MODULE 5

Module V (8 Hours)

The Memory System – basic concepts, semiconductor RAM memories - organization – static and dynamic RAM, Structure of larger memories, semiconductor ROM memories, Speed, Size and cost, Cache memory – mapping functions – replacement algorithms, Virtual memory – paging and segmentation.

- Hamacher, Vranesic & Zaky, “Computer Organization” (5th Ed), McGraw Hill.

VIRTUAL MEMORY

- In most modern computer systems, the physical main memory is not as large as the address space of the processor.
  - For example, a processor that issues 32-bit addresses has an addressable space of 4G bytes. The size of the main memory in a typical computer with a 32- bit processor may range from 1G to 4G bytes.
- If a program does not completely fit into the main memory, the parts of it not currently being executed are stored on a secondary storage device, typically a magnetic disk.
- As these parts are needed for execution, they must first be brought into the main memory, possibly replacing other parts that are already in the memory.
- These actions are performed automatically by the operating system, using a scheme known as virtual memory.

VIRTUAL MEMORY

- Application programmers need not be aware of the limitations imposed by the available main memory.
- Under a virtual memory system, programs and hence the processor, reference instructions and data in an address space that is independent of the available physical main memory space.
- The binary addresses that the processor issues for either instructions or data are called virtual or logical addresses.
- These addresses are translated into physical addresses by a combination of hardware and software components.
- If a virtual address refers to a part of the program or data space that is currently in the physical memory, then the contents of the appropriate location in the main memory are accessed immediately.
- Otherwise, the contents of the referenced address must be brought into a suitable location in the memory before they can be used.

VIRTUAL MEMORY

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- Figure 15 shows a typical organization that implements virtual memory.
- A special hardware unit, called the Memory Management Unit (MMU), keeps track of which parts of the virtual address space are in the physical memory.
- When the desired data or instructions are in the main memory, the MMU translates the virtual address into the corresponding physical address.
- Then, the requested memory access proceeds in the usual manner.

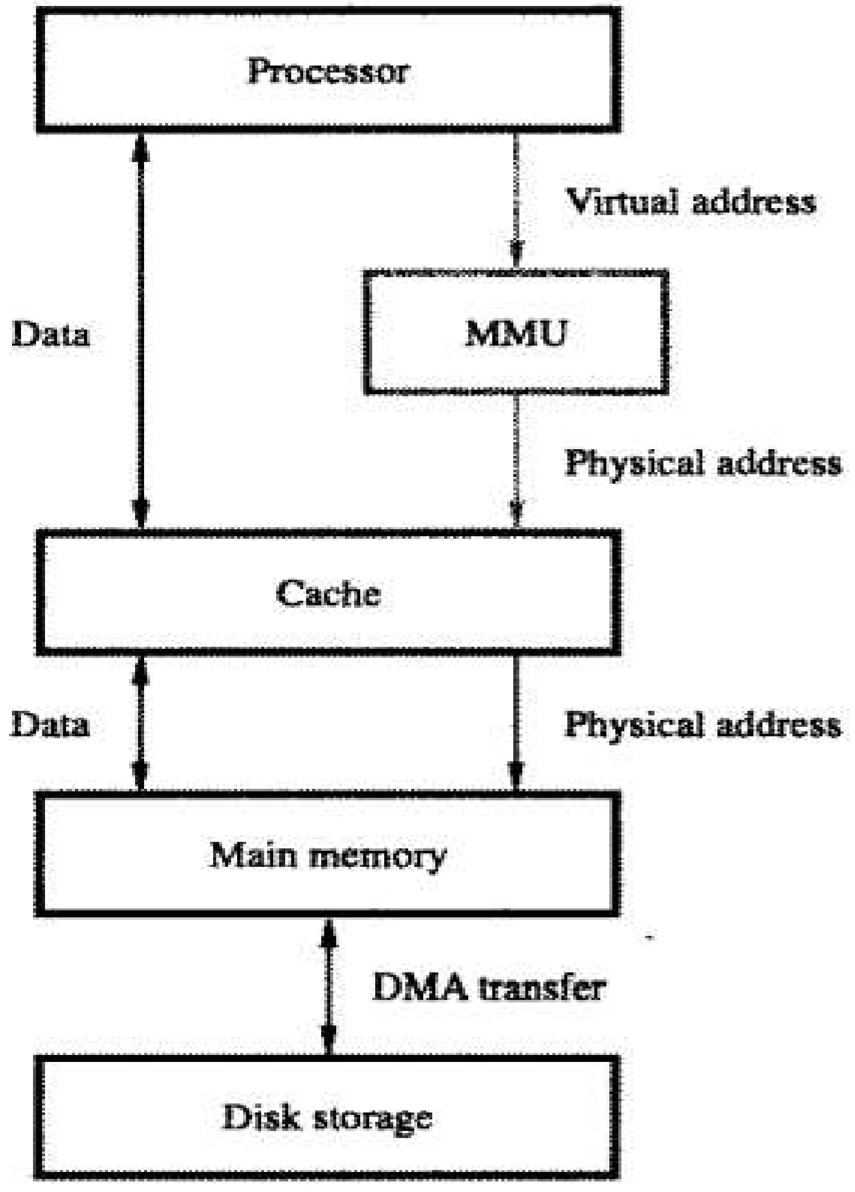

Figure 15 Virtual memory organization.

VIRTUAL MEMORY

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- If the data are not in the main memory, the MMU causes the operating system to transfer the data from the disk to the memory.

- Such transfers are performed using the DMA scheme.

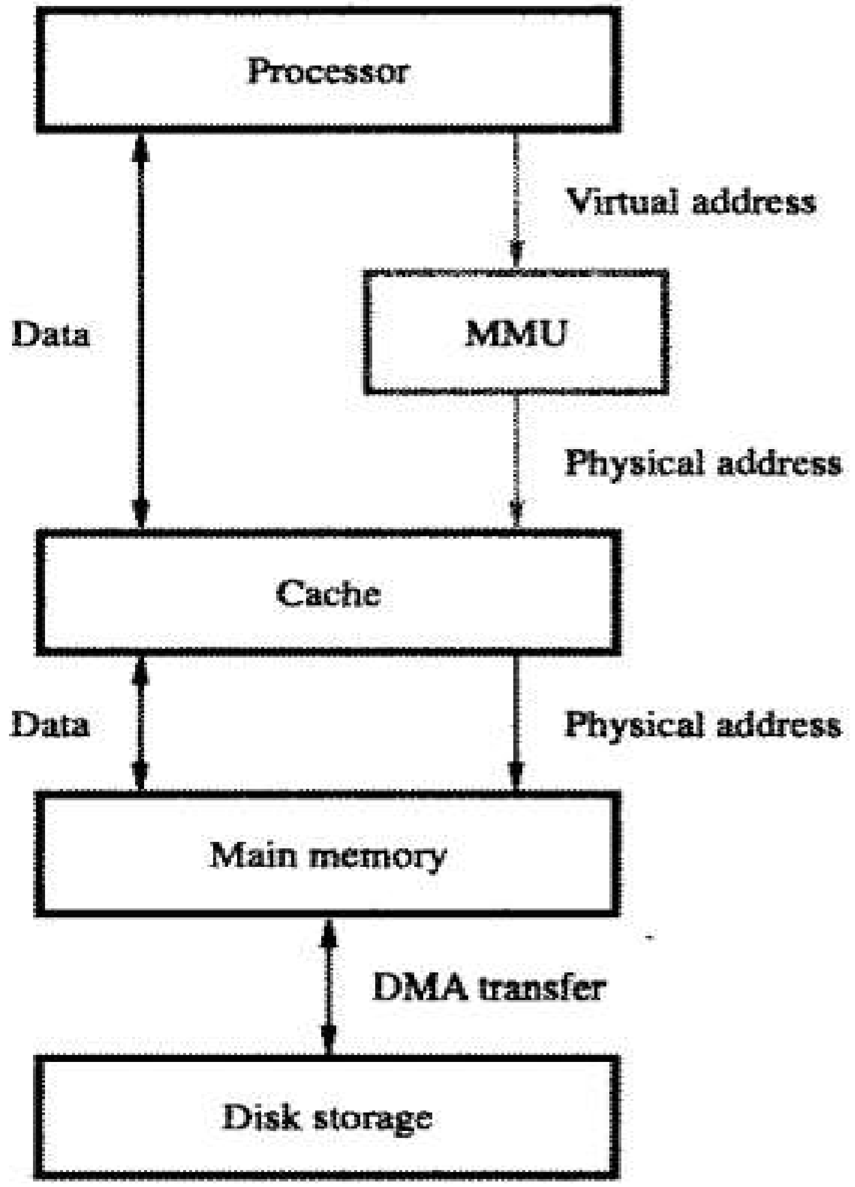

Figure 15 Virtual memory organization.

Address Translation

- A simple method for translating virtual addresses into physical addresses is to assume that all programs and data are composed of fixed-length units called pages, each of which consists of a block of words that occupy contiguous locations in the main memory.
- Pages commonly range from 2K to 16K bytes in length.
- They constitute the basic unit of information that is transferred between the main memory and the disk whenever the MMU determines that a transfer is required.

Address Translation

- Pages should not be too small, because the access time of a magnetic disk is much longer (several milliseconds) than the access time of the main memory.
- The reason for this is that it takes a considerable amount of time to locate the data on the disk, but once located, the data can be transferred at a rate of several megabytes per second.
- On the other hand, if pages are too large, it is possible that a substantial portion of a page may not be used, yet this unnecessary data will occupy valuable space in the main memory.

Address Translation

- The cache bridges the speed gap between the processor and the main memory and is implemented in hardware.
- The virtual-memory mechanism bridges the size and speed gaps between the main memory and secondary storage and is usually implemented in part by software techniques.
- Conceptually, cache techniques and virtual-memory techniques are very similar. They differ mainly in the details of their implementation.

Address Translation

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- A virtual-memory address-translation method based on the concept of fixed-length pages is shown schematically in Figure 16.
- Each virtual address generated by the processor, whether it is for an instruction fetch or an operand load/store operation, is interpreted as a virtual page number (high-order bits) followed by an offset (low-order bits) that specifies the location of a particular byte (or word) within a page.

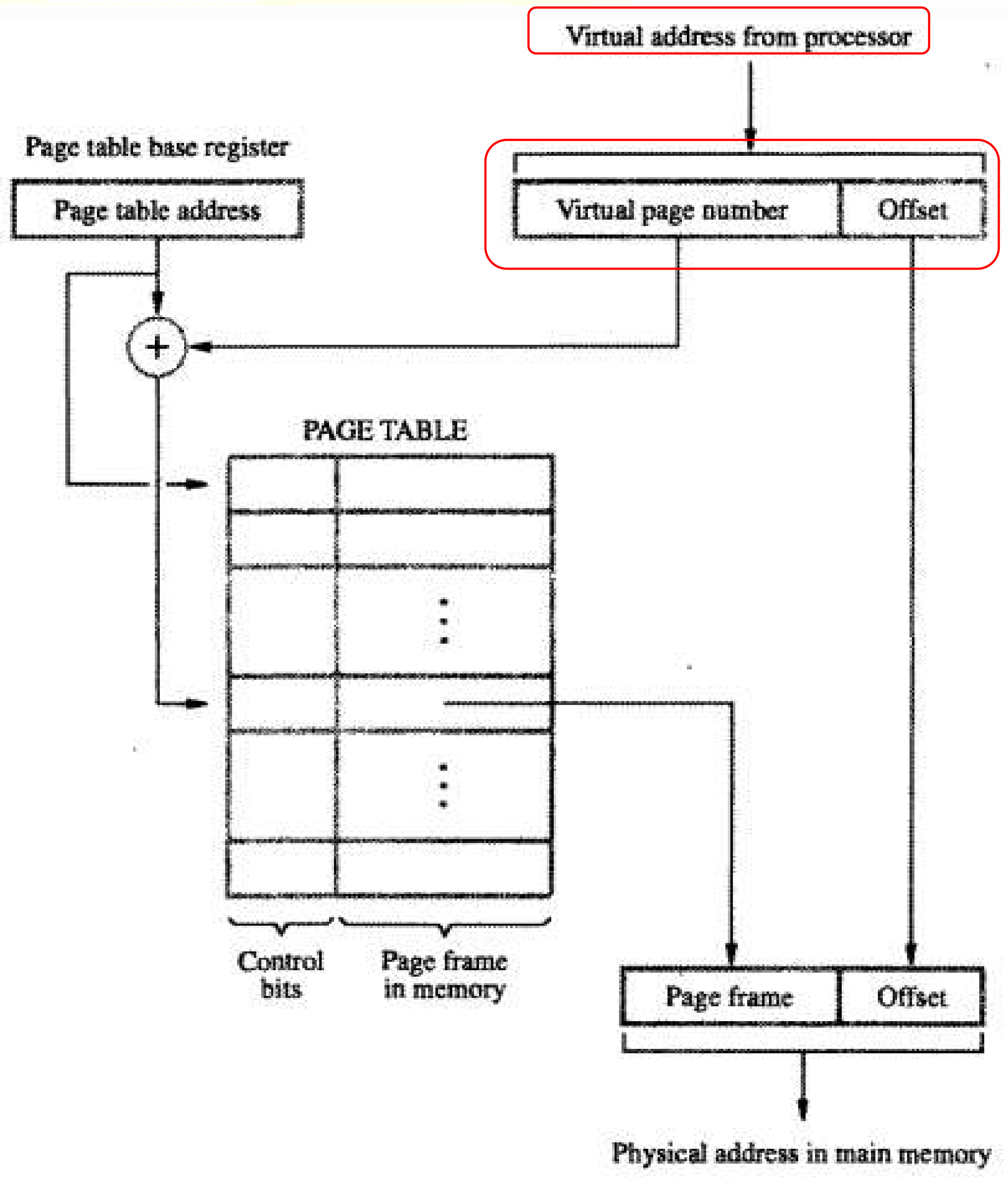

Figure 16 Virtual-memory address translation.

Address Translation

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- Information about the main memory location of each page is kept in a page table.
- This information includes the main memory address where the page is stored and the current status of the page.
- An area in the main memory that can hold one page is called a page frame.

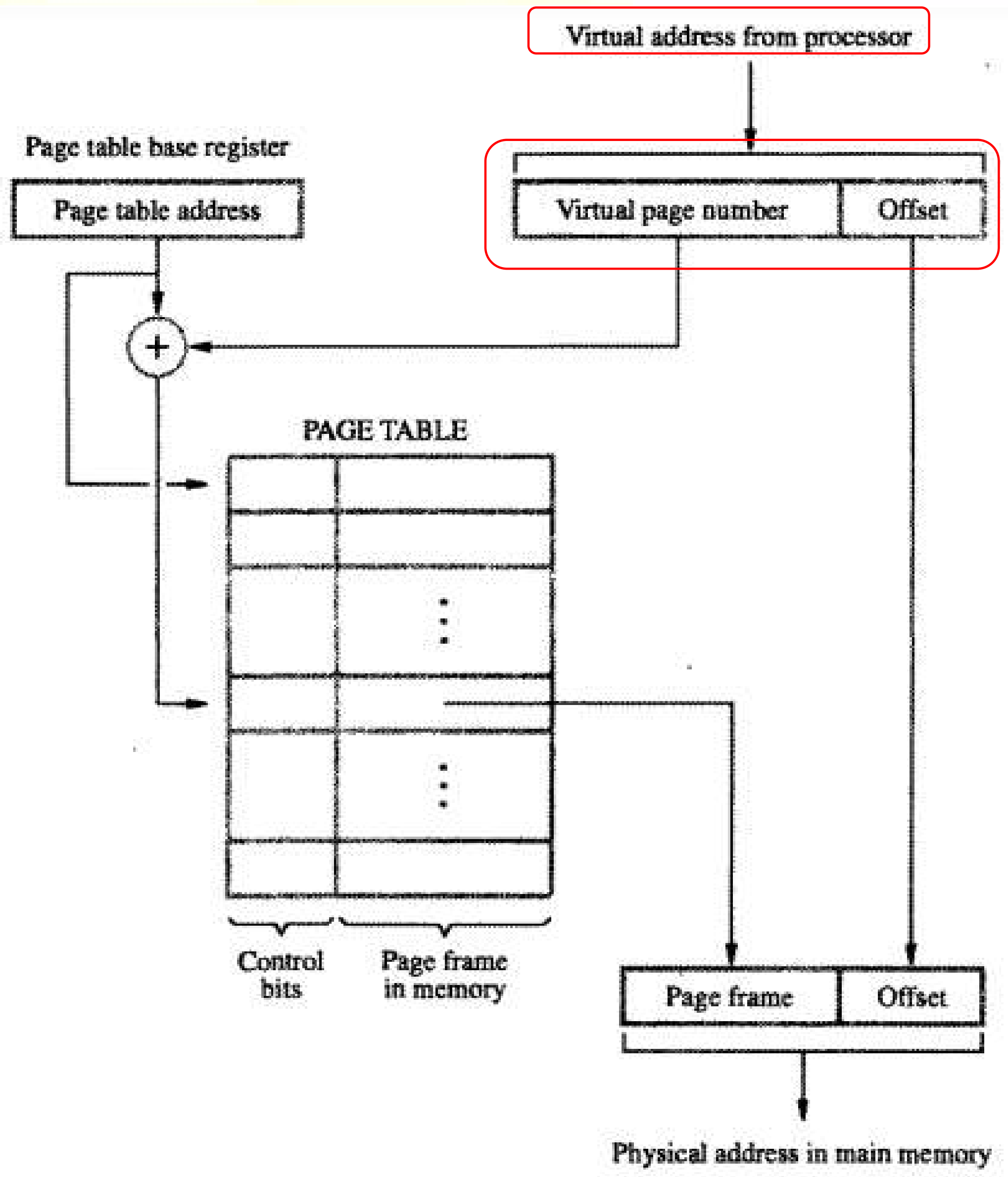

Figure 16 Virtual-memory address translation.

Address Translation

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- The starting address of the page table is kept in a page table base register.
- By adding the virtual page number to the contents of this register, the address of the corresponding entry in the page table is obtained.
- The contents of this location give the starting address of the page if that page currently resides in the main memory.

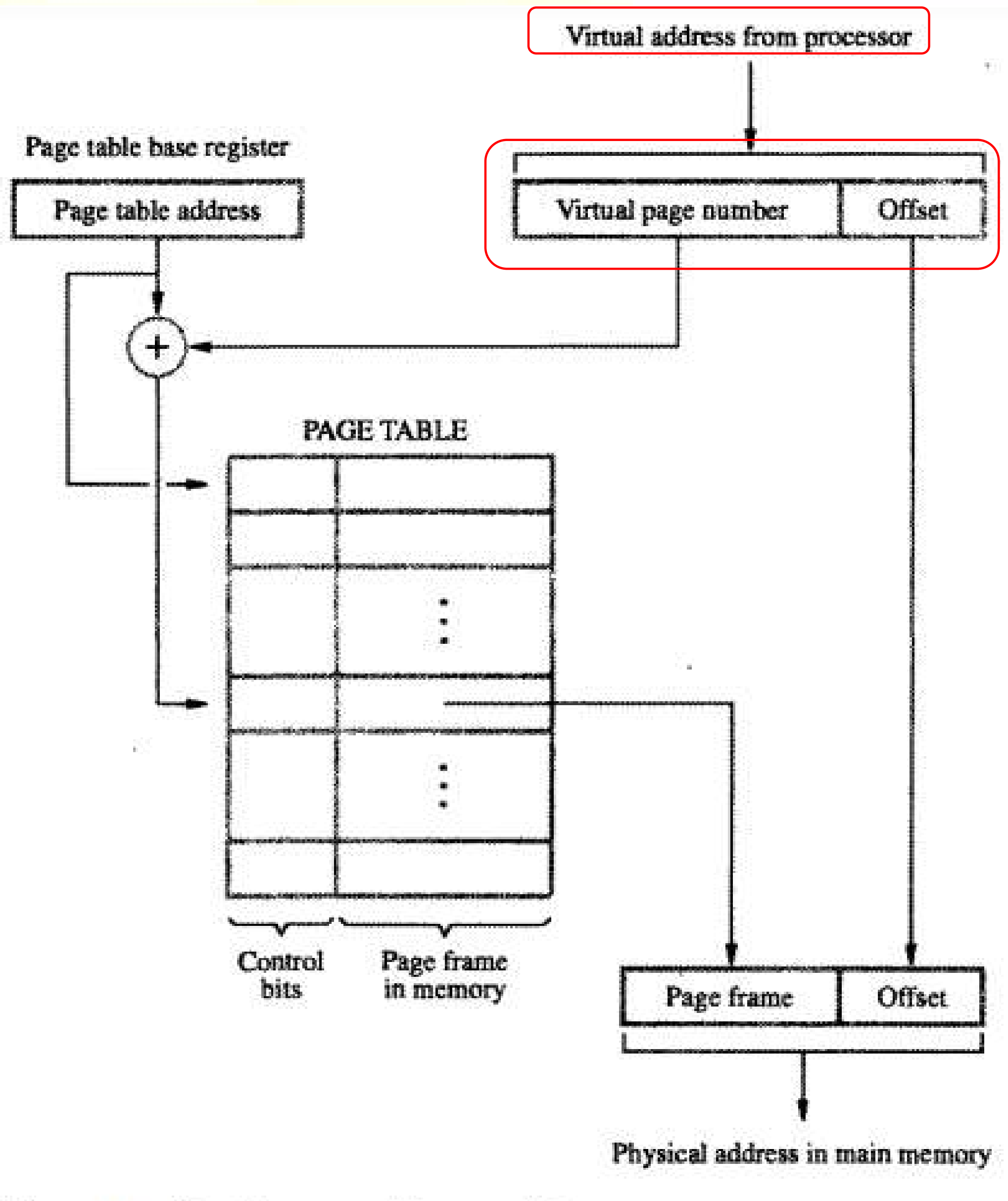

Figure 16 Virtual-memory address translation.

Address Translation

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- Each entry in the page table also includes control bits that describe the status of the page while it is in the main memory.
- One bit indicates the validity of the page, that is, whether the page is actually loaded in the main memory.
- It allows the operating system to invalidate the page without actually removing it.

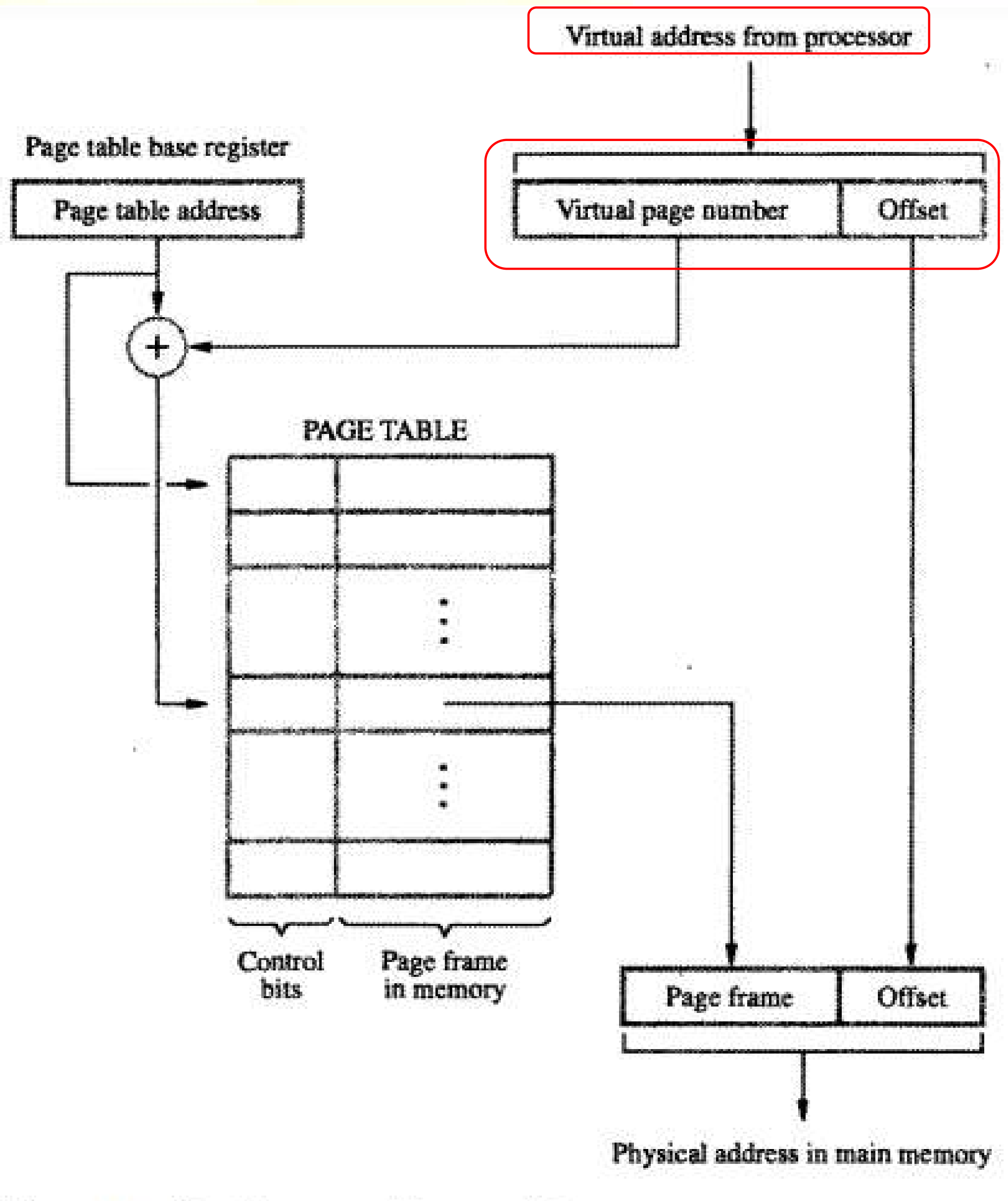

Figure 16 Virtual-memory address translation.

Address Translation

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

- Another bit indicates whether the page has been modified during its residency in the memory.
- As in cache memories, this information is needed to determine whether the page should be written back to the disk before it is removed from the main memory to make room for another page.
- Other control bits indicate various restrictions that may be imposed on accessing the page.
- For example, a program may be given full read and write permission, or it may be restricted to read accesses only.

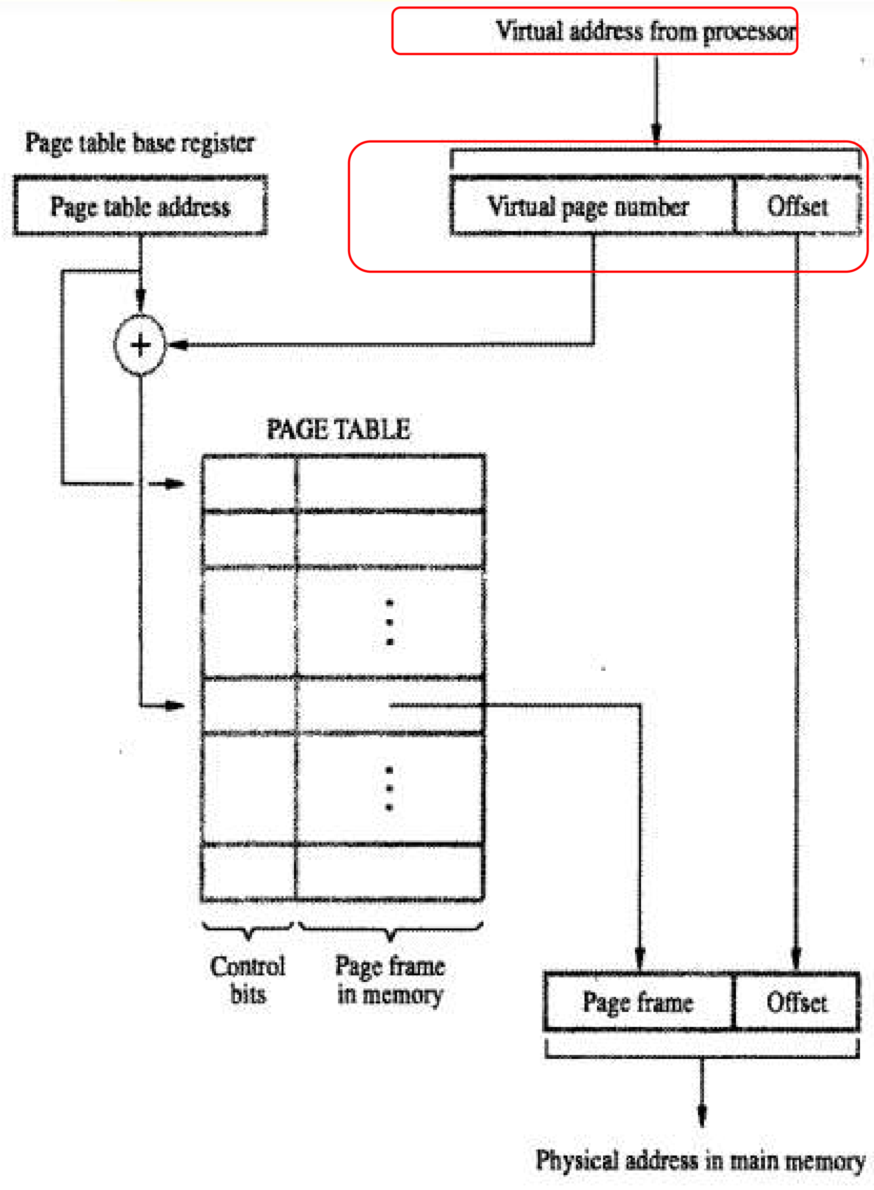

Figure 16 Virtual-memory address translation.

Translation Lookaside Buffer

- The page table information is used by the MMU for every read and write access.
- Ideally, the page table should be situated within the MMU.
- Unfortunately, the page table may be rather large and since the MMU is normally implemented as part of the processor chip, it is impossible to include the complete table within the MMU.
- Instead, a copy of only a small portion of the table is accommodated within the MMU, and the complete table is kept in the main memory.

Translation Lookaside Buffer

- The portion maintained within the MMU consists of the entries corresponding to the most recently accessed pages.
- They are stored in a small table, usually called the Translation Lookaside Buffer (TLB).
- The TLB functions as a cache for the page table in the main memory.
- Each entry in the TLB includes a copy of the information in the corresponding entry in the page table.
- In addition, it includes the virtual address of the page, which is needed to search the TLB for a particular page.

Translation Lookaside Buffer

- Figure 17 shows a possible organization of a TLB that uses the associative mapping technique.
- Set-associative mapped TLBs are also found in commercial products.
- The contents of the TLB be coherent with the contents of page tables in the memory.

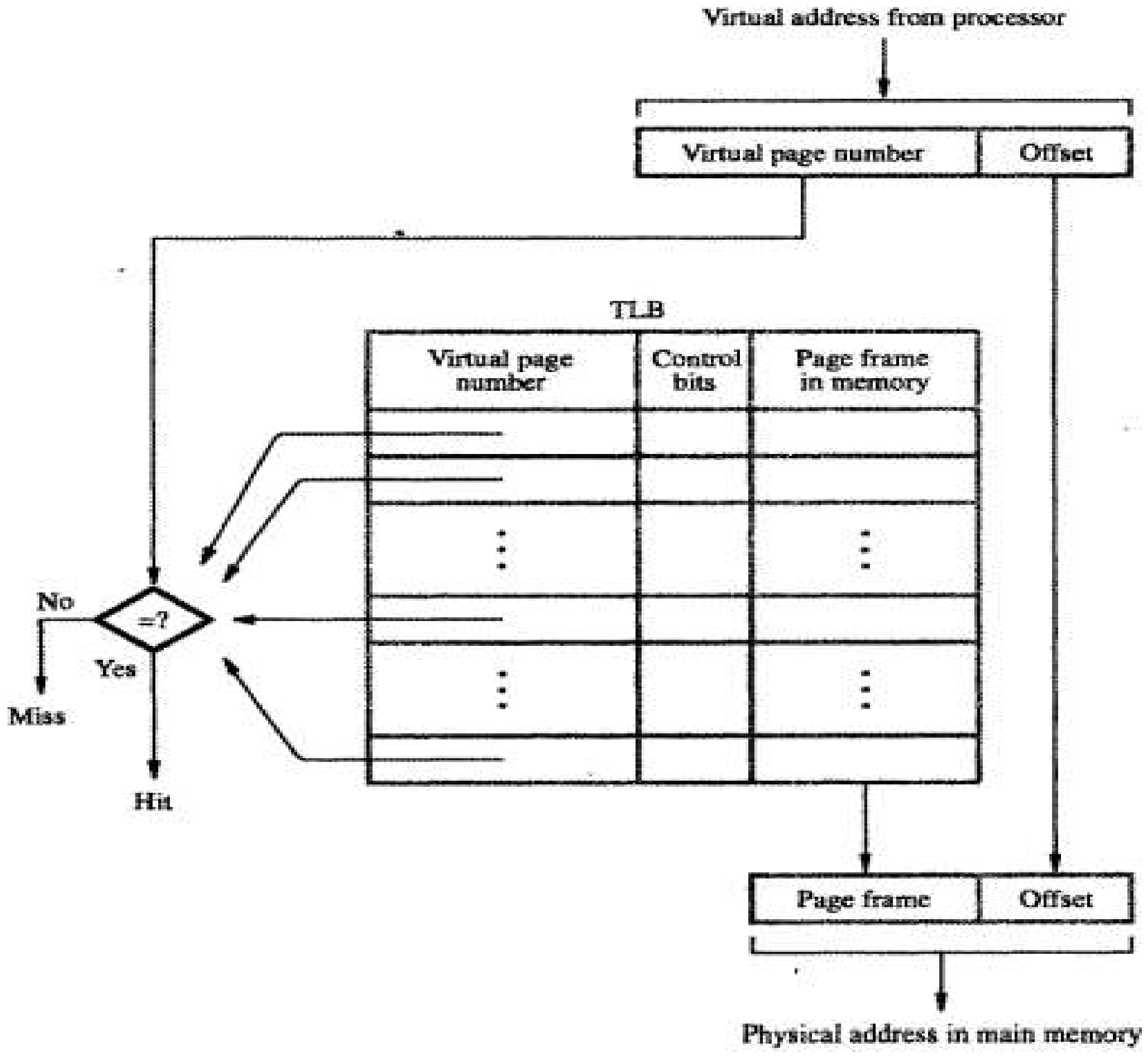

Figure 17 Use of an associative-mapped TLB.

[Hamacher, Vranesic & Zaky, "Computer Organization" (5th Ed), McGraw Hill]

Translation Lookaside Buffer

- It is essential to ensure that the contents of the TLB are always the same as the contents of page tables in the memory.
- When the operating system changes the contents of a page table, it must simultaneously invalidate the corresponding entries in the TLB.
- One of the control bits in the TLB is provided for this purpose.
- When an entry is invalidated, the TLB acquires the new information from the page table in the memory as part of the MMU’s normal response to access misses.

Translation Lookaside Buffer

Address translation proceeds as follows.

- Given a virtual address, the MMU looks in the TLB for the referenced page.
- If the page table entry for this page is found in the TLB, the physical address is obtained immediately.
- If there is a miss in the TLB, then the required entry is obtained from the page table in the main memory and the TLB is updated.

Page Faults

- When a program generates an access request to a page that is not in the main memory, a page fault is said to have occurred.
- The entire page must be brought from the disk into the memory before access can proceed.
- When it detects a page fault, the MMU asks the operating system to intervene by raising an exception (interrupt).
- Processing of the program that generated the page fault is interrupted, and control is transferred to the operating system.
- The operating system copies the requested page from the disk into the main memory.
- Since this process involves a long delay, the operating system may begin execution of another program whose pages are in the main memory.
- When page transfer is completed, the execution of the interrupted program is resumed.

Page Faults

- When the MMU raises an interrupt to indicate a page fault, the instruction that requested the memory access may have been partially executed.
- It is essential to ensure that the interrupted program continues correctly when it resumes execution.
- There are two options. Either the execution of the interrupted instruction continues from the point of interruption, or the instruction must be restarted.
- The design of a particular processor dictates which of these two options is used.

Page Faults

- If a new page is brought from the disk when the main memory is full, it must replace one of the resident pages.
- The problem of choosing which page to remove is just as critical here as it is in a cache and the observation that programs spend most of their time in a few localized areas also applies.
- Because main memories are considerably larger than cache memories, it should be possible to keep relatively larger portions of a program in the main memory.
- This reduces the frequency of transfers to and from the disk.

Page Faults

- Concepts similar to the LRU replacement algorithm can be applied to page replacement and the control bits in the page table entries can be used to record usage history.
- One simple scheme is based on a control bit that is set to 1 whenever the corresponding page is referenced (accessed).
- The operating system periodically clears this bit in all page table entries, thus providing a simple way of determining which pages have not been used recently.

Page Faults

- A modified page has to be written back to the disk before it is removed from the main memory.
- It is important to note that the write-through protocol, which is useful in the framework of cache memories, is not suitable for virtual memory.
- The access time of the disk is so long that it does not make sense to access it frequently to write small amounts of data.

Page Faults

- Looking up entries in the TLB introduces some delay, slowing down the operation of the MMU.
- Here again we can take advantage of the property of locality of reference.
- It is likely that many successive TLB translations involve addresses on the same program page. This is particularly likely when fetching instructions.
- Thus, address translation time can be reduced by keeping the most recently used TLB entries in a few special registers that can be accessed quickly.