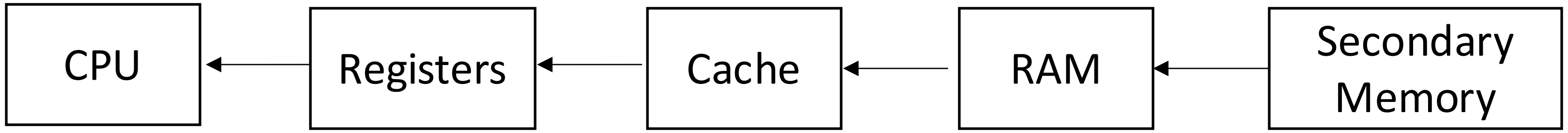
Unit-4

Memory Management

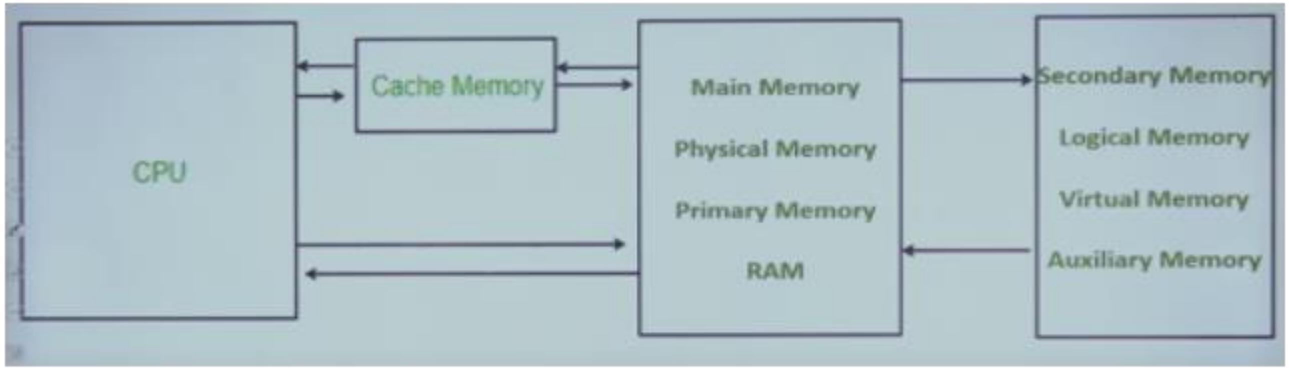
Principle of Concurrency

- Criteria of performance-

1. Size
2. Access Time
3. Cost



Locality of Reference



- Locality of reference refers to the tendency for a program to access a small portion of its memory at any given time, while the vast majority of its memory remains unused. There are two types of locality of reference-Temporal and spatial.
- **Spatial** – it says if CPU need some instruction from MM then there are maximum chances that next demand will be near by, therefore copy whole block in cache.
- **Temporal** –refers to the tendency for a program to access recently used memory locations repeatedly. This means that if a program accesses a memory location , it is likely to access that same location again in the near future.

Access Time

Assume access time of SM=1000ms

Access time of MM=100ms

Hit %o of RAM=90%

Total access time =hit ratio*RAM time +Miss ratio *[Penality+SM time]

Memory Allocation

1. Contiguous Allocation
    1. Fixed size partition
    2. Variable size partition
2. Non Contiguous allocation

Memory Allocation

1. Contiguous Allocation

Allocation will be sequential at time. Left space cannot be reused.

It can be of two type-

a. Fixed partitioning – Memory is divided in fixed size partitions. Size of partition cannot be changed. One partition can store only one process. it is again of two type-
   i. Equal size
   ii. Unequal size

Allocation can be contiguous only.

Disadvantage – Internal fragmentation, if size of process is not matched then process cannot be allocated.

Memory Allocation

1. Contiguous Allocation

a. Variable Size partitioning –

   i. As the process will come space will be allocated

   ii. Initial memory is available as single chunk

advantage – Internal fragmentation is not possible, Process size can be unlimited

Disadvantage- external fragmentation (holes)

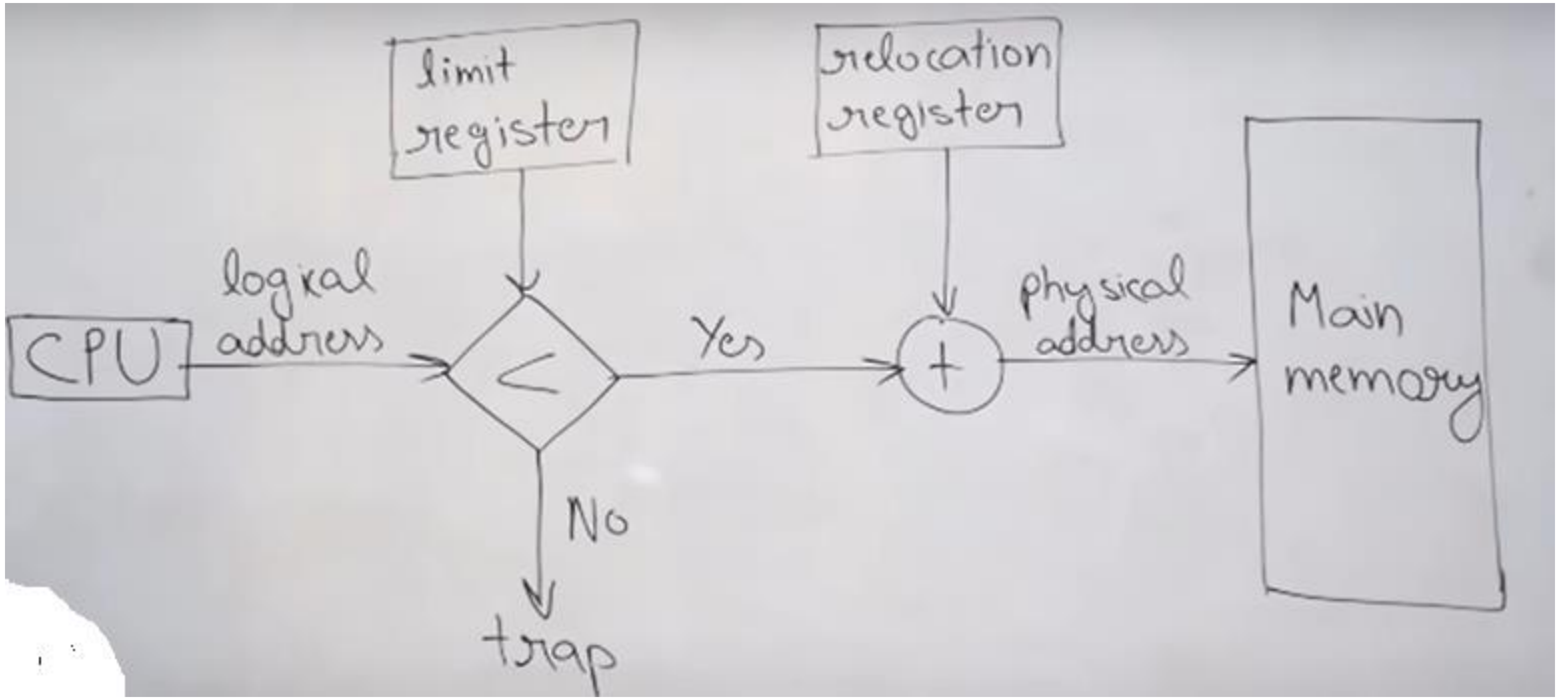
Memory Allocation

1. Contiguous Allocation

Allocation methods

1. First fit
2. Next fit
3. Best fit
4. Worst fit

Address Translation- Contiguous Allocation



CPU-

Limit Register-

Relocation Register-

Non-Contiguous Allocation

It is a memory allocation technique used in operating systems to allocate memory to processes that do not require a contiguous block of memory. In this technique, each process is allocated a series of non-contiguous blocks of memory that can be located anywhere in the physical memory.

There are two fundamental approaches to implementing noncontiguous memory allocation:

1. **Paging-** In paging, the memory is divided into fixed-size pages, and each page is assigned to a process. This technique is more efficient as it allows the allocation of only the required memory to the process.

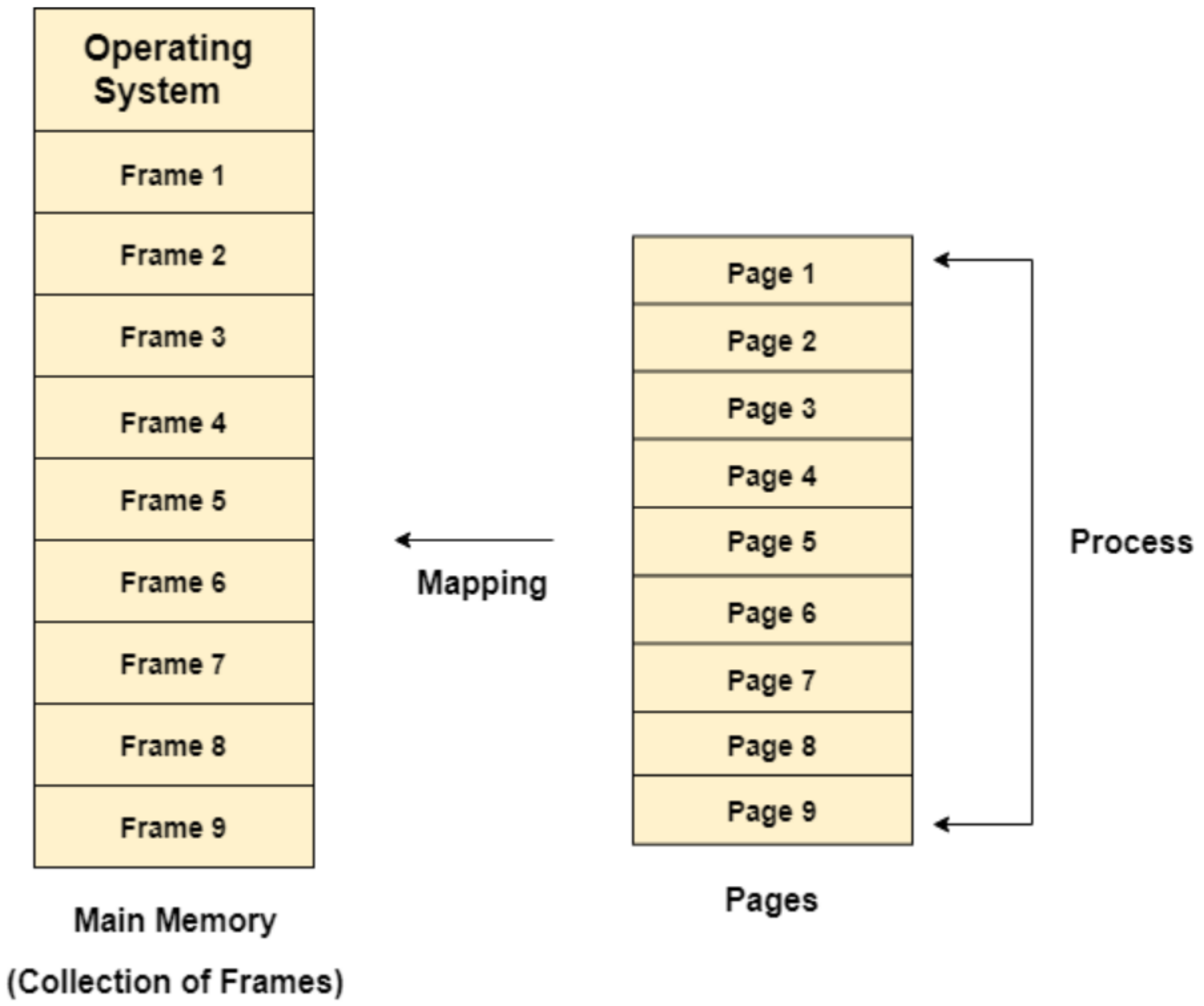
2. **Segmentation-** In segmentation, the memory is divided into variable-sized segments, and each segment is assigned to a process. This technique is more flexible than paging but requires more overhead to keep track of the allocated segments.

Non-Contiguous Allocation

Paging-

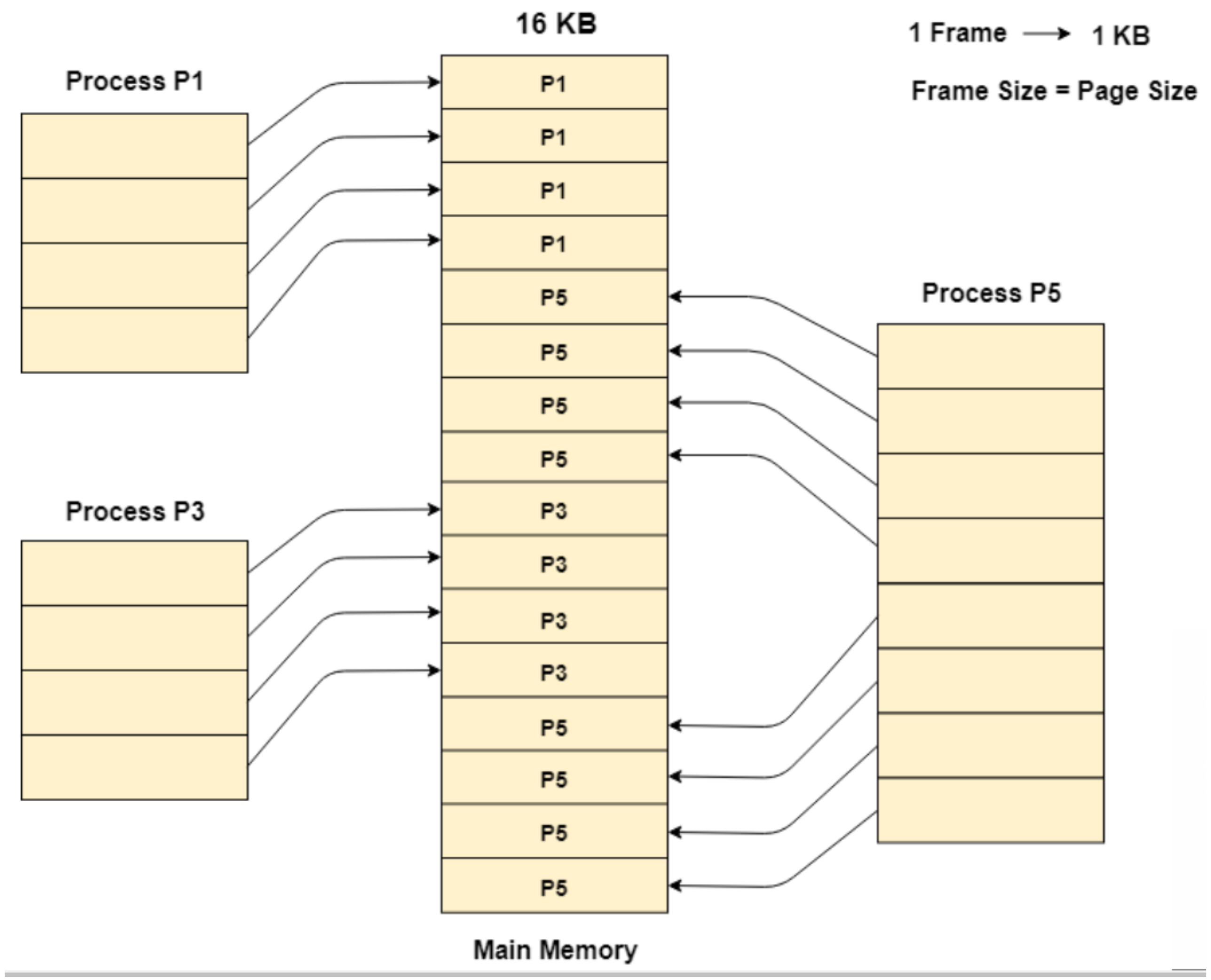
- Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.
- The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.
- One page of the process is to be stored in one of the frames of the memory. The pages can be stored at the different locations of the memory.
- Pages of the process are brought into the main memory only when they are required otherwise they reside in the secondary storage.

Paging



Operating System
Frame 1
Frame 2
Frame 3
Frame 4
Frame 5
Frame 6
Frame 7
Frame 8
Frame 9
Mapping
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 8
Page 9
Process
Pages
Main Memory
(Collection of Frames)

Paging


16 KB
1 Frame → 1 KB
Frame Size = Page Size
Process P1
Process P3
Process P5
P1
P1
P1
P1
P5
P5
P5
P5
P3
P3
P3
P3
P5
P5
P5
P5
Main Memory

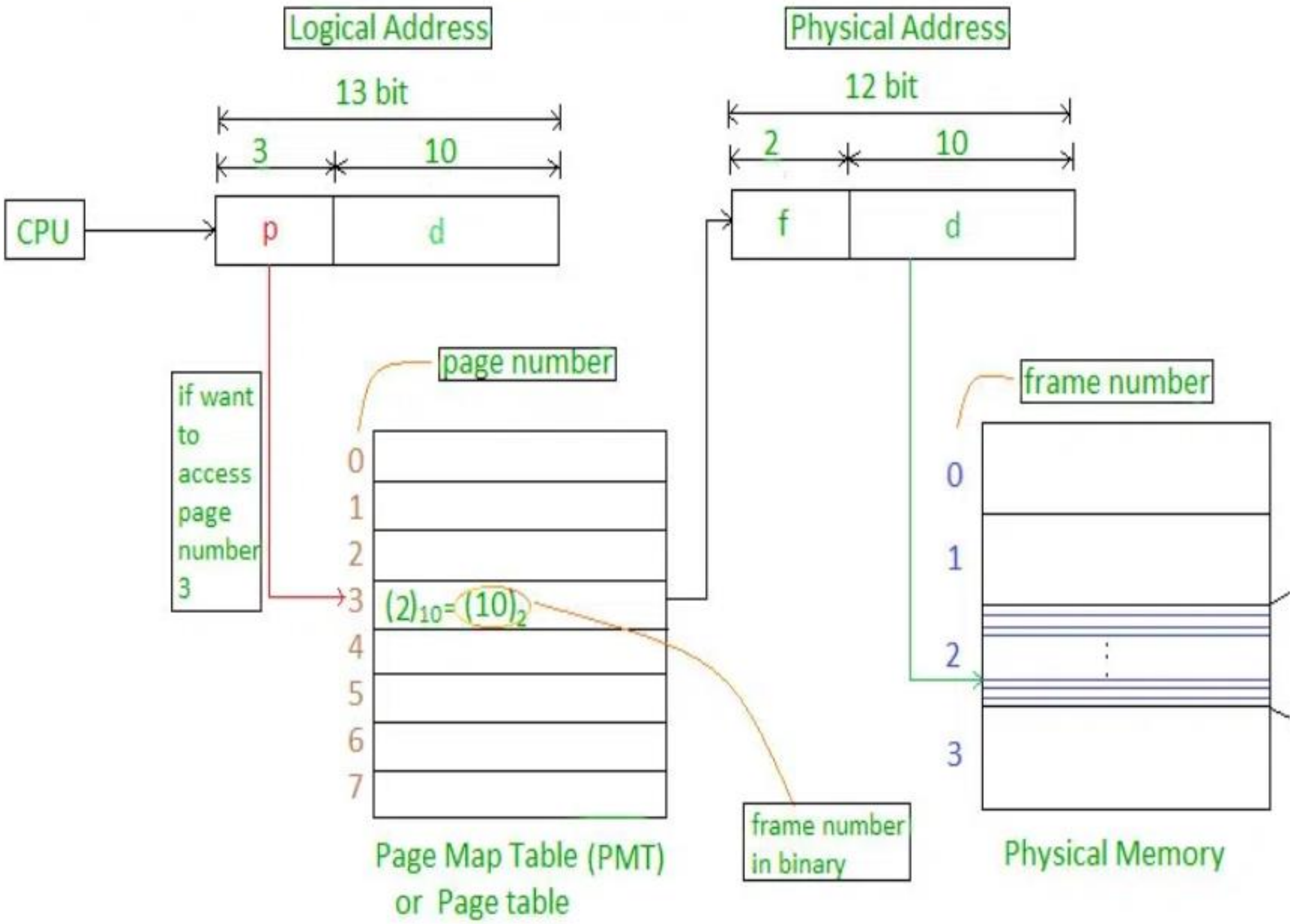
Paging (Address Translation)

- When a page is to be accessed by the CPU by using the logical address, the operating system needs to obtain the physical address to access that page physically.
- The logical address has two parts -Page Number, Offset.

PNO	offset

- Memory management unit of OS needs to convert the page number to the frame number.

Paging (Address Translation)



Paging (Address Translation)

Calculate Size of memory-

With n bits address size

- No of locations =2^n
- Size of memory=2^n x size of one location

Q1. Calculate memory size where one address is of 14bits and one location is of 1 byte.

Q2. Memory size if 64KB and memory is 2 byte addressable.

a) How many locations are there in memory?

b) How many address bits are required to represent address?

Paging (Address Translation)

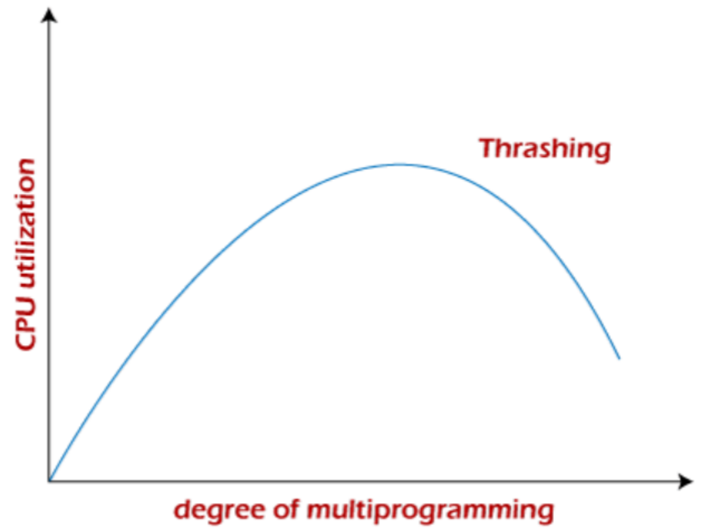
Calculate number of pages and number of frames-

Q. Let logical address is of 24bits and physical address is 16bits with page size of 1 KB. Find following-

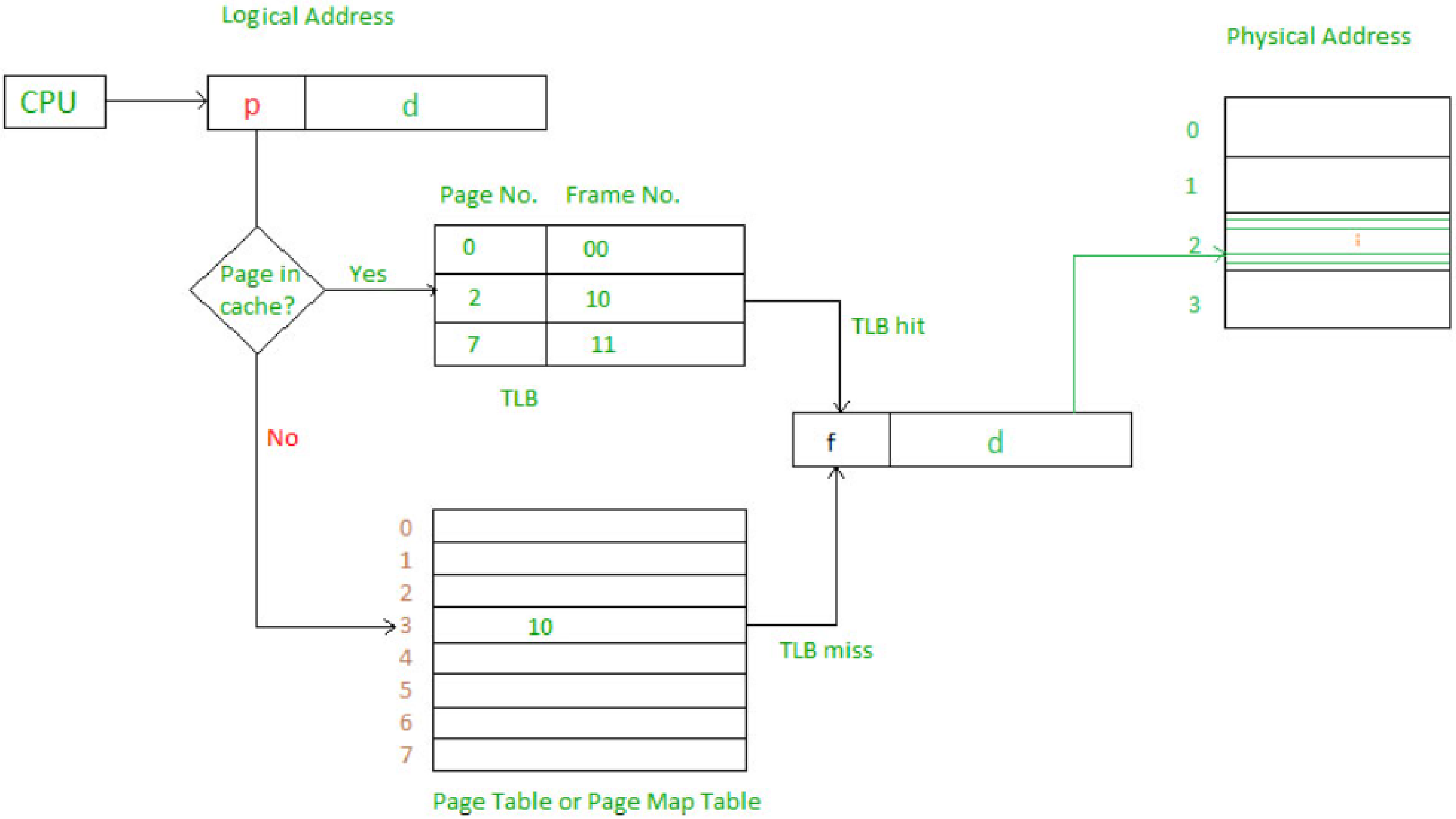
a. Size of SM
b. Size of MM
c. Number of pages
d. Number of frames

Thrashing

- **Page fault:** We know every program is divided into some pages. A page fault occurs when a program attempts to access data or code in its address space but is not currently located in the system RAM.
- **Swapping:** Whenever a page fault happens, the operating system will try to fetch that page from secondary memory and try to swap it with one of the pages in RAM. This process is called swapping.
- ***Thrashing*** is when the page fault and swapping happens very frequently at a higher rate, and then the operating system has to spend more time swapping these pages. This state in the operating system is known as thrashing. Because of thrashing, the CPU utilization is going to be reduced or negligible



Translation Lookaside Buffer



Translation Lookaside Buffer

Steps in TLB hit

- CPU generates a virtual (logical) address.
- It is checked in TLB (present).
- The corresponding frame number is retrieved, which now tells where the main memory page lies.

Steps in TLB miss

- CPU generates a virtual (logical) address.
- It is checked in TLB (not present).
- Now the page number is matched to the page table residing in the main memory (assuming the page table contains all PTE).
- The corresponding frame number is retrieved, which now tells where the main memory page lies.
- The TLB is updated with new PTE (if space is not there, one of the replacement techniques comes into the picture i.e either FIFO, LRU or MFU etc).

Translation Lookaside Buffer

Total access time with TLB

- Let RAM access time =x sec
- TLB access time=y sec, where y is very very smaller than x

Effective memory access time =hit(y+x)+miss(y+x+x)

1. Page will be searched in TLB (y sec) then frameno. Will be searched in Main memory (x sec)

2. If page not found in TLB then time =y + 2x

Q. Find Effective memory access time if TLB access time is 2sec with hit ratio of 90% and main memory time is 80sec.

Page Replacement Algorithms

First in first out (FIFO)

4,7,6,1,7,6,1,2,7,2

Frames=3 and frames=4

a) Use demand paging and calculate the no. of page faults and page hits.

b) What is the hit and miss ratio

Total =hit+miss

Hit ratio=(hit/total)*100

Page Replacement Algorithms

Belady's Anomaly- is a phenomenon where increasing the number of page frames results in an increase in the number of page faults for a given memory access pattern.

Q. 1,2,3,4,1,2,5,1,2,3,4,5

Frames=3 and 4

Page Replacement Algorithms

Optimal Page Replacement-

Predict the future and find which page is not going to used for the longest time in the future, replace it.

Q1. 4,7,6,1,7,6,1,2,7,2

Frames=3

Q2. 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3

Frames=4

Page Replacement Algorithms

Least Recently Used (LRU)-

Replace the page which is not used since long time

Q1. 4,7,6,1,7,6,1,2,7,2

Frames=3

Q2. Use FIFO and LRU and find Total hit and miss values.

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

Frames=3

Page Replacement Algorithms

Most Recently Used (MRU)-

Replace the page which is used recently.

Q1. 4,7,6,1,7,6,1,2,7,2

Frames=3

Q2. Use FIFO and LRU and find Total hit and miss values.

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

Frames=3

Practice

Q1. A paging system uses 48-bit addresses, each of which specifies 1 byte (B) of memory. The system has a main memory unit of 512MB, and a page size of 16KB. Answer the following questions:

a) How many frames does the system memory contain?

b) How many bits does the system use to maintain displacements (i.e offsets), and how many bits does the system use to maintain page numbers?

Q2. There are four process P1, to P4 stored in Virtual Memory. P1 process takes 2 pages {1,4}, P2{3,5,6}, P3{2,7} and P4{0,8,9,10}. Each page can contain 5 instructions. The number of frames in MM are 5 in which 2^{nd} frame is busy. CPU has demanded P2 process 10^{th} instruction and P4 process 4^{th} instruction. Design address mapping diagram with proper data structures and addressing.