NOTES

Unit 4

Date .....

Virtual Memories

→ In most modern computer systems, physical main memory is not as large as address space spanned by an address issued by processor

→ When a program does not completely fit into main memory, parts of it not currently being executed are stored on secondary storage devices (like Magnetic disks)

→ When a new segment of program is to be moved into full main memory, it must replace another segment already in memory

→ OS moves programs & data automatically b/w main memory & secondary storage.

NOTES

Date .....

∴ application programmer does not need to be aware of limitations imposed by available main memory

Techniques that automatically move program & data blocks into physical main memory when they are required for execution are called virtual-memory techniques.

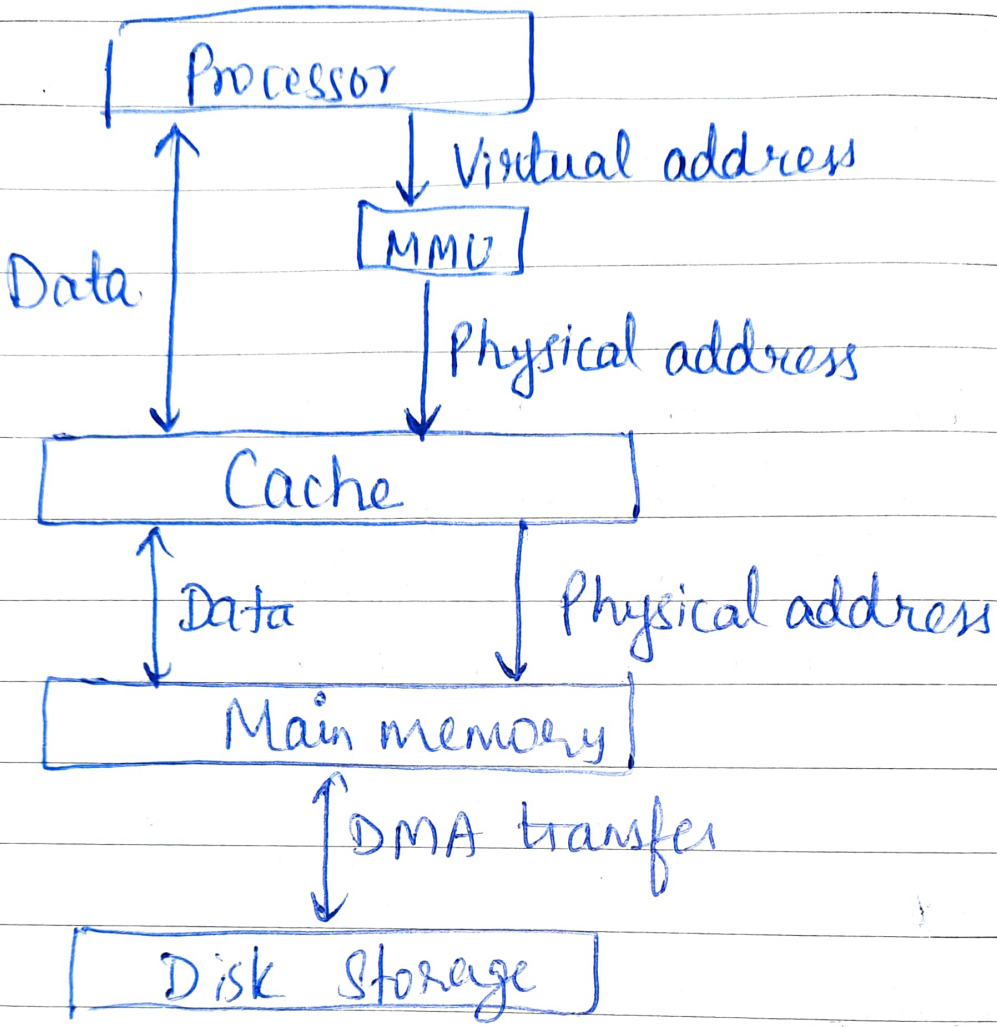
∴ program & thus processor reference and instruction & data space that is independent of available main memory space

Virtual or logical addresses
↳ binary addresses that processor issues for either instruction or data

translated into physical addresses by a combination of hardware & software components.

NOTES

Date .....



Virtual memory organization

Memory Management Unit (MMU)
(special hardware unit)
↳ translates virtual addresses into physical addresses

- When desired data/instructions are in Main memory, these are fetched to Cache

NOTES

Date .....

If data are not in main memory, MMU causes OS to bring data into memory from disk

Address translation from virtual address into physical address

processes by which data stored to & then retrieved from storage disk

- paging
- segmentation

① Paging

Assume all programs & data composed of fixed-length units called pages, each of which consists of a block of words that occupy contiguous loc. in main memory → frame

Page range generally 2K to 16K bytes in length

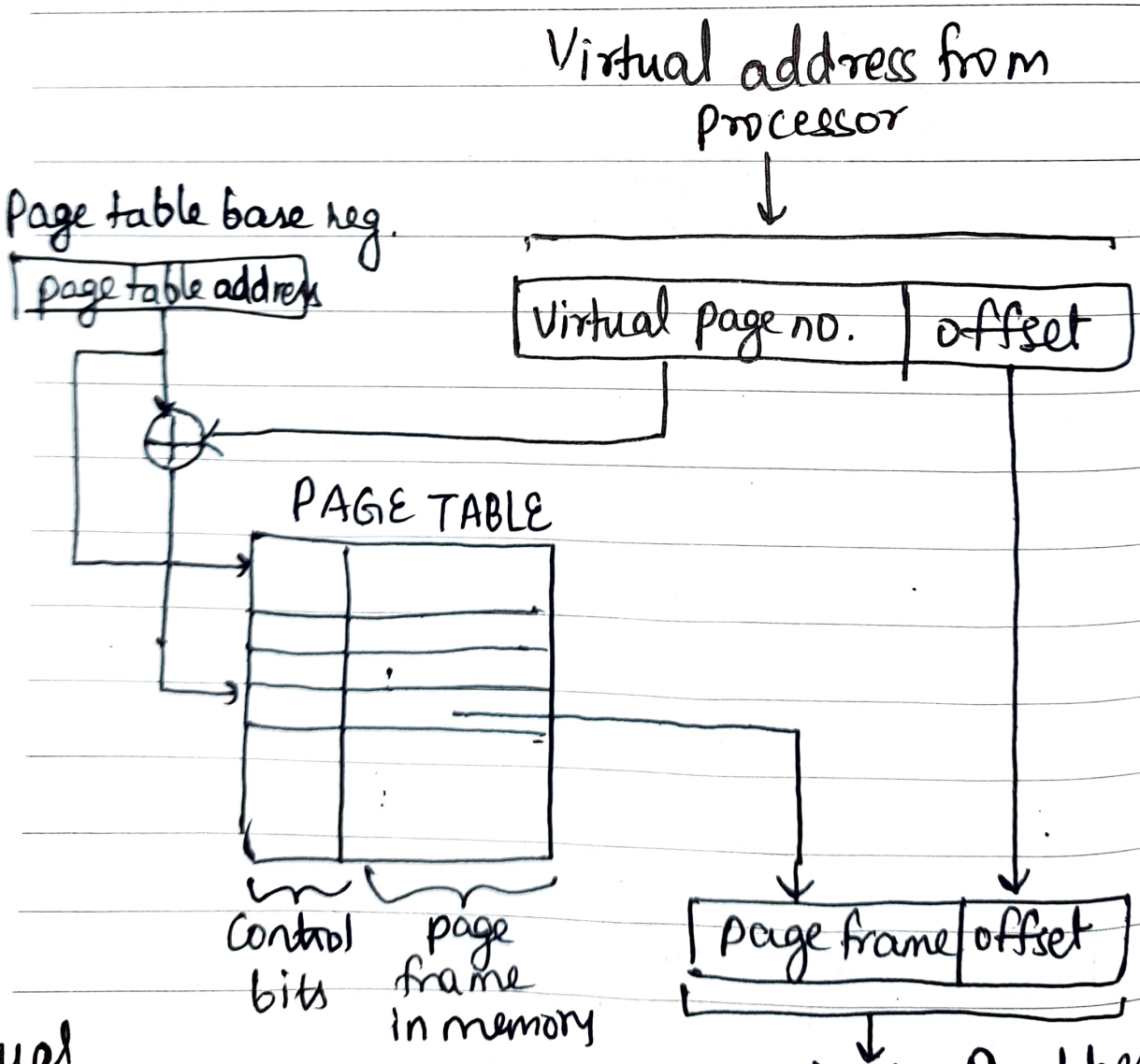
Page → basic unit of info to be moved b/w MM & disk

NOTES

Date .....

→ cache bridges speed gap b/w processor & MM & is implemented in hardware.

→ virtual memory mechanism bridges size & speed gaps b/w MM & secondary storage & is usually implemented in part by software techniques



Virtual address translation

NOTES

Date .....

Concept of fixed length pages

Virtual address generated by processor

comprises
- Virtual page no (high-order bits)
- Offset (low-order bits)
  - ↳ location of particular byte (or word) within page

Info about MM loc. of each page kept in page table
(including MM address where page is stored & current status of page)

An area in main memory that can hold one page → page frame

Starting address of page table kept in page table base register

Virtual page no. + contents of page table base reg → address of entry in page table

gives starting address of page if it currently resides in MM | whose content

NOTES

Date .....

Each entry in page table also includes some control bits that describe status of page while it is in MM.

e.g. ↳ one bit indicates validity of page (whether page is actually loaded in MM)

- Another bit indicates whether page has been modified during its residency in memory
- Other control bits indicate various restrictions that may be imposed on accessing page (eg. full read and write permission or read access only permission)

- MMU uses page table info for every read and write access thus ideally page table should reside within MMU but due to its large size, page table is kept in MM.

NOTES

Date .....

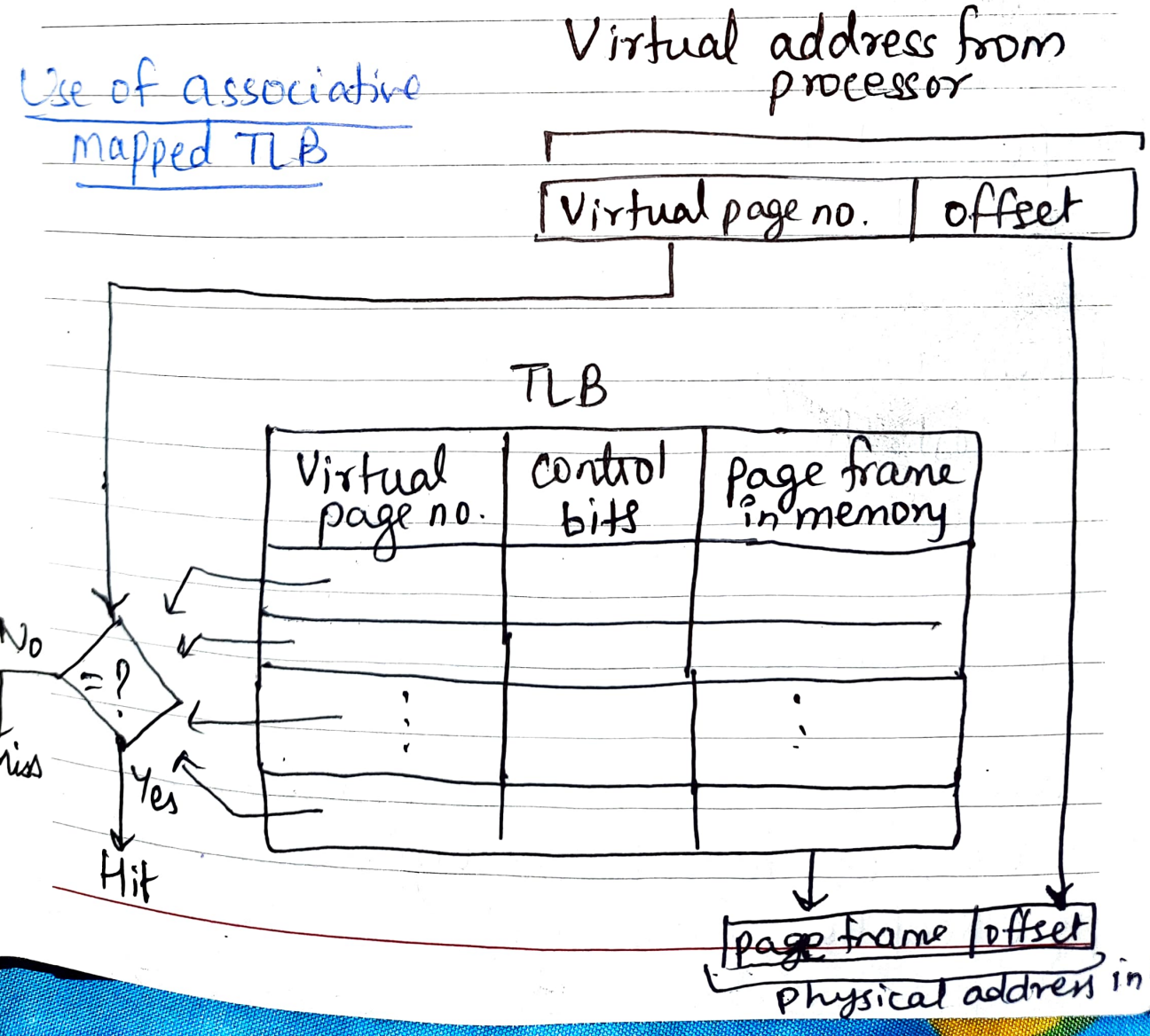
MMU implemented as part of CPU chip (along with primary cache)

A copy of small portion of page table can be accomodated within MMU.

↳ corresponding to most recently accessed pages

A small cache, usually called Translation Lookaside Buffer (TLB) is used in MMU for this purpose.

Use of associative mapped TLB



NOTES

Date .....

Address translation process

→ Given a virtual address, MMU looks in TLB for referenced page.

→ If page table entry for this page is found in TLB, physical address is obtained immediately.

→ If there is a miss in TLB, then required entry is obtained from page table in MM and TLB is updated.

When a program generates an access request to a page that is not in MM, a page fault occurs. Thus whole page must be brought from disk into MM before access can proceed.

When page fault is detected, MMU asks OS to intervene by raising an exception (interrupt)

NOTES

Date .....

Page replacement

If new page is brought from disk when MM is full, it must replace one of resident pages.

Concepts similar to LRU replacement algorithm can be applied to page replacement

A modified page has to be written back to disk before it is removed from MM.

↳ Write-through protocol

Like cache memories is not suitable for virtual memory.

as access time of disk is very long.

Advantages of paging

- On programmer level, paging is transparent function & does not require intervention
- No external fragmentation
- No internal fragmentation on updated os
- Frames donot have to be contiguous

NOTES

Date .....

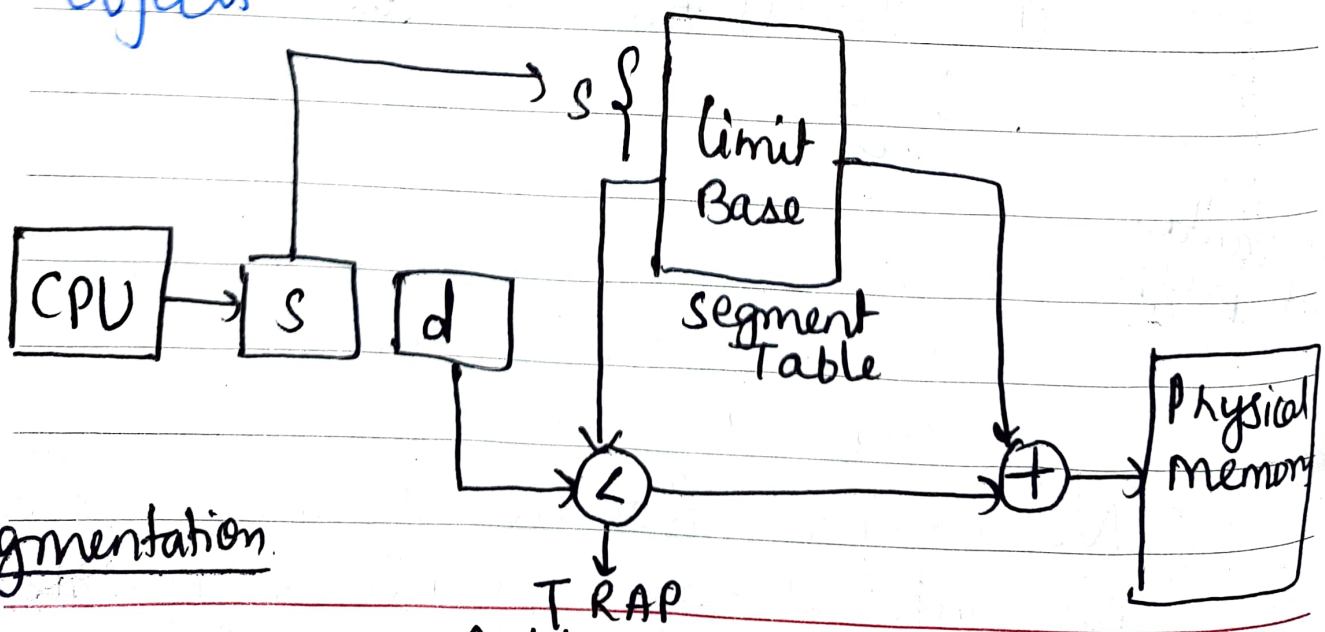
Disadv of paging

- Paging causes internal fragmentation on older systems
- Longer memory lookup times than Segmentation; remedy with TLB memory caches

② Segmentation

↳ virtual process that creates address space of various sizes in computer system, called segments

→ Each segment is different virtual address space that directly corresponds to process objects



Segmentation

NOTES

Date .....

When process executes, segmentation assigns related data into segments for faster processing.

Segmentation function maintains a segment table that includes physical addresses of seg., size & other data.

- segment no. (s) → no. of bits required to represent segment
- segment offset (d) → no. of bits required to represent size of segment.

→ Segmentation translates CPU-generated virtual addresses into physical addresses that refer to a unique physical memory location.

Challeng of external fragmentation due to segmentation →

- External fragmentation occurs when unusable memory is located outside of allocated memory blocks.
- Issue is that system may have enough memory to satisfy process request, but

NOTES Date .....

available memory is not contiguous location

→ In time, fragmentation worsens & significantly slows segmentation process.

Adv of Segmentation

- No internal fragmentation
- Segment tables consumes less space as compared to page tables
- Average segment sizes are larger than most page sizes allowing segments to store more process data.
- Less processing overhead.
- Simpler to relocate segments than to relocate contiguous address spaces on disk.
- Takes up less memory due to smaller Segment tables as compared to page

NOTES

Date .....

tables.

Disadv of segmentation

- Requires programmer intervention
- Subject to serious external fragmentation.
- Linux only supports segmentation in 80x86 µPs. Porting Linux to different architectures is problematic because of limited segmentation support.

Key differences b/w paging and segmentation

Parameter	Paging	Segmentation
① Size	fixed block size for pages & frames determined by computer hardware	Variable size segments that are user defined.
② Fragmentation	Older systems subject to internal fragmentation by not allocating entire pages to memory	external fragmentation

Parameter	Paging	Segmentation
③ Tables	Page tables direct MMU to page loc. & status. • Slower process than segment tables but TLB cache memory accelerates it	Segmentation tables contain Seg. ID & info • faster than direct paging table lookups.
④ Availability	Widely available on CPUs and as MMU chips	Windows Servers may support backwa Compatibility, while Linux has very limited suppor

③. Segmented Paging used by modern computers

Main memory → divided into → variable sized segments → further divided into → smaller fixed size pages on disk

NOTES

Date ...../...../.....

info about each segment / page

Each segment contains a page table, and there are multiple page tables per process.

Segment table has info about every segment

Segment tables mapped to page tables, & page tables are mapped to individual pages within a segment

Adv

- less memory usage
- more flexibility on page sizes
- simplified memory allocation
- additional level of data access security over paging.
- does not cause external fragmentation unlike segmentation