计算机体系结构

周学海
xhzhou@ustc.edu.cn
0551-63492149
中国科学技术大学

关于本章的术语

- **Miss： 缺失||失效**
- **Miss rate：缺失率||失效率**
- **Miss penalty：缺失代价||失效开销**
- **multilevel inclusive: 多级包容**
- **Multilevel exclusive：多级不包容||多级互斥**
- **Stall：停顿**
- **Write-through: 写直达||直写**
- **Write-back： 写回**

review-高级Cache优化方法

- **缩短命中时间**
  - 1、小而简单的第一级Cache
  - 2、路预测方法
- **增加Cache带宽**
  - 3、Cache访问流水化
  - 4、无阻塞Cache
  - 5、多体Cache
- **减小失效开销**
  - 6、关键字优先和提前重启
  - 7、合并写
- **降低失效率**
  - 8、编译优化
- **通过并行降低失效开销或失效率**
  - 9、硬件预取
  - 10、编译器控制的预取

Summary

Technique	Hit time	Band-width	Miss penalty	Miss rate	Power consumption	Hardware cost/ complexity	Comment
Small and simple caches	+			–	+	0	Trivial; widely used
Way-predicting caches	+				+	1	Used in Pentium 4
Pipelined cache access	–	+				1	Widely used
Nonblocking caches		+	+			3	Widely used
Banked caches		+			+	1	Used in L2 of both i7 and Cortex-A8
Critical word first and early restart			+			2	Widely used
Merging write buffer			+			1	Widely used with write through
Compiler techniques to reduce cache misses				+		0	Software is a challenge, but many compilers handle common linear algebra calculations
Hardware prefetching of instructions and data			+	+	–	2 instr., 3 data	Most provide prefetch instructions; modern high-end processors also automatically prefetch in hardware.
Compiler-controlled prefetching			+	+		3	Needs nonblocking cache; possible instruction overhead; in many CPUs

Figure 2.11 Summary of 10 advanced cache optimizations showing impact on cache performance, power consumption, and complexity. Although generally a technique helps only one factor, prefetching can reduce misses if done sufficiently early; if not, it can reduce miss penalty. + means that the technique improves the factor, – means it hurts that factor, and blank means it has no impact. The complexity measure is subjective, with 0 being the easiest and 3 being a challenge.

4.1 Cache的基本概念

存储系统的层次结构

Cache基本知识

4.2 Cache的基本优化方法

4.3 Cache的高级优化方法

4.4 存储器技术与优化

4.5 虚拟存储器 - 基本原理

4.4 存储器技术及优化

DRAM芯片 → 内存模组

存储器技术与优化

- **存储器的访问源**
  - 取指令、取操作数、写操作数和I/O
- **存储器性能指标**
  - 容量、速度和每位价格
  - 访问时间（Access Time)
  - 存储周期（Cycle Time)
- **种类：DRAM和SRAM**
  - Memory：DRAM
  - Cache：SRAM

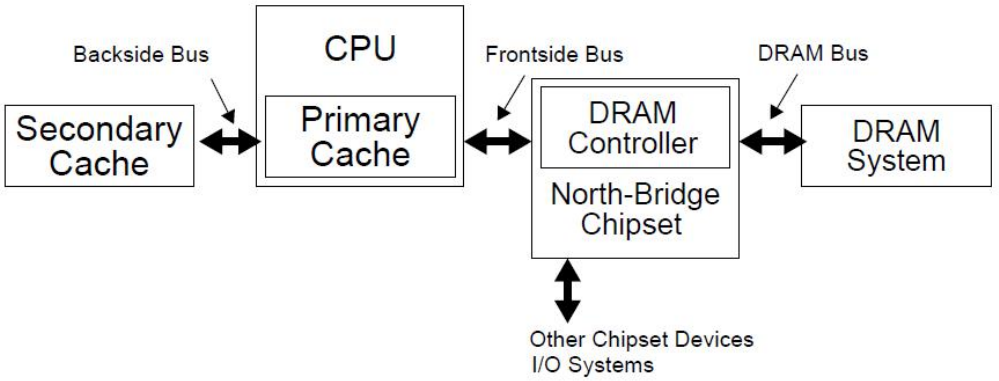
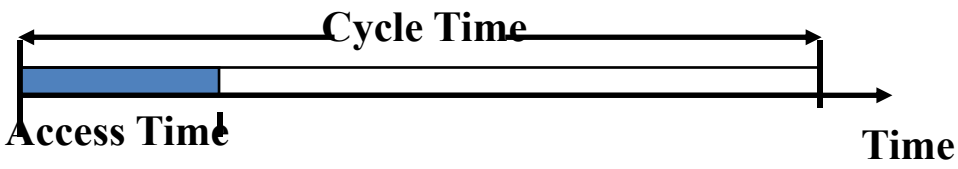


Figure 4.1: Memory System Architecture



内存子系统

- **内存子系统从CPU 到DRAM芯片的逐层关系**
  - Memory Controller
  - Channel
  - DIMM
  - rank
  - chip
  - bank
  - DRAM Array
    - row/column.

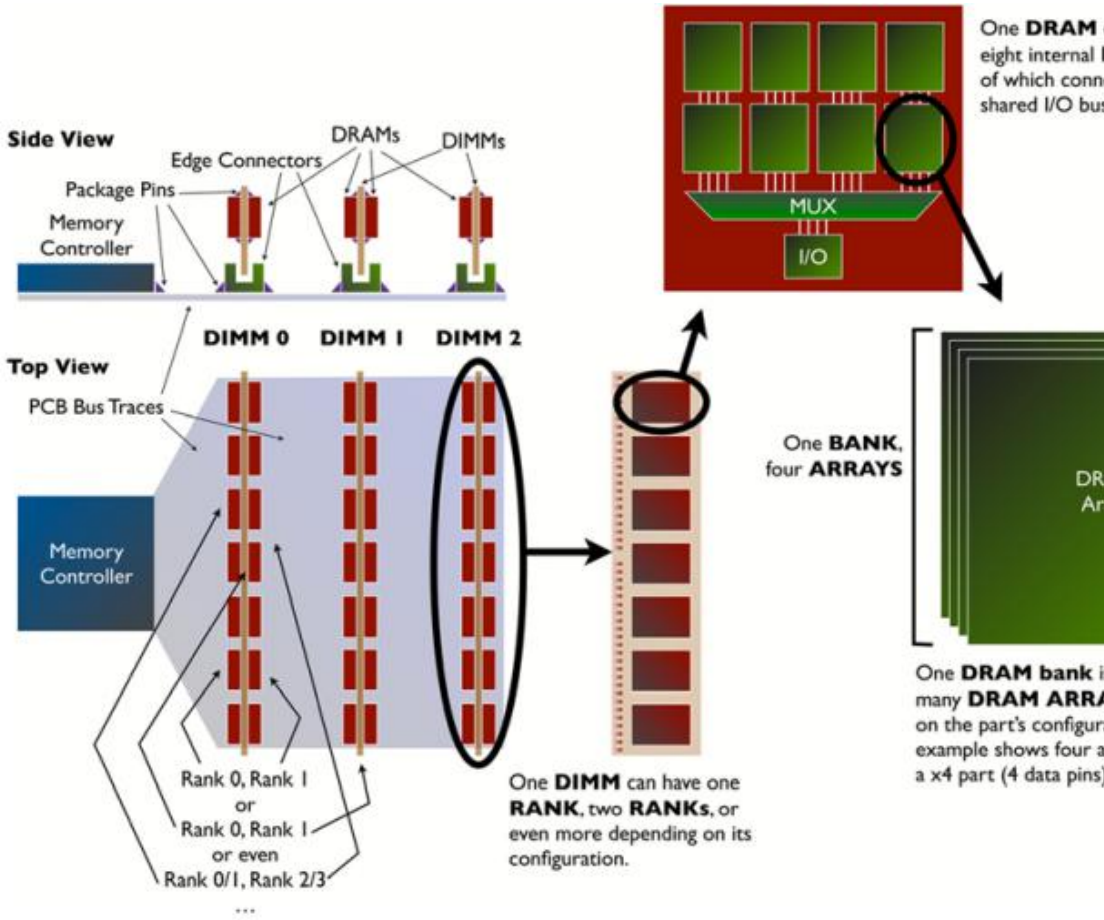
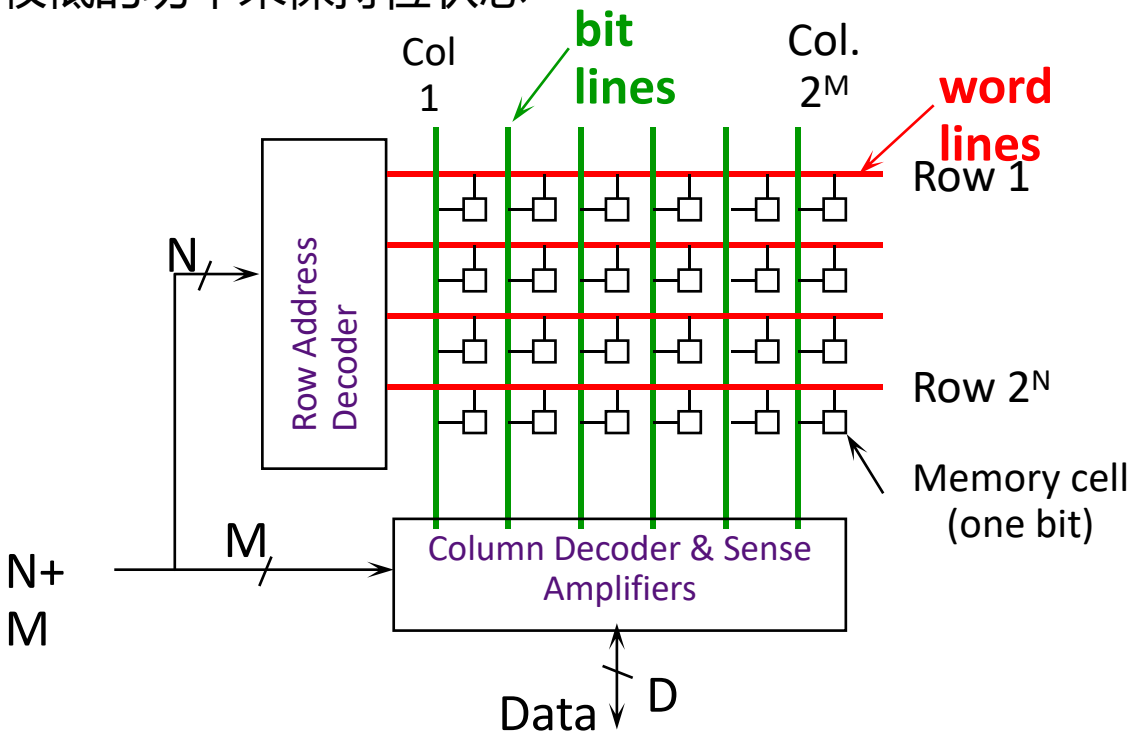


Figure 1.5: DIMMs, ranks, banks, and arrays. A system has potentially many DIMMs, ea may contain one or more ranks. Each rank is a set of ganged DRAM devices, each of which ha many banks. Each bank has potentially many constituent arrays, depending on the parts da

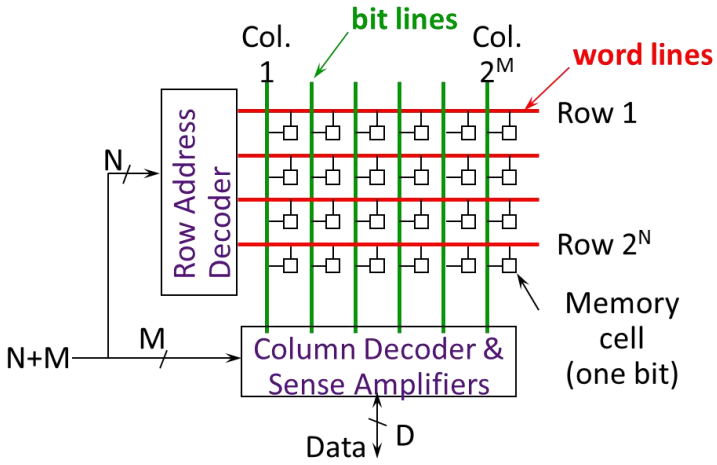
DRAM 阵列

- **DRAM**
  - 破坏性读：读后需要重新写回，必须要周期性的刷新；每位1个 transistor
  - 地址线复用:
    - Lower half of address: column access strobe (CAS)
    - Upper half of address: row access strobe (RAS)
- **SRAM**
  - 每位6个transistors；只需较低的功率来保持位状态



DRAM 阵列

- **DRAM是由单个信息位构成的阵列**
  - 通过行选择线和列选择线访问
  - 所有DRAM都由这些阵列构成
  - 不同的结构根据性能的需求选择的阵列数可能不同
- **所有DRAM的访问至少三个阶段**
  - Precharge， row access, column access
- **DRAM 的性能**
  - Latency
    - 地址信号有效到第一组数据信号有效所需要的时间
    - 处理器发出请求到所请求的第一组数据到达处理器输入引脚所需要的cycle数
  - Bandwidth
    - 第一组数据到达后，后续数据到达的速率



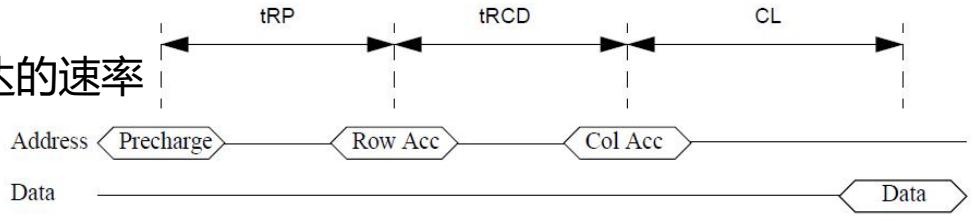
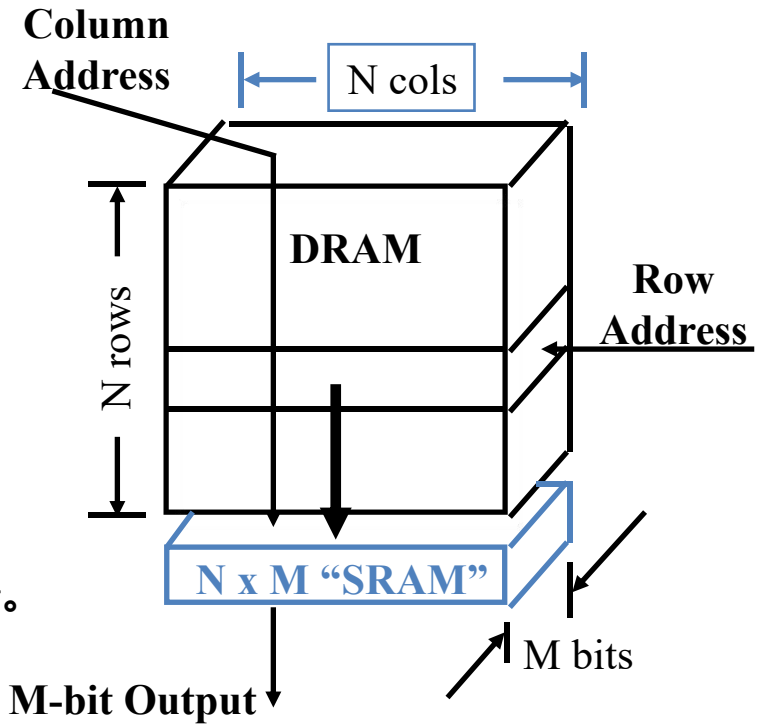
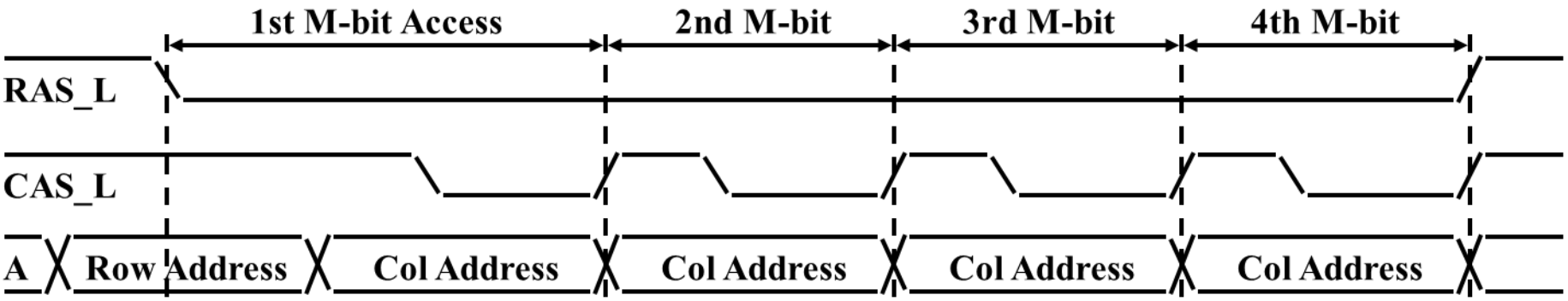


Figure 3.2: Timing Phases of a DRAM Access

DRAM device性能优化

- Fast Page Mode Operation (异步)
  - Multiple accesses to same row
  - A bank include Multiple DRAM Array
- Synchronous DRAM (同步)
  - Added clock to DRAM interface
  - Burst mode with critical word first
  - Multiple banks on each DRAM device
  - Single Data Rate (SDR) - **单速率数据传输**
    - 接受一个命令，一个时钟周期内传输一个数据。
  - Double data rate (DDR) - **双速率数据传输**
    - 在不增加时钟速度的情况下传输两倍的数据





SDRAM Device Architecture

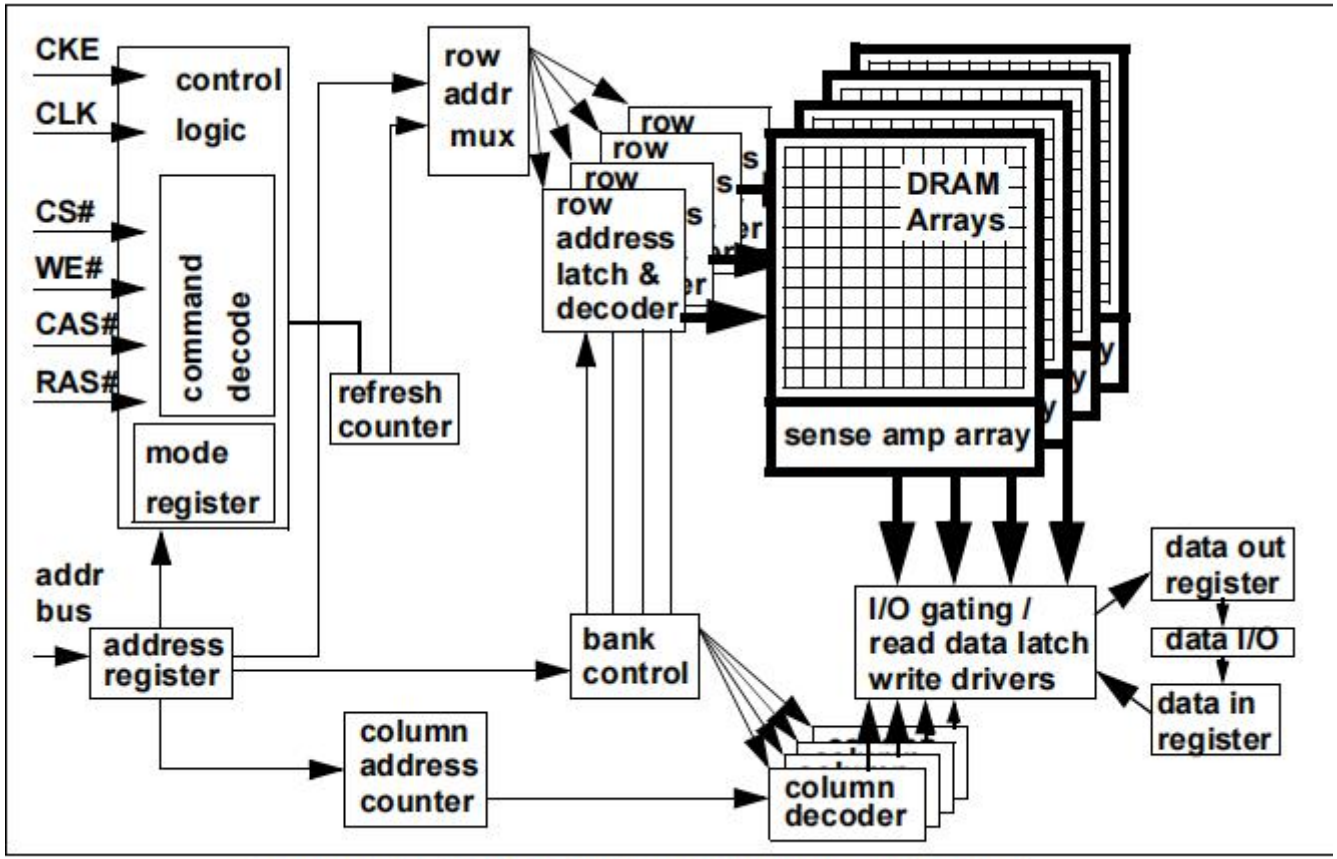
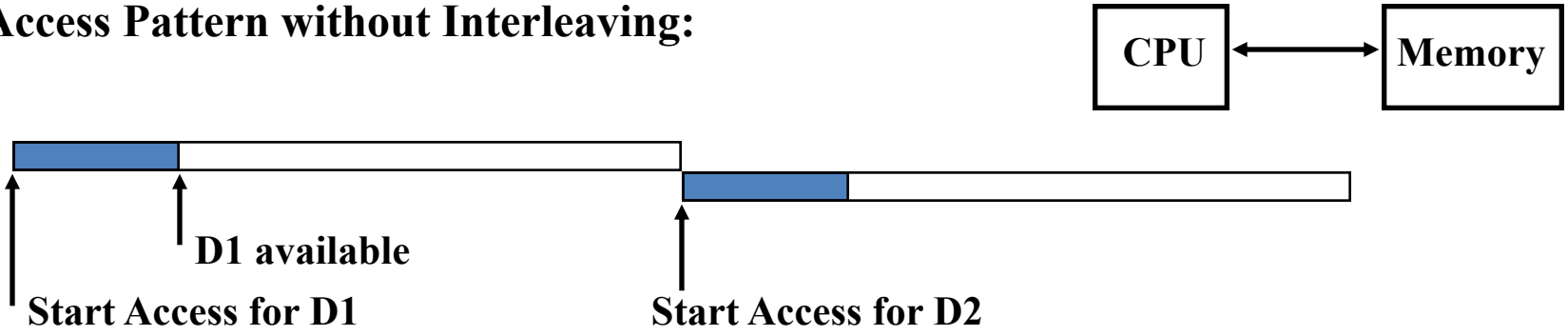


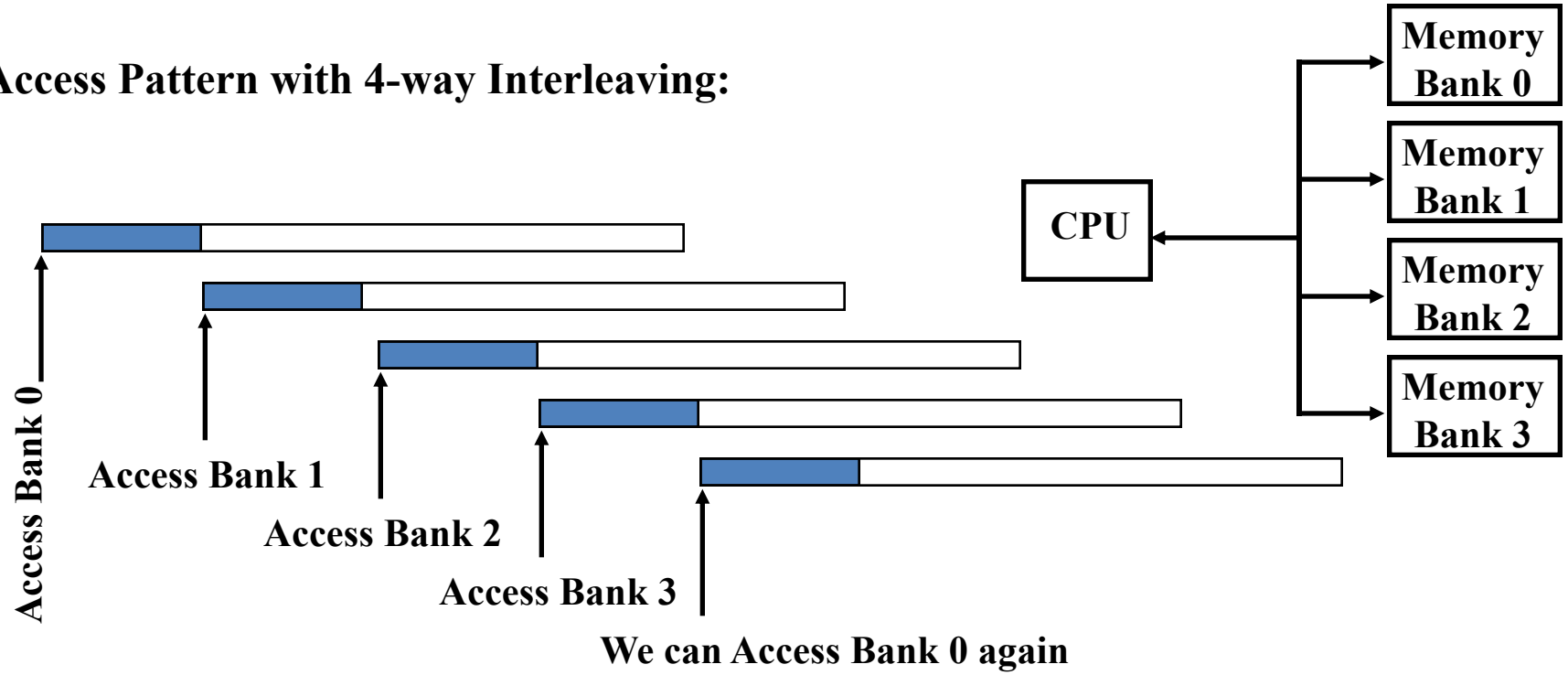
Figure 2.14: SDRAM Device Architecture with 4 Banks.

Increasing Bandwidth - Interleaving

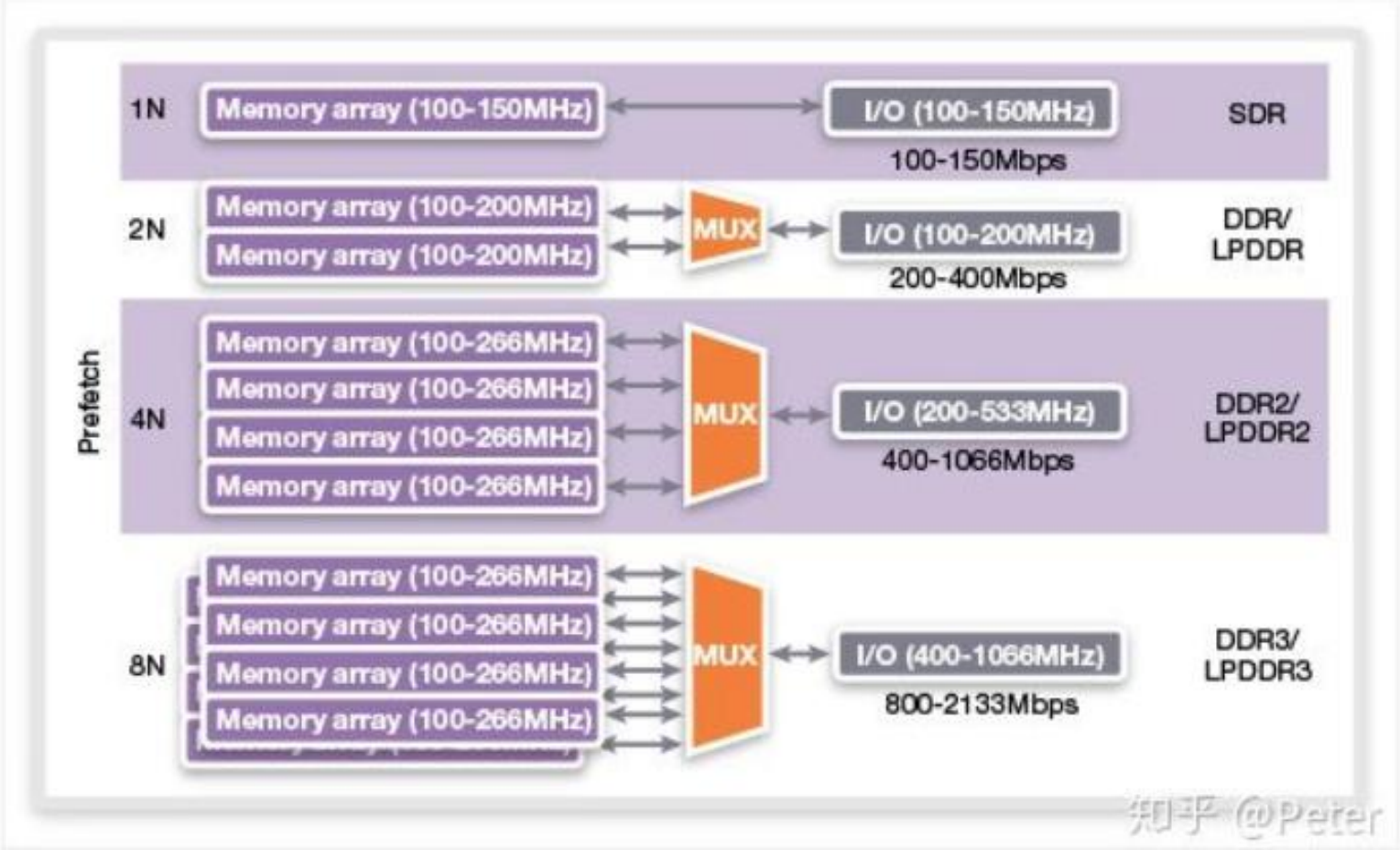
Access Pattern without Interleaving:



Access Pattern with 4-way Interleaving:

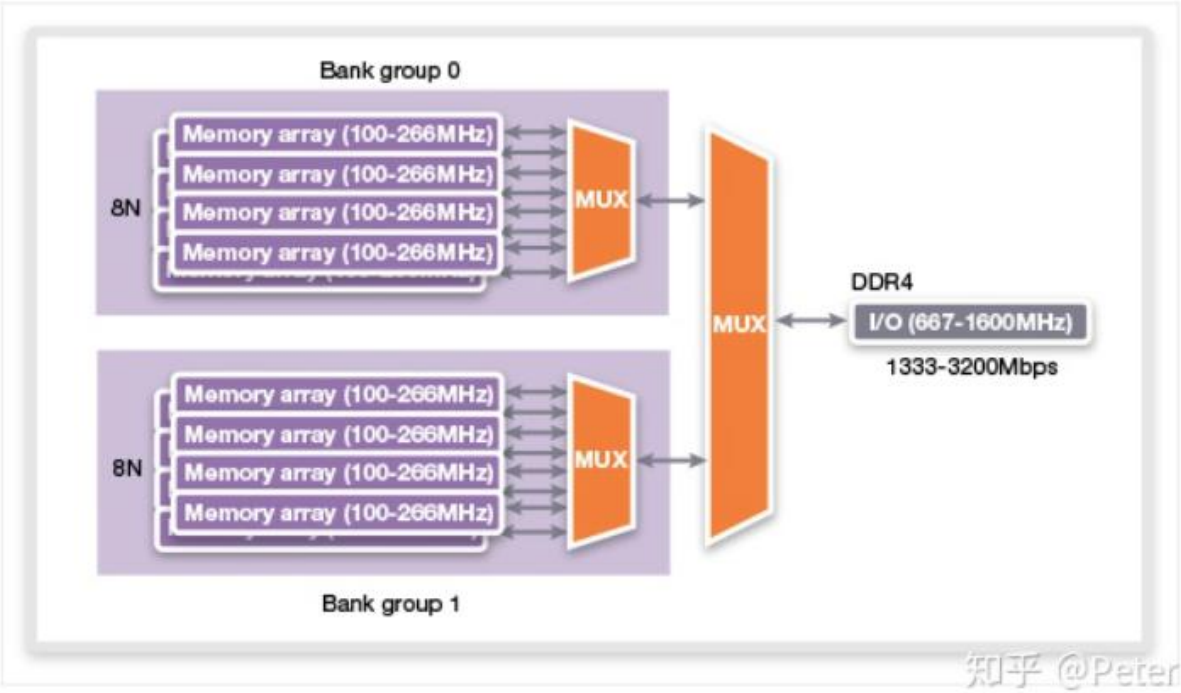


SDR、DDR、DDR2、DDR3



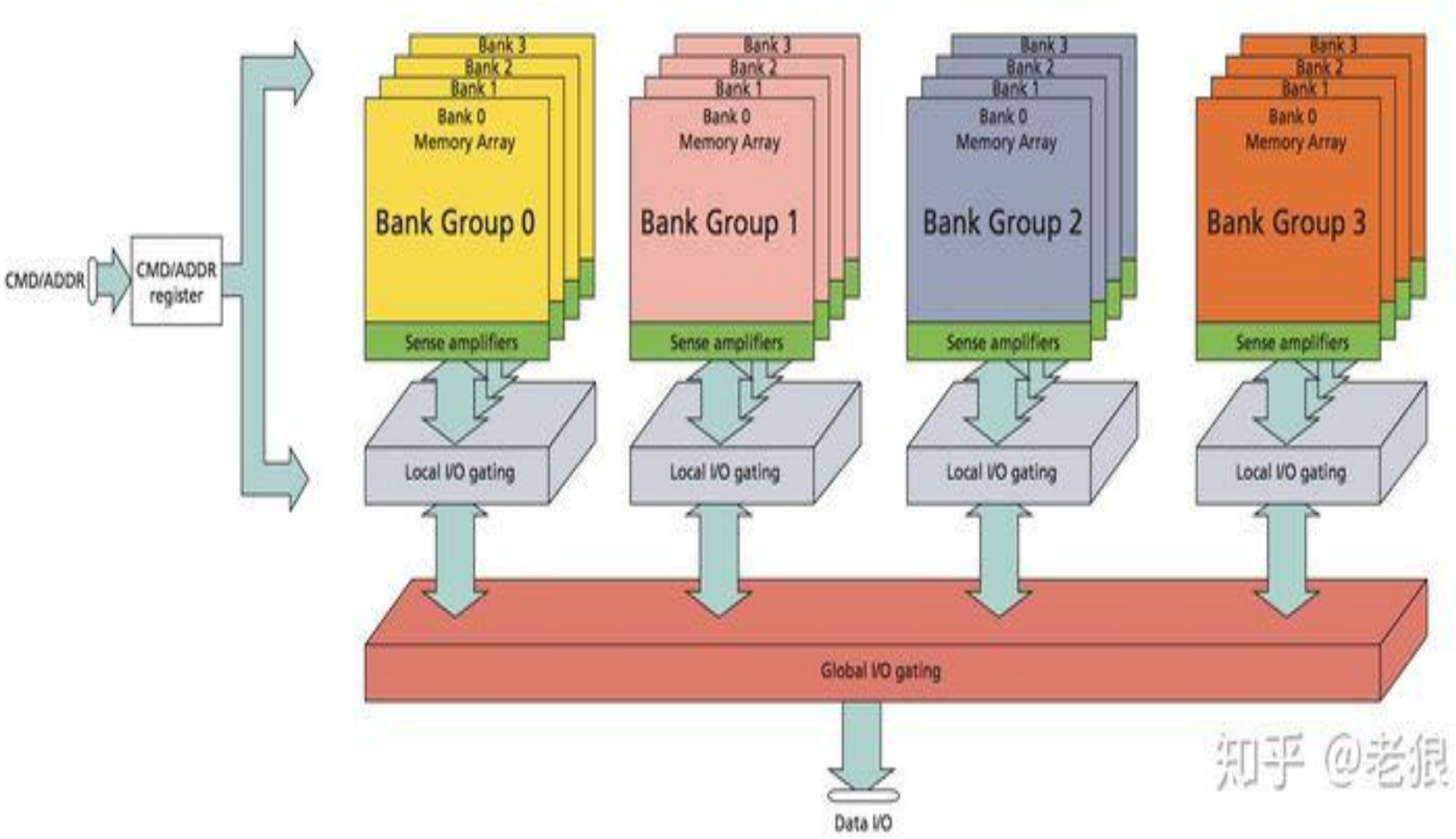
- **DDR：增加了预取机制**
  - DDR 2-bits; DDR2 4-bits DDR3 8-bits

DDR4



- **预取：每组预取8-bits**
- **增加了Bank Group：每个Bank Group可以独立读写数据**

DDR4 Bank Group



Memory 功耗

Reducing power in SDRAMs:

- Lower voltage
- Low power mode (ignores clock, continues to refresh)

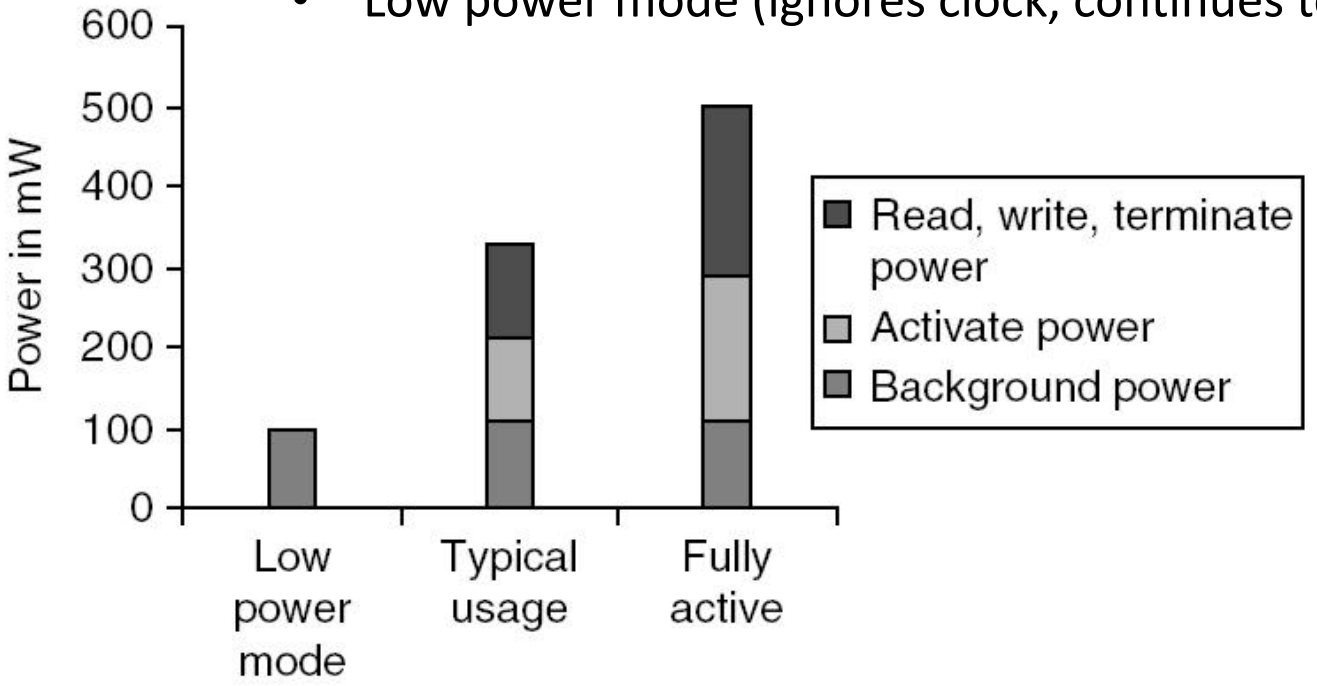


Figure 2.6 Power consumption for a DDR3 SDRAM operating under three conditions: low-power (shutdown) mode, typical system mode (DRAM is active 30% of the time for reads and 15% for writes), and fully active mode, where the DRAM is continuously reading or writing. Reads and writes assume bursts of eight transfers. These data are based on a Micron 1.5V 2GB DDR3-1066, although similar savings occur in DDR4 SDRAMs

4.4 存储器技术及优化

DRAM芯片 → 内存模组

内存子系统

- **内存子系统从CPU 到DRAM芯片的逐层关系**
  - Memory Controller
  - channel
  - DIMM
  - rank
  - chip
  - bank
  - row/column.

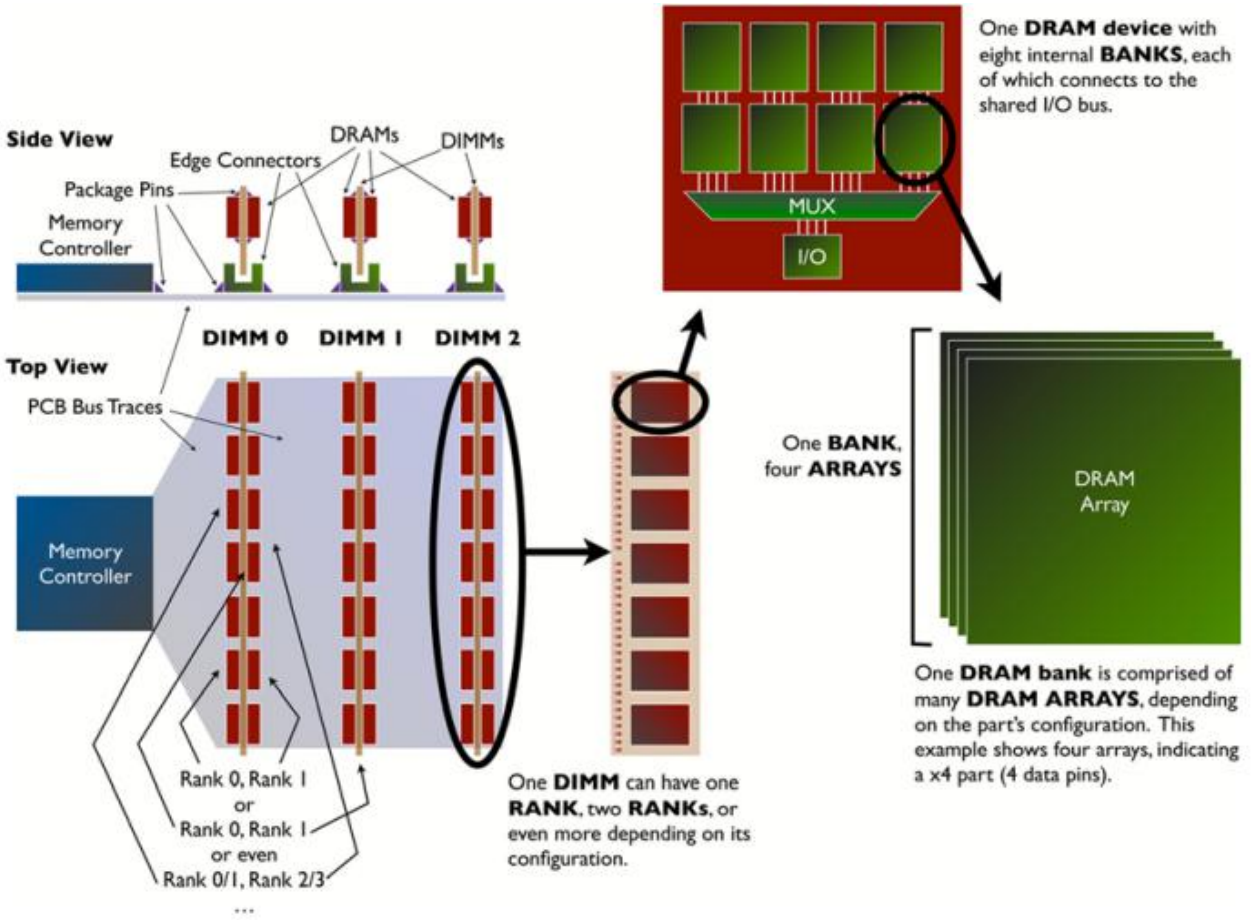


Figure 1.5: DIMMs, ranks, banks, and arrays. A system has potentially many DIMMs, each of which may contain one or more ranks. Each rank is a set of ganged DRAM devices, each of which has potentially many banks. Each bank has potentially many constituent arrays, depending on the parts data width.

Rank

- DIMM上可独立寻址一组存储芯片
- 一条DIMM可包含多个Rank
- DIMM的一面或两面，也称“DIMM级bank”
- 通过位扩展匹配内存总线的宽度。例如：
  - DRAM X4 16片 64bits
  - DRAM X8 8片 64bits

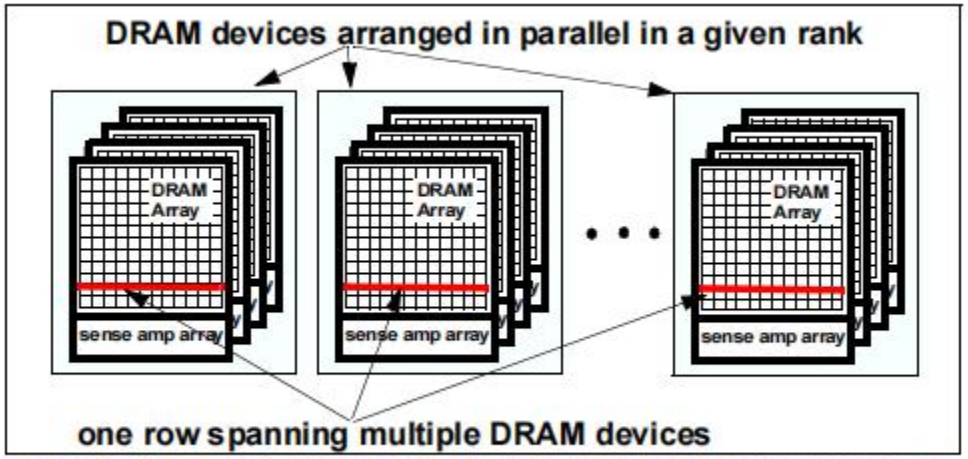


Figure 3.7: DRAM devices with 4 banks, 8192 rows per bank, 512 columns per row, and 16 bits per column.

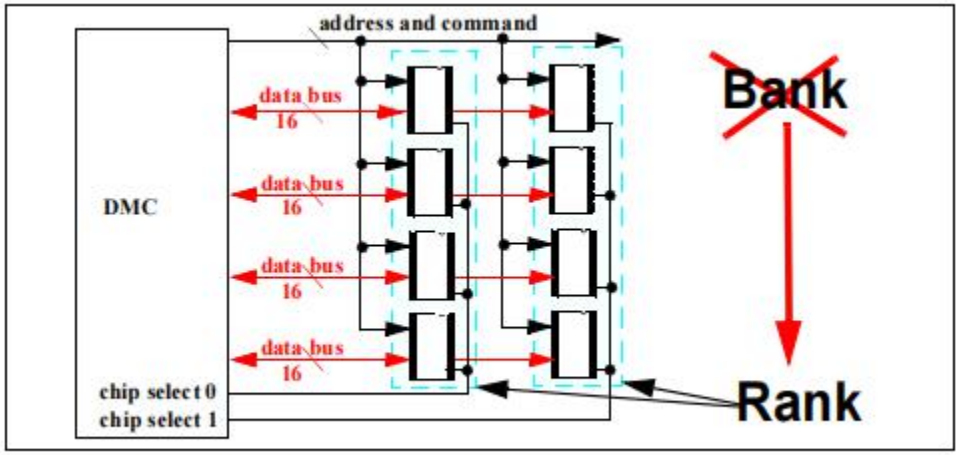
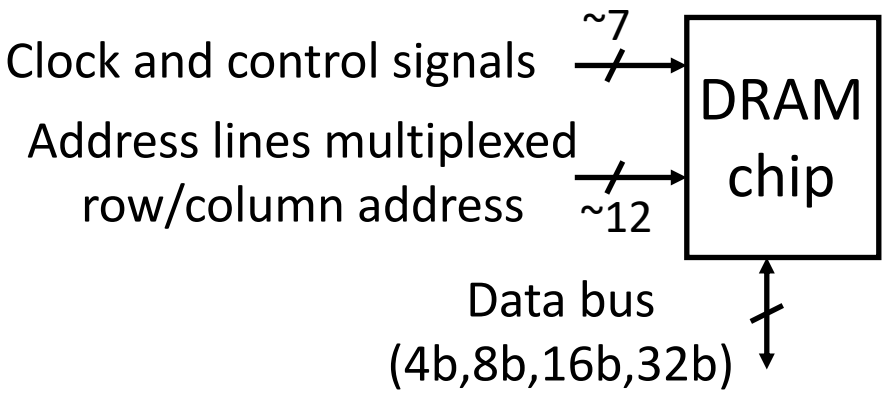
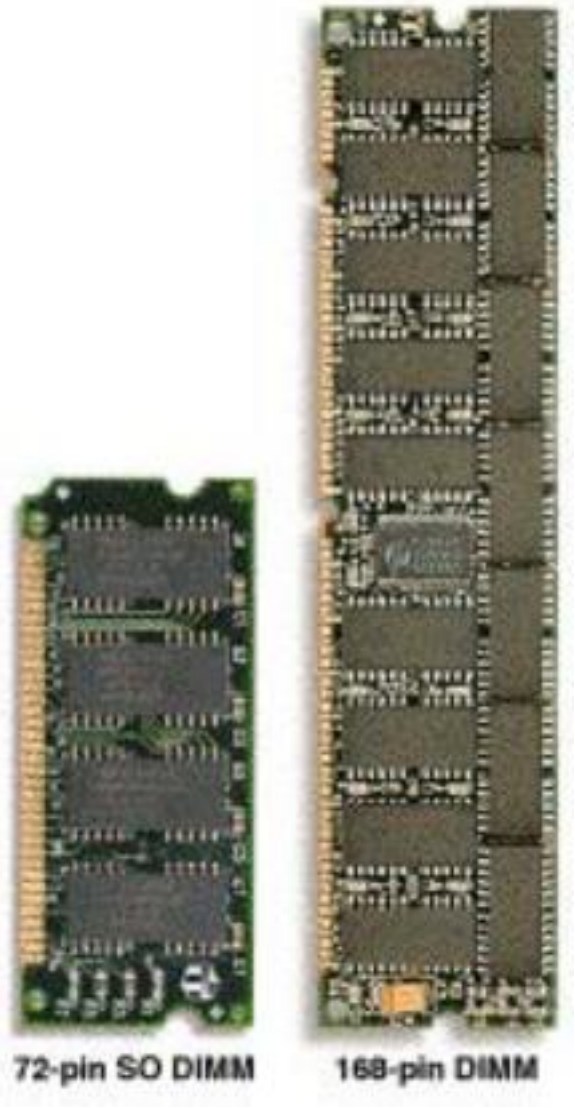


Figure 3.5: Memory System with 2 ranks of DRAM devices.

内存模组-DIMM





- **DIMM (Dual Inline Memory Module) 包含多个芯片，时钟/控制/地址信号并行连接(有时需要缓冲区驱动信号到所有芯片)**
- **Data pins 位扩展连接以形成更多的位数(e.g., 64-bit data bus using 16x4-bit parts)**

不同规格的DIMM

Standard	Clock rate (MHz)	M transfers per second	DRAM name	MB/sec /DIMM	DIMM name
DDR	133	266	DDR266	2128	PC2100
DDR	150	300	DDR300	2400	PC2400
DDR	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1066–1600	2133–3200	DDR4-3200	17,056–25,600	PC25600

Figure 2.14 Clock rates, bandwidth, and names of DDR DRAMS and DIMMs in 2010. Note the numerical relationship between the columns. The third column is twice the second, and the fourth uses the number from the third column in the name of the DRAM chip. The fifth column is eight times the third column, and a rounded version of this number is used in the name of the DIMM. Although not shown in this figure, DDRs also specify latency in clock cycles as four numbers, which are specified by the DDR standard. For example, DDR3-2000 CL 9 has latencies of 9-9-9-28. What does this mean? With a 1 ns clock (clock cycle is one-half the transfer rate), this indicate 9 ns for row to columns address (RAS time), 9 ns for column access to data (CAS time), and a minimum read time of 28 ns. Closing the row takes 9 ns for precharge but happens only when the reads from that row are finished. In burst mode, transfers occur on every clock on both edges, when the first RAS and CAS times have elapsed. Furthermore, the precharge in not needed until the entire row is read. DDR4 will be produced in 2012 and is expected to reach clock rates of 1600 MHz in 2014, when DDR5 is expected to take over. The exercises explore these details further.

内存系统结构框图

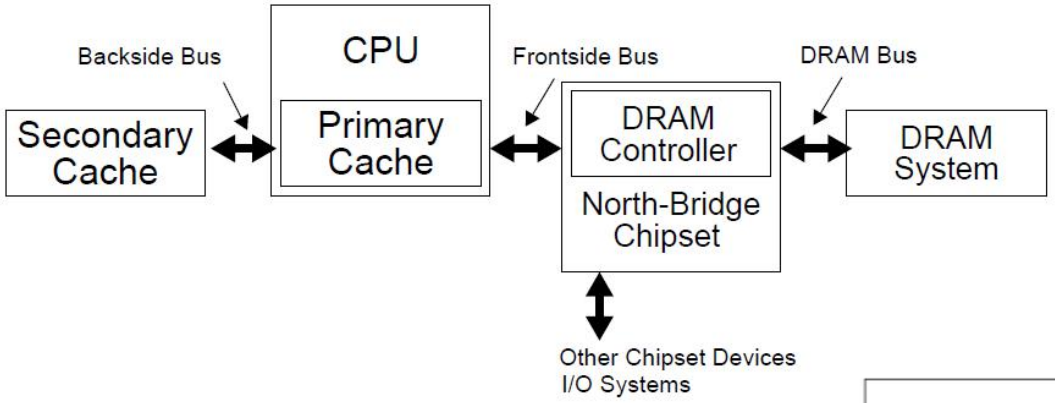
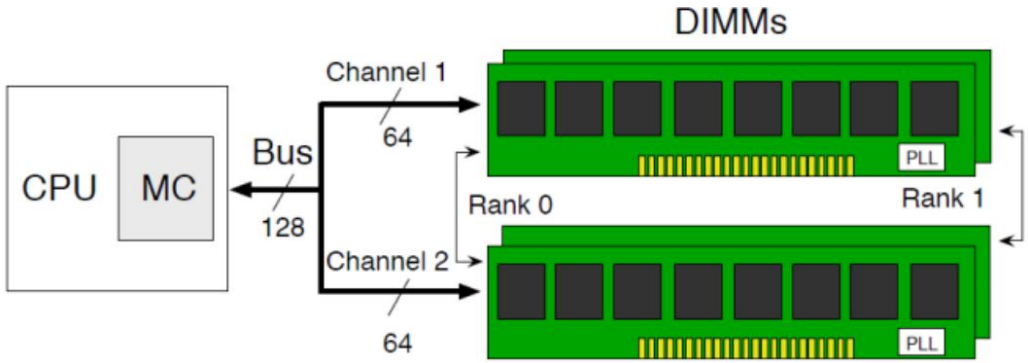
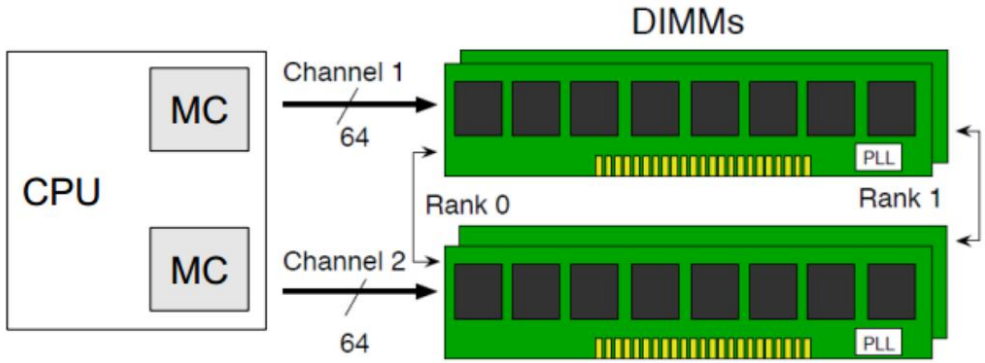


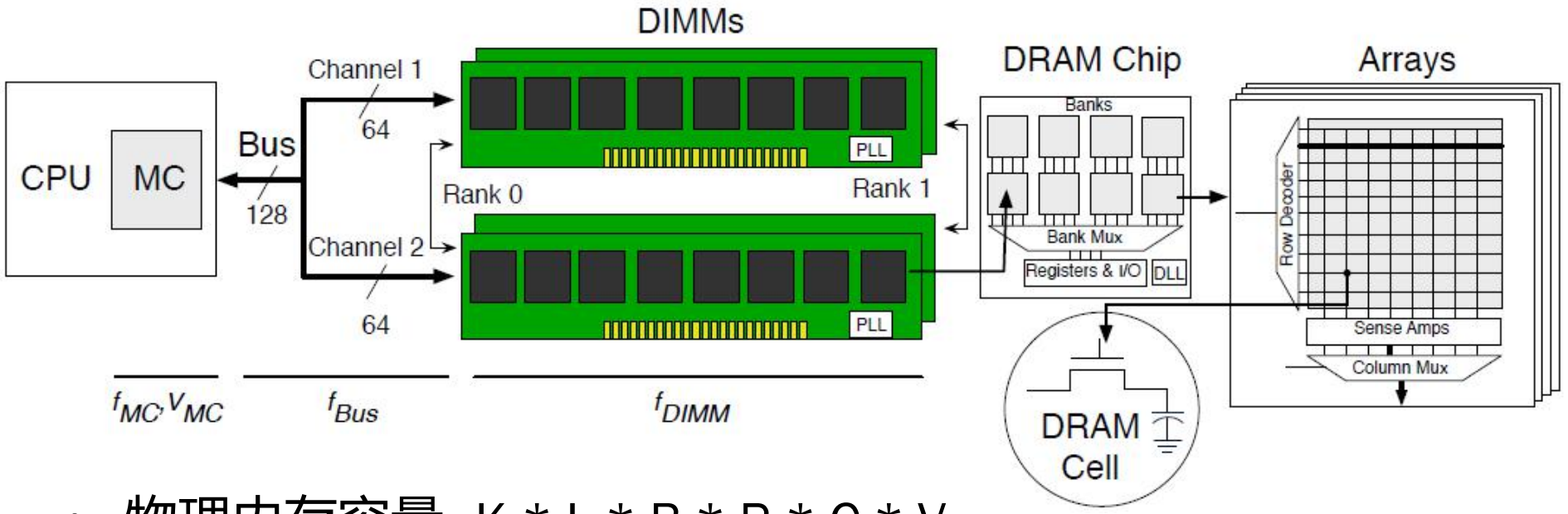
Figure 4.1: Memory System Architecture



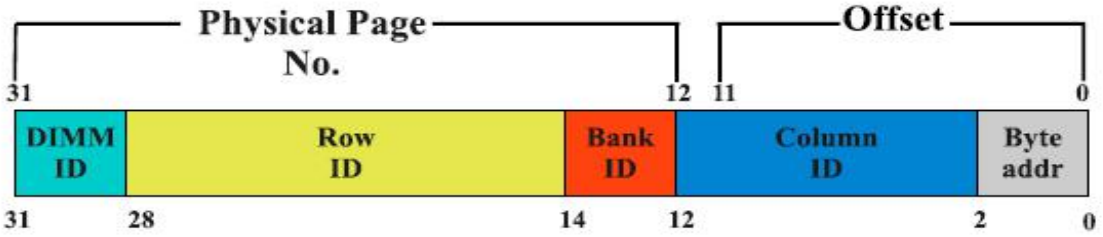
内存控制器
Channel



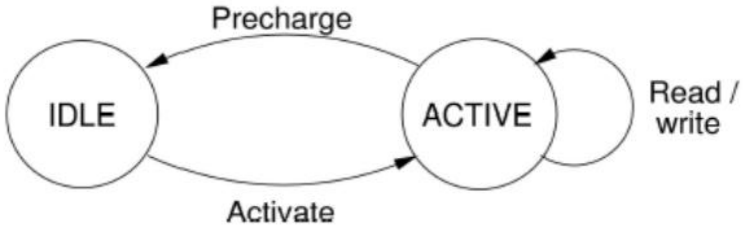
Channel：Multi-rank



- 物理内存容量=K * L * B * R * C * V.
  - K：channels(DIMM ID?)
  - L：ranks per channel
  - B：banks per rank
  - R：rows per bank
  - C：columns per row
  - V：bytes per column.



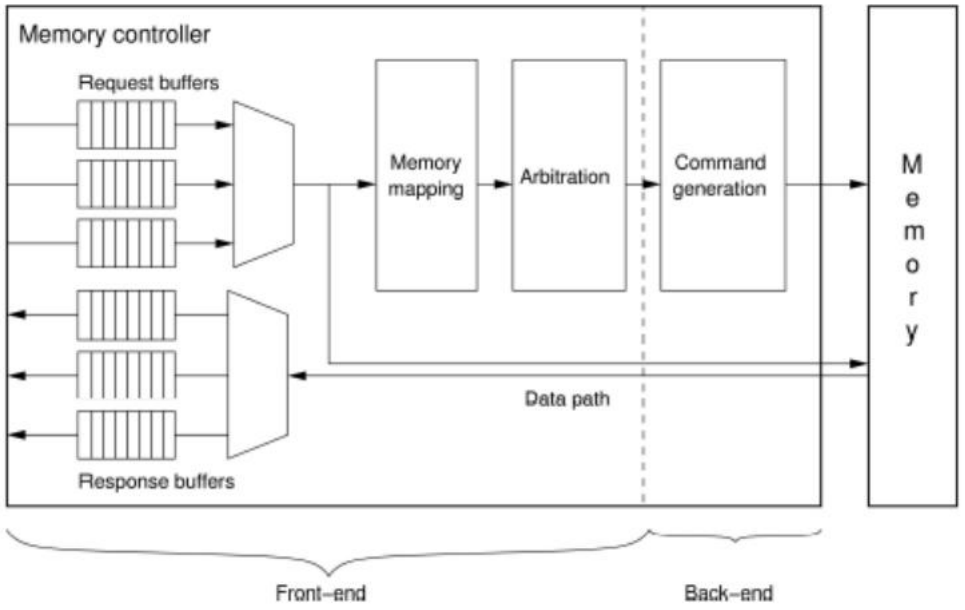
DRAM Command Summary



No operation	NOP	Ignores all inputs
Activate	ACT	Activate a row in a particular bank
Read	RD	Initiate a read burst to an active row
Write	WR	Initiate a write burst to an active row
Precharge	PRE	Close a row in a particular bank
Refresh	REF	Start a refresh operation

内存控制器

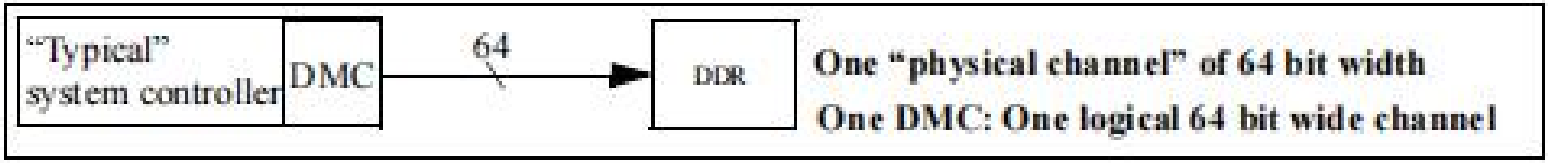
- **控制器前端**
  - Request/Response Buffers
  - Memory mapping
  - Arbiter
- **控制器后端**
  - Command Generator
- **满足读写时序**



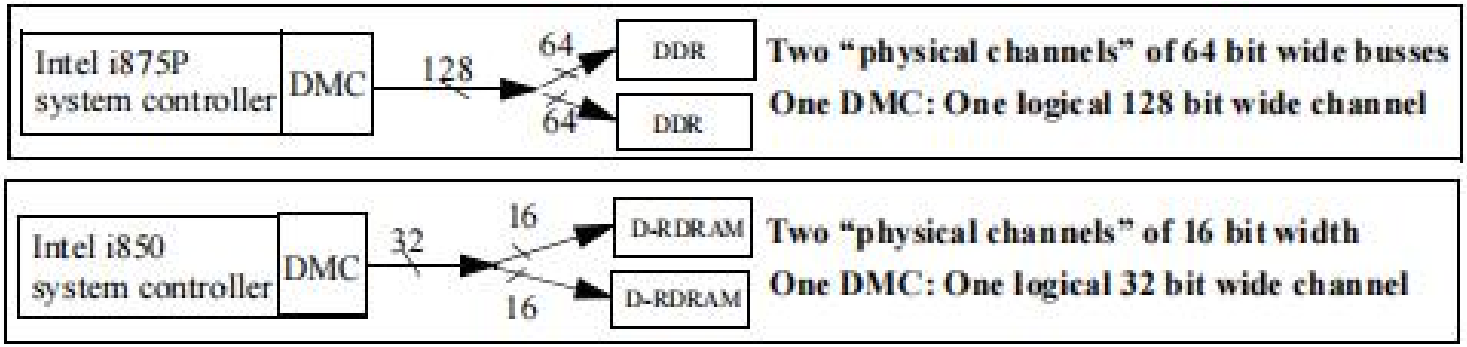
Command delay	Cycles
ACT to PRE	9
ACT to ACT (same bank)	12
ACT to ACT (diff. bank)	2
ACT to RD/WR	3
WR to RD turn-around	2

内存控制器与Channel

- **典型的每个内存控制器管理一个Channel，每个Channel 位宽64位 (DIMM)**



- **其他配置。例如：**



4 Channels of Interleaved FPM Devices

- **FPM devices 4个Channels多体交叉方式访问**

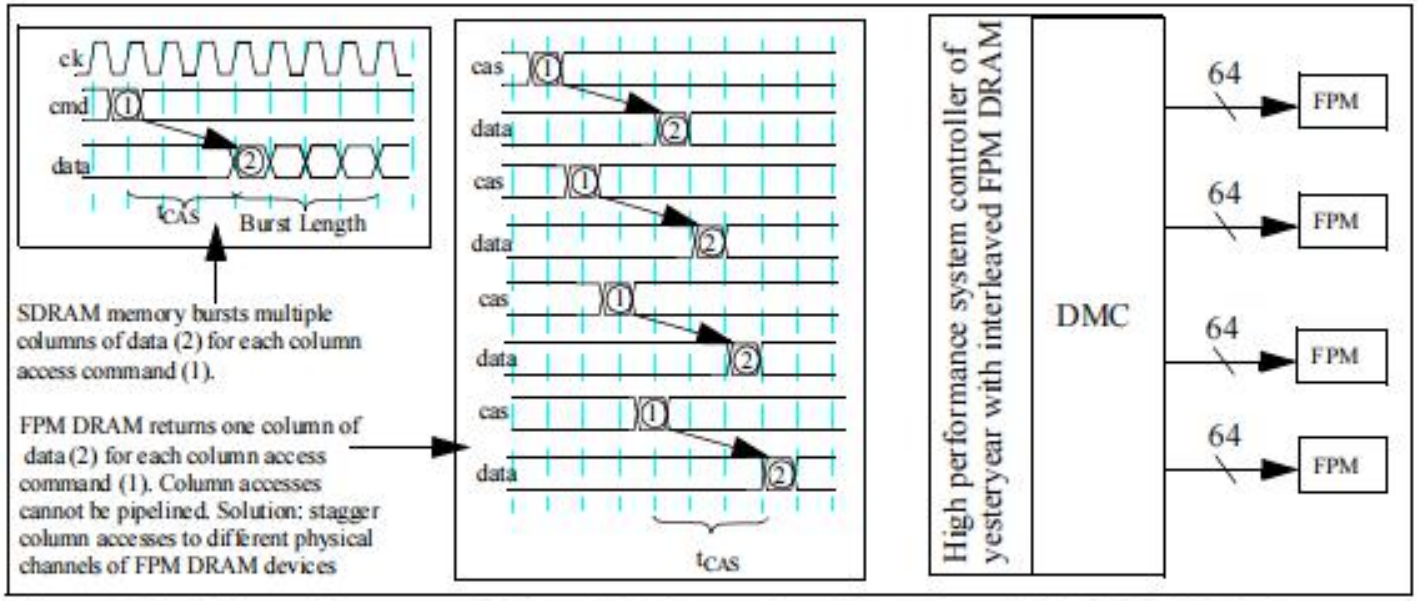


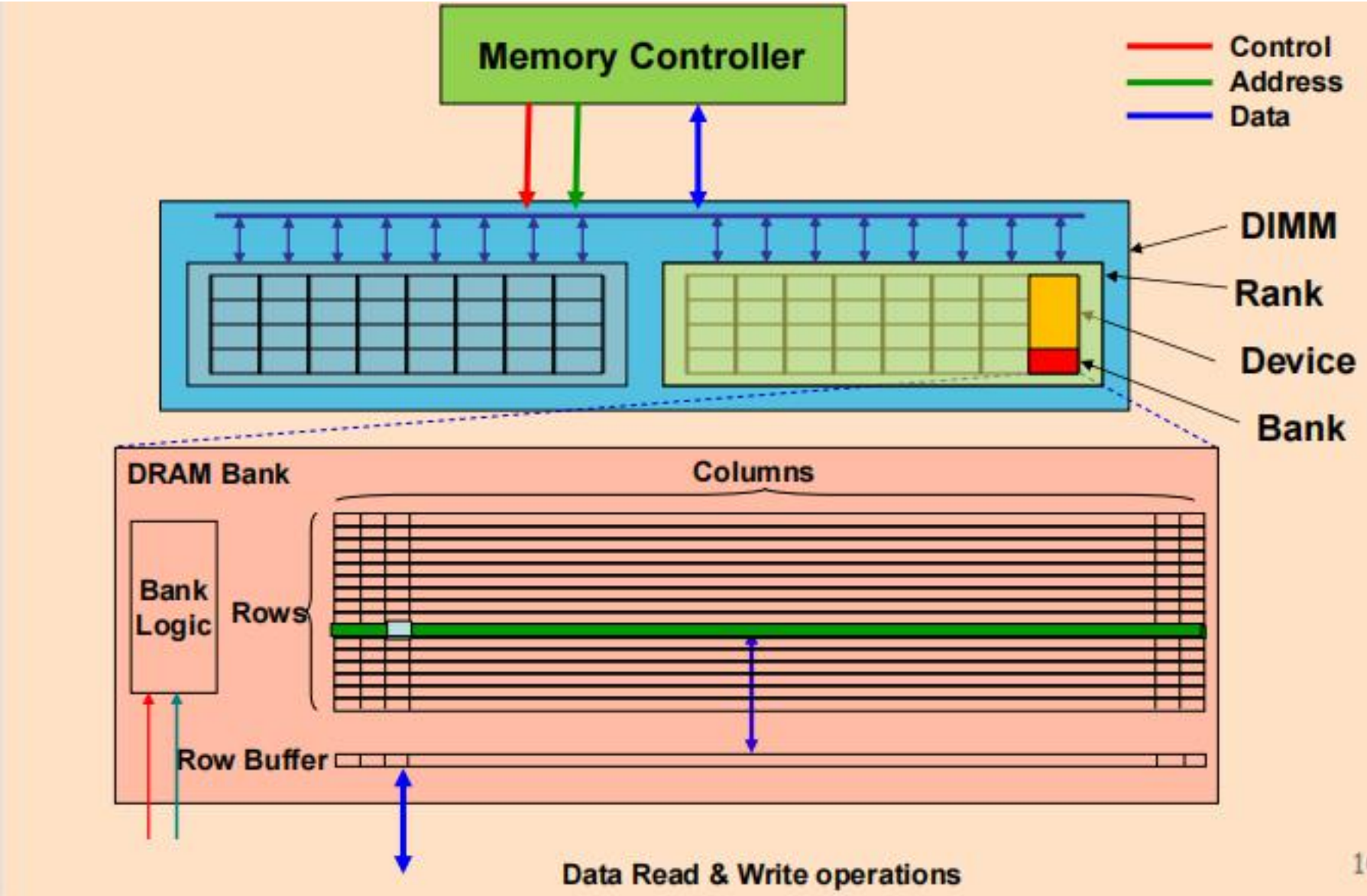
Figure 3.4: High performance DMC with 4 channels of interleaved FPM DRAM devices.

参考文献

Davis, B. T. (2001). Modern dram architectures, University of Michigan: 221.

Jacob, B. (2009). The Memory System, Morgan & Claypool.

小结：内存子系统


Memory Controller
Control
Address
Data
DIMM
Rank
Device
Bank
DRAM Bank
Columns
Bank Logic
Rows
Row Buffer
Data Read & Write operations

10

4.5 虚拟存储

虚拟存储回顾

TLB

虚拟存储器 - 基本原理

- **允许应用程序的大小，超过主存容量。目的是提高存储系统的容量**
- **帮助OS进行多进程管理**
  - 每个进程可以有自己的地址空间
  - 提供多个进程空间的保护
  - 可以将多个逻辑块映射到共享的物理存储器上
  - 静态重定位和动态重定位
    - 应用程序运行在虚地址空间
    - 虚实地址转换对用户是透明的
- **虚拟存储管理的是主存 - 辅助存储器这个层面上**
  - 失效：页失效或地址失效
  - 块：页或段

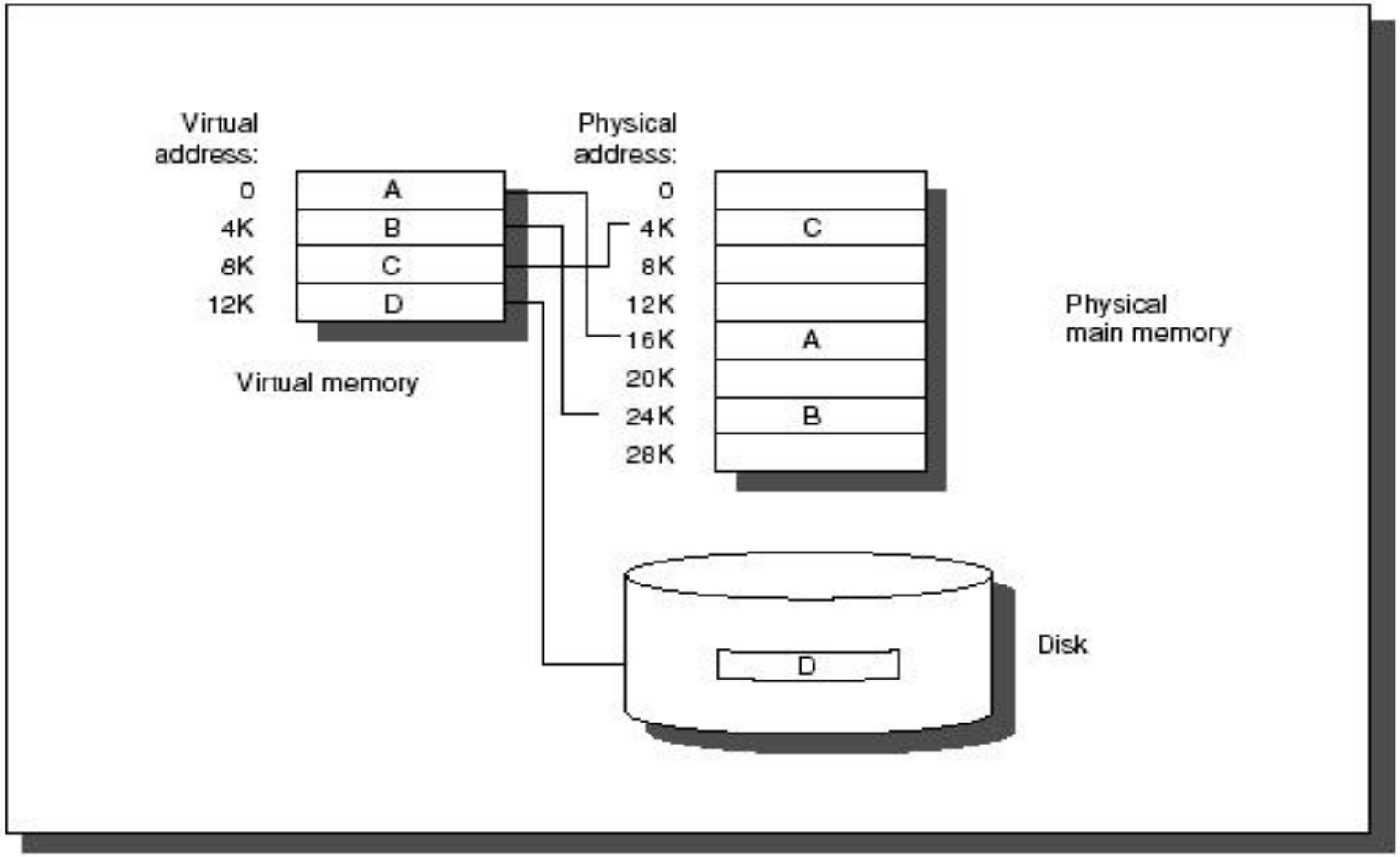


FIGURE 5.31 The logical program in its contiguous virtual address space is shown on the left. It consists of four pages A, B, C, and D. The actual location of three of the blocks is in physical main memory and the other is located on the disk.

Cache与VM的区别

- **目的不同**
  - Cache是为了提高访存速度
  - VM是为了提高存储容量
- **替换的控制者不同**
  - Cache失效由硬件处理
  - VM的页失效通常由OS处理
    - 一般页失效开销很大，因此替换算法非常重要
- **地址空间**
  - VM空间由CPU的地址尺寸确定
  - Cache的大小与CPU地址尺寸无关
- **下一级存储器**
  - Cache下一级是主存
  - VM下一级是磁盘，大多数磁盘含有文件系统，文件系统寻址与主存不同，它通常在I/O空间中，VM的下一级通常称为SWAP空间

虚拟存储器页式管理的典型参数与Cache的比较

Parameter	First-level cache	Virtual memory
Block (page) size	16–128 bytes	4096–65,536 bytes
Hit time	1–3 clock cycles	100–200 clock cycles
Miss penalty	8–200 clock cycles	1,000,000–10,000,000 clock cycles
(access time)	(6–160 clock cycles)	(800,000–8,000,000 clock cycles)
(transfer time)	(2–40 clock cycles)	(200,000–2,000,000 clock cycles)
Miss rate	0.1–10%	0.00001–0.001%
Address mapping	25–45 bit physical address to 14–20 bit cache address	32–64 bit virtual address to 25–45 bit physical address

Figure C.19 Typical ranges of parameters for caches and virtual memory. Virtual memory parameters represent increases of 10–1,000,000 times over cache parameters. Normally first-level caches contain at most 1 MB of data, while physical memory contains 256 MB to 1 TB.

- 从表中看 （与Cache参数相比）
  - **除了失效率较低，其他参数都比Cache大**

页式管理和段式管理

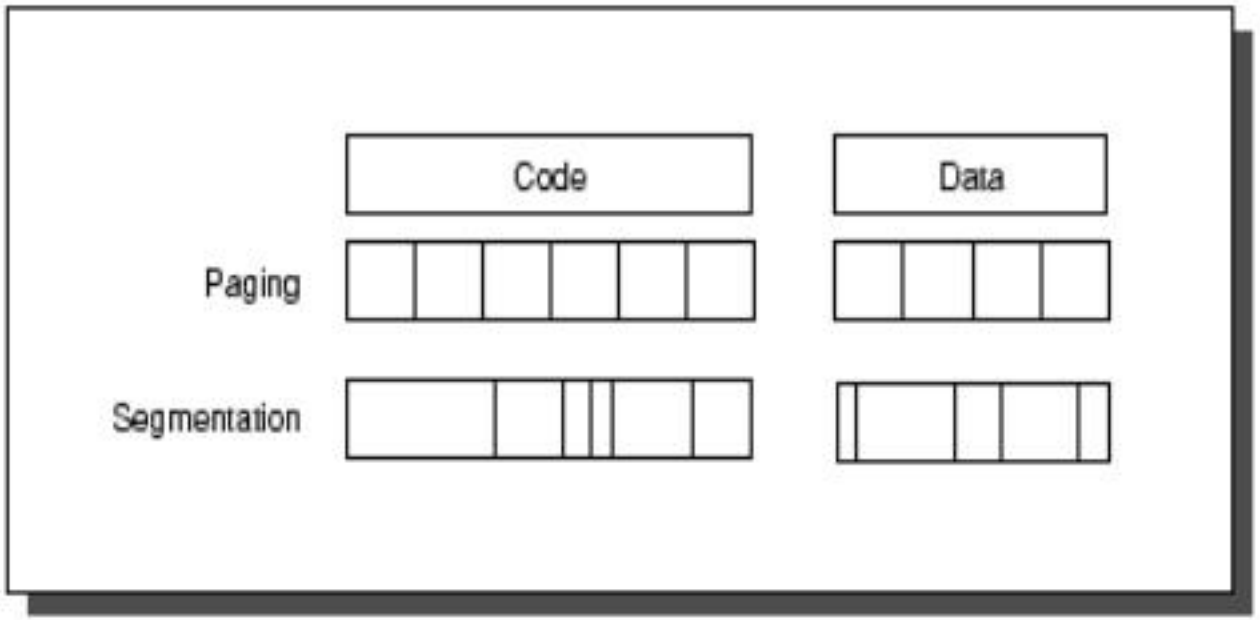


FIGURE 5.33 Example of how paging and segmentation divide a program.

页式管理和段式管理

Aspect	Page	Segment
Words/Address	One - contains page and offset	Two - possible large max-size hence need Seg and offset address words
Programmer visible	No	Sometimes yes
Replacement	Trivial - due to fixed size	Hard - need to find contiguous space ==> GC necessary or wasted memory
Memory Inefficiency	Internal fragmentation - wasted part of a page	External fragmentation - due to variable size blocks
Disk Efficiency	Yes - adjust page size to balance access and transfer time	Not always - segment size varies

- VM可分为两类：页式和段式
  - 页式：每页大小固定
  - 段式：每段大小不等
  - 两者区别：

VM的四个问题 (1/2)

- **映象规则**
  - 选择策略：低失效率（复杂映象算法） vs. 高失效率（简单映射方法），**由于失效开销很大，一般选择低失效率方法，即全相联映射**
- **查找算法 - 用附加数据结构**
  - 固定页大小 - 用页表
    - VPN - > PPN
    - Tag标识该页是否在主存
  - 可变长段 - 段表
    - 段表中存放所有可能的段信息
    - 段号 - >段基址 再加段内偏移量
    - 可能存在许多小尺寸段
  - 页表
    - 页表中所含项数：一般为虚页的数量
    - 功能: VPN - >PPN，方便页重新分配，有一位标识该页是否在内存

VM的四个问题（2/2)

- **替换规则**
  - LRU是最好的
  - 但真正的LRU方法，硬件代价较大
  - 用硬件简化，通过OS来完成
    - 为了帮助OS寻找LRU页，每个页面设置一个 use bit
    - 当访问主存中一个页面时，其use bit置位
    - OS定期复位所有使用位，这样每次复位之前，使用位的值反映了从上次复位到现在的这段时间中，哪些页曾被访问过。
    - 当有失效冲突时，由OS来决定哪些页将被换出去。
- **写策略**
  - 总是用写回法，因为访问硬盘速度很慢。

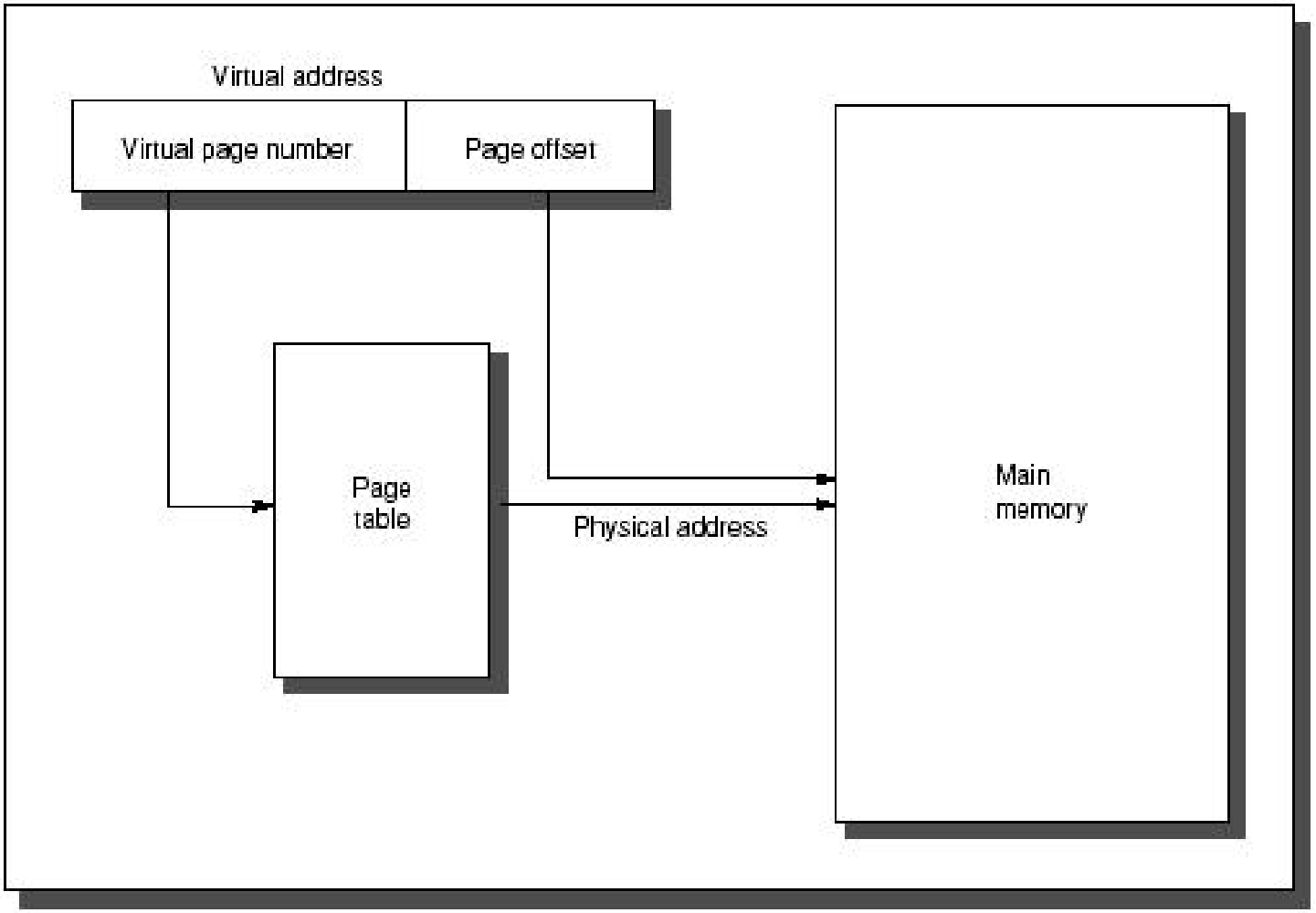
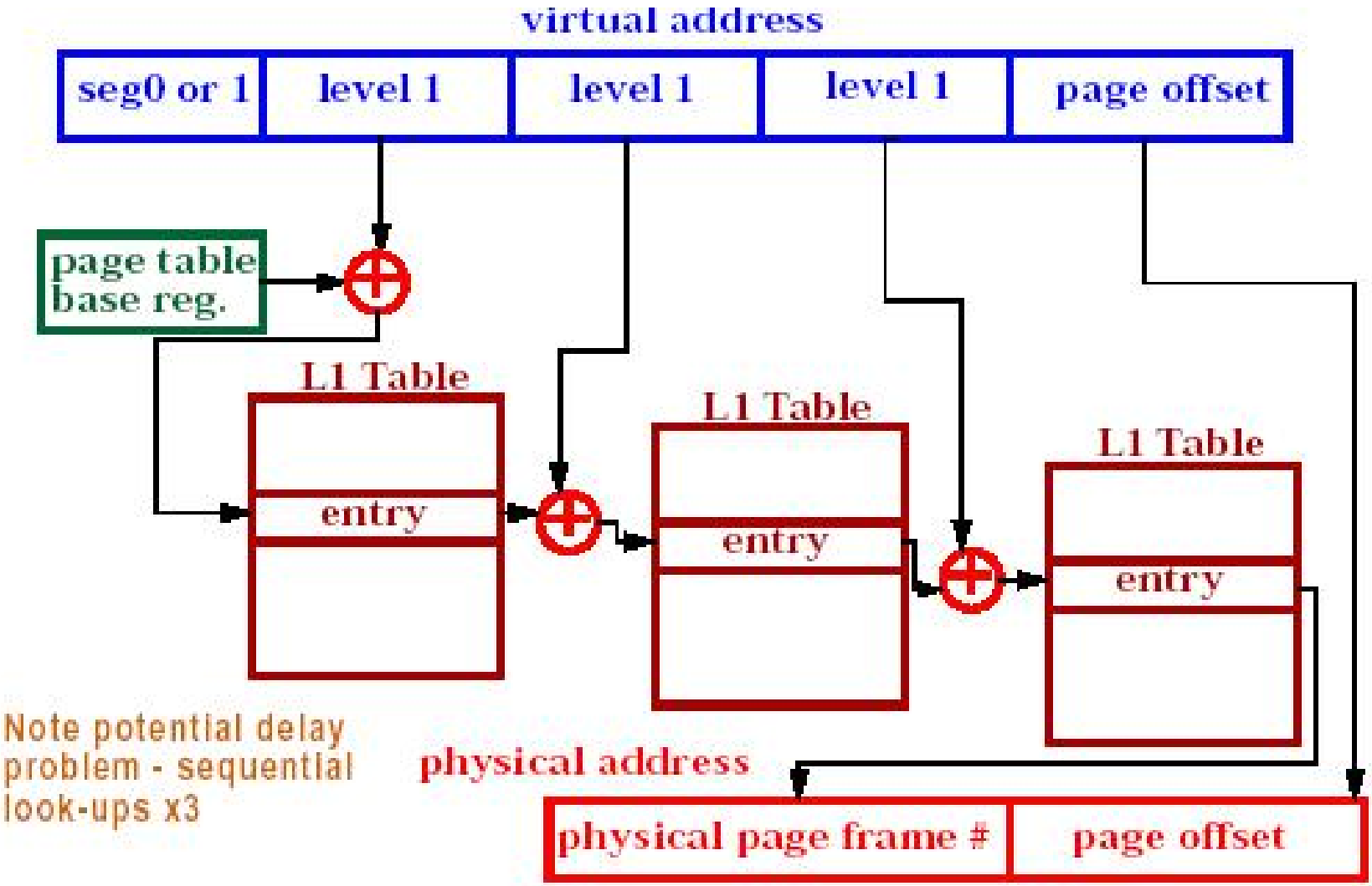


FIGURE 5.35 The mapping of a virtual address to a physical address via a page table.

页面大小的选择

- **页面选择较大的优点**
  - 减少了页表的大小
  - 如果局部性较好，可以提高命中率
- **页面选择较大的缺点**
  - 内存中的碎片较多，内存利用率低
  - 进程启动时间长
  - 失效开销加大

Alpha VPN - >PPN



review -内存子系统

- **内存子系统的组成**
  - Memory Controller
  - Channel
  - DIMM（内存模组）
  - rank
  - chip
  - bank
  - DRAM Array
    - row/column
- **性能优化的手段**
  - FPM
  - SDRAM
    - SDR，DDR
  - 多存储体多体交叉
  - 独立多存储体，并行访问

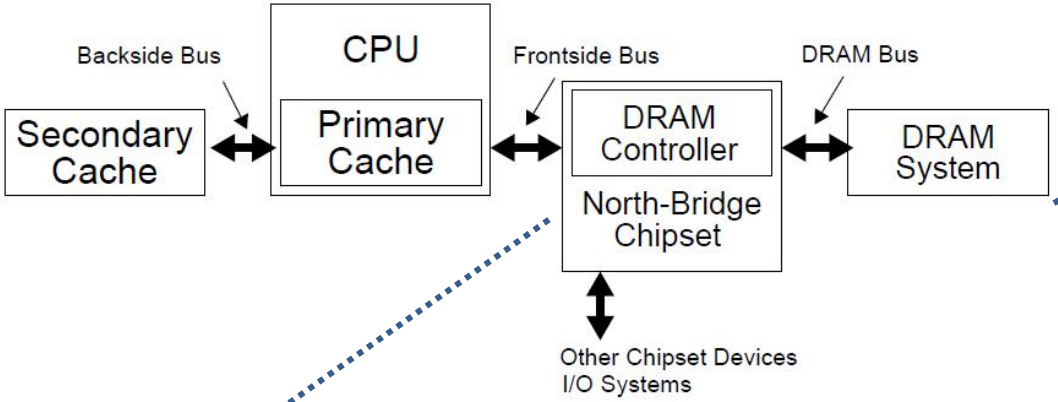
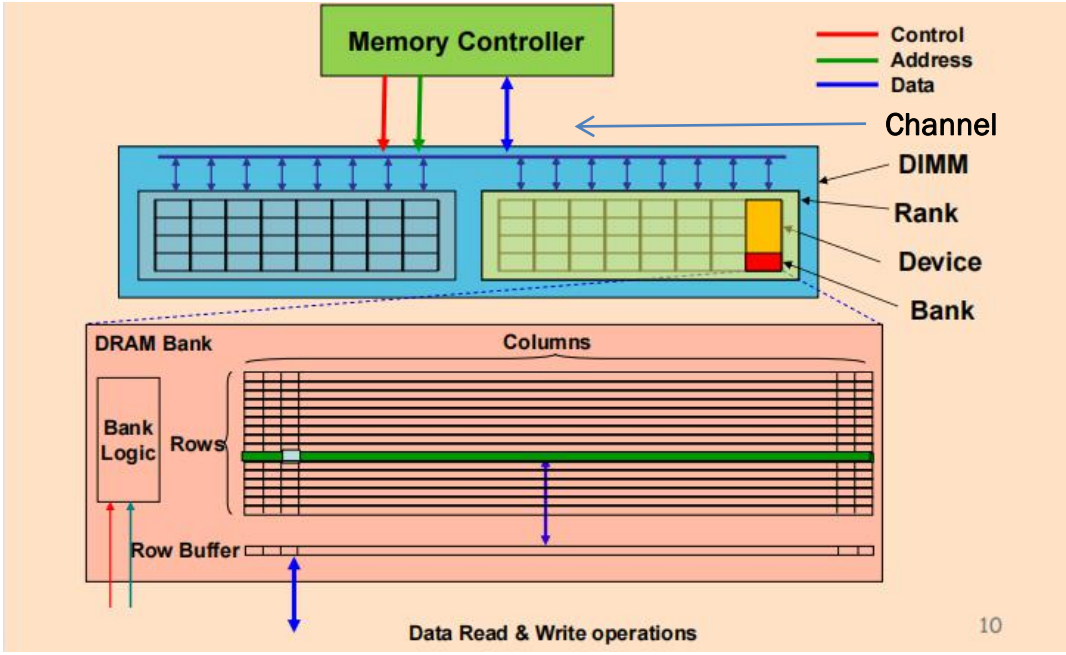
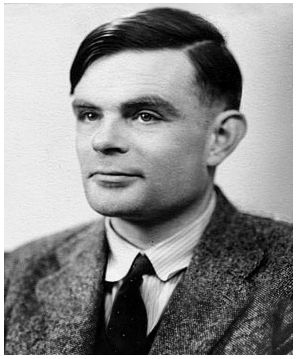


Figure 4.1: Memory System Architecture



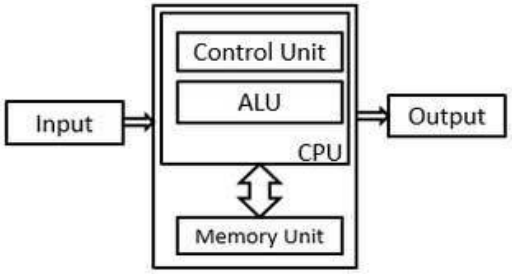
Review：虚拟存储-存储器地址空间

- **线性地址空间(Linear address space):**
  - 连续非负整数地址的有序集合{0, 1, 2,… }
  - 图灵计算模型(1936)中的存储器：一条无限长的带子



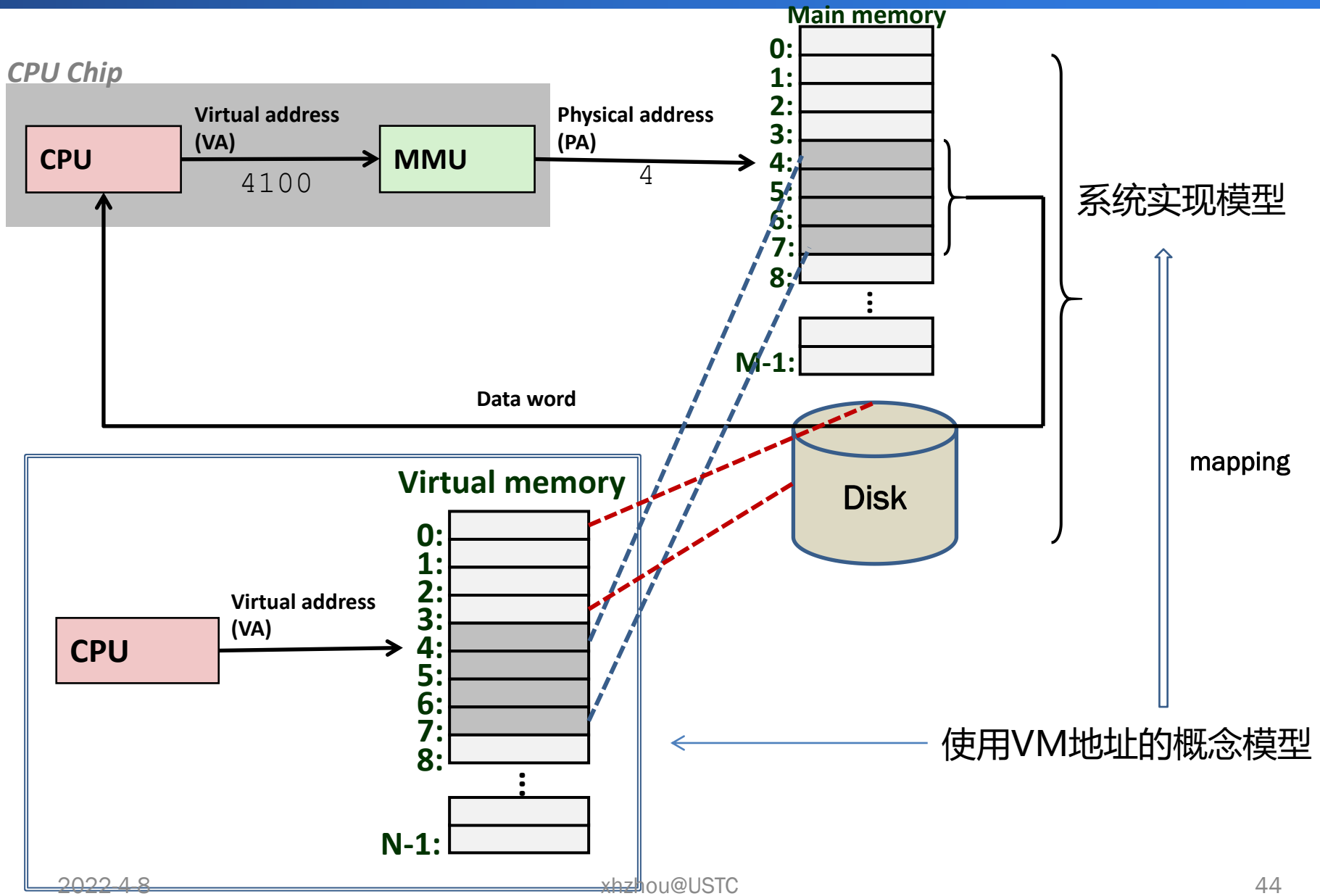
- **虚拟地址空间(Virtual address space):**
  - 虚拟地址的集合N = 2^n 、{0, 1, 2, …, N-1}，地址长度为n

- **物理地址空间(Physical address space):**
  - 物理地址的集合M = 2^m、{0, 1, 2, ..., M-1}，地址长度为m



Von Neumann Model 1945年

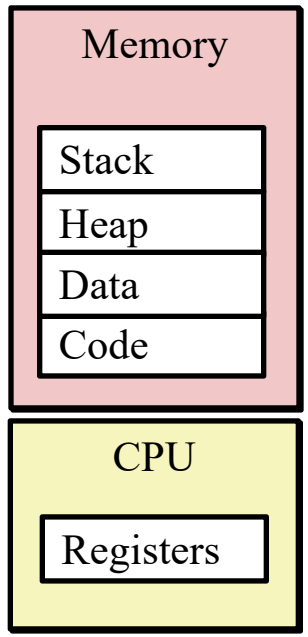
使用虚拟地址的计算机系统（从CA的角度）



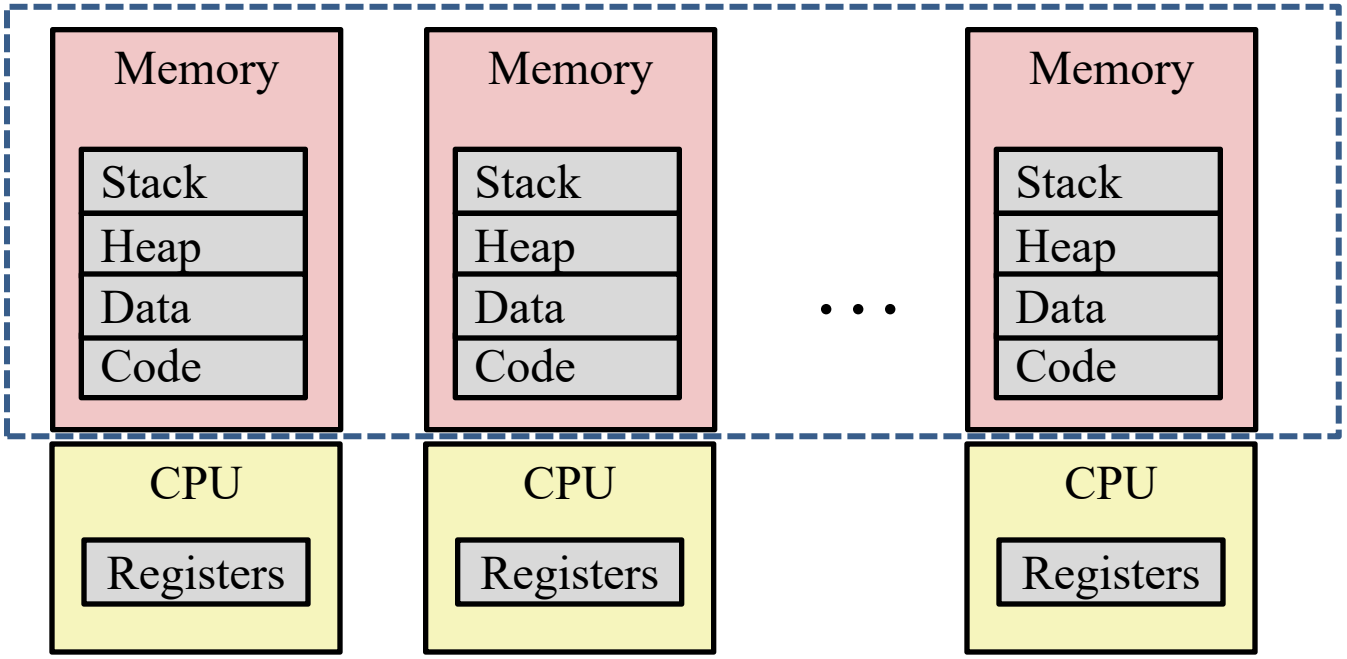
从OS的角度看VM： Processes

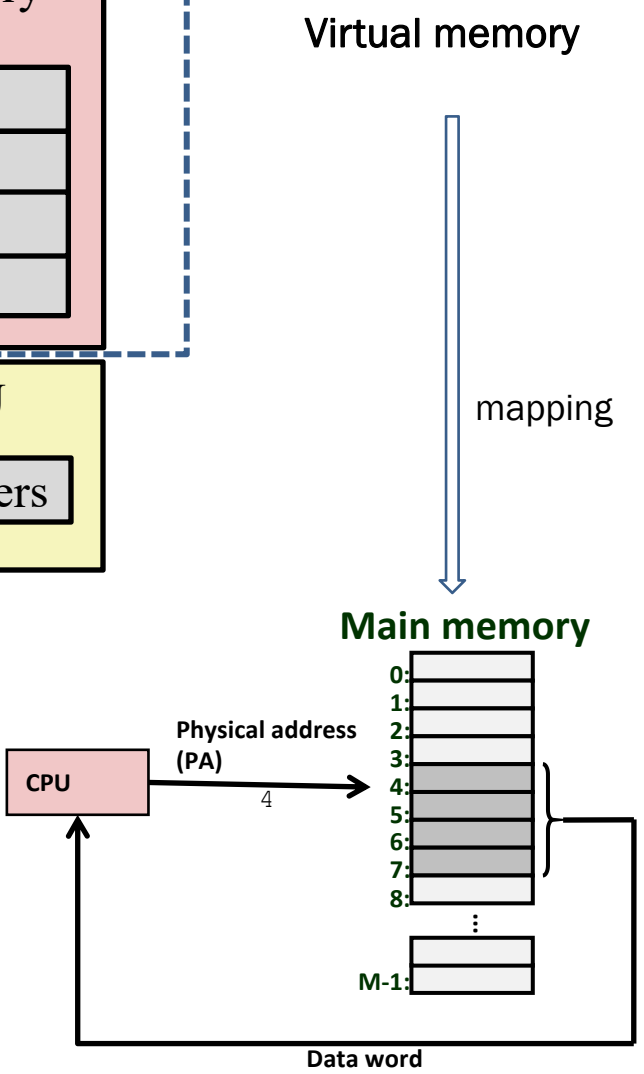
- **进程: A *process* 是正在运行的程序的实例**
  - One of the most profound ideas in computer science
  - Not the same as “program” or “processor”

- **Process 为每个程序提供两个关键抽象:**
  - **逻辑控制流（*Logical control flow*）**
    - **每个程序好像都有一个独占使用的CPU**
    - **内核提供了上下文切换（context switching）机制实现进程间的切换**
  - **私有地址空间（*Private address space*）**
    - **每个程序好像都有一个独占使用的主存.**
    - **内核提供了虚拟存储机制**



多进程并发: 给用户的“幻觉”



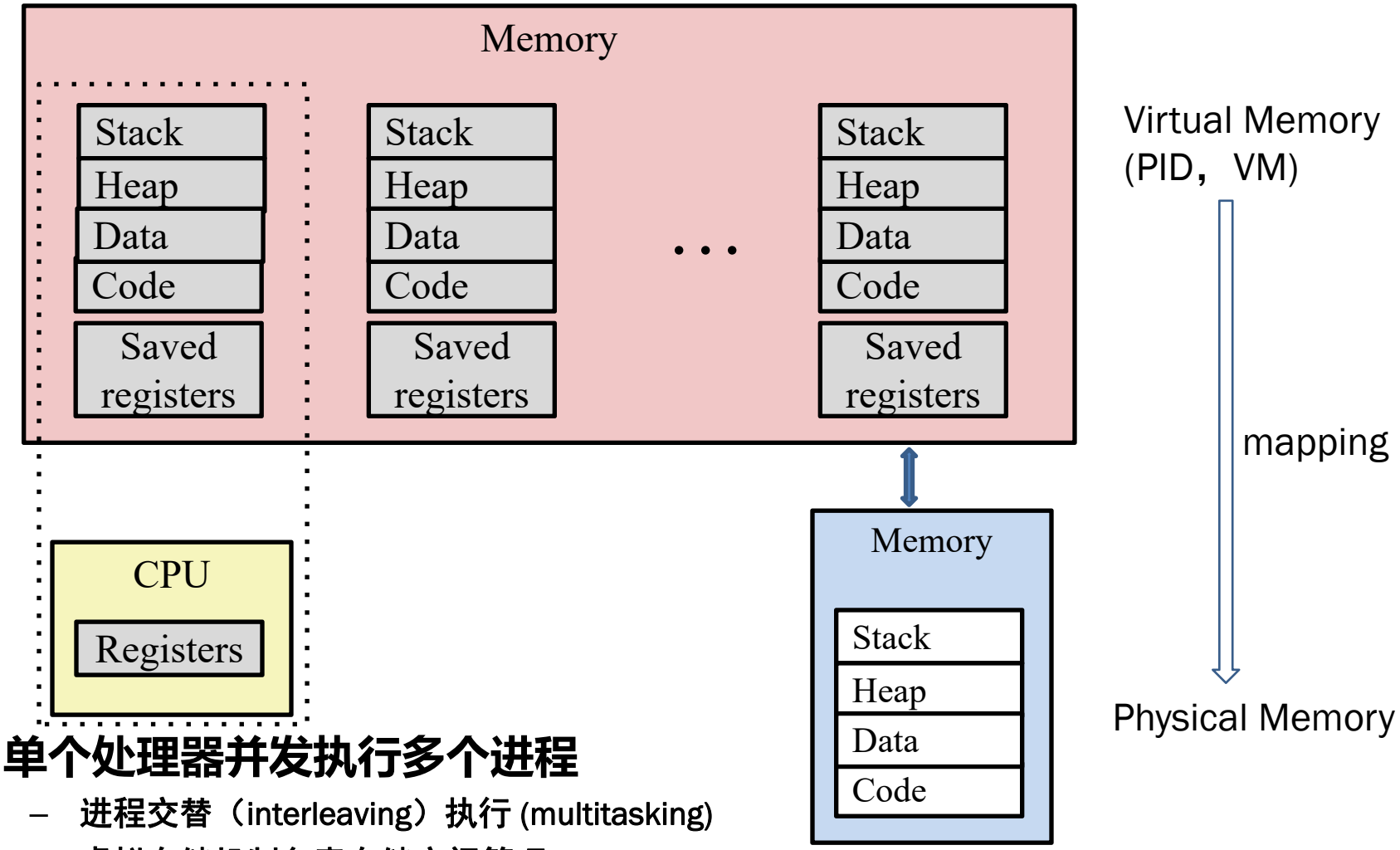


- **Computer 同时运行多个进程，包括：**
  - 一个或多个用户的应用
    - Web browsers, email clients, editors, ...
  - 后台任务
    - Monitoring network & I/O devices

Multiprocessing Example

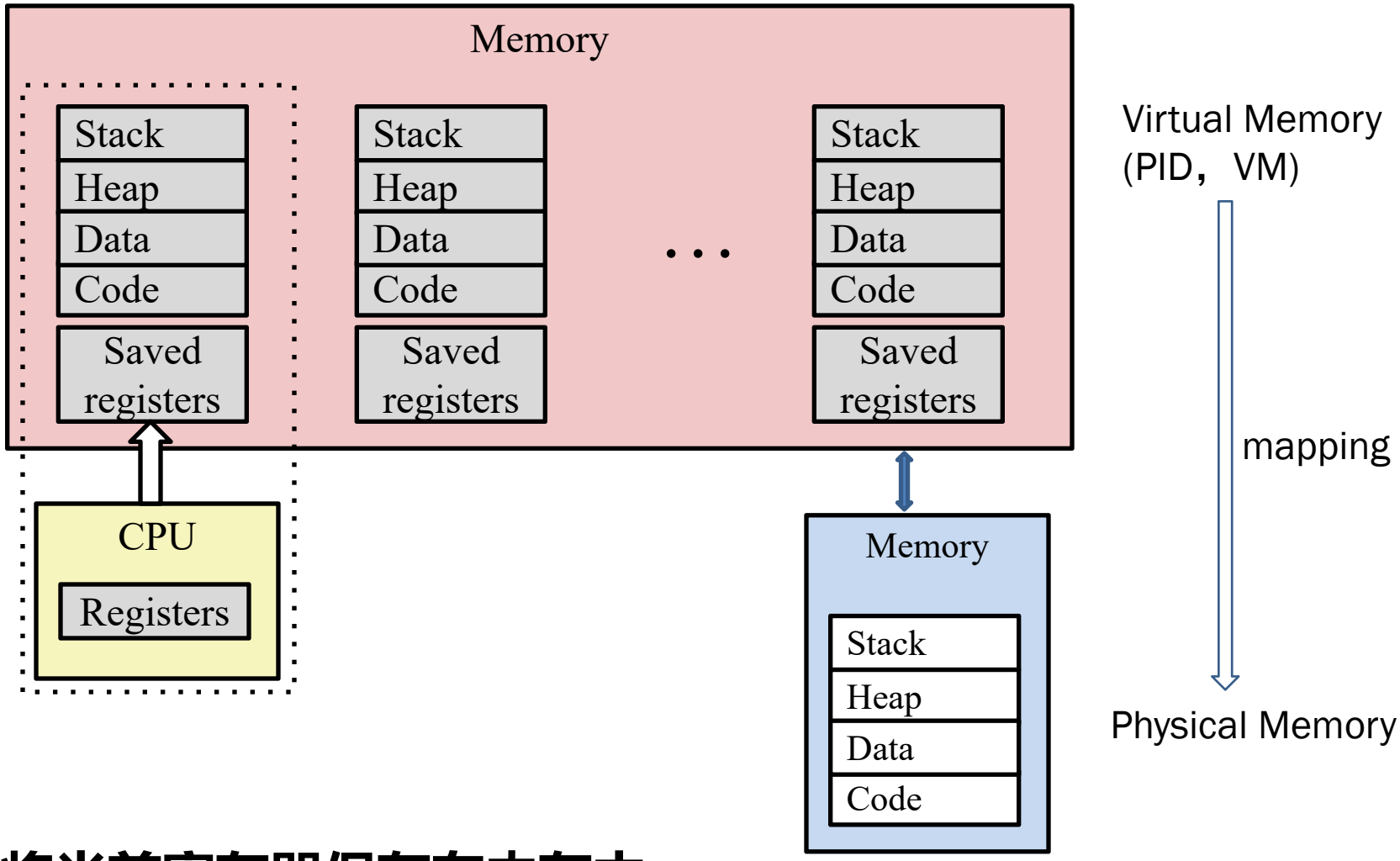
- **运行在Mac上的部分程序（进程）**
  - System has 123 processes, 5 of which are active
  - Identified by Process ID (PID)

多进程并发: 实际情形（单核）



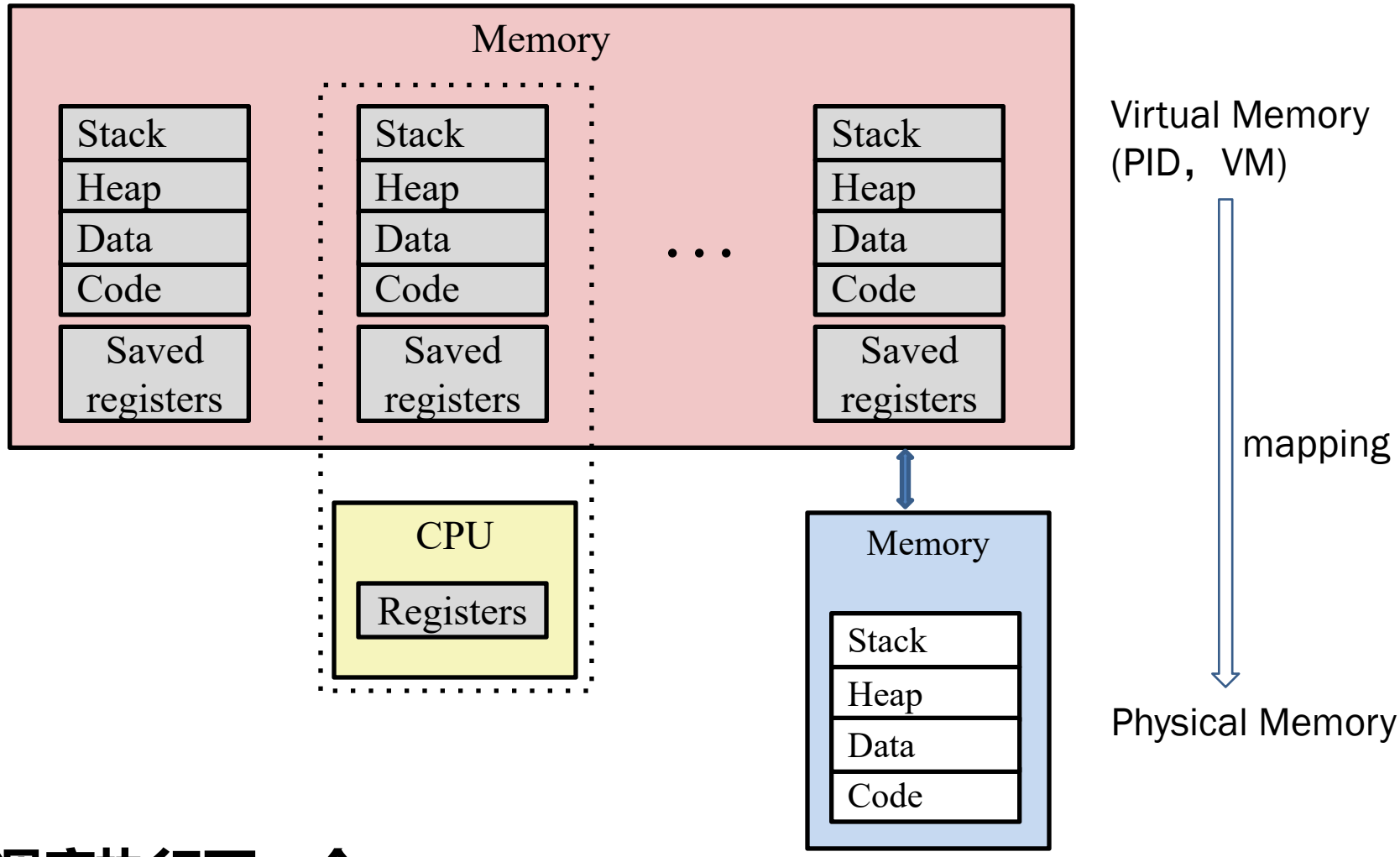
- **单个处理器并发执行多个进程**
  - 进程交替（interleaving）执行 (multitasking)
  - 虚拟存储机制负责存储空间管理
  - 处于非运行状态的”现场“(Register values)保存在存储器中

多进程并发: 实际情形（单核）



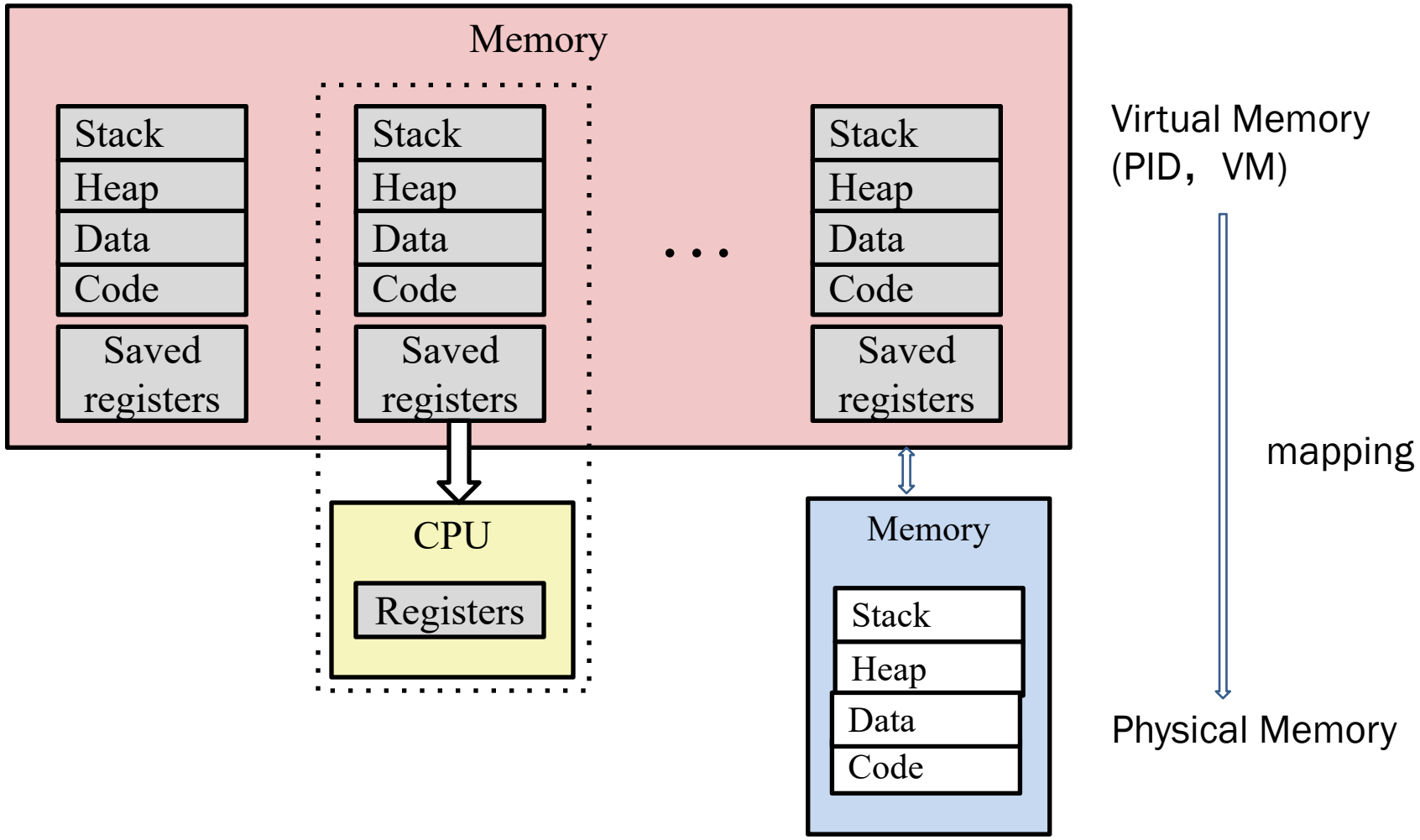
- **将当前寄存器保存在内存中**

多进程并发: 实际情形（单核）



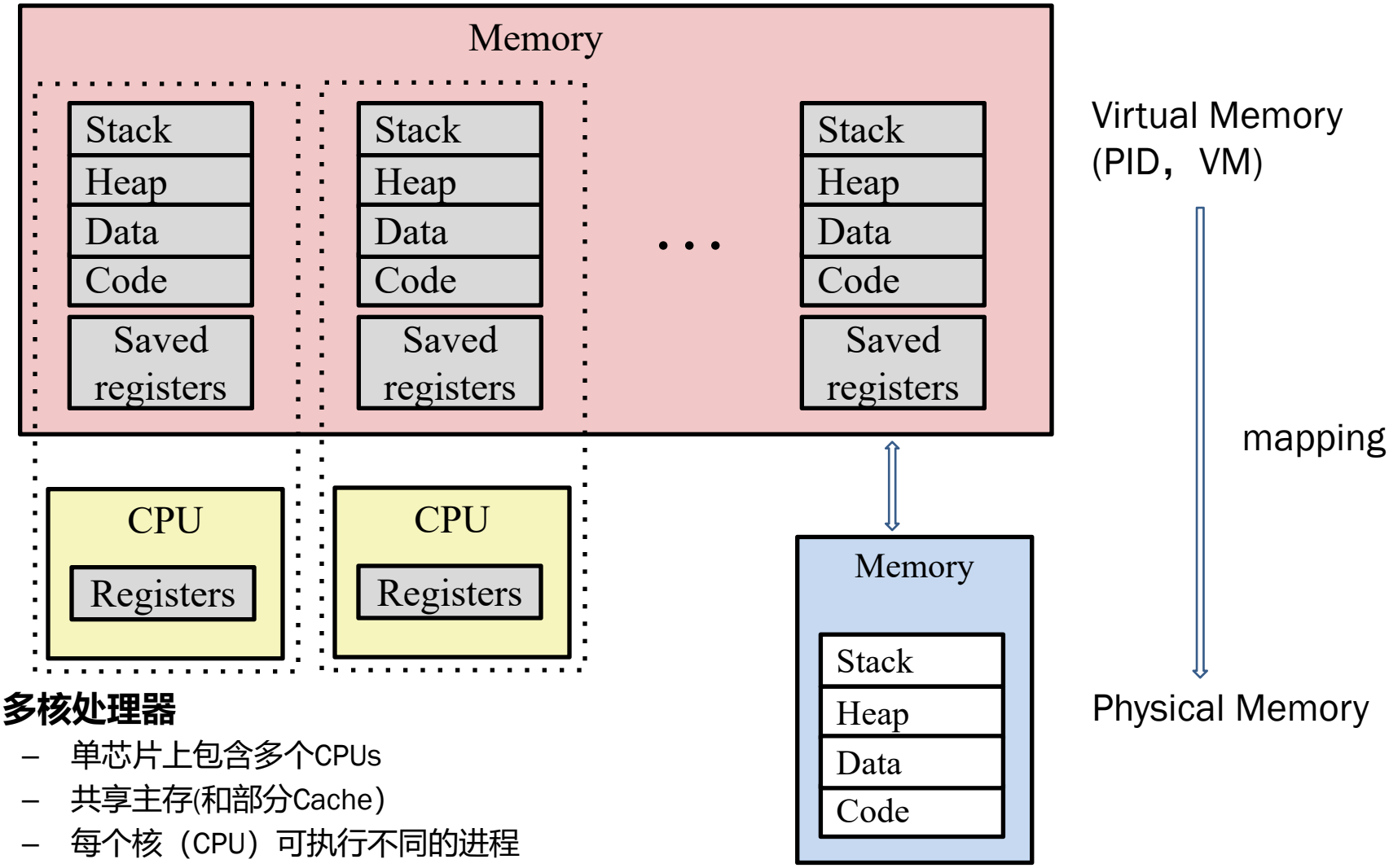
- **调度执行下一个process**

多进程并发: 实际情形（单核）



- **加载保存的寄存器 并 切换地址空间** (context switch)

多进程并发: 实际情形（多核）



- **多核处理器**
  - 单芯片上包含多个CPUs
  - 共享主存(和部分Cache)
  - 每个核（CPU）可执行不同的进程
    - 由内核调度进程到CPU上运行

4.5 虚拟存储

虚拟存储回顾 → TLB

如何加速虚拟地址到物理地址的转换？

TLB （Translation look-aside Buffer）

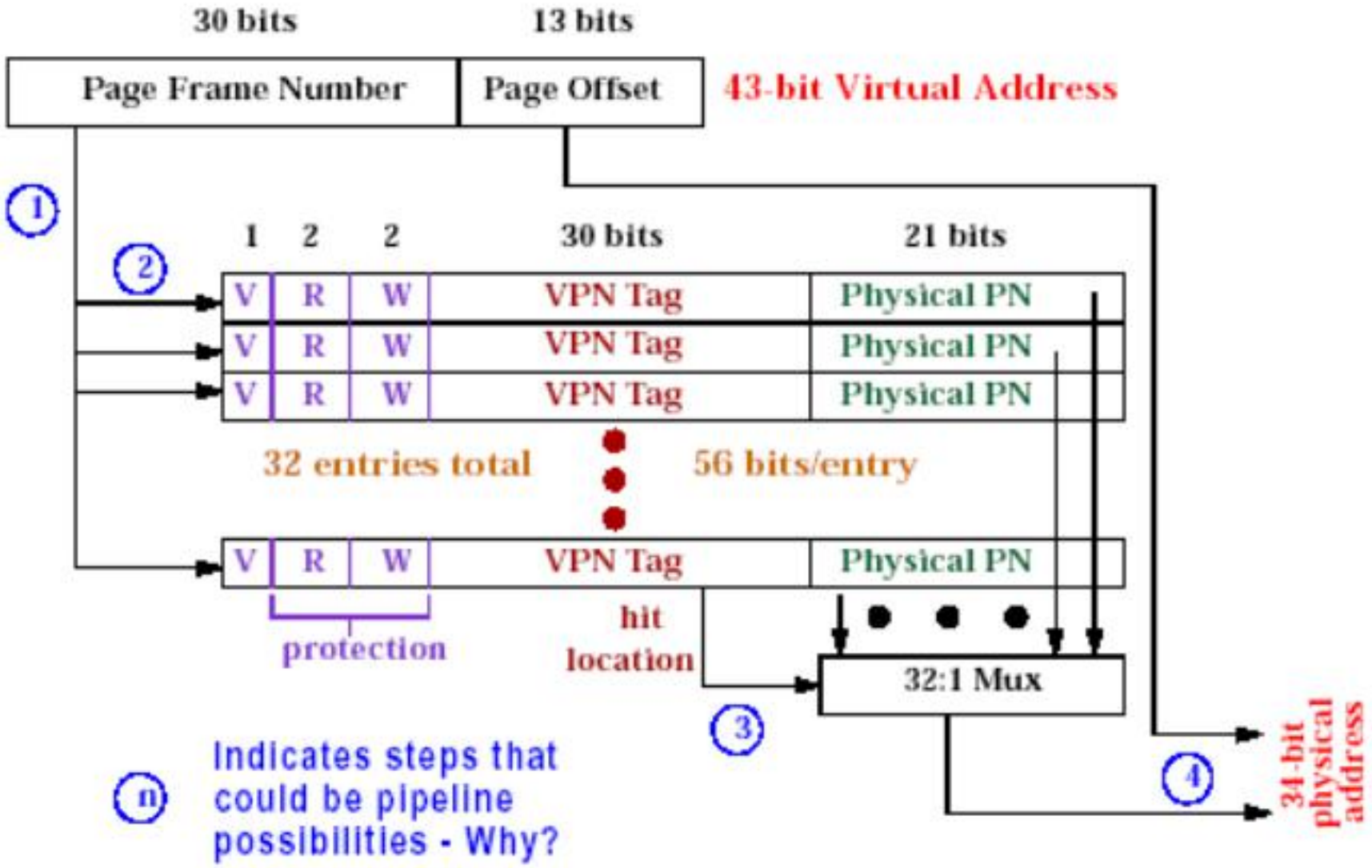
- **页表一般很大，存放在主存中**
  - 导致每次访存可能要两次访问主存，一次读取页表项，一次读写数据
  - 解决办法：采用 TLB
- **TLB**
  - **存放近期经常使用的页表项，是整个页表的部分内容的副本**
  - 基本信息：

    VPN##PPN##Protection Field##use bit ## dirty bit
  - OS修改页表项时，需要刷新TLB，或保证TLB中没有该页表项的副本
  - TLB必须在片内
    - 速度至关重要
    - TLB过小，意义不大
    - TLB过大，代价较高
    - 相联度较高（容量小）

TLB的典型参数

- **block size - same as a page table entry - 1 or 2 words**
- **hit time - 1 cycle**
- **miss penalty - 10 to 30 cycles**
- **miss rate - 0.1% to 2%**
- **TLB size - 32 B to 8 KB**

举例：Alpha 21064的TLB



Summary of Virtual Memory and Caches

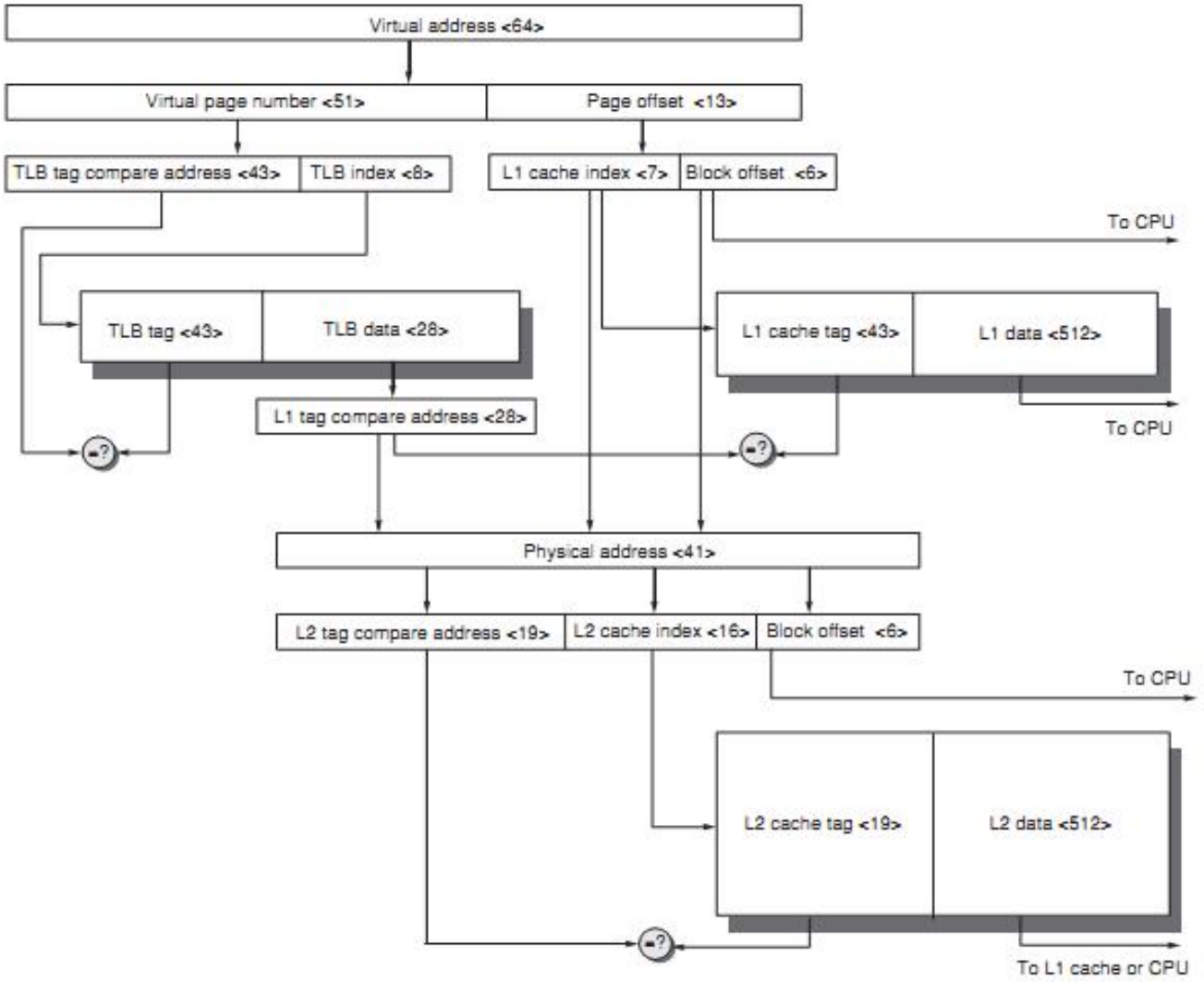


Figure C.24 The overall picture of a hypothetical memory hierarchy going from virtual address to L2 cache access. The page size is 8 KB. The TLB is direct mapped with 256 entries. The L1 cache is a direct-mapped 8 KB, and the L2 cache is a direct-mapped 4 MB. Both use 64-byte blocks. The virtual address is 64 bits and the physical address is 41 bits. The primary difference between this simple figure and a real cache is replication of pieces of this figure.

Acknowledgements

- **These slides contain material developed and copyright by:**
  - John Kubiatowicz (UCB)
  - Krste Asanovic (UCB)
  - John Hennessy (Standford)and David Patterson (UCB)
  - Chenxi Zhang (Tongji)
  - Muhamed Mudawar (KFUPM)
- **UCB material derived from course CS152、CS252、CS61C**
- **KFUPM material derived from course COE501、COE502**