COMPUTER ORGANIZATION AND ARCHITECTURE (COA)

EET 2211
4TH SEMESTER – CSE & CSIT
CHAPTER 8, LECTURE 30

CHAPTER 8 – OPERATING SYSTEM SUPPORT

TOPICS TO BE COVERED

- Memory Management

LEARNING OBJECTIVES

- Understand the reason for memory partitioning.
- Explain the various techniques used for memory partitioning.
- Assess the relative advantages of paging and segmentation.

ALREADY COVERED

- Operating system overview
- Scheduling

Contd.

TYPES OF SCHEDULING

TYPES OF SCHEDULING	OPERATION
Long term scheduling	The decision to add to the pool of processes to be executed.
Medium term scheduling	The decision to add to the number of processors that are partially or fully in main memory.
Short term scheduling	The decision as to which available process will be executed by the processor.
I/O scheduling	The decision as to which process is pending I/O request shall be handled by an available I/O device.

MEMORY MANAGEMENT

- It refers to management of primary/main memory.
- The task of sub-division of the memory to accommodate multiple processes in the multi-programming systems is carried out dynamically by the OS and is known as Memory management.

❖ SWAPPING

- ✓ It is an efficient way of memory management.
- ✓ It is an I/O operation which enhances performance of the system.

Contd.

- We have a long-term queue of process requests stored on disk.
- When processes are completed they are moved out of main memory.
- If none of the processes in memory are in ready state, not idle, the processor swaps one of these processes back out to disk into an intermediate queue.
- It is a queue of existing processes that have been temporarily out of the memory.
- Then execution starts with a new process form long-term queue.

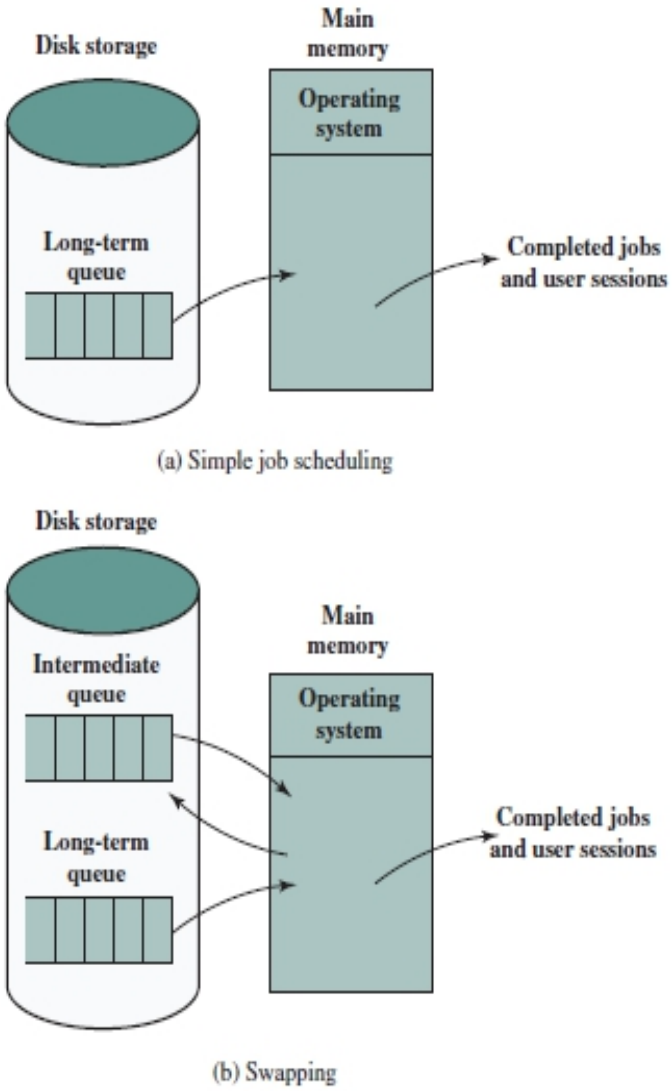


Fig.1: The use of Swapping [Source: Computer Organization and Architecture by William Stallings]

Contd.

PARTITIONING

- Simplest method is to use **fixed-size partitions.**
  - Partitions are of fixed size but not equal in size.
  - A process coming into the memory is placed in the smallest available partition that will hold it.
  - With the use of unequal fixed-size partitions there is wastage of memory.
  - E.g. a process that requires 3M bytes of memory would be placed in 4M partition, wasting 1M.

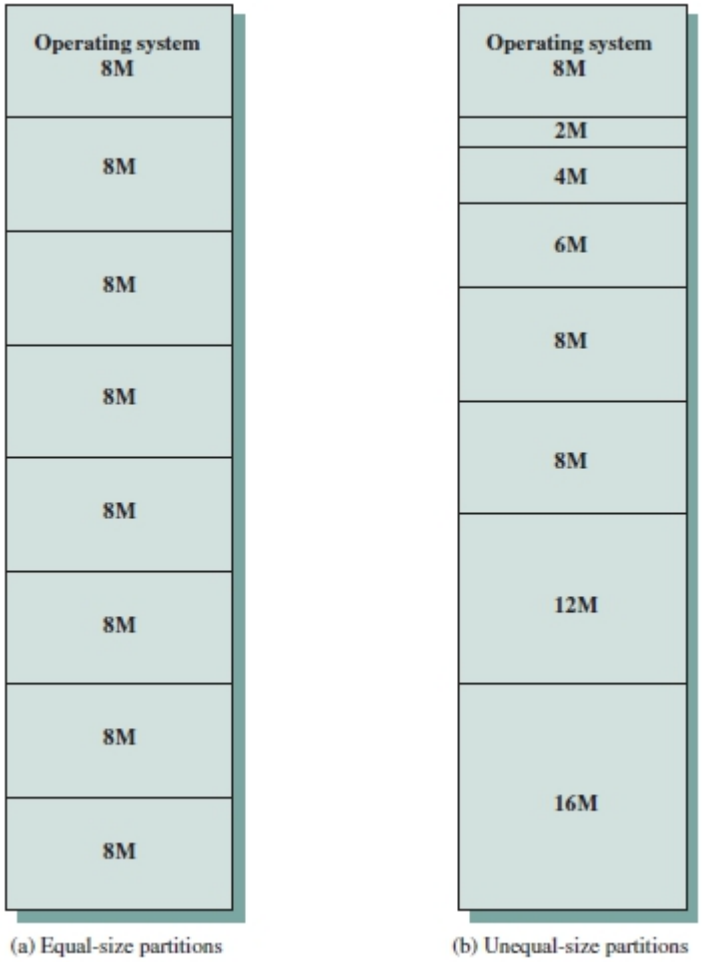


Fig.2: Example of Fixed Partitioning of 64-Mbyte Memory [Source: Computer Organization and Architecture by William Stallings]

Contd.

- Another efficient way is to use **variable-size partitions.**
- When a process is coming into memory it is allocated exactly as much memory as it requires.
- An example using 64 Mbytes of main memory is shown in the figure3 .

Contd.

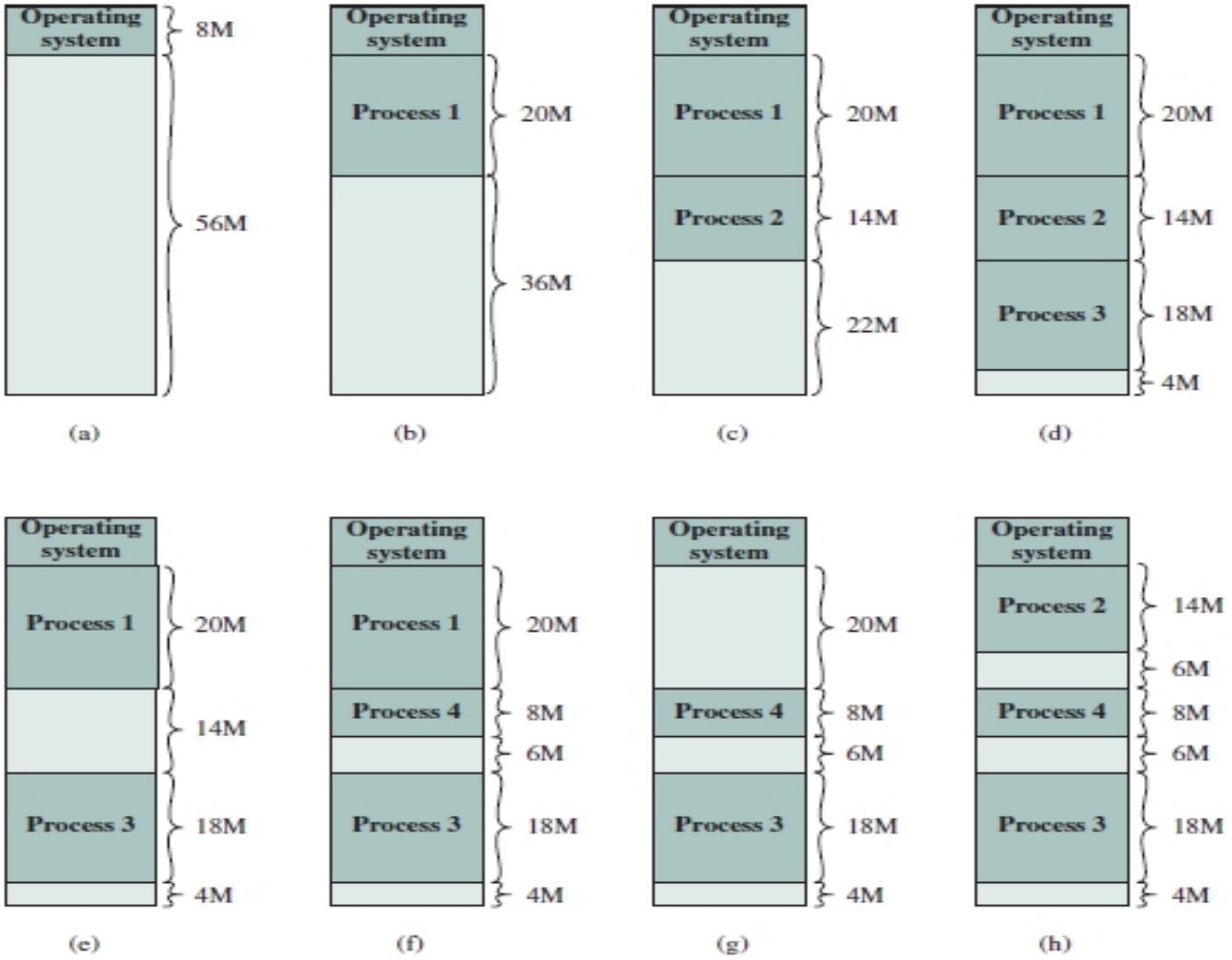


Fig. 3: The effect of Dynamic Partitioning [Source: Computer Organization and Architecture by William Stallings]

Contd.

- **PAGING**
  - Memory and each process is partitioned into equal and small fixed-size pages.
  - Pages are assigned to memory chunks known as frames or page frames.

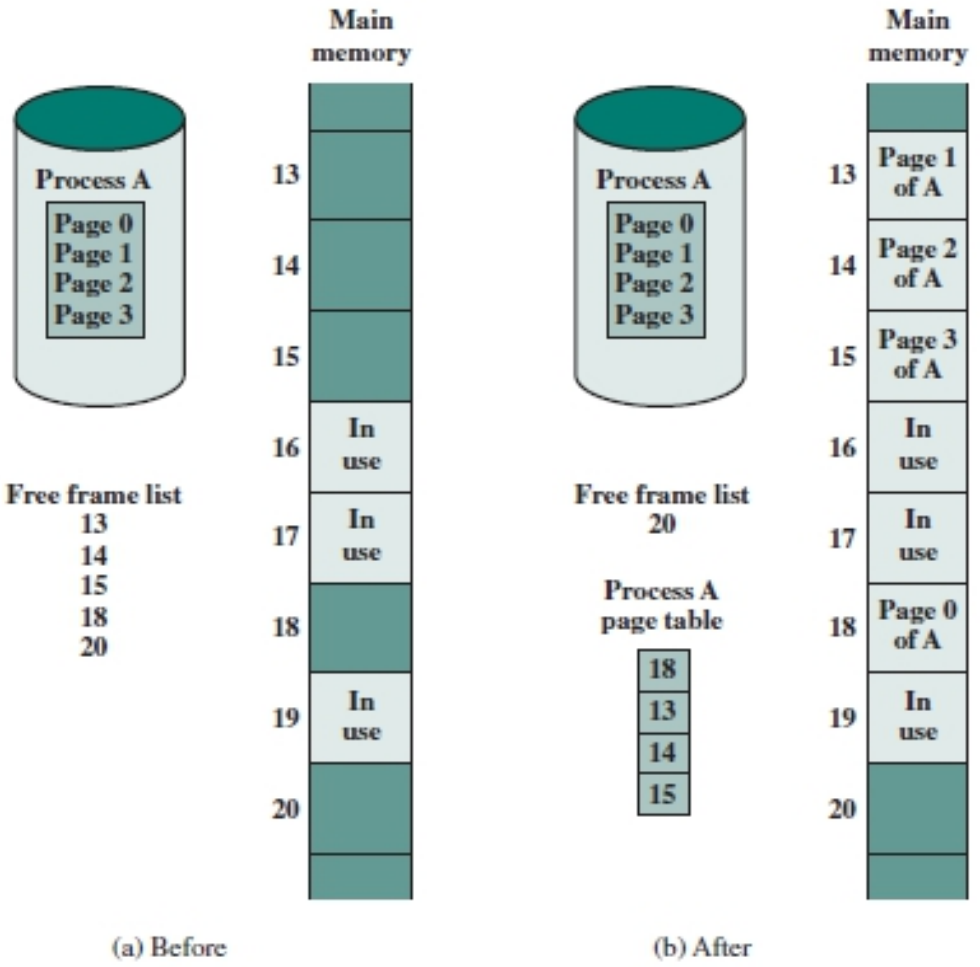


Fig.4: Allocation of Free Frames [Source: Computer Organization and Architecture by William Stallings]

Contd.

- OS maintains a page table for each process.
- Page table shows the frame location for each page of the process.
- Logical address is the location of a word relative to the beginning of the program.
- The processor uses the page table and produces a physical address (frame number and relative address).

Contd.

VIRTUAL MEMORY

1. DEMAND PAGING

- It means that each page of a process is brought in only when it is needed i.e. on demand.
- It facilitates not to load an entire process into main memory.
- With demand paging, OS and the hardware devices ways to structure the program into pieces that can be loaded one at a time for too large programs.
- When a process executes only in main memory it is referred to as **real memory**.
- When a programmer or user perceives a larger memory it is known as **Virtual memory**. It allows for very effective multiprogramming.

Contd.

- ✓ When a process is running a register holds the starting address of the page table for that process.
- ✓ The page number of a virtual address is used to index that table and look up the corresponding frame number.
- ✓ Frame number combined with offset of virtual address results in real address.
- ✓ Virtual memory schemes store page tables in virtual memory rather than real memory.

Contd.

VIRTUAL MEMORY

2. PAGE TABLE STRUCTURE

- It is a basic mechanism for reading a word form memory.
- It involves the translation of a virtual or logical address consisting of page number and offset into a physical address consisting of frame number and offset.
- It is of variable length depending on the size of a process.

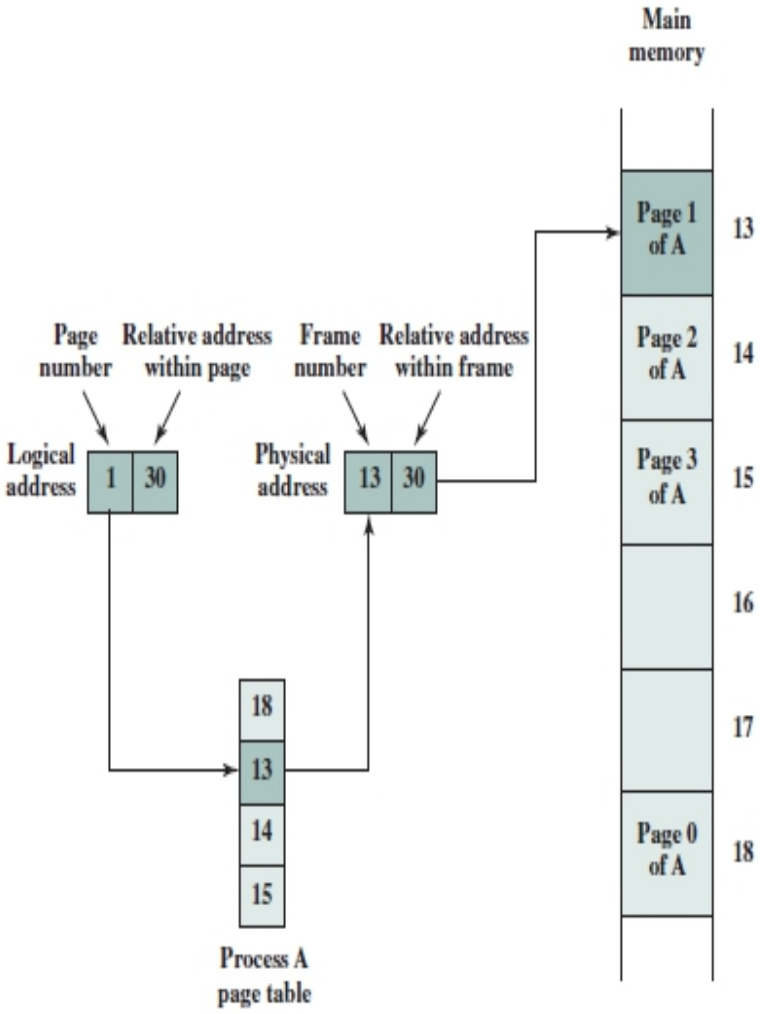


Fig.5: Logical and Physical Addresses [Source: Computer Organization and Architecture by William Stallings]

Contd.

TRANSLATION LOOKASIDE BUFFER

- TLB are virtual memory schemes that make use of special cache for page table entries.
- This cache functions in the same way as a memory cache and contains the page table entries that have been most recently used.
- By the principle of locality, most virtual memory references will be to locations in recently used pages.
- Most references will involve page table entries in the cache.

Contd.

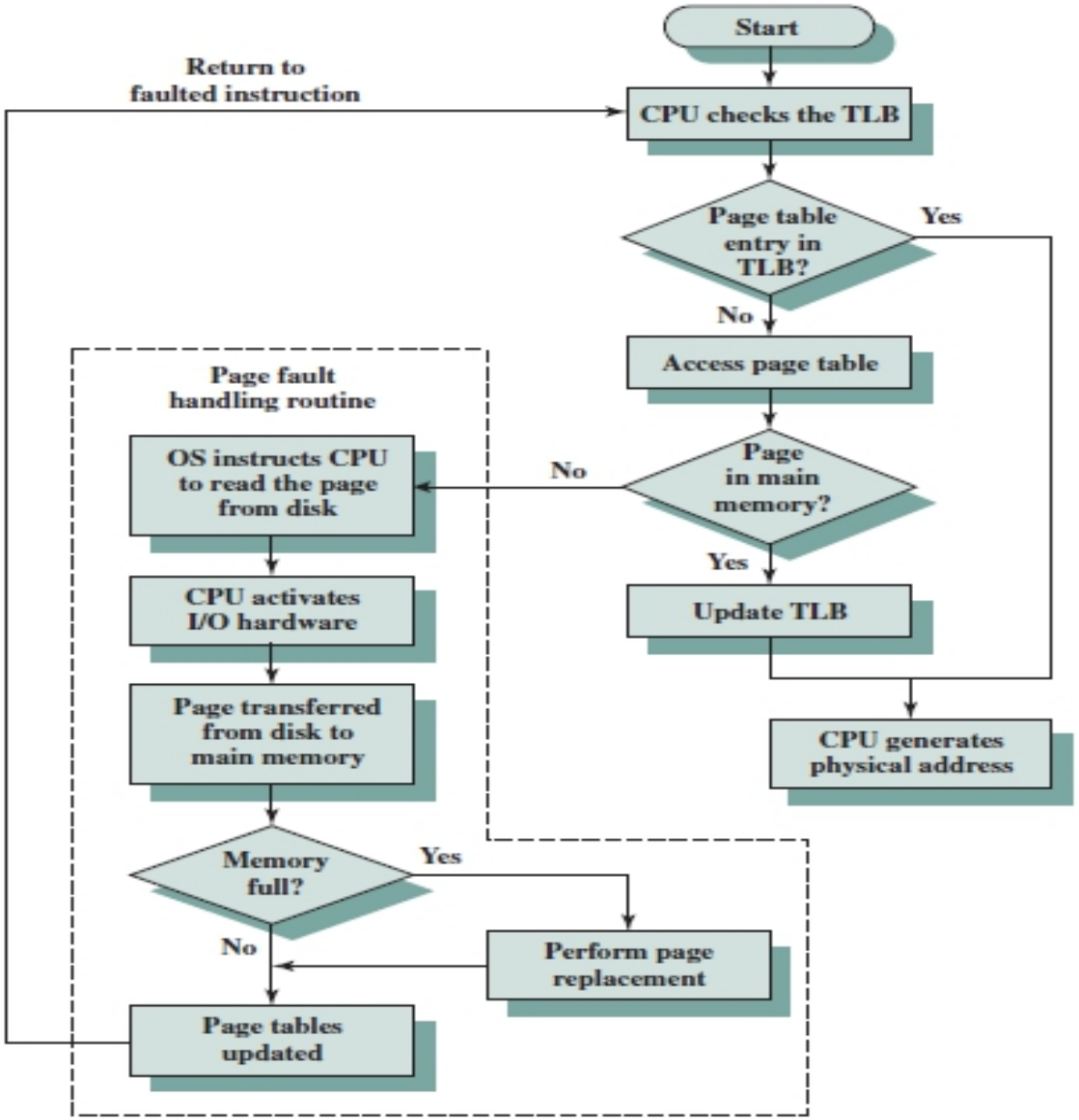


Fig.6: Operation of Paging and Translation Lookaside Buffer (TLB) [Source: Computer Organization and Architecture by William Stallings]

Contd.

- A virtual address is in the form of a page number, offset.
- The memory consults the TLB to see if the matching page table entry is present.
- If present then physical address is generated.
- If not then entry is accessed form a page table.

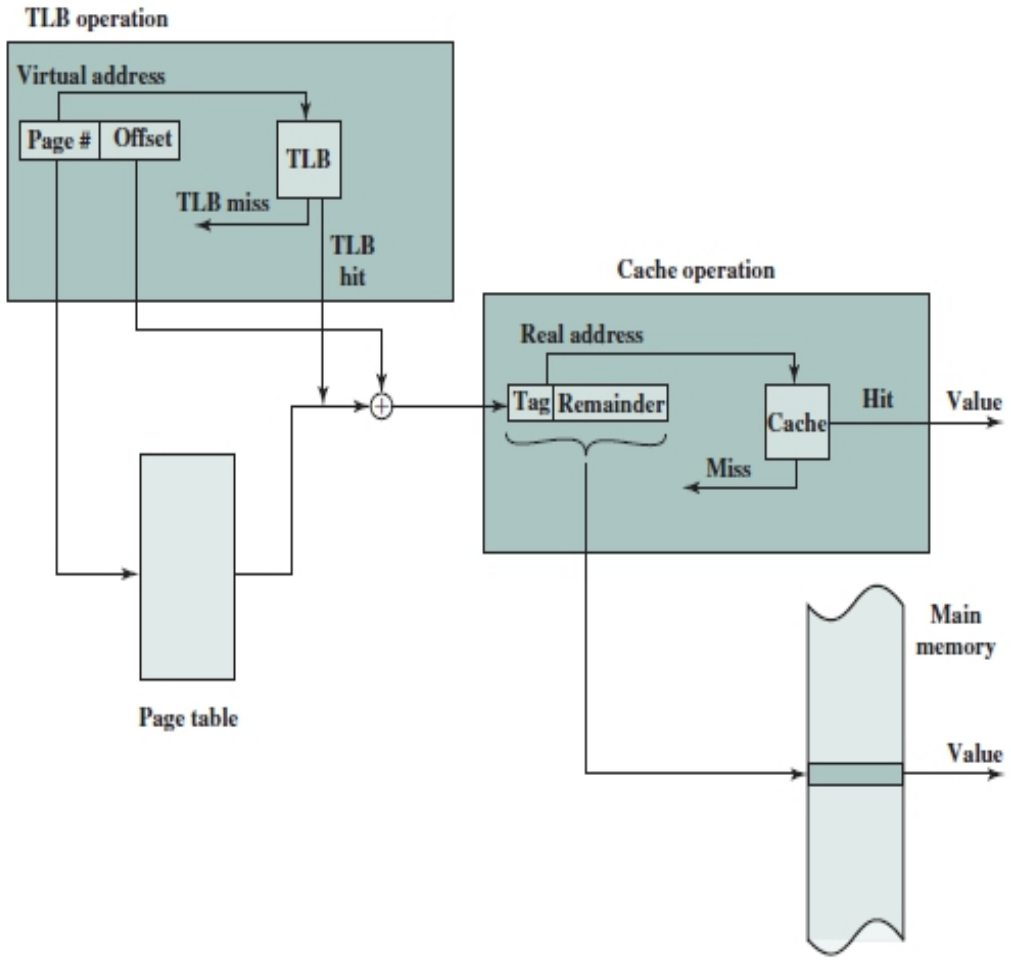


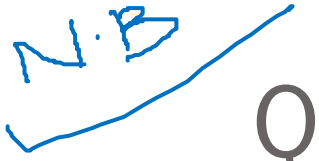
Fig.7: Translation Lookaside Buffer and Cache Operation [Source: Computer Organization and Architecture by William Stallings]

Contd.

- **SEGMENTATION**
- It is another way in which addressable memory can be sub-divided.
- It is only visible to the programmer.
- It helps in organizing programs and data.
- It acts as a means for associating privilege and protection attributes with instruction and data.
- It allows the programmer to view memory as consisting of multiple address spaces or segments.
- Segments are of variable dynamic size.
- OS assigns programs and data to different segments.

ADVANTAGES OF SEGMENTATON OVER NON-SEGMENTED ADDRESS SPACE

1. It simplifies the handling of growing data structures.
2. It allows the programs to be altered and recompiled independently without requiring an entire set of programs to be relinked and reloaded.
3. It lends itself to sharing among processors.
4. It helps in protection of the system administrator.



QUESTIONS

1. What is the difference between a process and a program?
2. What is the purpose of swapping?
3. If a process may be dynamically assigned to different locations in main memory, what is the implication for the addressing mechanism?
4. Is it necessary for all of the pages of a process to be in main memory while the process is executing?
5. Must the pages of a process in main memory be contiguous?
6. Is it necessary for the pages of a process in main memory to be in sequential order?
7. What is the purpose of a translation lookaside buffer?
8. Describe exactly how a virtual address generated by the CPU is translated into a physical main memory address?
9. Give reasons that the page size in a virtual memory system should be neither very small nor very large.

THANK YOU