Read/Write operations on cache in case of Hit

* + In Read operation main memory is not involved.
  + In Write operation two things can happen.

1. Cache and main memory locations are updated simultaneously **(“ Write Through ”)** OR
2. Update only cache location and mark it as “ Dirty or Modified Bit ” and update main memory location at the time of cache block removal **(“ Write Back ” or “ Copy Back ”)** .

# Read/Write operations on cache in case of Miss Read Operation

* + When addressed word is not in cache Read Miss occurs there are two ways this can be dealt with

1. Entire block of words that contain the requested word is copied from main memory to cache and the particular word requested is forwarded to CPU from the cache **( Load Through )** (OR)
2. The requested word from memory is sent to CPU first and then the cache is updated **( Early Restart )**

# Write Operation

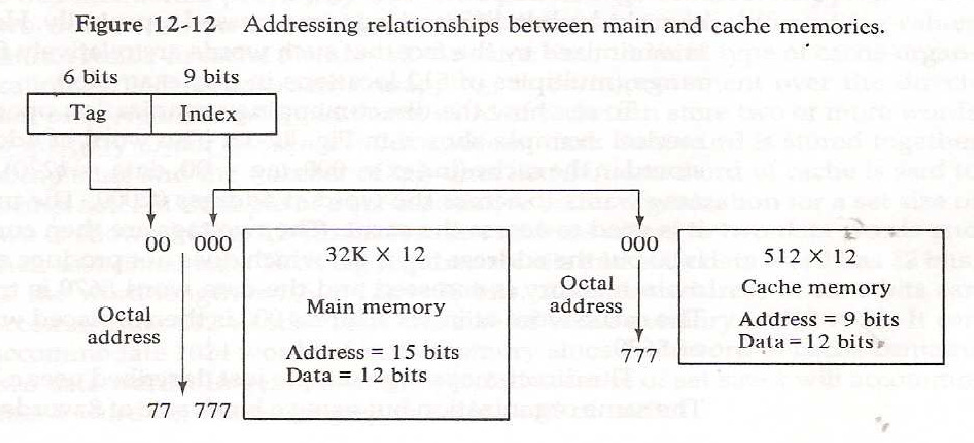
* + If addressed word is not in cache Write Miss occurs
  + If write through protocol is used information is directly written in to main memory
  + In write back protocol , block containing the word is first brought in to cache , the desired word is then overwritten.

# Mapping Functions

* + Correspondence between main memory blocks and those in the cache is specified by a memory mapping function
  + There are three techniques in memory mapping
  1. Direct Mapping
  2. Associative Mapping
  3. Set Associative Mapping

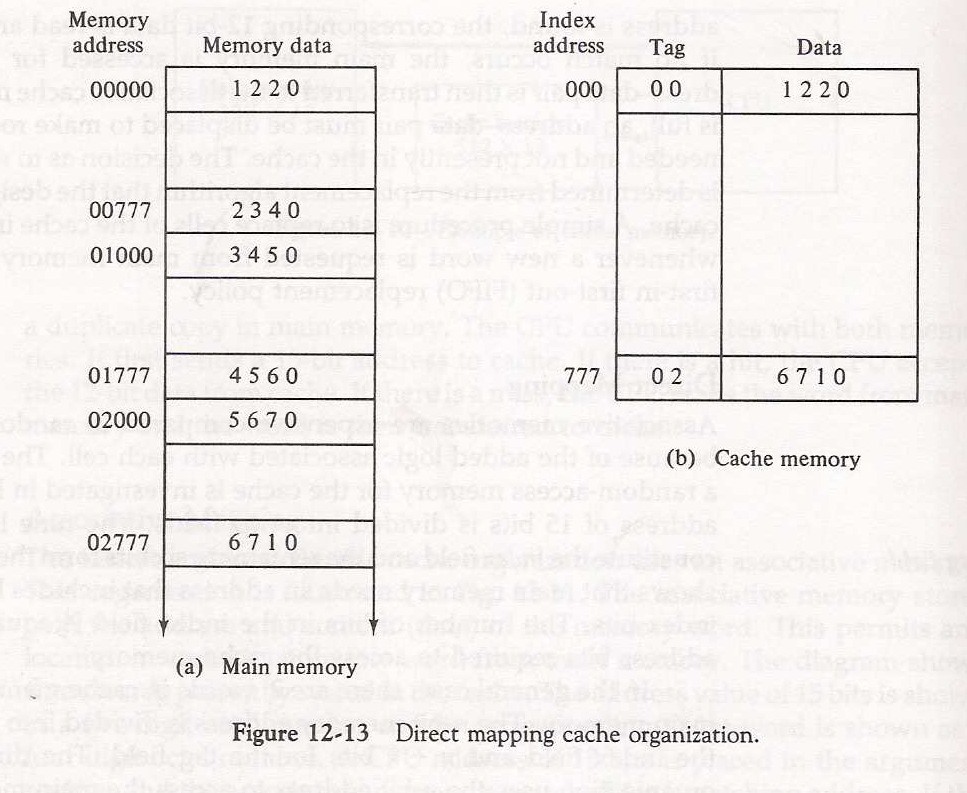
# Direct mapping:

A particular block of main memory can be brought to a particular block of cache memory. So, it is not flexible.

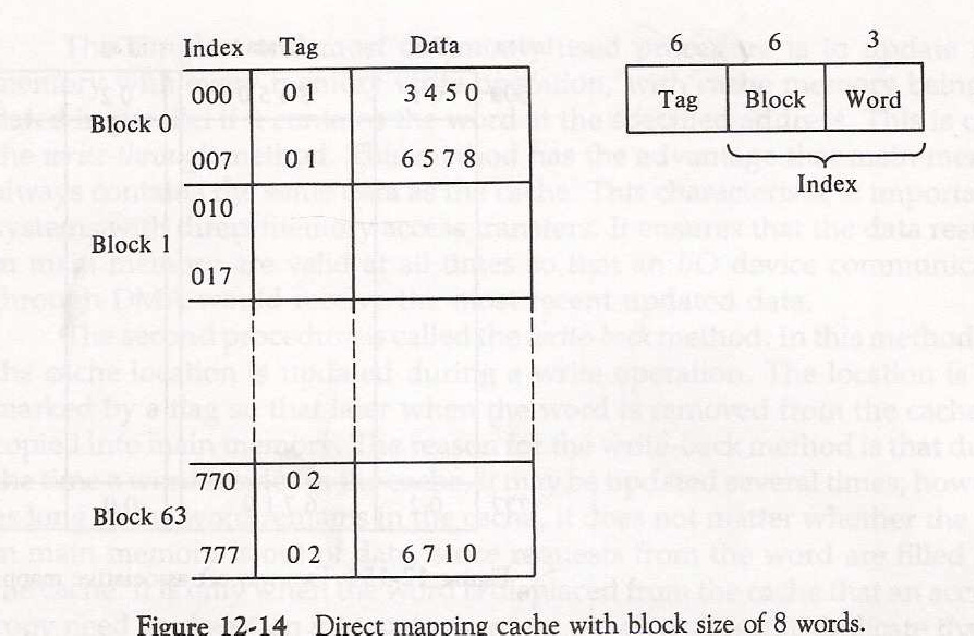


In fig 12-12. The CPU address of 15 bits is divided into two fields. The nine least significant bits constitute the index field and remaining six bits form the tag field. The main memory needs an address

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.



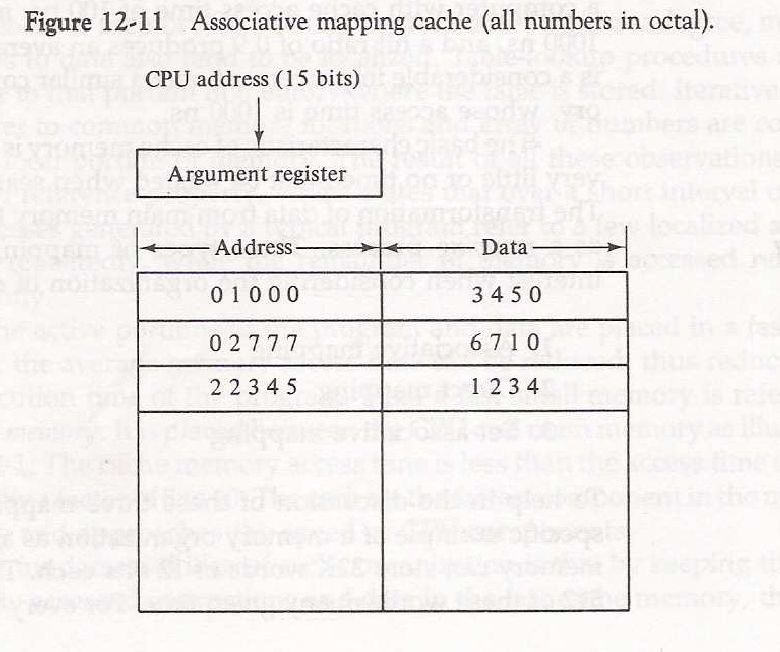
The direct mapping cache organization uses the n- bit address to access the main memory and the k-bit index to access the cache.Each word in cache consists of the data word and associated tag. When a new word is first brought into the cache, the tag bits are stored alongside the data bits.When the CPU generates a memory request, the index field is used 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 anfd 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.



In fig 12-14, The index field is now divided into two parts: Block field and The word field. In a 512 word cache there are 64 blocks of 8 words each, since 64X8=512. The block number is specified with a 6 bit field and the word with in the block is specified with a 3-bit field. Th etag field stored within the the cache is common to all eight words of the same block.

# Associative mapping:

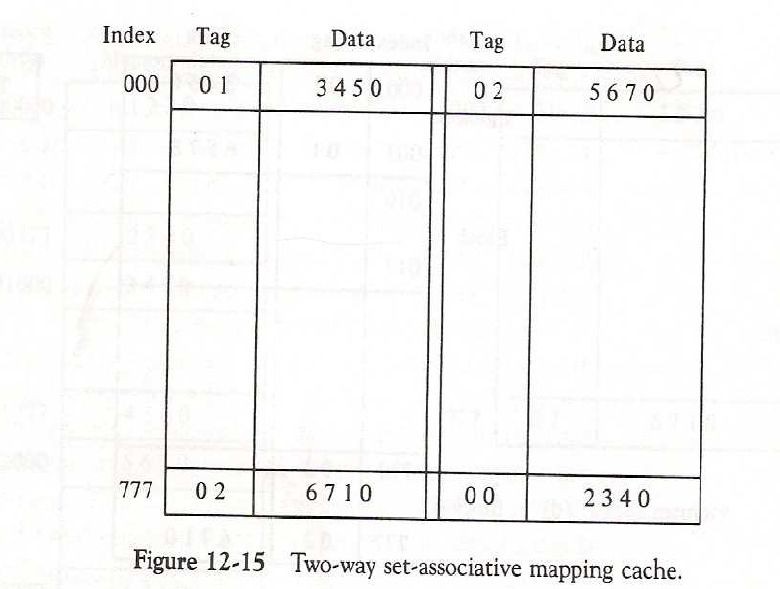
In this mapping function, any block of Main memory can potentially reside in any cache block position. This is much more flexible mapping method.



In fig 12-11, The associative memory stores both address and content(data) of the memory word. This permits any location in cache to store any word from main memory.The diagram shows three words presently stored in the cache. The address value of 15 bits is shown as a five-digit ctal number and its corressponding 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 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.

# Set-associative mapping:

In this method, blocks of cache are grouped into sets, and the mapping allows a block of main memory to reside in any block of a specific set. From the flexibility point of view, it is in between to the other two methods.



The octal numbers listed in Fig.12-15 are with reference to the main memory contents. When the CPU generats a memory request, the index valus of the address is used to access the cache. The tag field of the CPU

address is then compared with both tags in the cache to determine if a match occurs. The comparison logic dine by an associative search of the tags in the set similar to anassociative memory search thus the name “Set Associative”.

# Replacement Policies

* + When the cache is full and there is necessity to bring new data to cache , then a decision must be made as to which data from cache is to be removed
  + The guideline for taking a decision about which data is to be removed is called replacement policy Replacement policy depends on mapping
  + There is no specific policy in case of Direct mapping as we have no choice of block placement in cache Replacement Policies

# In case of associative mapping

* + - A simple procedure is to replace cells of the cache in round robin order whenever a new word is requested from memory
    - This constitutes a First-in First-out (FIFO) replacement policy

# In case of set associative mapping

* + - Random replacement
    - First-in First-out (FIFO) ( item chosen is the item that has been in the set longest)
    - Least Recently Used (LRU)( item chosen is the item that has been least recently used by CPU)