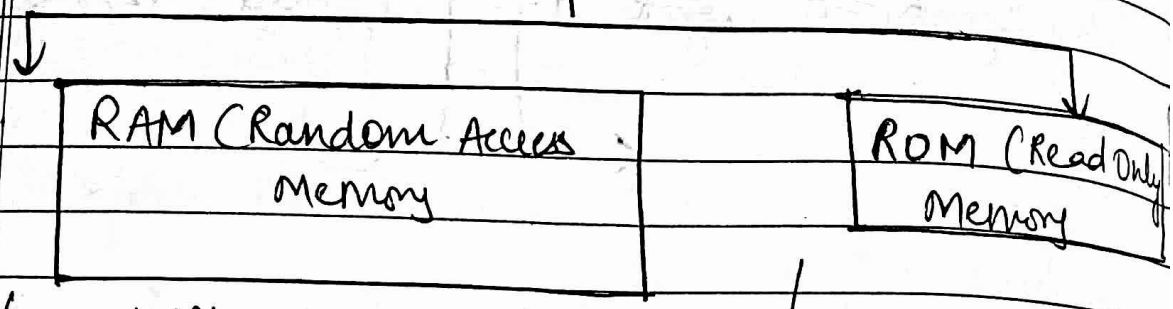
Unit-4
Main Memory

Page No:
Date: / /

Main Memory & its types



RAM
- ↳ volatile
- ↳ R/W memory
- ↳ Access method Random
- ↳ writing data is faster

ROM
- ↳ non-volatile
- ↳ only for read
- ↳ Access method Random
- ↳ writing data is slower

Types of RAM.

1) **SRAM** :- Static RAM.
   - * called static because data retain (does not change till power is ON)
   - * uses flip-flop as memory cell.
   - * size of memory cell is larger
   - * flip flop stores '0' and '1'
   - * memory density (memory cell / area) is low. So, access time is less.
   - * faster
   - * costlier
   - * used in cache memory.

Page No.
Date: / /

2) DRAM :- Dynamic RAM
* Called dynamic because data changes in the presence of 'power' also
* uses capacitor as memory cell.
* Size of memory cell is smaller
* presence of electric charge on capacitor show '1' and absence show '0'.
* Due to natural tendency of charge decay on capacitor, the data '1' becomes '0' after electric charge decay below threshold level.
* Memory density is high. So access time is high.
* Slower and cheaper
* used as primary memory

Types of ROM

① PROM :- Programmable ROM.
* one time programmable by user.
* Data once written cannot be erased.

② EPROM :- Erasable PROM
* multi time programmable by user.
* Data once written can be erased by UV rays.
* whole chip is erased and written again.
* If error, complete chip need to be rewrite again.
* More time re-programming

Date: / /

③ EEPROM :- Electrical EPROM

* multi times programmable by user
* Data once written can be erased using electrical signal
* byte / block level erasing & written again.
* less time in reprogramming

④ Flash Memory :- * Similar to EEPROM
* multi time programmable by user
* Data once written can be erased using electrical signal.
* bit level erasing and written again.
* Super fast in reprogramming so, called flash memory.

Introduction to Memory

* In computer memory is the most essential component of the normal functioning of any system, without it computer can't perform simple tasks.
* Computer memory is any physical device, used to store data, information or instruction temporarily or permanently.

Date: / /

* It is the collection of storage units that stores binary information in the form of bits.

* The memory block is split into a small number of components, known as cells.

* Each cell has a unique address to store data in memory, ranging from zero to memory size - one.

* Eg:- If the size of computer memory is 64k words, the memory units have 64 * 1024 = 65536 locations or cells.

Therefore the address of the memory is cells varies from 0 to 65535

Characteristics of memory

Following are the different features of the memory system that includes :-

1. Location :- It represents the internal or external location of the memory in a computer.

Internal memory is inbuilt in computer memory, it is also known as primary memory. Eg:- registers, cache and main memory.

Page No:
Date: / /

External memory is the separate storage device from the computer like disk, tape, USB pen drive.

2. Capacity :- Storage capacity can vary in external and internal memory

External device storage capacity is measured in terms of bytes.

Internal memory is measured with bytes or words.

The storage word length can vary in bits, like 8, 16 or 32 bits.

3. Access Methods :-

(i) Sequential

→ Start at the beginning and read through in order
→ Access time depends on location of data and previous location.
→ Ex: tape

(ii) Direct

→ Individual blocks have unique address
→ Access is by jumping to vicinity plus sequential search

→ Access time depends on location and previous location.
→ Eg:- Disk

(iii) Random :-

→ Individual addresses identify locations exactly.
→ Access time is independent of locations or previous access.
→ Eg:- RAM.

(iv) Associative

→ Data is located by a comparison with contents of a portion of the store.
→ Access time is independent of location or previous access.
→ Eg:- Cache.

4. Unit of Transfer :- A unit of transfer measures the transfer rate of bits that can be read or write in or out of the memory devices.

The transfer rate of data can be different in external and internal memory.

Internal memory: The transfer rate of bits is mostly equal to the word size.

External memory: The transfer rate of bit

Date: / /

or unit is not equal to the word length, it is always greater than a word or may be referred to as block

5. Organisation:- Physical arrangement of bits into words.

Not always obvious
Eg:- interleaved

6. Performance:-

(i) Access time:- Time between presenting the address and getting the valid data.

(ii) Memory cycle Time:- Time may be required for the memory to "recover" before next access.

Cycle time is access + recovery.

(iii) Transfer Rate:- Rate at which data can be moved.

7. Physical Type:-

→ Semiconductor - RAM
→ Magnetic - Disk and Tape
→ Optical - CD & DVD
→ Others - Bubble
Hologram.

Page No:
Date: / /

8. Physical characteristics:-

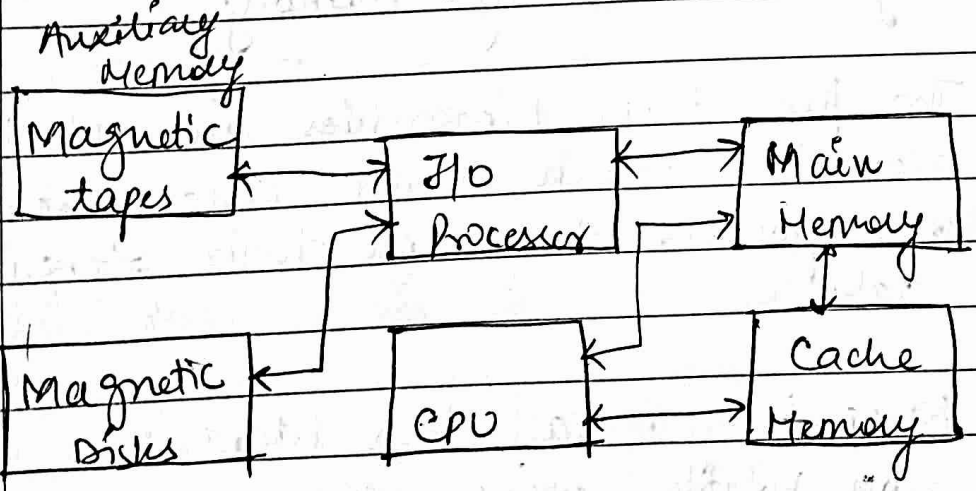
→ Decay
→ Volatility
→ Erasable
→ Power consumption

Memory Hierarchy

* The memory unit is an essential component in any digital computer since it is needed for storing programs and data.

* One memory unit does not have enough space to accomodate all the programs used in a typical computer.

* Therefore, it is more economical to use low cost storage device to serve as backup for storing that information which is not currently used by CPU.



Components of a typical memory hierarchy

Page No:
Date: / /

* Main memory directly communicates with the CPU.

* Devices which people provide backup storage, known as auxiliary memory.

* Cache is a special very high speed memory, sometimes which is used to increase the speed of processing.

* The total memory capacity of computer can be visualised as being a hierarchy of component.

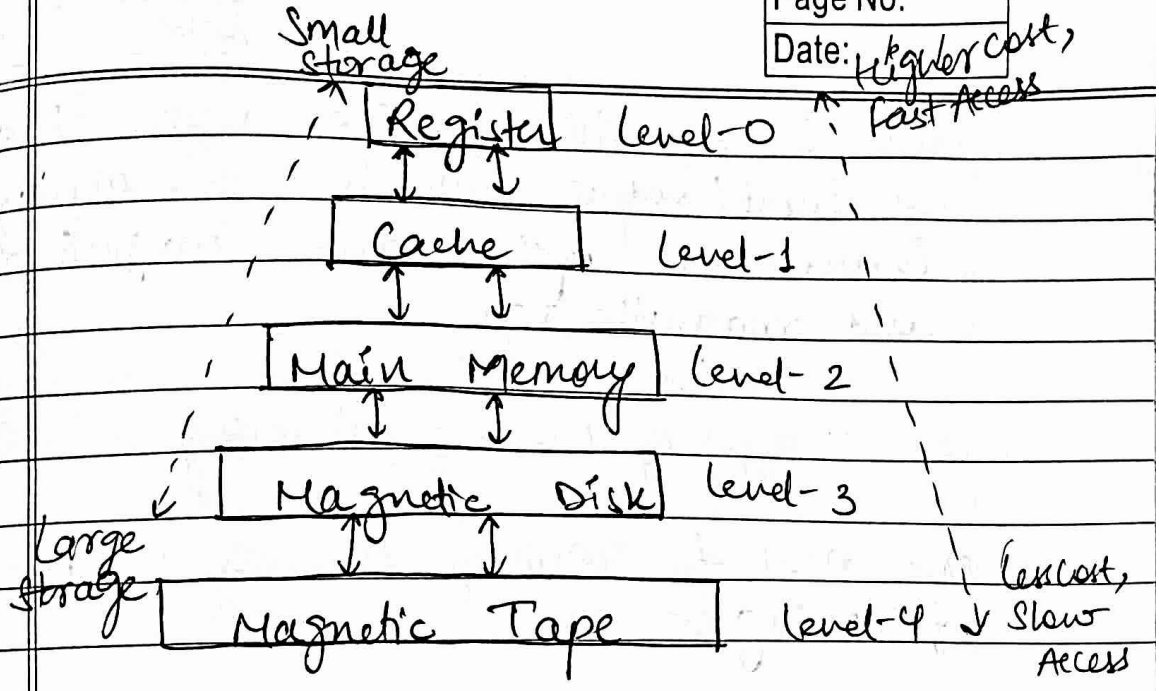
* Memory hierarchy is to obtain the highest possible access ~~period~~ speed while minimizing the total cost of the memory system

* Here the devices are arranged in a manner fast to slow, that is from register to Tertiary memory.

* The first three hierarchies are volatile memories which mean when there is no power, they lose their stored data!

* Whereas the last two hierarchies are non-volatile which means they store the data permanently.

Page No:
Date:



* The memory in a computer can be divided into five hierarchies based on the speed as well as use.

* The processor can move from one level to another based on its requirements.

* The five hierarchies in the memory are registers, cache, main memory, magnetic discs, and magnetic tapes.

Primary Memory

* It is also known as internal memory, and this is accessible by the processor directly. This memory includes main, cache, as well as CPU registers.

Secondary Memory

* It is also known as external memory, and

this is accessible by the processor through an input/output module. This memory includes an optical disk, magnetic disk and magnetic tape.

Advantages of Memory Hierarchy

The need for memory hierarchy includes the following :-

* Memory distribution is simple and economical.
* Removes external destruction.
* Data can be spread all over.
* Permits demand paging & pre-paging.
* Swapping will be more proficient.
* Reduces average cost per bit of entire memory system of computer
* It maintains average data transfer rate of entire memory system.
* It makes possible to recover data, if unknowingly modified or corrupted at any level of the hierarchy.

Main Memory

* The main memory acts as the central storage unit in a computer system.
* It is a relatively large and fast memory which is used to store programs and data during the run time operations.
* The primary technology used for the main

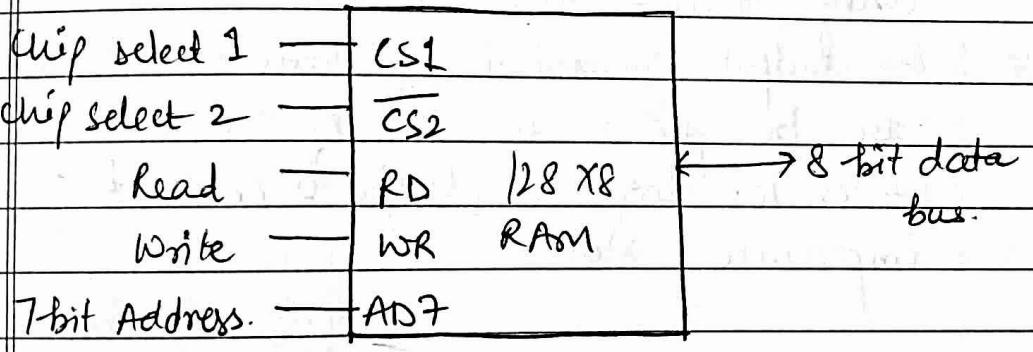
Page No:
Date: / /

memory is based on semiconductor integrated circuits.

* The integrated circuits for the main memory are classified in two categories:-
(i) RAM (Random Access Memory) integrated circuit chips.
(ii) ROM (Read Only Memory) integrated circuit chips.

RAM chips

* Integrated circuits RAM chips are needed classified into two possible operating modes:
(i) Static
(ii) Dynamic



Block Diagram.

CS1	CS2	RD	WR	Memory function	State of data bus
0	0	X	X	Inhibit	High-Impedance
0	1	X	X	Inhibit	High-Impedance
1	0	0	0	Inhibit	High-Impedance
1	0	0	1	Write	Input data to RAM
1	0	1	X	Read	Output data to RAM
1	1	X	X	Inhibit	High-Impedance

Page No:
Date: / /

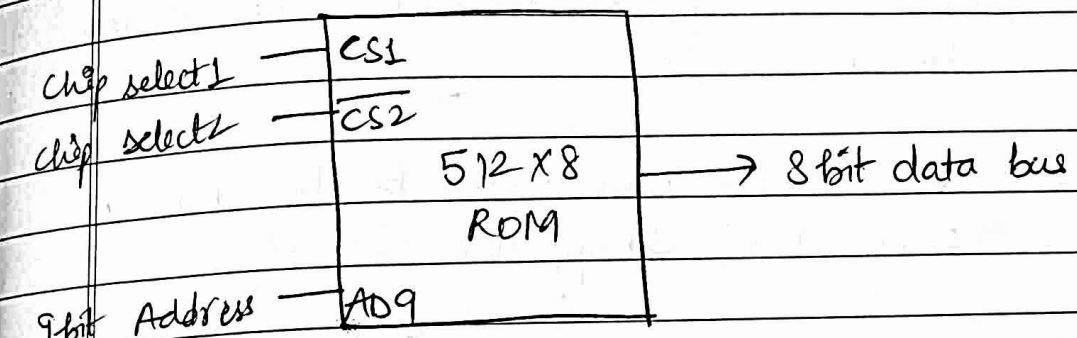
* RAM chips are available in a variety of sizes and are used as per the system requirement.
* 128 x 8 RAM chip has a memory capacity of 128 words of eight bits (one byte) per word.
* This requires a 7bit address and an 8bit bidirectional data bus.
* The 8bit bidirectional data bus allows the transfer of data either from memory to CPU during a read operation or from CPU to memory during a write operation.
* The read and write inputs specify the memory operation.
* The two chip select (CS) control inputs are for enabling the chip only when the microprocessor selects it.
* The bidirectional data bus is constructed using three-state buffers.
* The output generated by three-state buffers can be placed in one of the three possible states ie. logic 1, logic 0 or a high-impedance state.

ROM chips.

* The primary component of the main memory is RAM integrated circuit chips, but a portion of memory may be constructed with ROM chips.
* A ROM memory is used for keeping programs and data that are permanently resident in the computer.

Page No:
Date: / /

* Apart from the permanent storage of data, the ROM portion of main memory is required for storing an initial program known as bootstrap loader.
* The key function of the bootstrap loader program is to start the computer software operating when power is turned on.



* ROM chips are available in a variety of sizes and are used as per the system requirement.
* ROM can only perform read operation and hence the data bus can only operate in an output mode.
* The 9 bit address bit lines in the ROM chip specify any one of the 512 bytes stored in it.
* The value for chip select 1 and chip select 2 must be 1 and 0 for the unit to operate.
* Otherwise, the data bus is said to be in a high-impedance state
* 512 x 8 ROM chip has a memory capacity of 512 words of eight bits (one byte) per word.
* For the same size chip, it is possible to have more bits of ROM than RAM, hence

Page No:
Date: / /

diagram specifies 512 byte ROM while RAM has only 128 byte.

8.1 (a) How many 128 x 8 RAM chips are needed to provide a memory capacity of 2048 bytes?

(b) How many lines of the address bus must be used to access 2048 bytes of memory? How many of these lines will be common to all chips?

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

Soln: (a) No. of RAM chips = $\frac{\text{Total RAM size}}{\text{One RAM size}}$

$= \frac{2048 \text{ bytes}}{128 \times 8} = \frac{2048 \times 8}{128 \times 8}$

$= \frac{2^{11}}{2^7} = 2^4$

$= \boxed{16 \text{ chips}}$

(b) No. of addresses lines to access 2048 bytes of memory = 2048 bytes

$= 2048 \times 8$

(11) ← address lines ↓ ; 8 → data lines

$= 2^{11}$ → No. of address lines

$= \boxed{11}$

Page No:
Date: / /

No. of common addr lines = No. of addr lines in 1 chip

$= 128 \times 8$ → 7 → No. of common address lines

$= 7$

∴ No. of RAM chips = 16

decoder size = 4×16

↓

No. of decoded lines = 4

RAM, ROM Address Mapping

Q1. A computer employs RAM chips of 256×8 and RAM chips of 1024×8. The computer system needs 2K bytes of RAM, 4K bytes of ROM and four interface units, each with 4 registers. A memory mapped I/O configuration is used. The 2 highest order bits of the address bus are assigned 00 for RAM, 01 for ROM and 10 for interface registers.

(i) How many RAM & ROM chips are needed.
(ii) Draw a memory address map for the system.
(iii) Give the address range in hexadecimal for RAM, ROM & I/O interface.

Sol(i) No. of RAM chips = $\frac{\text{Total RAM size}}{\text{one RAM size}}$

$= \frac{2k \text{ bytes}}{256 \times 8}$

$= \frac{2k \times 8}{256 \times 8} = \frac{2k}{256}$

$= \frac{2^{11}}{2^8} \Rightarrow 2^3 =$ **8 RAM chips**

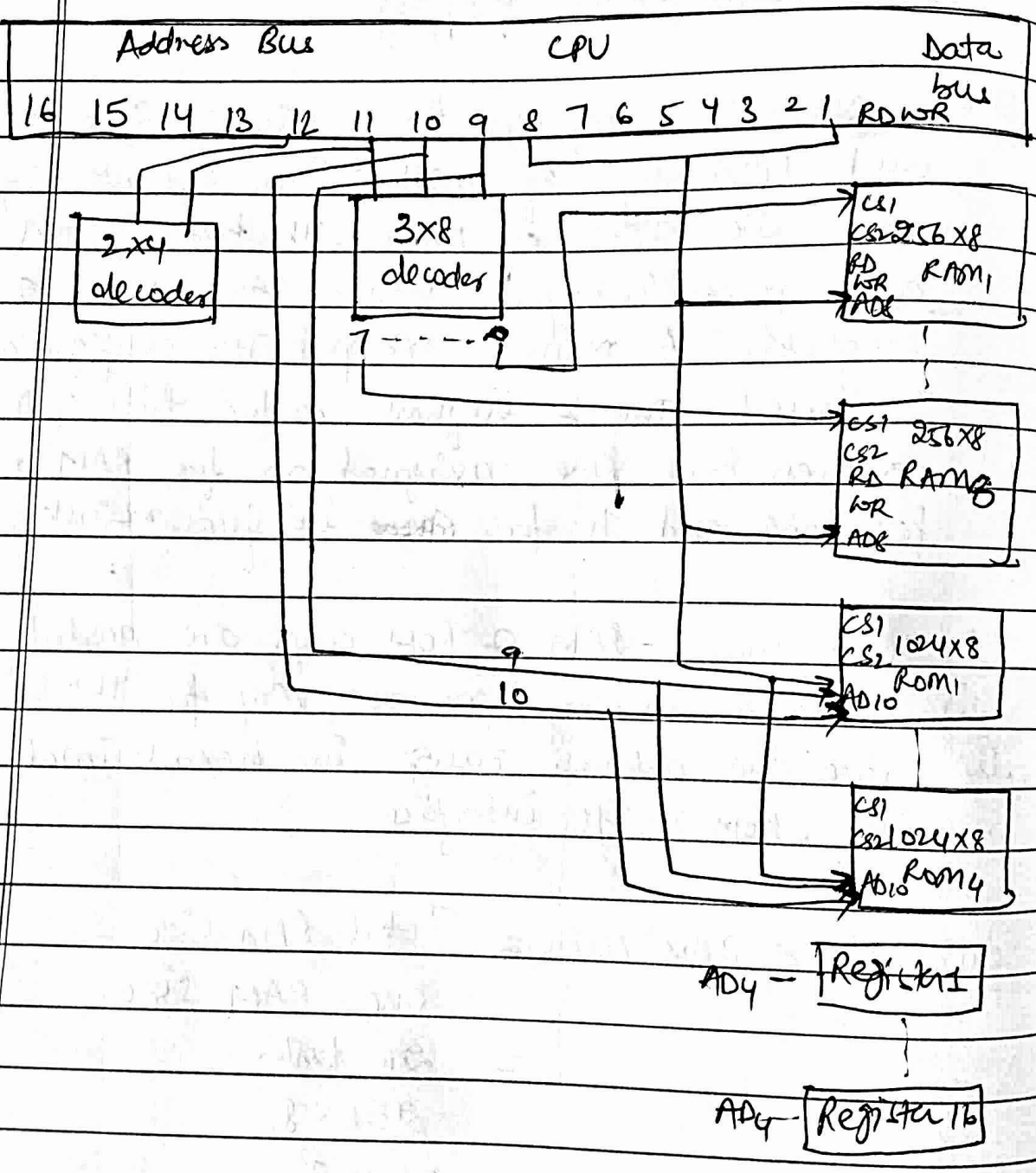
Date: / /

(ii) No. of ROM chips = $\frac{\text{Total ROM Size}}{\text{One RAM Size}}$

$= \frac{4k \text{ bytes}}{1024 \times 8} = \frac{4k \times 8}{1024 \times 8}$

$= \frac{4 \times 1024}{1024} =$ **4 ROM chips**

(iii) No. of I/O interface registers = $4 \times 4 =$ **16 registers**



Page No:
Date: / /

(iii) (iii) Memory Address Map

Components	Address	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
RAM	0000 - 07ff	00 000 ⟷ xxxxxxxx (3×8 decoder)
ROM	4000 - 4fff	0100 ⟷ xxxxxxxxxx (2×4 decoder)
I/O Interface	8000 - 800f	10 00000 00000 xxxx

RAM ← Start address 0000000000000000
End address 00000 1111111111111

ROM ← Start address 0100 0000000000000
End address 0100 1111111111111111

I/O Interface ← Start addr. 1000000000000000
End addr. 1000000000001111

Qe A computer system employs 128×8 q RAM chips, 512×8 q ROM chip. The system needs a total q 512 bytes q RAM & 512 bytes q ROM. Find the no. q RAM, ROM chips and draw memory address map.

Soln i) No. q RAM chips = Total RAM size / One RAM size

= 512 bytes / 128×8 = 512×8 / 128×8

= 4 RAM chips

Date: / /

No. of ROM chips = $\frac{\text{Total ROM Size}}{\text{one ROM Size}}$

$= \frac{512 \text{ bytes}}{512 \times 8}$

$= \frac{512 \times 8}{512 \times 8}$

= 1 Rom chip.

(ii) Memory Address Map.

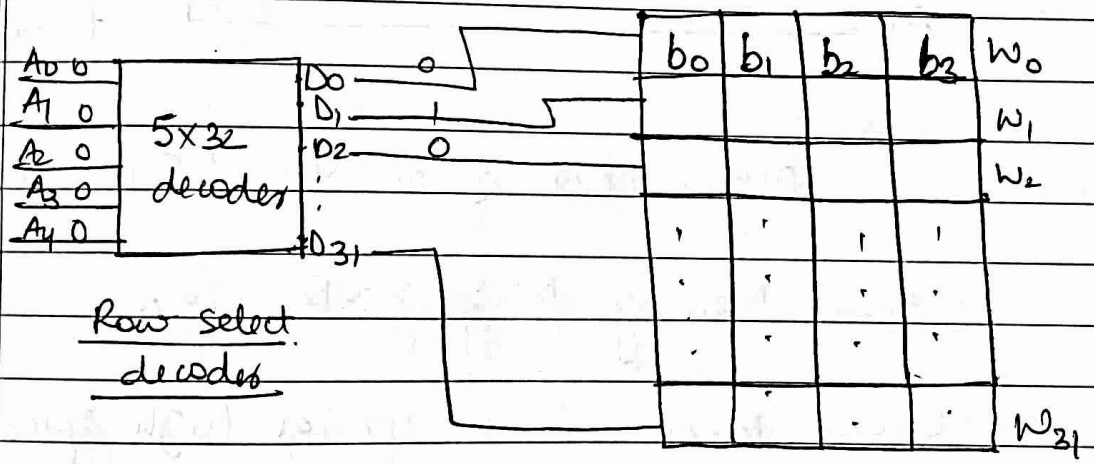
Chips	Address Range	Address lines
		16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RAM_1	0000H - 007F H	0 0 0 0 0 0 0 2×4 XXXXXXX decoder
		0000 0000 0000 0000
		0000 0000 0110 0110
RAM_2	0080 - 00ff	000 000 00 XXXXXXX
		0000 0000 1000 0000
		0000 0000 1111 1111
RAM_3	0100 - 017f	000000 010 XXXXXXX
		000000 010 0000000
		000000 010 1111111
RAM_4	0180 - 01ff	000000 011 XXXXXXX
		000000 011 0000000
		000000 011 1111111
ROM	0200 - 03ff H	000000 1 XXXXXXXXX
		0000 0010 0000 0000
		0000 0011 1111 1111

Page No:
Date: / /

2D and 2.5D RAM.

2D RAM :-

* In 2D RAM organisation, a RAM of $2^n \times m$ consisting 2^n memory words of m bits each.
* It has a row select decoder of size $n \times 2^n$ to select 1 out of 2^n memory words.
* Each row (memory word) have m column (1 column for each individual bit).
* In 2D RAM organisation, hardware is fixed.
* It requires more no. of logic gates.
* 2D RAM is more complex.



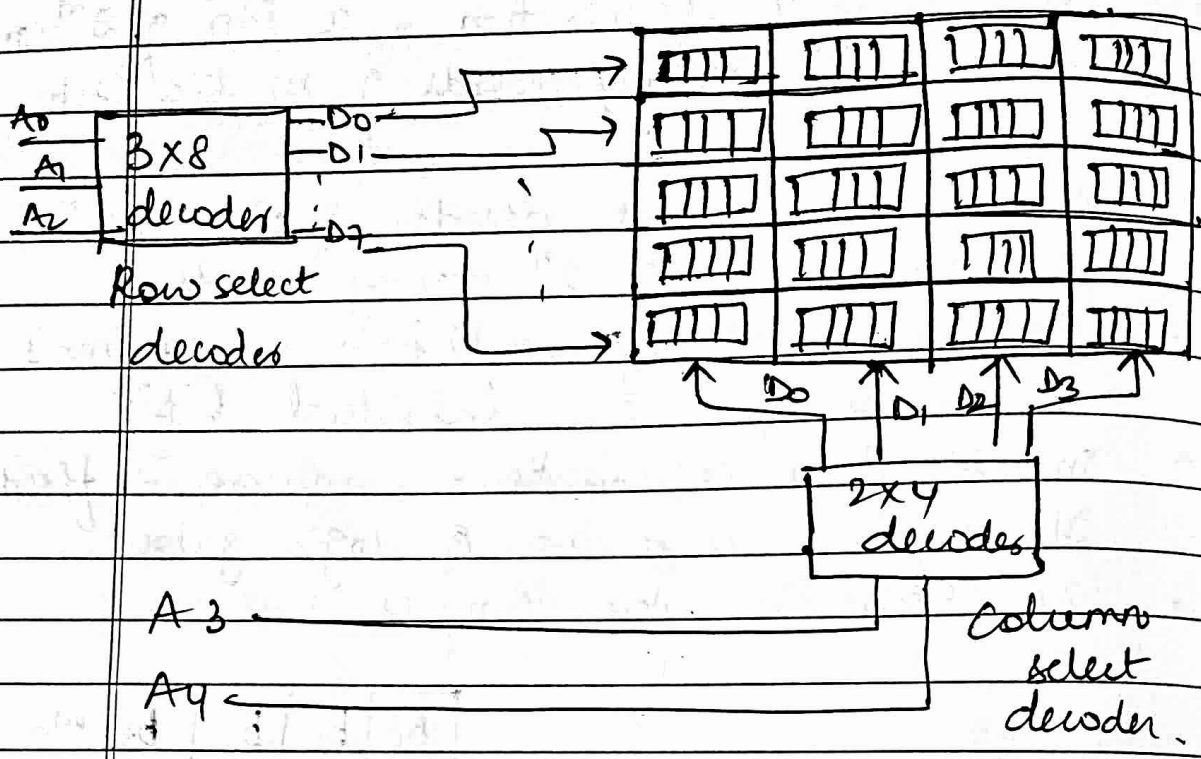
Organisation of 32×4 2D RAM.

2.5D RAM :-

* In 2.5D RAM organization, the no. of address lines are divided into approximately equal parts, one for row select decoder and another for column select decoder.

Page No.
Date: / /

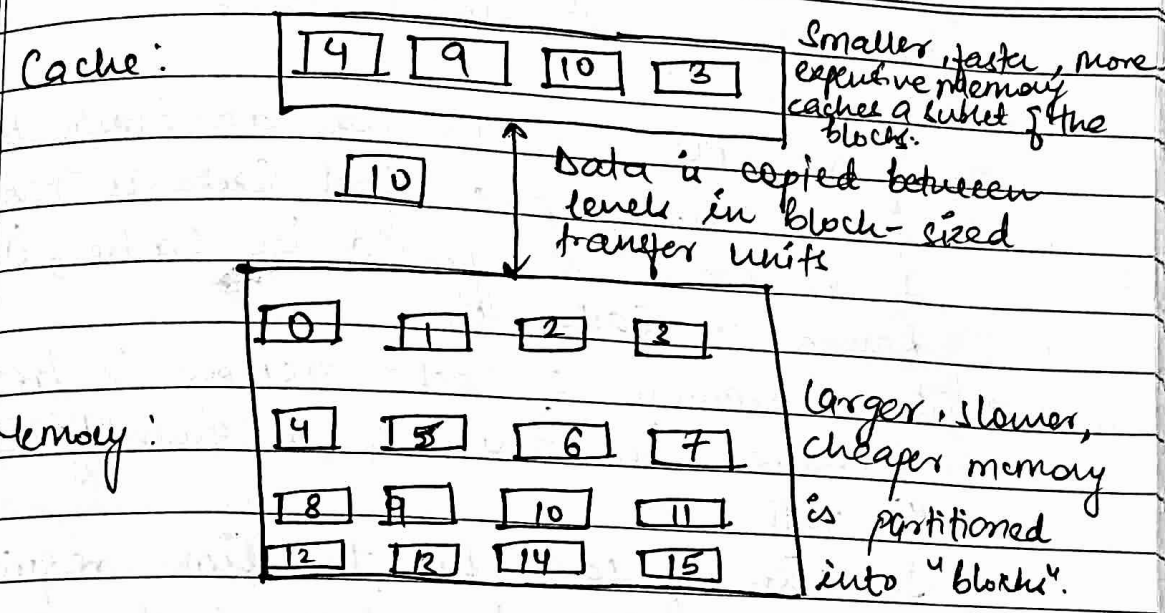
* In 2.5D RAM, hardware is variable
* It requires less no. of logic gates.
* 2.5D RAM is less complex.



Organisation of 32×4 2.5D RAM

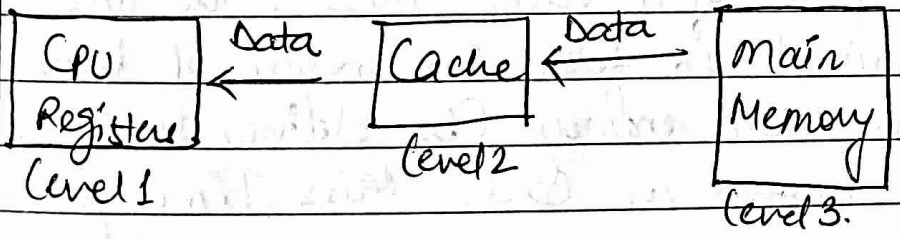
Cache Memory

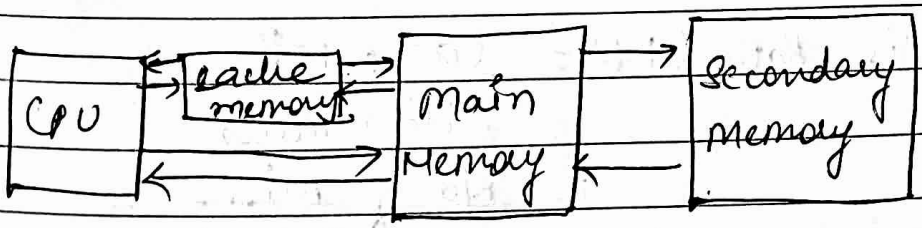
* Cache Memory is a special high speed memory.
* It is costlier than main memory, but economical than CPU registers.
* It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
* Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU.



Why Cache is needed?

* The cache memory is required to balance the speed mismatch between the main memory and the CPU.
* The clock of the processor is very fast, while the main memory access time is comparatively slower.
* Hence, the processing speed depends more on the speed of the main memory.





Date: / /

Cache Memory Performance Issue

* If the CPU needs to read data/instruction from memory, then first Cache is examined. If the word is found in Cache, it is known as Cache Hit.
* The performance of cache memory is frequently measured in terms of a quantity called Hit ratio.
* In case of cache hit the time required to access data from cache is known as Cache hit time.

$$\text{Hit Ratio (h)} = \frac{\text{Hit}}{\text{Hit + miss}} = \frac{\text{No. of hits}}{\text{Total accesses}}$$

* If the CPU needs to read data/instruction from memory, then first Cache is examined. If the word is not found in Cache, it is known as Cache Miss.
* Miss Ratio is the fraction of accesses that are not in the cache ie (1-h).
* In case of a cache miss, the time required to fetch the required block from main memory (to deliver to CPU), is known as Cache Miss Time penalty.

$$\text{Miss Ratio (1-h)} = \frac{\text{Cache Miss}}{(\text{Hit + Miss})} = \frac{\text{No. of Miss}}{\text{No. of Total Accesses}}$$

Page No:
Date: / /

* Cache performance can be approved by using higher cache block size, higher associativity, reduce miss rate, reduce miss penalty, and reduce the time to hit in the cache.

Average Memory Access time = Hit time + Miss Rate × Miss penalty

$t_a = h \times t_c + (1-h) \times (t_c + t_m) = t_c + (1-h) \times t_m$

Where,

t_a = Average memory access time
h = hit ratio.
t_c = Cache memory access time.

Cache Memory Mapping Techniques.

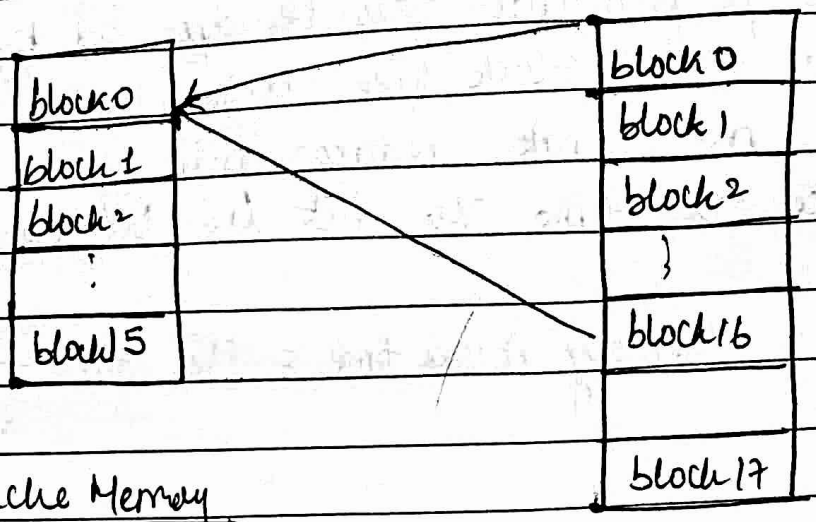
Cache memory mapping means how data is copied (mapped) from main memory to cache memory.

Mapping Techniques.

1) Direct Mapping :- In this mapping, main memory blocks are copied to a fixed block of cache memory, but one at a time.

Let cache memory block no. = C_b
Main memory block no. = m_b
No. of block in cache = C
No. of block in main memory = m

Date: / /



Cache Memory

Main Memory

Cache memory block no. = (main memory block no.) mod (No. of Cache Blocks)

$$C_b = m_b \bmod c$$

Main memory size = 512 x 8
Cache memory size = 64 x 8
Block size = 4 words.

No. of main memory blocks = Total main memory words / block size
= 512/4 = 128 blocks

No. of cache memory blocks = Total cache memory words / block size
= 64/4 = 16 blocks

Cache memory block no. (C_b) = 0 mod 16 = 0
= 16 mod 16 = 0
= 32 mod 16 = 0
= 48 mod 16 = 0.

Page No:
Date: / /

No. of main memory blocks in one block of cache = No. of main memory blocks / No. of cache memory blocks

$= \frac{128}{16} = 8$

Block size = 4 words

Main memory address = 512 x 8 (size) ↓ words

Main memory address = $512 = 2^9$

= 9 bits.

3 bits	4 bits	2 bits
Tag bits	Block bits	word bit

← index bits → = 6 bits

Main memory address (9 bits)

Cache memory word length = 11 bits.

Q. A digital computer has a memory unit of 64K x 16 and a cache memory of 1K words. The cache uses "Direct Mapping" with a block size of 4 words.

(a) How many bits are there in the tag, index, block and word field of address format.

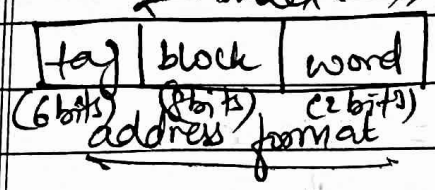
(b) How many bits are there in each ~~format~~ word of cache, and how are they divided

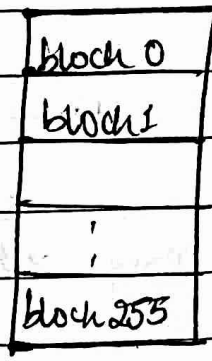
into functions? Include a valid bit.

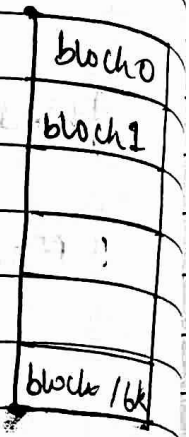
(f) How many blocks can the cache accomodate?

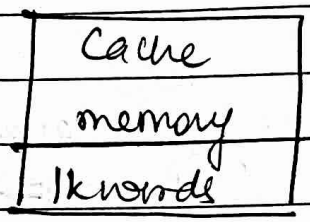
Soln Direct Mapping :-

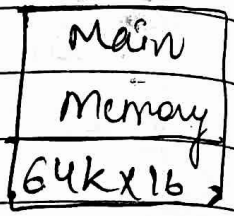
(a)











Tag = No. of main memory block in one block of cache

$$\text{tag} = \frac{\text{No. of main memory blocks}}{\text{No. of cache memory blocks}}$$

=

$$\text{Cache memory block no} = (\text{main memory block no}) \bmod (\text{No. of cache memory blocks})$$

$$\text{No. of cache memory block} = \frac{\text{No. of words in cache m/m}}{\text{block size}}$$

$$= \frac{1k}{4} = \frac{1 \times 10^{24}}{4} = 256$$

No. of cache memory blocks = 256

Planned:

Page No:
Date: / /

No. of main memory block = No. of block words in main memory / block size

$= \frac{64k}{4} = 16k = 16 \times 1024.$

No. of main memory block = 16×1024

$tag = \frac{16 \times 1024}{256} = 64 = 2^6$

No. of tag bits = 6

(b)

valid	tag	data
1 bit	6 bits	16 bits

$16 + 6 + 1 = 23$ bits.

c) ~~Direct Mapping~~

2) Associative Mapping :- In associative mapping, main memory blocks are copied into any block of cache memory.

Let Main memory size = 512×8 (512 words, 8 = word size)

Cache memory size = 64×8 (64 words, 8 = word size)

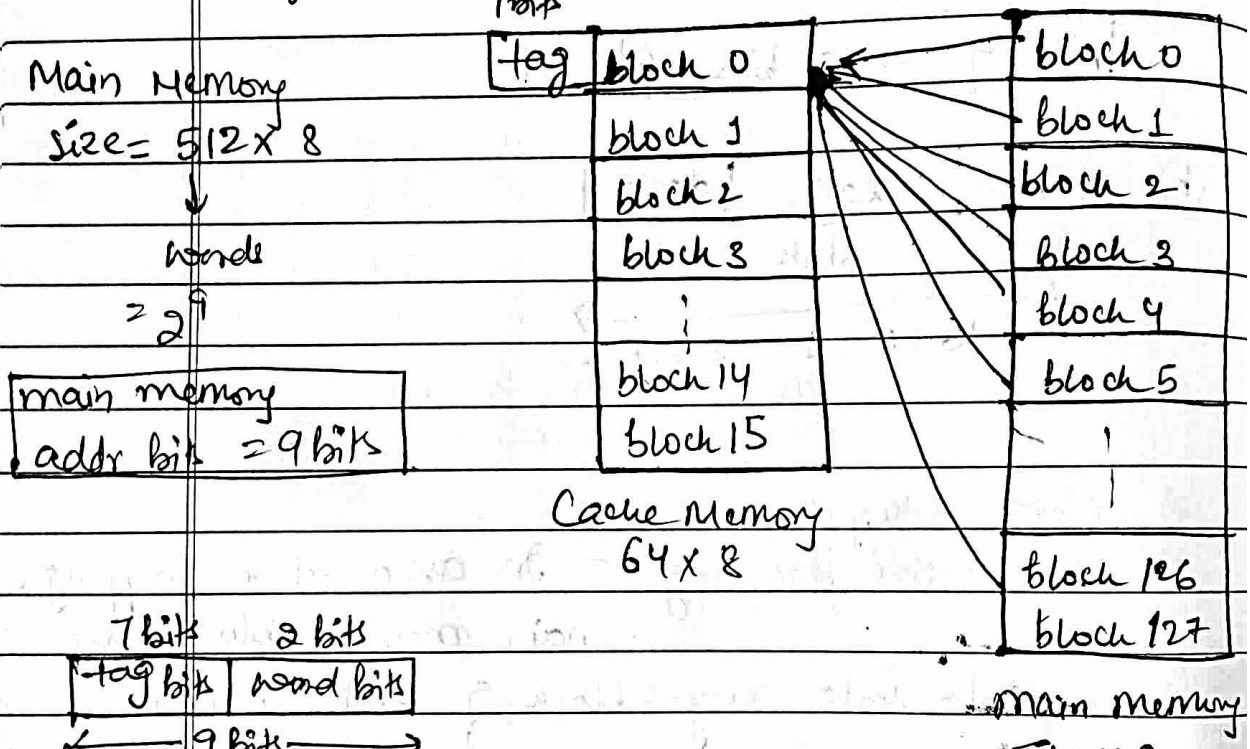
block size = 4 words

Page No:
Date: / /

No. of main memory blocks $= \frac{512}{4} = 128$ blocks

No. of cache memory blocks $= \frac{64}{4} = 16$ blocks

No. of main memory block in one block of Cache $= 128 = 2^7$



Associative mapping

3) Set Associative mapping :- It is a combination of direct mapping and associative mapping.

Set Associative = Direct + Associative
mapping mapping mapping

Planned:

Page No:
Date: / /

In this, cache memory is divided into set.

Set = group of blocks
Blocks = group of words.

$$\text{Cache memory Set No.} = (\text{main memory block no.}) \bmod (\text{No. of sets in cache memory})$$

$$\text{No. of sets in cache memory} = \frac{\text{No. of cache memory blocks}}{\text{set size}}$$

$$\text{No. of blocks in Cache memory} = \frac{\text{No. of words in Cache memory}}{\text{block size}}$$

$$\text{No. of blocks in main memory} = \frac{\text{No. of words in main memory}}{\text{block size}}$$

Let set size = 2 (Two-way set associative)
Block size = 4 words
main memory size = 512 x 8
cache memory size = 64 x 8

$$\text{No. of main memory blocks} = \frac{512}{4} = \boxed{128 \text{ blocks}}$$

$$\text{No. of cache memory blocks} = \frac{64}{4} = \boxed{16 \text{ blocks}}$$

$$\text{No. of sets in Cache memory} = \frac{16}{2} = \boxed{8 \text{ sets}}$$

Page No:
Date: / /

Cache memory set No. $\Rightarrow 0 \bmod 8 = 0$

$\Rightarrow 8 \bmod 8 = 0$

$\Rightarrow 16 \bmod 8 = 0$

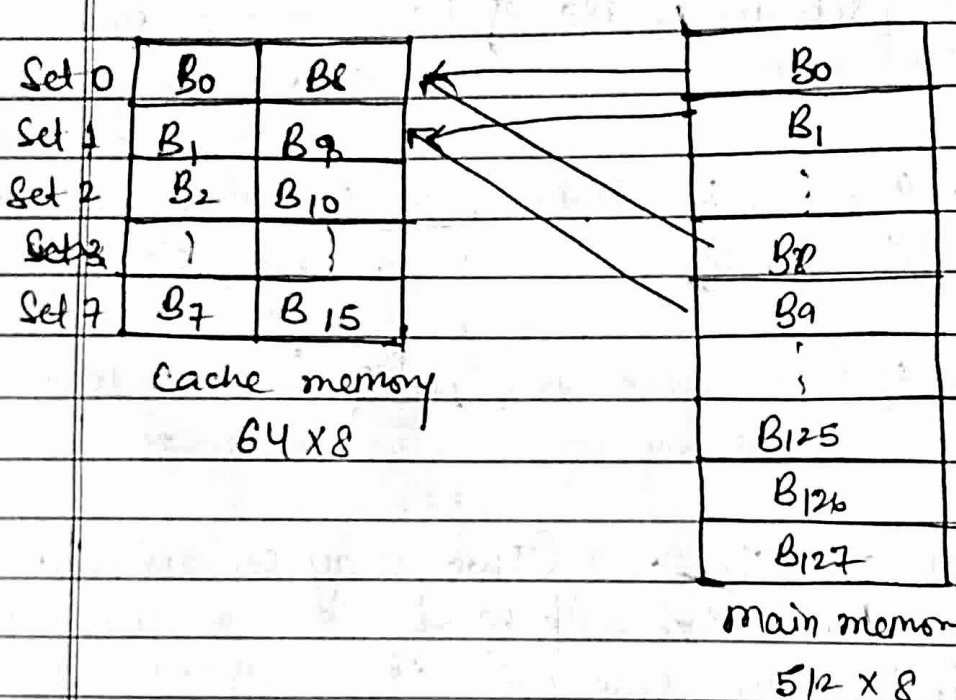
$\Rightarrow 24 \bmod 8 = 0$

$\Rightarrow 1 \bmod 8 = 1$

$\Rightarrow 9 \bmod 8 = 1$

$\Rightarrow 17 \bmod 8 = 1$

$\Rightarrow 25 \bmod 8 = 1$



No of main memory blocks in one set of cache (2 at a time) = $\frac{\text{main memory blocks}}{\text{Cache memory blocks}} = \frac{128}{8} = 16$

tag $= 2^4$

Page No:
Date: / /

Q1 A two-way set associative mapping cache memory uses blocks of four words. The cache can accomodate a total of 2048 words of main memory. The main memory size is 128K X 32.

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

(b) what is the size of cache memory.

Soln. Set-Associative mapping

Set size = 2
Block size = 4

Set 0	B_0	B_{256}
Set 1	B_1	B_{257}
	⋮	
Set 255		

2048 X 32
Cache memory.

B_0
B_1
B_2
⋮
B_{256}
B_{257}
5258
⋮

128K X 32
main memory.

No. of cache memory block = Total words in cache / Block size

$= \frac{2048}{4} =$ 512 blocks

Page No.
Date: / /

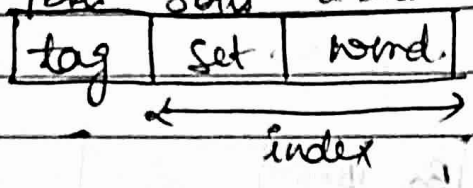
No. of Cache memory sets = Total block in cache / Set size

$= \frac{512}{2} = \boxed{256 \text{ sets}}$

No. of main memory blocks = Total words in main mem / Block size

$= \frac{128\text{ k}}{4} = 32\text{k}$

(a)



address format

tag = main memory blocks / Cache memory sets $= \frac{32\text{k}}{256} = 128$

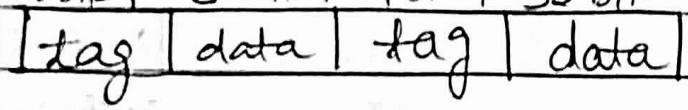
$= 2^7 = 7$ bits

Set = No. of sets in cache

$= 256 = 2^8 = 8$ bits

7bits + 32bits + 7bits + 32 bits = 78 bits

(b)

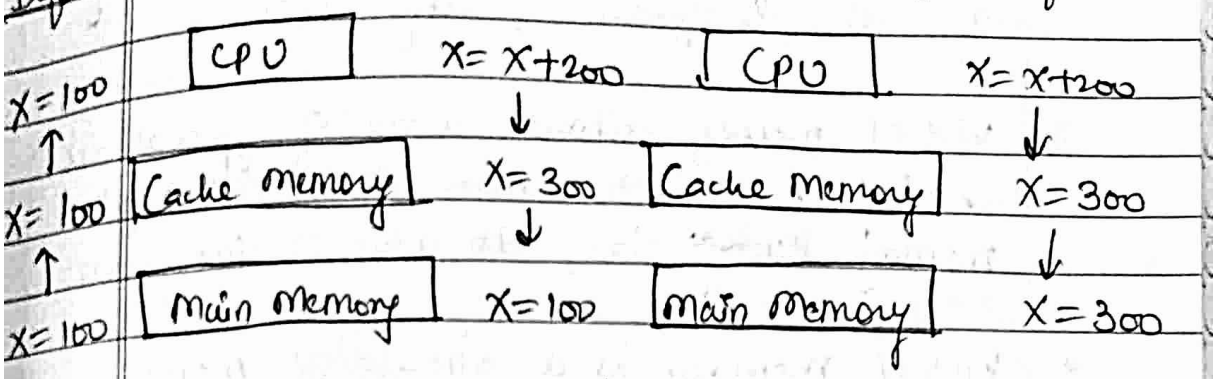


Cache size = 1024 x 78

Planned:

Page No:
Date: / /

Write Back v/s Write through in cache



	Write Back	Write Through
1.	Data is updated only in Cache Memory.	1. Data is updated both in cache memory and main memory.
2.	Simple method to implement	2. Complex method to implement.
3.	When a cache block is not required then selected cache block is written back to main memory to UPDATE the main memory.	3. When cache block is not required, there is no need to written back to main memory.
4.	Commonly used method.	4. Unpopular method.
5	Unreliable method.	5. Reliable method
6	No data Redundancy	6. Data Redundancy
7.	No wastage of time as data update occurs only in cache memory	7. Wastage of time as each data updates both in cache and main memory

Page No:
Date: / /

Virtual Memory :- Is a concept of that gives an illusion to the user that he has sufficient memory to execute any application program of any size.

* Virtual memory allows a no. of applications having total size more than the main memory ~~having~~ size, to run at the same time.
* Virtual memory is a simulated memory that is written to a file on the hard drive. This file is called page file or swap file. It is used by operating system to simulate physical RAM by using hard disk space.
* In windows 1.0, 2.0 versions, there was no virtual memory, so we were not able to run a no. of applications due to error run out of RAM space.
* However from windows 3.0 onwards, concept of virtual memory was introduced.
* To implement it, a portion of hard drive is reserved by the system. This portion can be either a file or a separate partition. In windows, it is a file called pagefile.sys. In linux, a separate partition is used for virtual memory.
* When the system needs more memory, it maps some of its memory addresses out to the hard disk drive. This extra memory does not actually exist in RAM, it is the

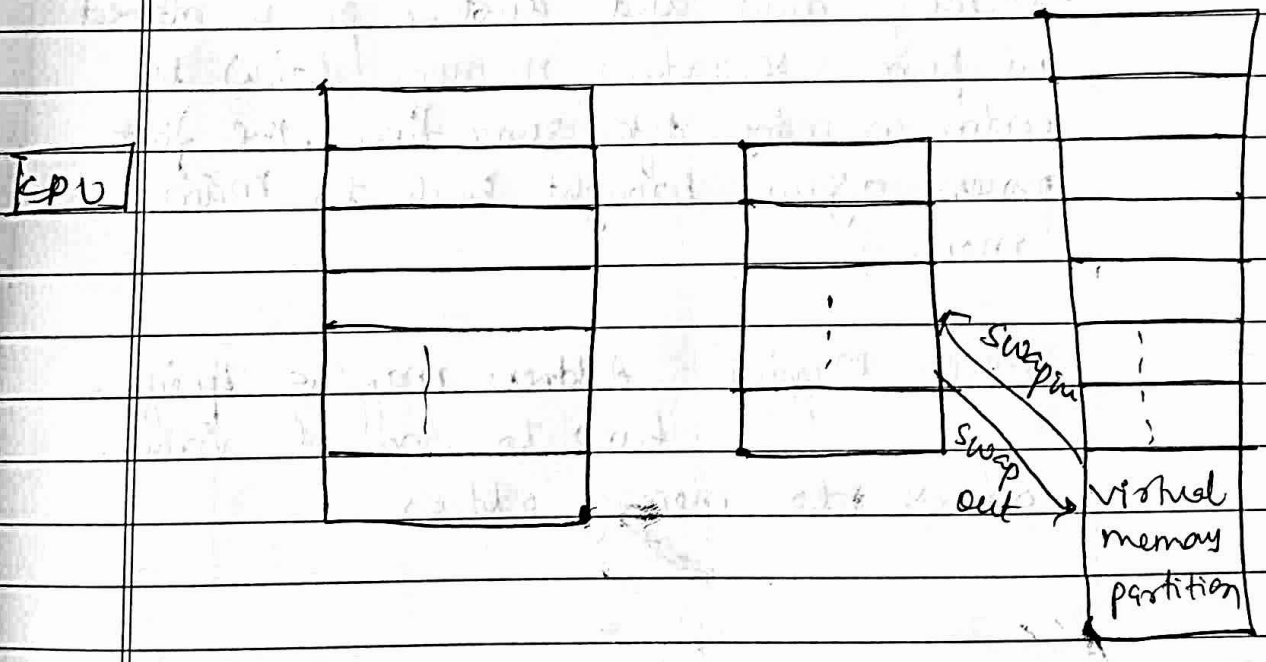
Page No:
Date: / /

Storage space on the disk drive.

* The more RAM (main memory) your computer has, the faster your programs will run.
* If lack of RAM is slowing our computer, then we can increase virtual memory to compensate. However adding more RAM gives high speed rather than virtual memory because it takes more time to swap data from hard disk than from RAM.
* for windows 10,

initial virtual memory size = 1.5 × RAM size
= 1.5 × 4GB
= 6 GB.

Maximum virtual memory size = 3 × initial size
= 3 × 6 GB
= 18 GB.



Page No:
Date: / /

Virtual address (Logical Address) :- Each address in virtual memory is called virtual address

Address space :- Set of all virtual addresses is called address space

Memory Address (Physical Address) :- Each address in main memory is called memory address

Memory space :- The set of all memory addresses is called memory space

Swapping :- Swapping is a mechanism in which a process temporarily moved out from main memory to secondary memory (disk) and another process moved in from secondary memory (disk) to main memory. After some time, the first process again brought back to main memory.

Address Mapping :- Address mapping specify how to convert virtual address to memory address.

Page No:
Date: / /

Virtual memory Implementation

- Using Paging
- Using Segmented paging.

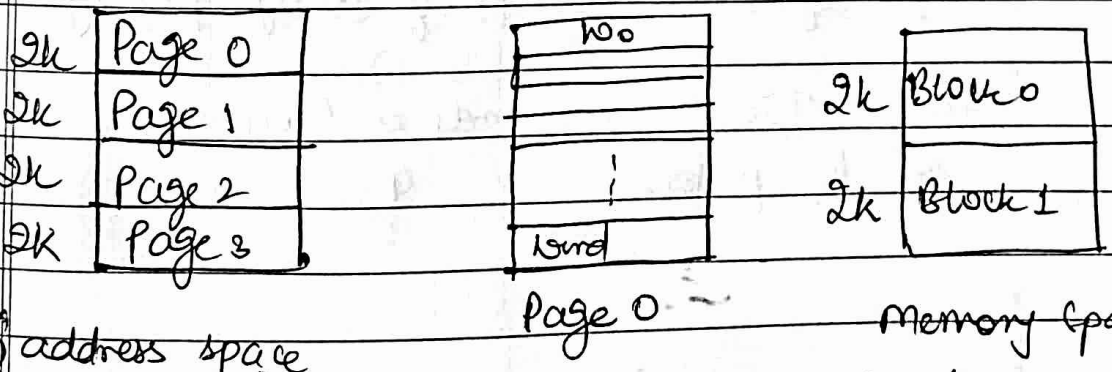
Address Mapping using pages

Page:- Virtual memory is divided into equal size groups. Each group is called page.

① Virtual memory (address space) is a group of pages

② Each page consists of a no. of words called page size.

Let virtual memory size (addr space) = 8k words
Page size = 2k words
= block size.



8 address space (virtual memory) 8k words

Page 0

Memory space (main memory) 4k words.

Page No:
Date: / /

Block (Frame):- Main memory is divided into equal size of groups. Each group is called Block (frame).

① Main memory (memory space) is group of blocks (frames)

② Each block (frame) consists of a no. of words called Block size.

Page size = Block size

(a) No. of pages = $\frac{\text{address space size}}{\text{page size}}$

(b) No. of blocks = $\frac{\text{memory space size}}{\text{block size}}$

(c) virtual address bits = n
address space = 2^n.

d) memory address bits = m bits
memory space = 2^m.

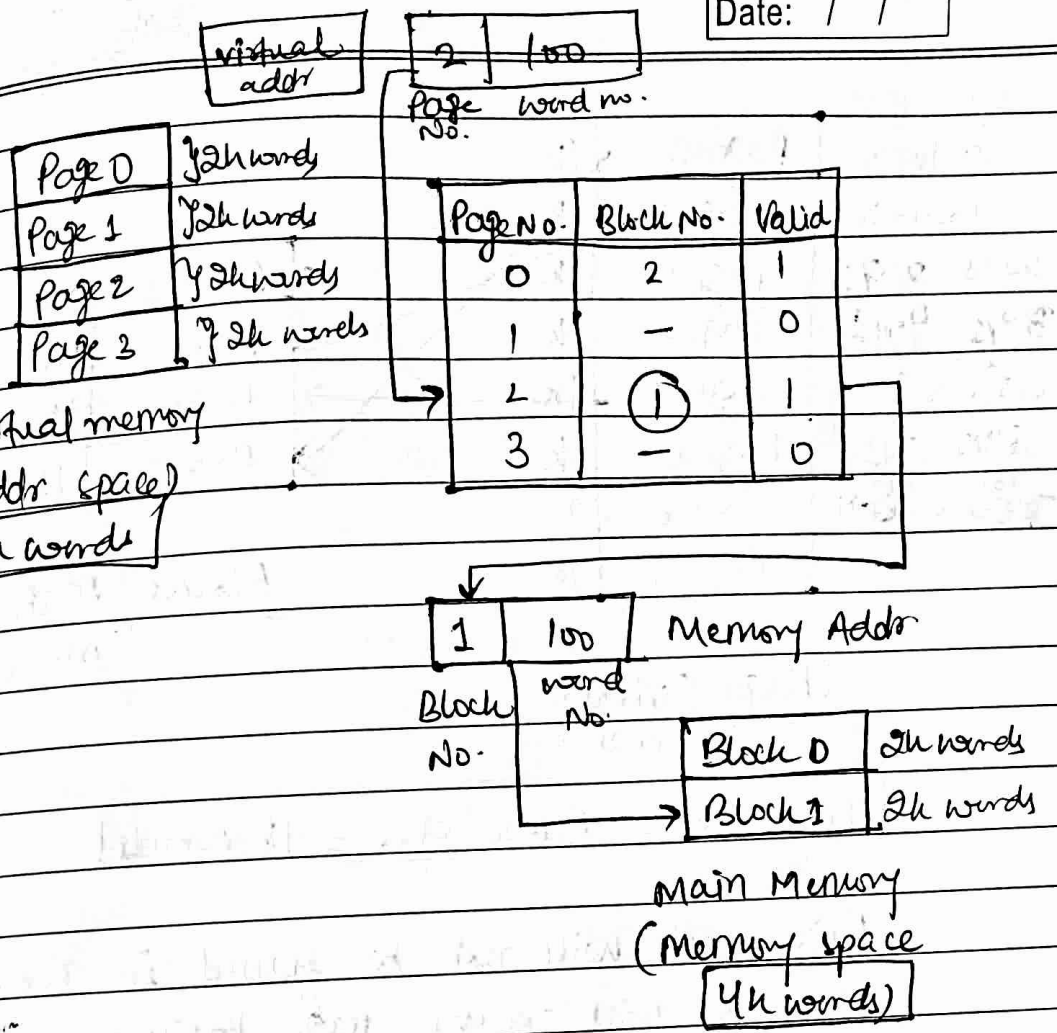
Sample of address mapping using paging

Page size = 2k words = Block size

No. of pages = $\frac{8k}{2k} = 4$

Planned:

Page No:
Date: / /

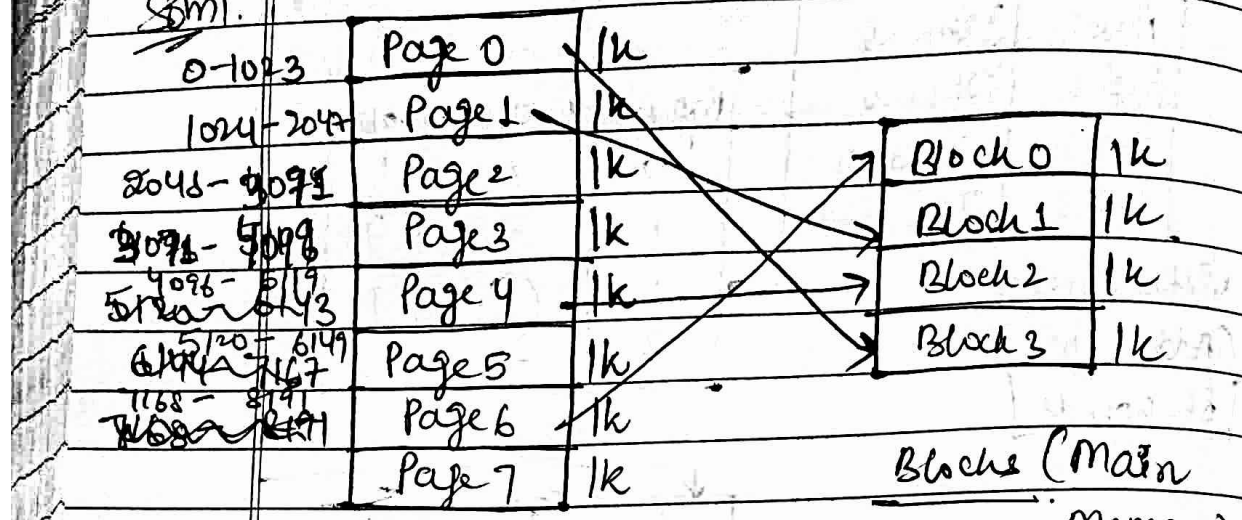


Q A virtual memory has page size of 1k words. There are eight pages and four blocks. The associative memory page table contains the following entries. Make a list of all virtual addresses (in decimal) that will cause a page fault.

Page	Block
0	3
1	1
4	2
6	0

Page No:
Date: / /

Sol.



Page size = Block size = 1k words

Pages that will not be found in main memory will causes page fault.

Page No.	Virtual addr (in decimal)
2	2048 - 3071
3	3072 - 4095
5	5120 - 6143
7	7168 - 8191

Segmented Paging :- In this, the virtual memory is first divided into segments, which are further divided into pages, pages are further divided into words.

Segments :- ① Can be subroutine, array of data, table of symbols, Stack or main program.

Page No:
Date: / /

② Each segment is a group of pages.
③ Each page is a group of words.

Page 0	Segment 0
Page 1	
Page 0	
Page 1	Segment 1
Page 2	
Page 3	
Page 0	Segment 2
Page 1	

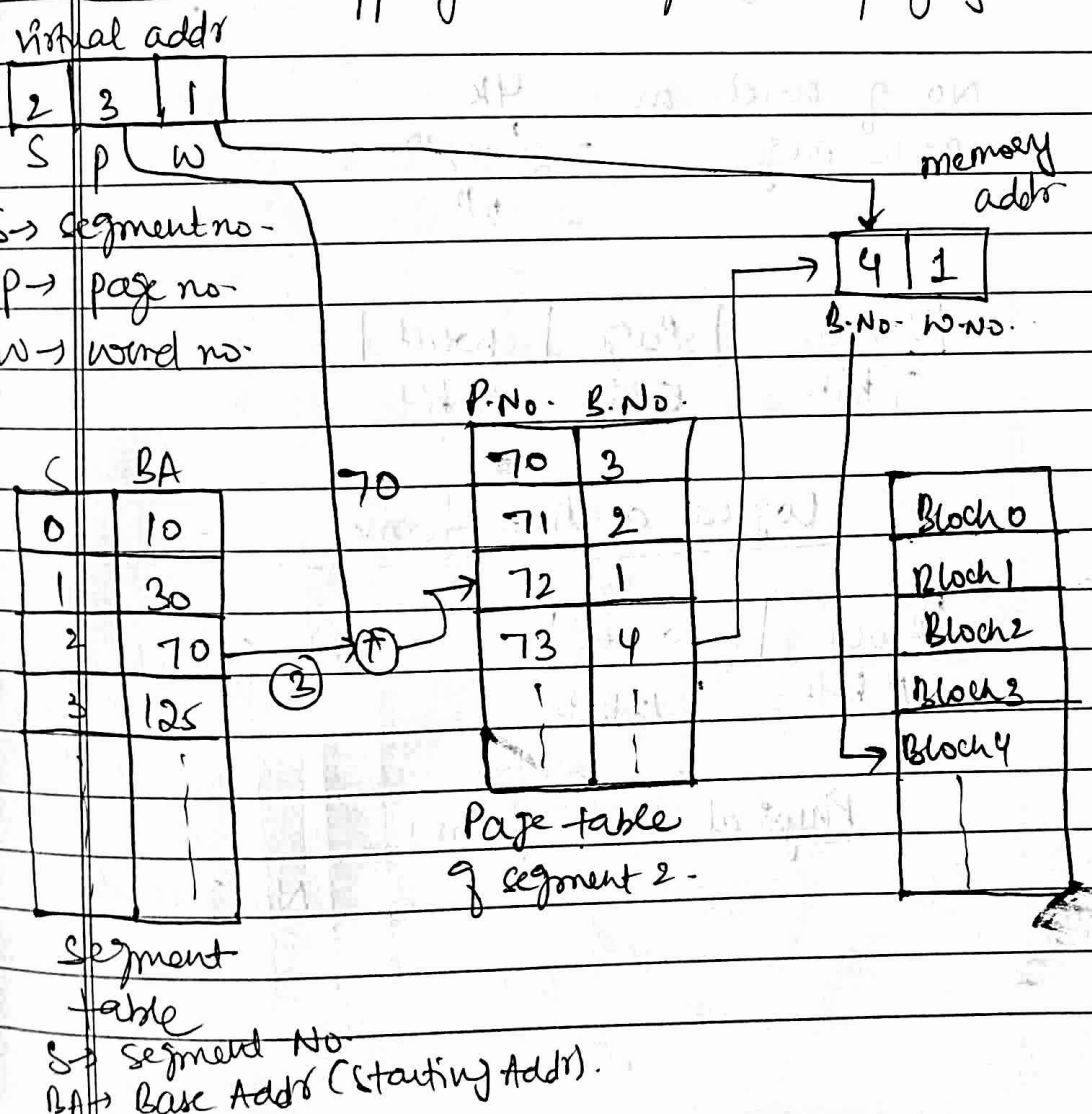
Virtual memory (Addr space).

Block 0
Block 1
Block 2
Block 3

Main Memory (Memory space)

Page size = Block size

Address mapping in segmented paging



Page No:
Date: / /

Virtual Addr	Segment No.	Page No.	word No.

Memory Addr	Block No.	Word No.

Q) The logical addr space in a computer system consists of 128 segments. Each segment consists of 32 pages of 4k words in each. Physical memory consists of 4k blocks of 4k words in each. Formulate the logical and physical address formats.

Soln) No. of segments = 128
$= 2^7$

No. of pages in each segment = 32
$= 2^5$

No. of words in each page = 4k
$= 2^2 \times 2^{10}$
$= 2^{12}$

Segment	Page	word
7 bits	5 bits	12 bits

Logical address format

Block	word
12 bits	12 bits

No. of blocks = 4k
$= 2^2 \times 2^{10}$
$= 2^{12}$

Physical Addr format

No. of words = 4k
$= 2^2 \times 2^{10}$
$= 2^{12}$

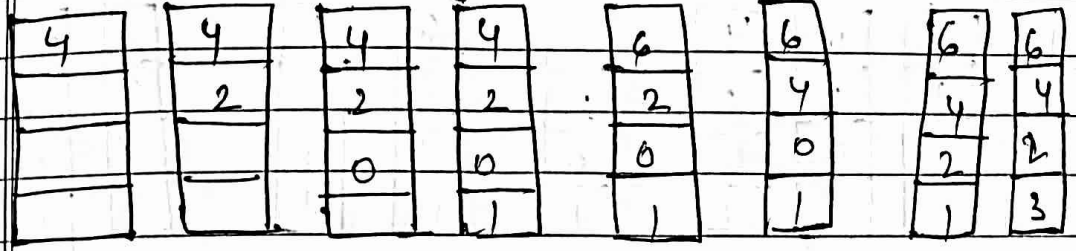
Page No:
Date: / /

Page Replacement Algorithms in memory organisation

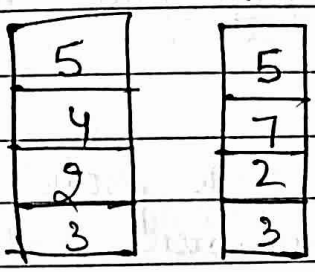
1) FIFO (First in First out) :- Page comes first in memory will be moved out first.

Page Reference :-

4 2 0 1 2 6 1 4 0 1 0 2 3



5 7



No. of page faults = 10

Assume :-

(a) Addr space = 8k words

↳ No. of pages = $\frac{\text{Addr space}}{\text{page size}} = \frac{8k}{1k} = 8$ pages

(b) Memory space = 4k words

Page size = block size = 1k words

↳ No. of blocks = $\frac{\text{memory space}}{\text{block size}} = \frac{4k}{1k} = 4$ blocks

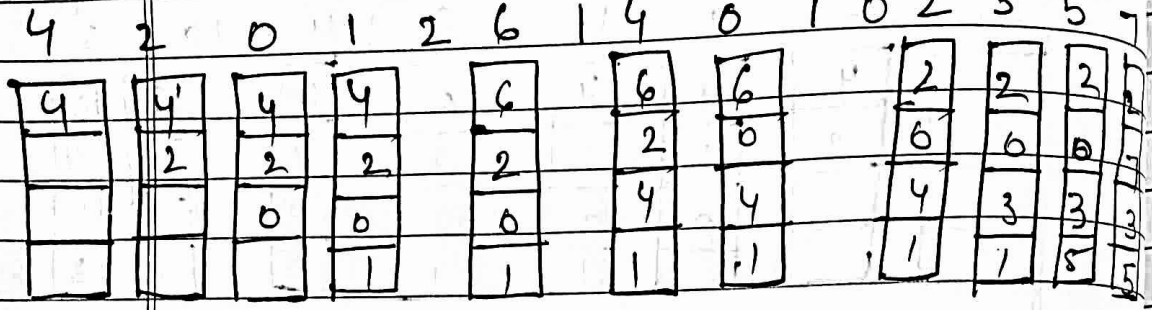
Page No:
Date: / /

2) LRU (least Recently Used) Page Replacement Algorithm.

Replace the page that has been used least recently.

Page Reference :-

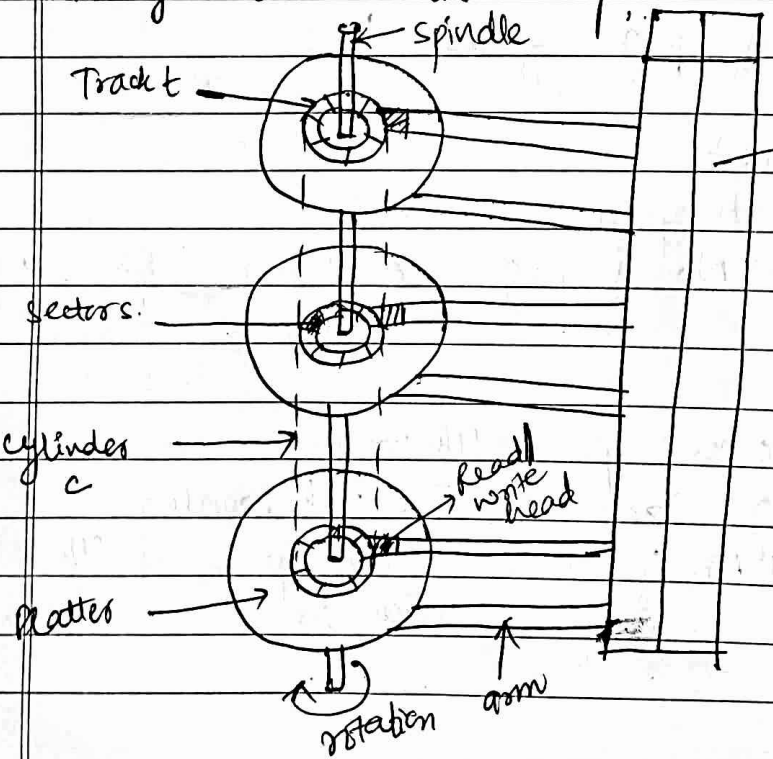
4 2 0 1 2 6 1 4 0 1 0 2 3 5



No. of blocks = 4

No. of page faults = 11

Magnetic Disk :- Magnetic disk provide a large amount of secondary storage to modern computer



Planned:

Page No:
Date: / /

Platter :- A round surface (disk) containing magnetic coating.

Track :- A circle on the disk surface containing data.

Cylinder :- Set of tracks accessed simultaneously from read/write head.

Sector :- A portion of track on the disk surface. Each sector stores same no. of bytes. So, the density of bit is more in sectors near the centre of the disk.

Block :- The smallest unit of data that can be written or read from the disk.

Read/write head :- A transducer attached to an arm for reading/writing data to/from the disk surface.

Arm Assembly :- A mechanical unit holding all heads and arms.

Drive Motor :- The drive (AC) motor rotates the platters at a fixed speed (eg: 3600 rpm)

Head motion :- A mechanism is required to move the head assembly in/out.

Seek Time :- Time taken by R/W head to move to correct track.

Page No:
Date: / /

Sol: Average latency time = $\frac{1}{2 \times \text{rotational speed}}$

$= \frac{1}{2 \times 7200 \text{ rpm}}$

$= \frac{1}{2 \times 120 \text{ rps}} = \frac{1}{240}$

$= 0.00416 \text{ sec}$

$= 4.16 \text{ ms}$.

Data transfer rate = $\frac{\text{No. of rotations}}{\text{sec}} \times \text{track} * \frac{\text{capacity}}{\text{No. of R/W head}}$

$= 120 \times 30 \times 0.5 \times 1$

$= 1800 \text{ KB/sec}$

$= \frac{1800}{2^{10}} = \frac{1800}{1024} = \boxed{1.76 \text{ MB/s}}$

Q2 What is the average access time for transferring 512 bytes of data with following specifications.

(i) Average seek time = 5ms
(ii) Disk rotation = 6000 rpm
(iii) Data rate = 40 KB/sec
(iv) Controller overhead = 0.1ms

Sol: Average access time = 5ms

Average latency time = $\frac{1}{2 * 6000 \text{ rpm}} = \frac{1}{2 * \frac{6000}{60} \text{ rps}}$

$= \frac{1 \text{ s}}{2 \text{ v}} = 0.005 \text{ s}$

$= 5 \text{ ms}$.

Page No:
Date: / /

Data transfer time = $\frac{512\ B}{40\ KB} = \frac{2^9\ B}{40 \times 2^{10}}$

$= \frac{1}{4 \times 2} = \frac{1}{8 \times 2}$

$= 0.0125\ sec = 12.5\ msec$

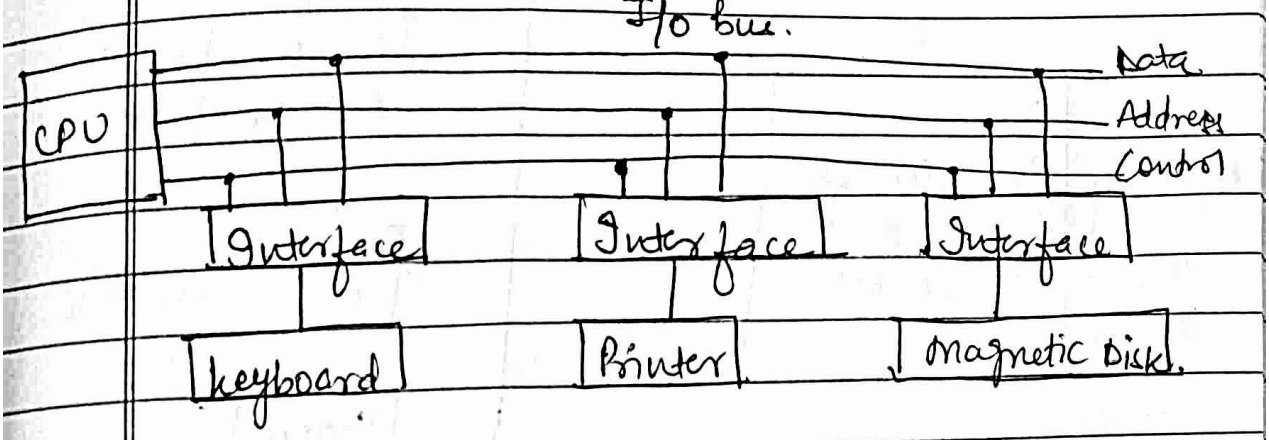
Average access time :- $5 + 5 + 12.5 + 0.1 + 0$

$= 22.6\ msec$

Unit-5 Input-Output

Page No:
Date: / /

I/O Interface:- It is a hardware circuit used between CPU and I/O devices to supervise and synchronise all input and output transfer.



Requirement (Need of I/O Interface)

* I/O devices are electromechanical and electro-magnetic while CPU & memory are electronic devices. so they have different signals is required.

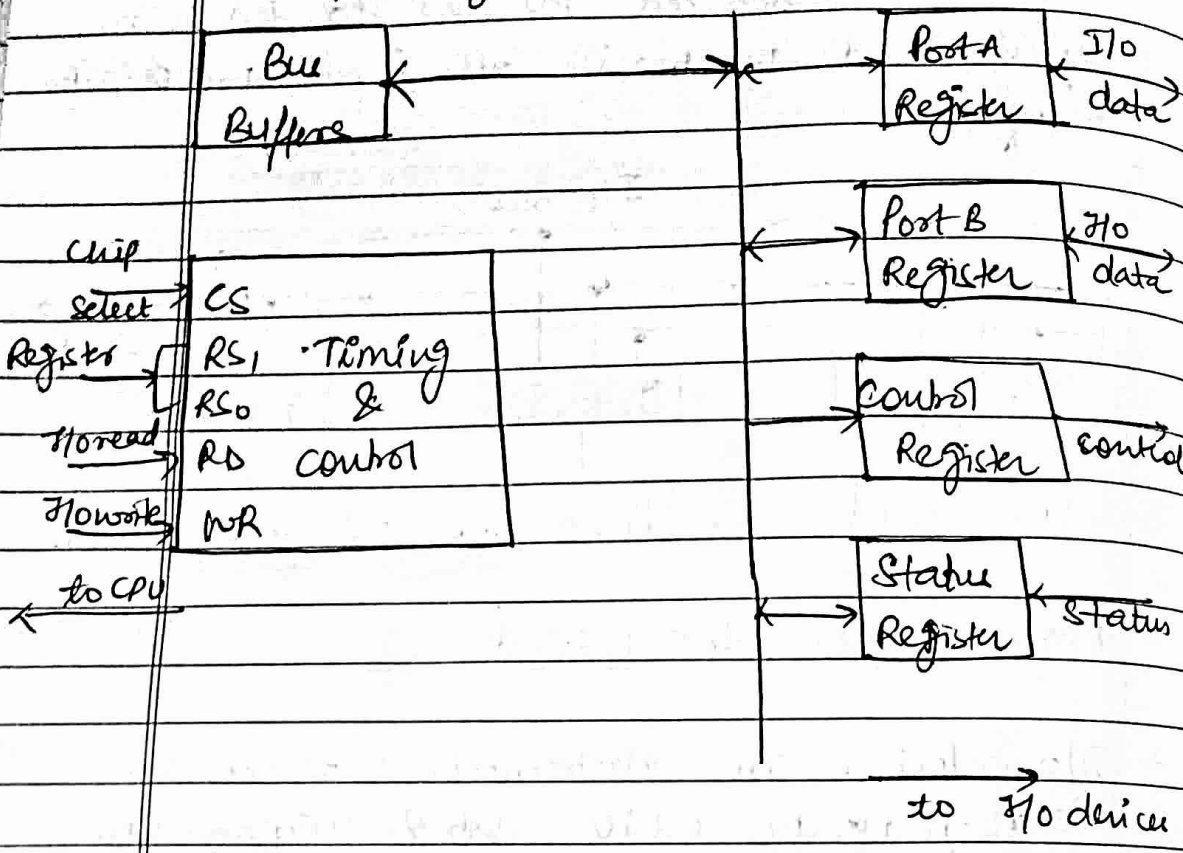
* Data transfer rate of CPU is faster as compared to I/O devices. So, synchronisation is needed.

* Data format of I/O devices are different from word format of CPU and memory.

* Operating modes of I/O devices are different from each other and so each I/O device must be controlled so that it will not disturb the operation of other I/O devices to CPU.

Page No:
Date: / /

Example of I/O Interface



CS	RS_1	RS_0	Register select
0	x	x	None
1	0	0	Port A Register
1	0	1	Port B Register
1	1	0	Control Register
1	1	1	Status Register.

I/O commands

* I/O commands are the instructions given by the CPU to I/O interface or I/O devices to perform specific operations

Planned:

Page No:
Date: / /

Types of I/O commands

1) Control command :-

2) Status command :- This command is given by CPU to check status of I/O devices.

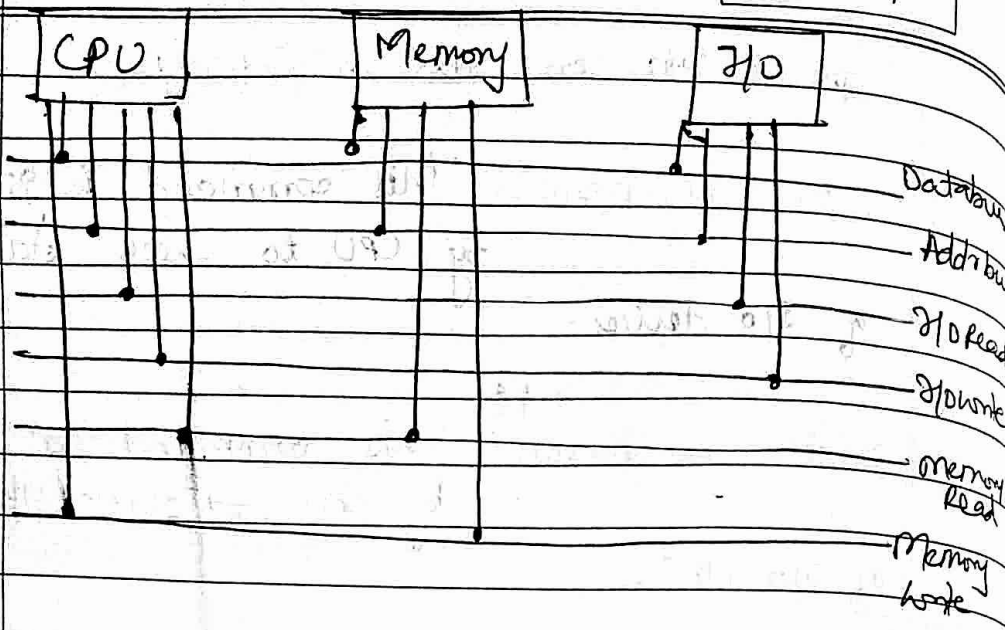
3) Data Input command :- This command is given by CPU to read any data from input devices.

4) Data output command :- This command is given by CPU to write any data to output devices.

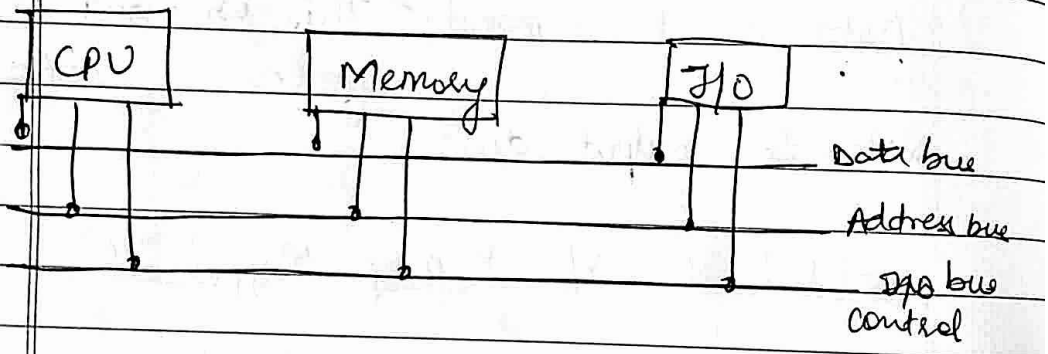
Isolated I/O v/s Memory Mapped I/O.

Isolated I/O :- In Isolated I/O, the address and data bus are common, but control bus are different. For memory, it has "memory read" and "memory write" control lines. For I/O device it has seperate "I/O read" "I/O write" control line.

Page No:
Date: / /



Memory Mapped I/O :- Address, data and control lines bus are common for memory & I/O devices.



Asynchronous Data Transfer

* A data transfer is said to be asynchronous data transfer, when two independent units are their own internal clock. These independent units does not make use of common clock.

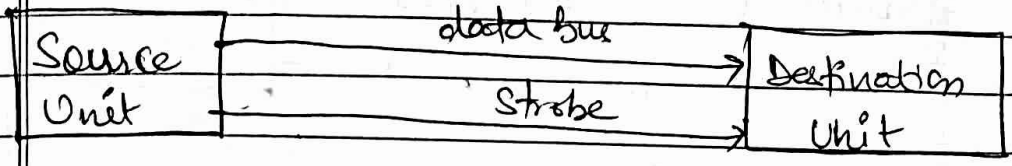
Planned.

Page No:
Date: / /

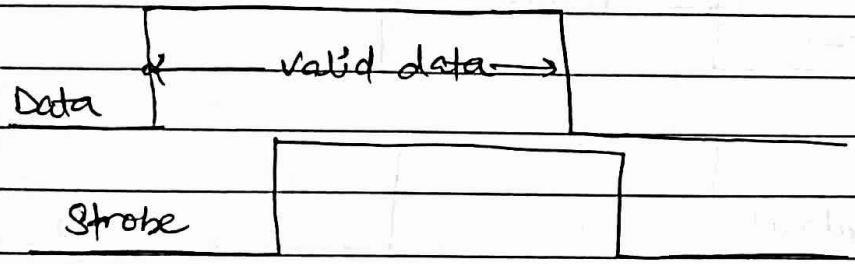
Types of Asynchronous data transfer

(1) Strobe control :-

(i) Source initiated strobe

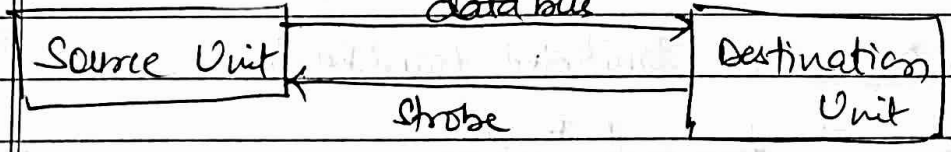


(a) Block diagram.

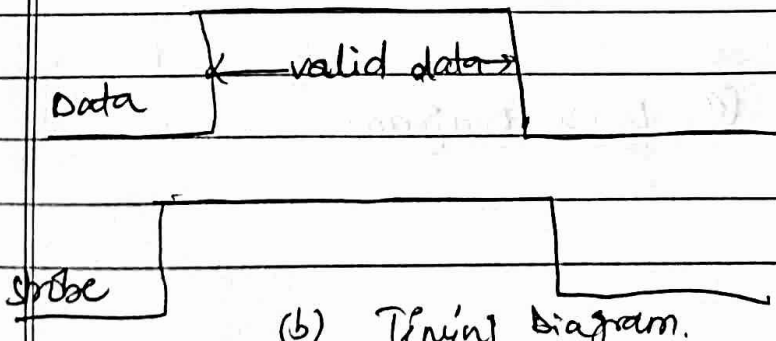


(b) Timing diagram.

(ii) Destination initiated strobe



(a) Block diagram.

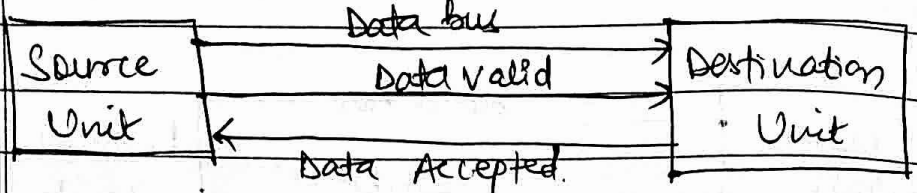


(b) Timing diagram.

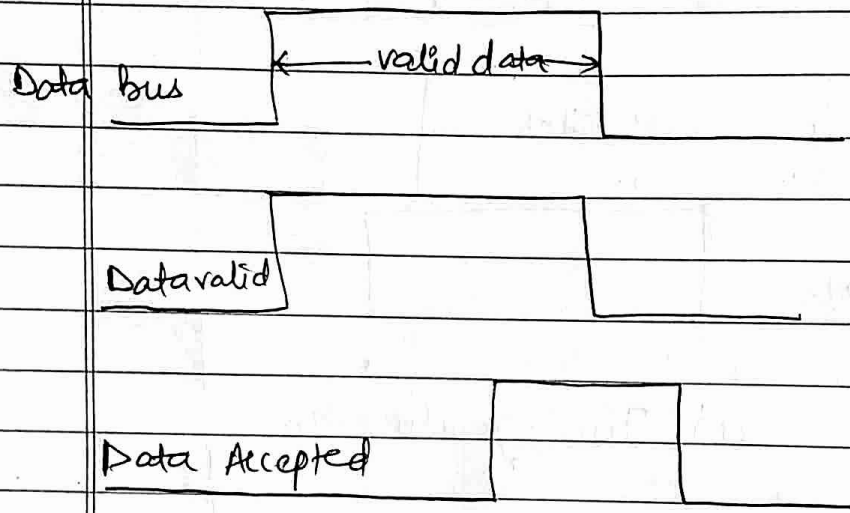
Page No:
Date: / /

2) Handshaking Method :- (Strobe control + Acknowledgement sig

i) Source- Initiated handshaking

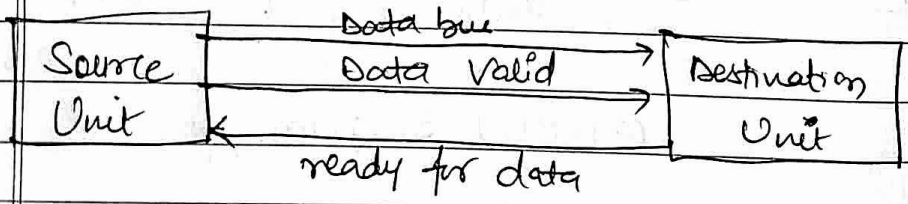


(a) Block Diagram



(b) Timing Diagram

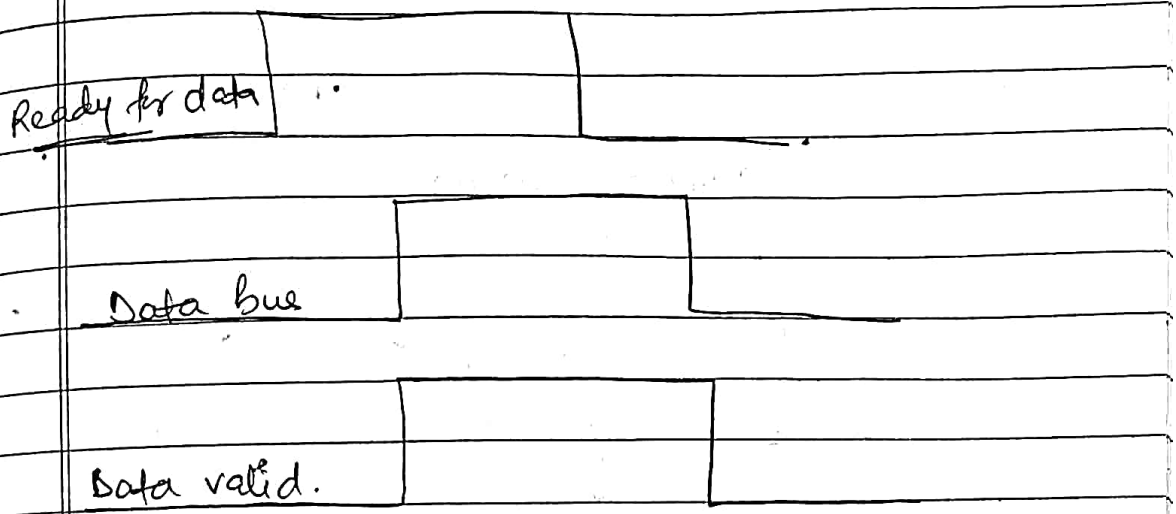
ii) Destination Initiated handshaking.



(a) Block Diagram

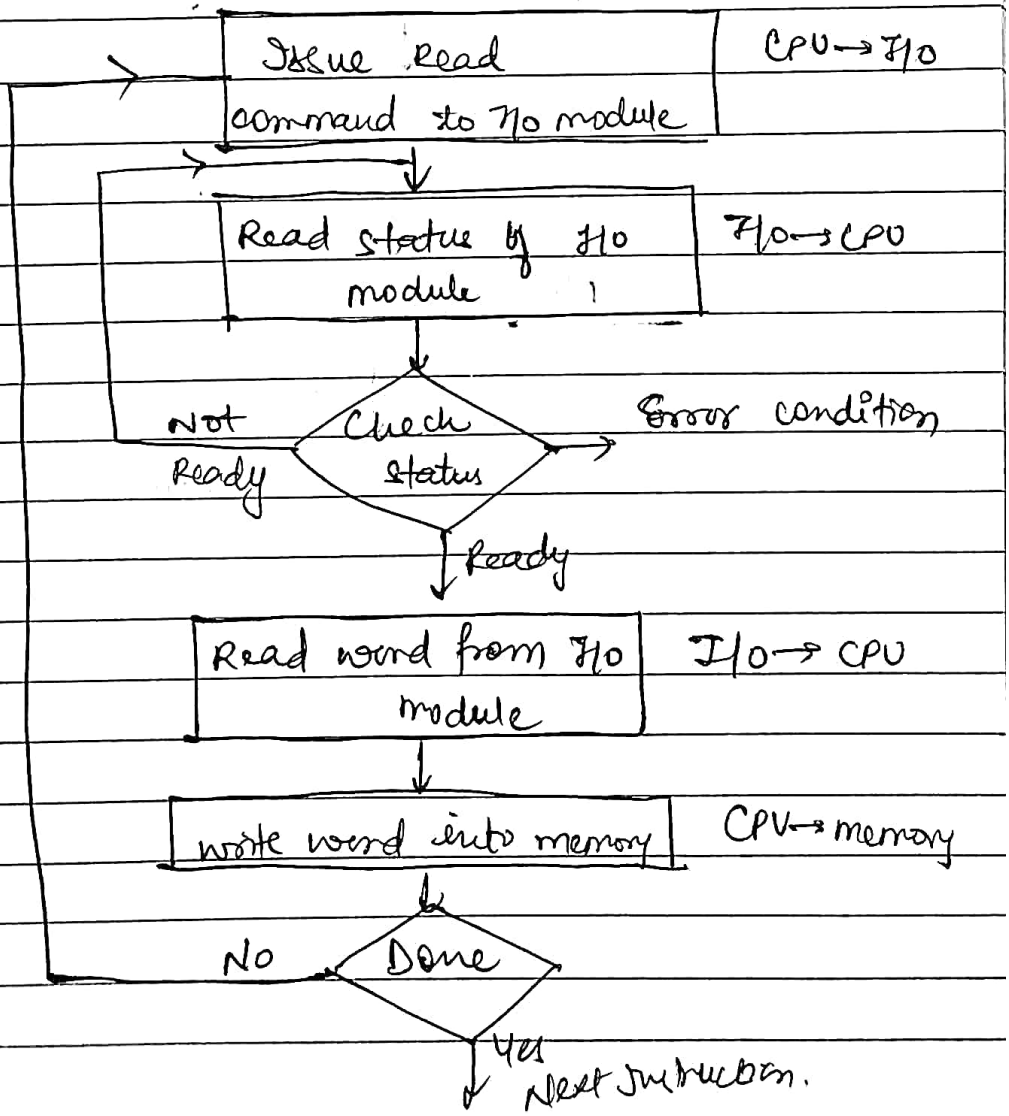
Planned:

Page No:
Date: / /



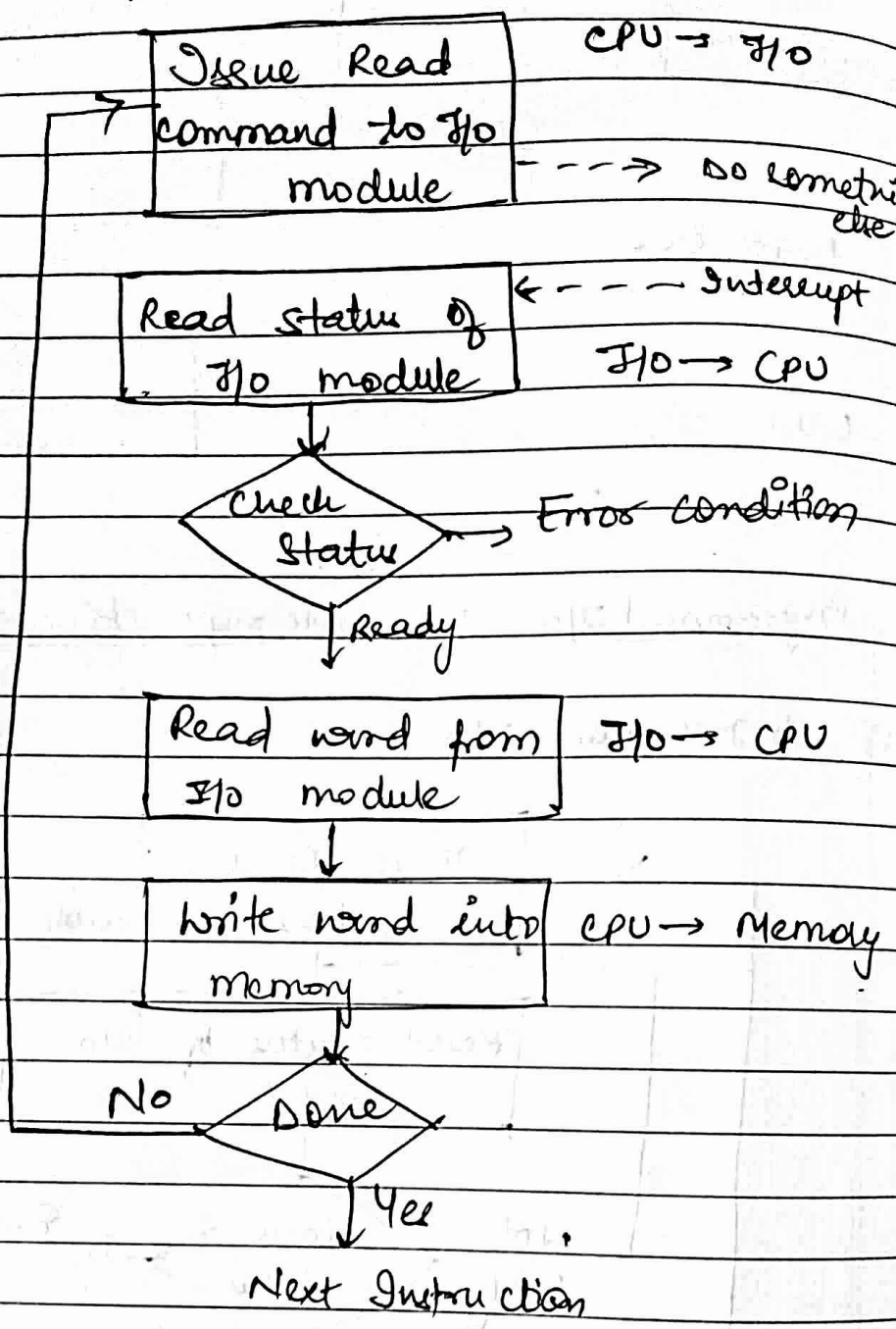
Programmed I/O VS Interrupt driven I/O.

(1) Programmed I/O.



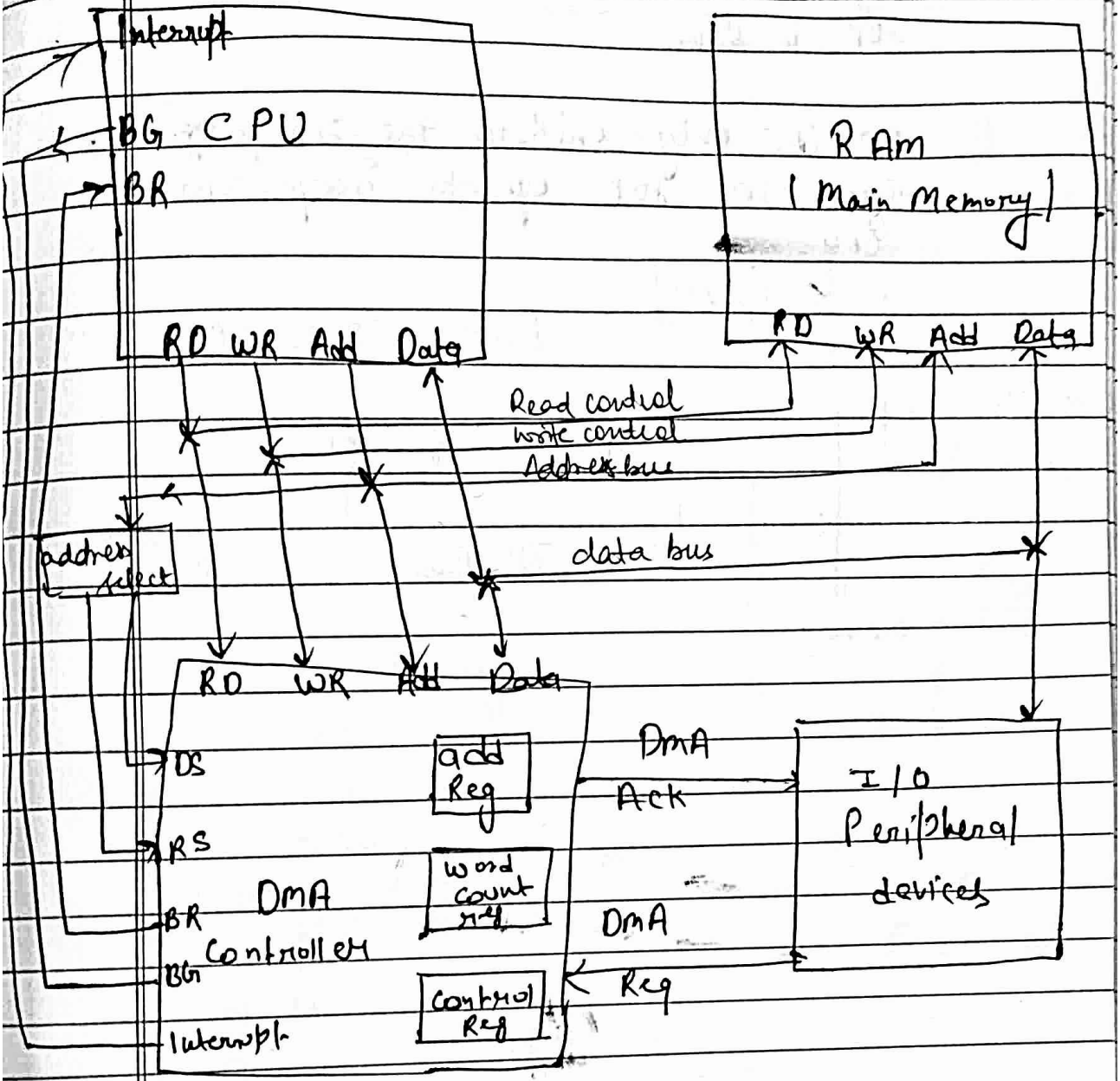
Page No:
Date: / /

2) Interrupt driven I/O.



Page No:
Date: / /

DMA (Direct Memory Access)



Input-Output Processor (IOP)

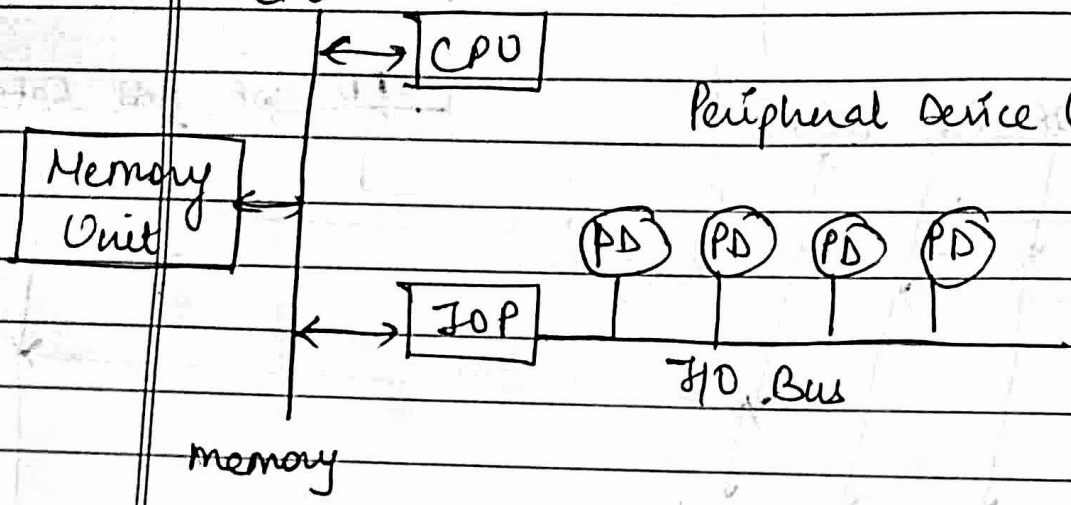
① The IOP is similar to CPU except that it is designed to handle only I/O processing.

② The IOP fetches & executes I/O instructions to facilitates I/O transfer. If required, It can perform all the functions of CPU.

Page No:
Date: / /

(3) Both CPU and IOP exists in the system however CPU is the master while IOP is slave.

(4) The CPU only initiates the I/O program, after that IOP operates independent of the CPU.



Page No:
Date: / /

CPU - IOP Communication