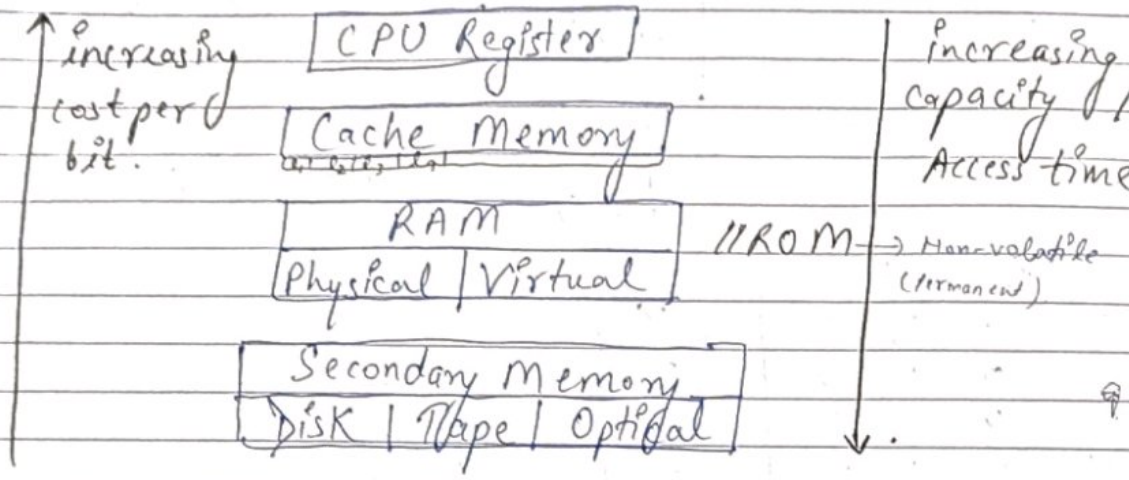
Date - 19/08/19

UNIT-4

MEMORY ORGANIZATION -

MEMORY HIERARCHY -



* CPU Register is the smallest & fastest memory.
* Cache memory have very less space.
* RAM (Primary Memory).
  → Primary Memory
  → Voldtile
  → Semiconductor.

Ram are divided into two types static & Dynamic.
static RAM are made from use of flip-flop.

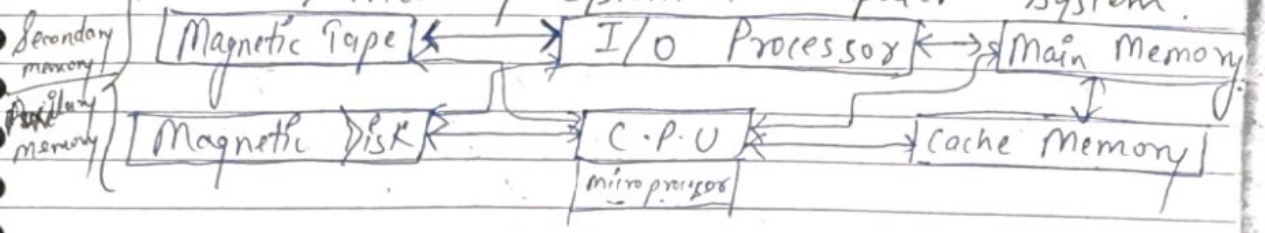
BIOS - Basic Input. Output System Programme
BIOS is also known as boot step loader.

Scanned by CamScanner

Peripheral devices → I/O devices.

ROM ← PROM, EPROM, EEPROM → Electronic erasable programable.

→ Memory System in Computer System.



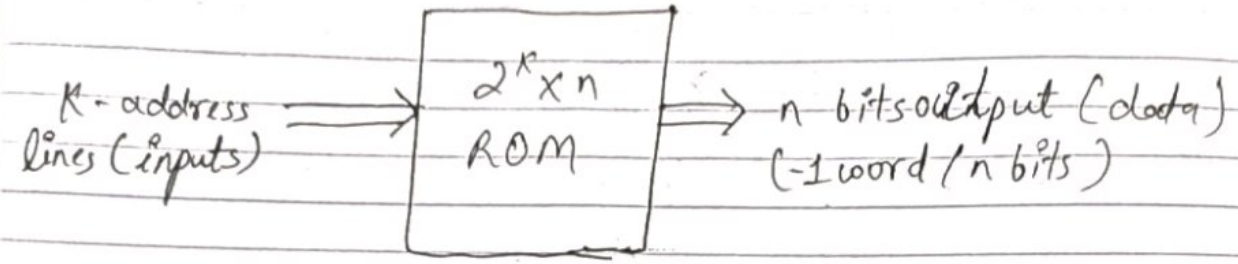
READ ONLY MEMORY (ROM) -

A ROM is essentially a memory device in which permanent binary information stores. The binary information must be specified by the designer & in then embeded in the unit to form the required interconnection pattern. ROM come with special internal electronic fuses that can be programmed for a specific configuration. Once the pattern is established, it stays with the unit even when power is turned off & on again.

A ROM is a device that includes both the decoder & the OR gates within a single IC package. The connection b/w the output of the decoder & the inputs of the OR gates can be specified for each particular configuration.

#NOTE - Generally, a $2^K \times n$ ROM consisting a $K \times 2^K$ decoder & n OR Gates. It contains K address lines (inputs). It contains 2^K words of n bits each.

Scanned by CamScanner

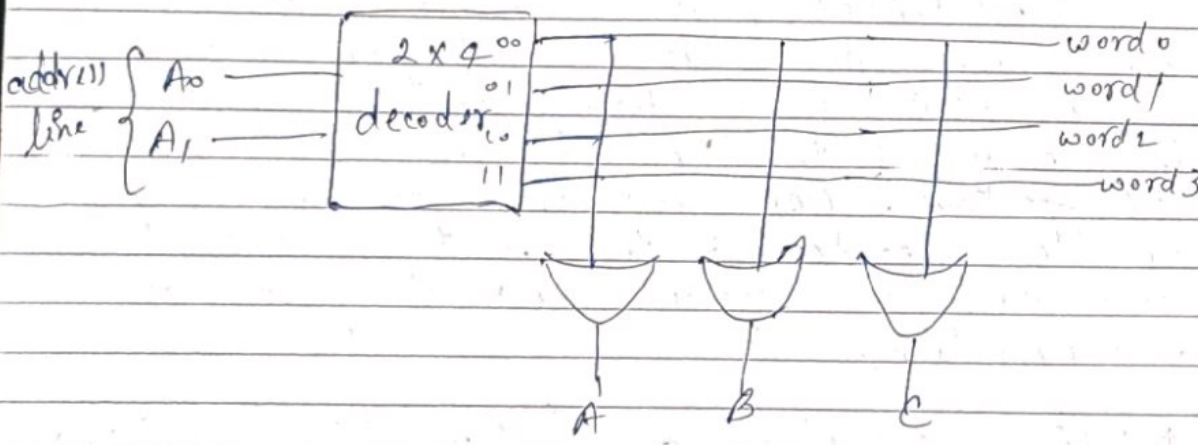


BLOCK DIAGRAM OF ROM

No. of words = 2^K
Capacity of ROM = $2^K \times n$ bits
word size = n bits
address lines = K

Q 4 x 3 ROM ⇒ $2^2 \times 3$.

2x4 decoder, address lines = 2 for gates = 3



Scanned by CamScanner

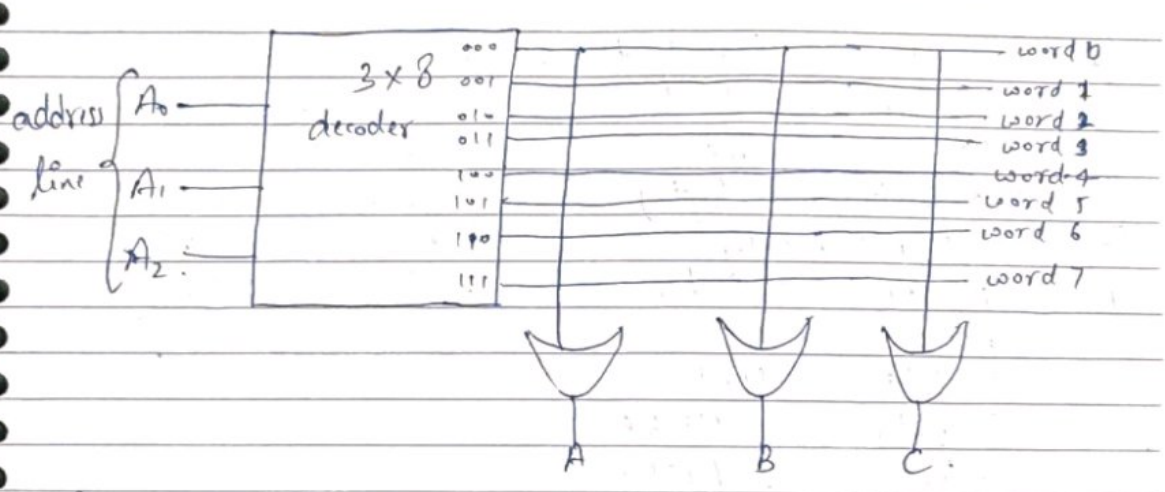
$2^k \times n = 2^b \times 3$.

$2^4 \times 3$.

Ex = 8×3 ROM & 16×3 ROM & 32×4 ROM.

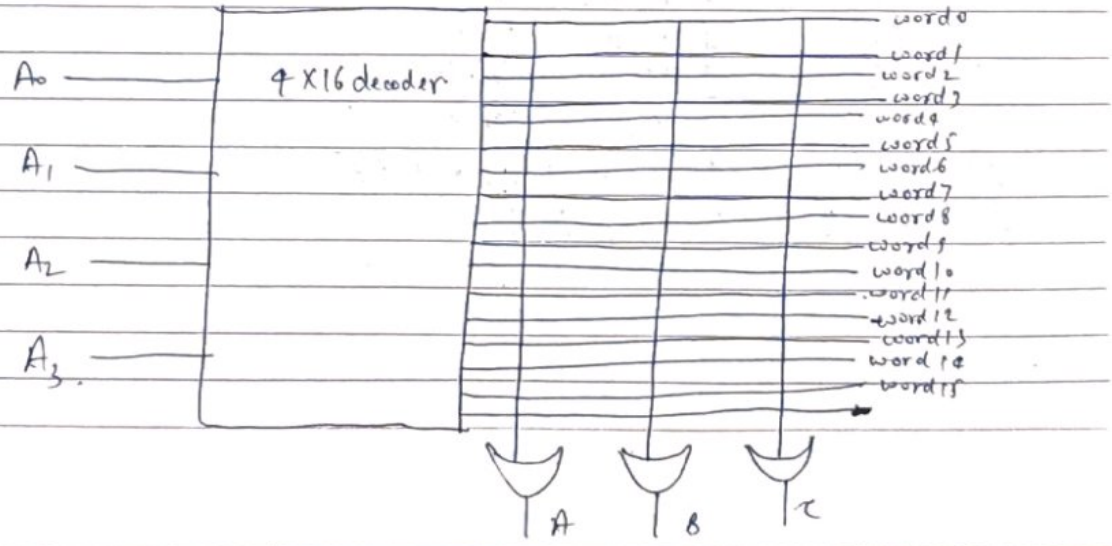
For 8×3 ROM -

3×8 decoder, 3 OR Gates, 3 address lines, word size 3 bits, no. of word = 8



For 16×3 ROM -

4×16 decoder, 3 OR Gates, 4 address lines, word size 3 bits, no. of words = 16.



Scanned by CamScanner

for 32×4 ROM.

5×32 decoder, 4 OR Gates, 5 address lines, word size 4, no. of words = 32.

Ex- 32×8 ROM. $2^5 \times 8$. | Here K=5 & n=8.

a	Word size (output size)	8 - n
b	no. of word	32 - 2^K
c	address in lines	5 - K
d	capacity (in byte)	32×8/8 = 32 byte.
e	size of decoder	5×32 - K×2^K
f	no. of OR Gates	8 - n.

1 bit	0, 1	
4 bits	1 Nibble	
8 bits	1 Byte	
1024 (2^{10}) byte	1 KB	
2^{10} KB	1 MB	
2^{10} MB	1 GB	
2^{10} GB	1 TB	Tera
2^{10} TB	1 PB	Peta
2^{10} PB	1 EB	Exa
2^{10} EB	1 ZB	Zeta
2^{10} ZB	1 YB	Yotta
2^{10} YB	1 Bronto Byte (BB)	
2^{10} BB	1 Geop Byte	

Scanned by CamScanner

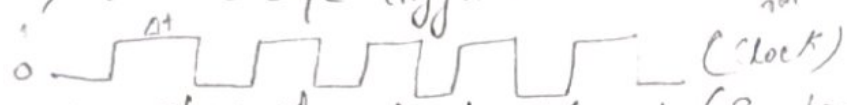
TYPES OF ROMs.

For small quantities it is more economical to use a second type of ROM called a Programmable Read Only Memory (PROM). When ordered, PROM units contain all 0's (or 1's) in every bit of the stored words. The fuses in the PROM are blown by application of current pulses through the output terminals. A blown fuse defines one binary state & an unblown link represents the other states.

The Hardware procedure for programming ROMs or PROMs is not changeable & one's program, the fixed pattern is permanent & cannot be altered. A third type of ROM available is called Erasable PROM (EPROM), can be restructured to the initial value (all 0's or 1's). When an EPROM is placed under a special ultraviolet light for a given period of time, the shortwave radiation discharges the internal gates. After discharging the ROM returns to its initial state & can be reprogrammed. Certain ROMs can be erased with electrical signals instead of ultraviolet light, & these are called electrically erasable PROM (EEPROM). These ROMs can be used to erase data from the desired position.

Scanned by CamScanner

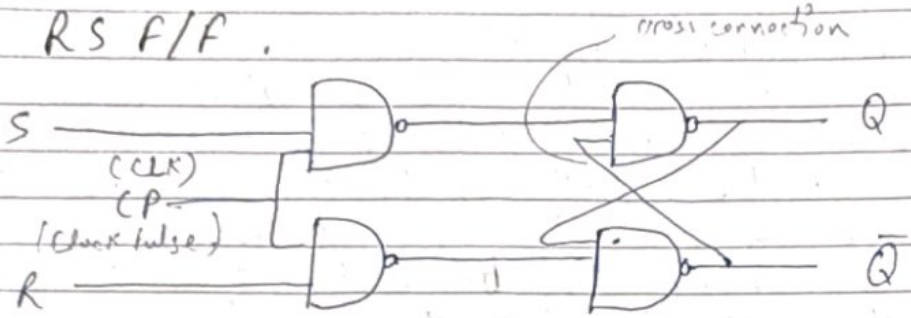
→ +ve/-ve edge triggered.



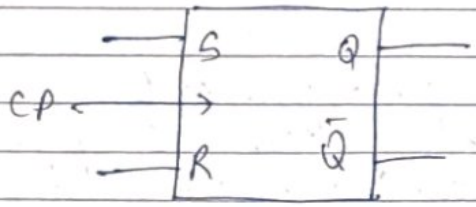
→ +ve/-ve ~~edge~~ level triggered. (Synchronous)

FLIP-FLOP

⇒ RS F/F.



LOGIC DIAGRAM



FLIP FLOP BLOCK DIAGRAM

Q_t (previous state output)	S (set)	R (Reset)	Q_{t+1} (next/present state)
0	0	0	0 (No change state)
1	0	0	1 (No change state)
0	0	1	0 (Reset state)
1	0	1	0 (Reset state)
0	1	0	1 (Set state)
1	1	0	1 (Set state)
0	1	1	forbidden State
1	1	1	forbidden State

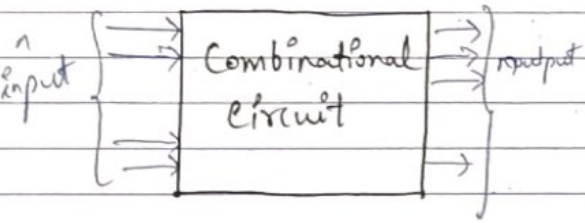
Characteristic Table

Scanned by CamScanner

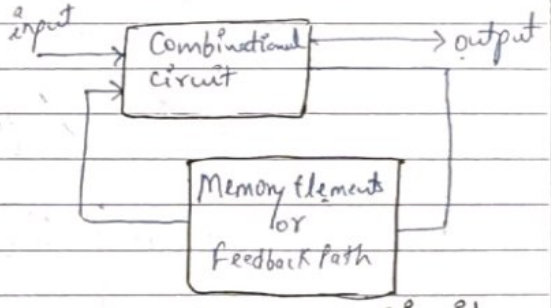
COMBINATIONAL & SEQUENCIAL CIRCUIT

Logic circuits for digital systems may be combinational or sequencial. A combinational circuit consist of logic gates whose output at any time are determine directly from the present combination of inputs without regard to previous inputs / outputs. A combinational circuit performs a specific information, processing operation fully specified logically by a set of boolean function.

Sequencial circuits employ memory elements (binary cells) or feedback path in addition to logic gates. Therefore, there outputs are a function of the inputs & the state of the memory elements (previous output). The state of memory elements in turn, is a function of ~~function~~ previous inputs.



Combinational Circuit



Sequential Circuit.

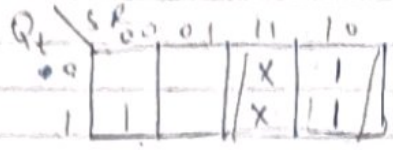
There are two type of Sequential circuits Synchronous & Asynchronous.

The behaviour of a Asynchronous circuit depends upon the on the order in which it's input

Scanned by CamScanner

signal change & can be affected by at any instant of time.

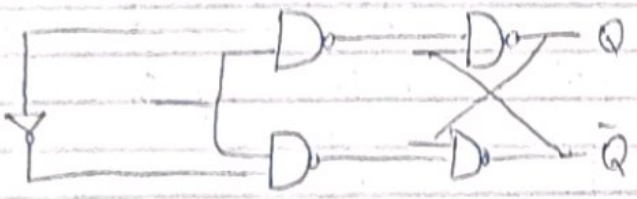
In Synchronous sequential circuit additional timing signals are used to control the output according to a specific time interval.



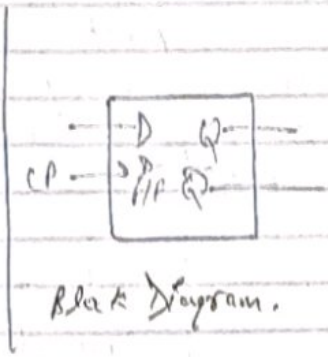
$Q_{t+1} = S + \bar{R} Q_t$ → Characteristic Equation.

$SR = 0$ (both S & R ≠ 1)

FF D-F/F.



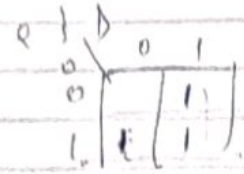
Logic Diagram



Block Diagram.

Q_t	D	Q_{t+1}
0	0	0 Reset
0	1	1 Set
1	0	0 Reset
1	1	1 Set

Characteristics Table



$Q_{t+1} = D$

↓

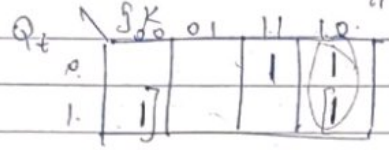
Characteristic Equation.

Scanned by CamScanner

J-K F/F

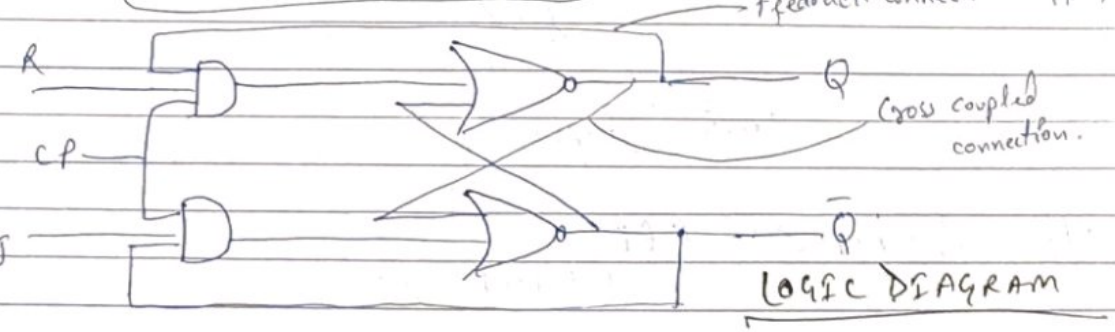
Characteristics Table :-

Q_t	J	K	Q_{t+1}
0	0	0	0 (No change)
0	0	1	0 (Reset)
0	1	0	1 (Set)
0	1	1	1 (Toggle/Complement)
1	0	0	1 (No change)
1	0	1	0 (Reset)
1	1	0	1 (Set)
1	1	1	0 (Toggle/Complement)

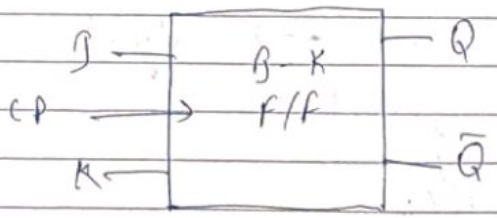


characteristics equation.

$$Q_{t+1} = Q_t K + \bar{Q}_t J + J\bar{K}.$$



LOGIC DIAGRAM



BLOCK DIAGRAM

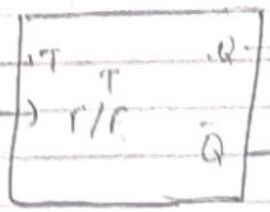
Scanned by CamScanner

T- F/F.

Characteristics Table

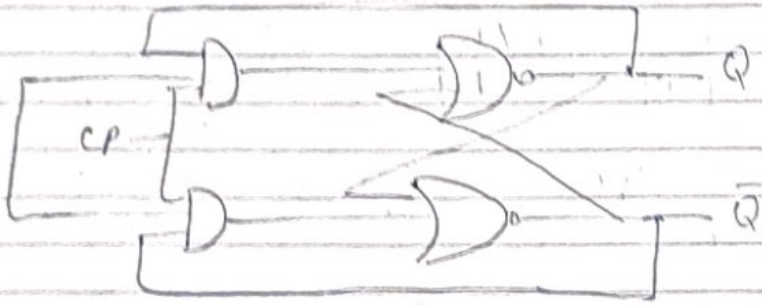
Q_t	T	Q_{t+1}
0	0	0 (No change)
0	1	1 (Toggle)
1	0	1 (No change)
1	1	0 (Toggle)

Block Diagram.



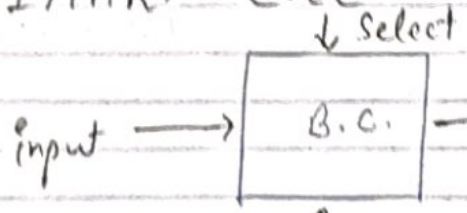
Q_t \ T	0	1
0	0	1
1	1	0

$Q_{t+1} = Q_t \bar{T} + \bar{Q}_t T$ → characteristic equation.



Logic Diagram

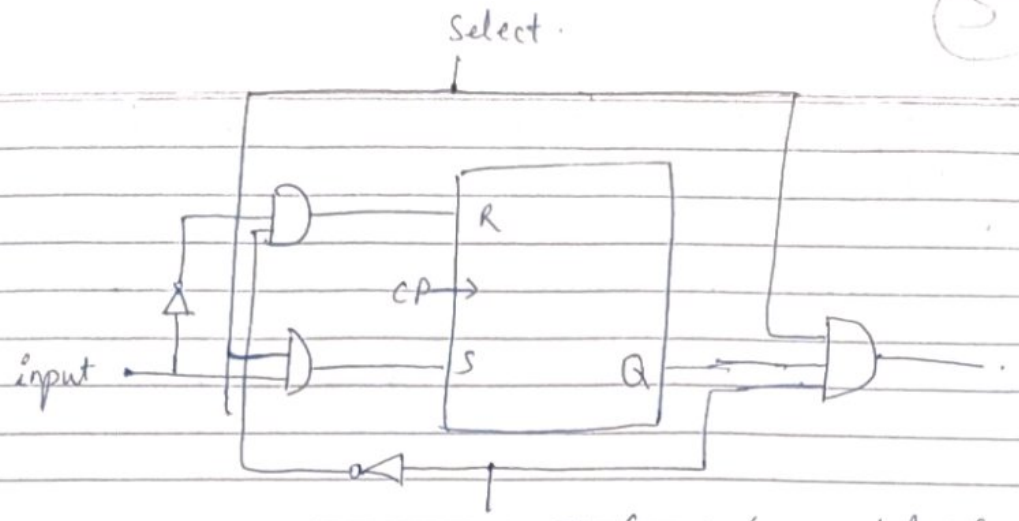
BINARY CELL -



R/$\bar{W}$
(Read/write control input)

BLOCK LOGIC DIAGRAM

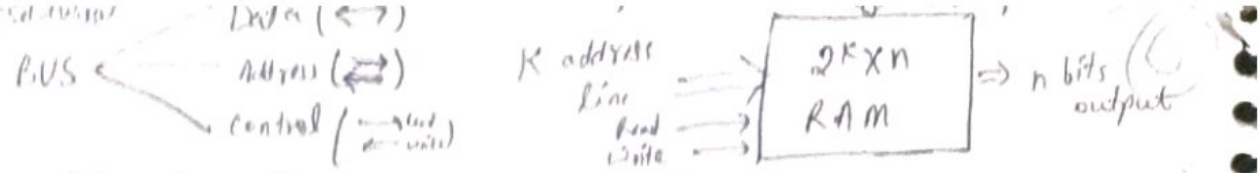
Scanned by CamScanner



LOGIC DIAGRAM

The Binary Storage cell is the basic building block of a memory unit. The binary cell stores one bit in its internal flip-flop. It has three inputs & one output. The select input enables the cell for reading or writing & the read/write input determines the cell operation when it is selected. A 1's in the read/write input provides the read operation by forming a path from the flip-flop to the output terminal. A 0's in the read/write input provides the write operation by forming a path from the input terminal to the flip-flop.

Scanned by CamScanner



LOGICAL CONSTRUCTION OF 4 x 3 RAM:

(4 rows, 3 col.)

4 x 3 = 2^2 x 3 = 2^K x n.

2^K → no. of words

n → word size (input/output size).

$2^K \times n$ → capacity (bits).

no. of binary cell = 12.

address line = 2

no. of words = 4

word size = 3.

capacity = 12 bits

2-> Memory Organization (Memory Decoding -

no. of AND Gate = 4.

no. of input/gate = 2.

for, 1024 x 1.

decoder size = 10×2^{10}

no. of AND Gate = 2^{10}.

no. of input/Gate = 10.

Scanned by CamScanner

7

Scanned by CamScanner

e.g. # MEMORY EXPANSION -

Two ways to increase memory -

* We increase no. of words
* We increase word size.

e.g. Q 1K × 8 RAM → 1024 × 8 ⇒ $2^{10} \times 8$.

Using 1K × 8 RAM make 4K × 8 RAM & 1K × 16 RAM

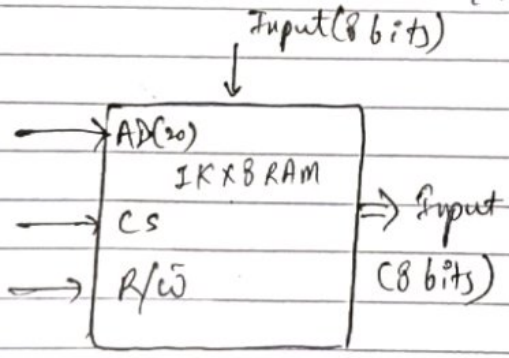
⇒ Given we have 1K × 8 RAM & using this RAM chip construct (i) 4K × 8 RAM & (ii) 1K × 16 RAM.

In 1K × 8 RAM chip.

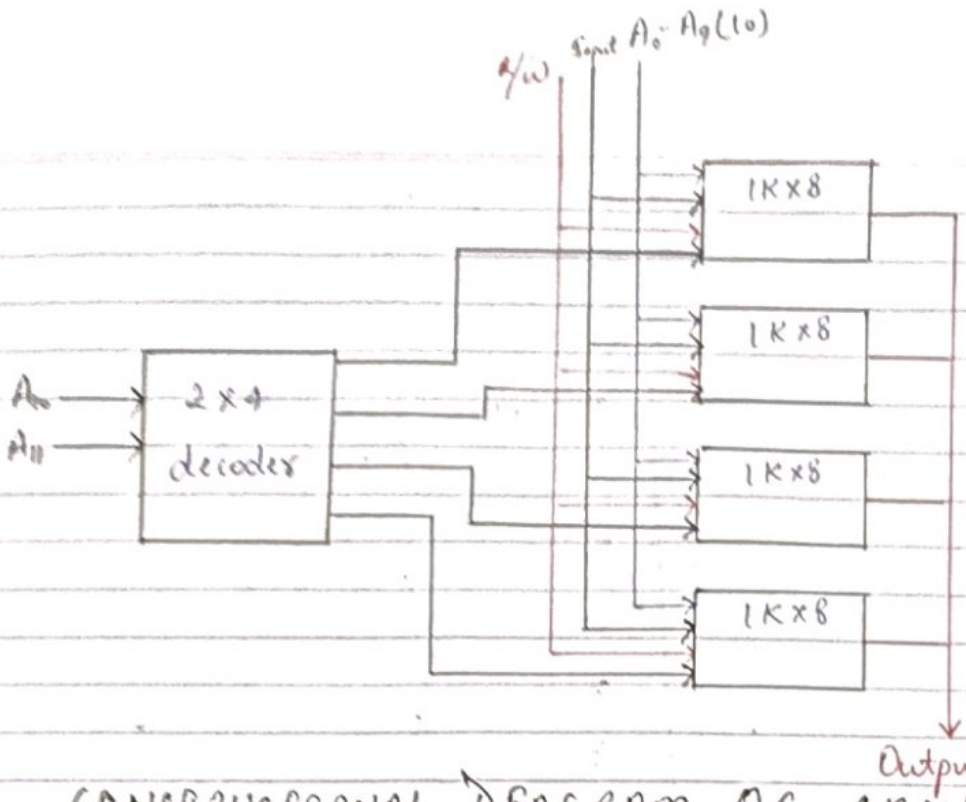
⇒ no. of words = 1K = 1024
⇒ address line = 10.
⇒ word size (I/O size) = 8 bit.
⇒ Capacity = 1K × 8.

In 4K × 8 RAM →

⇒ no. of RAM chips = $\frac{4K \times 8}{1K \times 8} = \frac{2^{12}}{2^{10}} = 4$.



Scanned by CamScanner

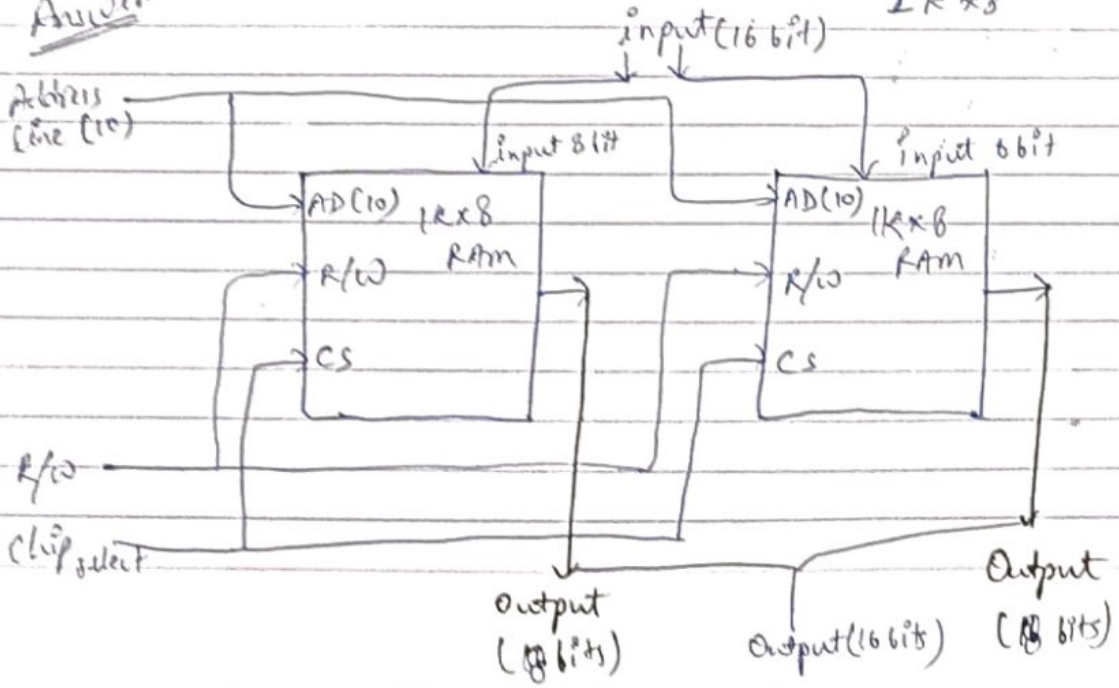


CONSTRUCTIONAL DIAGRAM OF 4K×8 RAM.

Q In 1K×16 RAM →

⇒ no. of RAM chips of 1K×8 = $\frac{1K\times16}{1K\times8}$ = 2

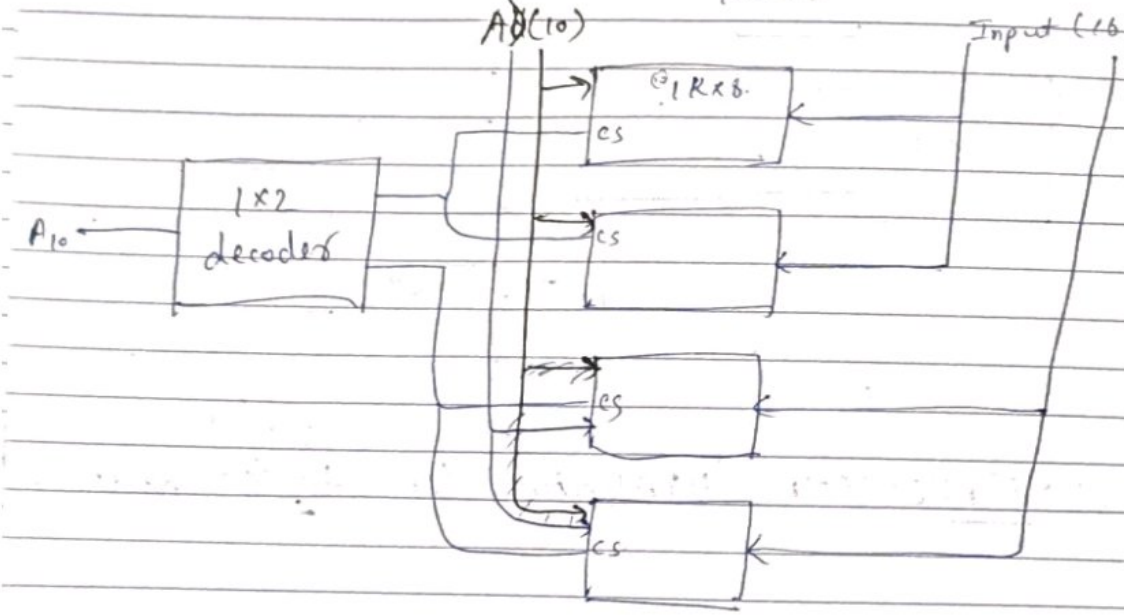
Answer-



Scanned by CamScanner

(iii) 2K x16 RAM. Design using of 1K x8.

no. of 1K x8 chips = $\frac{2K \times 16}{1K \times 8}$ = 4.



Scanned by CamScanner

Q EXERCISE - 1 (A)

(A) How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 byte.

(B) How many lines of the address must be used to access 2048 bytes? How many these lines are connected to the address inputs of all chip.

(C) How many lines must be decoded for the chip select input? Specify the size of the decoder.

Q EXERCISE - 2.

(a) A Computer uses RAM chips of 1024 x 1 capacity. (i) How many chips are needed & how should their address lines be connected to provide a memory capacity of 1024 bytes.

(b) How many chips are needed to provide a memory capacity of 16K (16 x 1024) bytes? Explain in words how the chips are to be connected.

⇒ Answer exercise - 1 (A) $\frac{2048}{128 \times 8} = \frac{2^{11} \times 2^3}{2^7} = 16$

(B) 11 lines, 7 lines

(C) 4 lines, 4 x 16.

Exercise - 2 (B) (a) $\frac{1024 \times 8}{1024 \times 1} = 8$

(b) $\frac{2^{14} \times 2^3}{2^{10}} = 128$ chips

Scanned by CamScanner

Q Exercise 3 Cirmit

An Integrated RAM chip has a cap of 1024 words of 8 bit each.

(i) How many address & data lines are the chip?

(ii) How many chips are needed to cons a 16K x16 RAM?

(iii) How many address in data lines the 16 K x16 RAM?

(iv) What size decoder is needed construct the 16 K x 16 RAM from 1K x8 RAM chips? What are the to the decoder & where are it's ou connected?

Answ (i) 10 , 8

(ii) 32

(iii) 14 address lines

(iv) 4 x 16 ,

2½ or 2.5 Memory Organiz (Coincident decoding).

e.g. 1024 x 1 RAM

decoder size = 10 x 1024

AND Gate = 1024

Inputs / Gate = 10

Scanned by CamScanner

12

Q Exercise 3 Circuit

An Integrated RAM chip has a capacity of 1024 words of 8 bit each. (input/output lines. (n))

(i) How many address & data lines are in the chip?

(ii) How many chips are needed to construct a 16K x16 RAM?

(iii) How many address in data lines in the 16K x16 RAM?

(iv) What size decoder is needed to construct the 16K x16 RAM from the 1K x8 RAM chips? What are the inputs to the decoder & where are it's output connected?

Ans (i) 10, 8

(ii) 32

(iii) 14 address lines

(iv) 4 x 16,

2½ or 2.5 Memory Organization. (Coincident decoding).

e.g. 1024 x1 RAM

decoder size = 10 x 1024

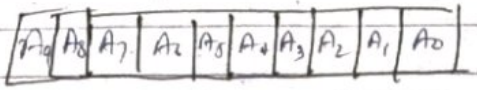
AND Gate = 1024

Inputs / Gate = 10

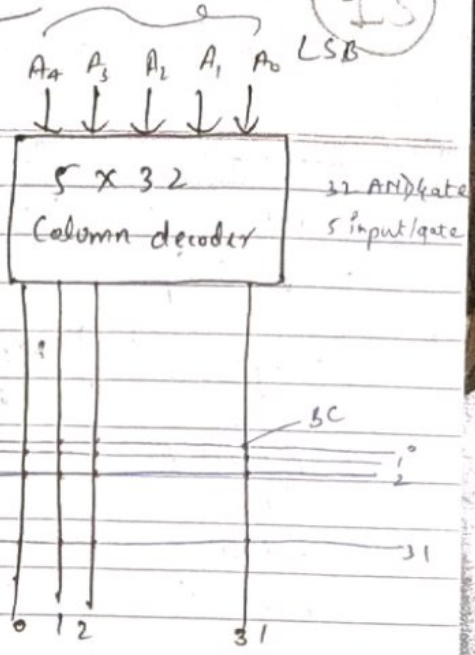
Scanned by CamScanner

Register is a collection of flip-flop

MAR – Memory Address Register



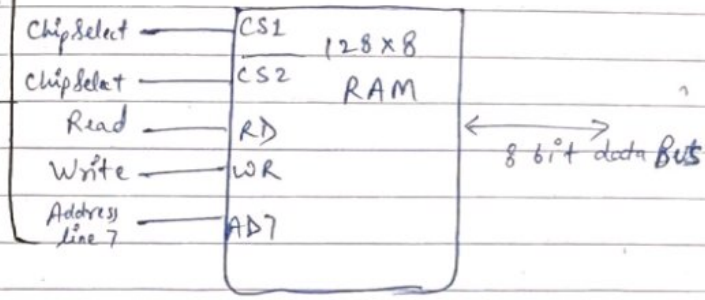
Memory Address.

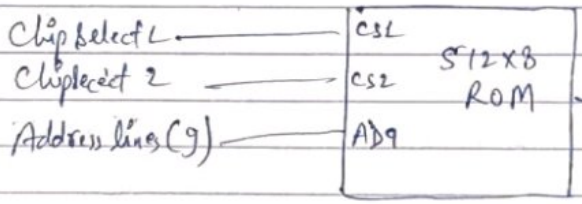


V. Imp

MEMORY ADDRESS MAP.

CS1	CS2	RD	WR	Memory function	State of Data Bus
1	0	0	1	write	Input data to RAM
1	0	1	x	read	output data from RAM





$CS1 = 1, \overline{CS2} = 0$ (Read Operation)

Scanned by CamScanner

no. of RAM chips required = 4 = 512×8 / 128×8.
no. of ROM chips require = 1 = 512×8 / 512×8. (14)
Address line or Address Bus both are same.

Q Construct a Computer System with 512 bytes of RAM & 512 bytes of ROM using the RAM chips 128×8 & ROM chip 512×8.

Answer

Component	Hexadecimal Range	Address Bus. 10 9 8 7 6 5 4 3 2 1 (2×4 decoder)
RAM 1	0000H - 007FH	0 0 0 x x x x x x x
RAM 2	0080H - 00FFH	0 0 1 x x x x x x x
RAM 3	0100H - 017FH	0 1 0 x x x x x x x
RAM 4	180H - 01FFH	0 1 1 x x x x x x x
ROM 1	0000H - 01FFH	1 x x x x x x x x x

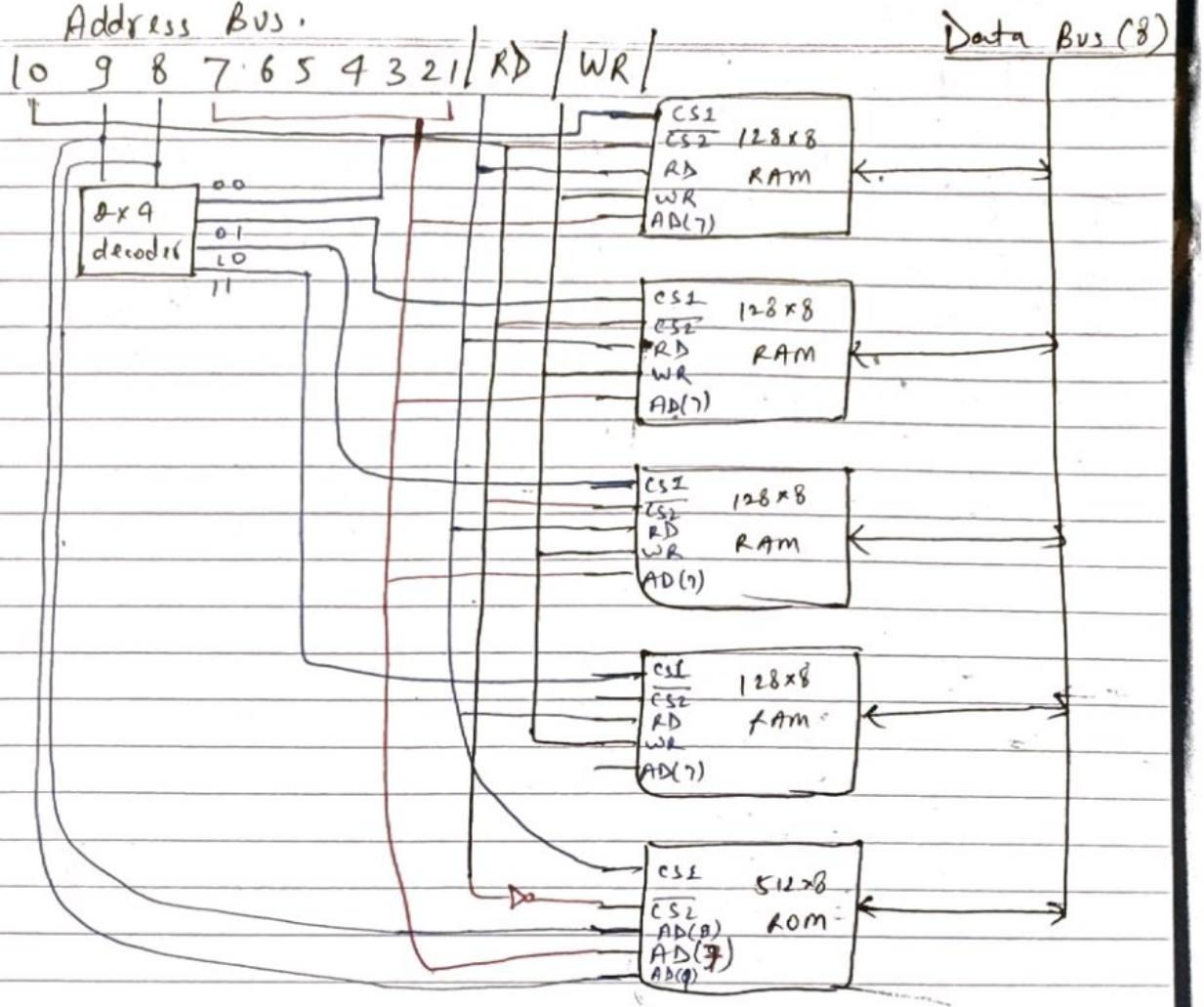
(0-127) → (0000H - 007F

Q

Scanned by CamScanner

15

(Input/Output(8)).

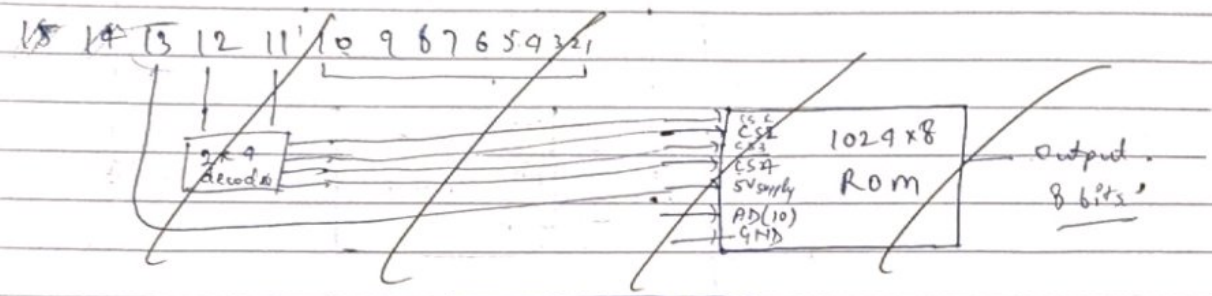


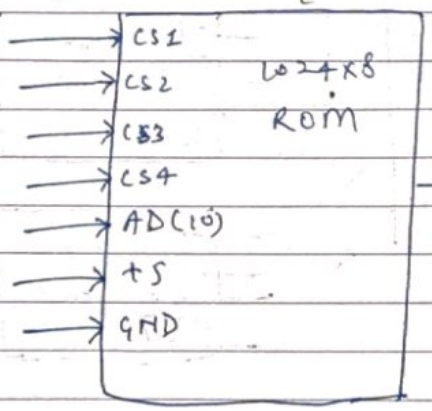
Scanned by CamScanner

2048×8 / 256×8 · 2^11 / 2^8 · 2^3 (8) · 10+2+1 · 1024×8 · 2^12 / 2^10 (16) · 2×4 · 2^10

Exercise -

Q A ROM chip of 1024 x 8 bits has four select input & operates from a 5V power supply. How many pins are needed for the IC package? Draw a block diagram & label all input & all outputs terminal in the ROM.





Total 24 Pins.

Exercise -

Q A computer Employees RAM chips of 256 x 8 & ROM chips of 1024 x 8. The computer system needs 2K bytes of RAM, 4K bytes of ROM & four interface units of four registers each. A memory - mapped I/O configuration is used. The two highest ordered bits of

Scanned by CamScanner

the address bus are assigned to 00 for RAM, 01 for ROM & 10 for interface unit (registers). –

(a) How many RAM & ROM chips are required?
(b) Draw a memory map for the system.
(c) Given the address range in the hexadecimal for the RAM, ROM & Interface.

Answer – (a) No. of RAM chips = $\frac{2^{11} \times 8}{2^8 \times 8} = 2^3 = 8$

No. of ROM chips = $\frac{2^{12} \times 8}{2^{10} \times 8} = 2^2 = 4$

(b)

Component	Hexadecimal Range	Address BUS (16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1)
RAM1	0000H – 00FFH	0 0 0 0 0 0 0 0 X X X X X X X X
RAM2	0100H – 01FFH	0 0 0 0 0 0 0 1 X X X X X X X X
RAM3	0200H – 02FFH	0 0 0 0 0 0 1 0 X X X X X X X X
RAM4	0300H – 03FFH	0 0 0 0 0 0 1 1 X X X X X X X X
RAM5	0400H – 04FFH	0 0 0 0 0 1 0 0 X X X X X X X X
RAM6	0500H – 05FFH	0 0 0 0 0 1 0 1 X X X X X X X X
RAM7	0600H – 06FFH	0 0 0 0 0 1 1 0 X X X X X X X X
RAM8	0700H – 07FFH	0 0 0 0 0 1 1 1 X X X X X X X X
ROM1	2000H – 23FFH	0 1 0 0 0 0 X X X X X X X X X X
ROM2	2400H – 27FFH	0 1 0 0 0 1 X X X X X X X X X X
ROM3	2800H – 2BFFH	0 1 0 0 1 0 X X X X X X X X X X
ROM4	2C00H – 2FFFH	0 1 0 0 1 1 X X X X X X X X X X
Interface	4000H – 400FH	1 0 0 0 0 0 0 0 0 0 0 0 X X X X

Total Register = 4 × 4 (16)

Scanned by CamScanner

2½ or 2.5 MEMORY ORGANIZATION (COINCIDENT DECODING) -

In 2D organization for designing $2^K \times n$ RAM, we require a decoder with K inputs & 2^K outputs i.e., 2^K AND Gates with K inputs per gate. We can reduce the total no. of gates & the inputs per gates by employing two decoders (a row decoder & a column decoder), which one decoder performs the row selection & other performs the column selection. Therefore, we only require two decoder of size $K/2 \times 2^{K/2}$.

For example - If K=10, in 2D organization we require 10×2^{10} size decoder with 1024 AND Gates of 10 inputs each.

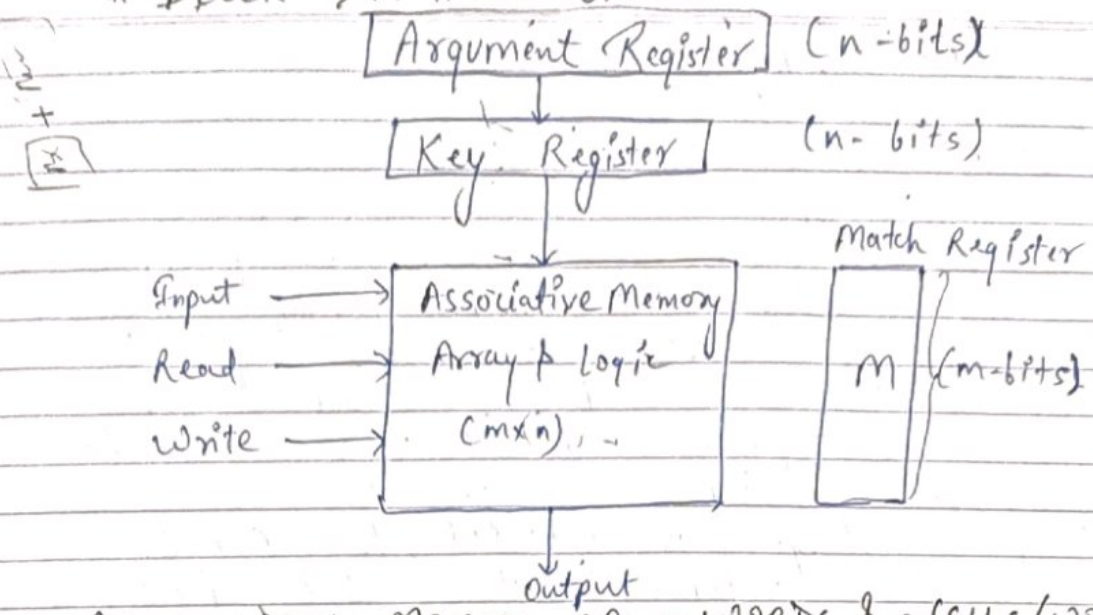
But in 2½ D organization we only require two decoders of size 5×32 with 32 AND Gates of total no. of AND Gates (32+32) 64 of 5 inputs each.

ASSOCIATIVE MEMORY - (CONTENT ADDRESSABLE MEMORY (CAM)) -

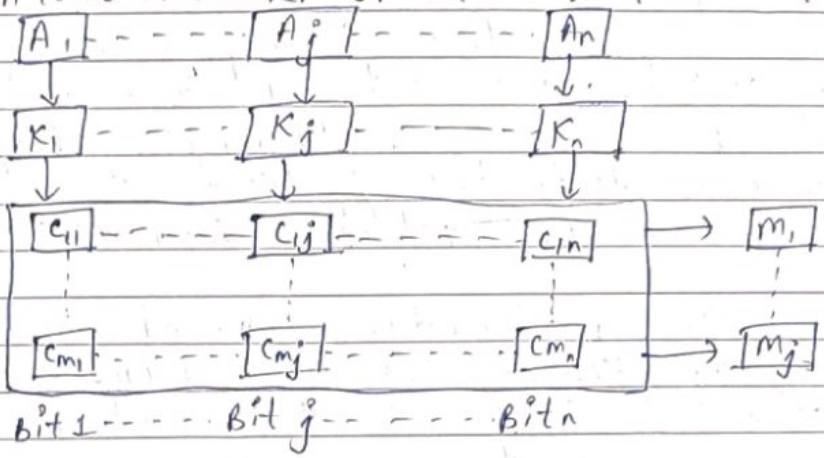
P.T.O.

Scanned by CamScanner

* BLOCK DIAGRAM OF CAM -



* ASSOCIATIVE MEMORY OF m WORDS & n CELLS/WORD



MASKING (AND Operation).

A = 1 1 0 1 0 1 0 1 | 1 1 0 0 0 1 0 0

AND K = 1 1 1 1 1 1 1 1 | 1 1 1 0 0 0 0 0

1 1 0 1 0 1 0 1 | 1 1 0 0 0 0 0 0

Scanned by CamScanner

CACHE MEMORY

Imp

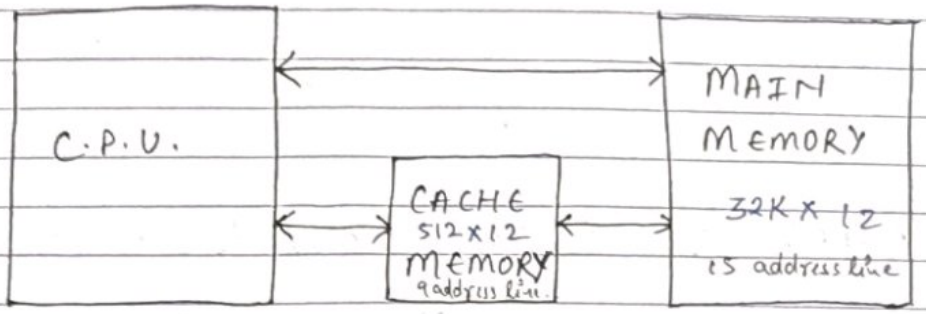
* LOCALITY OF REFERENCE -

Analysis, of a large no. of programmes as shown that the references to memory at any given interval of time tends to be confined within a few localized area in memory. This phenomenon is known as the property of locality of references. When a programe loop is executed, the C.P.U. repeadily refers to the set of instructions in memory that within the loop. Everytime a given sub routine (sub program or function) is called, it's set of instructions are called from memory. Therefore, loops & sub routines tends to localised the reference to memory for fetching instructions.

The fundamental idea of cache organization is that by keeping the most frequently accessed instructions & datta in the fast & small cache memory, the average memory access time will increase near to the C.P.U's access rate. If the active portion of the programe & data are placed in a fast small memory, the average memory access time can be reduce. Thus, reducing the total execution time of the programe. The cache memory is placed b/w the C.P.U. & a main memory.

Scanned by CamScanner

The basic operation of cache is as follows - When the C.P.U. needs to access the memory, the cache is examined. If the word is found in cache (cache hit), it is read from the cache memory. If the word addressed by the C.P.U. is not found in the cache (cache miss) the main memory is accessed to read the word.



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

$$\text{hit ratio} = \frac{\text{no. of cache hits}}{(\text{no. of cache hits} + \text{no. of cache miss})}$$

Range (0-1)

(no. of cache hits + no. of cache miss) → TOTAL REFRENCES

NOTE -
Cache Memory has two parts L_1 which is primary & L_2 which is secondary.
Fastest Order $\{ L_1 > L_2 \}$.

Scanned by CamScanner

Imp

ex- Q Calculate the average memory access time of a computer system with cache access time of 100ns, a main memory access time of 1000ns & a hit ratio of 0.9.

Solution-

Total average access time = (Hit ratio x cache access) + {Remaining Hit Ratio x (cache access + main memory access)}.

$= 0.9 \times 100 + 0.1 \times (100 + 1000)$

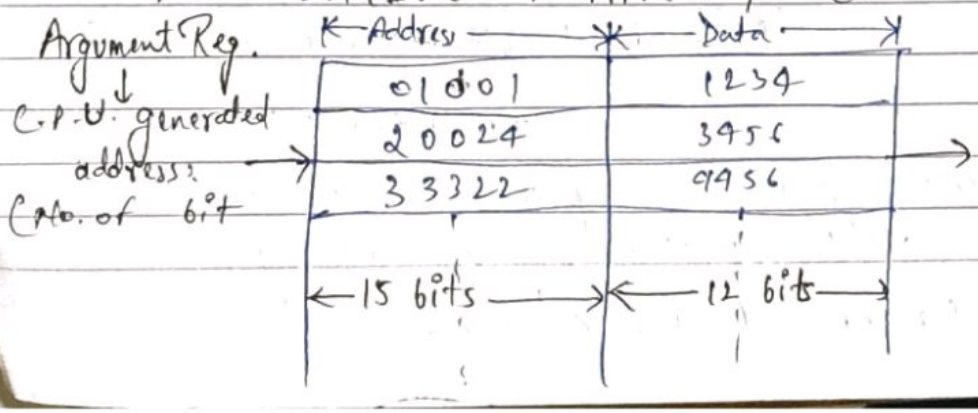
$= 200ns.$

TYPES OF CACHE ORGANIZATION-

The transformation of data from main memory to cache memory is referred to as mapping process. Three type of mapping procedure are used when considering the organization of cache memory.

* ASSOCIATIVE MAPPING
* DIRECT MAPPING
* SET-ASSOCIATIVE MAPPING

* ASSOCIATIVE MAPPING - (In octal → 3 bit).

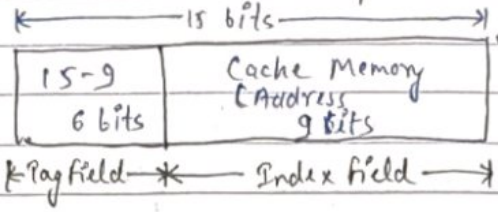


Scanned by CamScanner

DISADVANTAGES

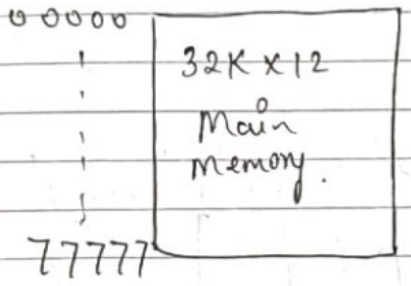
* Associative Memories are expensives compare to Random Access Memories (RAM) because of the added logic associated with each cell.

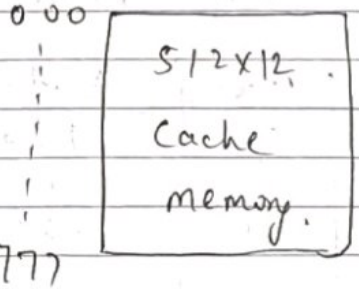
* DIRECT MAPPING

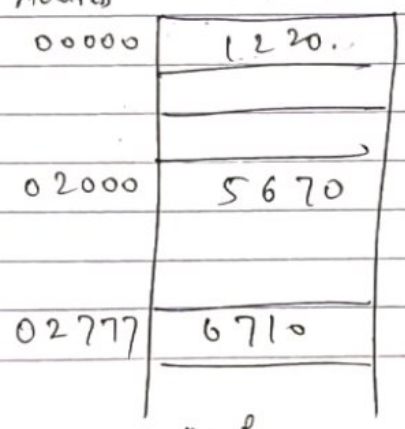


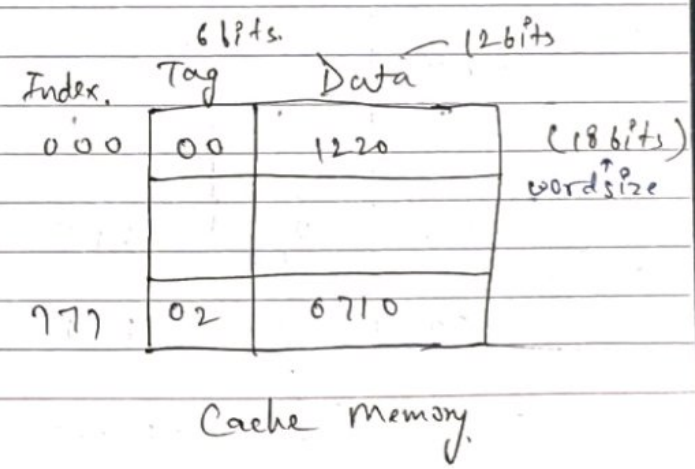
ex-

If n-bits Main Memory Address & K-bits Cache Memory Address.
Index field - K bits.
Tag field - (n-K) bits.









Scanned by CamScanner

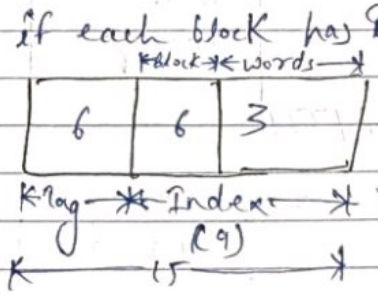
* DISADVANTAGES OF DIRECT MAPPING.

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

We can organize direct mapping cache organization with no. of blocks size. In this case the index field is divided into two parts - the block field part & the word field part. The tag field store within the cache is common to all words of the same block. Everytime a cache miss occurs, an entire block must be transferred from main memory to cache memory.

	Index	Tag	Data
Block 0	000	01	3456
	⋮	⋮	⋮
	007	01	2225
Block 1	010	---	----
	⋮		
	017	---	----
	⋮	⋮	⋮
Block 63	770	02	655
	⋮	⋮	⋮
	777	02	555

Octal cache memory



Scanned by CamScanner

Data → 16.

$2^7 5^-$

EXERCISE -

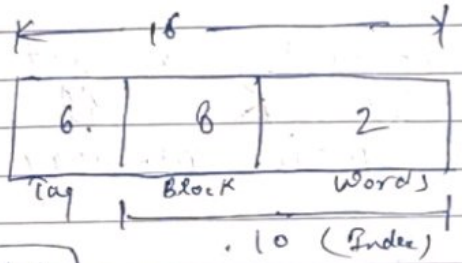
Q A Digital Computer has a memory unit of 64K x16. & a cache memory of 1K words. The cache uses direct mapping with a block size of four words.

(a) How many bits are their in the tag, index, block & word fields of the address format?

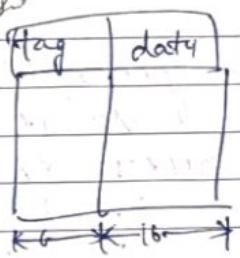
(b) How many bits are their in each word of cache, How are they divided into functions? Include a valid bit.

(c) How many blocks can the cache accomodate

Solutions - (a)



(b) ~~16~~



6 + 16 + include a valid bit

23.

(c) 256.

Scanned by CamScanner

(xxx)

(26)

SET ASSOCIATIVE MAPPING -

The third type of cache organization, called set-associative mapping. is an improvement over the direct mapping organization is that each word of cache can store two or more words of memory under the same index addresses.

Each data word is stored together with its tag & the no. of tag data items in one word of cache is said to form a **set**.

In General, a set associative cache of set size K will accomodate K words of main memory in each word of cache.

When the C.P.U. generates a memory reference, the index value of the address is used to access the cache. The tag field of the C.P.U. address is than compared with both tags (K=2). In the cache to determine if a match occurs. The comparison logic is done by an associative search of the tags in the set similar to an associative memory search. Therefore, the name set associative. The hit ratio will improve as the set size increases because more words with the

Scanned by CamScanner

same index but different tags can reside in cache.

ex Two way Associative Mapping Cache.

Index	Tag	Data	Tag	Data
000	01	3450	02	6710
⋮				
777	00	2310	03	5543

(K = 2)

one word of cache = two words of main memory

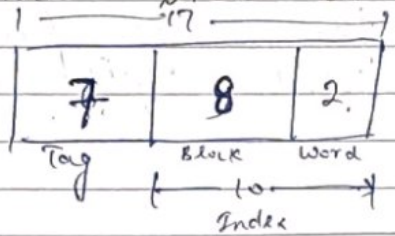
Exercise

Q A two way set associative cache memory uses blocks of 4 words. The cache can acomodate a total of 2048 words from main memory. The main memory size is 128K x 32.

(a) formulate all information required to construct the cache memory.

(b) What is the size of cache memory.

$\Rightarrow \frac{2048}{2} = 2^{10}$



1024 words/32 bit in cache memory

Data = 32
Tag = 6.

(b) 78 bits.

Size = 1024 x 78 bit

Scanned by CamScanner

WRITING INTO CACHE -

There are two approaches used for writing into cache.

* WRITE THROUGH APPROACH -

The simplest & most commonly used procedure is to update main memory with every memory write operation, with cache memory being updated in parallel. If it contains the word at the specified address. This is called the write through method. This method has the advantage that main memory always contains the same data as the cache.

* WRITE BACK APPROACH -

The second procedure is called the write back method. In this method only the cache location is updated during a write operation. The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory.

Scanned by CamScanner

Set → Block (line) → words.

VALID BIT -

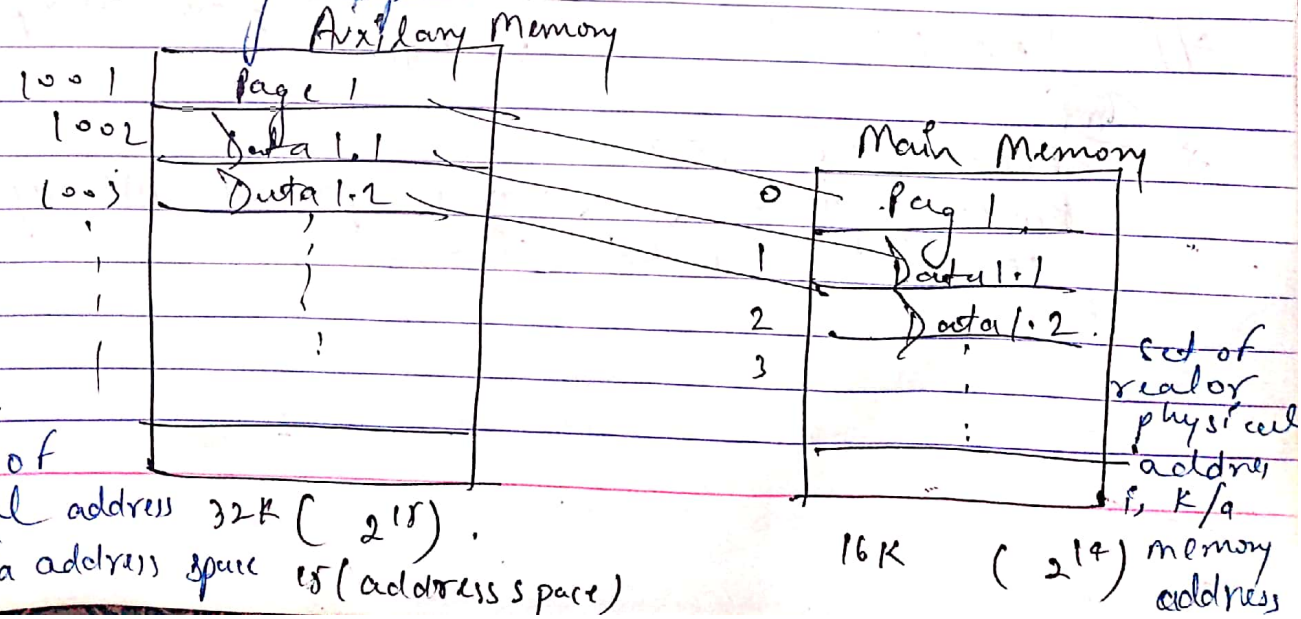
After initialization the cache is considered to be empty, but in effect it contains some non-valid data.. Therefore, a single bit include with each word stored in cache to indicate whether or not words contain valid data, this bit is known as valid bit.

The introduction of the valid bit means that a word in cache is not replaced by another word unless the valid bit is said to 1 (one). If the valid bit happens to be 0 (zero), the new word automatically replaces the invalid data.

(Imp.) 2nd Sessional start -

VIRTUAL MEMORY. (15 marks).

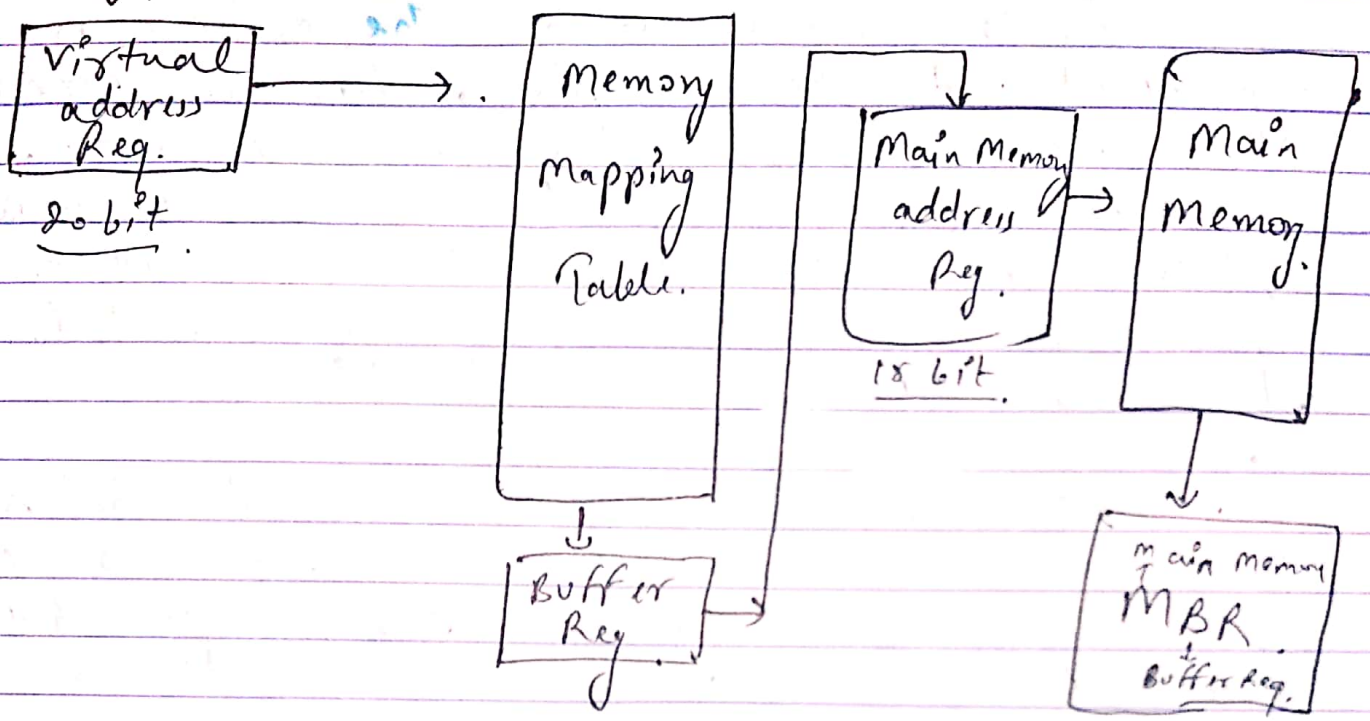
→ Dynamic address translation.
→ Address Space / (Virtual address space).
→ Memory Space.



Generally, the size of virtual memory is greater than size of main memory.

C.P.U

virtual address



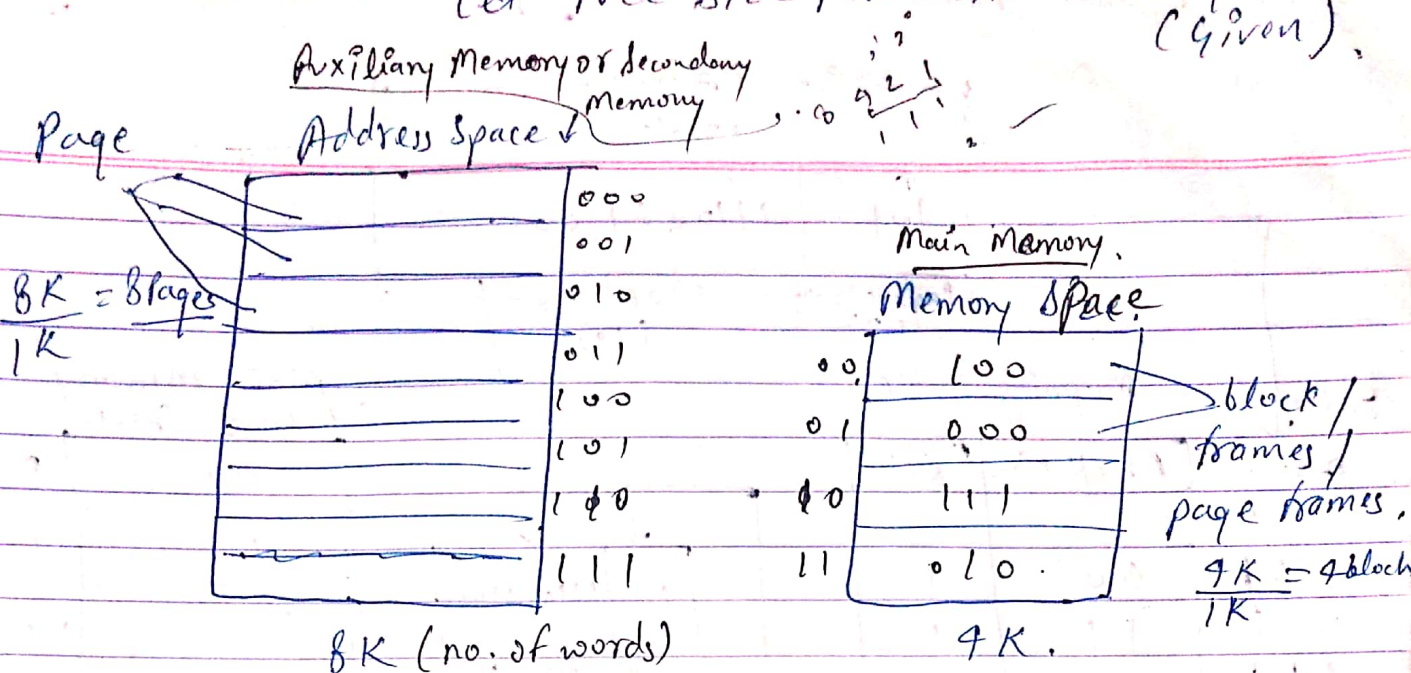
The above process is known as dynamic address translation.

The conversion of virtual address into main memory address with the help of mapping table automatically in dynamic time. is known as dynamic address translation (mapping).

PARTITION ← fixed / Variable (Address Mapping).

* (Imp) fixed or paging -
(Partition the address space & memory space into fixed size).

Let free size partition = 1K (Page size) (Given).

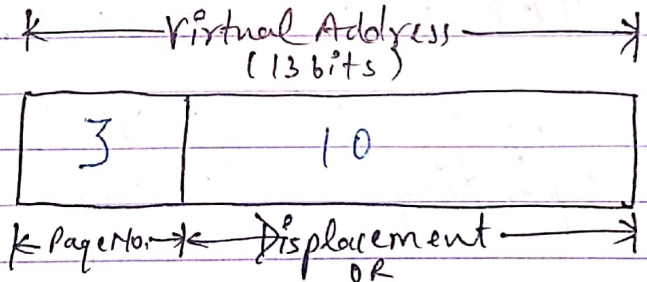


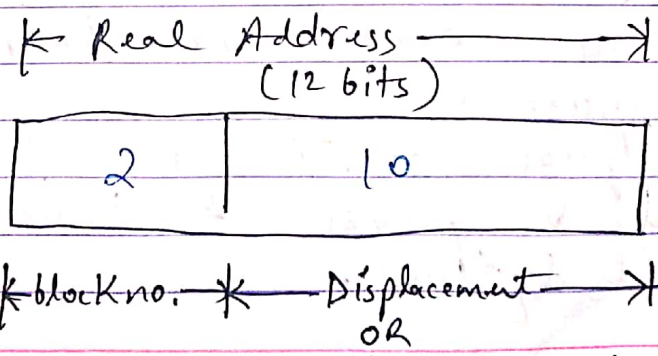
Virtual address = Address Space = 13 bits

Real or Physical address = Memory or Main Memory Space = 12 bits

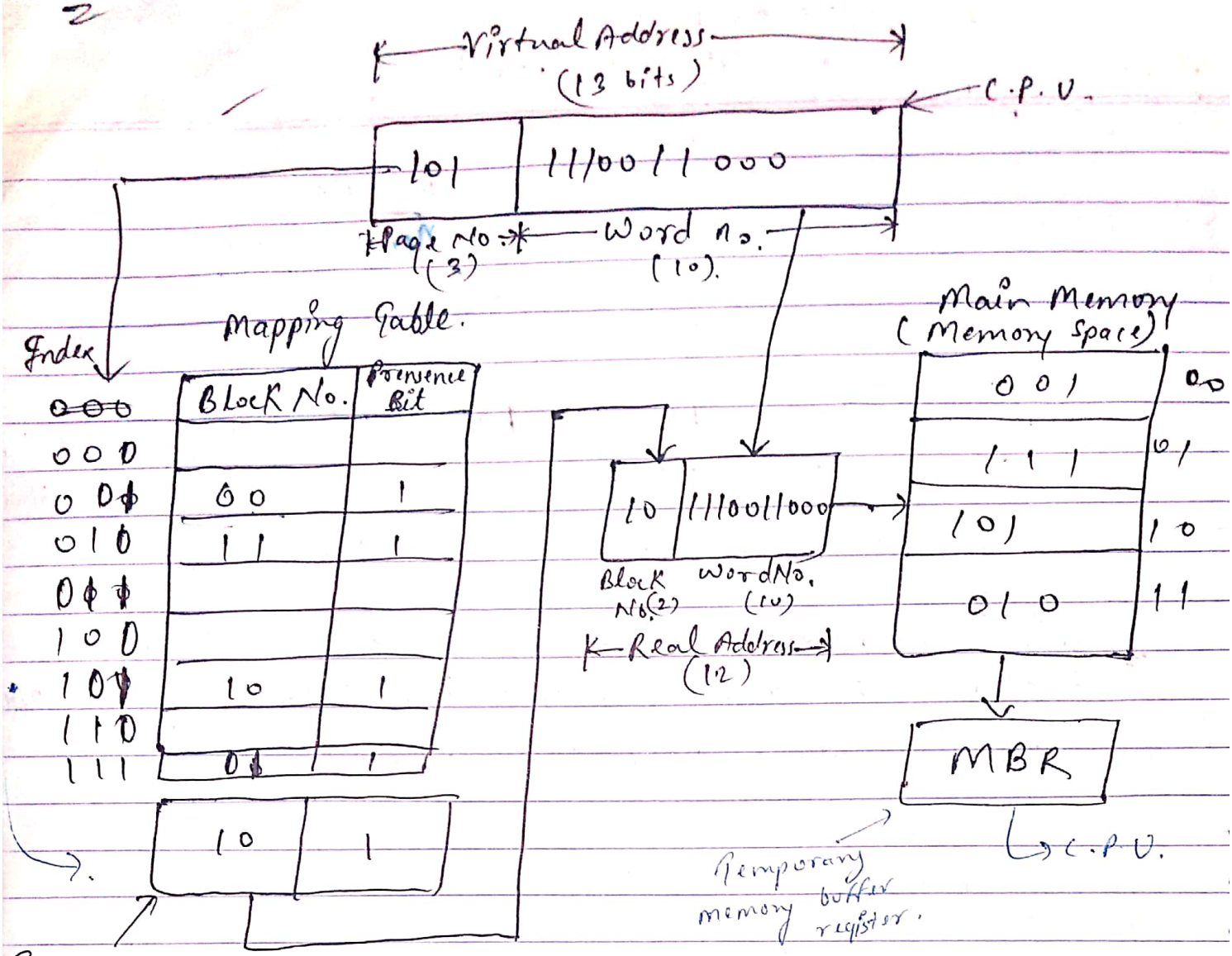
Programe / C.P.U always generate virtual addreis.

If C.P.U called page is not found in main memory then a signal is generate & this signal is known as Page fault.





Q How Virtual Address Map to Real Address?



MEMORY TABLE IN PAGING SYSTEM

Q An address space is specified by 24 bits & the corresponding memory space by 16 bits.

(a) How many words are there in the address space?

(b) How many words are there in the memory space?

(c) If a page consist of 2K words, how many pages & blocks are there in the system.

2^{24-11} (13)

Page

2^{12} 2^{10} $2^{(n)}$

2^1 K

Address Space bit

Answer - (a) memory space (16) $2^{24} = 2^4 \cdot 2^{10} \cdot 2^{10} = 2^4 M = 16M$.

(b) $2^{16} = 2^6 \cdot 2^{10} = 64K$.

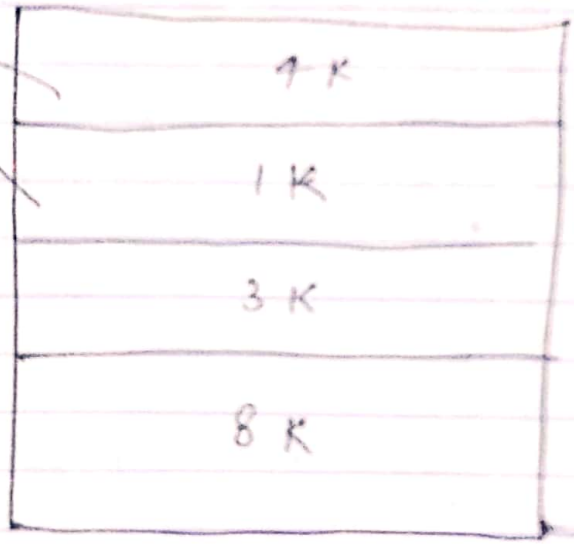
(c) $\frac{2^{21}}{2K} = \frac{2^{14} K}{2^1 K} = 2^{13}$. (No. of Pages)

$\frac{2^{16}}{2K} = \frac{2^6 K}{2^1 K} = 2^5$ (No. of Blocks)

Size of a Page = Size of a Block = Size of

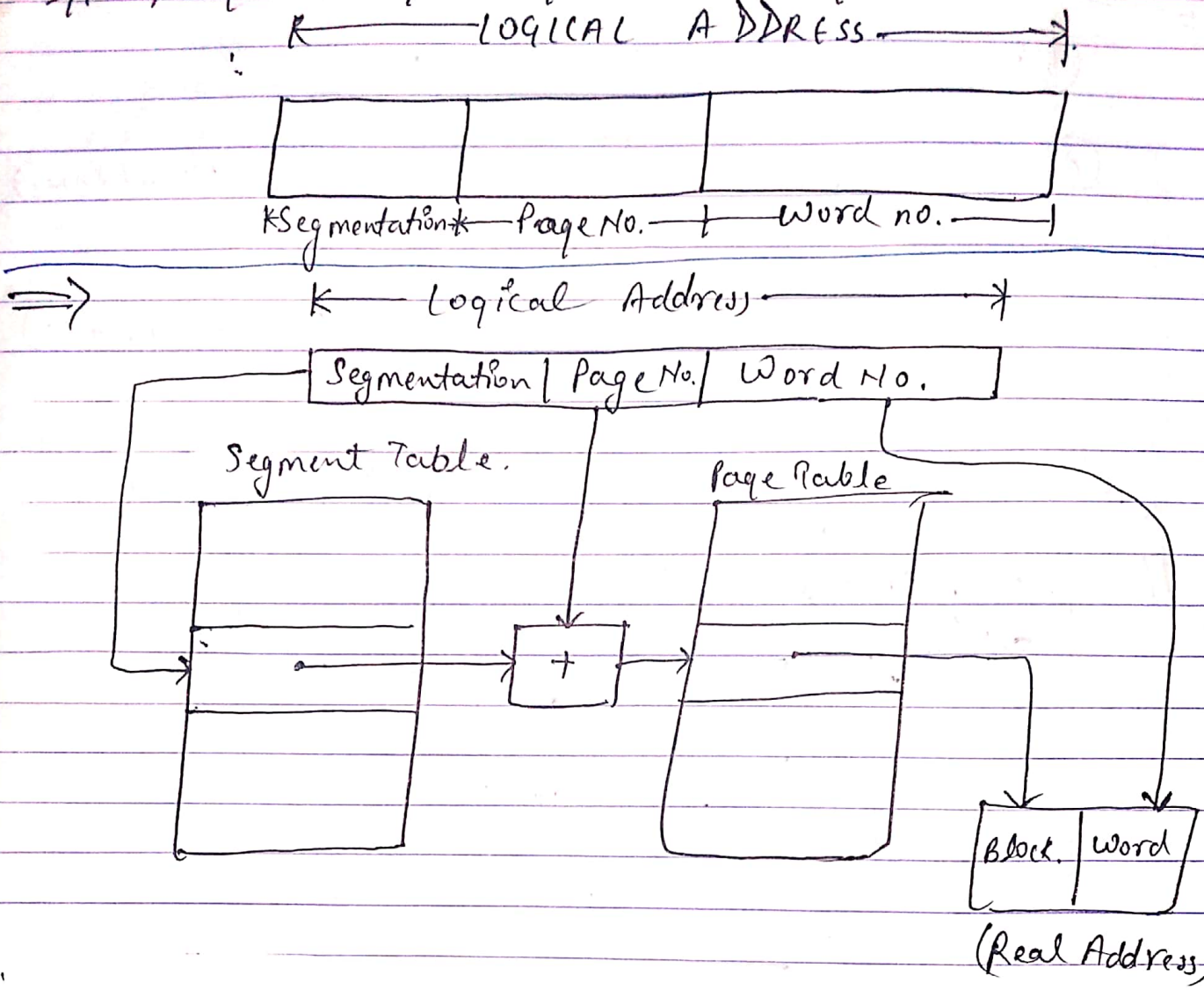
If SEGMENTATION.

Segments.



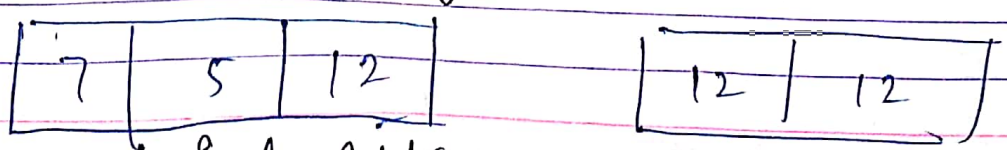
Variable Size Partitioning Partitioning.

SEGMENTED PAGE MAPPING -



Q A logical address space in a computer system consist of 128 segments. Each segment can have upto 32 pages of 4K words in each. Physical memory consists of 4K ~~words~~ blocks in ~~each~~ of 4K words in each. Formulate the logical & physical address formats -

Solution-



Logical Address

Real Address

(Associative memory)

TLB (Translation lookaside buffer).

TLB is a type of associative memory which is used to store the mapping table of virtual memory.

PAGE REPLACEMENT ALGORITHMS

*When a page fault occurs in a virtual memory system, it signifies that the page referenced by the C.P.U is not in main memory. A new page is then transferred from auxiliary memory to main memory. If main memory is full, it would be necessary to remove a page from a memory block to make room for the new page. Therefore, ~~The~~ policy a page replacement algorithm is used for choosing pages to remove from the main memory. Some algorithms are -

* FIFO (First In First Out) Algorithm-

The first in first out algorithm select for replacement the page that has been in memory the longest time.

* LRU (Least Recently Used) Algorithm-

This algorithm select the page for replacement that least recently used.

Exercise -

Q 1 A virtual memory has a page size of 1K words. There are 8 pages & 4 blocks. The associative memory page table contains the following entries.

Page	Block
0	3
1	1
4	2
6	0

make a list of all virtual addresses (in decimal) that will cause a page fault if used by the C.P.U..

Q 2- A virtual memory system has an address space of 8K words, a memory space of 4K words, & Page & block size of 1K words. The following page reference changes occur during a given time - interval. ~~If the same~~ is referenced ago Determine the four pages that are reside in main memory after each page referenced change if the replacement

algorithm used is
(a) FIFO & (b) LRU. — Reference string.

4 2 0 1 2 6 1 4 0 1 0 2 3 5 7

Solution (1) Exercise Q 1 Answer :

Page fault occur in Page 2, 3, 5 & 7.

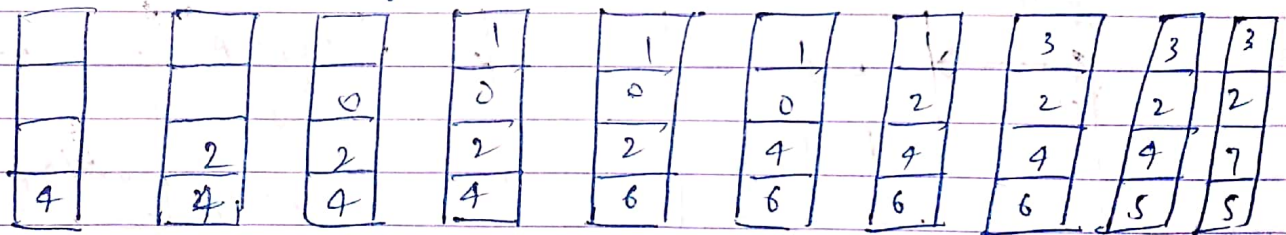
& the address list are.

Virtual Address List -

2 - (2048 - 3071)
3 - (3072 - 4095)
5 - (5120 - 6145)
7 - (7168 - 8191)

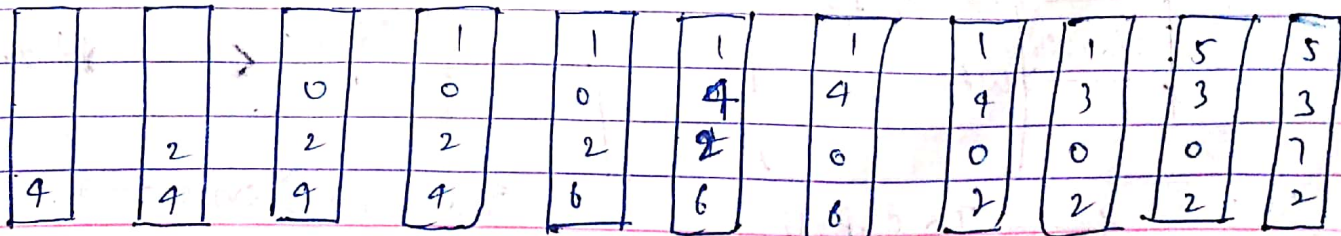
Solution (2)

(a) Using FIFO.



Total Page fault = 10.

(b) Using LRU



Total Page fault = 11.