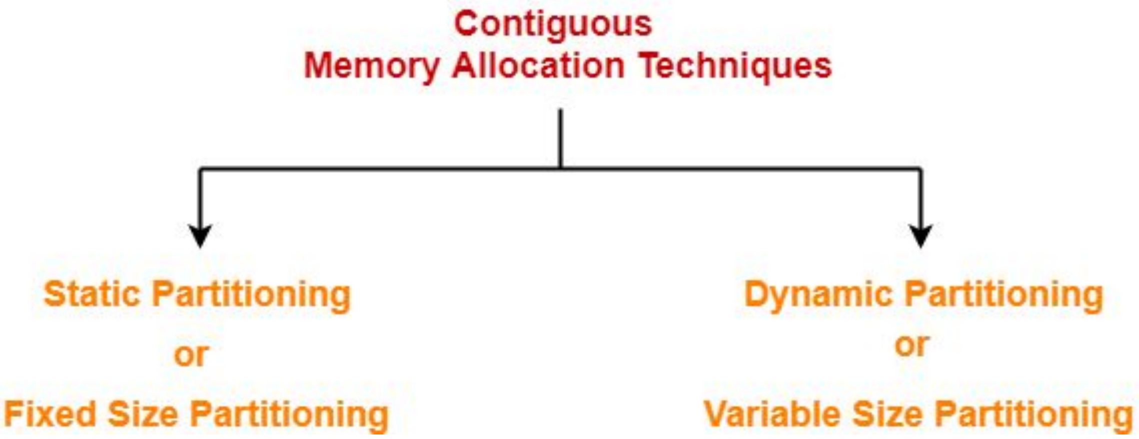
# **Contiguous Memory Allocation-**

- Contiguous memory allocation is a memory allocation technique.
- It allows to store the process only in a contiguous fashion.
- Thus, entire process has to be stored as a single entity at one place inside the memory.

## Techniques-

There are two popular techniques used for contiguous memory allocation-



Contiguous
Memory Allocation Techniques

Static Partitioning
or
Fixed Size Partitioning

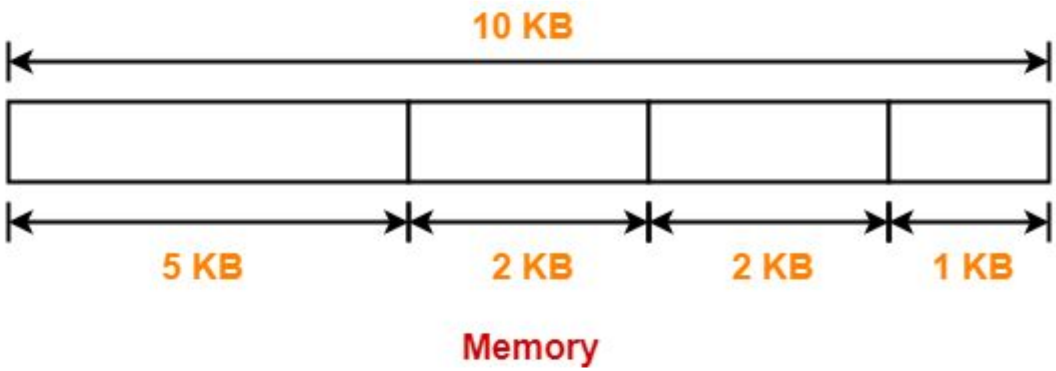
Dynamic Partitioning
or
Variable Size Partitioning

## Static Partitioning-

- Static partitioning is a fixed size partitioning scheme.
- In this technique, main memory is pre-divided into fixed size partitions.
- The size of each partition is fixed and can not be changed.
- Each partition is allowed to store only one process.

## Example-

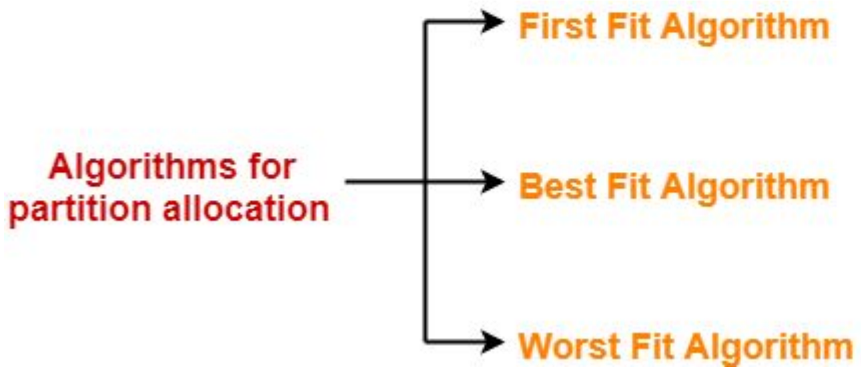
Under fixed size partitioning scheme, a memory of size 10 KB may be divided into fixed size partitions as-



- These partitions are allocated to the processes as they arrive.
- The partition allocated to the arrived process depends on the algorithm followed.

# Algorithms for Partition Allocation-

Popular algorithms used for allocating the partitions to the arriving processes are-



## 1. First Fit Algorithm-

- This algorithm starts scanning the partitions serially from the starting.
- When an empty partition that is big enough to store the process is found, it is allocated to the process.
- Obviously, the partition size has to be greater than or at least equal to the process size.

## 2. Best Fit Algorithm-

- This algorithm first scans all the empty partitions.
- It then allocates the smallest size partition to the process.

## . Worst Fit Algorithm-

- This algorithm first scans all the empty partitions.
- It then allocates the largest size partition to the process.

## Important Points-

For static partitioning,

- Best Fit Algorithm works best.
- This is because space left after the allocation inside the partition is of very small size.
- Thus, internal fragmentation is least.
- Worst Fit Algorithm works worst.
- This is because space left after the allocation inside the partition is of very large size.
- Thus, internal fragmentation is maximum.

# Internal Fragmentation

- It occurs when the space is left inside the partition after allocating the partition to a process.
- This space is called as internally fragmented space.
- This space can not be allocated to any other process.
- This is because only static partitioning allows to store only one process in each partition.
- Internal Fragmentation occurs only in static partitioning.

## External Fragmentation

- It occurs when the total amount of empty space required to store the process is available in the main memory.
- But because the space is not contiguous, so the process can not be stored.

## Contiguous Memory Allocation- Dynamic Partition

- Dynamic partitioning is a variable size partitioning scheme.
- It performs the allocation dynamically.
- When a process arrives, a partition of size equal to the size of process is created.
- Then, that partition is allocated to the process.

## Partition Allocation Algorithms-

- he processes arrive and leave the main memory.
- As a result, holes of different size are created in the main memory.
- These holes are allocated to the processes that arrive in future.

**Popular partition allocation algorithms are-**

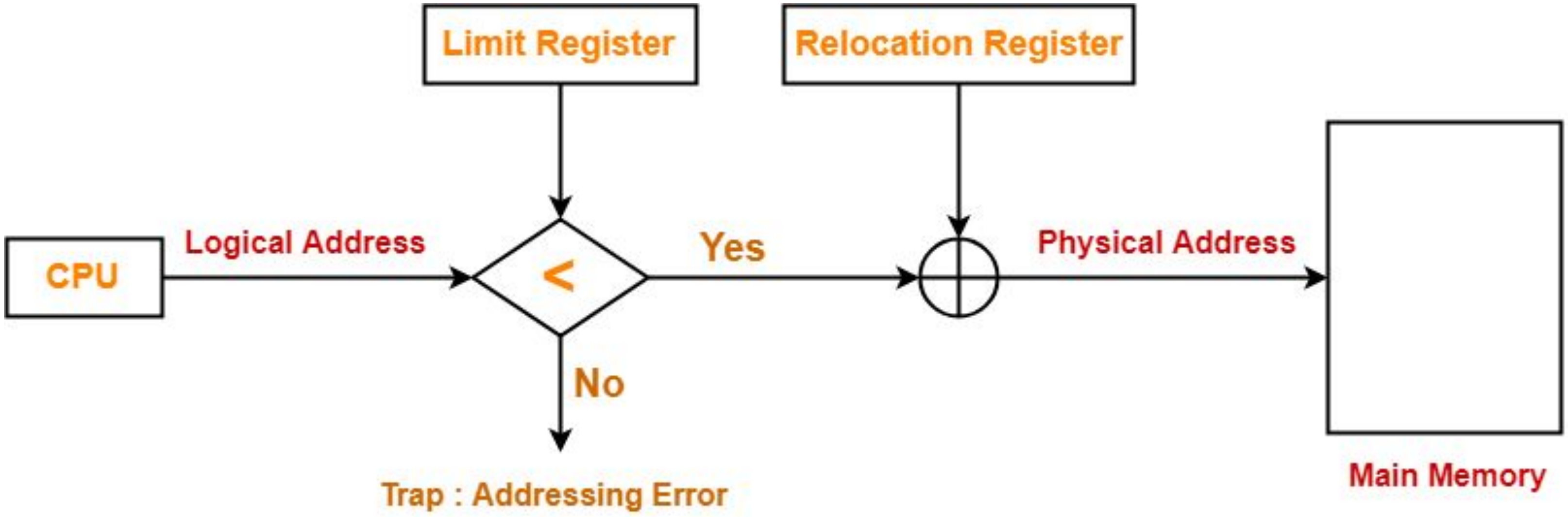
- First Fit Algorithm
- Best Fit Algorithm
- Worst Fit Algorithm

## **Important Points-**

For dynamic partitioning,

- Worst Fit Algorithm works best.
- This is because space left after allocation inside the partition is of large size.
- There is a high probability that this space might suit the requirement of arriving processes.
- Best Fit Algorithm works worst.
- This is because space left after allocation inside the partition is of very small size.
- There is a low probability that this space might suit the requirement of arriving processes.

# Translating Logical Address into Physical Address-



Translating Logical Address into Physical Address

- CPU always generates a logical address.
- A physical address is needed to access the main memory.

## Step-01:

- The translation scheme uses two registers that are under the control of operating system.
- During context switching, the values corresponding to the process being loaded are set in the registers.

These two registers are-

- Relocation Register
- Limit Register

- Relocation Register stores the base address or starting address of the process in the main memory.
- Limit Register stores the size or length of the process.

## Step-02:

- CPU generates a logical address containing the address of the instruction that it wants to read.

## Step-03:

- The logical address generated by the CPU is compared with the limit of the process.
- Now, two cases are possible-

## Case-01: Generated Address >= Limit

- If address is found to be greater than or equal to the limit, a trap is generated.
- This helps to prevent unauthorized access.

## Case-02: Generated Address < Limit

- The address must always lie in the range [0, limit-1].
- If address is found to be smaller than the limit, then the request is treated as a valid request.
- Then, generated address is added with the base address of the process.
- The result obtained after addition is the address of the memory location storing the required word.

# Non-Contiguous Memory Allocation-

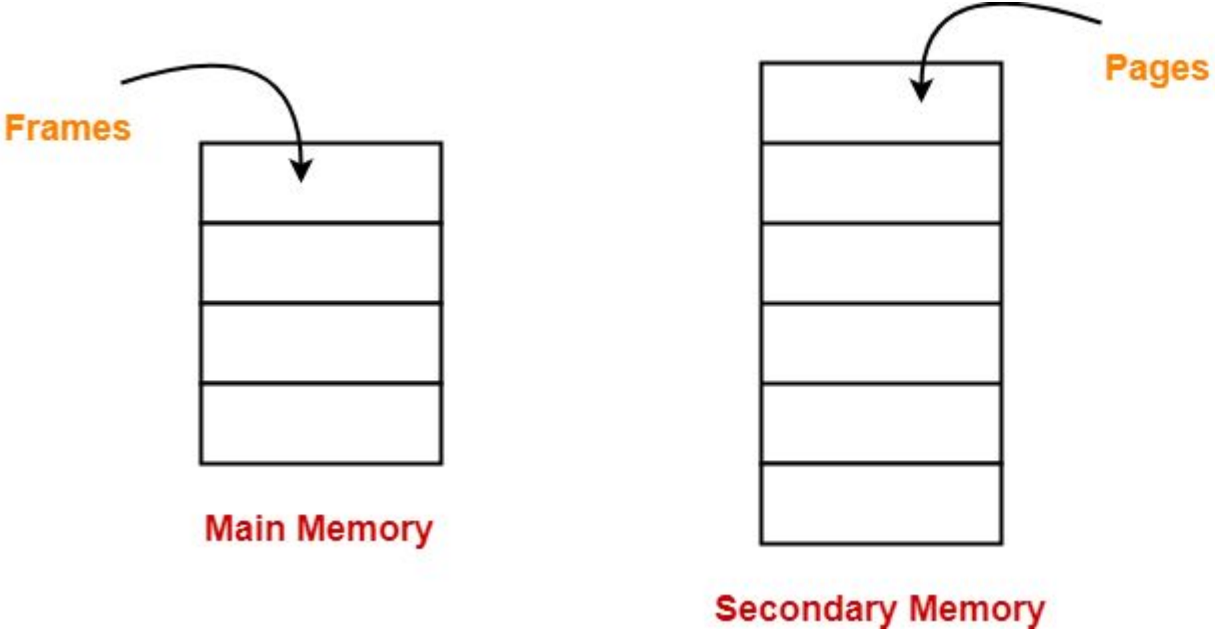
- Non-contiguous memory allocation is a memory allocation technique.
- It allows to store parts of a single process in a non-contiguous fashion.
- Thus, different parts of the same process can be stored at different places in the main memory.

# Techniques-

Paging

# Paging-

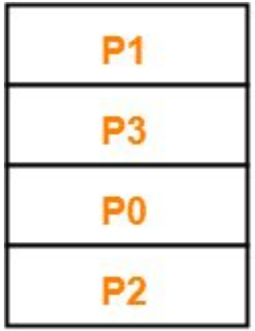
- Paging is a fixed size partitioning scheme.
- In paging, secondary memory and main memory are divided into equal fixed size partitions.
- The partitions of secondary memory are called as **pages**.
- The partitions of main memory are called as **frames**.



- Each process is divided into parts where size of each part is same as page size.
- The size of the last part may be less than the page size.
- The pages of process are stored in the frames of main memory depending upon their availability.

# Example-

- Consider a process is divided into 4 pages $P_0$, $P_1$, $P_2$ and $P_3$.
- Depending upon the availability, these pages may be stored in the main memory frames in a non-contiguous fashion as shown-



Main Memory

## Translating Logical Address into Physical Address-

- CPU always generates a logical address.
- A physical address is needed to access the main memory.

## **Step-01:**

CPU generates a logical address consisting of two parts-

1. Page Number
2. Page Offset

Page Number specifies the specific page of the process from which CPU wants to read the data.

Page Offset specifies the specific word on the page that CPU wants to read.

## Step-02:

For the page number generated by the CPU,

- **Page Table** provides the corresponding frame number (base address of the frame) where that page is stored in the main memory.

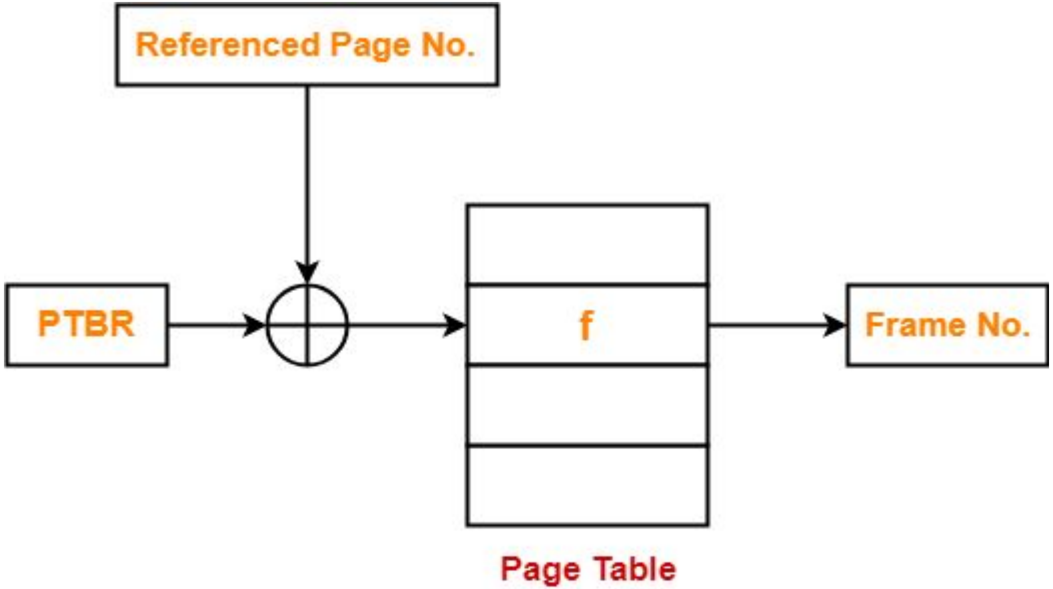
## Page Table-

- Page table is a data structure.
- It maps the page number referenced by the CPU to the frame number where that page is stored.

## Characteristics of Page Table

- Page table is stored in the main memory.
- Number of entries in a page table = Number of pages in which the process is divided.
- Page Table Base Register (PTBR) contains the base address of page table.

## Working-

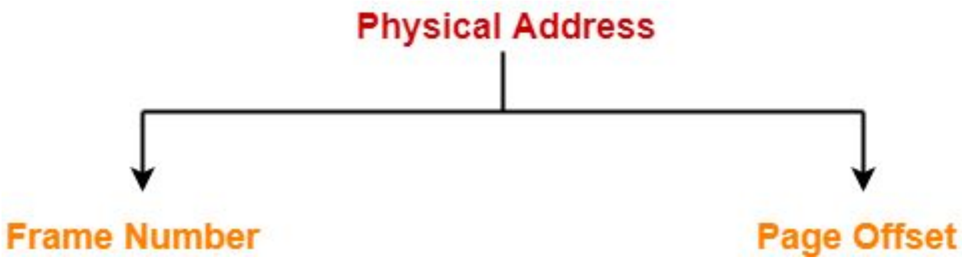


Obtaining Frame Number Using Page Table

- Page Table Base Register (PTBR) provides the base address of the page table.
- The base address of the page table is added with the page number referenced by the CPU.
- It gives the entry of the page table containing the frame number where the referenced page is stored.

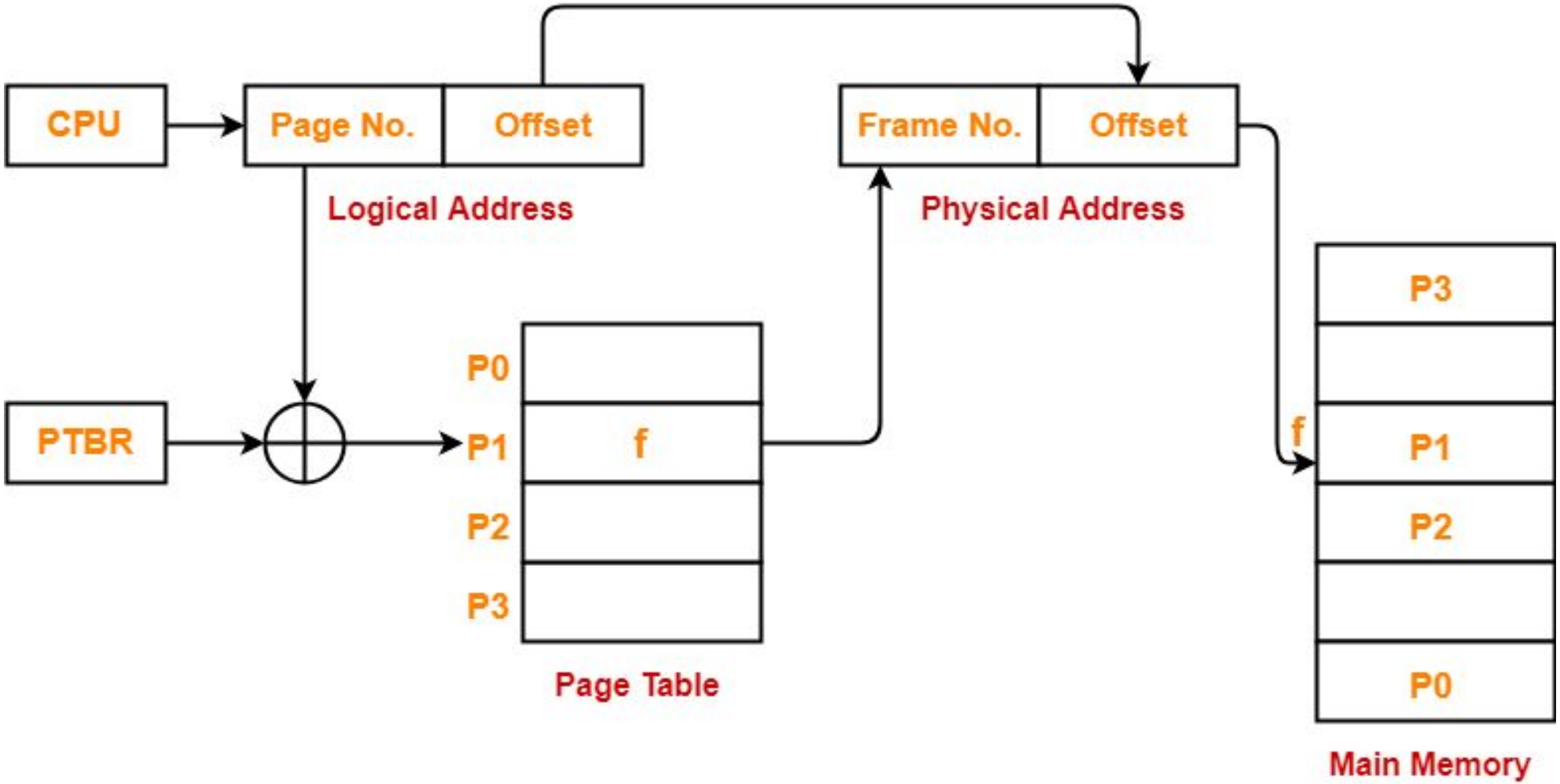
## Step-03:

- Frame number specifies the specific frame where the required page is stored.
- Page Offset specifies the specific word that has to be read from that page.



- Frame number specifies the specific frame where the required page is stored.
- Page Offset specifies the specific word that has to be read from that page.

## Diagram-



Translating Logical Address into Physical Address-

# **Advantages-**

- It allows to store parts of a single process in a non-contiguous fashion.
- It solves the problem of external fragmentation.

The disadvantages of paging are-

- It suffers from internal fragmentation.
- There is an overhead of maintaining a page table for each process.
- The time taken to fetch the instruction increases since now two memory accesses are required.

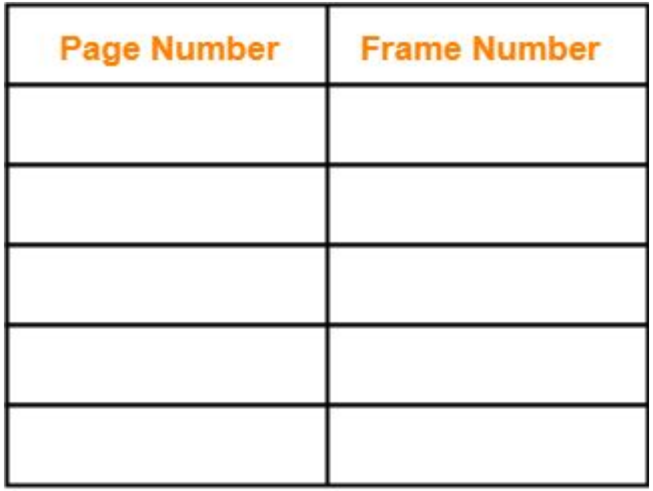
# Translation Lookaside Buffer-

- Translation Lookaside Buffer (TLB) is a solution that tries to reduce the effective access time.
- Being a hardware, the access time of TLB is very less as compared to the main memory.

## Structure-

Translation Lookaside Buffer (TLB) consists of two columns-

1. Page Number
2. Frame Number



| Page Number | Frame Number |
|---|---|
| | |
| | |
| | |
| | |
| | |

Translation Lookaside Buffer

# Translating Logical Address into Physical Address-

The logical address generated by the CPU is translated into the physical address using following steps-

**Step-01:**

CPU generates a logical address consisting of two parts-

1. Page Number
2. Page Offset

**Step-02:**

- TLB is checked to see if it contains an entry for the referenced page number.
- The referenced page number is compared with the TLB entries all at once.

# Now, two cases are possible-

## Case-01: If there is a TLB hit-

- If TLB contains an entry for the referenced page number, a TLB hit occurs.
- In this case, TLB entry is used to get the corresponding frame number for the referenced page number.

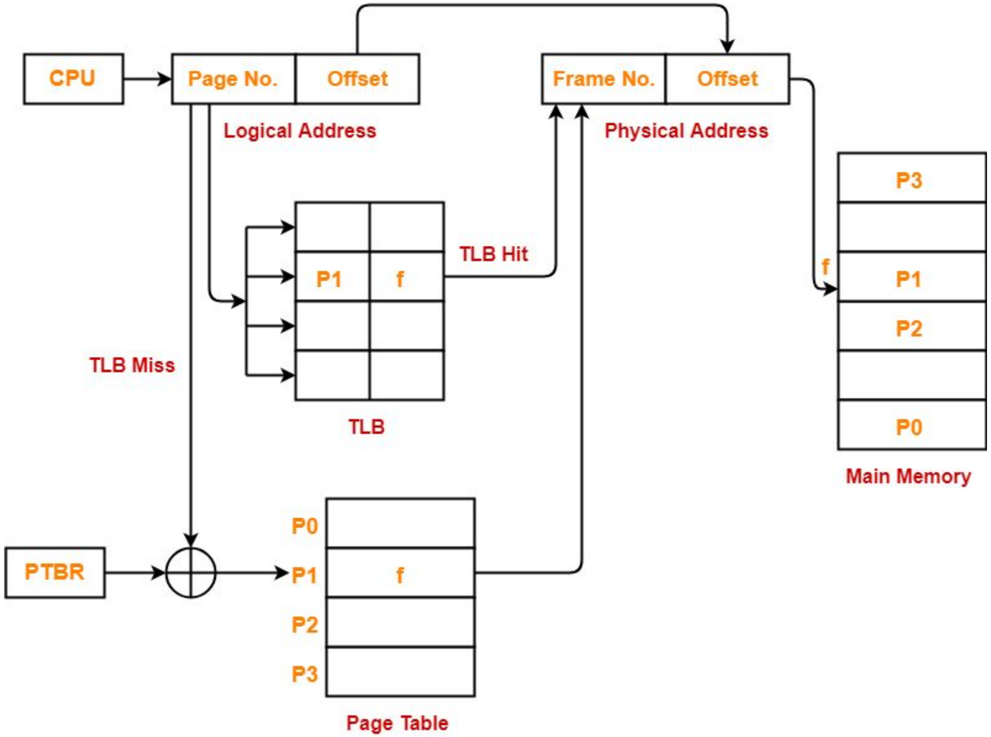
## Case-02: If there is a TLB miss-

- If TLB does not contain an entry for the referenced page number, a TLB miss occurs.
- In this case, page table is used to get the corresponding frame number for the referenced page number.
- Then, TLB is updated with the page number and frame number for future references.

# **Step-03:**

- After the frame number is obtained, it is combined with the page offset to generate the physical address.
- Then, physical address is used to read the required word from the main memory.

# Diagram-



Translating Logical Address into Physical Address

# Important Points-

## Point-01:

- Unlike page table, there exists only one TLB in the system.
- So, whenever context switching occurs, the entire content of TLB is flushed and deleted.

## Point-02:

When a new process gets scheduled-

- Initially, TLB is empty. So, TLB misses are frequent.
- With every access from the page table, TLB is updated.
- After some time, TLB hits increases and TLB misses reduces.

## Advantages-

The advantages of using TLB are-

- TLB reduces the effective access time.
- Only one memory access is required when TLB hit occurs.

## Disadvantages-

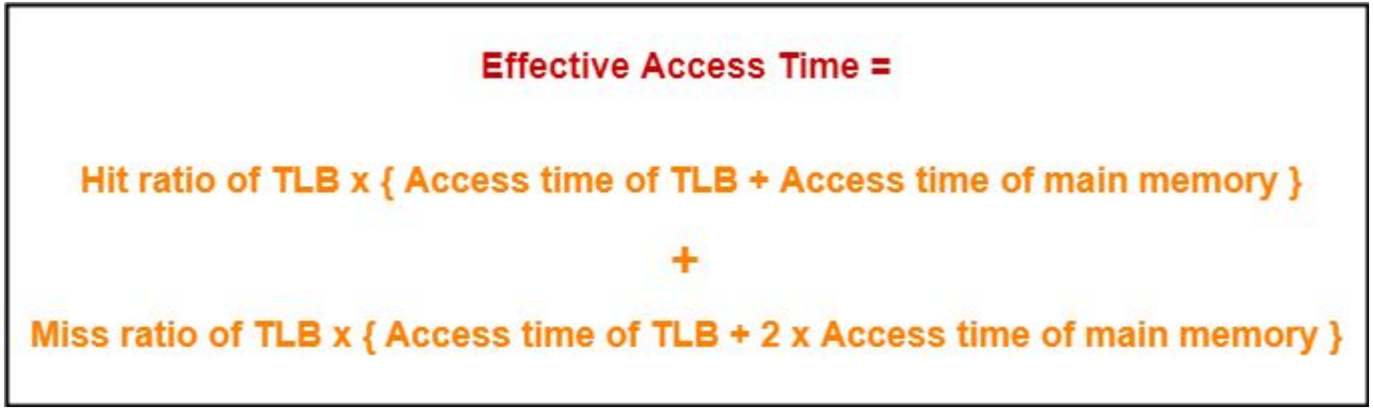
A major disadvantage of using TLB is-

- After some time of running the process, when TLB hits increases and process starts to run smoothly, a context switching occurs.
- The entire content of the TLB is flushed.
- Then, TLB is again updated with the currently running process.

This happens again and again.

Other disadvantages are-

- TLB can hold the data of only one process at a time.
- When context switches occur frequently, the performance of TLB degrades due to low hit ratio.
- As it is a special hardware, it involves additional cost.



Effective Access Time =

Hit ratio of TLB x { Access time of TLB + Access time of main memory }

+

Miss ratio of TLB x { Access time of TLB + 2 x Access time of main memory }

## Page Fault-

- When a page referenced by the CPU is not found in the main memory, it is called as a **page fault.**
- When a page fault occurs, the required page has to be fetched from the secondary memory into the main memory.