# 第30 - 31讲 存储划分技术：简单分页技术

# Simple Paging Technique

- Partition memory into small equal-size chunks and divide each process into the same size chunks.

- The chunks of a process are called **pages** and chunks of memory are **called frames** **（页框、页帧）.**

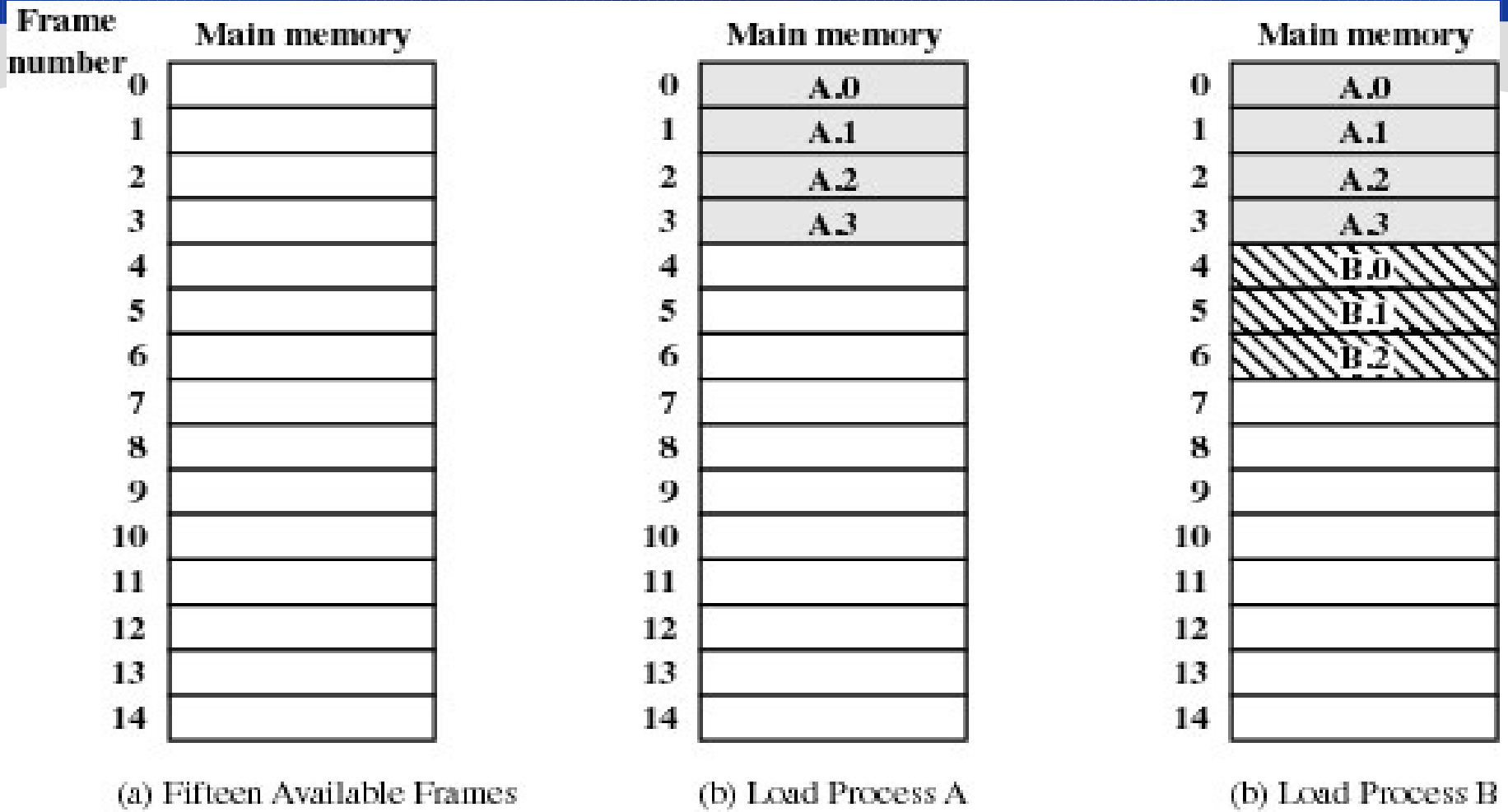



**Figure 7.9 Assignment of Process Pages to Free Frames**

电子科技大学
University of Electronic Science and Technology of China

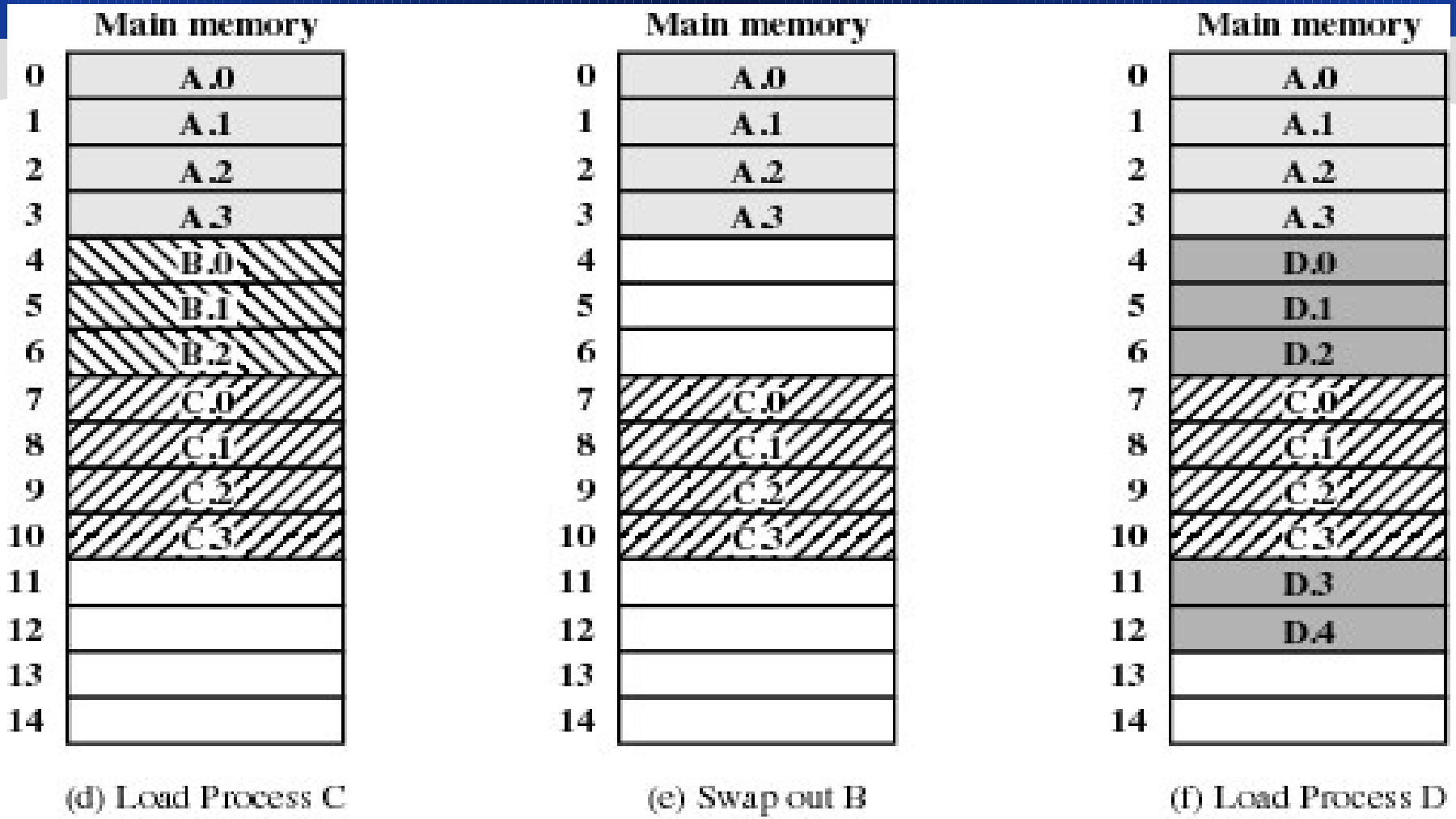



**Figure 7.9 Assignment of Process Pages to Free Frames**

电子科技大学
University of Electronic Science and Technology of China

# Simple Paging Technique （简单分页技术）

- Operating system maintains a page table （页表） for each process.
  - contains the frame location for each page in the process.
  - logical address consists of a page number （页号） and an offset （偏移量） within the page.
- A single free-frame list （空闲页框表） of all frames in main memory that are currently unoccupied and available for pages.

# Page Tables for Example

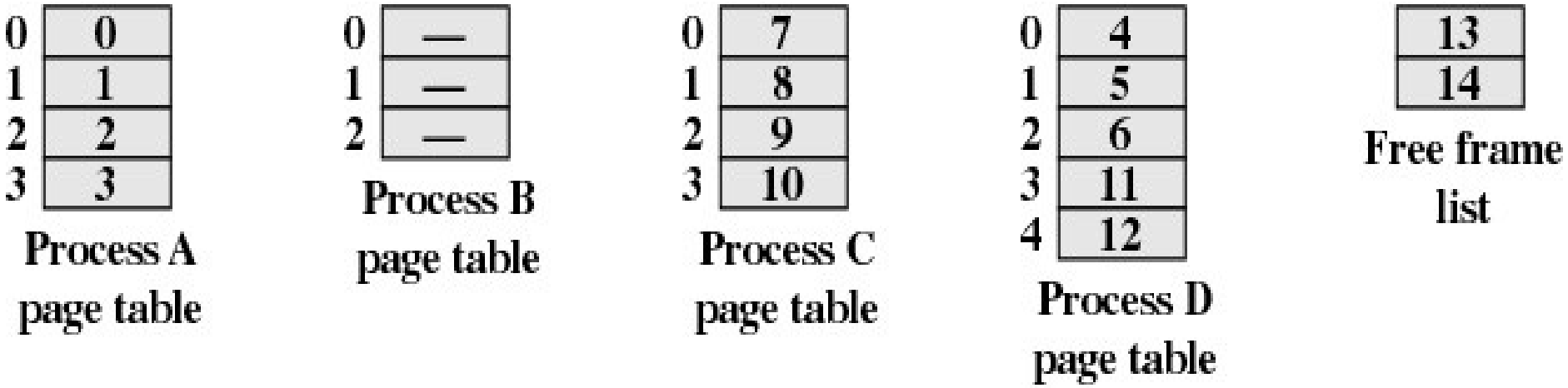



Figure 7.10 Data Structures for the Example of Figure 7.9 at Time Epoch (f)

# Logical-to-Physical Address Translation

- Address translation is done by processor hardware (MMU，存储管理单元).
- The processor must know how to access the page table of the current process, using page table register (*页表寄存器*).
- Logical address :page number, offset.
- Physical address:frame number, offset.

# Logical-to-Physical Address Translation

在分页系统中，地址变换步骤：

- 根据逻辑地址及页大小求出页号和页内偏移；
- 用页号检索页表，查找指定页对应的页框号；
- 根据页框号、页内偏移以及页框尺寸求出物理地址

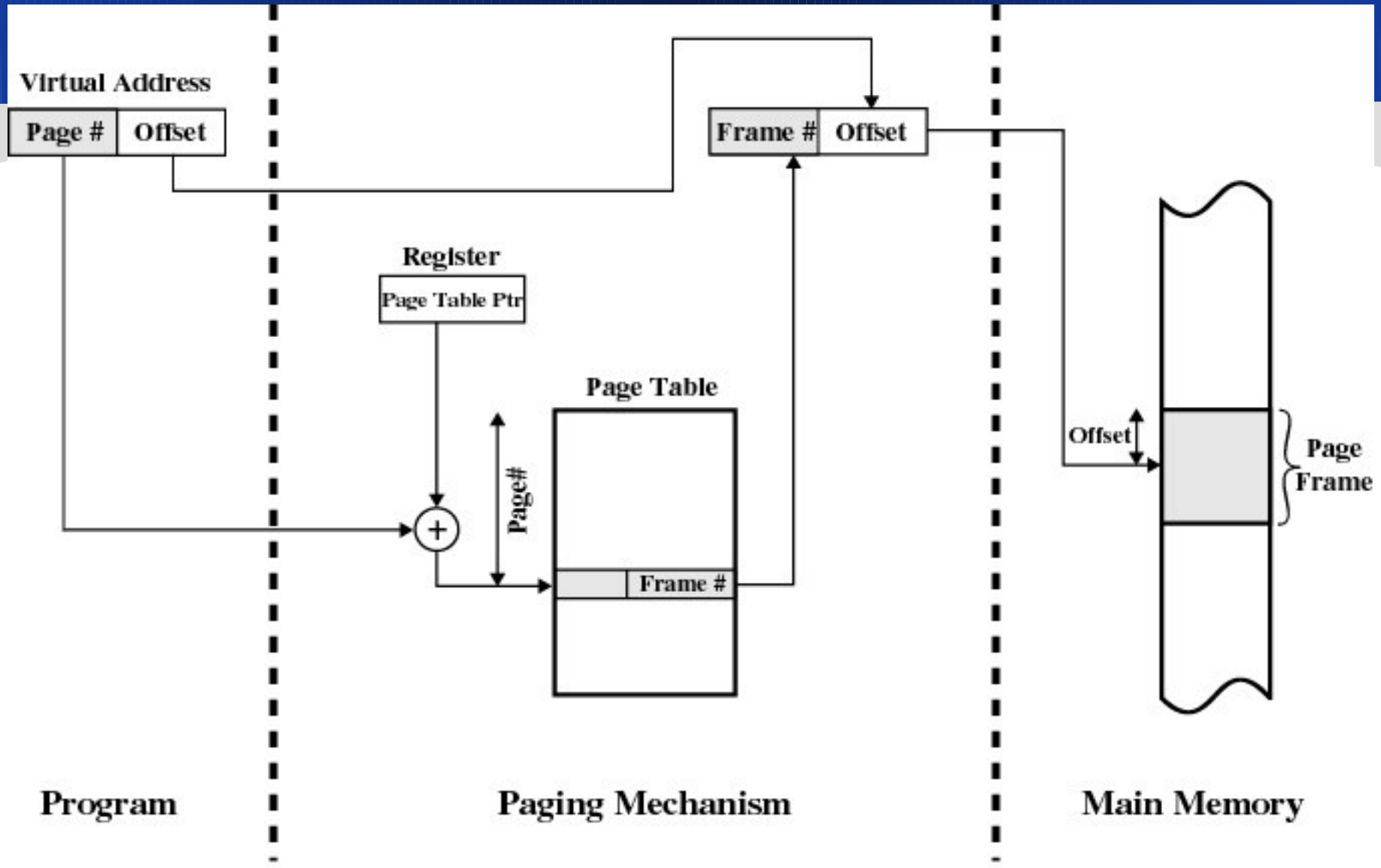



Figure 8.3 Address Translation in a Paging System

# Translation Lookaside Buffer ( 快表 - 加快页表检索速度 )

- Each virtual memory reference can cause two physical memory accesses.
  - one to fetch the page table.
  - one to fetch the data.
- To overcome this problem a high-speed cache is set up for page table entries.
  - called the TLB - Translation Lookaside Buffer.

# Translation Lookaside Buffer

- Contains page table entries that have been most recently used.
- Functions same way as a  memory cache.

# Translation Lookaside Buffer

- Given a virtual address, processor examines the TLB.
- If page table entry is present **(a hit)**, the frame number is retrieved and the real address is formed.
- If page table entry is not found in the TLB **(a miss)**, the page number is used to index the process page table.

# Translation Lookaside Buffer

- First checks if page is already in main memory .
  - if not in main memory a page fault is issued.
- The TLB is updated to include the new page entry.

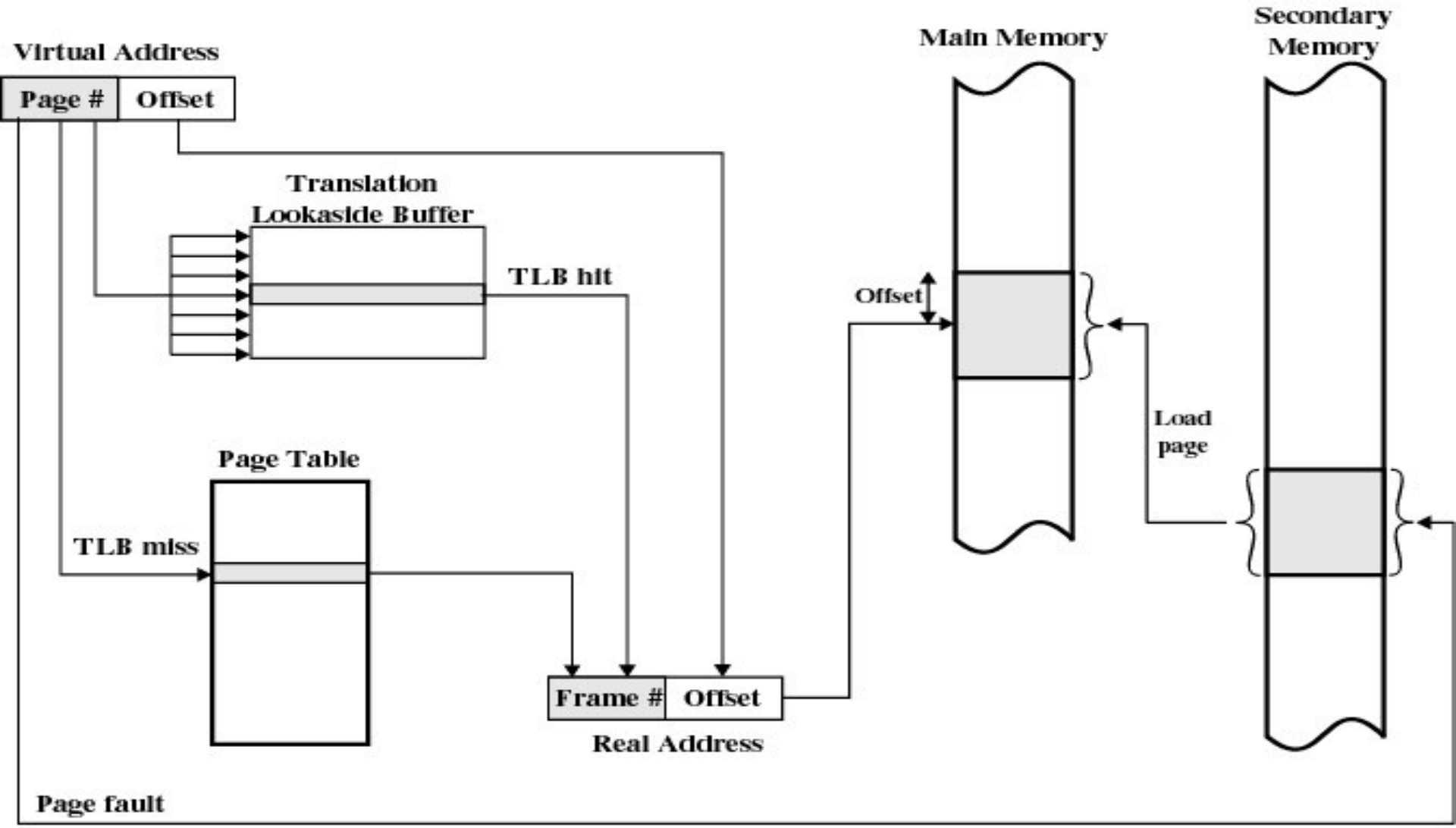



Figure 8.7 Use of a Translation Lookaside Buffer

# Size of Page & Page Tables

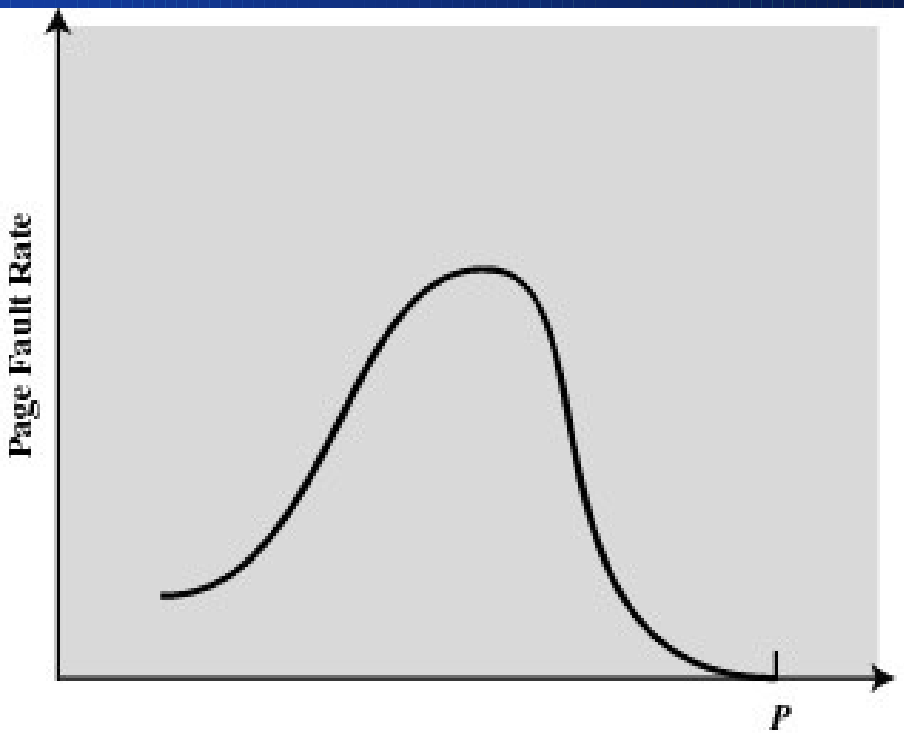
1. Page size
2. Page table size
3. For huge page table
   - Organization
   - Storage

# Page Size

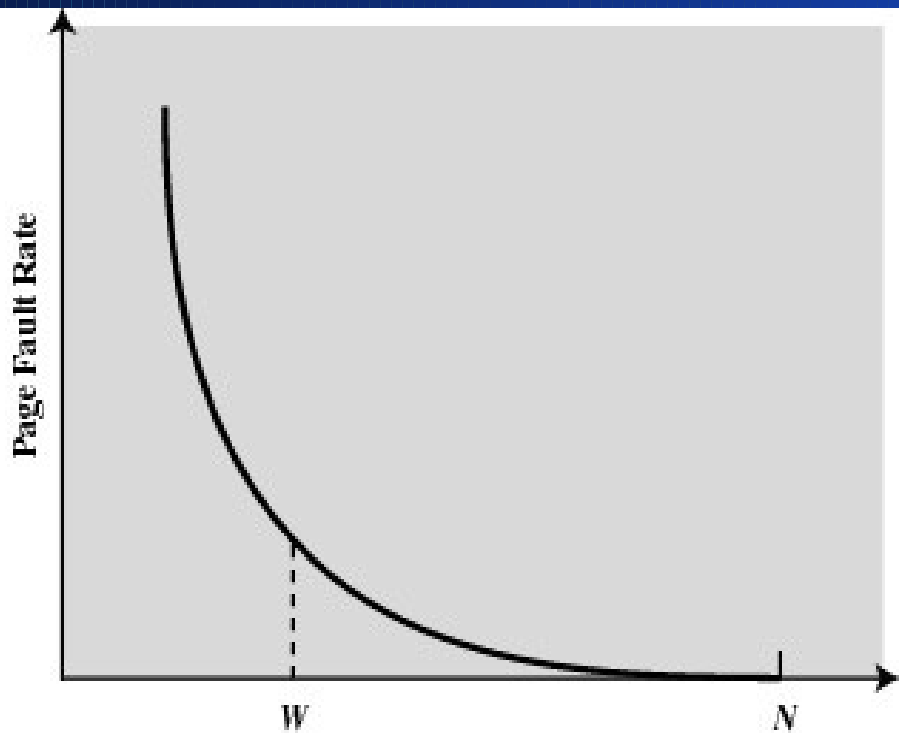
- **Less internal fragmentation, small page size**
  - Smaller page size, more pages required per process.
  - More pages per process means larger page tables.
  - Larger page tables means large portion of page tables in virtual memory.
- **Secondary memory favors large page size**
  - Secondary memory is designed to efficiently transfer large blocks of data .

# Page Size

- **Small page size**
  - large number of pages will be found in main memory.
  - As time goes on during execution, the pages in memory will all contain portions of the process near recent references.
  - Page faults （页面访问失败，缺页） low.
- **Increased page size**
  - causes pages to contain locations further from any recent reference.
  - Page faults rise.




(a) Page Size




(b) Number of Page Frames Allocated

$P$ = size of entire process
$W$ = working set size
$N$ = total number of pages in process

**Figure 8.11 Typical Paging Behavior of a Program**

电子科技大学
University of Electronic Science and Technology of China

# Page Size

- Multiple page sizes provide the flexibility needed to effectively use a TLB.
- Large pages can be used for program instructions.
- Small pages can be used for threads.
- Most operating system support only one page size.

# Example Page Sizes

**Table 8.2 Example Page Sizes**

| Computer | Page Size |
|---|---|
| Atlas | 512 48-bit words |
| Honeywell-Multics | 1024 36-bit word |
| IBM 370/XA and 370/ESA | 4 Kbytes |
| VAX family | 512 bytes |
| IBM AS/400 | 512 bytes |
| DEC Alpha | 8 Kbytes |
| MIPS | 4 kbyes to 16 Mbytes |
| UltraSPARC | 8 Kbytes to 4 Mbytes |
| Pentium | 4 Kbytes or 4 Mbytes |
| PowerPc | 4 Kbytes |

# Page Table Structure

- 许多计算机系统支持大容量虚拟内存。比如，在 VAX 系统中，每个进程使用的虚拟内存最大容量为 $2^{31}$(= 2G) 个字节。
- 大容量虚拟内存的实现需要大页表。比如，在 VAX 系统中，页面的大小为 $2^9$ (=512 ) 个字节，因此如果某进程使用的虚拟内存容量为系统规定的最大容量， 那么该进程的页表将有多达 $2^{22}$ (= 4M) 个表项。
- 下面介绍三种解决“大页表占用大内存”的方法

# Virtual Page Tables

- The entire page table may take up too much main memory.
- Page tables are also stored in virtual memory.
- When a process is running, part of its page table is in main memory.
- How to deal with it?

# 多级页表 Two-Level Scheme for 32-bit address

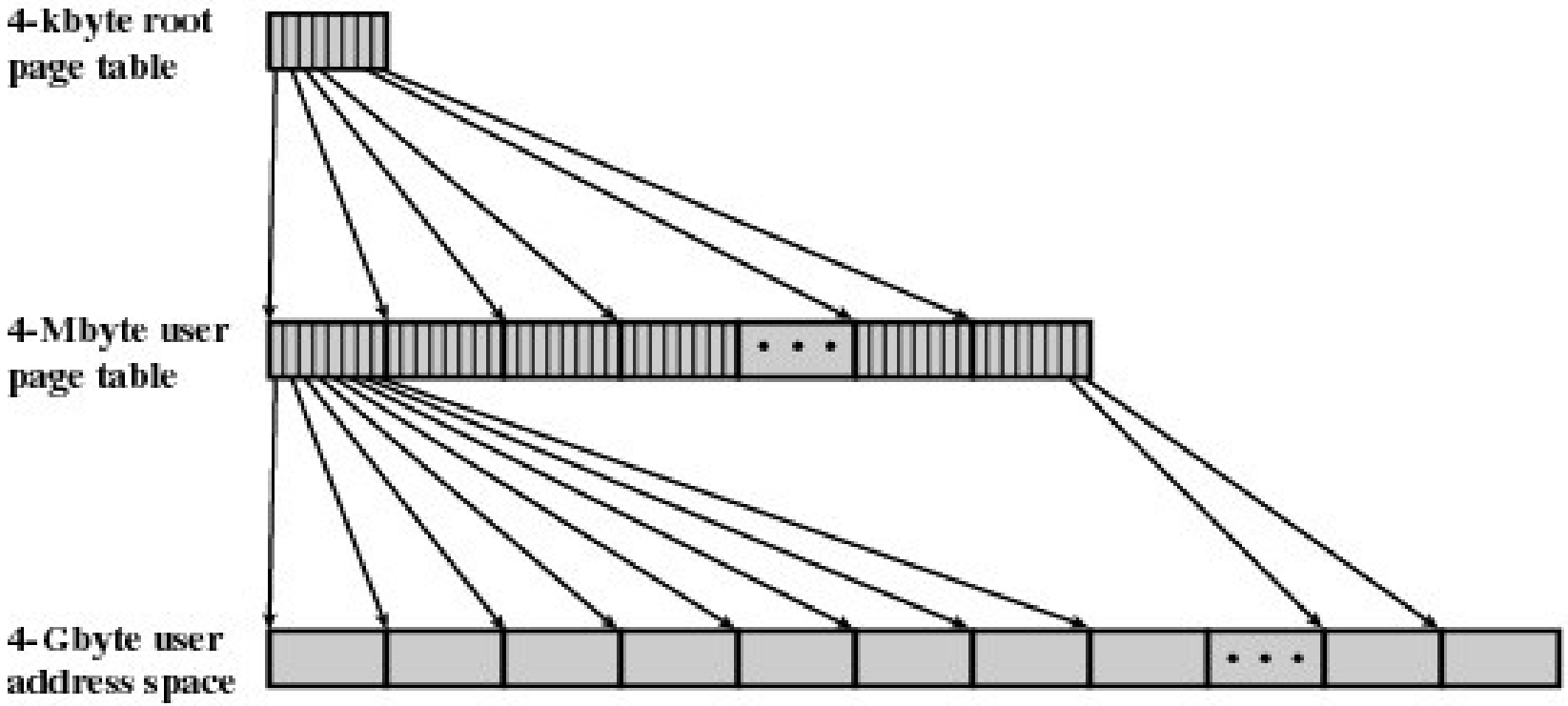
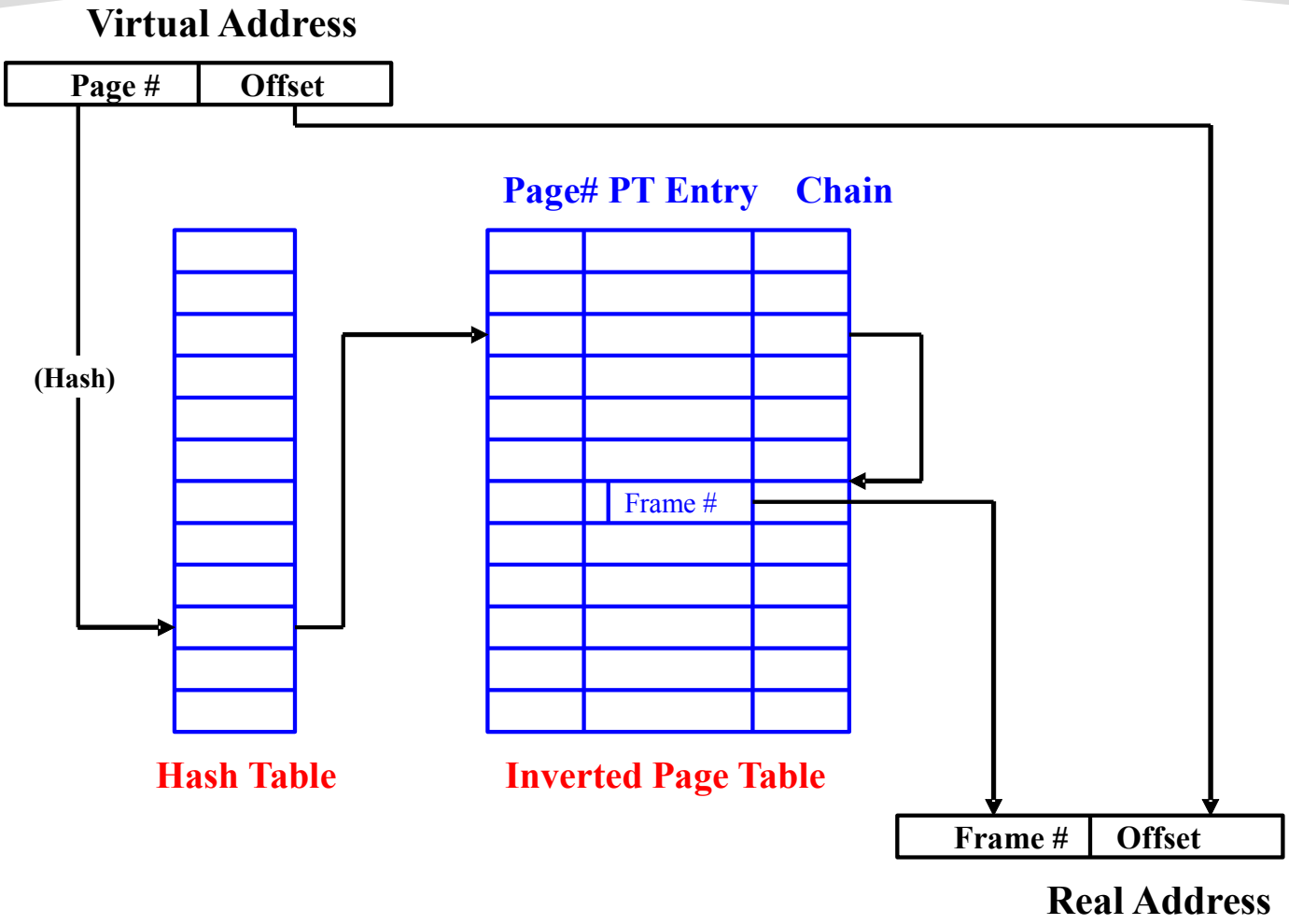



Figure 8.4 A Two-Level Hierarchical Page Table [JACO98a]

# Inverted Page Table（反置页表）




**Inverted Page Table Structure**

电子科技大学
University of Electronic Science and Technology of China