

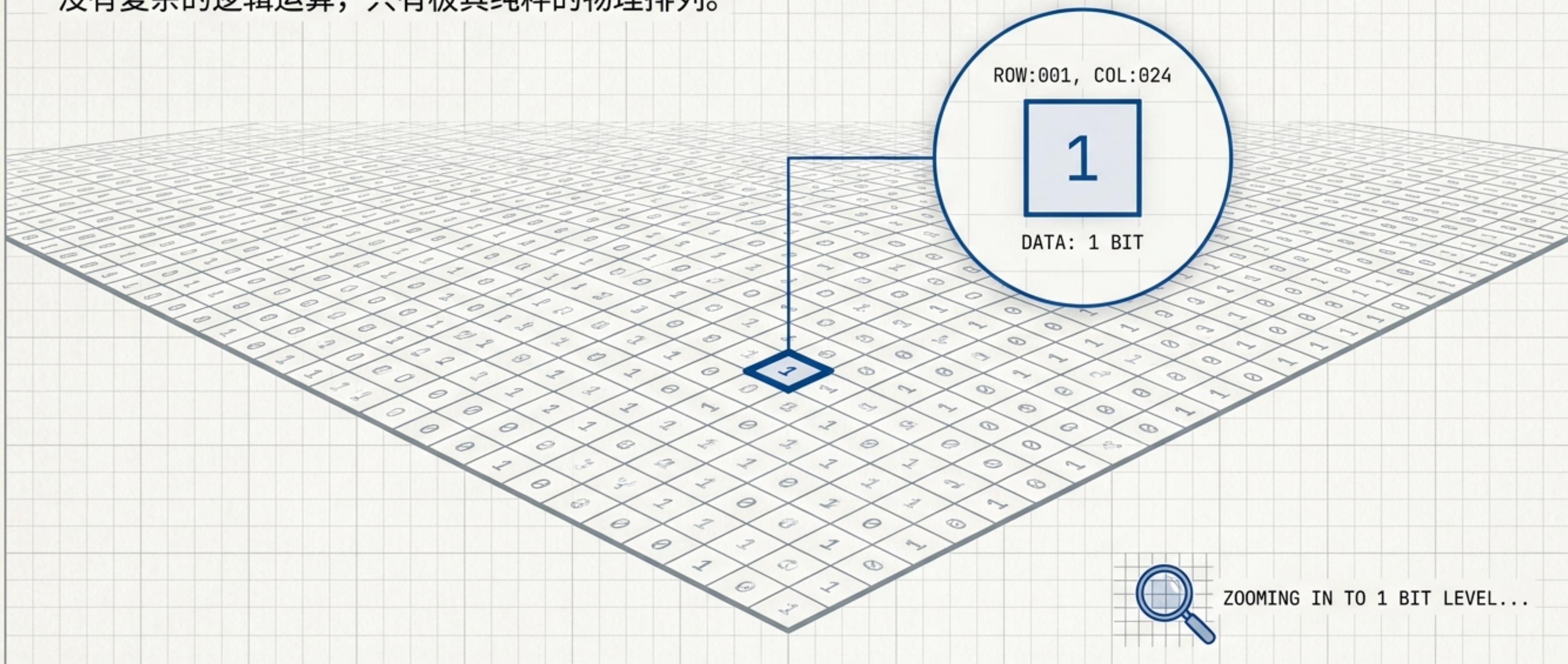
存储的物理学

从几十飞法的电容，到百亿美金的商业帝国

过去五十年，几乎所有的计算都建立在极其微小、甚至随时在漏电的两个零件上。今天，我们将把视野放大一亿倍，看透内存运转的物理真相。

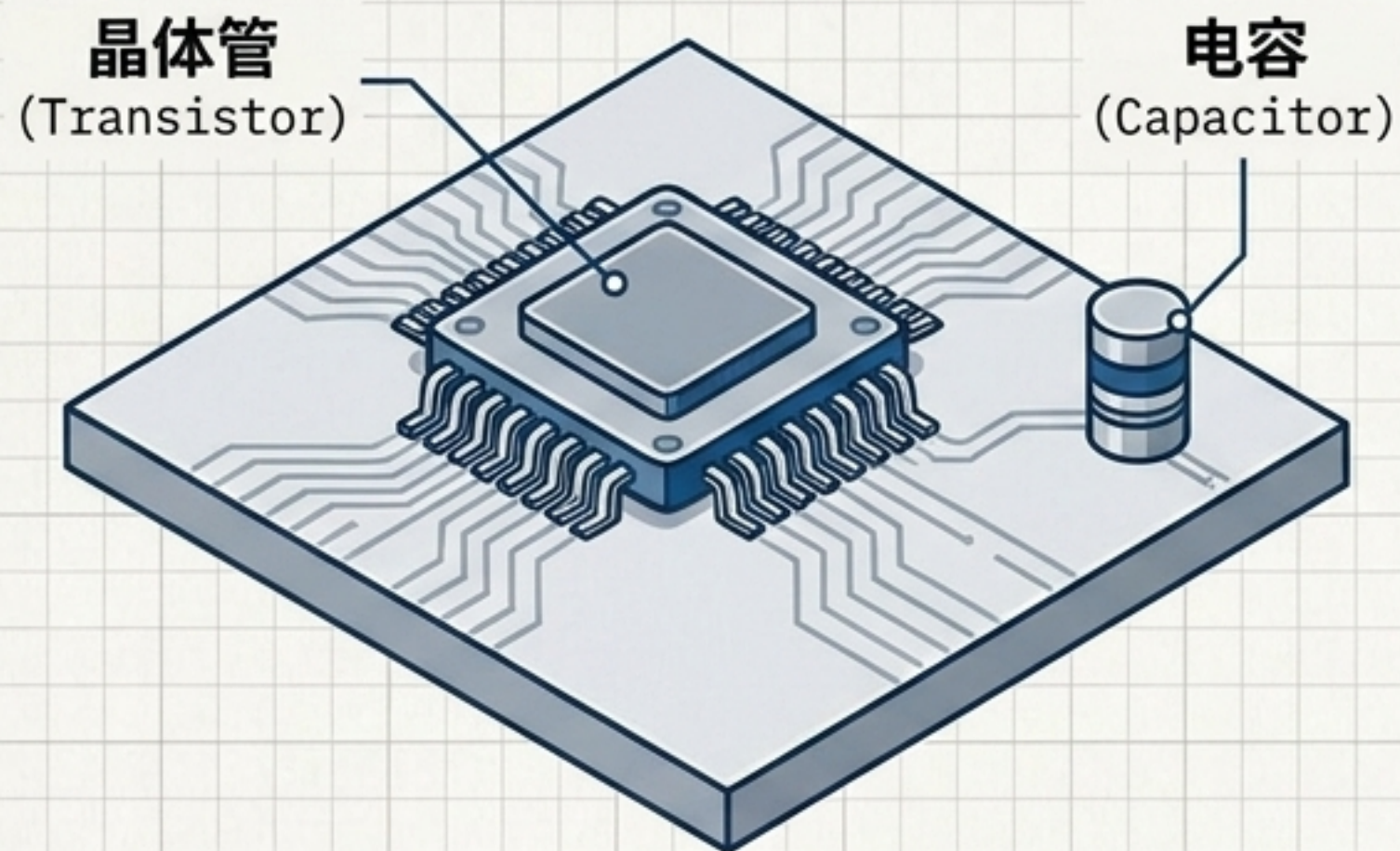
内存，本质上是一个浩瀚的矩阵网格

每一格 (Cell) 只做一件事：存储一个比特 (0 或 1)。
没有复杂的逻辑运算，只有极其纯粹的物理排列。

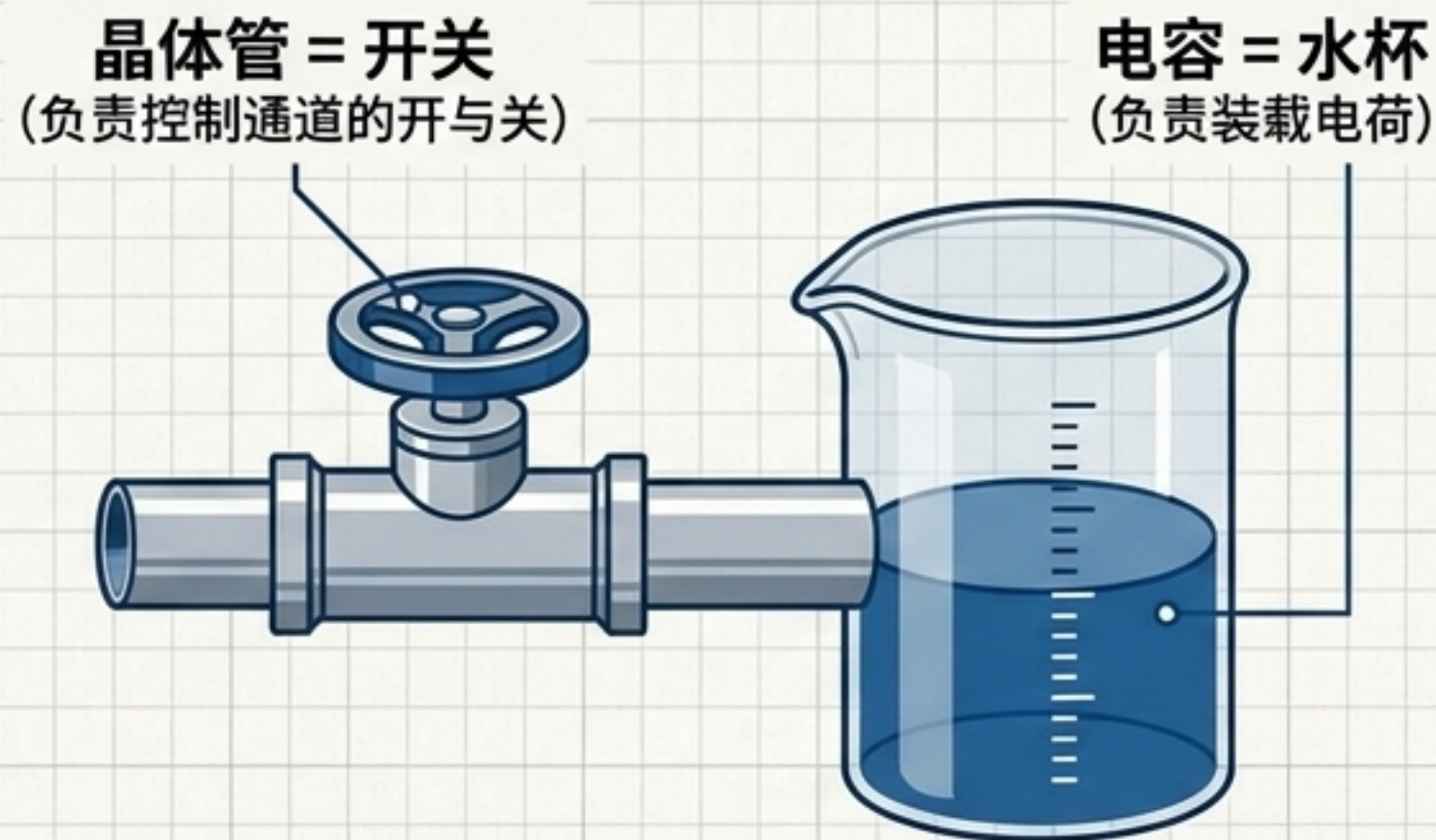


存下一个比特的全部家当：一个开关，一个水杯

现实 (The Reality)



比喻 (The Metaphor)

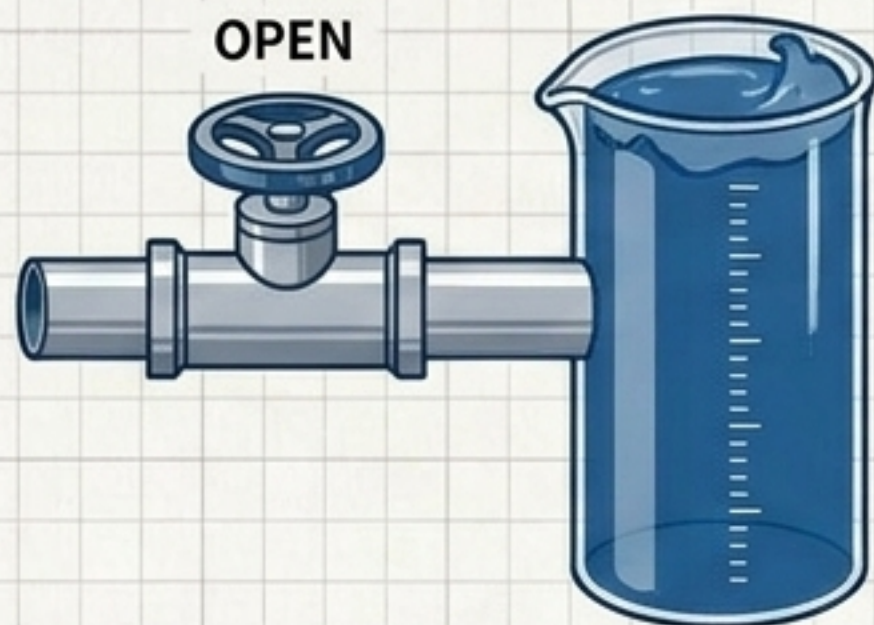


有电荷就是 1，空了就是 0。就这么朴素。

读与写，纯粹的物理动作

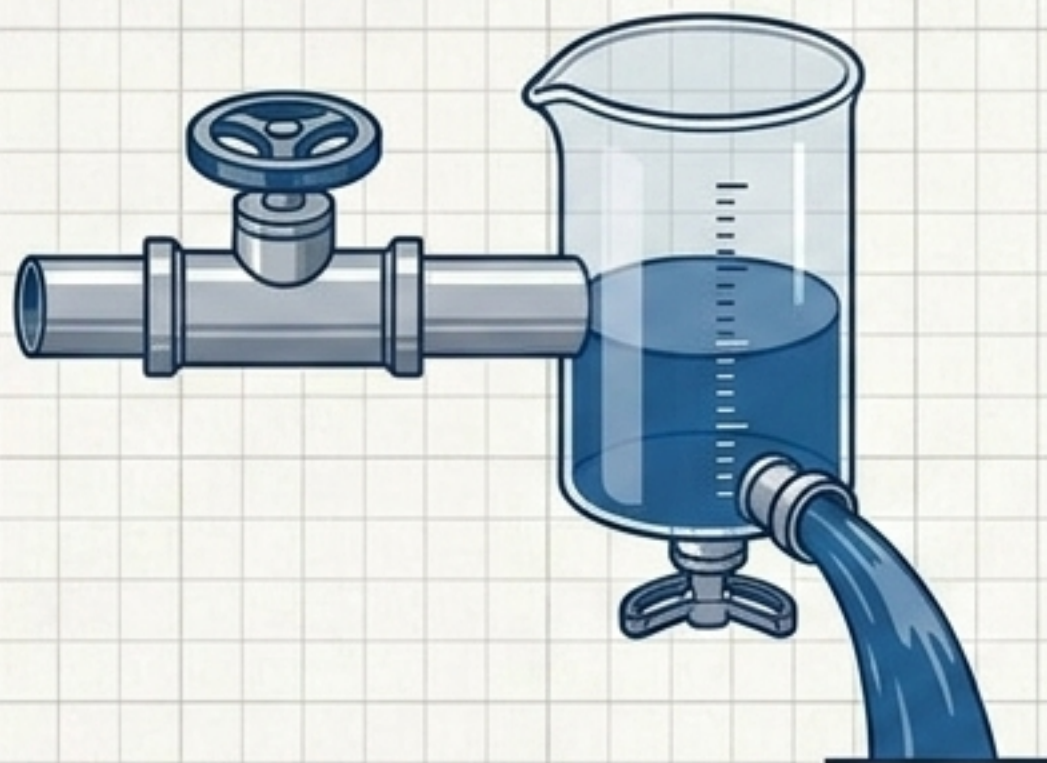
数据不是虚无缥缈的代码，而是真实存在于硅片上的电子集结与消散。

写 1 (Write 1)



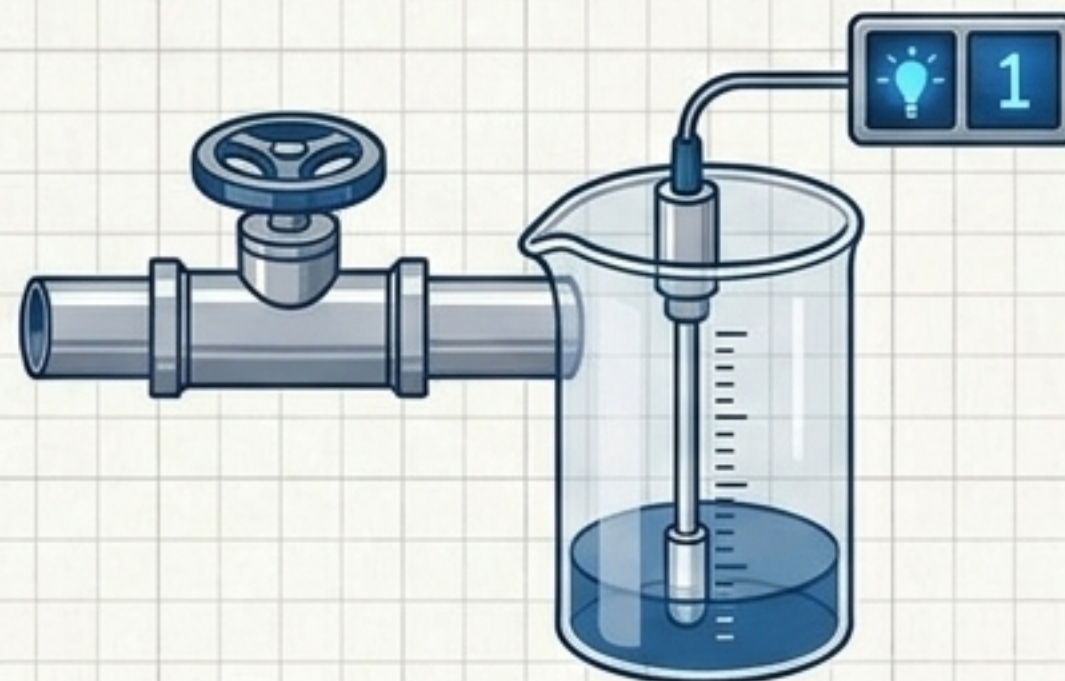
开关打开 → 电流注入 → 电容蓄满水

写 0 (Write 0)



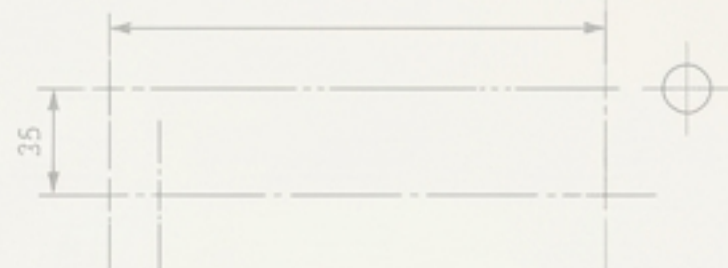
开关打开 → 底部阀门放水 → 电容排空

读取 (Read)



开关打开 → 传感器检测杯底是否有水
→ 判定 1 或 0

致命缺陷：极其微小的容量，毫秒级的漏电



1

极其微小的电容

容量仅为 几十飞法
(Tens of Femtofarads)

2

自然漏电

里面的电荷在 毫秒级
就会自然漏光。

3

强制刷新机制

每隔几十毫秒，必须把所有
电容读一遍再重新写回去
(注水)。在此期间，内存
无法进行任何读写工作。



它得不停地给自己抄笔记，不抄就忘。而在抄笔记期间，它什么活儿都干不了。

为什么内存的进步，远远慢于处理器？

存这个比特的物理单元本身变成了瓶颈。内存频率的提升，永远追不上 CPU 的狂飙。

旧制程 OLD NODE



较大的电容，漏电相对较慢，
刷新压力尚可承受。



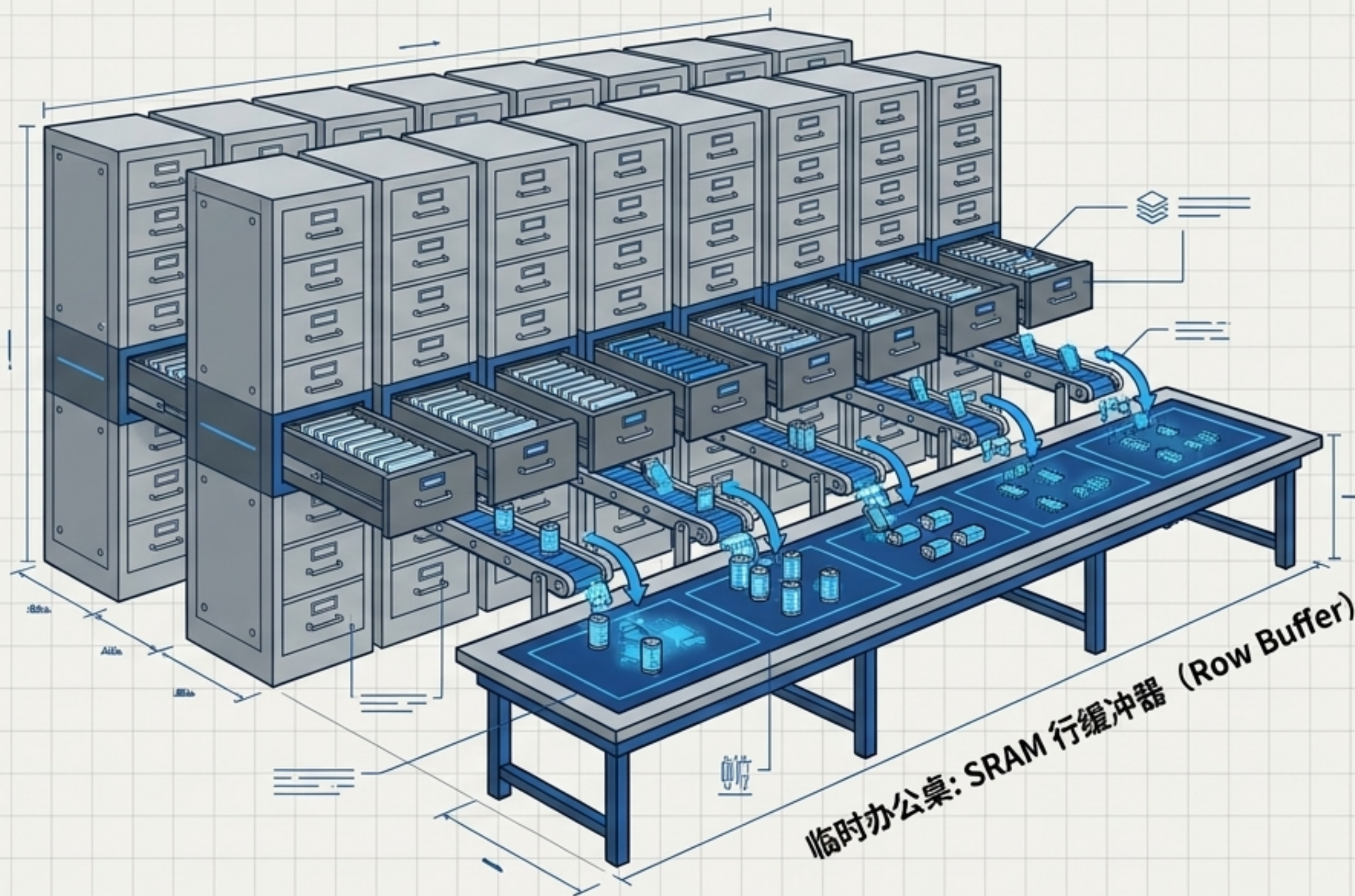
摩尔定律的副作用

现代制程 MODERN NODE



制程每缩小一次，电容随之变小，
但漏电孔无法等比例缩小。漏电相对极快，
必须以极高频率疯狂刷新。

矩阵寻址：每一次读取，都是牵一发而动全身



要访问某一格，
不能只拿那一格。

必须先选定「行」。
一整排电容的电荷会被
全部强制读出，倾倒入
前面的临时长桌
(SRAM 行缓冲器)
上。

然后，才能从这张桌
子上挑选你真正需要的
「列」。

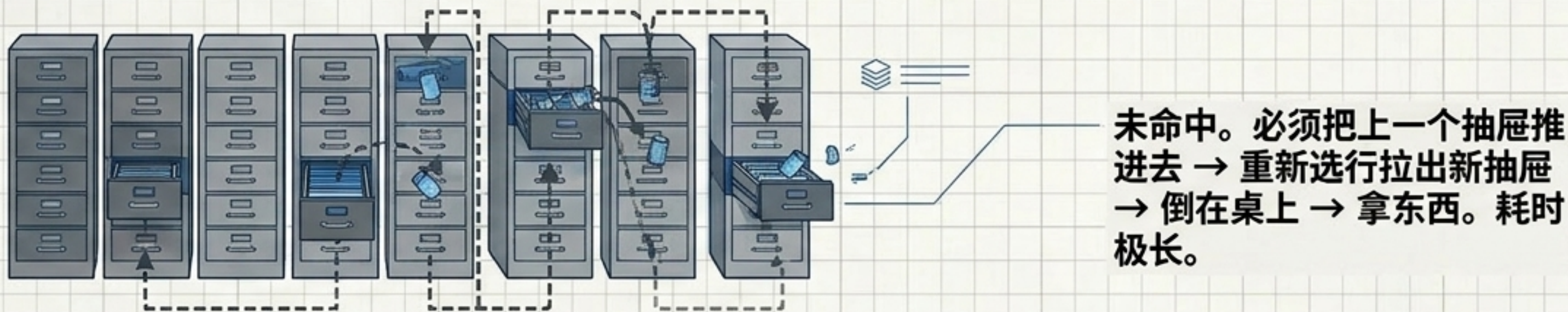
顺序与随机的鸿沟：为什么局部性如此重要？

顺序访问比随机访问快几十倍，这就是软件工程中天天强调的「内存局部性」的物理真相。

顺序访问
(Sequential Access)
— 极速



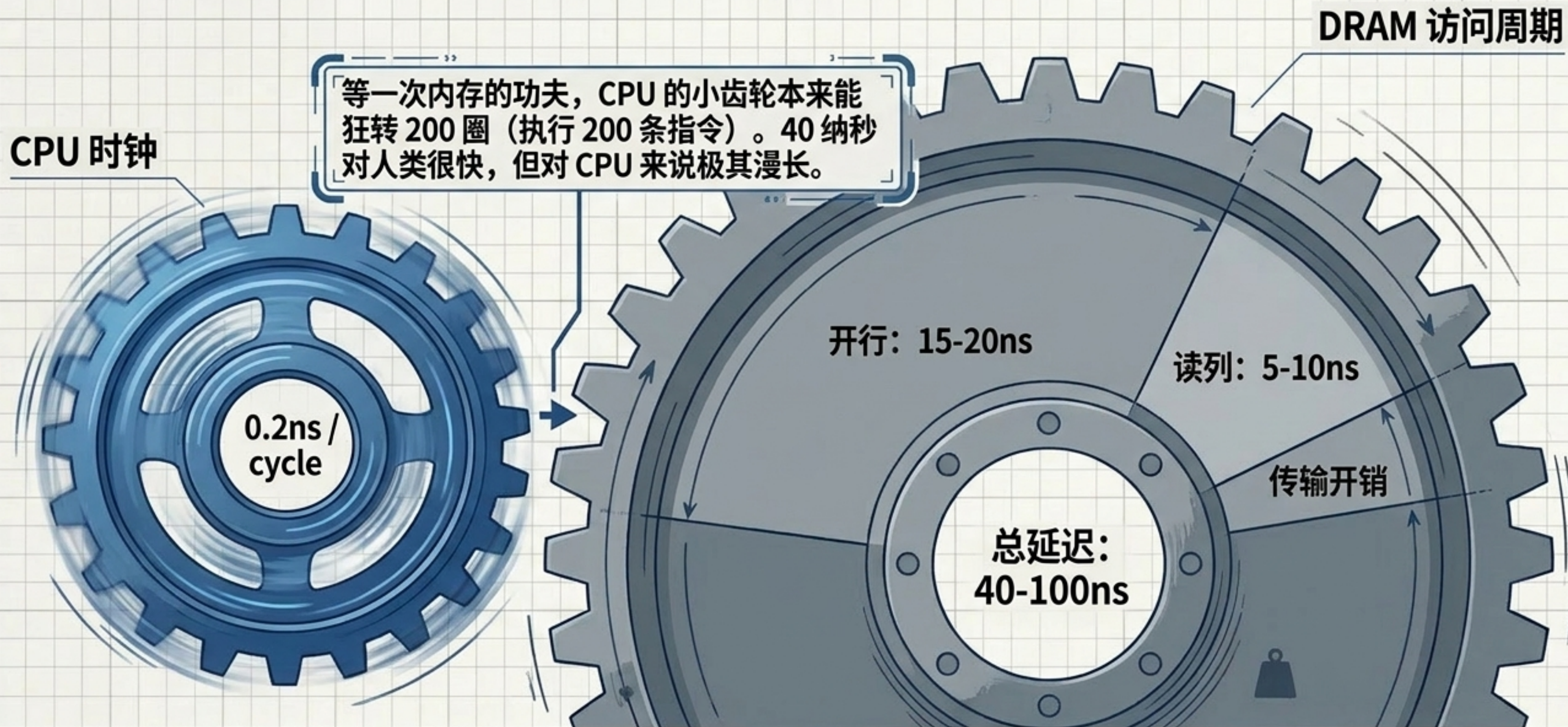
随机访问
(Random Access)
— 极慢



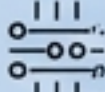
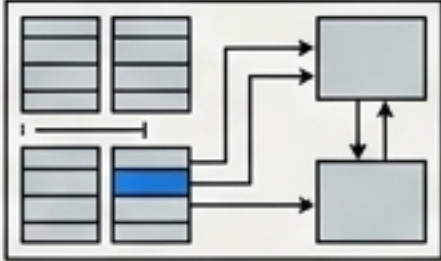
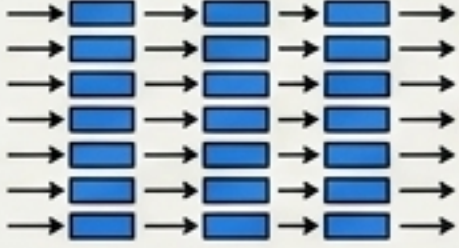
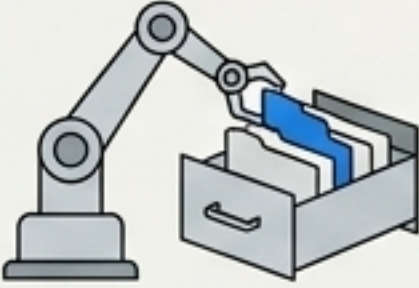
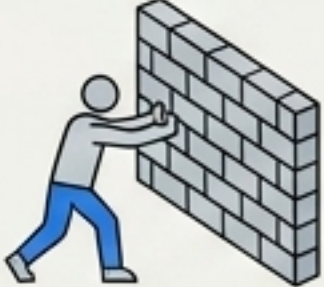
CPU 缓存的预读、矩阵乘法的可预测扫描，本质上都是在迎合这套「抽屉与桌子」的物理规则。

存储墙 (The Memory Wall)：处理器的时间相对论

处理器大部分时间其实在空转，在绝望地干等内存。



对抗存储墙：算力双雄的解法

	 CPU（逻辑派）	 GPU（吞吐派）
核心策略	押注局部性，堆砌超大缓存（SRAM） 	掩盖延迟，堆砌海量线程（Threads） 
时间流速体验	试图让内存显得像 0.2ns 一样快 	只要活跃线程够多，就不在乎内存慢 
运作隐喻	智能预测你下一步要拿哪个抽屉，提前备好 	一个工人等抽屉，立刻切换另一个工人去算 
致命弱点	预测失败（缓存未命中）时延迟极高 	当 Batch Size 为 1（无线程可切）时，墙再次出现 

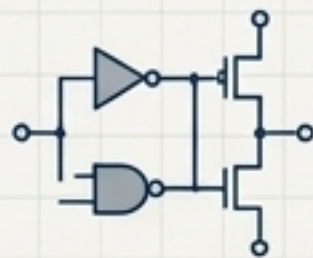
科技树的分歧：为什么台积电不代工 DRAM？

同样叫 5nm，两边的工艺参数完全是两回事。DRAM 无法借用逻辑芯片的产线。



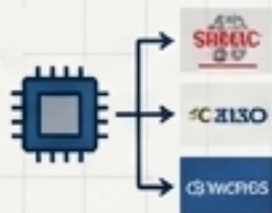
逻辑工艺（如 台积电 TSMC）

- **核心追求：**极高的开关速度与 计算功耗控制



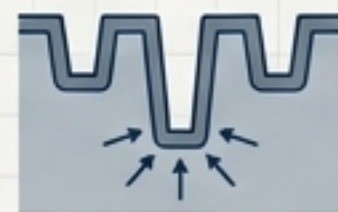
- **优化方向：**晶体管的极速响应，逻辑门电路的密度。

- **产线特点：**通用代工模式，服务海量无晶圆厂芯片设计公司。



存储工艺（DRAM 专线）

- **核心追求：**极限的电容密度与 严苛的漏电控制



- **优化方向：**在极小空间内挖出巨大的深坑做电容，并死磕漏电率。

- **产线特点：**必须拥有独立研发体系和专门的晶圆制造工厂（IDM）。



百亿美金的护城河：三大寡头的终极游戏

全世界玩得起这个游戏的只剩三家公司。这解释了为什么内存的底层创新，多年来一直落后于处理器。



极高的资本门槛

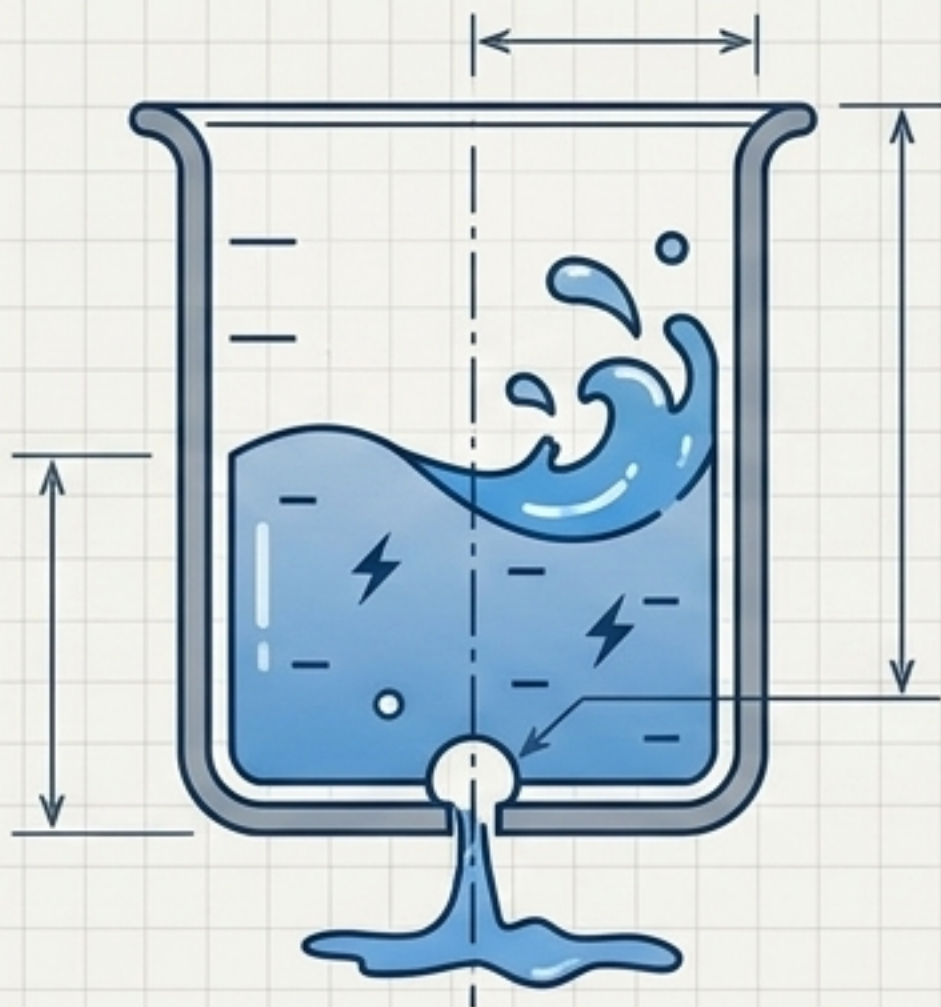
百亿美元级的单代资本开支，不允许任何试错。

极高的技术门槛

长达十年以上的闭源工艺积累，无法通过外部购买 IP 追赶。

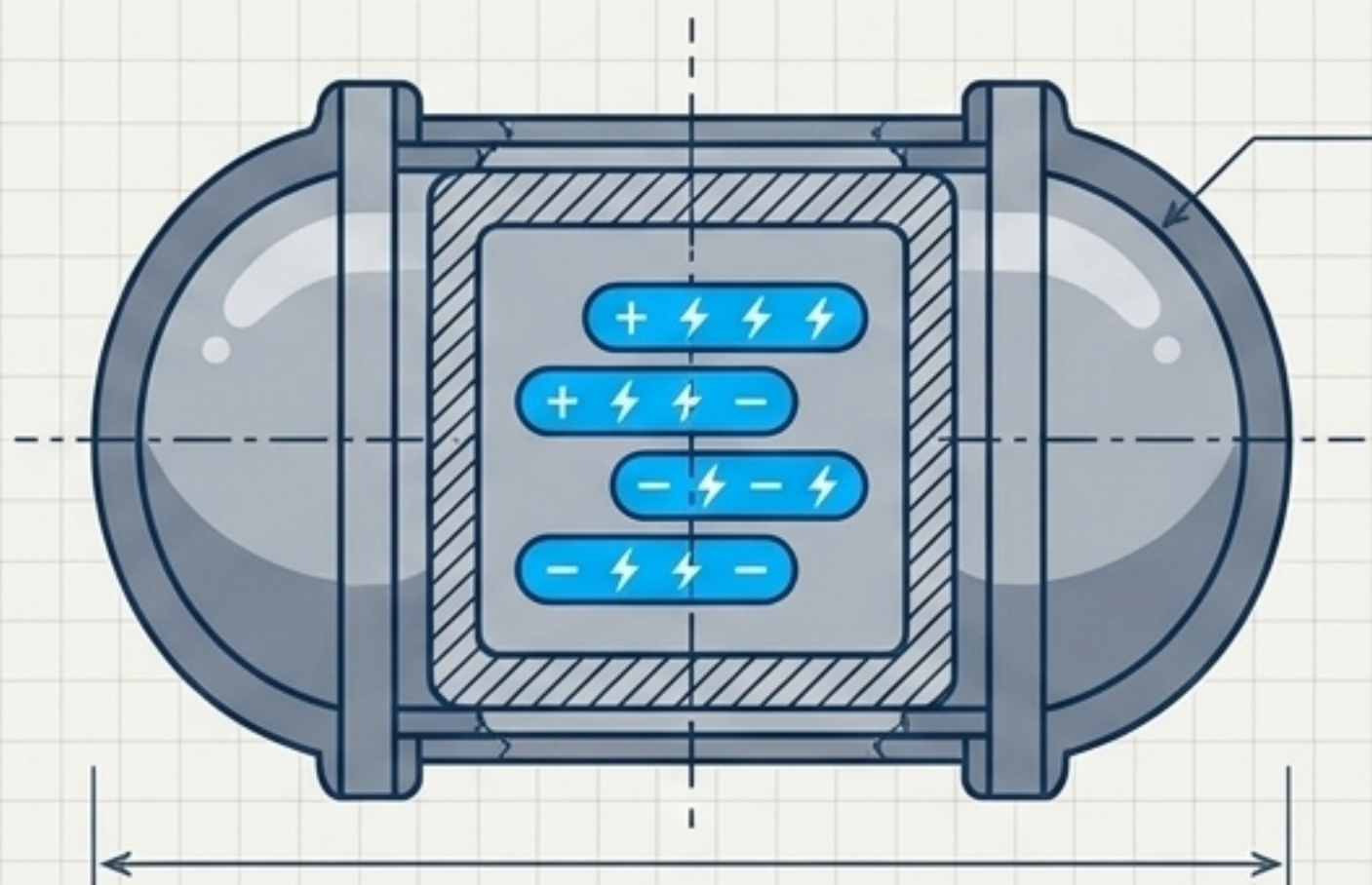
同样是存数据，DRAM 与 NAND 的物理宿命

DRAM（运行内存）



靠电容装电荷。断电（或停止刷新）数据瞬间灰飞烟灭。速度极快。

NAND（硬盘存储）



将电子死死「困」在绝缘层里。断电也跑不出来，代价是读写速度慢三个数量级。

计算机物理分工的底层逻辑

存储双雄的宿命之战，一锤定音的技术选型总结。

[维度名称]	DRAM（运行内存）	NAND（硬盘存储）
物理机制	电容蓄水，极易漏电	绝缘层彻底困住电子
状态维持	必须通电且疯狂刷新 (易失性)	彻底断电，数据永存 (非易失性)
速度级别	纳秒级 (ns)	微秒/毫秒级 (慢千倍)
系统分工	运行工作台 (RAM)	永久仓库 (Storage)

顿悟时刻：电容漏电的蝴蝶效应

软件里的疯狂优化、GPU的算力狂飙、半导体产业的格局.....根源，都在于极微观世界里那会漏水的杯子。

物理源头：微观世界里，几十飞法的电容会不可避免地漏电。

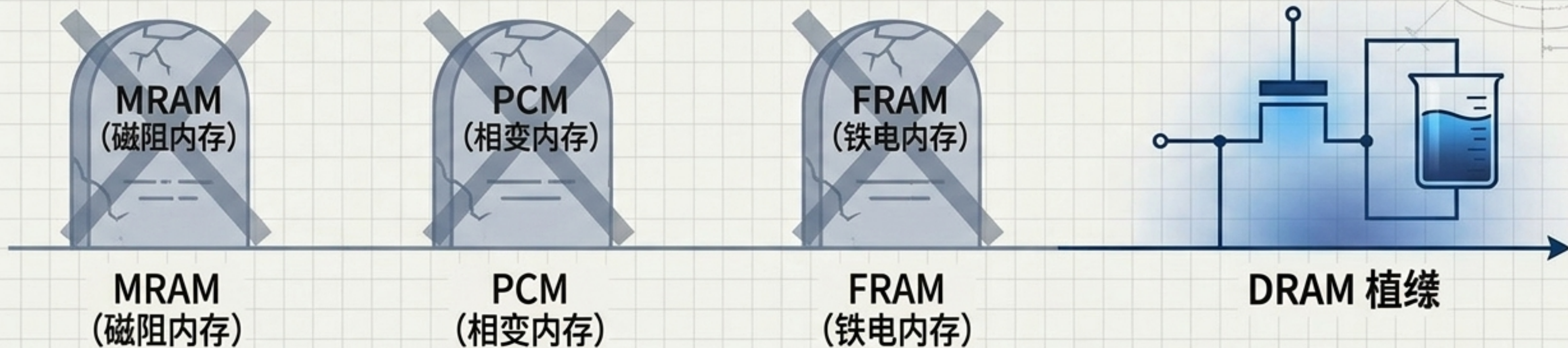
硬件妥协：必须每几十毫秒全局停止工作，进行刷新。

软件补救：为避开寻址延迟，必须通过预测压榨「内存局部性」。

系统分化：CPU 堆砌海量缓存，GPU 堆砌海量线程，来掩盖物理劣势。

商业壁垒：为在极小空间死磕漏电控制，造就了三大寡头百亿美金的独立产线。

并不完美的五十周年，却是最好的解法



过去几十年，无数人试图换掉 DRAM，
但全败给了成本或良率。

计算机工程史上最伟大的妥协与挣扎，
在可见的未来，仍将由这个朴素的「一个晶体管 + 一个电容」继续支撑。