Unit-9

Memory Organization:

⊛. Memory Hierarchy:-

Main goal of memory Hierarchy is to obtain the highest possible access speed while minimizing the total cost of the memory system.

fig. Memory Hierarchy

⊛. Primary (Main) Memory:- The memory which is used by the CPU during program execution is called main memory. It is directly connected with CPU. It is relatively large and fast memory used to store programs and data during the computer operation. Semiconductor integrated circuit is the principle technology used for main memory. RAM, ROM and Cache memory are main memories.

a) Random Access Memory (RAM):- RAM chips are available in two modes static and dynamic.

Static RAM → It consists of internal flip-flops to store binary information. It is easier to use and has shorter read/write cycles.

Dynamic RAM → It stores binary information in the form of electric charges in capacitors. The stored charge tends to discharge with time, so dynamic RAM (DRAM) words are refreshed every few milliseconds to restore the decaying charge.

Scanned with CamScanner

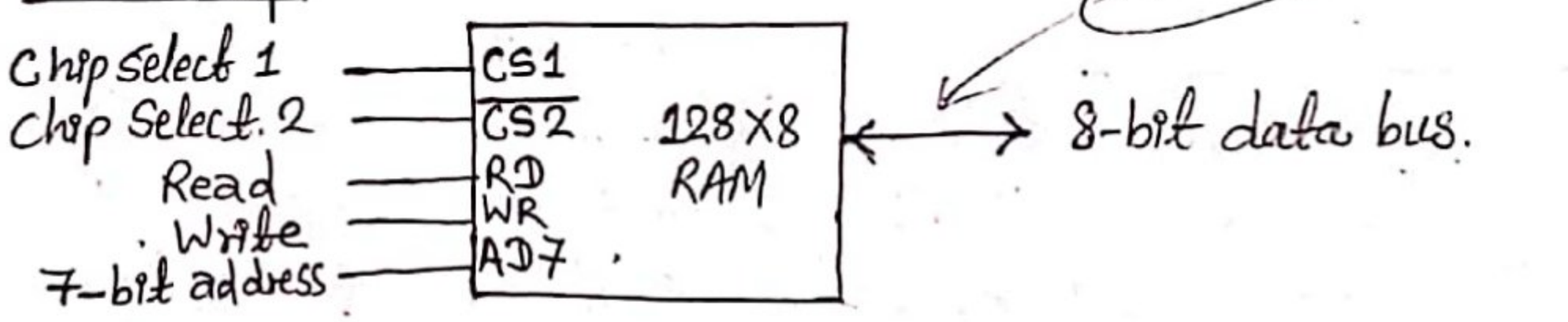
b> Read-Only Memory (ROM):- Random access ROM chips are used for storing programs that are permanently resident in computer and for tables of constants that do not change once computer is manufactured. The content of ROM remain unchanged after power is turned off and on again.

Bootstrap loader → It is initial program whose function is to start the computer operating system when power is turned on and is stored in ROM portion of main memory.

Computer startup → The startup of a computer consists of turning the power on and starting the execution of an initial program. Thus when power is turned on, the hardware of the computer sets the PC to the first address of the bootstrap loader. The bootstrap program loads the portion of the OS from the disk to main memory and control is then transferred to the OS, which prepares the computer for general use.

⊛. RAM and ROM Chips:-

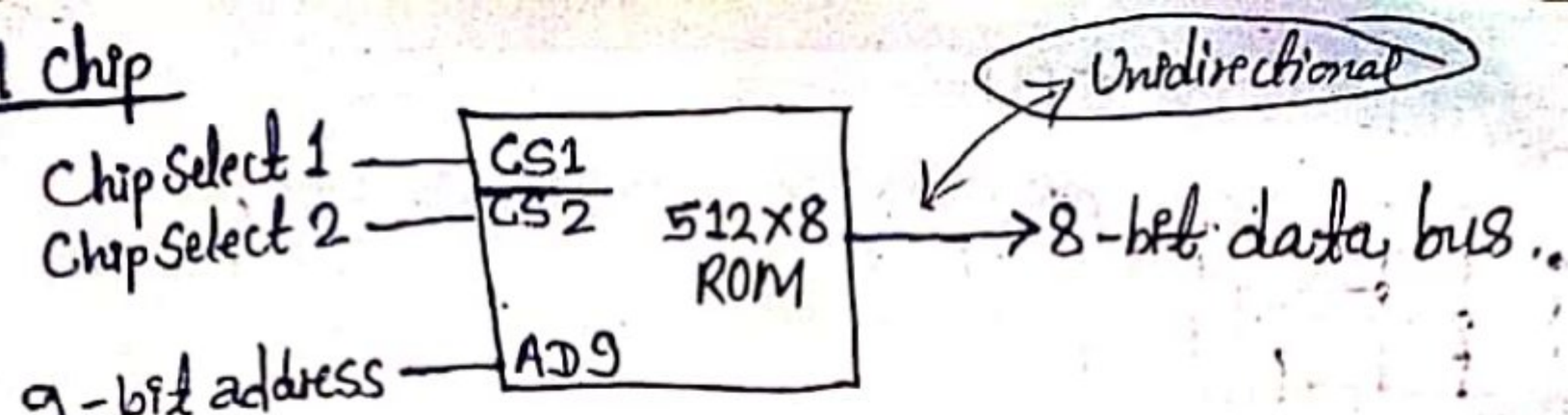
RAM Chip



Two control signals (CS) are used for enabling the RAM chip. Bar above CS 2 indicates that chip is enabled only when CS 1 = 1 and CS2 = 0. RD and WR are read and write control signals that are used to define mode of transfer. Since the size of Ram is of 128 words so we need 7-bit address. Working of chip is described by following function table:

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

ROM chip



Two control signals (CS) are used for enabling ROM chip. Bar above CS2 indicates that chip is enabled only when CS1=1 and CS2=0. RD and WR are not used here because ROM is read-only memory. Since the size of RAM is of 512 words so we need 9-bit address. Working of ROM chip can also be described by similar function table as for RAM chip excluding RD and WR column.

⊛. Memory Address Map:-

Memory address map is the process of assigning address space to a memory system of a computer. Suppose a memory system with 128 words of RAM and 512 words of ROM. If we use RAM chip with 128 words we need to use 4 RAM chips. For this situation memory address map can be done as given in the table below:

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

Address lines 1–7 are used to represent address of RAM chips because their size is 128 words. but address line 1-9 are used to represent address of ROM chip because size of ROM is 512 words.

⊛. 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 selects the byte within the chips and other lines in the address bus selects a particular chip through its chip select inputs.

Scanned with CamScanner

7.

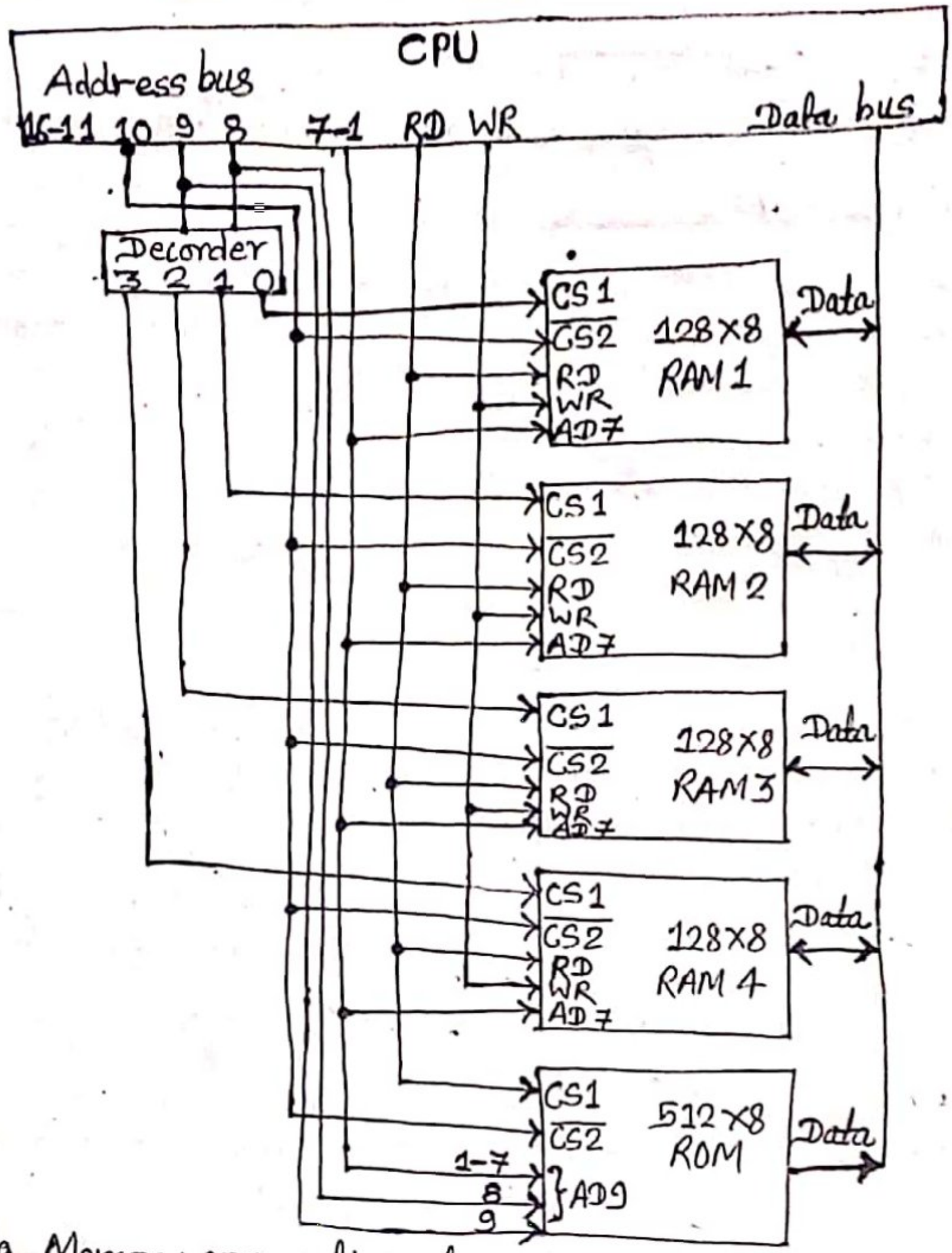


fig. Memory connection to CPU assuming 128X8 RAM and 512X8 ROM.

⊛. Auxiliary (Secondry) Memory:- The most common auxiliary memory devices used in computer systems are magnetic disks and magnetic tapes.

Magnetic disks → A magnetic disk is a circular plate constructed of metal or plastic coated with magnetized material. Bits are stored in magnetized surface in spots along concentric circles called tracks. The tracks are commonly divided into sections called sectors. Disks that are permanently attached to the unit assembly and cannot be removed by the occasional user are called hard disks. A disk drive with removable disks is called a floppy disk.

Scanned with CamScanner

Magnetic tape → Magnetic tape is a strip of plastic coated with a magnetic recording medium. 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. Read/write heads are mounted one in each track so that data can be recorded and read as a sequence of characters.

⊛. Associative Memory:-

⊛ Hardware Organization:.

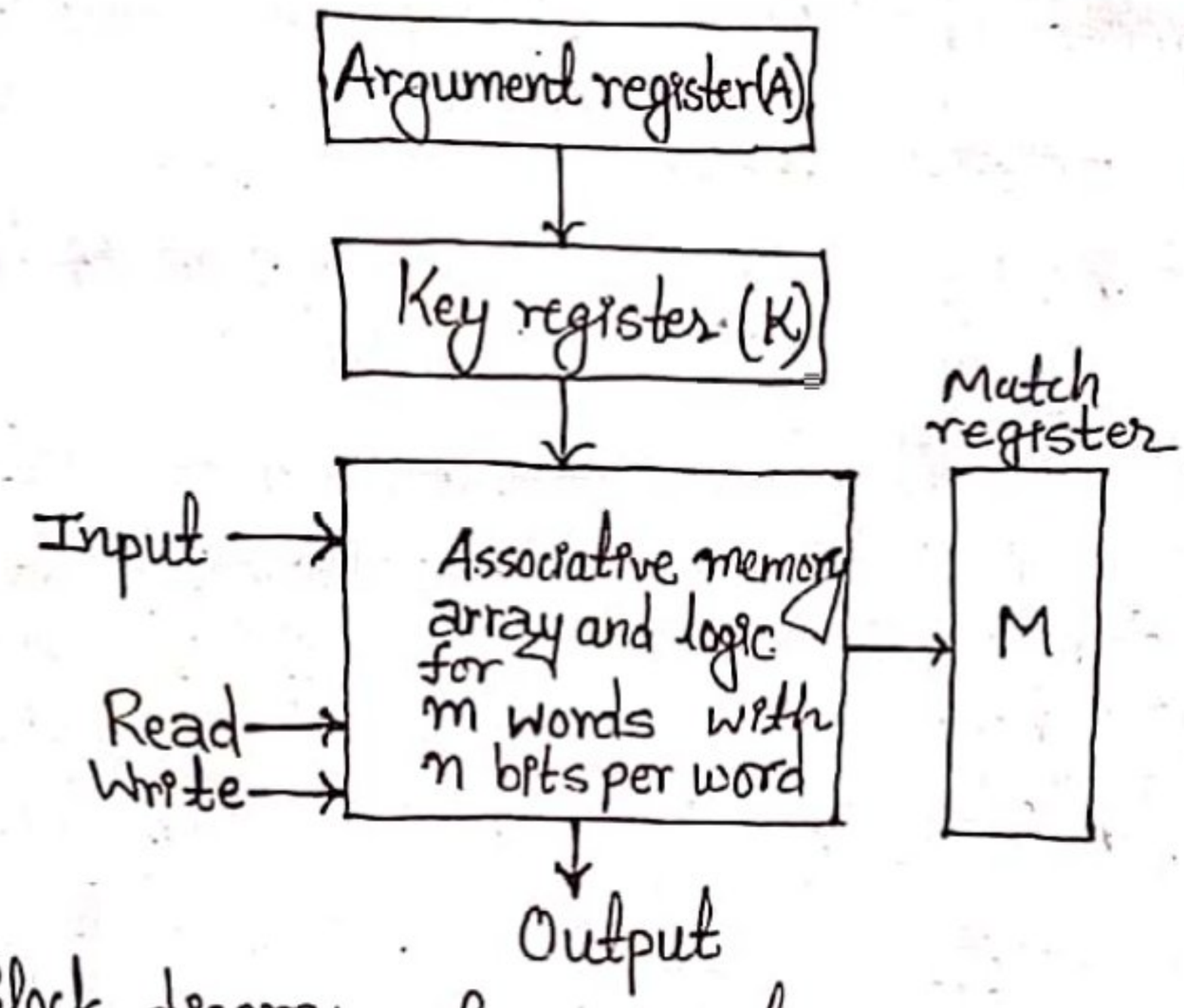


fig. Block diagram of associative memory.

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 word. The ~~M register~~ match register (M) has m bits, one for each memory word. The key provides a mask or identifying piece of information which specifies how the reference to memory is made.

⊛. Match Logic:

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

7. Word i is equal to the argument in A if $A_j = F_{ij}$ for $j = 1, 2, \ldots n$. 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

$$x_j = A_j F_{ij} + A_j' F_{ij}'$$

where, $x_j = 1$ if the pairs of bits in position j are equal
otherwise, $x_j = 0$.

For a word i to be equal to the argument in A we must have all x_j variables equal to 1. This is the condition for setting the corresponding match bit M_i to 1. The Boolean function for this condition is:

$$M_i = x_1 x_2 x_3 \ldots x_n.$$

and constitutes the AND operation of all pairs of matched bits in a word.

⊛ 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. It is then necessary to scan the bits of the match register one at a time. The matched words are read in sequence by applying a read signal to each word line whose corresponding M_i bit is 1.

⊛ Write Operation: An associative memory must have a write capability for storing the information to be searched. Writing in an associate memory can take different forms, depending on the application. If the entire memory is loaded with new information at once prior to search operation then the writing can be done by addressing each location in sequence. If unwanted words have to be deleted and new words inserted one at a time, then there is a need for a special register to distinguish between active and inactive words.

Scanned with CamScanner

⊛. Cache Memory:

Cache is a fast small capacity memory that should hold that information which is most likely to be accessed. The cache memory access time is less than the access time of main memory by a factor of 5 to 10: Cache is the fastest component in memory hierarchy. It is placed between the CPU and main memory as in the figure below:-

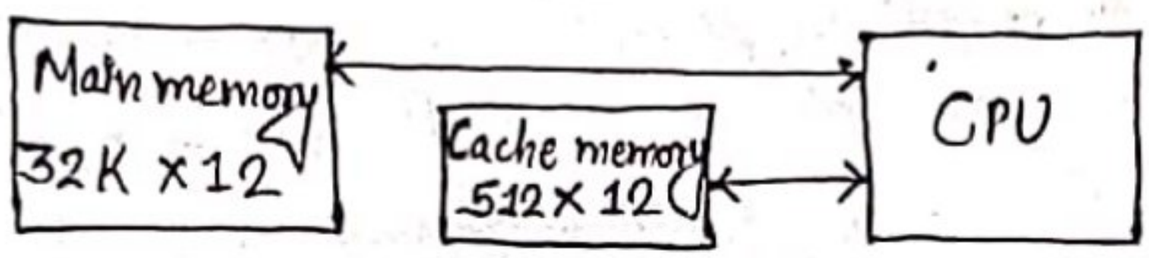


fig. Example of cache memory.

⊛. Locality of refrence: Analysis of a large number of typical programs has shown that the refrences to memory at any given interval of time tend to be confined within a few localized areas in memory. This phenomenon is known as locality of refrence. Loops and subroutines tend to localize the refrences to memory for fetching instructions, this is the reason for this property.

Temporal Locality → The information which is used currently is likely to be in use in near future. For e.g. Reuse of information in loops.

Spatial locality → If a word is accessed, adjacent (near) words are likely accessed soon. For eg. Related data items (arrays) are usually stored together; instructions are executed sequentially.

→ The property of locality of Refrence makes the Cache memory systems work.

⊛. Hit & Miss Ratio:

The performance of cache memory is frequently measured in terms of a quantity called hit ratio. When the CPU refers to memory and finds the word in cache, it is said to produce a hit. If the word is not found in cache, it is in main memory then it counts as a miss. The ratio of number of hits divided by the total CPU refrences to memory (hits plus misses) is the hit ratio.

The hit ratio is best measured experimentally by running representative programs in the computer and measuring the number of hits and misses during a given interval of time. Hit ratios of 0.9 or higher have been reported. This high ratio verifies the validity of the locality of refrence property.

The average memory access time of a computer system can be improved considerably by use of a cache. If the hit ratio is high enough so that the most of the time the CPU accesses the cache instead of main memory, the average access time is closer to the access time of the fast cache memory.

V.Imp

⊛. Mapping:- The transformation of data from main memory to cache memory is referred to as a mapping process. Following three types of mapping procedures are of practical interest when considering the organization of cache memory:

→ OR Cache Mapping

a) Associative mapping → This is the fastest and most flexible method of cache organization. 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. This organization is as below:

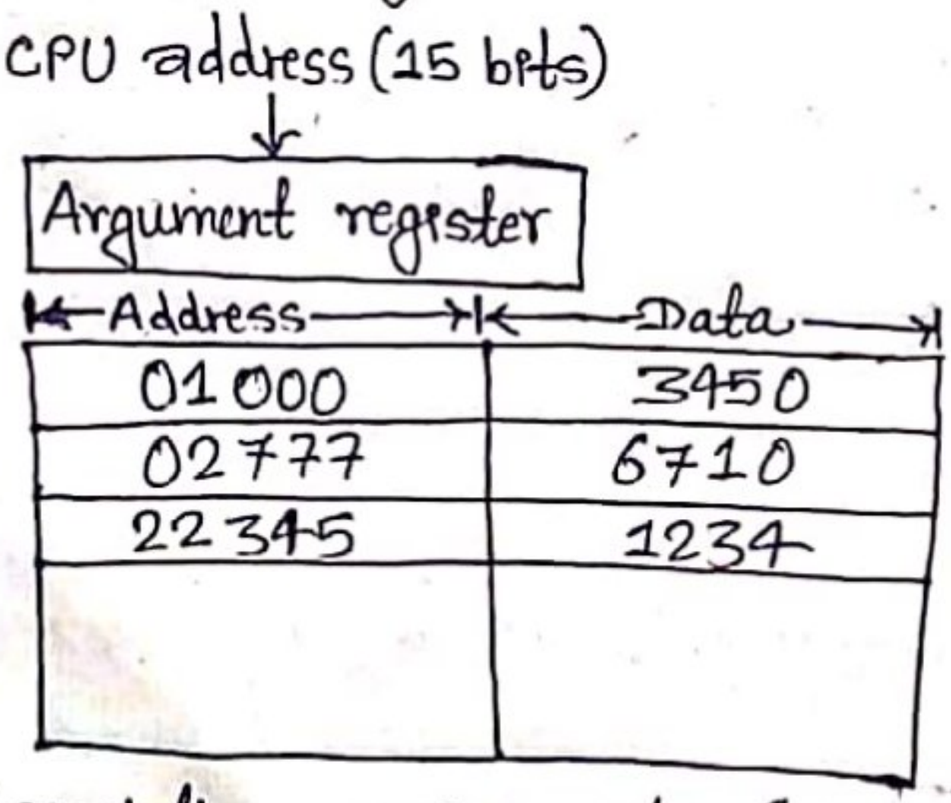


fig. Associative mapping cache (all numbers in octal).

b) Direct mapping → Main memory locations can only be copied into one location in the cache. This is accomplished by dividing main memory into pages that correspond in size with the cache.

Scanned with CamScanner

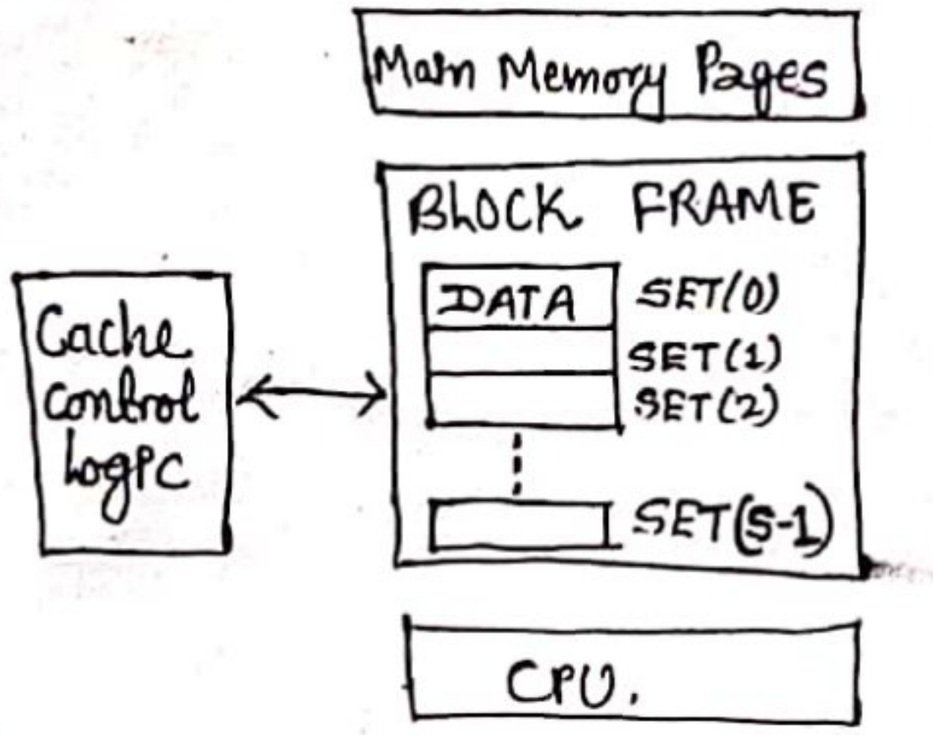


fig. Example of direct mapping used in cache memory.

c) 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 direct mapping organization. So, 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.

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

fig. Example of Set-Associative mapping used in cache memory.

⊛ Write Policies/Writing into Cache: If the operation is write, then there are two ways that the system can proceed. The first is write-through method. This method update the main memory with every memory write operation, with cache memory being updated in parallel if it contains the word at the specified address. This method has the advantage that main memory always contains the same data as the cache.

The second method is 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.

Scanned with CamScanner