Memory Hierarchy

Zatin Gupta.

→ Why do we require additional storage.

→ It is more economical to use low cost storage devices to serve as a backup for storing info. that is not currently used by CPU.

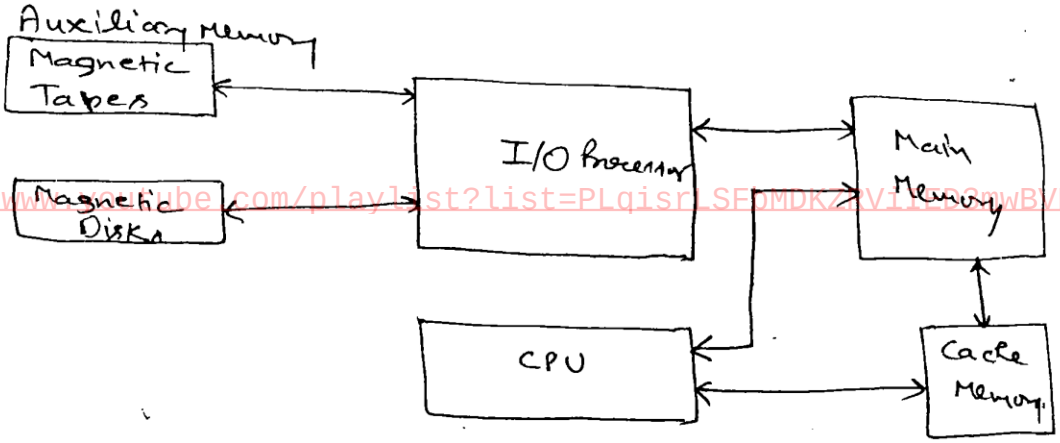
Main Memory: It communicates directly with CPU. It stores only the program or data, which is currently required by CPU.

Auxiliary Memory: It provides back up storage.

Ex: Magnetic Disks, Magnetic Tape.

Memory Hierarchy:

* It consist of all storage devices employed in computer system



- Slow magnetic tapes are used to store removable files.
- Magnetic disks used as back up storage.
- Main Memory communicate directly with CPU & with auxilia memory devices through I/O Processor.

* When program not residing in main memory, are required by C they are brought in auxiliary memory.

Cache Memory:

- Very High speed memory.
- Used to increase the speed of processing by making current programs & data available to CPU at a rapid rate
- Used to compensate for speed difference b/w main memor access time & processor logic (CPU logic)

* Technique used to compensate for mismatch in operating speed is to employ an extremely fast, small cache b/w CPU & main memory whose access time is close to processor logic clock cycle time.

- It is used for storing segments of programs currently being executed in CPU & temp. data frequently needed in present calculations.

I/O Processor:

- It transfer data b/w auxiliary memory & main memory.

* As storage capacity of memory increases, cost per bit for storing info. decreases & access time of memory becomes longer.

Auxiliary Memory: large storage, relatively inexpensive, low access speed.

Cache memory: very small, relatively expensive, very high access speed.

* Memory access speed increase, so does its relative cost.

Goal: To obtain highest possible average access speed while minimizing total cost of entire ~~memory~~ memory system.

* CPU has not direct access to auxiliary memory,

* Access time ratio b/w cache & main memory is 1 to 7.

* Auxiliary memory average access time is usually 1000 times that of main memory.

* Block size in auxiliary memory ranges from 256 to 2048 words, while cache block size is from 1 to 16 words.

Multiprogramming:

- It refers to two or more independent programs in different parts of memory hierarchy.
- In this, when one program is waiting for input or output transfer there is another program ready to utilize CPU.

Semiconductor RAM Memories

Semiconductor RAM or RAM or R/W Memory or Scratch Pad Memory

- Semiconductor RAM refers to semiconductor IC Memories, that can be used in 'Read' mode as well as in 'Write' mode
- It uses either a read cycle or a write cycle depending on type of request. Read cycle has shorter time than write cycle.
- Semiconductor memories are <u>non destructive Read Out & volatile memories</u>.

Nondestructive: Data stored in memory is not destroyed by procedure used to read data from memory cells.

Volatile: Requires electrical power to maintain storage. For this reason UPS & battery back up system are used in semiconductor memory to maintain constant power.

RAM (Random Access Memory):- Data can be read & written in any desired order from any location.

Term 'RAM' is not used for Read only memory, although a ROM can be random access.

Two main types of RAM:-

(1) S RAM (Static RAM) — Faster than DRAM, Requires more logic, more expensive. Does not need to be refreshed again & again. Bit is stored as voltage in flip flops.

(2) D RAM (Dynamic RAM) — Bit is stored as charge in MOSFET & capacitor. It requires refreshing for recharging capacitors. Reduced power consumption

Examples of Semiconductor Memories →

(1) Magnetoresistive Random Access Memory (MRAM)

(2) Flash Memory (EEPROM) - Non volatile

(3) ROM

— Difference b/w RAM & ROM —

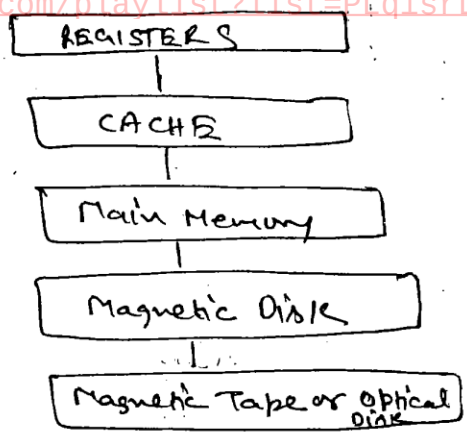
RAM — R/W Memory, for temporary storage (volatile)
ROM — Read only Memory, for Permanent storage (Nonvolatile)
} Both are Random Access Memory.

* Main Memory in Computer system has two parts i.e. —

(1) RAM
(2) ROM — for storing boot strap loader.

Boot Strap loader:

- It is a program whose function is to start computer operating S/W when power is turned on.
- It loads a portion of O.S. from disk to main memory & control is then transferred to O.S.



PREPARED BY: MR. PIYUSH GUPTA, MR. ZATIN GUPTA

https://www.youtube.com/playlist?list=PLqisrLSFbMDKZRViIED3mwBVHl5c4LG-L

RAM Chip:- RAM CHIP :-

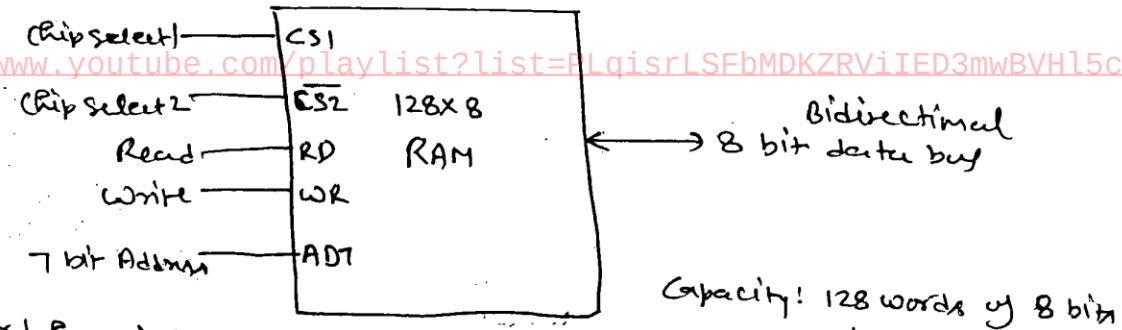
Bidirectional Bus: It allows transfer of data either from memory to CPU (during a read operation).

from CPU to memory (During a write operation).

* It can be constructed with 3-state buffer.

* 3 State buffer O/P can be placed in one of 3 possible states
  - A signal equivalent to logic 1
  - Signal equivalent to logic 0
  ] - Normal digital signals.
  - High Impedance state → like an open circuit, which means that O/P does not carry a signal & has no logic significance

Block Diagram:-



Capacity: 128 words of 8 bits per word.
word length = 8 Bits

* Each word has independent address & each have same no. of bits.

Function table:-

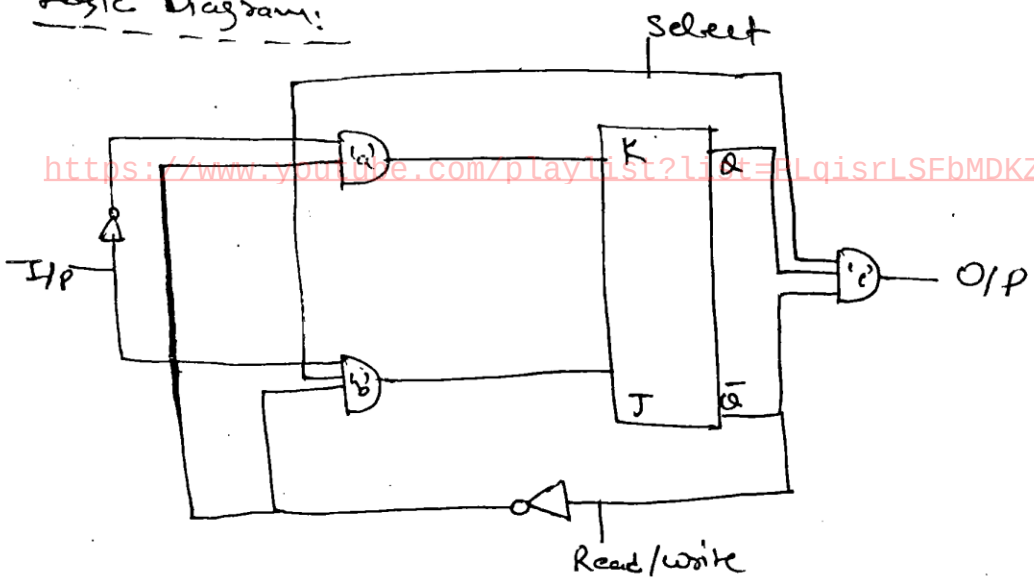
CS1	$\overline{CS2}$	RD	WR	Memory function	State of data Bus
0	0	X	X	Inhibit	High Impedance
0	1	X	X	Inhibit	
1	0	0	0	Inhibit	
1	0	0	1	Write	Input data to RAM
1	0	1	X	Read	Output data from RAM
1	1	X	X	Inhibit	High Impedance

CS: Chip Select: Control inputs are for enabling chip only when it is selected by microprocessor.

* Availability of more than one control input to select the chip facilitates the decoding of address lines when multiple chips are used in microcomputer.

* Read & write inputs are combined into one line R/W Sometimes.
* If chip select inputs are not enabled or if they are enabled but read or write inputs are not enabled, memory is inhibited & its data bus is in High impedence state.
* when CS1=1 & $\overline{CS2}$=0, memory can be placed in write or read mode.
* when WR=1 - Memory stores a byte from data bus into a location specified by address input lines.
* when RD=1: Content of selected byte is placed into data bus.

Logic Diagram:



RAM for storing one bit - (Binary cell) of RAM.

* Access time & cycle time in RAM are constant & independent of location accessed.

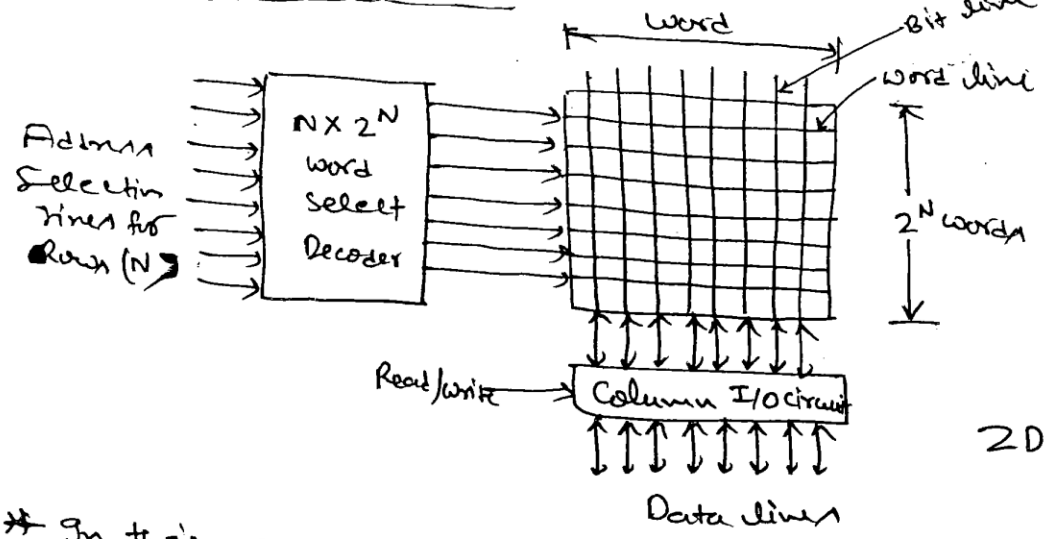
PREPARED BY: MR. PIYUSH GUPTA, MR. ZATIN GUPTA

https://www.youtube.com/playlist?list=PLqisrLSFbMDKZRViIED3mwBVHl5c4LG-L

Chip Organization: < 2D organization
                    < 2½D organization.

- Semiconductor memories are packaged in chips.

① 2D Memory Organization:



2D Organisation.

* In this memory on chip is considered to be a list of words in which any word can be accessed randomly

Q: In a memory of size 64 KB, how many words it has of length 16 bit. Ans: 32K words.

* In 2D, memory is organized on array of words.
* Each decoded line of decoder drives a word line so complete word can be input or output from the memory simultaneously.

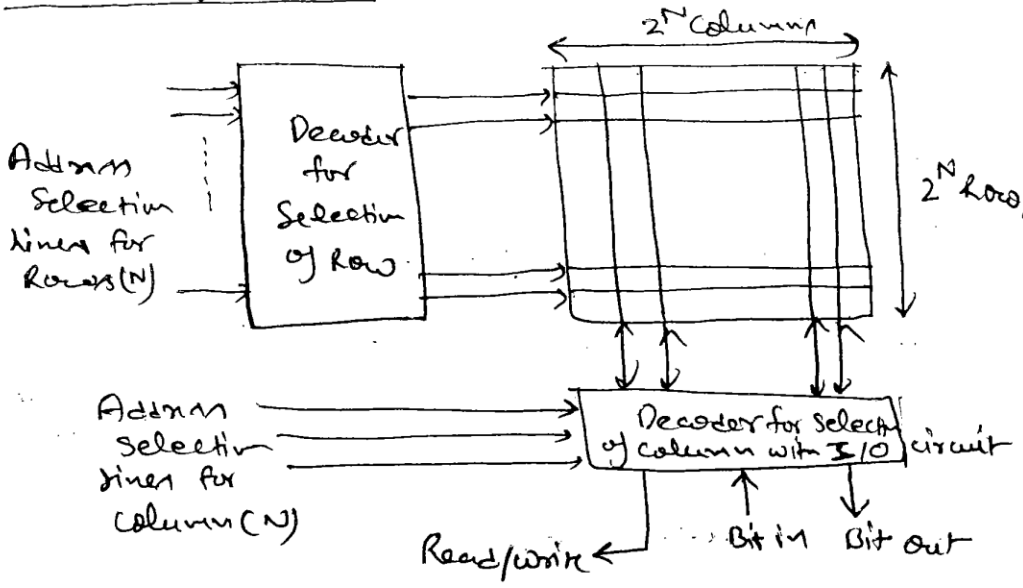
Write: Address decoder selects required word & bit lines are activated for a value 0 or 1, acc. to data line values.

Read: Value of each bit line is passed through data lines.

PREPARED BY: MR. PIYUSH GUPTA, MR. ZATIN GUPTA

https://www.youtube.com/playlist?list=PLqisrLSFbMDKZRViIED3mwBVHl5c4LG-L

② 2½ D organization:



* In 2½ D organization, bits of a word are spread over a number of chips.

Ex: 32 bit word can be stored on 4 chips containing 8 bits of the word.

Ideal organisation: 1 bit of a word on a single chip.

* A row line & a column line are connected to each memory cell.

* Addresses supplied to this chip are divided into Row & column address lines & are then used to input or output bit/bits from this memory chip.

Comparision of 2D & 2½ D Organisation:-

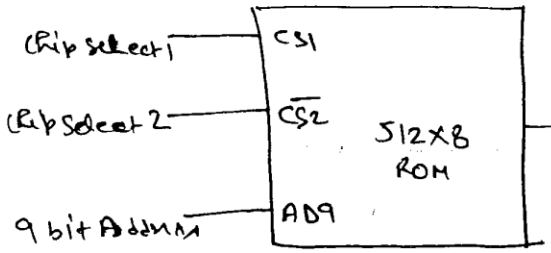
* 2½ D organisation of chips is supposed to be more advantageous. because-

(1) It requires less circuitry & gates.

(2) Chip have only one input/output pin in 2½ D while in 2D it has to have 16 or 32 input/output pins, thus in chip packages less no. of pins are required for 2½ D organisation.

ROM Memories

ROM Chip :



Block Diagram.

CS1	CS2	Operation
0	0	High Impedance
0	1	High Impedance
1	0	Read Operation
1	1	High Impedance

* For the same size chip, it is possible to have more bits of ROM than of RAM, because internal binary cells in ROM occupy less space than in RAM.

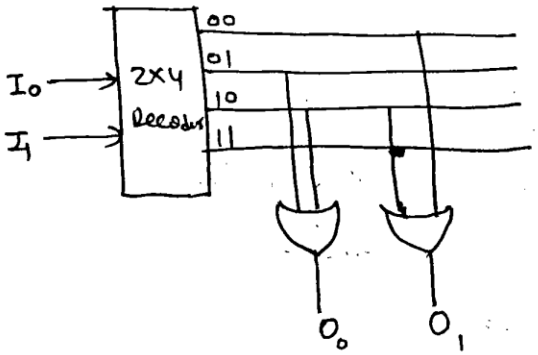
* Two chip select inputs must be CS1=1 & $\overline{CS2}$=0 for unit to operate, otherwise data bus is in a high impedance state.

* There is no need for a read or write control because unit can only read.

* When chip is enabled by two select inputs, byte selected by address lines appears on data bus.

Combinational circuit for ROM :-

Input		Output	
I_0	I_1	O_0	O_1
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



* ROM are memories on which it is not possible to write data when they are on line to the computer. They can only be Read.

* Non volatile in nature.

Types of ROM:-

(1) PROM: Programmable Read Only Memory :⇒
- Also non-volatile in nature.
- Writing process can be performed electrically.
- PROMs are more flexible & convenient than ROMs.

(2) EPROM: Erasable PROM:-
- Can be read & written electrically.
- Write operation is not simple.

Write: It requires erasure of whole storage cells by exposing the chip to ultra violet light.
- This erasure is time consuming process.
- Once all cells have been brought to same initial state then EPROM can be written electrically.

(3) EEPROM:
- It do not require prior erasure of previous contents.
- writing time is higher than reading time.

Adv.: Non volatile memory & can be updated easily.

Dis adv. ⇒ High cost, write operation take too much time.

* ROMs are made of cheaper & slower technology than RAMs.

** Flash Memory:- Can be programmed at high speed
- Cost & write time fall in b/w EPROM & EEPROM.

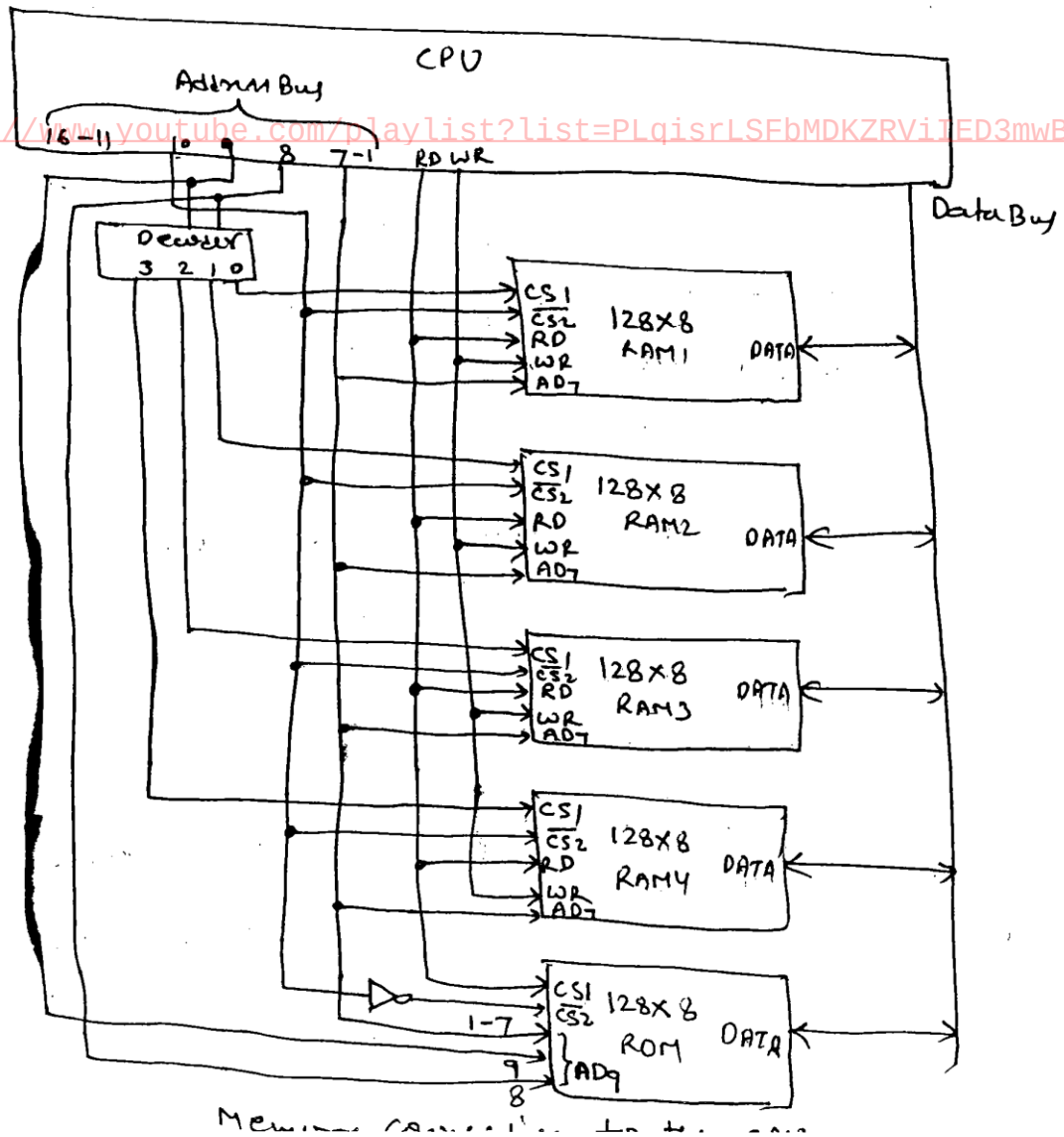
* In flash memory, entire memory can be erased in few seconds by using electric erasing technology.

Features of ROM Memories :-

Memory Type	Write time	Order of Read Time	No of write Cycle allowed
ROM	Once	Nano Sec.	One
PROM	Hours	Nano Sec.	Tens
EPROM	Minutes (Inc. time of Erase)	Nano Sec.	Hundreds
EEPROM	Milliseconds	Nano Sec.	Thousands.

Memory Address Map :-

- Mapping B/w RAM chips & Range of addresses in RAM.



Memory Address Map for Micro ~~com~~ pro computer (Above)-:

Component	Hexadecimal 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

CACHE MEMORY

Locality of Reference: References to memory at any given interval of time tend to be confined with in a few localized area in memory. This phenomenon is known as property of locality of Reference.

* Over a short interval of time, address generated by a typical program refer to a few localized areas of memory repeatedly, while remainder of memory is accessed relatively infrequently.

Cache memory: It is fast, & small memory for storing active portions of program & data for reducing average memory access time so it reduce total execution time of program.

* It is placed b/w CPU & Main memory.
* Cache memory access time is less than access time of main memory by a factor of 5 to 10.
* Fastest component in memory hierarchy & approaches the speed of CPU.
* Most frequently accessed instructions & data in fast cache memory, ~~average memory access time~~ cache is only a small fraction of size of main memory, a large fraction of memory requests will be found in fast cache because of locality of reference property of programs.

Basic Operation of cache:

* When CPU needs to access memory, cache is examined. If word is found in cache, it is read from fast memory. If the word addressed by CPU is not found in cache, main memory is accessed to read the word.

Hit Ratio: When CPU refers to memory & finds the word in cache, it is said to produce a 'hit'. If word is not found in cache, it is in main memory & it count as a 'miss'.

* Ratio of the number of hits divided by total CPU references to memory is hit ratio.

* Standard hit ratio is 0.9.

* Hit ratio which is high, varifies the validity of locality of refernce

Ex:

A computer with cache access time of 100 ns, main memory access time is 1000 ns, & hit ration is 0.9. Then find out average access time for accessing 100 words.

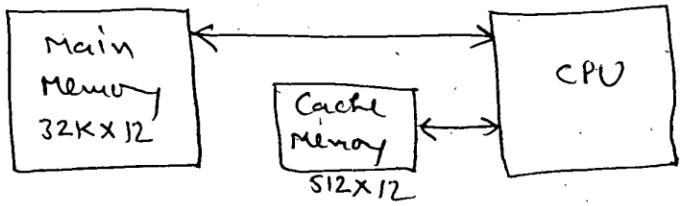
Avg. access time $= \frac{(0.9 \times 100 + 0.1 \times 1100)}{100} = \frac{90 + 110}{100} = \frac{200}{100} = 2$

$= 200$ ns Ans

Mapping: Transformation of data from main memory to cache memory is refered to mapping process.

Types:
(1) Associative Mapping
(2) Direct Mapping
(3) Set Associative Mapping.

Ex:

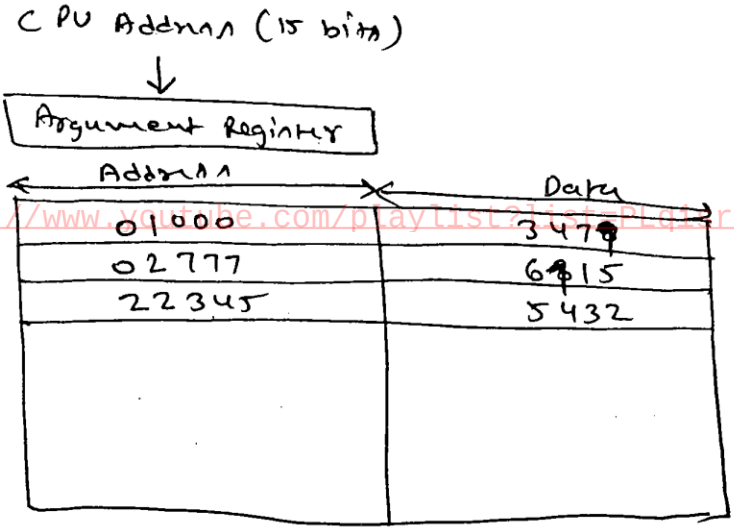


M.M. - 32K words of size 12 bit.
Cache - 512 words of size 12 bit

* for every word stored in cache, there is a duplicate copy in main memory.
* CPU sends address to cache, if there is a hit, it accepts data from cache, if there is a miss, it reads word from main memory & word is transferred to cache.

(1) Associative Mapping :-

- Fastest & most flexible cache organization.
- It stores both address & content of memory word.
- It permits any location in cache to store any word from main memory.



Associative Mapping Cache (All numbers are in octal)

* CPU address of 15 bits is placed in argument register & associative memory is searched for matching address. If address is found, 12 bit data is read & sent to CPU. If no match occurs, Main memory is accessed for the word. Address – Data pair is then transferred to associative cache memory.
* If cache is full, address – data pair must be replaced either. Decision as to what pair is replaced is determined from replacement algorithm that the designer chooses for cache.
* Whenever a new word is requested from main memory. It constitutes FIFO replacement policy.

(2) DIRECT MAPPING:

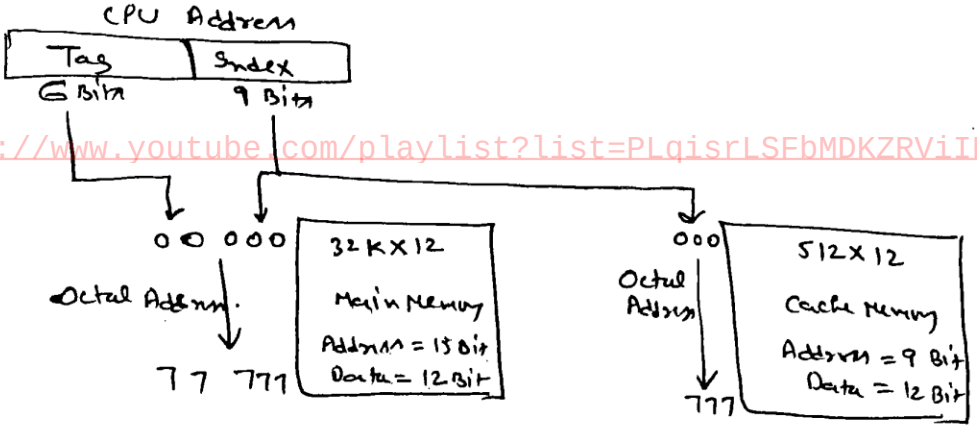
* Associative memory are expensive compared to RAM because of added logic associated with each cell.

Tag field: CPU Address of 15 bits is divided into two fields.

(1) Index field: Nine least significant bits.

(2) Tag field: Remaining Six bits.

* Main memory needs an address that includes both tag & index bits.
* No of bits in index is equal to no. of address bits required to access cache memory.



In general –

If No of words in cache memory = 2^K

No of words in Main Memory = 2^n

then n bit memory address is divided into two fields –

Tag field: size (n-k) bit

& Index field: size K bit.

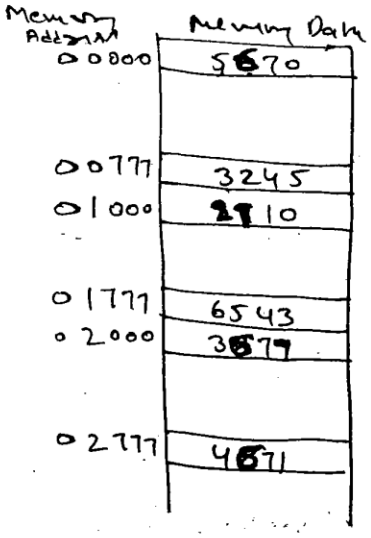
Then It uses n bit address to access main memory & K bit address to access cache memory.

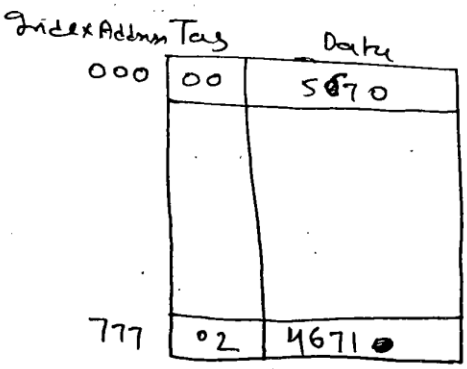
PREPARED BY: MR. PIYUSH GUPTA, MR. ZATIN GUPTA

https://www.youtube.com/playlist?list=PLqisrLSFbMDKZRViIED3mwBVHl5c4LG-L

Internal organization of words in cache memory –

* Each word in cache consist of data word & associated tag.





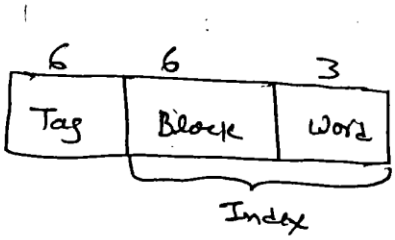
* When new word is brought into cache, tag bits are stored along with data bits.

* When CPU generates a request, index field is used for address to access the cache, the tag field of CPU Address is compared with tag in word read from cache. If the two tags match, then is a hit & desired word is in cache.

* If there is no match then is a miss & required word is read from main memory. It is then stored in cache together with new tag.

Direct mapping Cache with blocks:

	Index	Tag	Data
Block 0	000	01	3450
	007	01	6560
Block 1	010		
	017		
	⋮	⋮	⋮
Block 63	770	02	4321
	777	02	6710



Block size of 8 words

* Index field is now divided into two parts – (1) Block field
(2) Word field.

* Block no. is specified with 6 bit field & word with in block is specified with 3 bit field.

* Tag field is common to all eight words with in a block.

* When a miss occurs, an entire block of eight words must be transferred from main memory to cache memory.

(3) Set Associative Mapping: ++

Disadv. of Direct Mapping: Two words with same index in their address, but with different tag values can not reside in cache memory at the same time.

* In this each word of cache can store two or more words of memory under same index address.

Set: Each data word is stored together with its tag & 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

Two-way set Associative mapping cache.

* Each index address refers to two data words & their associated tags.

  Word length = 2 (6+12) = 36 Bits, 6 bits of each tag (No of tag = 2)
  12 Bits of data word (No of Words = 2).

* Now size of cache memory 512X36, It can accommodate 1024 words of main memory.

* A set associative cache of set size K will accommodate K words of main memory in each word of cache

* When CPU generates memory request, Index value of address is used to access cache. Tag field of CPU address is then compared with both tags in the cache to determine if match occurs.

* Hit ratio will improve as the set size increases bcoz more words with same index but different tag can reside in cache.

* Replacement Algorithms are FIFO (first in first out), LRU (Least Recently used).

  FIFO: It select for replacement the item that has been in the set longest.

  LRU: It select for replacement the item that has been least recently used by CPU.

* When a miss occurs then one of the above two algos are used.

PREPARED BY: MR. PIYUSH GUPTA, MR. ZATIN GUPTA

WRITING INTO CACHE :-

* For write operation there are two ways —

(1) To update main memory with every memory write operation with cache memory being updated in parallel if it contains the word at specified address.

Write through method.

Adv: — Main memory always contains same data as cache.
— Important in DMA transfers.
— It ensure validity of recent updated data.

(2) Write Back Method :-

— Only cache location is updated during write operation.
— location is then marked by a flag, so that later when word is removed from cache it is copied into main memory.

Reason: During the time a word resides in the cache, it may be updated several times. It does not matter whether copy in main memory is out of date, since requests from word are filled from the cache.

* When word is removed from cache, accurate copy of that word need be rewritten into main memory.

Q- A computer system has 4K word cache organized in block set associative manner with 4 Blocks per set, 64 words per block, The main memory contains 65536 Blocks. How many bits are there in each of TAG, SET and word fields.

Cache Initialization:

* Cache is initialized, when power is applied to the computer or when main memory is loaded with a complete set of programs from auxiliary memory.
* After initialization, cache is empty. (It contains nonvalid data).

Valid bit: It indicate that whether or not the word contains valid data.

* Cache is initialized by clearing all valid bits to 0.
* Valid bit of particular cache word is set to 1 the first time this word is loaded from main memory & stays set unless the cache has to be initialized again.
* If valid bit happens to be 0, new word automatically replaces invalid data.
* Initialization condition has the effect of forcing misses from cache until it fills with valid data.

PREPARED BY: MR. PIYUSH GUPTA, MR. ZATIN GUPTA

AUXILIARY MEMORY

Ex:
- Magnetic Disks
- Magnetic Tapes
→ Most common auxiliary memory.
- Magnetic Drums
- Magnetic Bubble Memory
- Optical Disks.

Imp. Characteristics of any device are
- Access mode
- Access Time
- Transfer Rate
- Capacity
- Cost.

Access Time: Average time required to reach a storage location in memory & obtain its contents called access Time.

Seek Time: Used in electromechanical devices disk & Tape.
- Required to position the read-write Head to a location

Latency: Time to reach particular sector under Read/write Head (track)

Transfer Time: Required to transfer data to or from the device.

* Because seek time is much longer than transfer time, auxiliary storage is organized in records or blocks.

Record: Specified no of characters or words.

Transfer Rate: No. of characters or words that the device can transfer per second, after it has been positioned at the beginning of the record.

Magnetic Drum:- Similar to disks in operation.
- Consist of high speed rotating surface coated wit magnetic recording medium.

- Rotating surface of a drum is cylinder.
* Recording surface rotates at uniform speed & is not started or stopped during access operation.
* Bits are recorded as magnetic spots on surface → write head.

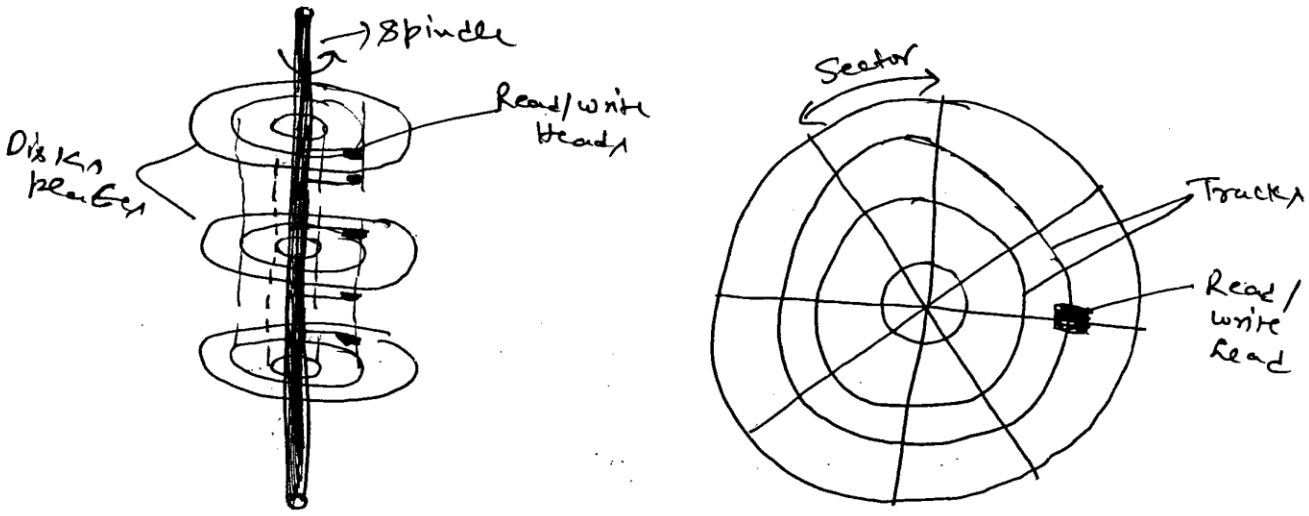
Read Head: Stored bits are detected by a change in magnetic field produced by recorded spot on the surface as it passes through read head.

* More information can be stored on a disk than on a drum of comparable size. So disks have replaced drums.

(2) Magnetic Disks:

- It is a circular plate constructed of metal or plastic coated with magnetized material.

- Both sides of disks are used for storage (Except upper most & lower most), all disks are stacked on a spindle with read/write heads available on each surface.



* All disks rotate together at high speed & are not stopped or started for access purposes.

Tracks: Bits are stored in magnetized surface in spots along concentric circles called tracks.

Sections: Tracks are commonly divided into sections called Sectors (Sectors).

- Sector is minimum quantity of information which can be transferred.

* Some disks are used single Read/write head. (Movable from one track to other).
* Some disks uses separate read/write heads for each track in each surface.

Address bits in disk system specifies -
- Disk Number.
- Disk Surface
- ~~Sector~~ Track Number
- ~~Track~~ Sector within the ~~sector~~ track.

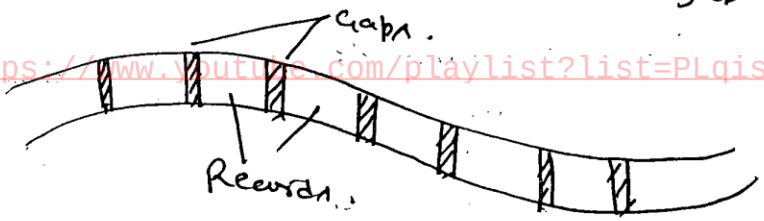
* After read/write Heads are positioned in specified track, system has to wait until rotating disk reaches the specified sector under the read/write head.
* Information transfer is very fast once beginning of sector has been reached.
* Disks have multiple heads & simultaneous transfer of bits from several tracks at the same time.

Density:-
* ~~A track~~ Sector in a given ~~sector~~ track near circumference is longer than a ~~track~~ sector near the center of disk.
* If bits are recorded with equal density, some tracks will contain more recorded bits than others.
* To make all records in a sector of equal length, some disks use a variable recording density with higher density on tracks near the center than on tracks near the circumference.

(3) Magnetic Tape: - Used in Cartridge & Cassette.
- Sequential Access Device

- It consist of electrical, mechanical & electronic components to provide the parts & control mechanism for a magnetic tape unit.

* It is a strip of plastic coated with a magnetic recording medium.
* Bits are recorded as magnetic spots on tape along several tracks.
* Read/write heads are mounted one in each track so that data can be recorded & read as a sequence of characters.
* Magnetic tape can be stopped, started to move forward or in reverse.
* Information is stored in blocks called record.
* Between every record there is a gap.



* Each record on tape has an identification bit pattern at beginning & end.
* By reading bit pattern at the end of record, control recognize ~~begin~~ beginning of gap.
* By reading bit pattern at the beginning, tape control identifies record number.
* A tape unit is addressed by specifying the record number & no. of characters in record.
* Records may be of fixed or variable length.

Virtual Memory

* It is a concept used in large computer systems that permits the user to construct programs as though large memory space available equal to total auxiliary memory.
* Address referenced by CPU goes through an address mapping from virtual address to physical Address in Main Memory.
* It provide an illusion that they have very large memory.
* It provides a mechanism for translating program generated address into correct main memory location.

Address Space & Memory Space:

Virtual Address: Address used by programmer.

Address Space: Set of virtual Addresses. (Set of Addresses generated by programs).

Physical Address: Address in main memory.

Memory Space: Set of physical Addresses. (Actual main memory locations)

* Address Space is larger than memory space due to virtual memory.

Ex:

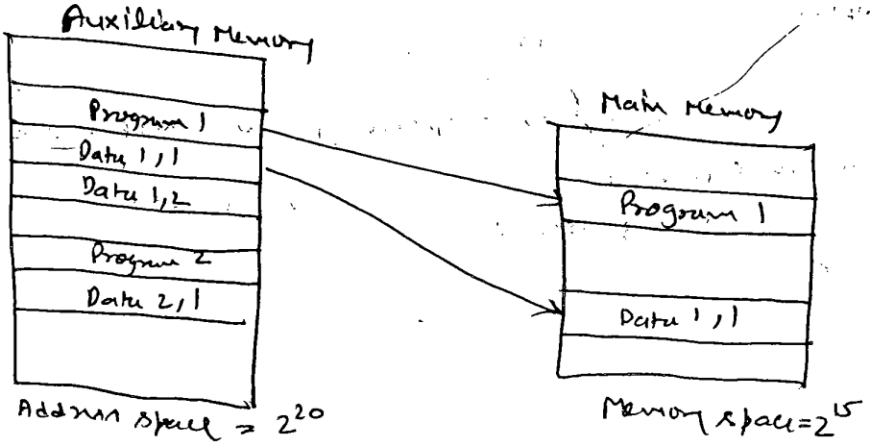
Main memory has capacity = 32K = 2^{15} words.

Auxiliary memory has capacity = 2^{20} = 1024 K words.

Now Address space (N) = 1024 K &

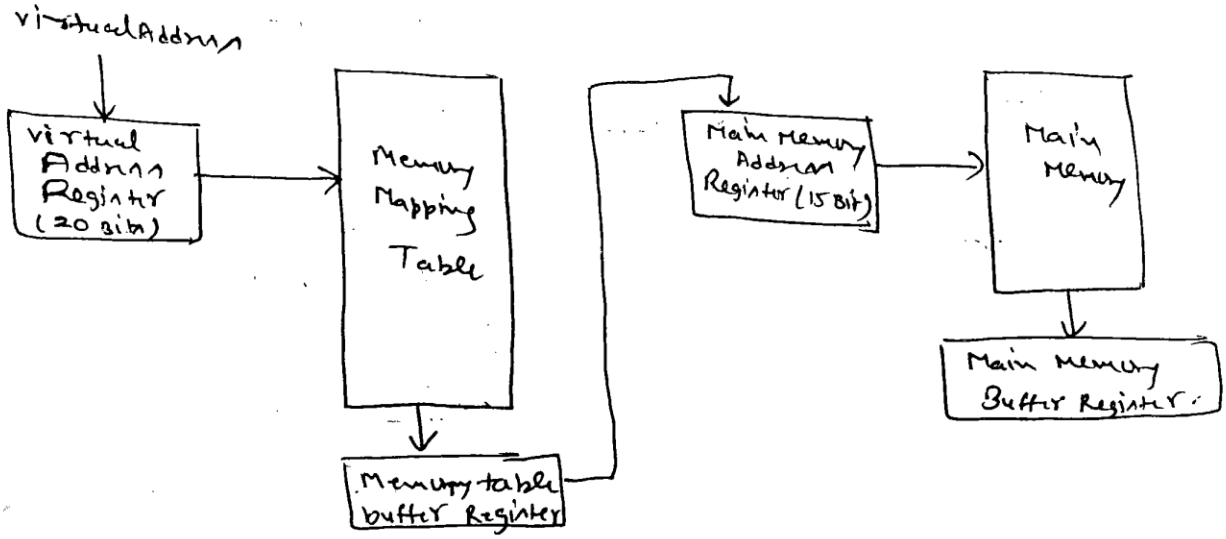
Memory Space (M) = 32 K.

Ex:



* Address field of instruction code consist of = 20 Bits.
  Physical memory Addresses specified with = 15 Bits.

Memory Table for mapping virtual Address :-



* Mapping is dynamic operation, means every address is translated immediately as a word is referenced by CPU.

* Storing of Mapping table :
  (a) A Seperate Memory – Additional memory unit is required as well as one extra memory access time.
  (b) In Main Memory – Table takes space from main memory & two accesses to memory.
  (c) Associative Memory.

Address Mapping Using Pages :-

* Physical memory is divided into group of equal size : Blocks. (from 64 to 4096 words)

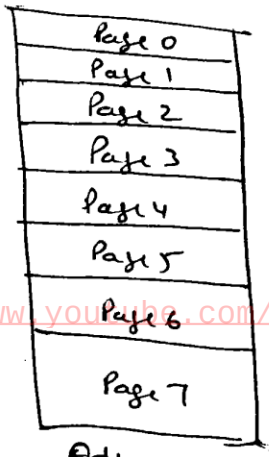
Page : Group of Address space of same size

Ex: If a page or block consist of 1K words then Address space is divided into 1024 pages & main memory is divided into 32 Blocks.

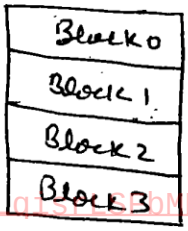
* Programs are also splitted into pages.
* Portions of programs are moved from auxiliary memory to main memory in records equal to the size of page.
* Page frame : Sometimes used to denote block.

Ex: Consider Address Space = 8K &
Memory Space = 4K.

* & we split each into group of 1K = 2^{10} words then -



Address Space: 2^{13}
N = 8K = 2^{13}

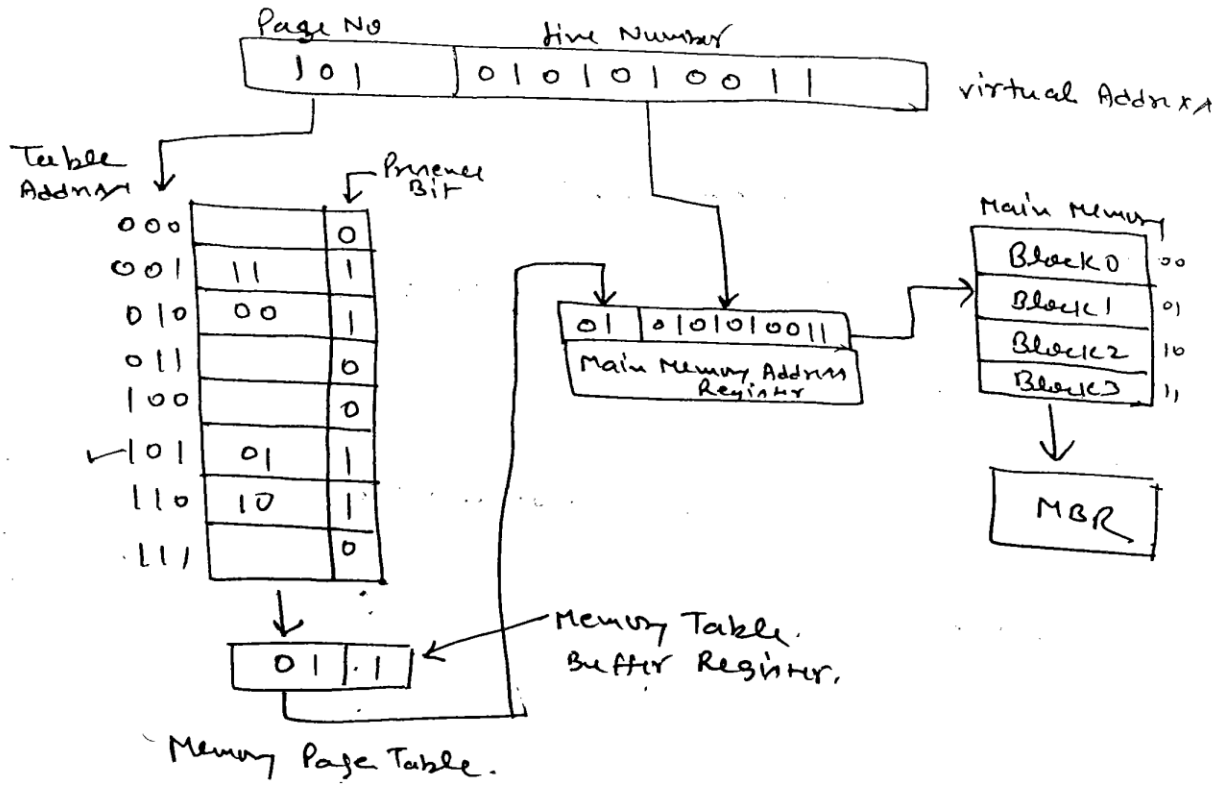


Memory Space
M = 4K = 2^{12}

* Virtual Address is considered to be represented by two numbers -
(1) Page Number Address
(2) A line with in the page

* In a computer with 2^p words per page,
p: Used to specify line Address.
- Remaining higher order bits are for page Number.

for above example -
10 bits are for specify line Address &
3 bits are for page No.



Memory Table in a paged System

* Memory page table :- for given example -
- It consist of 8 words (one for each page).
- Address in Page table, denotes - page Number. Content of word gives block number, where page is stored in main memory.
- Presence Bit in each location indicates whether page has been transferred to main memory. 0 means page is not available in main memory.

* Content of word in memory page table at page number address is read out into memory table buffer register.

* If presence bit is 1, A read signal to main memory transfers the content of word to main memory buffer register, ready to be used by CPU.

* If presence bit in the word is 0. It signifies that the content of the word referenced by virtual address does not reside in main memory. A call to the O.S. is then generated to fetch the required page from auxiliary memory & place it into main memory before resuming computation.

PAGE REPLACEMENT

* Virtual memory is combination of H/w & S/w techniques.

* Memory mgmt S/w system handles all S/w operations for efficient utilization of memory space - It must decide -
  (1) which page in main memory is to be removed.
  (2) when a new page is transferred from Auxiliary memory to main memory,
  (3) where the page is to be placed in main memory.

Page fault: Program is executed from main memory, until it attempts to reference a page that is still in auxiliary mem. This condition is called page fault.

* When page fault occurs, execution of present program is suspended until, required page is brought into main memory.

* Loading a page from auxiliary memory to main memory is basically an I/O operation, O.S. assign this task to I/O processor. In between control is transferred to next program in memory that is waiting to be processed in CPU.

* Page fault means page referenced by CPU is not in main memory.

* If main memory is full, necessary to remove a page from memory block.

Two algorithms for replacement —

(1) FIFO :- Selects for replacement the page that has been in memory for longest time.

(2) LRU :- Least recently used :— Remove the page, least likely to be referenced in immediate future.

* when page is loaded into memory its identification no. is pushed in to FIFO queue.

* It can be implemented by associating a counter with every page that is in main memory. When page is referenced counter is initialized with 0. After every fixed interval of time counter is incremented by 1. least recently used page is that which has highest count. (Counters are called aging registers)
This is more difficult to be implemented.

Q-1 A virtual memory system has an address space of 8k words, a memory space of 4K words, and page and block sizes of 1 K words. Following page reference changes occur during a given time interval —

4, 2, 0, 1, 2, 6, 1, 4, 0, 1, 0, 2, 3, 5, 7. Determine reference strings of length 4 pages that are resident in main memory after each page reference changes. If replacement algo used (1) FIFO (2) LRU