操作系统

Operating system

胡燕

大连理工大学

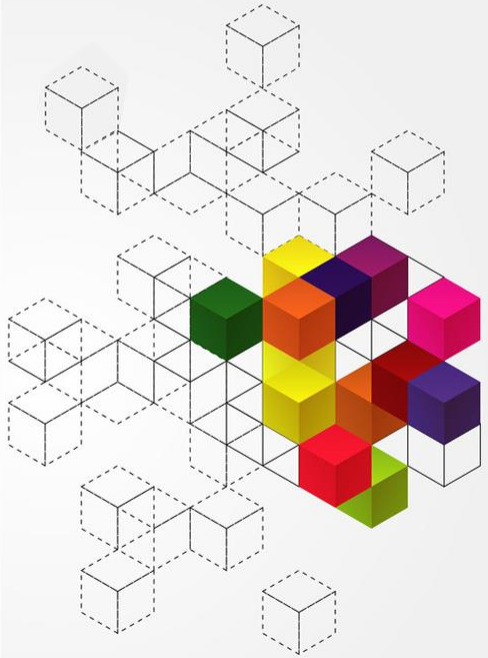
内容纲要

8.4 分页的硬件支持

一、分页硬件基本结构

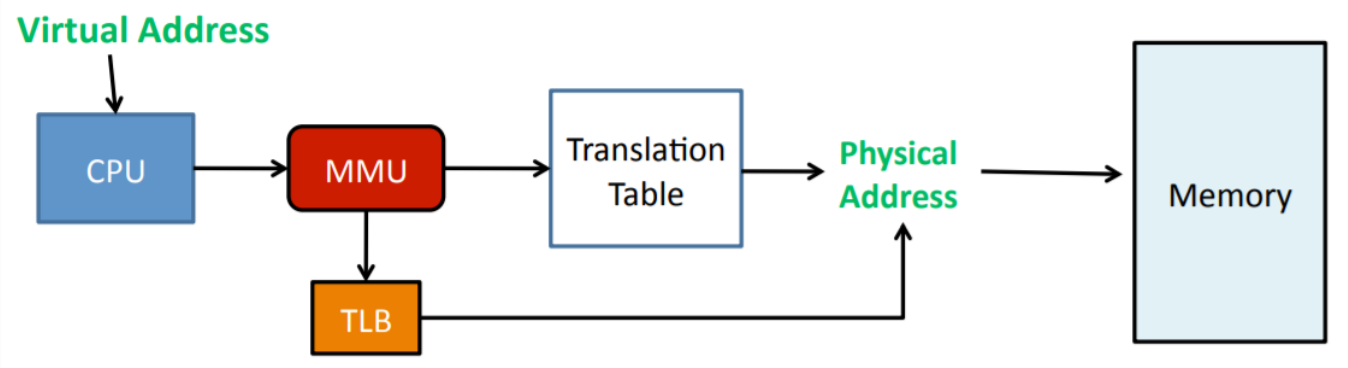
二、TLB

三、支持TLB的分页硬件性能评估



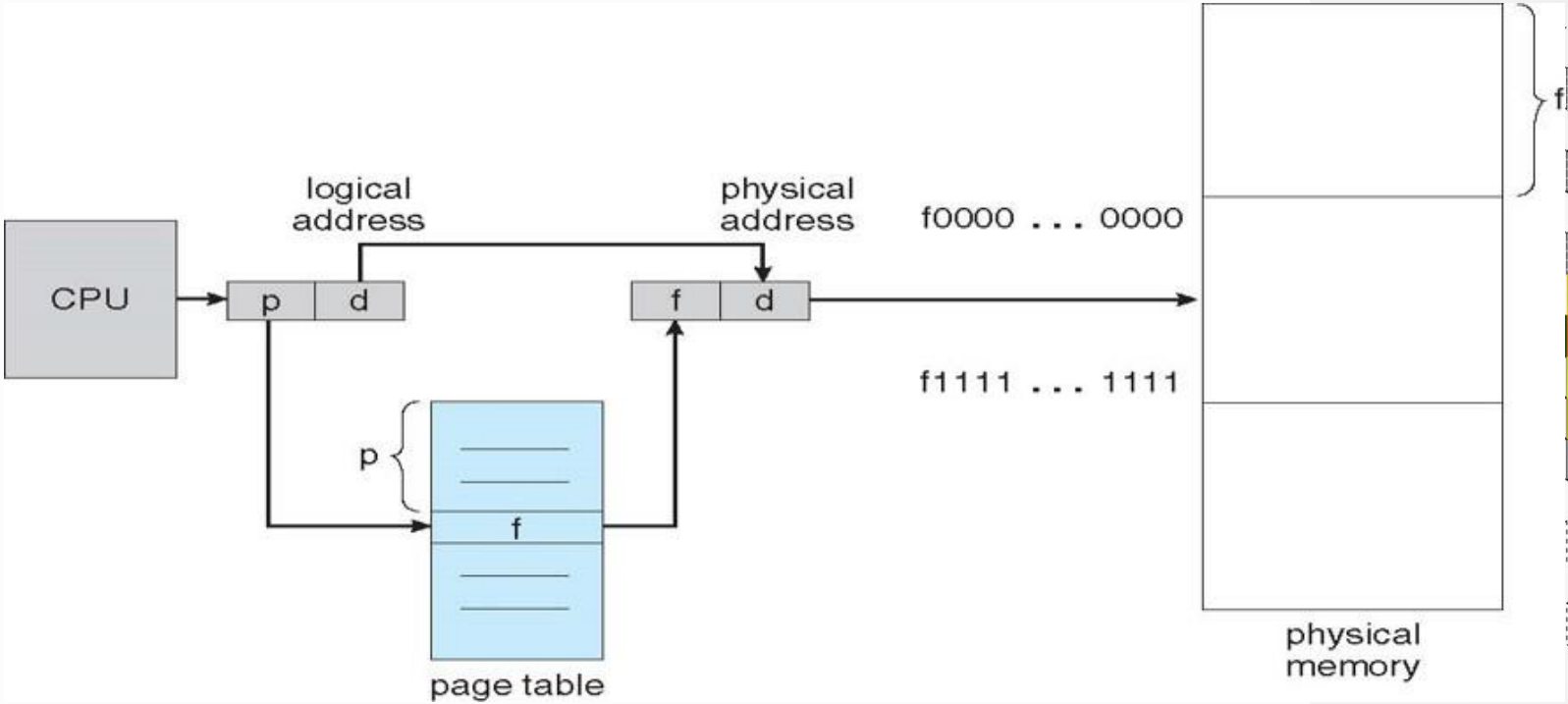
一、分页硬件基本结构

- 分页机制的实现需要硬件支持
- MMU(Memory Management Unit)
  - 用来支持逻辑地址（虚地址）到物理地址转换的硬件单元
  - CPU核心发出的地址都会交给MMU进行翻译



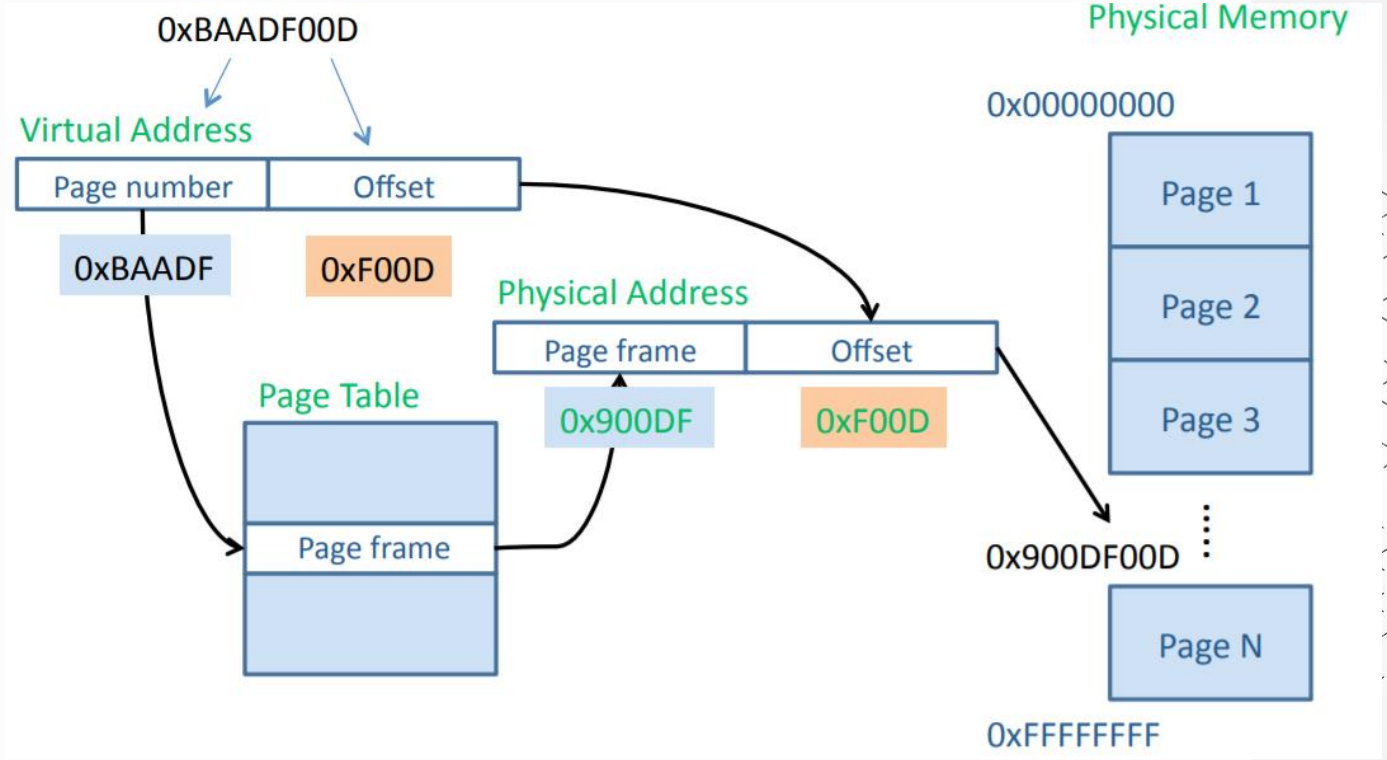
一、分页硬件基本结构

- 分页硬件的逻辑示意图



一、分页硬件基本结构

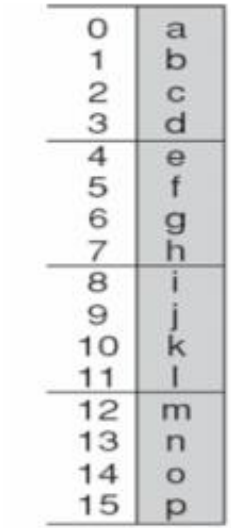
- **分页机制下的地址翻译示例**



填空题 4分

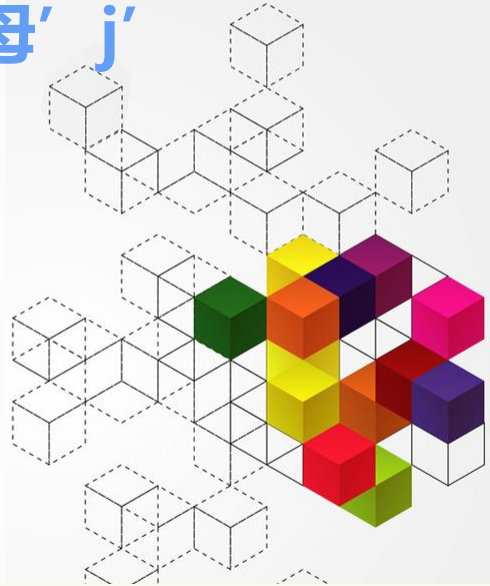
设置

系统中逻辑地址位数是4，物理地址位数是5，每个页的大小=4字节，某进程逻辑地址空间和页表如图所示。请问字母’c’的物理地址= [填空1]，字母’j’的物理地址= [填空2]。



logical memory

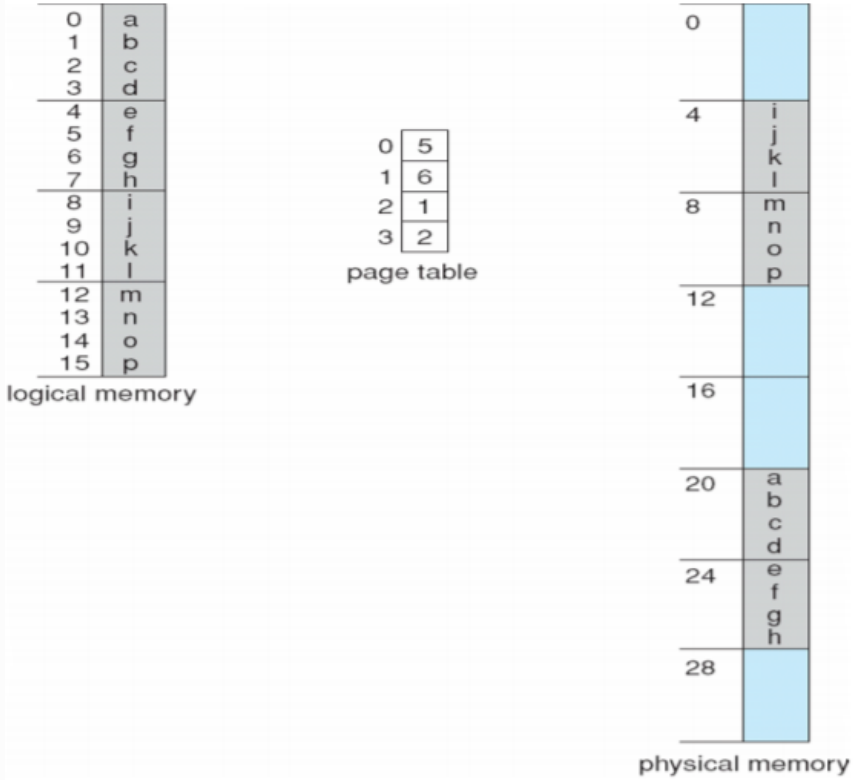
page table

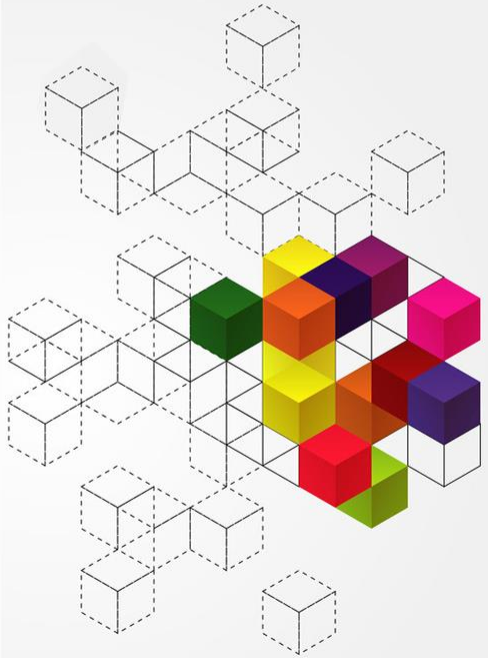


填空题需3.0以上版本雨课堂

作答

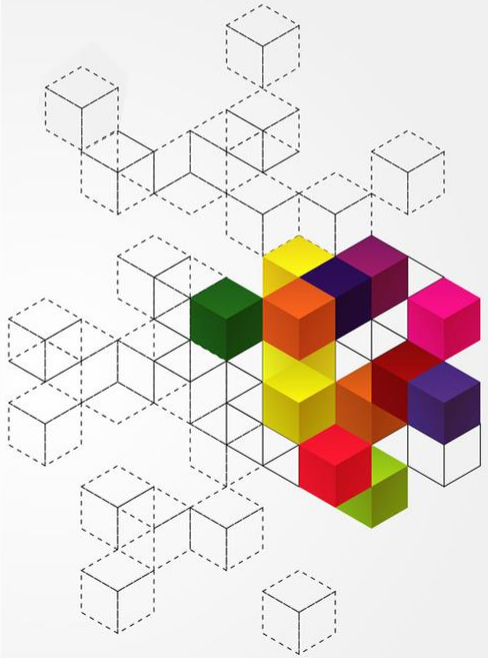
一、分页硬件基本结构


0 a
1 b
2 c
3 d
4 e
5 f
6 g
7 h
8 i
9 j
10 k
11 l
12 m
13 n
14 o
15 p
logical memory
0 5
1 6
2 1
3 2
page table
0
4 i j k l
8 m n o p
12
16
20 a b c d
24 e f g h
28
physical memory



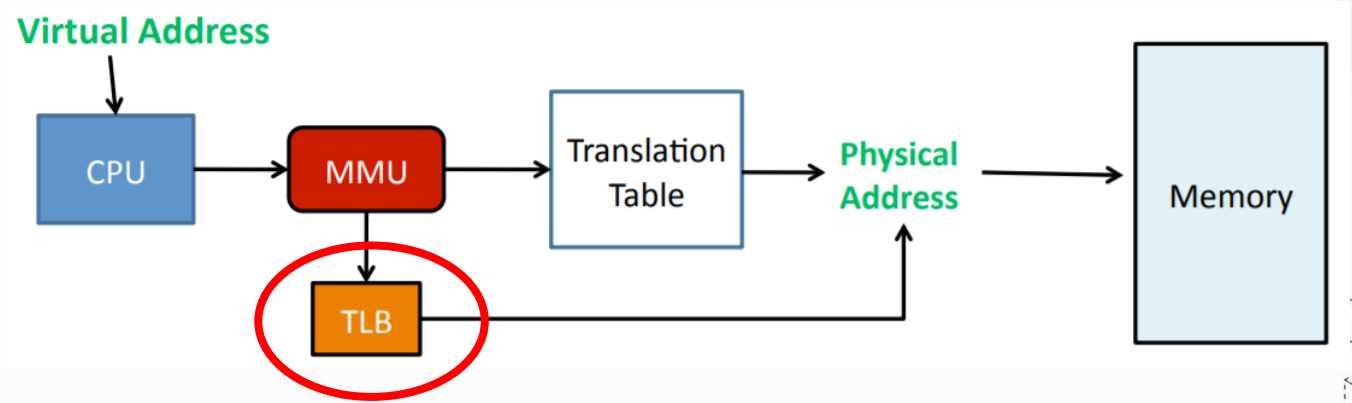
一、分页硬件基本结构

- 讨论题1：分析分页机制所带来的开销

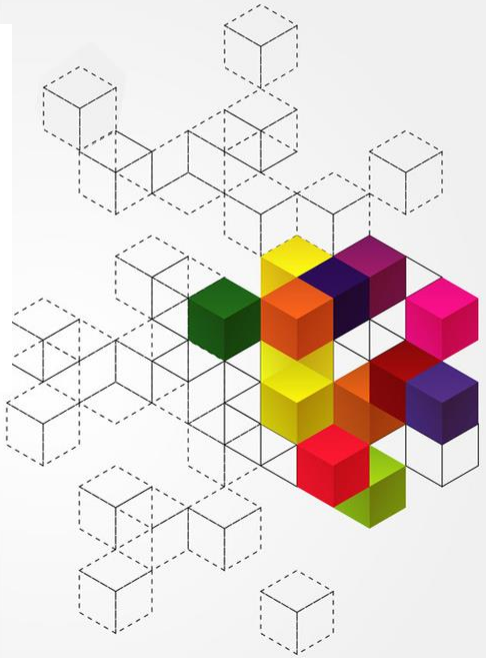


二、TLB

- **Paging is Slow!**

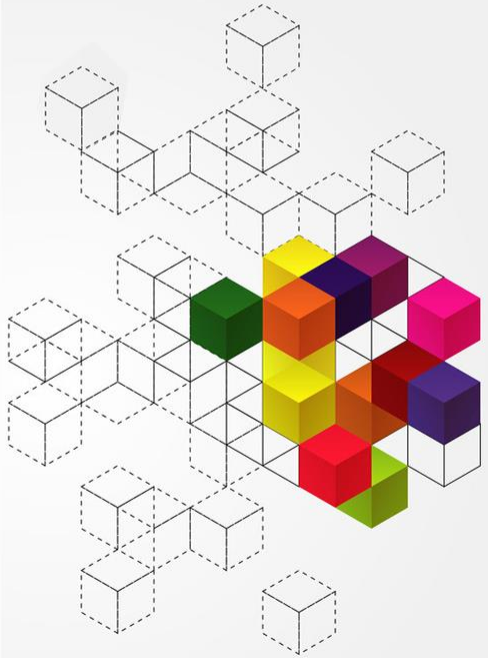


TLB comes to it’s rescue



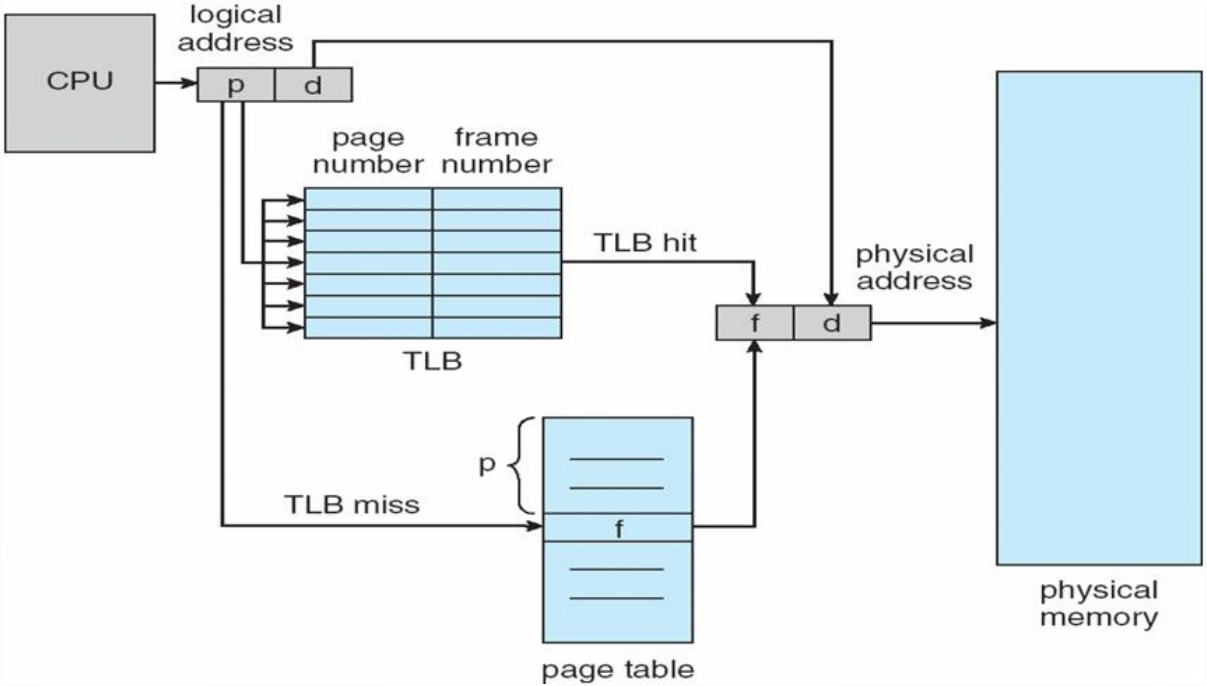
二、TLB

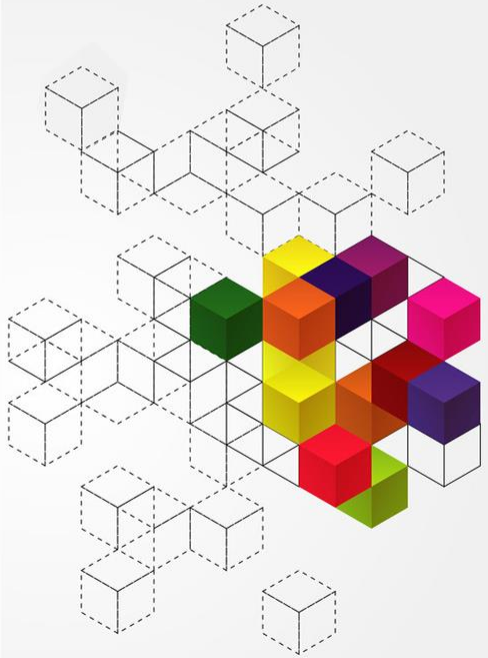
- **TLB (Translation Lookaside Buffer)**
  - **part of MMU**
  - **a hardware cache of popular virtual-to-physical address translations (aka, address translation cache)**



二、TLB

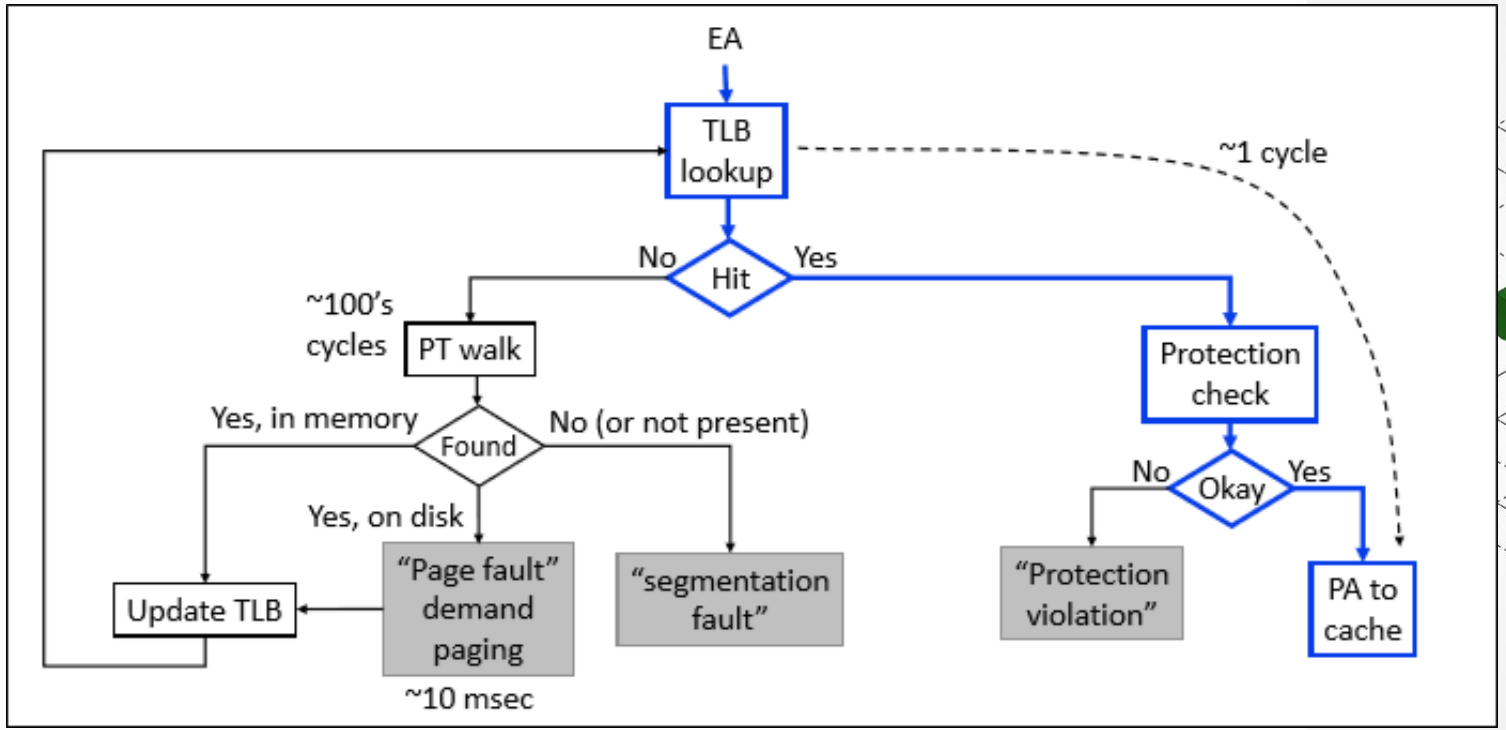
- 增加TLB缓存机制的页硬件逻辑示意图





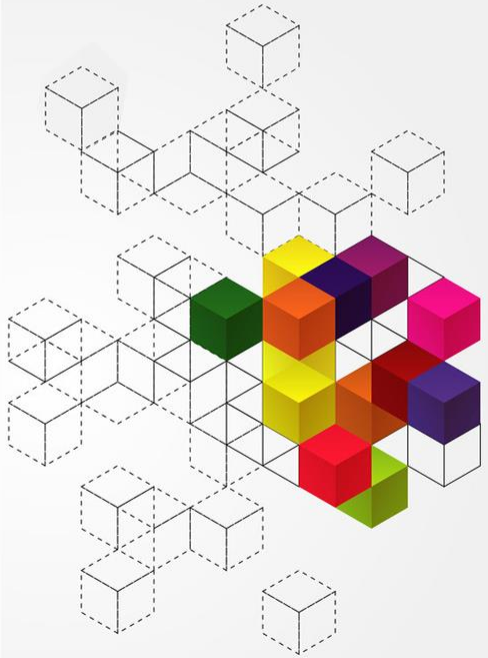
二、TLB

- 一次分页地址的完整寻址流程



一、分页硬件基本结构

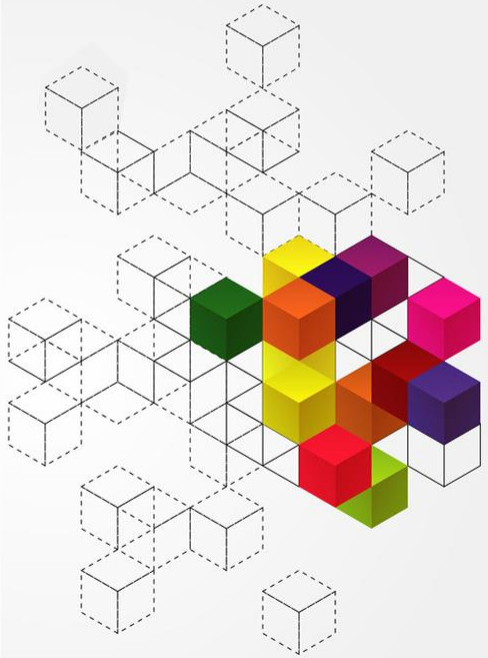
- 讨论：慕课堂讨论1.



二、TLB

- 快表结构示意

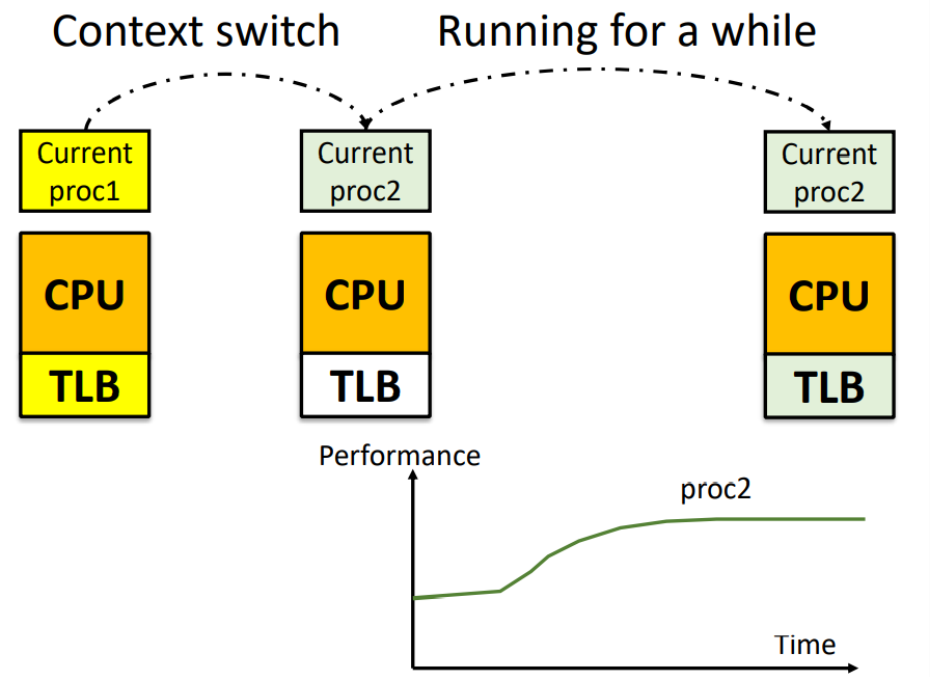
Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

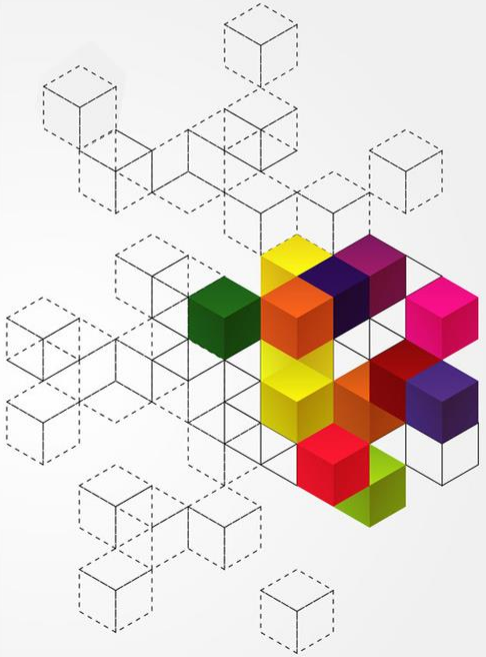


- 进程的Cold Start问题.

二、TLB

进程cold start示意 （proc2）





三、支持TLB的分页硬件性能评估

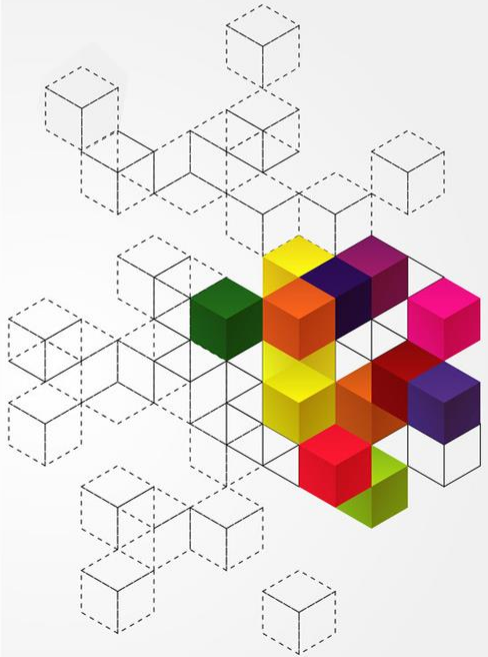
包含TLB的页式机制访存效率分析

- 假设单次内存访问的时间为1
- TLB的访问时间为ε（这是个远小于1的值）
- TLB的页表项访问命中率= α

内存有效访问时间

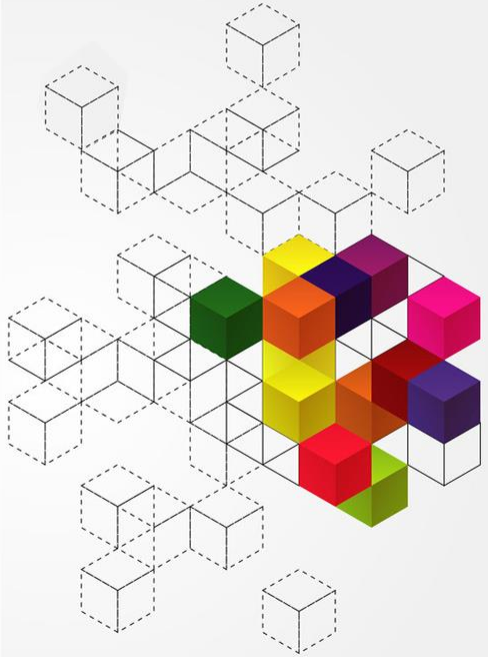
$$= (1 + \varepsilon)\,\alpha + (2 + \varepsilon)(1 - \alpha)$$

$$= 2 + \varepsilon - \alpha$$



本讲小结

- 分页的硬件支持



内容纲要

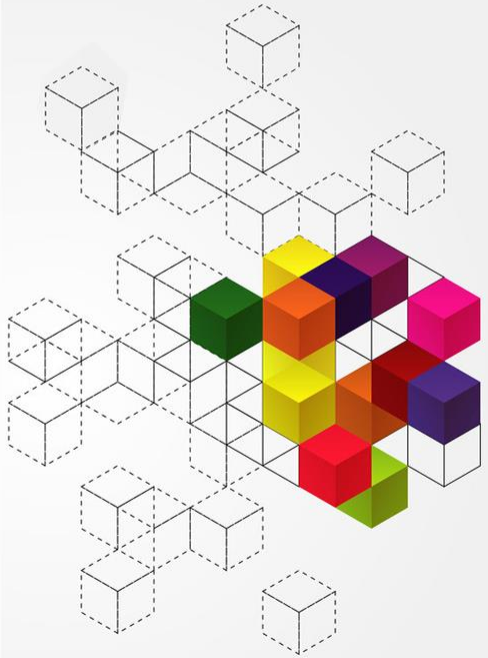
8.5 页表结构

一、页表结构

二、多级页表

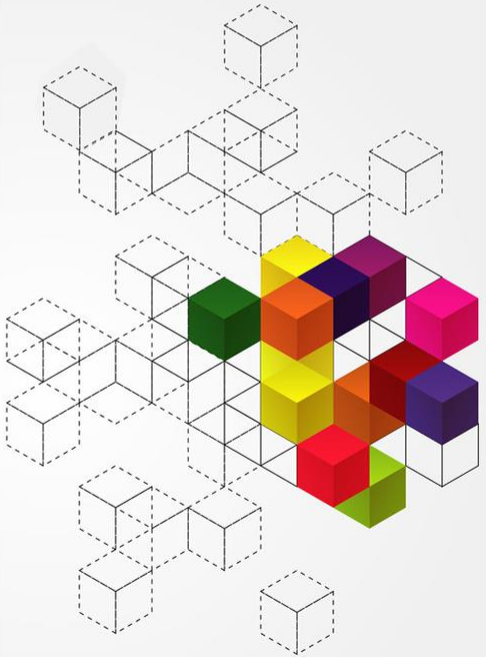
三、哈希页表

四、反置页表



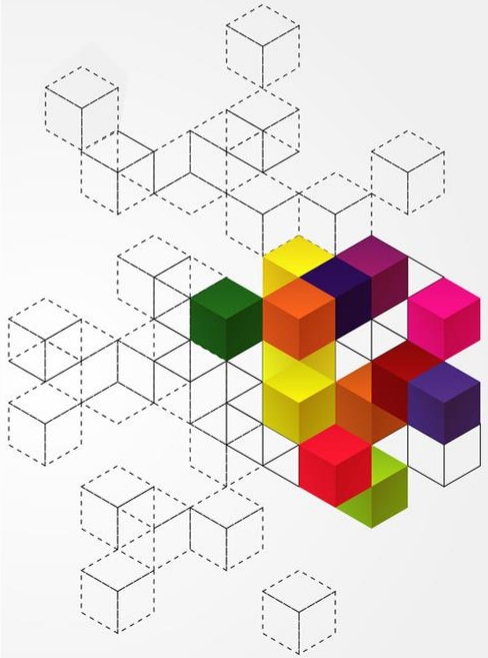
一、页表结构

- **计算机系统物理内存空间持续扩大，单级页表的结构已经无法满足实际需要**
  - 单级页表可能形成较大的页表。为了表达4G的逻辑地址空间，4k大小的页，需要的页表大小为4M字节大小的页表
  - 64位的逻辑地址空间，所需的页表则更为庞大



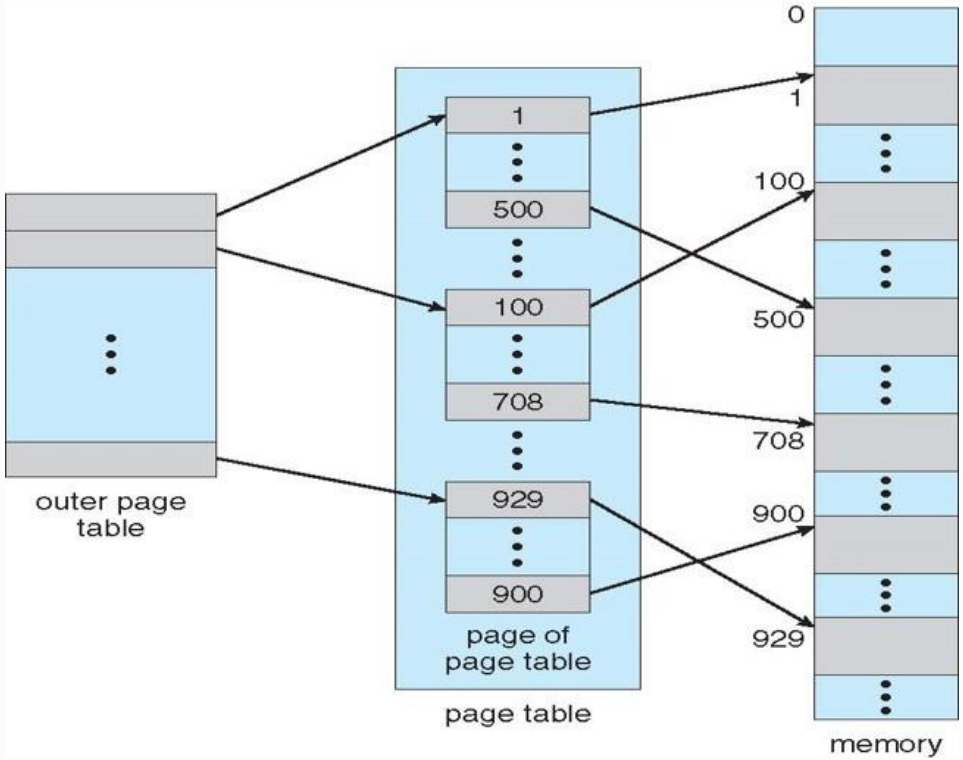
一、页表结构

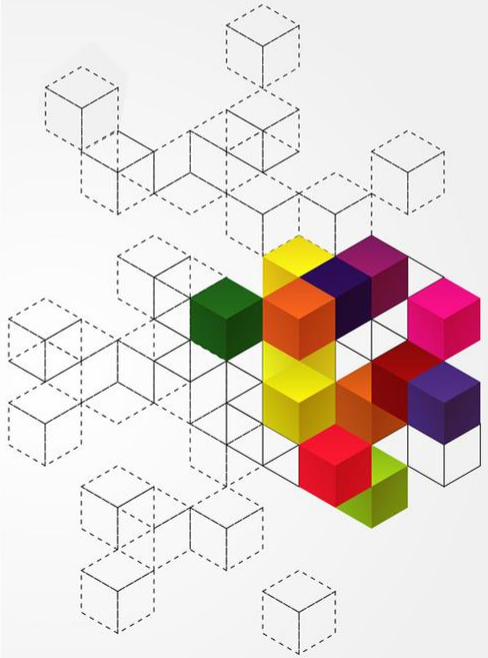
- **处理大地址空间的3种典型页表结构方案**
  - 多级页表（Hierarchical Page Tables）
  - 哈希页表（Hashed Page Table）
  - 反置页表（Inverted Page Table）



二、多级页表

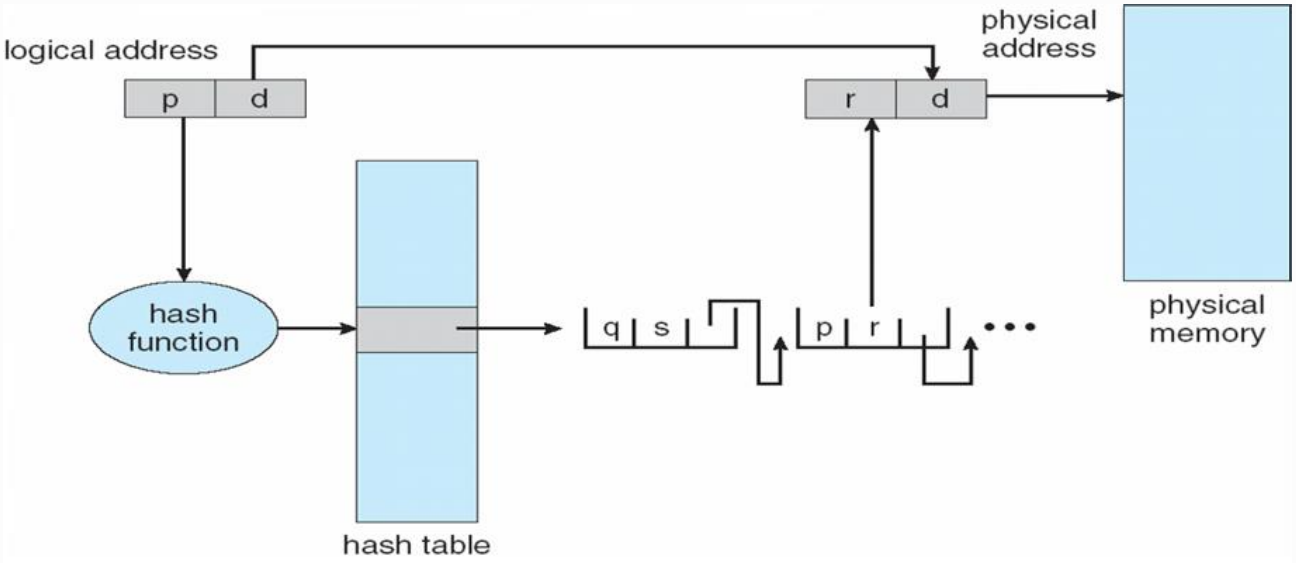
- **两级页表示意图**

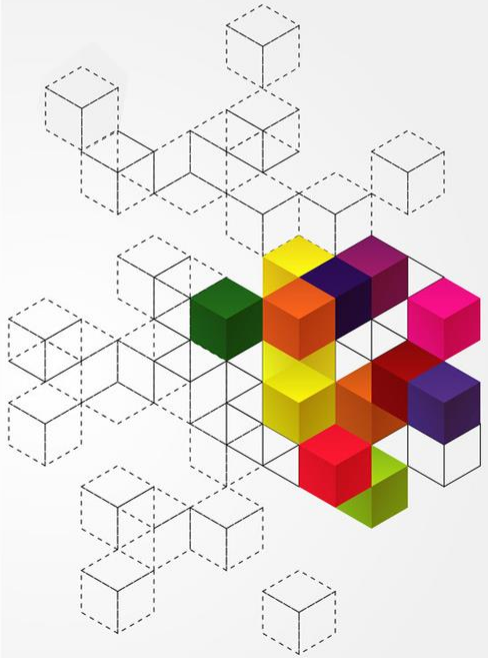




三、哈希页表

- **以哈希结构组织页表**

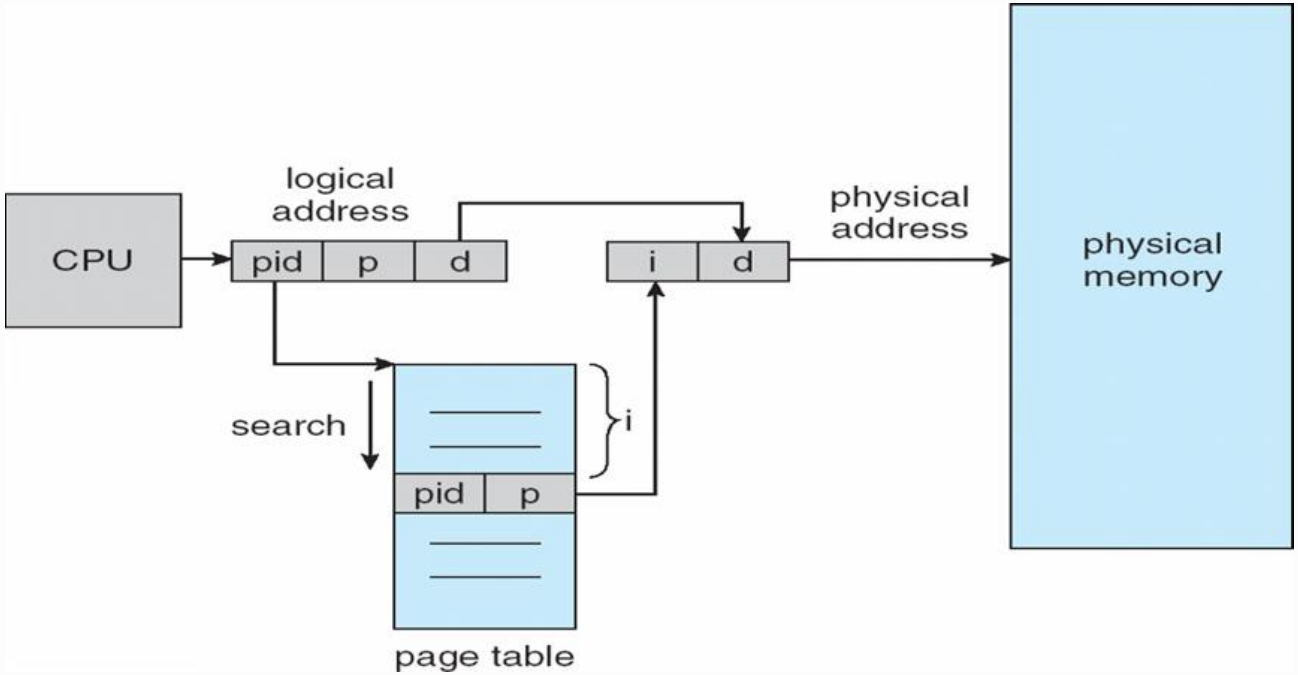


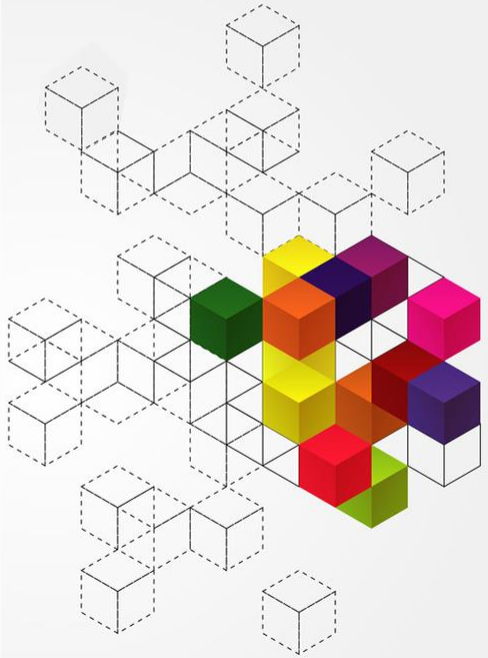


- 可应用于地址空间大于32位的CPU
- 进程只使用逻辑地址空间较小一部分的情形下，哈希页表可以有较高的效率

四、反置页表

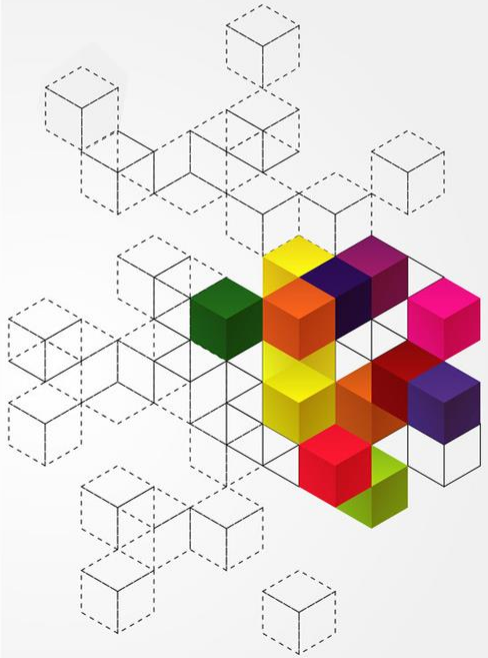
- **反置页表结构**





本讲小结

- 页表结构



内容纲要

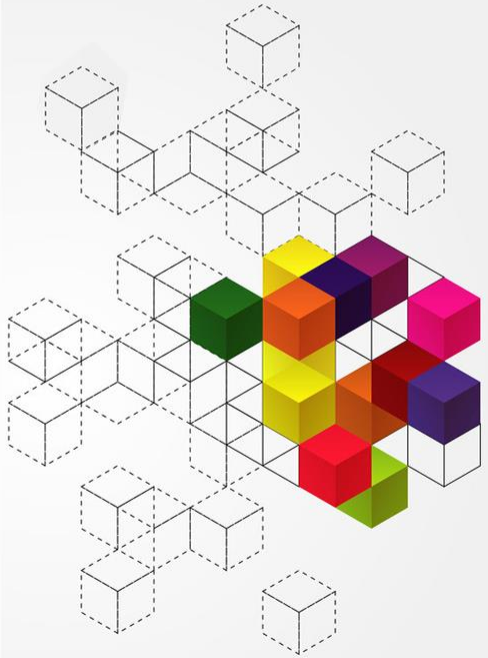
8.6 分段机制

一、分段概念

二、分段结构

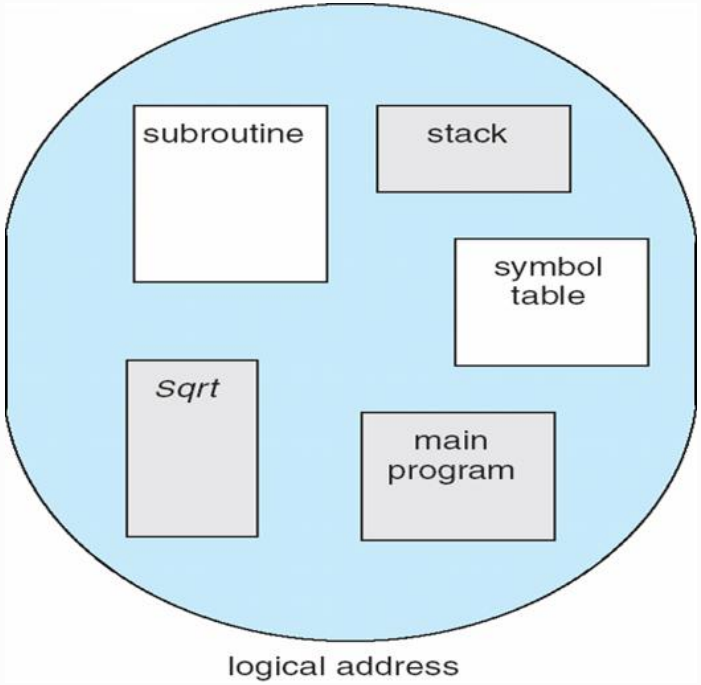
三、分段硬件支持

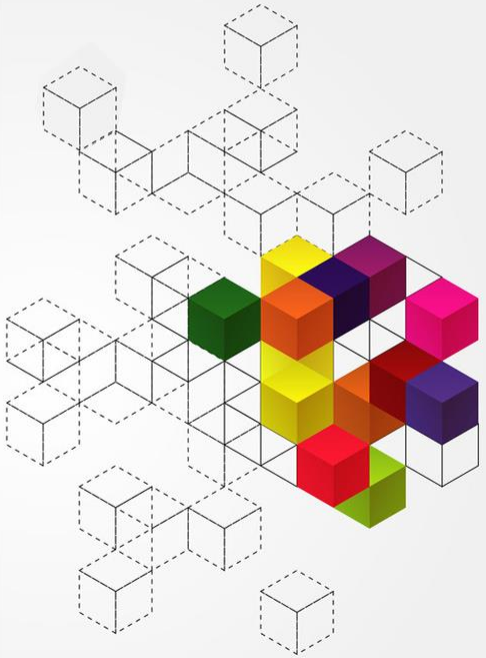
四、段式地址翻译



一、分段概念

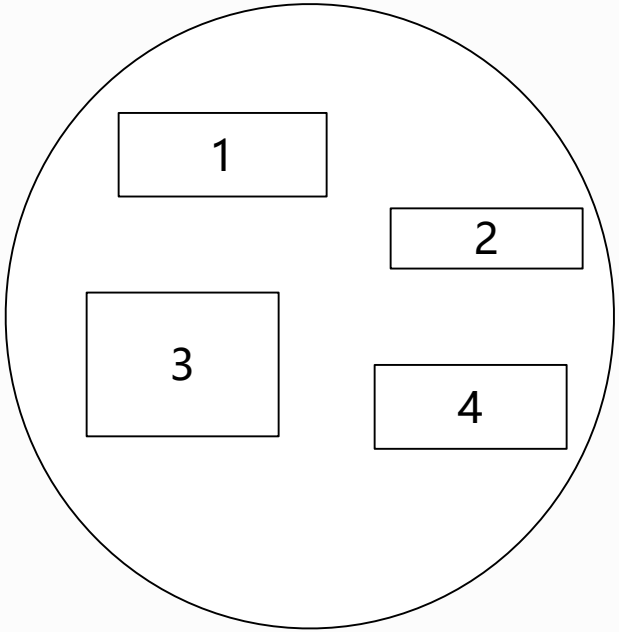
- 进程地址空间逻辑分块示意图



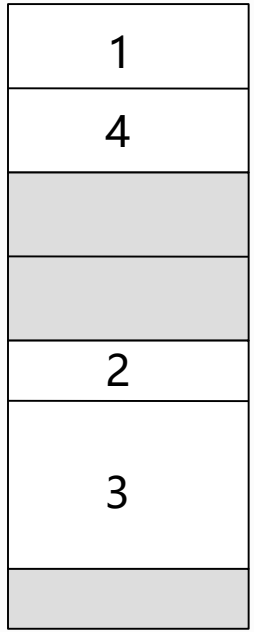


二、分段结构

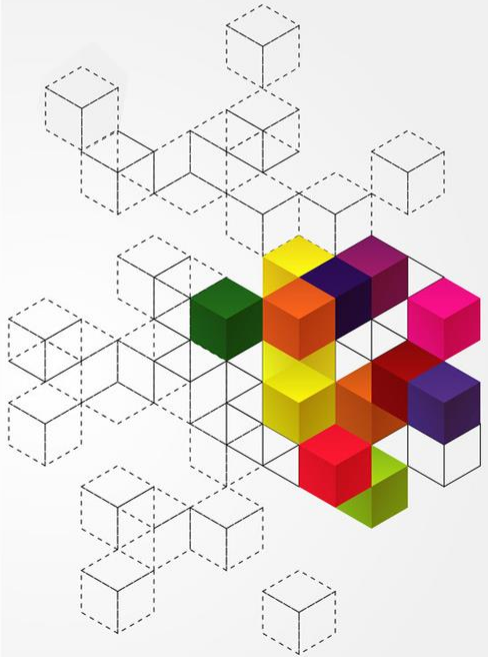
- 程序逻辑空间分段，每个段占据一块连续内存区域



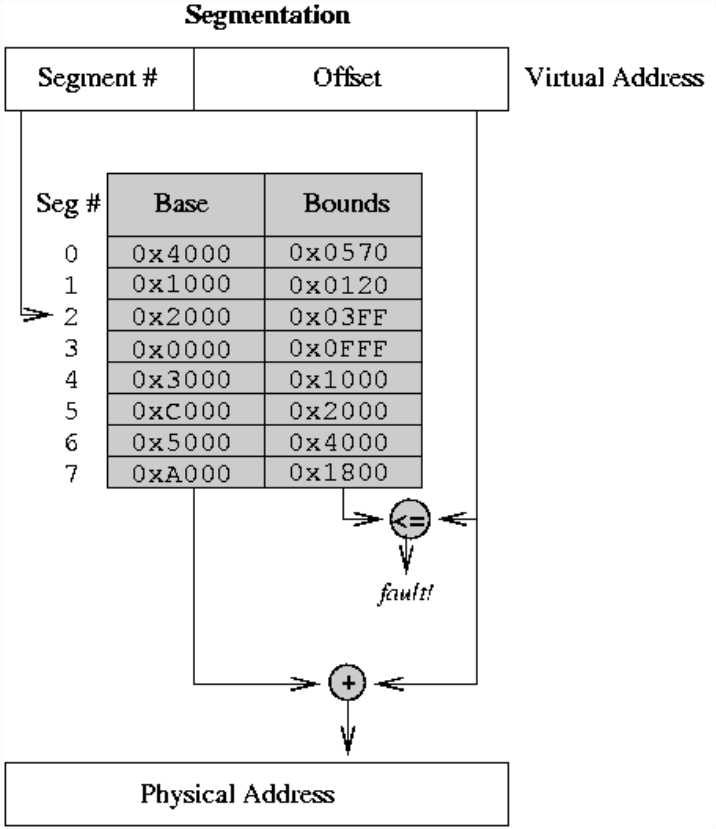
user space



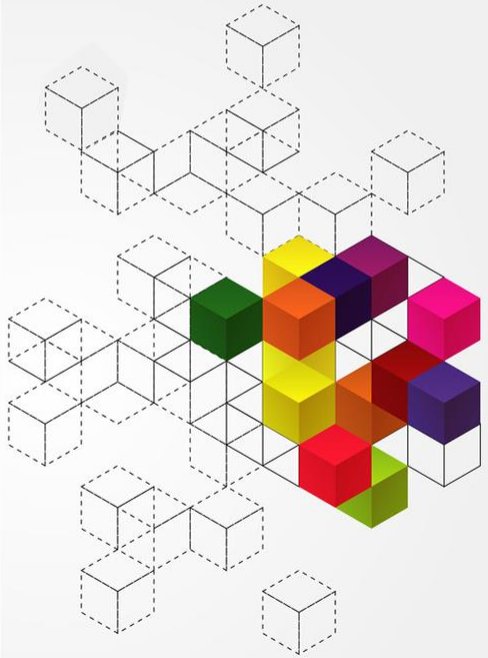
physical memory space



二、分段结构



- 分段地址，分为段号（Segment #）和段内偏移(Offset)



Segmentation Architecture

- Logical address consists of a two tuple:

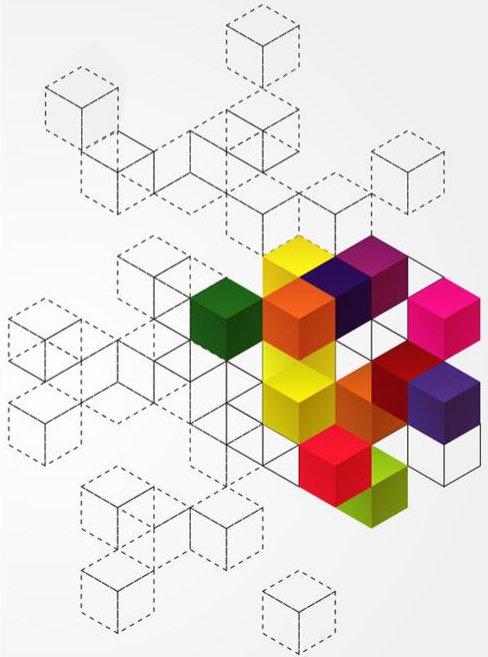
  <segment-number, offset>,

- **Segment table** – maps two-dimensional physical addresses; each table entry has:
  - **base** – contains the starting physical address where the segments reside in memory
  - **limit** – specifies the length of the segment

- **Segment-table base register (STBR)** points to the segment table's location in memory

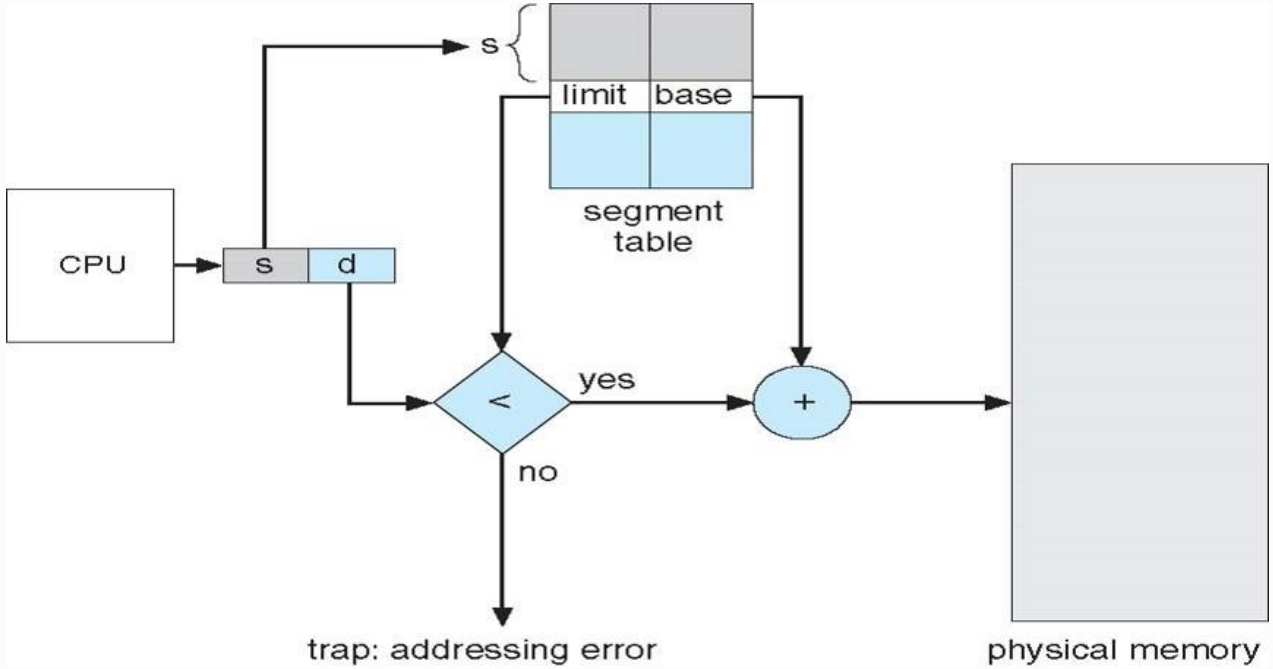
- **Segment-table length register (STLR)** indicates number of segments used by a program;

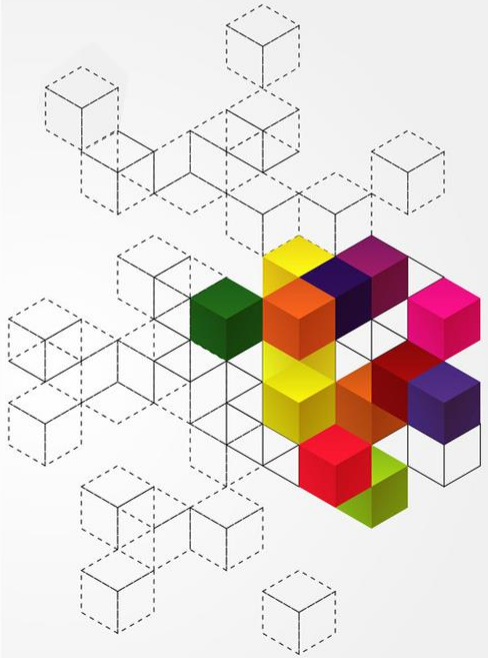
  segment number ***s*** is legal if ***s*** < **STLR**



三、分段硬件支持

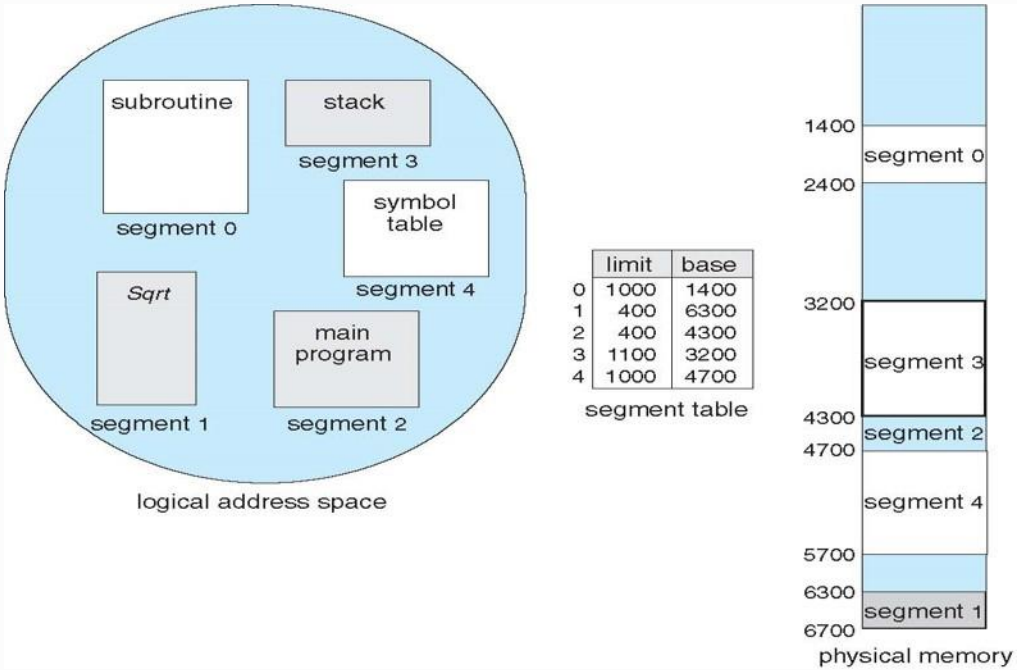
- **支持分段的硬件逻辑示意图**

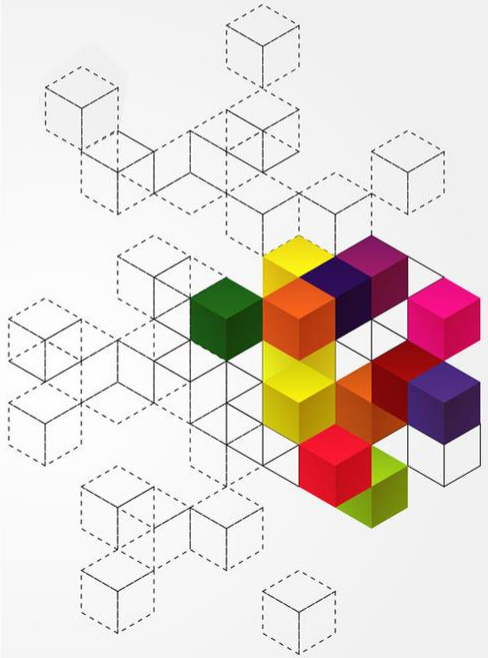




四、段式地址翻译

- 分段机制下的段地址转换



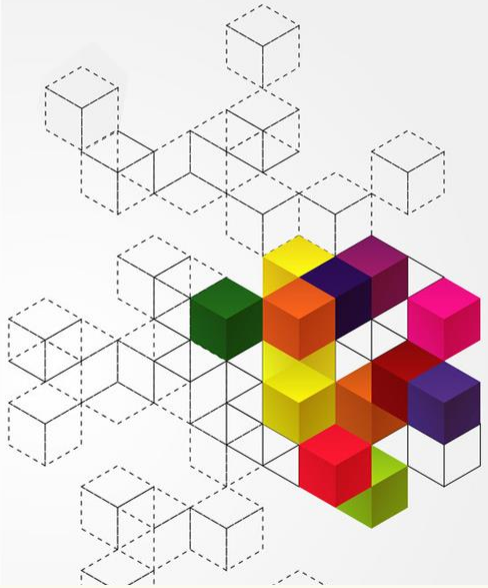


填空题 6分

设置

若某系统采取分段内存管理，其段表如下图所示，那么，逻辑地址（2，88）对应的物理地址是（ [填空1] ）；逻辑地址（4，100）对应的物理地址是（ [填空2] ）。

段号	基地址	段长
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

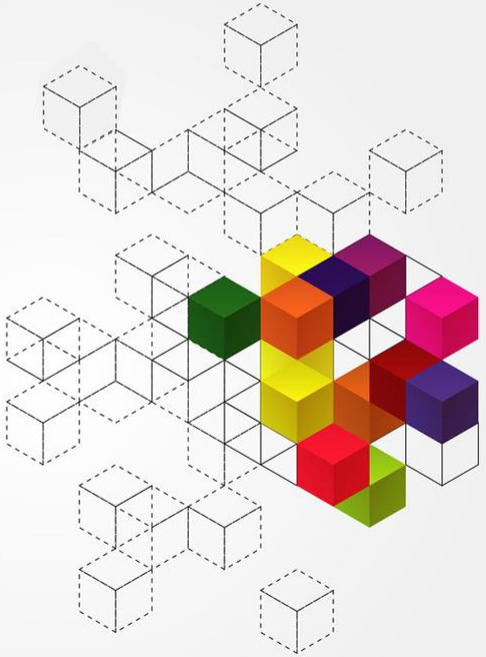


正常使用填空题需3.0以上版本雨课堂

作答

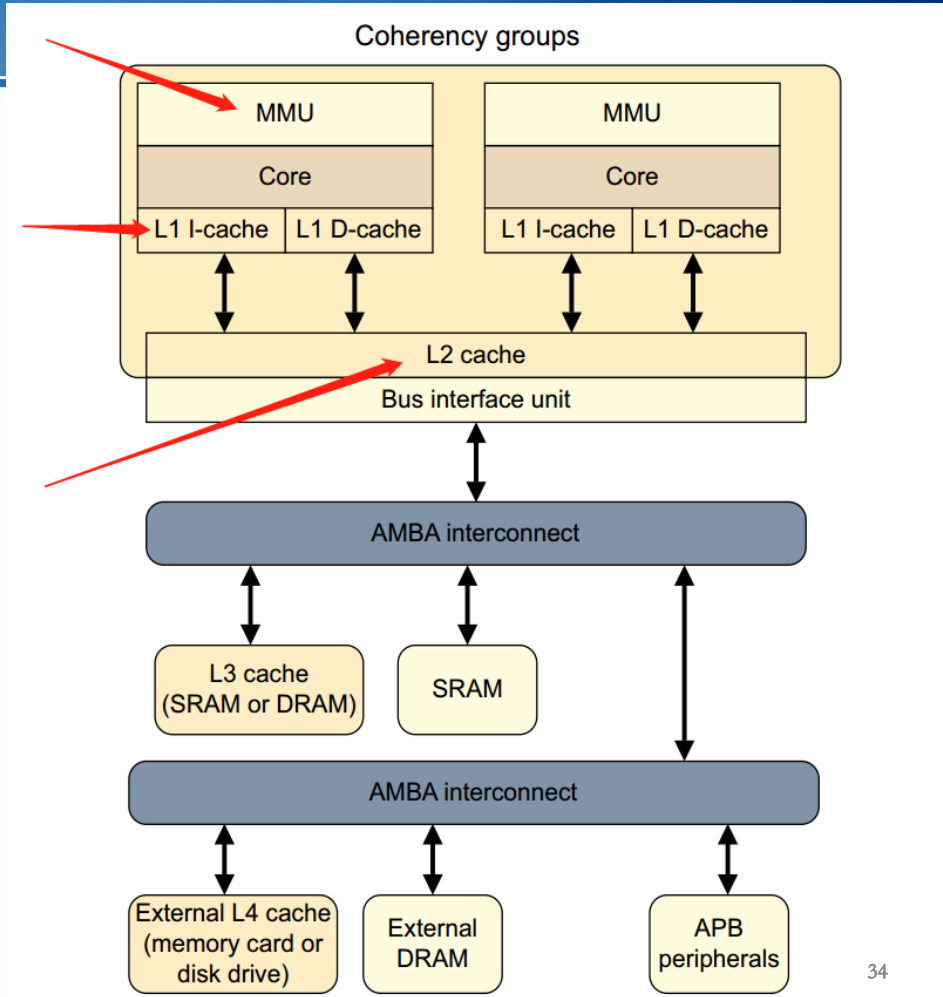
本讲小结

- 分段机制



E.1 ARM CPU MMU

- Arm架构
  - Arm架构的芯片在手机领域大行其道
  - 华为基于ARM的鲲鹏CPU也在云服务器端大规模部署
  - 苹果的MAC系列已开始用自研M1架构的芯片



E.1 ARM CPU MMU

- **Kunpeng 920**
  - **业界第一颗7nm数据中心ARM CPU**
  - **因遭受A国打压，原计划2023出的5nm Kunpeng 930会困难了**

E.1 ARM CPU MMU

- **Arm**