



动态存储器

2022年秋

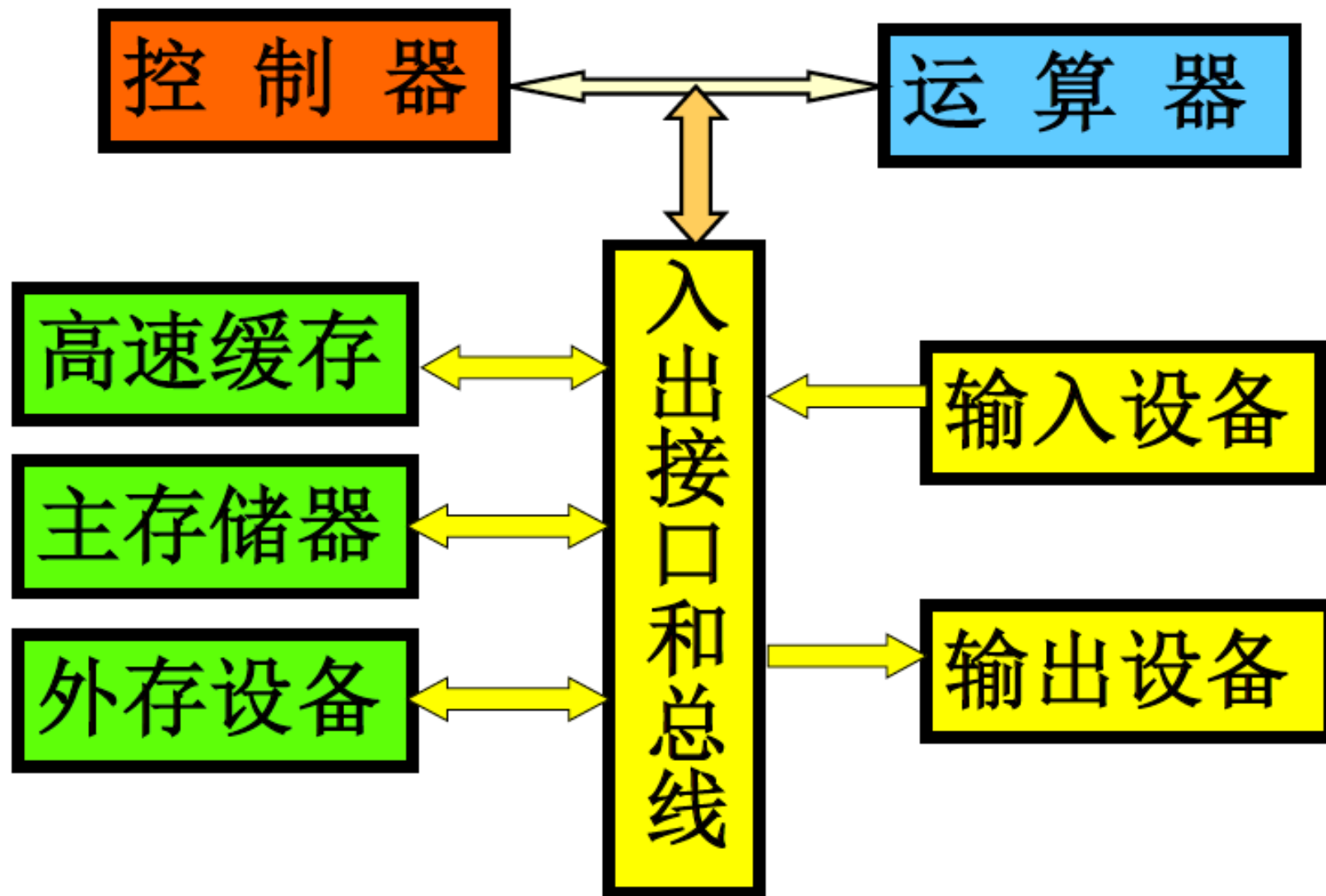
本单元内容提要

- 第一讲层次存储器系统概述及动态存储器
- 第二讲静态存储器及高速缓冲存储器
- 第三讲高速缓冲存储器的组成与运行原理
- 第四讲虚拟存储器的运行原理
- 第五讲磁表面存储设备的存储原理与组成
- 第六讲**RISC-V**系统异常处理和响应，虚拟内存

本讲概要

- 存储器系统功能
- 存储器系统的设计目标
- 需要解决的问题
- 层次存储器系统
- 动态存储器的组成与原理

计算机硬件系统



存储器地位和作用

- 存储程序使计算机走向通用。
- 计算机中用来存放程序和数据的部位，是Von Neumann结构计算机的重要组成，是计算机的中心。
- 程序和数据的特点
 - 源程序、汇编程序、机器语言程序
 - 各种类型的数据
 - 共同点：二进制数据

对存储介质的基本要求

- 能够有两个稳定状态来表示二进制中的“0”和“1”
- 容易识别
- 两个状态能方便地进行转换
- 几种常用的存储方式
 - 磁颗粒、半导体(电平/电容)、光

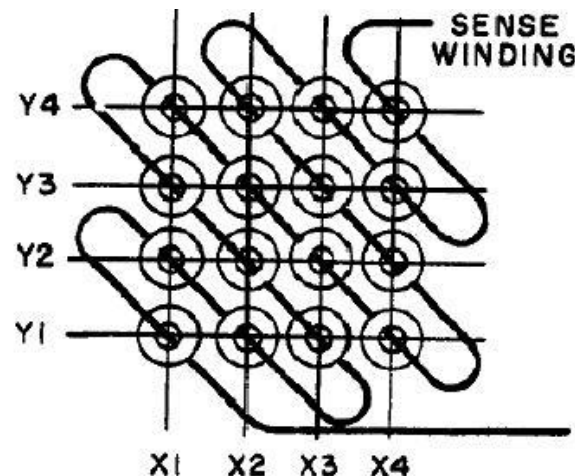
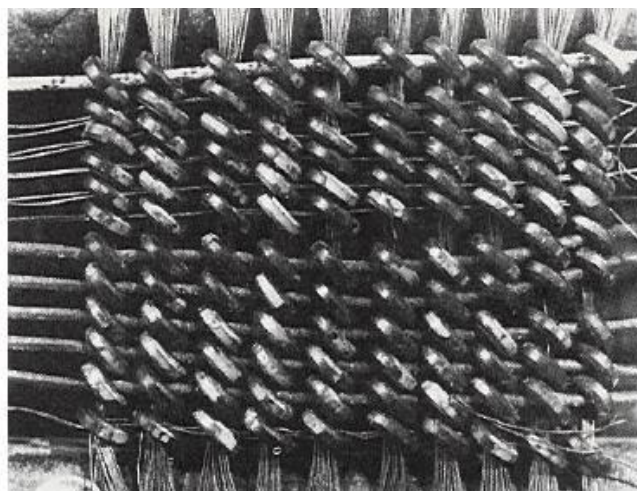
早期存储器

- 水银延迟线存储器
- EDSAC, 1949
- Maurice Wilkes
- 1967年Turing奖
- 存储原理
 - 水波



磁芯存储器

- 圆柱型陶瓷上涂磁粉
- 手工穿线，水手结
- 消磁后重写



半导体存储器

□ 存储原理

- MOS管寄生电容
- 触发器

□ 访问机制

- 随机访问

□ 分类

- ROM、RAM
- SRAM、DRAM

按访问方式分类

- 随机访问存储器（RAM）
 - 访问时间与存放位置无关
 - 半导体存储器
- 顺序访问存储器（SAM）
 - 按照存储位置依次访问
 - 磁带存储器
- 直接访问存储器（DAM）
 - 随机+顺序
 - 磁盘存储器
- 关联访问存储器（CAM）
 - 根据内容访问
 - Cache和TLB

存储器系统设计目标

□ 尽可能快的存取速度

- 应能基本满足CPU对数据的访问要求

□ 尽可能大的存储空间

- 可以满足程序对存储空间的要求

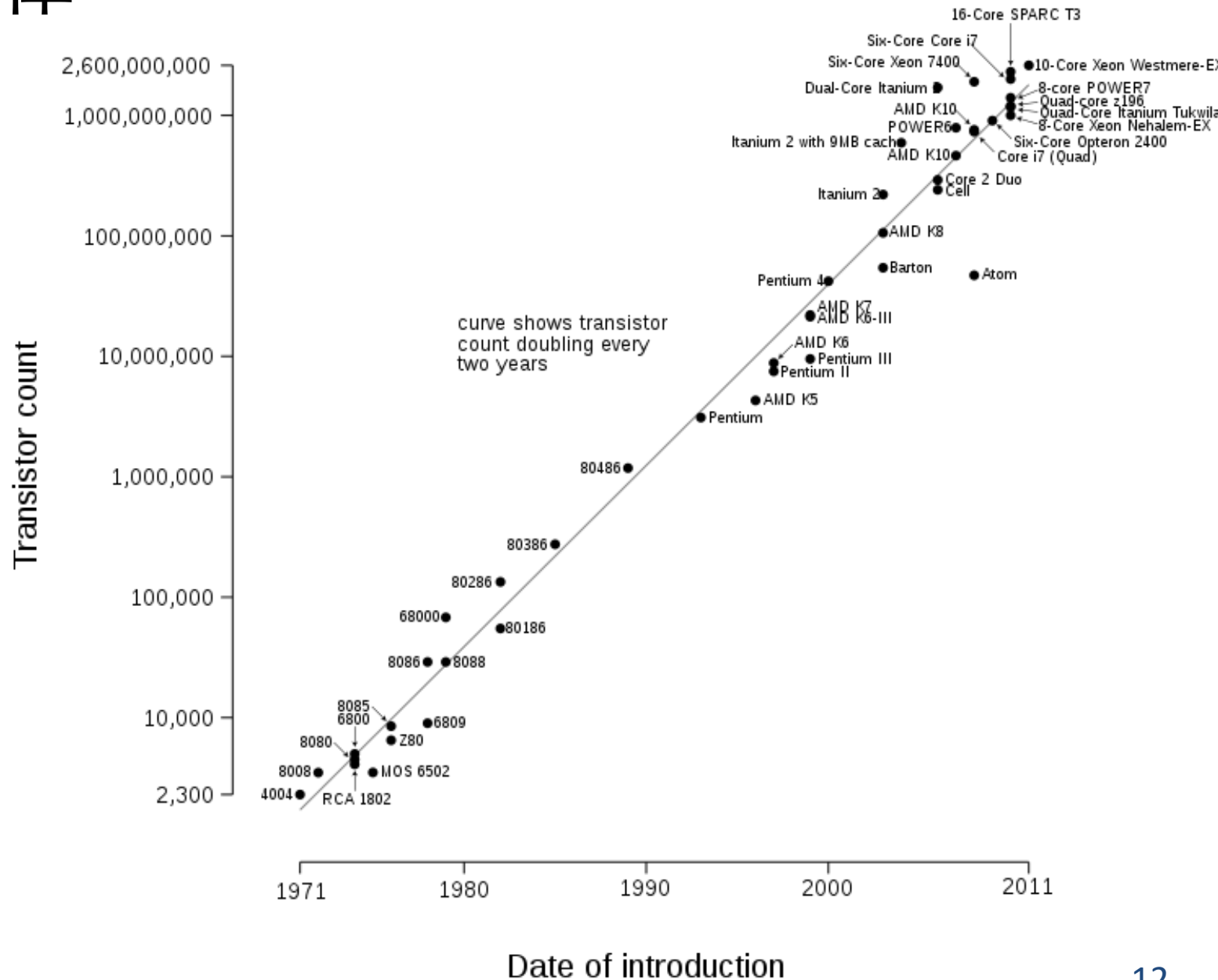
□ 尽可能低的单位成本（价格/位）

- 用户能够承受的范围内

□ 较高的可靠性

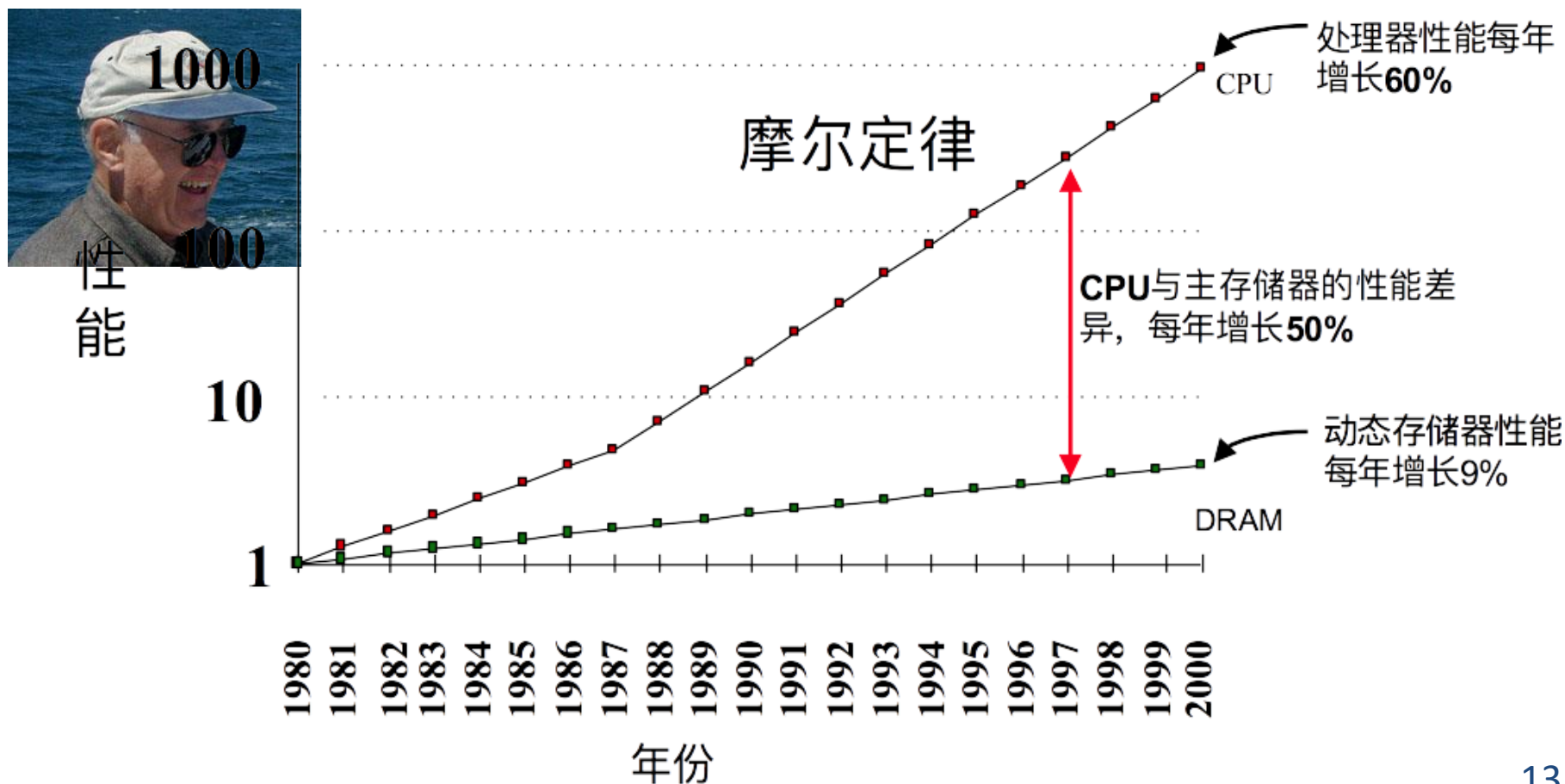
摩尔定律

Microprocessor Transistor Counts 1971-2011 & Moore's Law



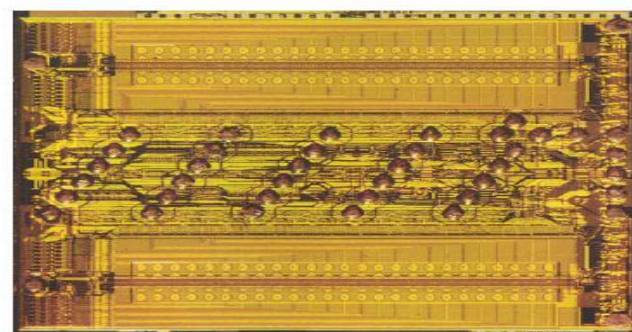
Moore定律

- 1965年，Intel公司创始人之一Gordon Moore提出
- 芯片上集成的晶体管数量每18个月翻一番



摩尔定律

年代	容量	价格 (\$/MB)	总访问时间 (新行 / 列)	列访问时间 (现访问行)
1980	64 Kbit	1500	250 ns	150 ns
1983	256 Kbit	500	185 ns	100 ns
1985	1 Mbit	200	135 ns	40 ns
1989	4 Mbit	50	110 ns	40 ns
1992	16 Mbit	15	90 ns	30 ns
1996	64 Mbit	10	60 ns	20 ns
1998	128 Mbit	4	60 ns	10 ns
2000	256 Mbit	1	55 ns	7 ns
2004	512 Mbit	0.25	50 ns	5 ns
2007	1 Gbit	0.05	40 ns	1.25 ns



存储器对性能的影响

□ 假定某台计算机的处理器工作在：

- 主频= 1GHz (机器周期为1 ns)
- $CPI = 1.1$
- 50% 算逻指令, 30% 存取指令, 20% 转移指令

□ 再假定其中10% 的存取指令会发生数据缺失，需要50个周期的延迟。

- $CPI = \text{理想CPI} + \text{每条指令的平均延迟} = 1.1 + (0.30 \times 0.10 \times 50) = 1.1 \text{ cycle} + 1.5 \text{ cycle} = 2.6 \text{ CPI!}$
- 也就是说，处理器58 %的时间花在等待存储器给出数据上面！

□ 每1% 的指令的数据缺失将给CPI附加0.5个周期！

存储器设计目标

□ 目标

- 大容量、高速度、低成本、高可靠性

□ 目前现实

- 大容量存储器速度慢
- 快速存储器容量小

□ 如何实现我们的目标呢？

- 层次存储器系统

问题

□ CPU clock rates $\sim 0.33\text{ns} - 2\text{ns}$ (3GHz-500MHz)

Memory technology	Access time in nanosecs (ns)	Access time in cycles	\$ per GB in 2012	Capacity
SRAM (on chip)	0.5-2.5 ns	1-3 cycles	\$4k	256 KB
SRAM (off chip)	1.5-30 ns	5-15 cycles	\$4k	32 MB
DRAM	50-70 ns	150-200 cycles	\$10-\$20	8 GB
SSD (Flash)	5k-50k ns	Tens of thousands	\$0.75-\$1	512 GB
Disk	5M-20M ns	Millions	\$0.05-\$0.1	4 TB

最新的一些磁盘性能数据

Comparing Drive Types

	HDD	Flash-based SSD
Durability and Noise	Loud and susceptible to shock	No moving parts!
Access Time	~ 12 ms ≈ 30M clock cycles	~ 0.1 ms ≈ 250K clock cycles
Relative Power	1	1/3
Cost	~ \$0.035 / GB	~ \$0.35 / GB
Capacity	500GB - 12TB	128GB - 2TB
Other Problems	Fragmentation	Limited Writes
Lifespan	5-10 years	Avg Failure rate 6 years

层次存储器系统

□ 高速度

- 静态存储器速度高
- 设置较小容量的高速缓冲存储器

□ 大容量

- 动态存储器价格适中，速度适中
- 可作为主存储器

□ 低成本

- 磁盘存储器价格低廉
- 作为辅助存储器，暂存CPU访问频率不高的数据和程序
- 作为虚拟存储器的载体

程序运行的局部性原理

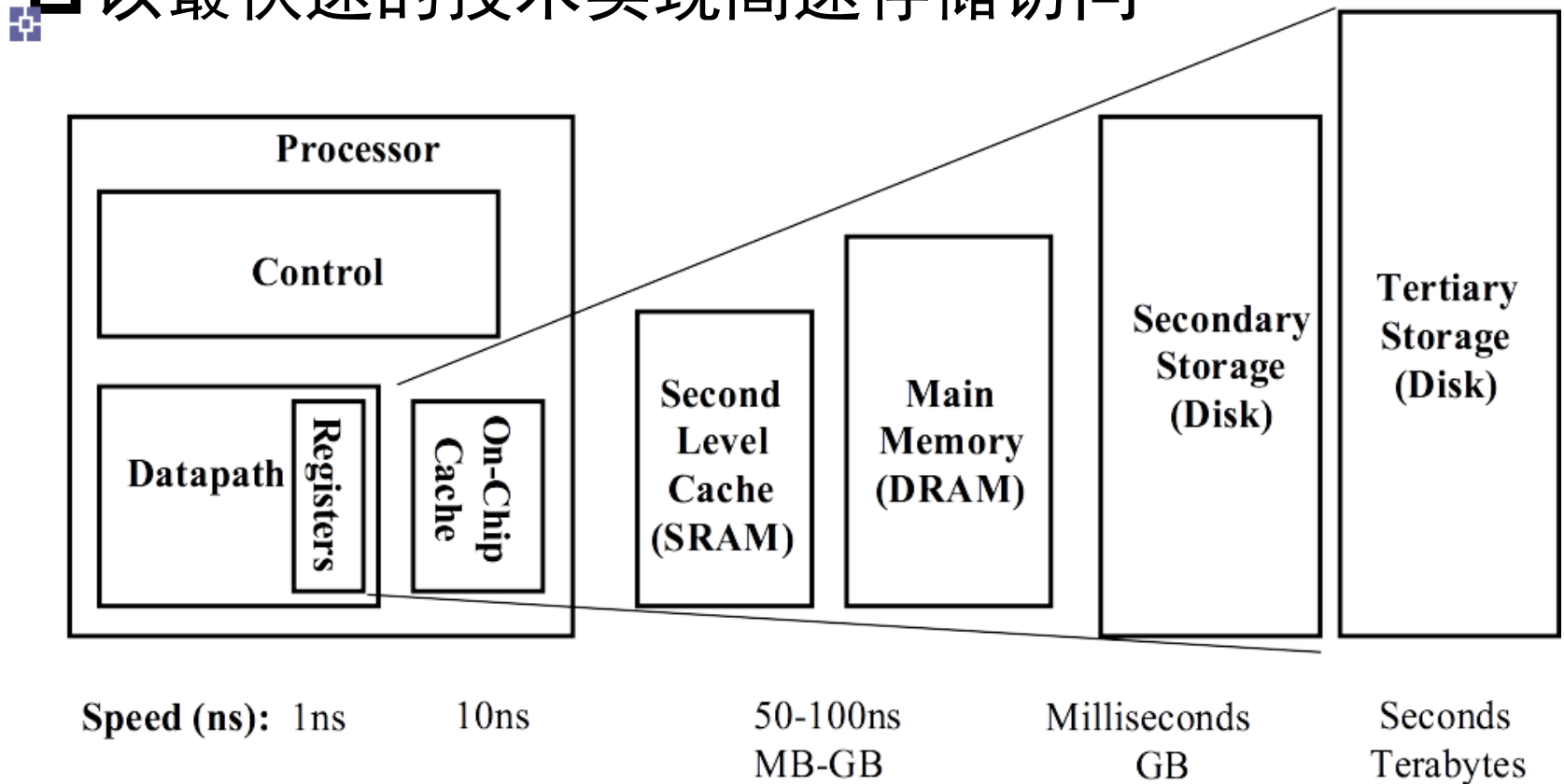
- 程序运行时的局部性原理表现在：
- 在一小段**时间**内，最近被访问过的程序和数据很可能再次被访问
- 在**空间**上这些被访问的程序和数据往往集中在一小片存储区
- 在访问**顺序**上，指令顺序执行比转移执行的可能性大(大约**5:1**)
- 合理地把程序和数据分配在不同存储介质中

层次之间应满足的原则

- (1). 一致性原则：处在不同层次存储器中的同一个信息应保持相同的值。
- (2). 包含性原则：处在内层的信息一定被包含在其外层的存储器中，反之则不成立,即内层存储器中的全部信息，是其相邻外层存储器中一部分信息的复制品。

层次存储器系统

- 利用程序的局部性原理:
- 以最低廉的价格提供尽可能大的存储空间
- 以最快速的技术实现高速存储访问



现代计算机存储器系统

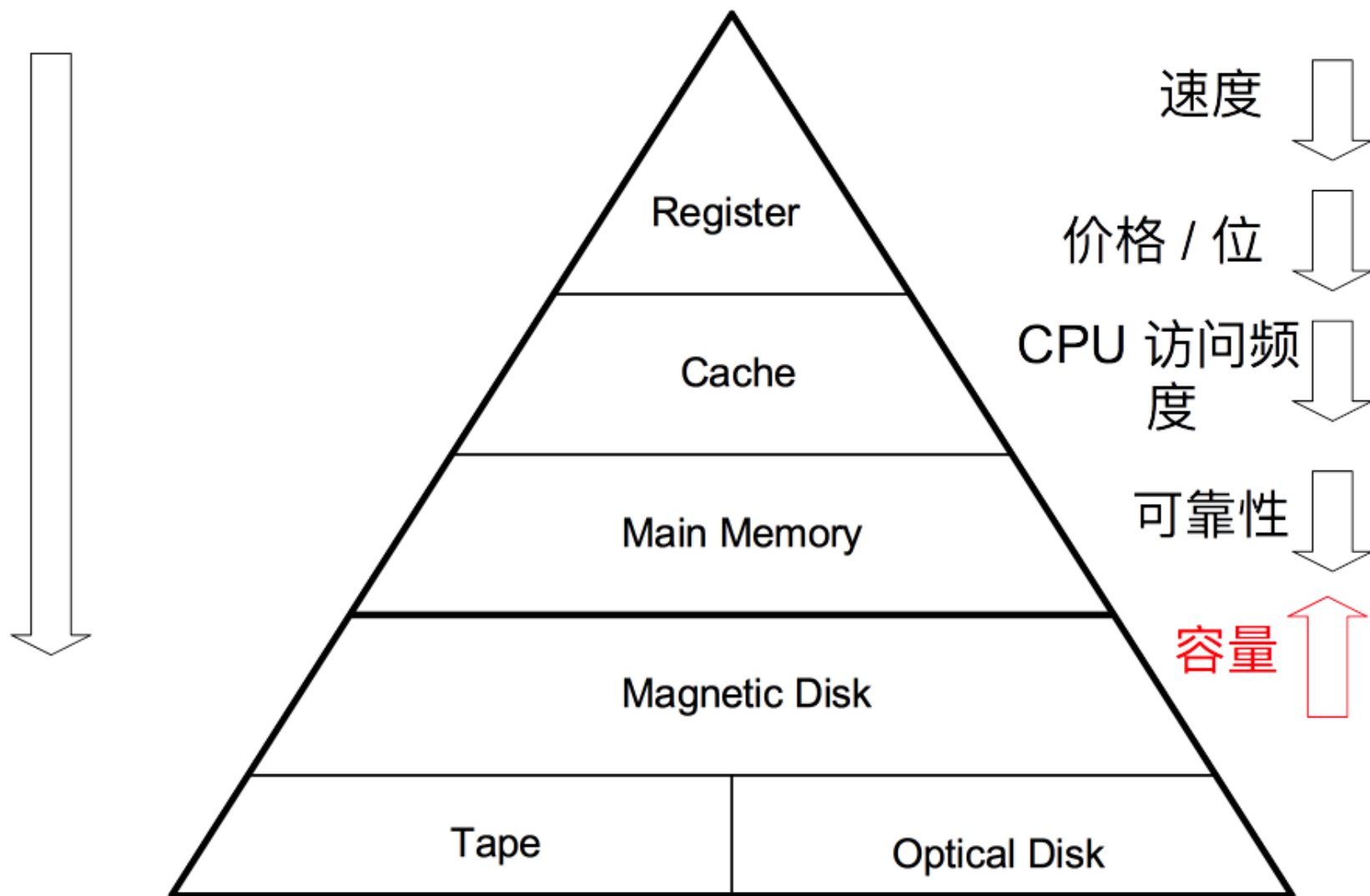
□ 主存储器

- 寄存器Register
- 高速缓存Cache
- 主存储器Main Memory

□ 辅助存储器

- 磁盘Disk
- 磁带Tape
- 光盘Compact Disc

不同类型存储器比较



并行技术

□主存的一体多字

- 一个读写体，每次多个字

□单字多体

- 多个读写体，交叉编址

□多端口存储器

主存储器的作用和连接

- 存储正处在运行中的程序和数据(或一部分)的部件，通过地址数据控制三类总线与**CPU**、与其它部件连通



地址总线

- 地址总线用于选择主存储器的一个存储单元（字或字节），其位数决定了能够访问的存储单元的最大数目，称为最大可寻址空间。例如，当按字节寻址时，20位的地址可以访问1MB的存储空间，32位的地址可以访问4GB的存储空间。

数据总线

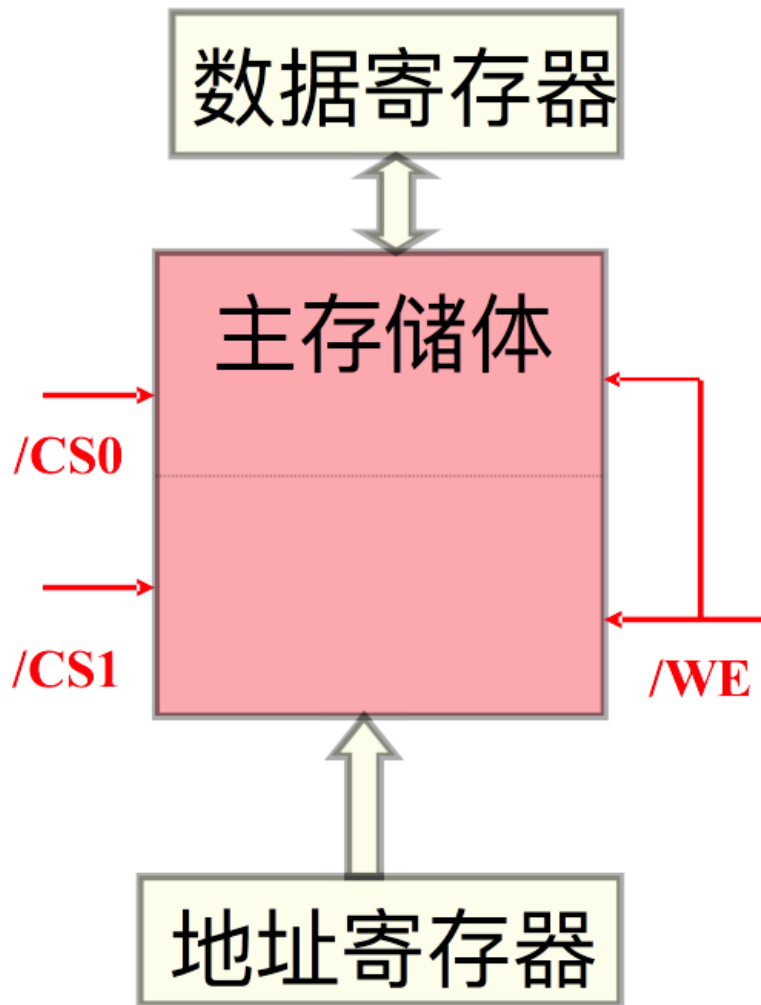
□ 数据总线用于在计算机各功能部件之间传送数据，数据总线的位数（总线的宽度）与总线时钟频率的乘积，与该总线所支持的最高数据吞吐（输入/输出）能力成正比。

Names	Memory clock	I/O bus clock	Transfer rate	Theoretical bandwidth
DDR-200, PC-1600	100 MHz	100 MHz	200 MT/s	1.6 GB/s
DDR-400, PC-3200	200 MHz	200 MHz	400 MT/s	3.2 GB/s
DDR2-800, PC2-6400	200 MHz	400 MHz	800 MT/s	6.4 GB/s
DDR3-1600, PC3-12800	200 MHz	800 MHz	1600 MT/s	12.8 GB/s
DDR4-2400, PC4-19200	300 MHz	1200 MHz	2400 MT/s	19.2 GB/s
DDR4-3200, PC4-25600	400 MHz	1600 MHz	3200 MT/s	25.6 GB/s
DDR5-4800, PC5-38400	300 MHz	2400 MHz	4800 MT/s	38.4 GB/s
DDR5-6400, PC5-51200	400 MHz	3200 MHz	6400 MT/s	51.2 GB/s

控制总线

- 控制总线用于指明总线的工作周期类型和本次入/出完成的时刻。总线的工作周期可以包括主存储器读周期、主存储器写周期、I/O设备读周期、I/O设备写周期，即用不同的总线周期来区分要用哪个部件（主存或I/O设备）和操作的性质（读或写）；还有直接存储器访问（DMA）总线周期等。

主存储器的读写过程



□读过程:

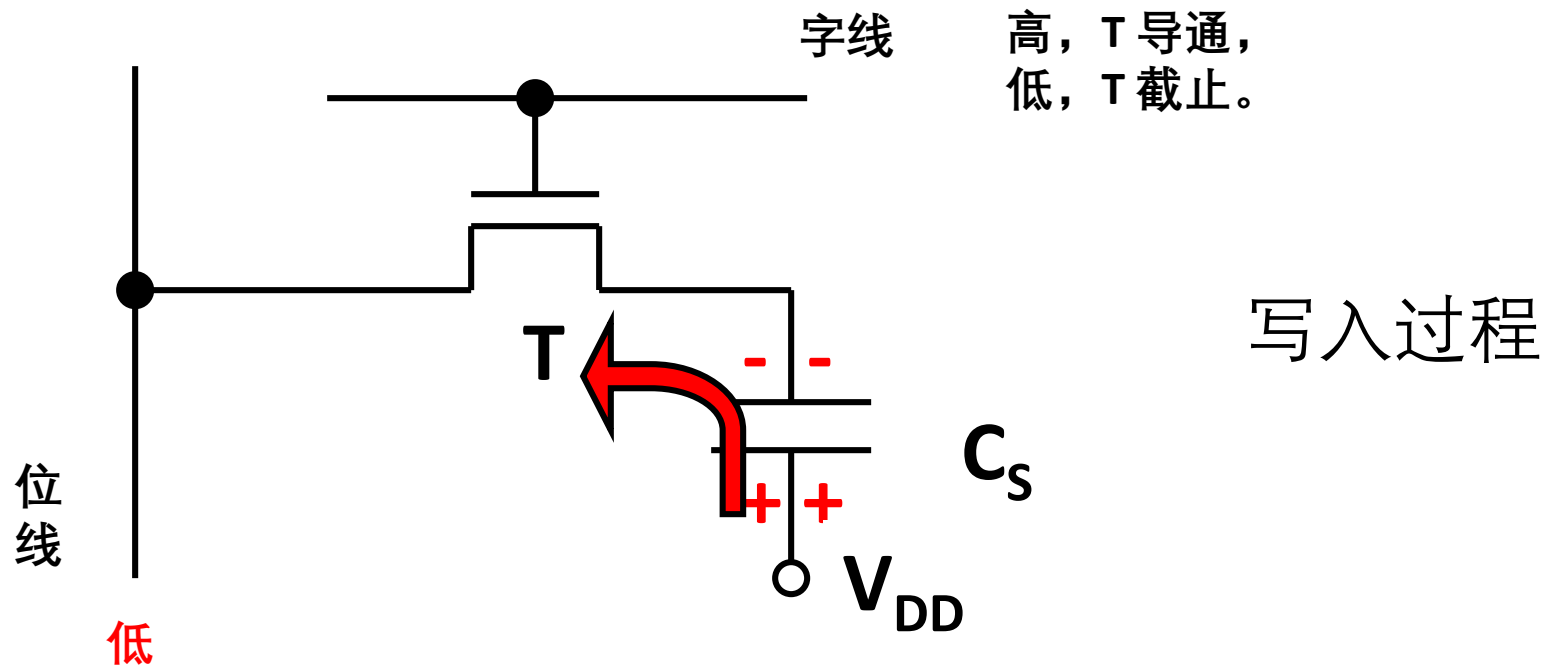
- 给出地址
- 给出片选与读命令
- 保存读出内容

□写过程:

- 给出地址
- 给出片选与数据
- 给出写命令

动态存储器的存储原理

- 动态存储器，是用金属氧化物半导体（MOS）的单个MOS管来存储一个二进制位（bit）信息的。信息被存储在MOS管T的源极的寄生电容CS中，例如，用CS中存储有电荷表示1，无电荷表示0。

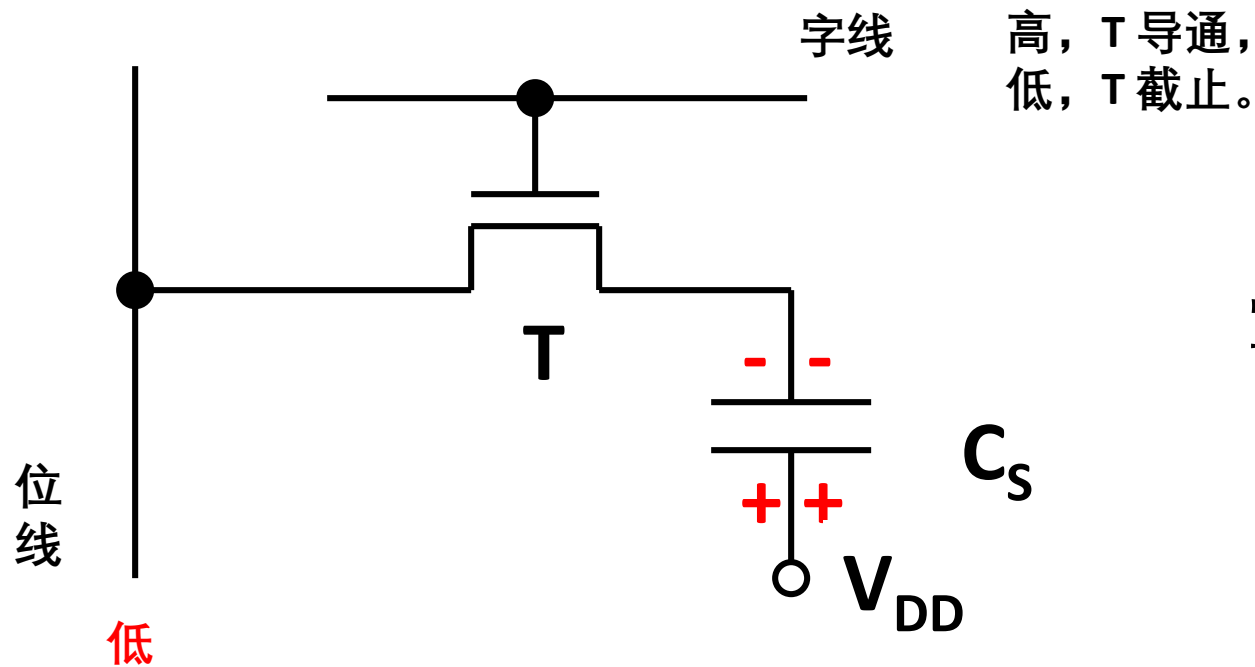


写 1：使位线为低电平，

若 C_s 上无电荷，则 V_{DD} 向 C_s 充电；

把 1 信号写入了电容 C_s 中。

若 C_s 上有电荷，则 C_s 的电荷不变，
保持原记忆的 1 信号不变。

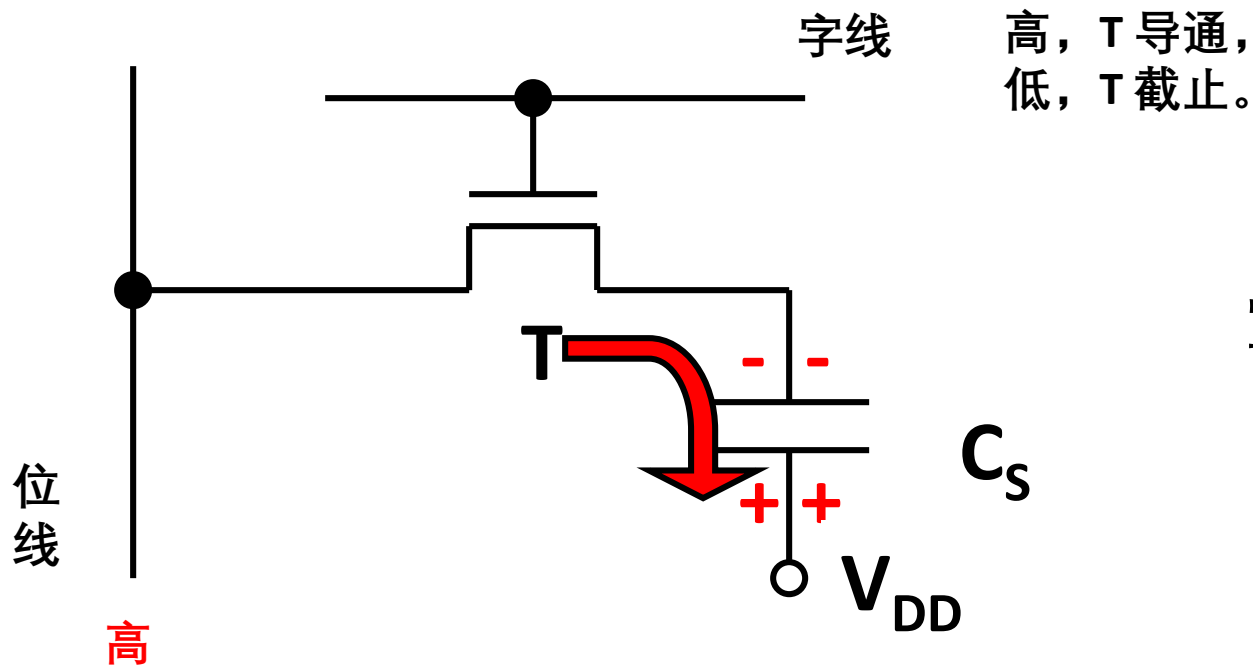


写 1：使位线为低电平，

若 C_s 上无电荷，则 V_{DD} 向 C_s 充电；

把 1 信号写入了电容 C_s 中。

若 C_s 上有电荷，则 C_s 的电荷不变，
保持原有的内容 1 不变；

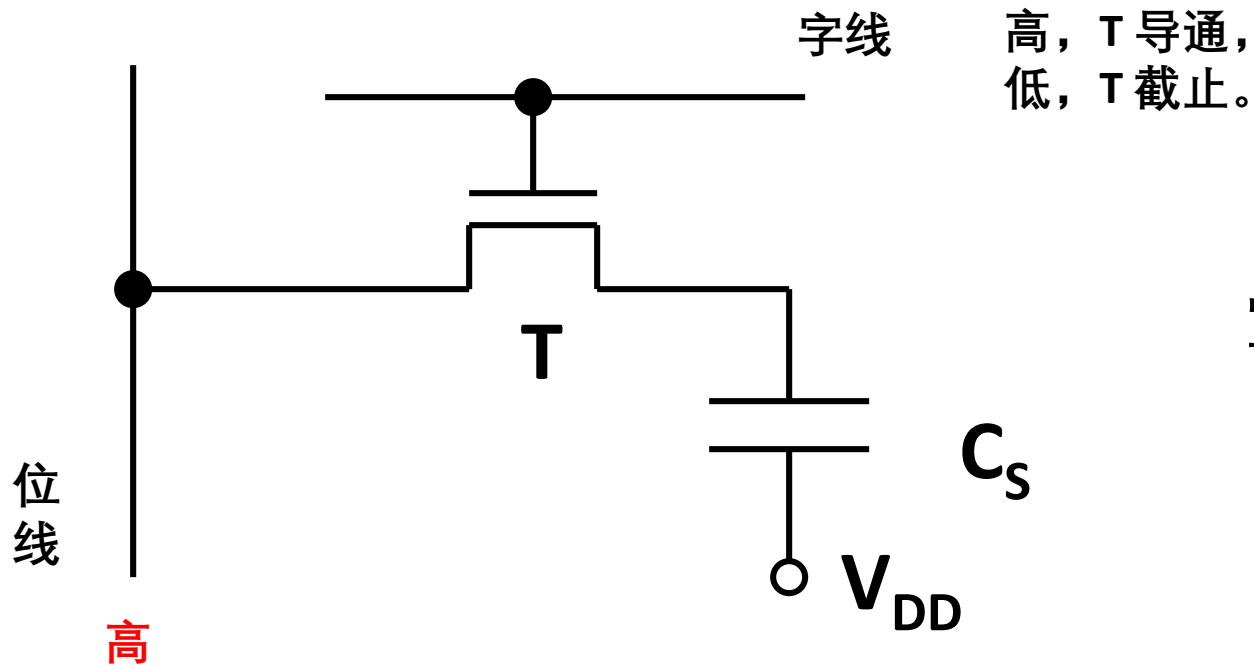


写 0：使位线为高电平，

若 C_s 上有电荷，则 C_s 通过 T 放电；

把 0 信号写入了电容 C_s 中。

若 C_s 上无电荷，则 C_s 无充放电动作，
保持原记忆的 0 信号不变。



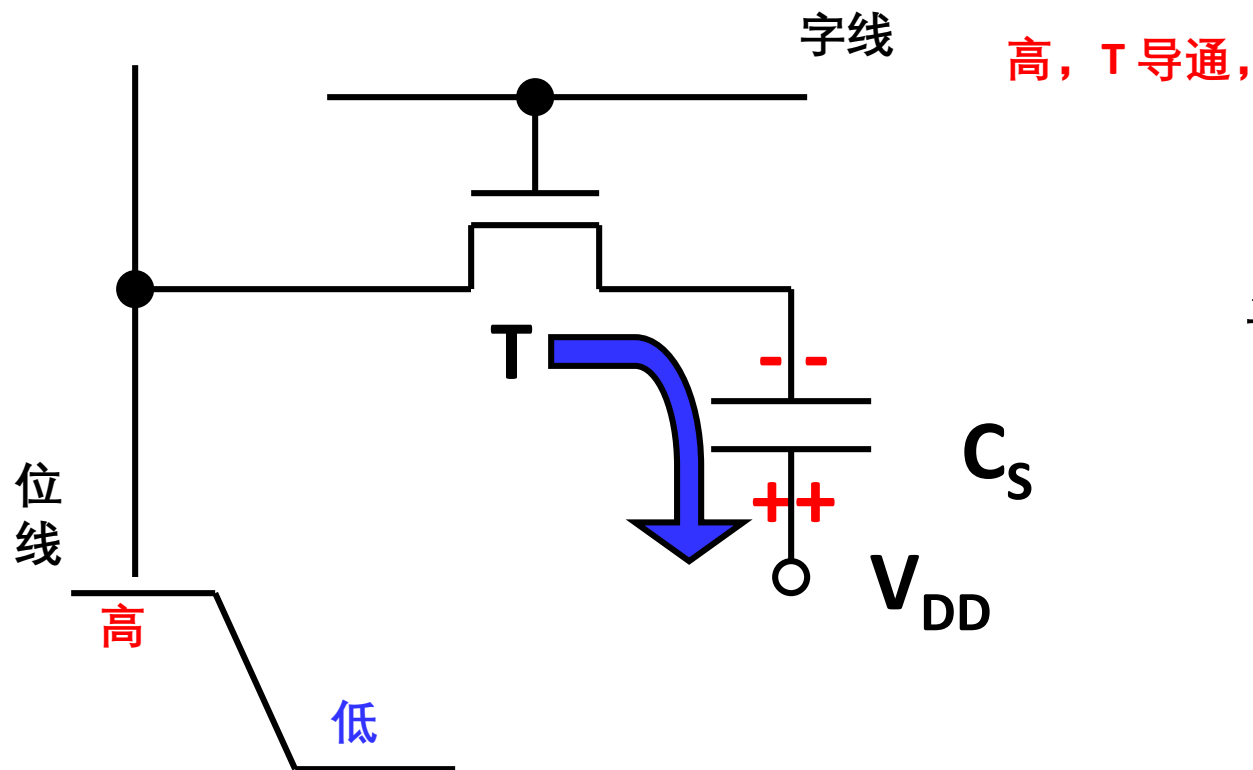
写入过程

写 0：使位线为高电平，

若 C_s 上有电荷，则 C_s 通过 T 放电；

把 0 信号写入了电容 C_s 中。

若 C_s 上无电荷，则 C_s 无充放电动作，
保持原记忆的 0 信号不变。



读操作：首先使位线充电至高电平，当字线来高电平后，T导通，

1. 若 C_s 上无电荷，则位线上无电位变化，读出为 0；
2. 若 C_s 上有电荷，则会放电，
并使位线电位由高变低，

接在位线上的读出放大器会感知这种变化，读出为 1。

动态存储器的工作特点

□ 破坏性读出

- 读出时被强制清零
- 预充电延迟

□ 需定期刷新

- 集中刷新
 - 停止读写，逐行刷新
- 分散刷新
 - 定时周期性刷新

□ 快速分页组织

破坏性读出：读操作后，被读单元的内容一定被清为零，必须把刚读出的内容立即写回去，通常称其为预充电延迟，它影响存储器的工作频率，在结束预充电前不能开始下一次读。

要定期刷新：在不进行读写操作时，DRAM 存储器的各单元处于断路状态，由于漏电的存在，保存在电容 C_s 上的电荷会慢慢地漏掉，为此必须定时予以补充，通常称其为刷新操作。刷新不是按字处理，而是每次刷新一行，即为连接在同一行上所有存储单元的电容补充一次能量。刷新有**两种常用方式**：

集中刷新，停止内存读写操作，逐行将所有各行刷新一遍；

分散刷新，每一次内存读写后，刷新一行，各行轮流进行。

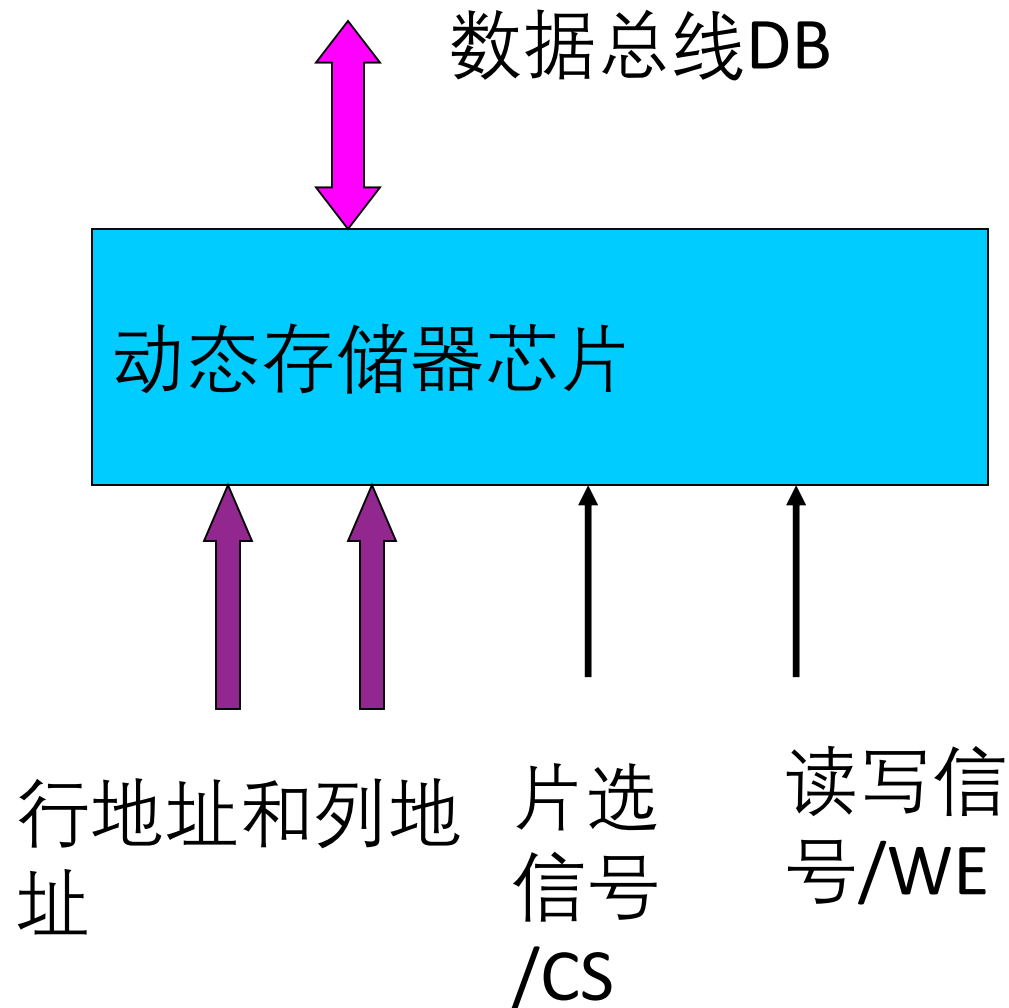
或在规定的期间内，如 2 ms，能轮流把所有各行刷新一遍。

快速分页组织的存储器：

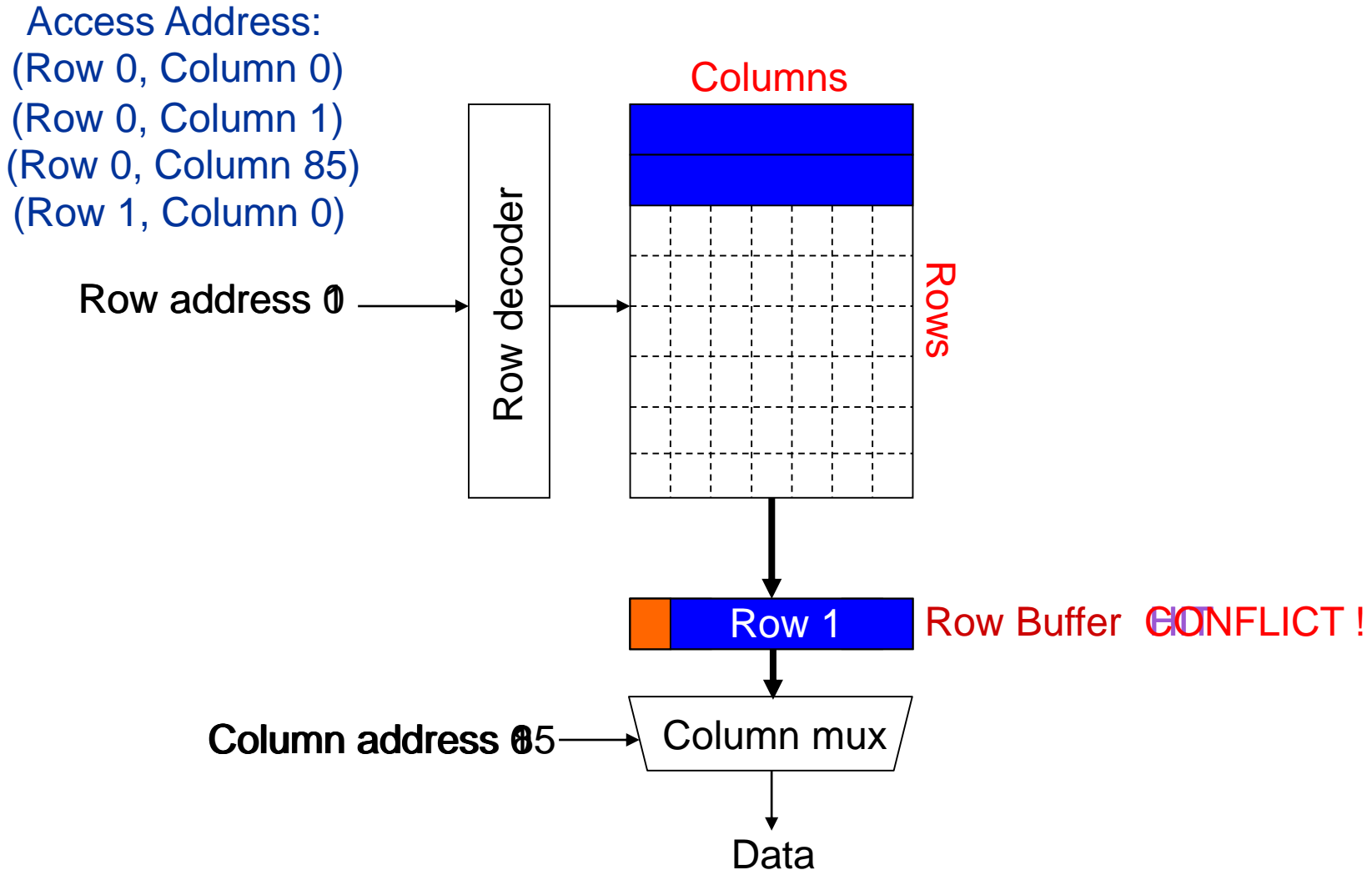
行、列地址要分两次给出，但连续地读写用到相同的行地址时，也可以在前一次将行地址锁存，之后仅送列地址，以节省送地址的时间，支持这种运行方式的被称为快速分页组织的存储器。

动态存储器读写过程

动态存储器集成度高，存储容量大，为节约管脚数，地址分为行地址和列地址

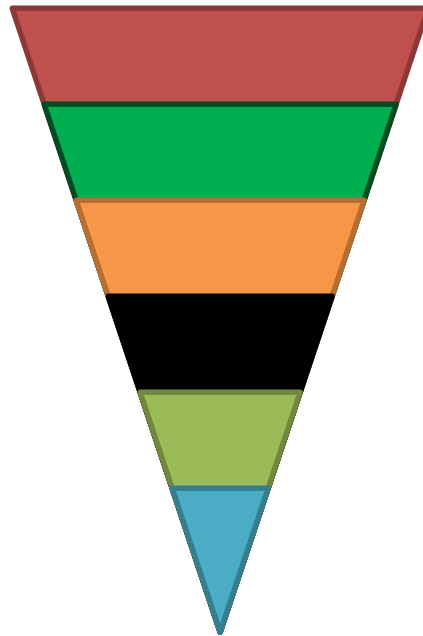


DRAM Bank Operation



DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column

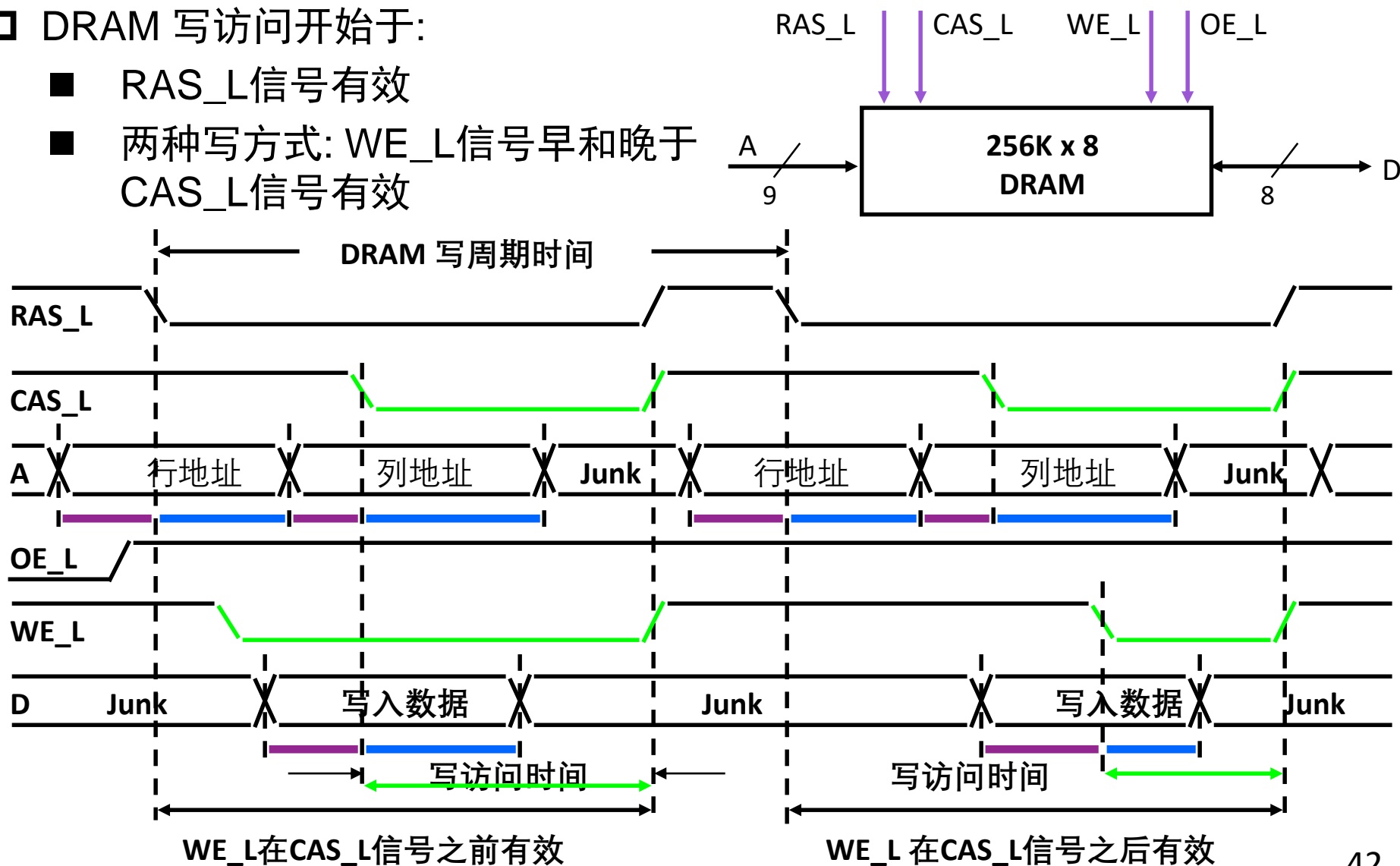


Top-Down View

DRAM 写时序

□ DRAM 写访问开始于:

- RAS_L信号有效
- 两种写方式: WE_L信号早和晚于CAS_L信号有效



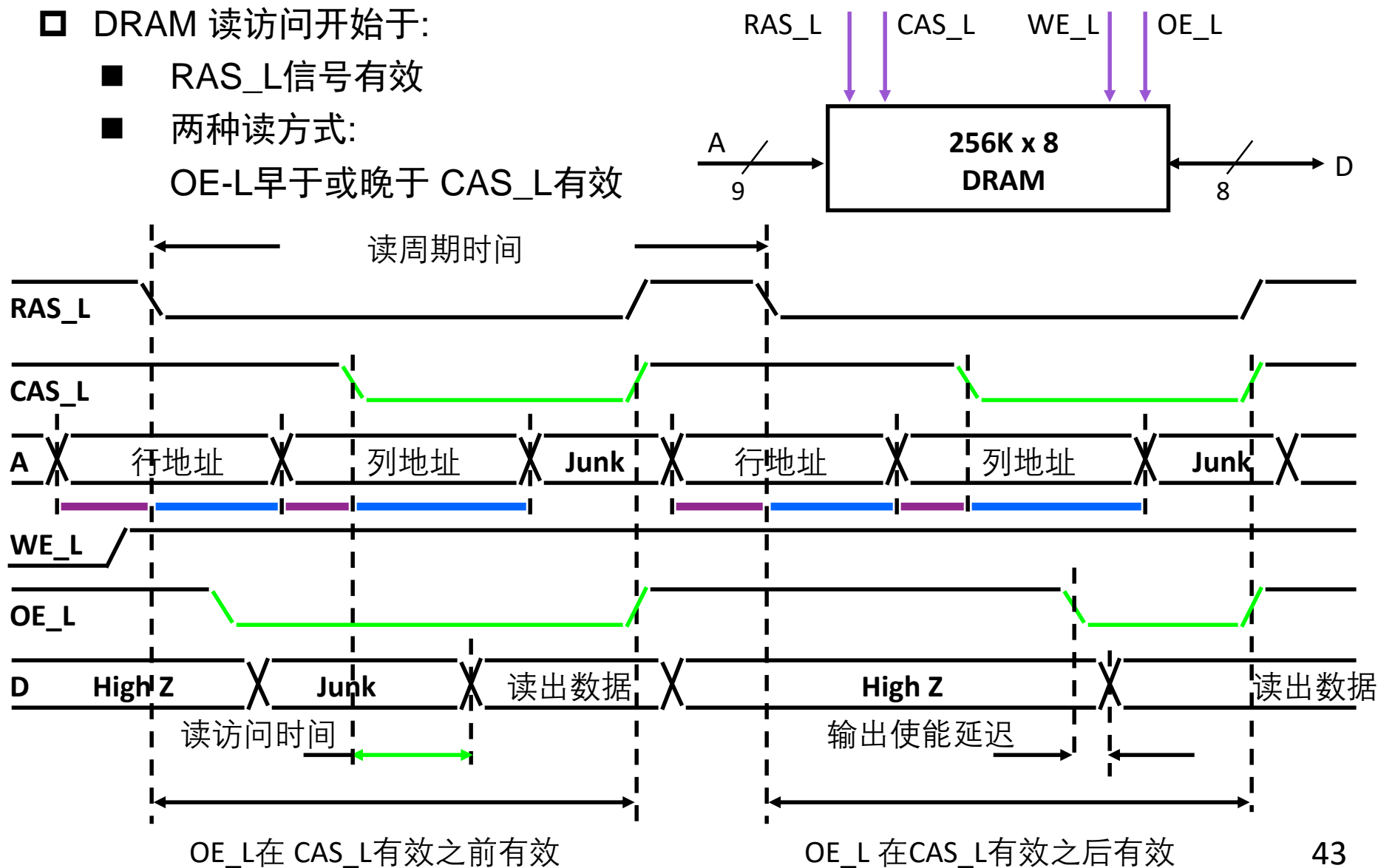
DRAM 读时序

□ DRAM 读访问开始于:

■ RAS_L信号有效

■ 两种读方式:

OE-L早于或晚于 CAS_L有效



小结

□ 程序的局部性原理:

- 时间局部性：最近被访问过的程序和数据很可能再次被访问
- 空间局部性：**CPU**很可能访问最近被访问过的地址单元附近的地址单元。

□ 利用程序的局部性原理:

- 使用尽可能大容量的廉价、低速存储器存放程序和数据。
- 使用高速存储器来满足CPU对速度的要求。

□ 动态存储器DRAM

- 电容充放电来存储数据
- 集成度高、容量大、能耗低、速度慢

阅读和思考

□ 阅读

□ 思考

- 程序的局部性原理指什么?为什么层次存储器系统能同时达到高性能/低成本/大容量的指标?

谢谢