15/7/22

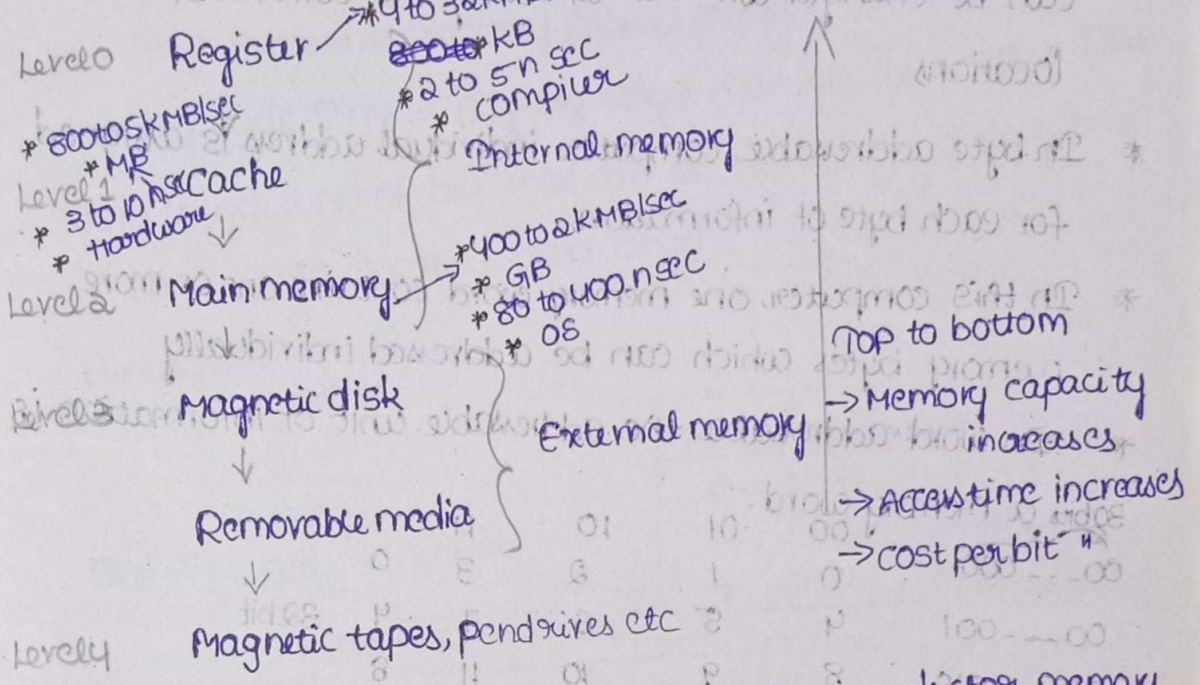
Memory Hierarchy

* This design is divided into 2 types

1. External memory or secondary memory:
   Comprising of Magnetic Disk, Optical Disk, Magnetic type i.e. peripheral storage Devices which are accessible by the processor via I/O module.

2. Internal memory or primary memory:
   Comprising of Main memory, cache memory & CPU registers
   This is directly accessible by the processor

Memory Hierarchy in computer system:



The main memory of a computer is semiconductor memory
The main memory unit of a computer basically consists of two kinds of memory

1. RAM 2. ROM

Random Access memories are volatile in memory
Read only memories are non-volatile in nature i.e., storage is permanent

Types of ROM:

1. PROM: Programmable Read only memory

   It can be programmed once as per user requirements

2. EPROM: Erasable programmable Read Only memory

   The contents of memory can be erased and stored new data into memory

3. EEPROM:

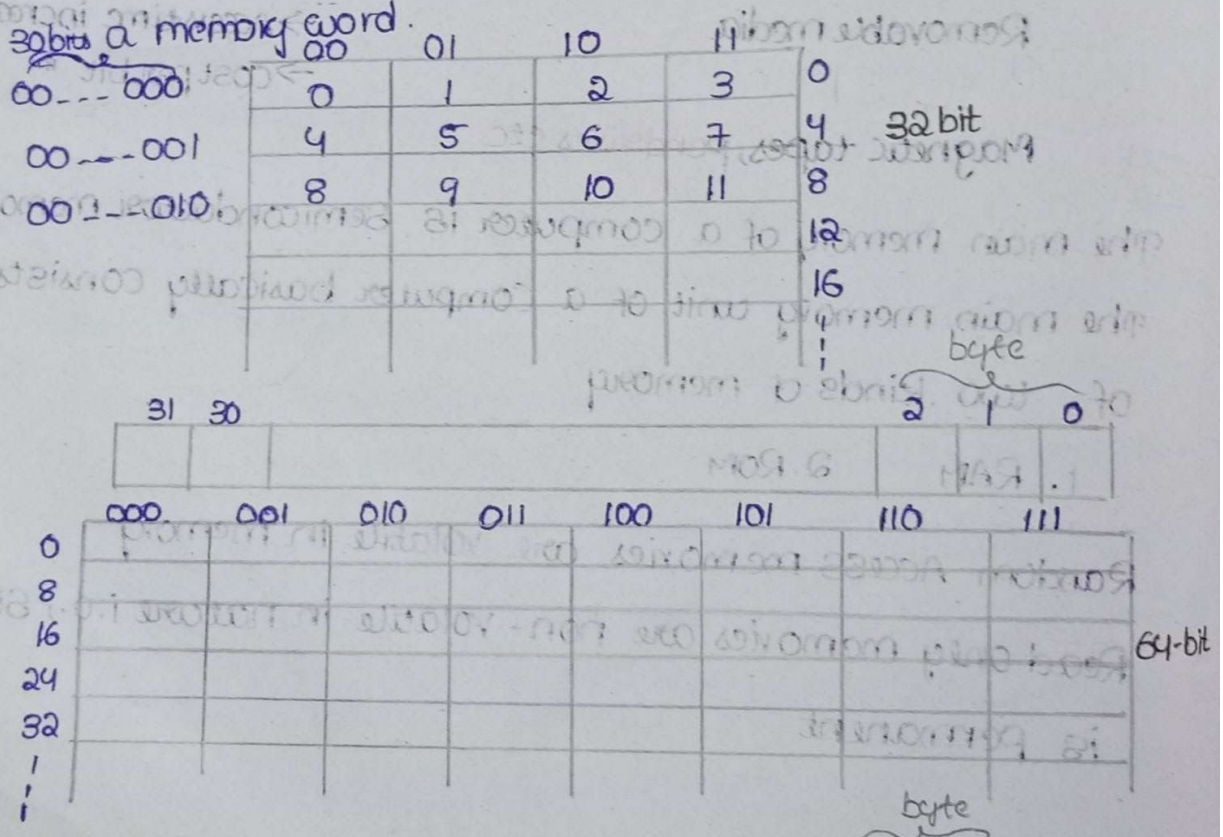
   Electrically Erasable programmable Read only memory

   In this type of memory the contents of particular location can be changed without affecting the contents of other locations

* In byte addressable computer individual address is assigned for each byte of information

* In this computer one memory word contains one or more memory bytes which can be addressed individually

* In word addressable the addressable unit of information is a memory word.



If MAR is k-bit long then total addressable memory locations will be 2^k

If MDR is n-bit long then n-bit of data is transferred in one memory cycle

Memory Access Time: Time that elapses btw the initiation of an operation and the completion of the operation is called Memory Access Time

Ex: The time btw Read and MFC

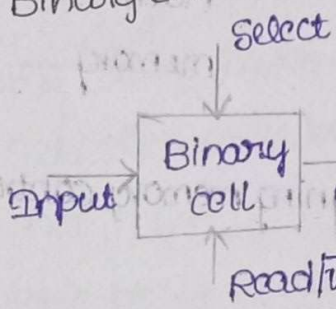
Memory cycle Time: The minimum time delay btw the initialization of two independent memory operations

Ex: Two Successive memory Read operation

Memory cycle Time is slightly larger than memory Access Time

Word length: No. of bits that can be transferred during one memory cycle is called word length

Binary cell:



Select	R/$\bar{w}$	OP
0	x	x
1	0	write
1	1	Read

SRAM	DRAM
* Static Random Access Memory	* Dynamic Random Access memory
* Static RAM is volatile	* Dynamic RAM is volatile.
* Memory cell is complex & larger	* Memory cell is simpler & smaller
* It is less dense i.e., Packing density is low	* It is more dense i.e, packing density is high (More cells for unique area)
* More expensive	* Less Expensive
* Faster access	* Slower access

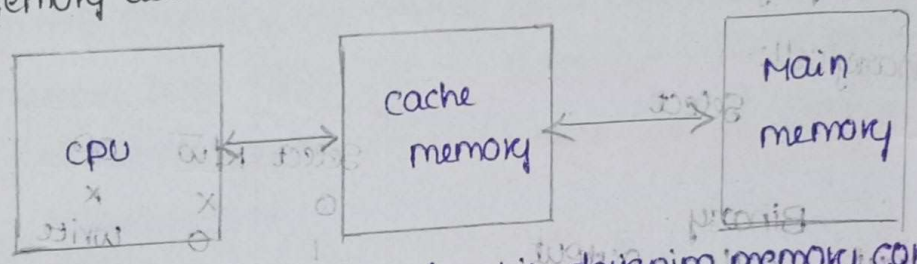
* DRAM requires the supporting refresh circuitary
* For larger memories, fixed cost of the refreshed circuitary is more than compensated for by the less cost of the DRAM cells

Cache Memory

Cache memory:

Analysis of large number of programs has shown that, a no. of instruction are executed repeatedly. This may be in the form of simple loops, nested loops or a few procedures that repeatedly call each other. It is observed that many instructions in each of the few localised areas of the program are repeatedly executed. This phenomenon is referred to as locality of reference

Memory access control & Data path:



Few assumptions are made while designing memory control circuitary

* The CPU doesn't know explicitly about the existence of cache
* The CPU simply makes read and write requests. The nature of these two operations are same whether cache is present or not
* The address generated by CPU always refers to the location of main memory

→ When the cache is full and a memory is referred that is not in the cache, a decision must be made which block should be removed from the cache to create space to

bring the new block to the cache that contains referenced word

→ Replacement algorithms are used to make the proper selection of the block must be replaced by the new one

→ When the write request is received from the CPU, there are two ways that system can proceed.

* In the first case cache location and the main memory location are updated Simultaneously. This is called write through protocol

* The alternative is to update cache location only, during replacement time, the cache block will be return back to main memory. If there is no new write operation in the cache block, this is not required to write back in the main memory

This information can be kept with the help of an associated bit. This bit is set while there is a write operation in the cache block. During replacement, it checks this bit, if it set, then write back the cache block in main memory otherwise not

This bit is known as dirty bit

This is called write once protocol

Cache memory Mapping Techniques:

* The mapping functions are used to map a block of main memory to block of cache memory

* Three different mapping functions are available

1. Direct Mapping
2. Associative Mapping
3. Set Associative Mapping

Direct Mapping:

* A particular block of main memory can be brought to particular block of cache memory
* In this technique block k of main memory maps into block k modulo M of cache memory where M is the no. of blocks in the cache
* Since more than one main memory block is map on to a given cache block position contention may arise for that position. This situation may occurs even when the cache is not full
* contention is resolved by allowing the new block to over write the currently decided block

So replacement algorithm trivial

Ex: 1024 k - main memory

256 k - cache memory

Block size - 128 words

Set size - 16 blocks

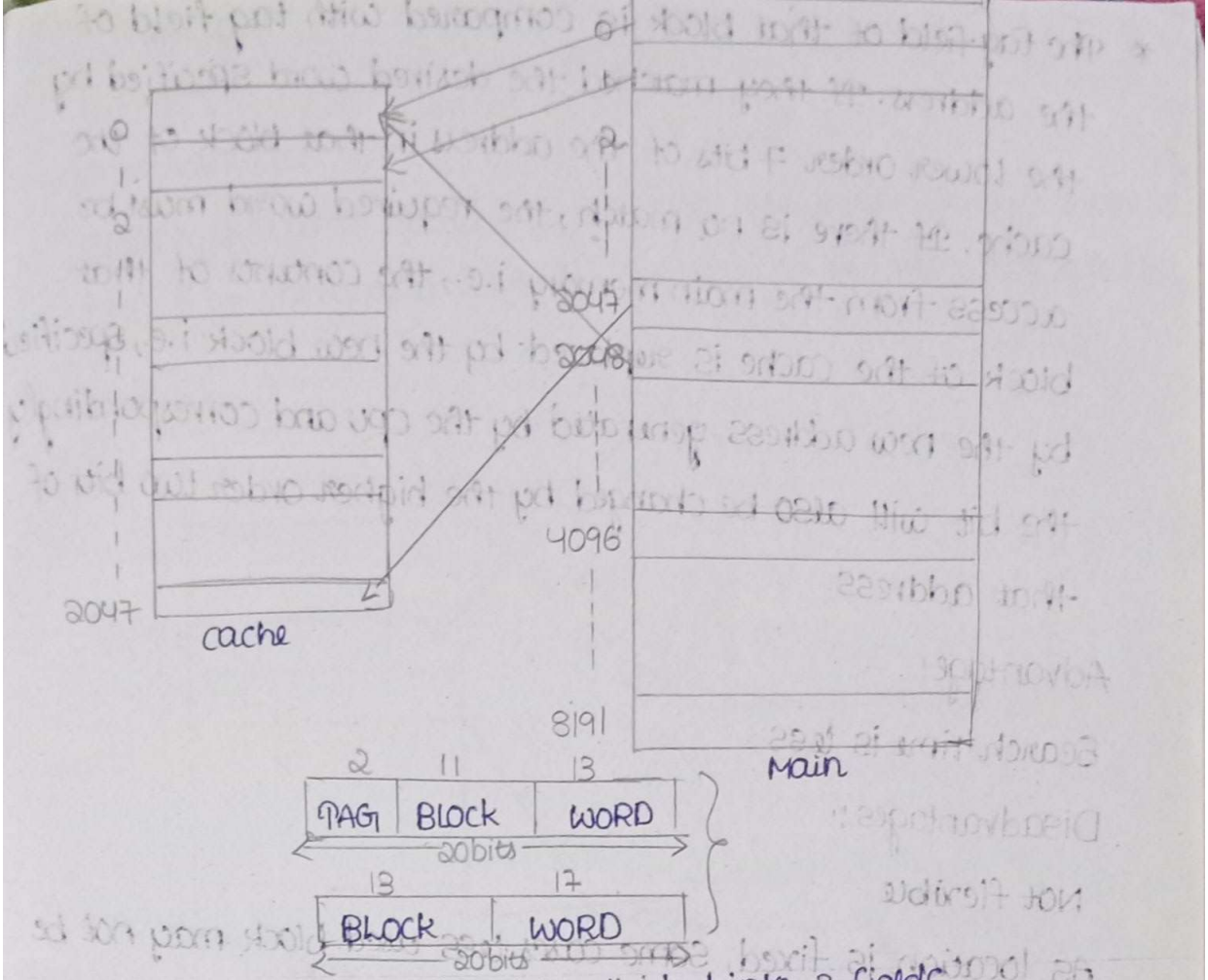
Sol: No. of blocks in cache memory $= \frac{256\,k}{128}$

$= \frac{2^8}{2^7} = 2^1 = 2$ blocks

No. of blocks in main memory $= \frac{2^{20}}{2^7}$

$= 2^{13}$

$= 8192$ blocks



* The main memory address is divided into 3 fields
* The field size depends upon memory capacity and the size of cache
* In the above example, the lower 7 bits of address is used to identify a word with a block
* Next 11 bits are used to identify a block out of 2048 block (which is the capacity of cache)

The remaining two bits are used as tag to identify the proper block of memory i.e, mapped to cache. when a new block is first brought into cache, the higher order two bits of main memory address are stored in two bit tag associated with its location in the cache. when CPU generates a memory request, the 11 bit address determines the corresponding cache block

* The tag field of that block is compared with tag field of the address. If they matched the desired word specified by the lower order 7 bits of the address in that block of the cache. If there is no match, the required word must be access from the main memory i.e., the contents of that block of the cache is replaced by the new block i.e., specified by the new address generated by the CPU and corresponding bit will also be changed by the higher order two bits of that address

Advantage:

Search time is less

Disadvantages:

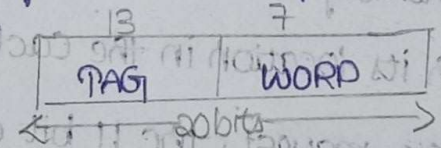
Not flexible

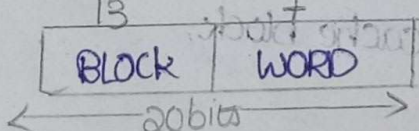
As location is fixed, some cases less used block may not be selected for replacement.

Associative Mapping:

In Associative mapping a main memory block can potentially reside in any cache block pattern. In this case the main memory block can be divided into two parts, lower order bits identify location of word with in block, high order bits identify the block.

In this example, 7 bits are used to identify a word within a block, and higher order 13 bits are used as tag bits to identify main memory block, which is resident in the cache



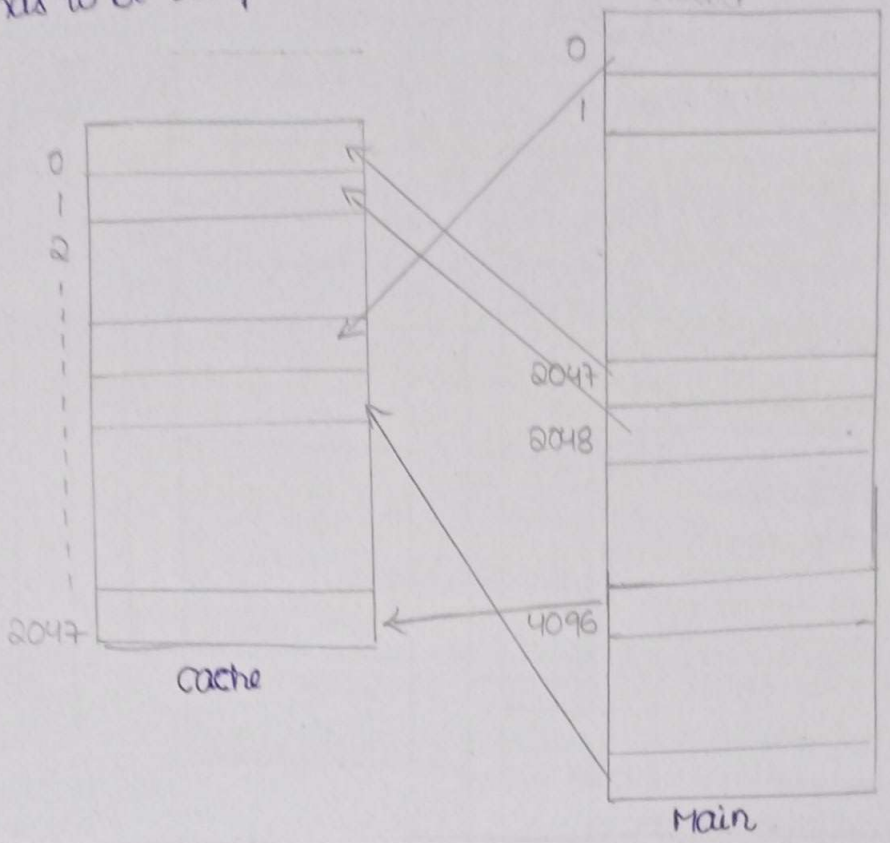


Advantages:

Flexible

Disadvantages:

Search overtime. Because tag field of the main memory address has to be compared with the tag field of all the cache blocks

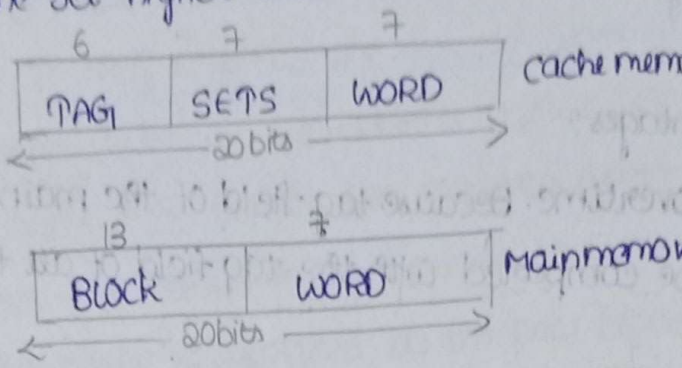


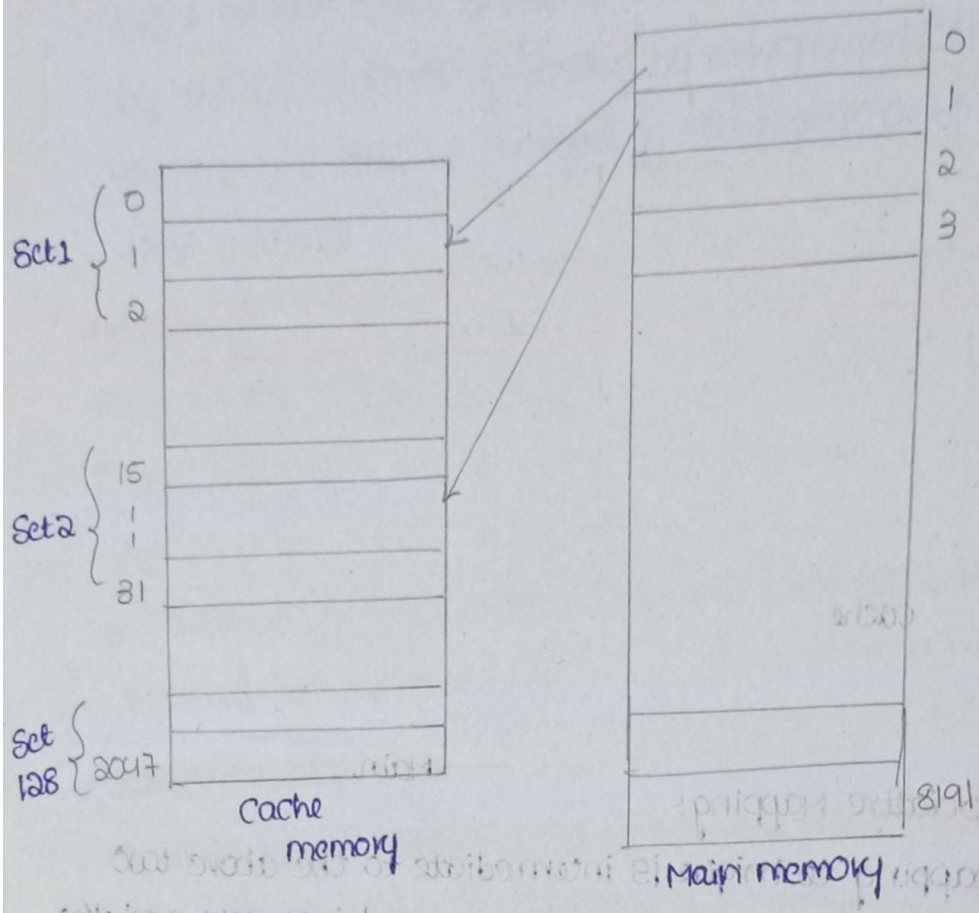
Set Associative Mapping:

This mapping technique is intermediate to the above two techniques. Blocks of the cache are grouped into sets and the mapping allows a block of main memory to reside in any block of the specific set. Therefore the flexibility of associativity mapping is reduced from full freedom to set of specific blocks. This also reduces the searching overhead because the restricted to no. of sets instead of no. of blocks, also contention problem of the direct mapping is caused by having few choices for block replacement

The main memory address is grouped into 3 bits. The lower order 7 bits are used to identify a word within a block. Since there are total 128 sets present, the next 7 bits are used

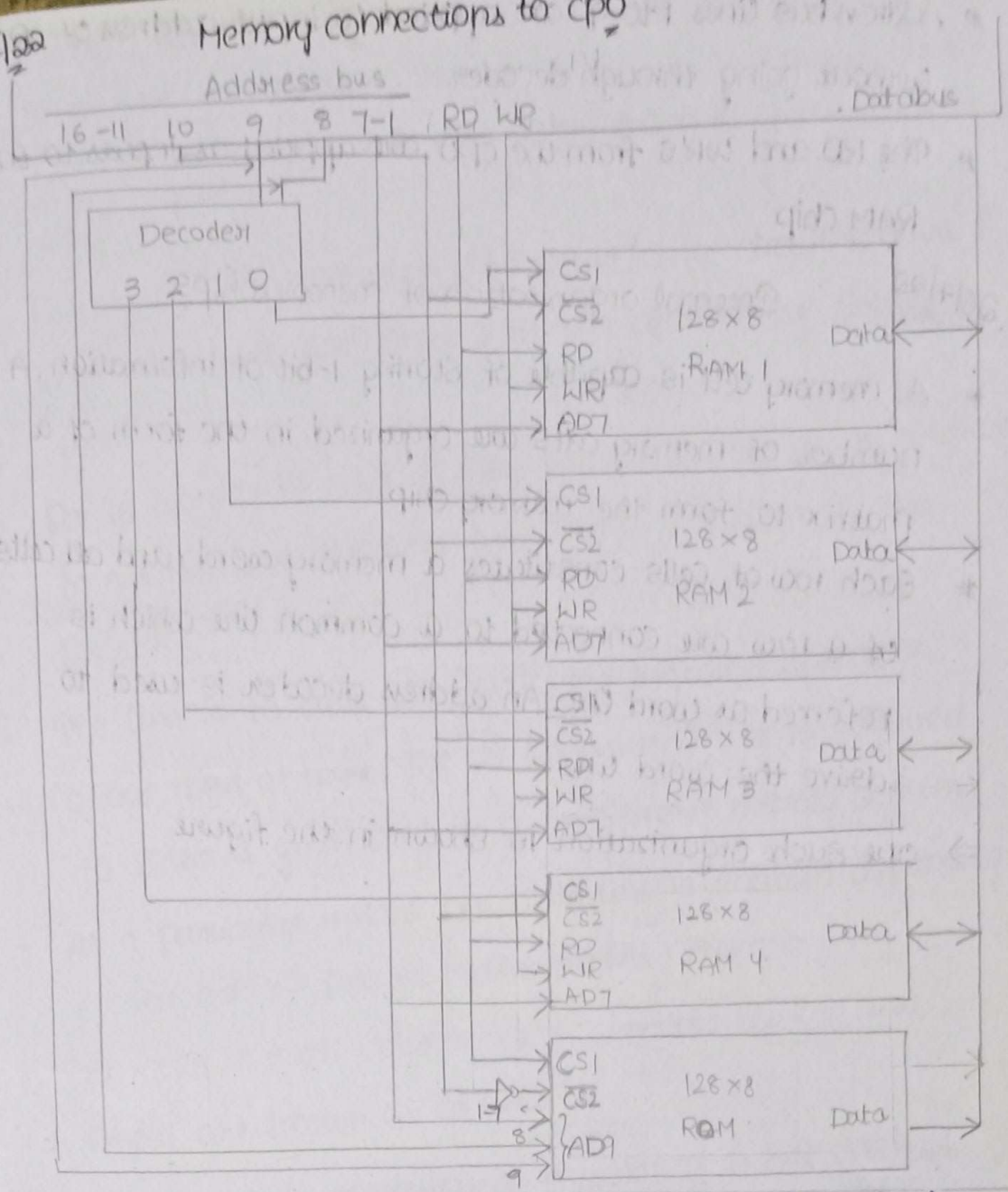
to identify the set higher order 6 bits are used as tag bits





26/7/22

Memory connections to CPU



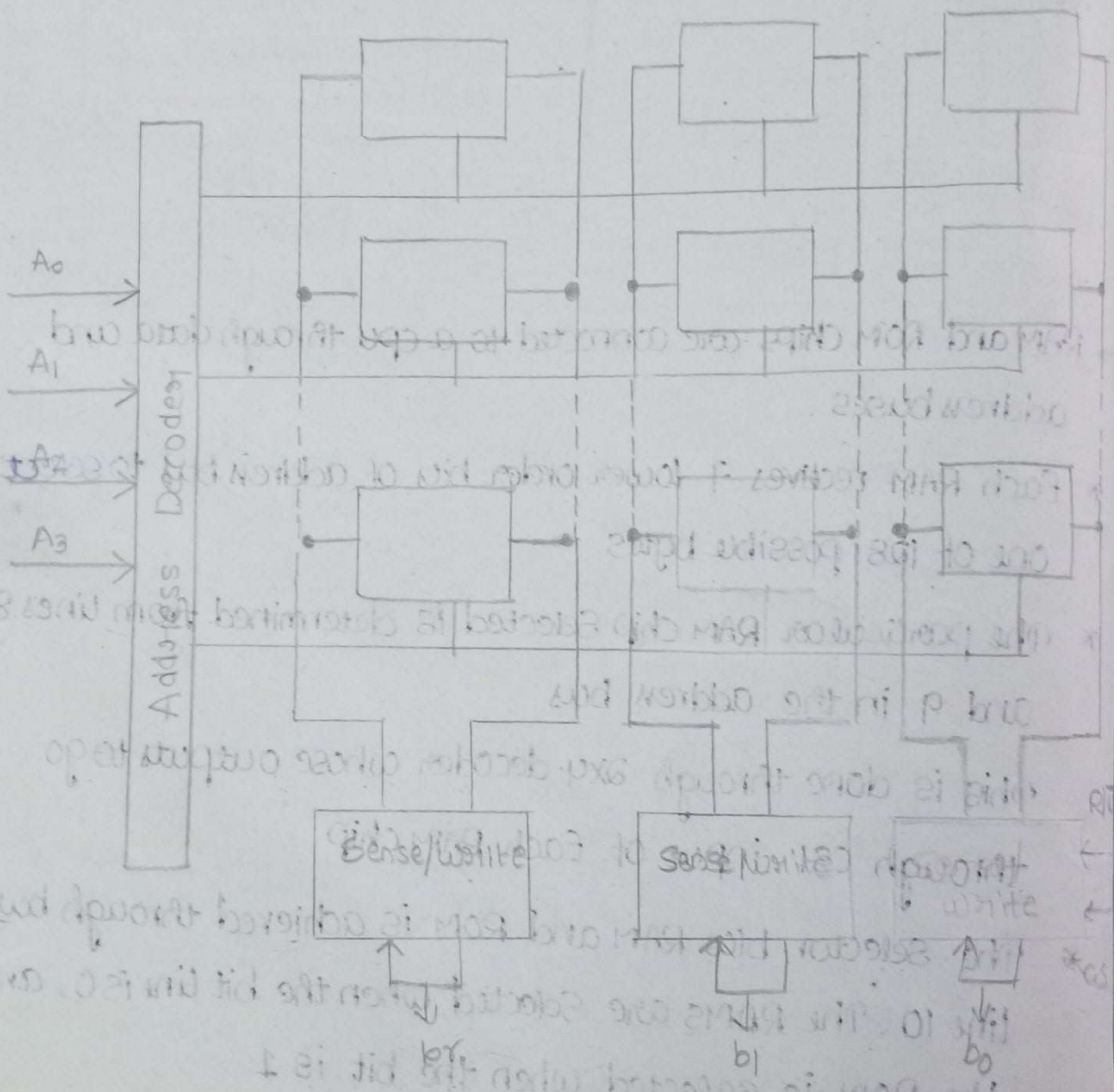
* RAM and ROM chips are connected to a cpu through data and address buses
* Each RAM receives 7 lower order bits of address bus to select one of 128 possible bytes
* The particular RAM chip selected is determined from lines 8 and 9 in the address bus
  This is done through 2x4 decoder whose outputs to go through CS1 input of each RAM chip
* The selector blw RAM and ROM is achieved through bus line 10. The RAMs are selected when the bit line is 0, and the ROM is selected when the bit is 1

* Address bus lines 1 to 9 are applied to input address of ROM without going through decoder
* The RD and WR's from the CPU are applied as inputs to Each RAM chip

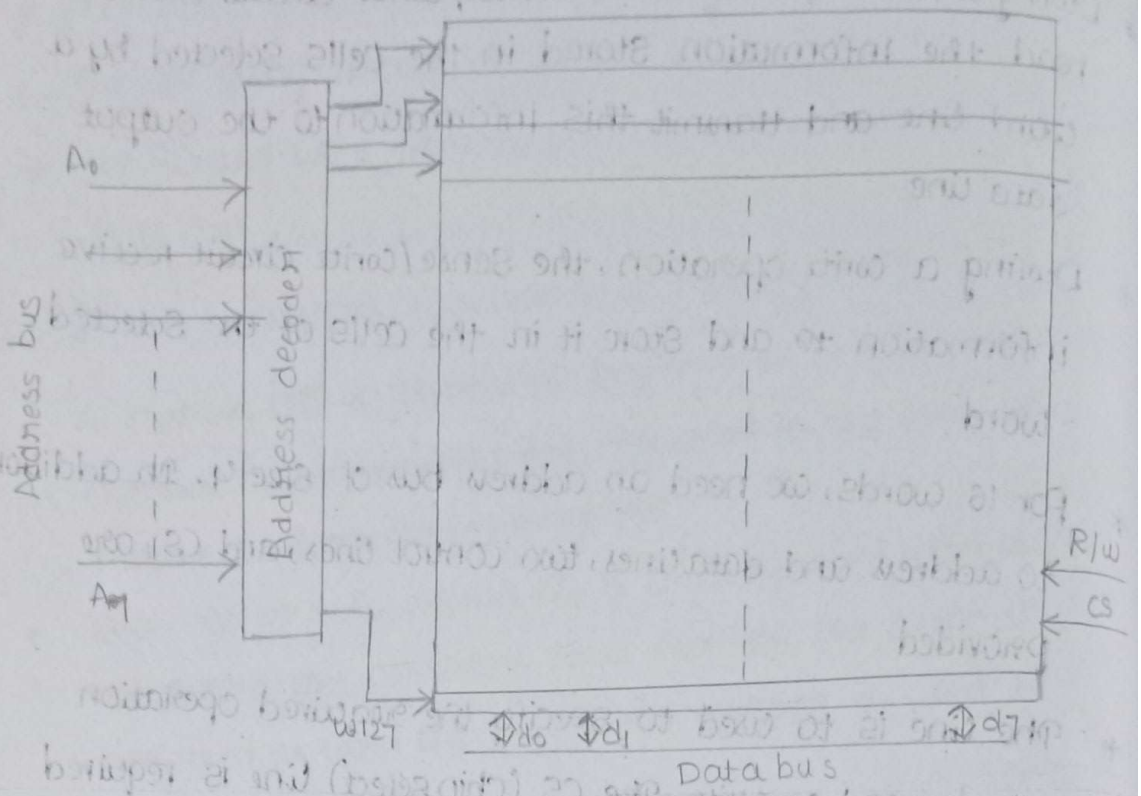
26/7/22 Internal organization of memory chips

* A memory cell is capable of storing 1-bit of information. A number of memory cells are organized in the form of a matrix to form the memory chip
* Each row of cells constitutes a memory word, and all cells of a row are connected to a common line which is referred as word line. An address decoder is used to drive the word line

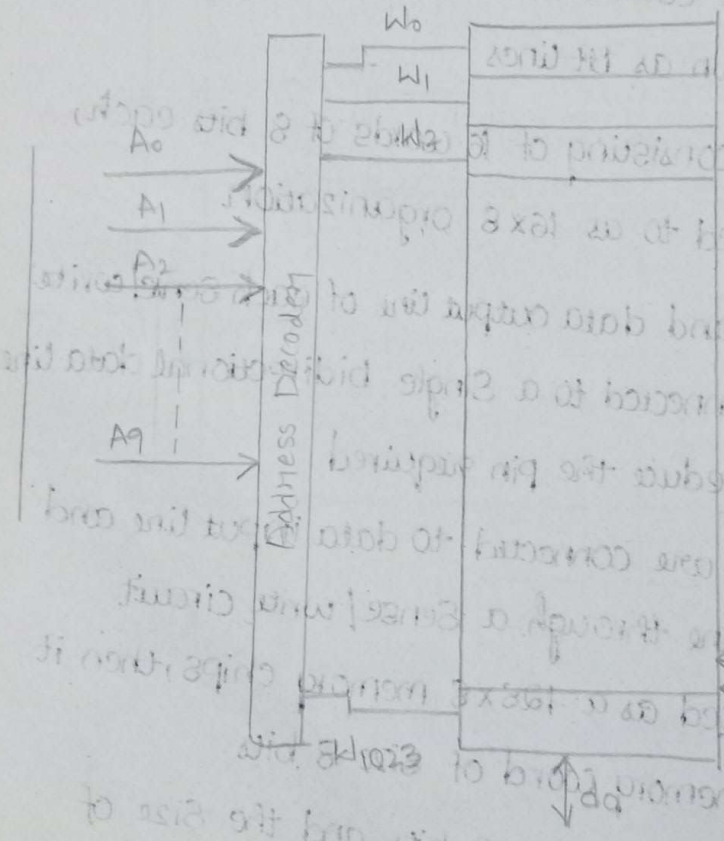
⇒ one such organization is shown in the figure



* During a Read operation, the Sense/write circuit Sense, or read the information stored in the cells selected by a word line and transmit this information to the output data line
* During a write operation, the Sense/write circuit receive information to and store it in the cells of the selected word
* For 16 words, we need an address bus of size 4. In addition to address and data lines, two control lines, and CS, are provided
* The line is to used to specify the required operation about read or write. The CS (chip select) line is required to select a given chip in a multiple chip memory system
* At a particular instant, one word line is enabled depending on the address present in the address bus
* The cells in each column are connected by two lines. These are known as bit lines
* A memory chip consisting of 16 words of 8 bits each, usually referred to as 16x8 organization
* The data input and data output line of each Sense/write circuit are connected to a single bidirectional data line in order to reduce the pin required
* These bits lines are connected to data input line and data output line through a Sense/write circuit
* If it is organised as a 128x8 memory chips, then it has got 128 memory word of size 8 bits.
* So, the size of data bus is 8 bits and the size of address bus is 7 bits (128x8)



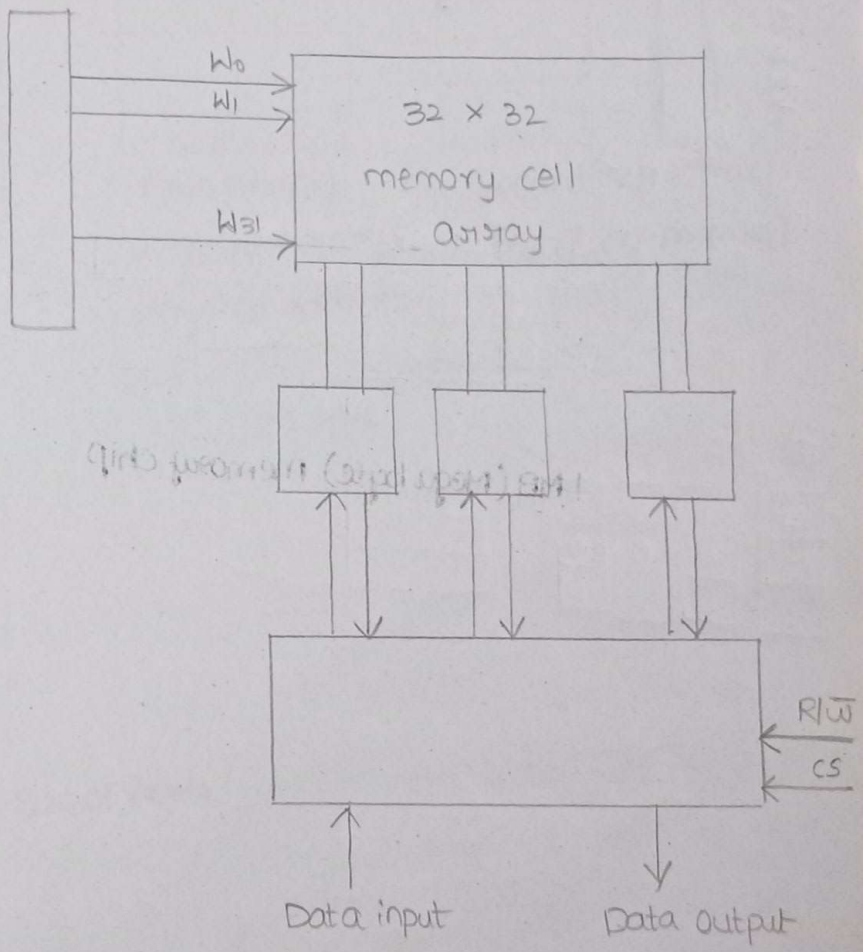
* If it is organized as a 1024x1 memory chips, then it has got 1024 memory words of size 1 bit only
* Therefore, the size of data bus is 1 bit and the size of address bus is 10 bits ($2^{10}=1024$).



* In second case, several memory words are organized in one row. In this case, address bus is divided into two groups
* one group is used to form the row address and the second group is used to form the column address

Consider the memory organization of 1024x1 memory chip

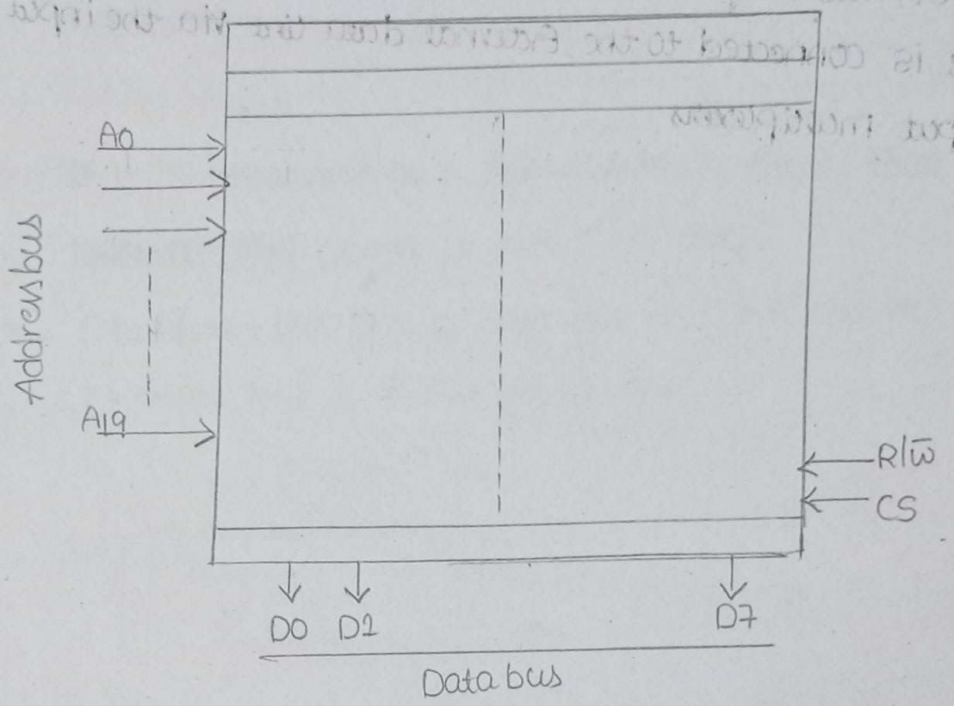
* The required 10-bit address is divided into two groups of 5 bits each to form the row and column address of the cell array
* A row address selects a row of 32 cells, all of which are accessed in parallel
* However, according to the column address, only one of these cells is connected to the External data line via the input output multiplexers



1M×8 Memory chips:

* The commercially available memory chips contain a much larger no. of cells. As for example, a memory unit of 1 MB (megabyte) size, organised as 1M×8 contains 2^20×8 memory cells
* It has got memory location (2^20) and each memory location contains 8 bits information. The size of address bus is 20 and the size of data bus is 8.

1M×8 memory chips



1 MB (Mega byte) memory chip

28/7/22

Virtual memory

↓

Paging

↓

Paging model

↓

Page faults & Page hits → Page is available

↳ The processor requested page is not available in main memory

↓

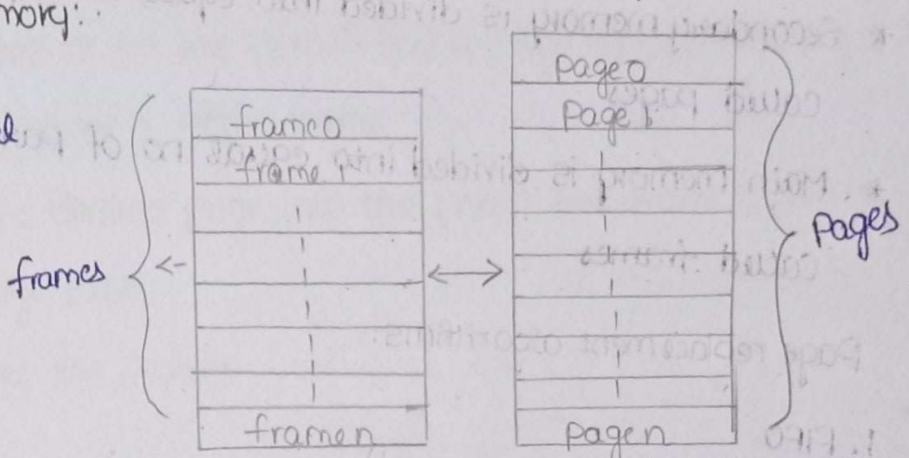
Page Replacement

Virtual memory:

↓

Not real

Non-contiguous memory allocation:



Page table:

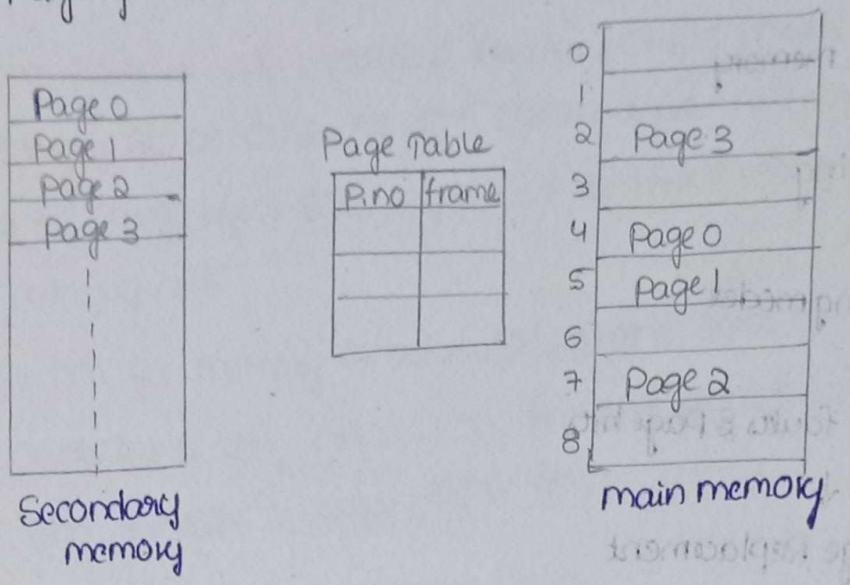
Page	Frame
0	4
1	5
2	7
3	2

Size of page = Size of frame

-for example

Page Size = 10kB

Paging model:



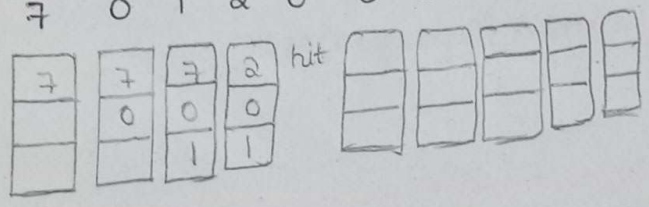
* Secondary memory is divided into equal no. of parts are called pages
* Main memory is divided into equal no. of parts are called frames

Page replacement algorithms:

1. FIFO
2. LRU (Least recently used)
3. OPT (optimal) ⇒ Best algorithm
   Drawback: It requires future knowledge of pages

FIFO: Drawback: Belady's Anamoly

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



Page replacement:

* In an operating System, page replacement is referred to a scenario in which a page from the main memory should be replaced by a page from Secondary memory
* page replacement occurs due to page faults

-> Basic page replacement algorithm:

1. Find the location of desired page on disk
2. Find a free frame
   * If there is a free frame, then use it
   * If there is no free frame, use a page replacement algorithm to select a victim frame
3. Read the desired page into the (new) free frame. update the page table
4. Restart the process