

2020-2021学年秋季学期

第七部分 半导体存储电路

授课团队：宋威

助 教：薛子涵

半导体存储电路

数字电路需要在电路中保存电路的状态、中间结果、配置信息甚至任何代码与数据。这就需要存储电路。

○ 半导体存储电路

○ 寄存器 (register)

- 锁存器 (latch) , 寄存器 (register)
- 高速电路, 无读写限制, 用于时序逻辑电路的设计
- 门电路的基本单元

○ 静态随机存储器 (SRAM: static random access memory)

- 缓存, 片上内存
- 高速电路, 面积小, 单一时刻只能读写一个数据, 用于在片上存储大量数据

○ 动态随机存储器 (DRAM: dynamic random access memory)

- 片内内存, 片外内存
- 速度较低, 面积很小, 读写带宽大, 地址受限

○ 非易失存储器 (non-volatile memory)

- 掉电不丢失内容的存储器
- 片上只读内存 (ROM) , 片外可编程ROM (Flash)

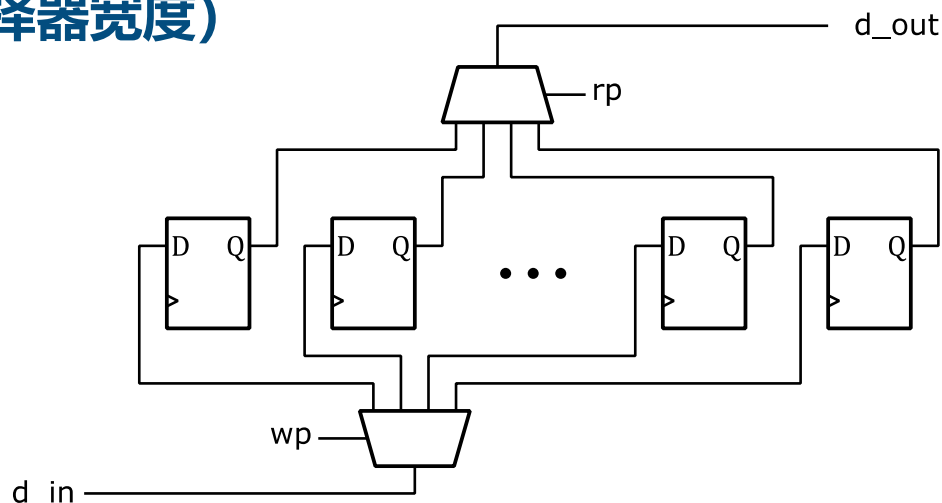
静态随机存储器

在芯片内部保存大量数据的高速数据存取电路。

- 大量：存储密度相对寄存器要高
- 高速：存储和读取速度相比其他高密度存储电路块
- 限制：每个周期读或写一个数据

○寄存器的问題：

- 面积大
- 速度也不一定快（选择器宽度）



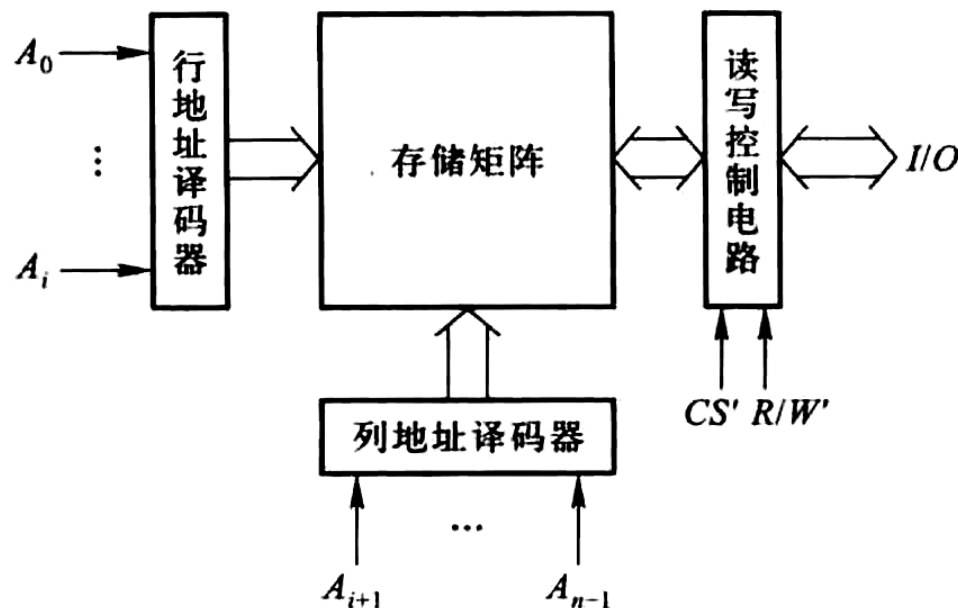
静态随机存储器

SRAM使用二维的存储矩阵而非一维存储向量

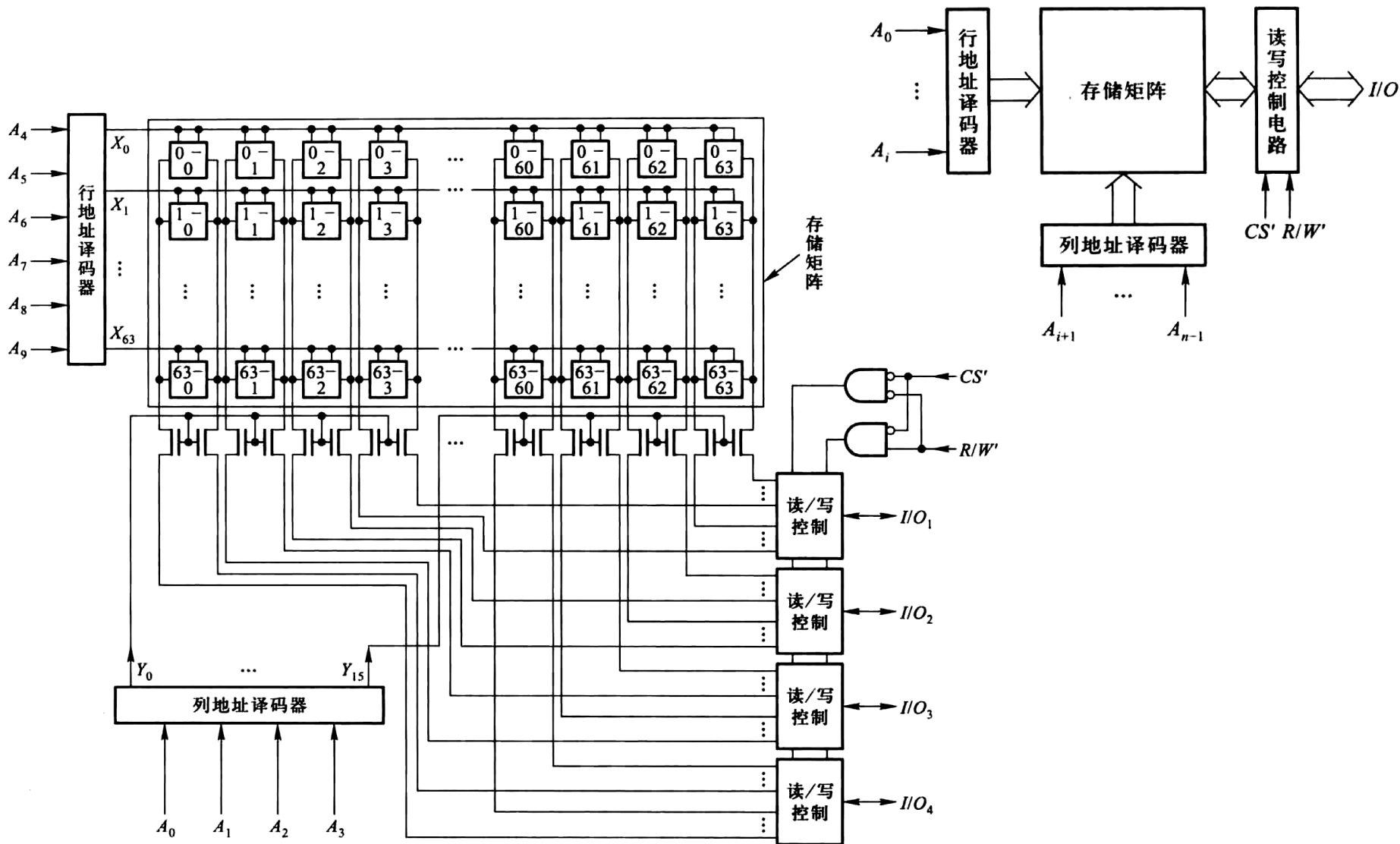
- 存储矩阵的密度更大
- 地址译码简单（类似选择器的问题）

- 1024x4的SRAM实际
实现为64x64的矩阵
行：6位地址
列：4位地址

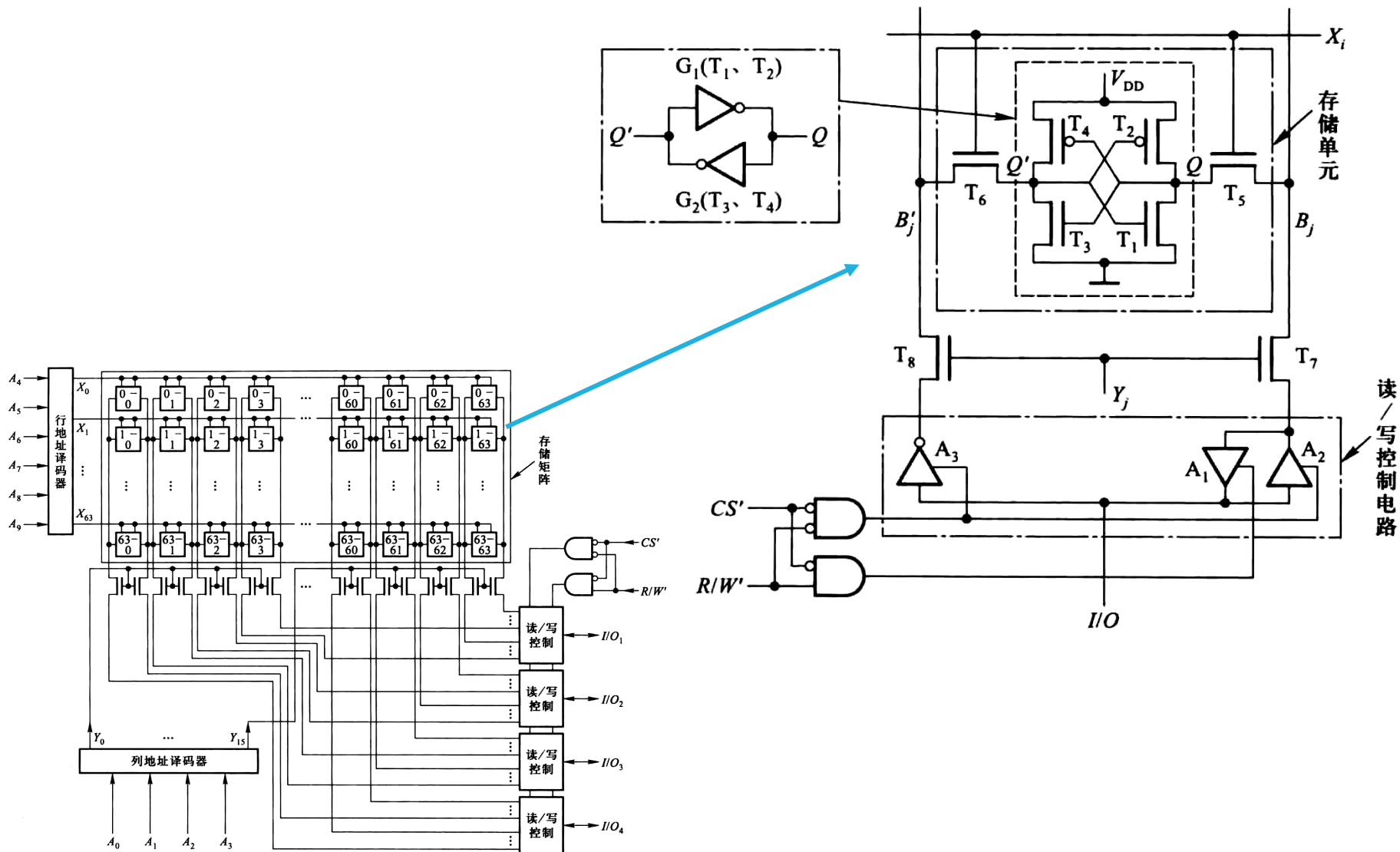
- CS' 片选
- R/W' 读写控制



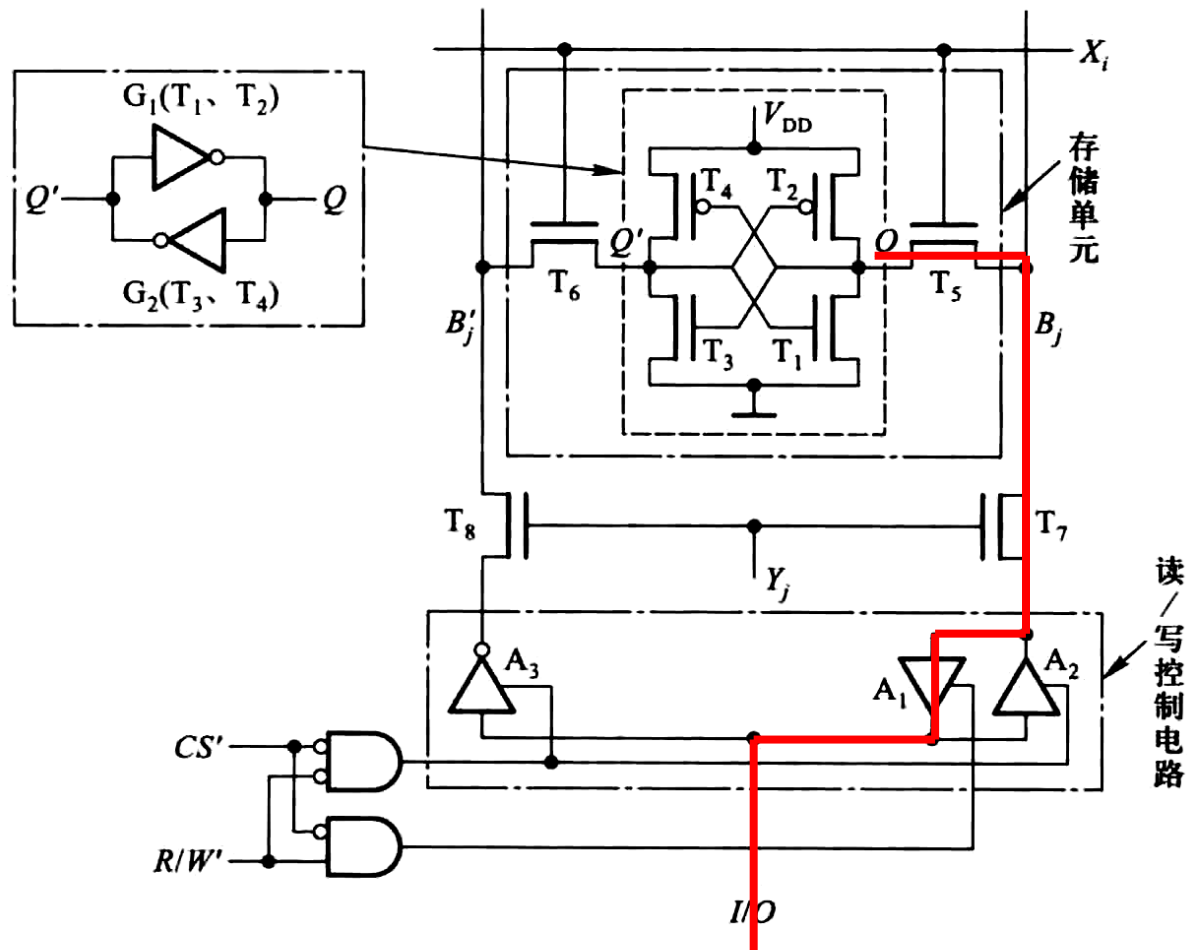
静态随机存储器



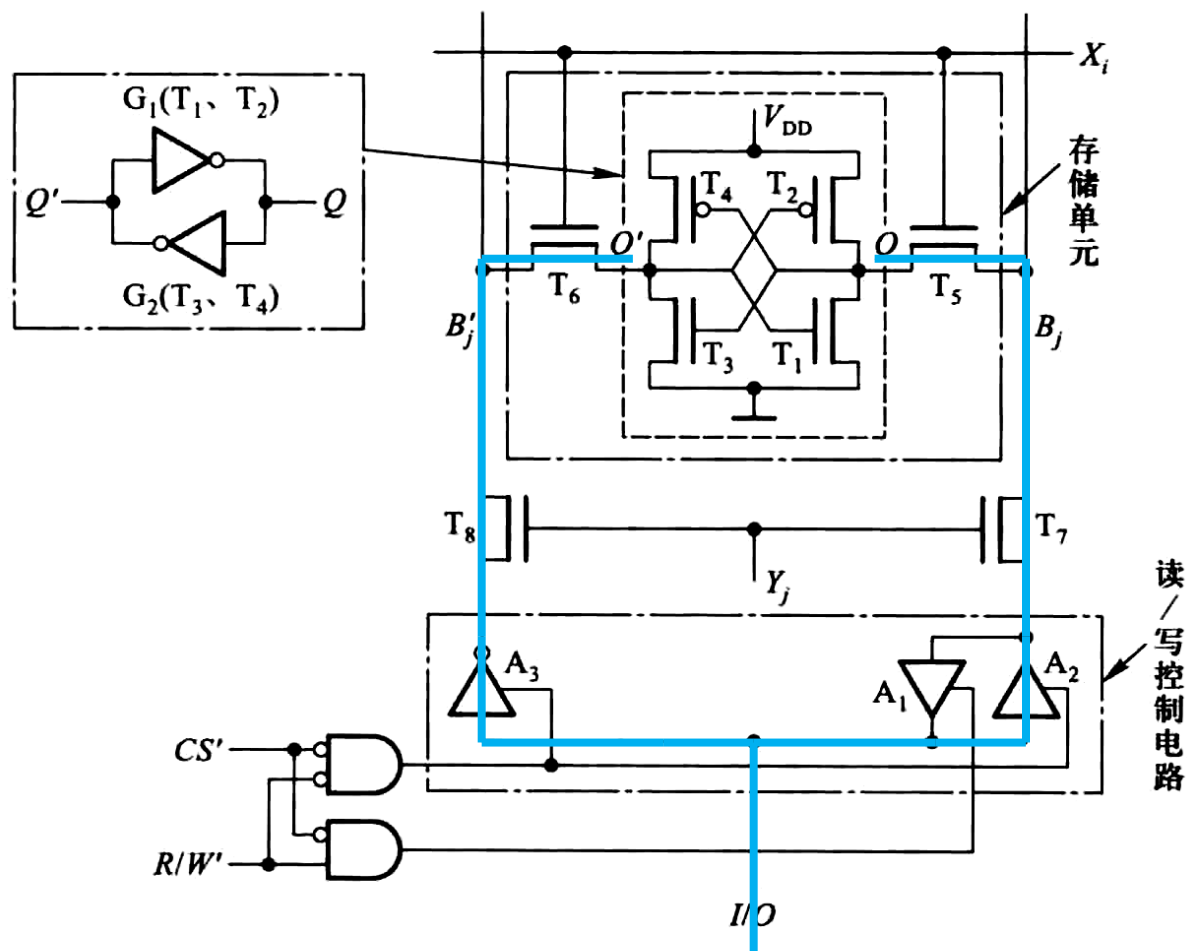
静态随机存储器



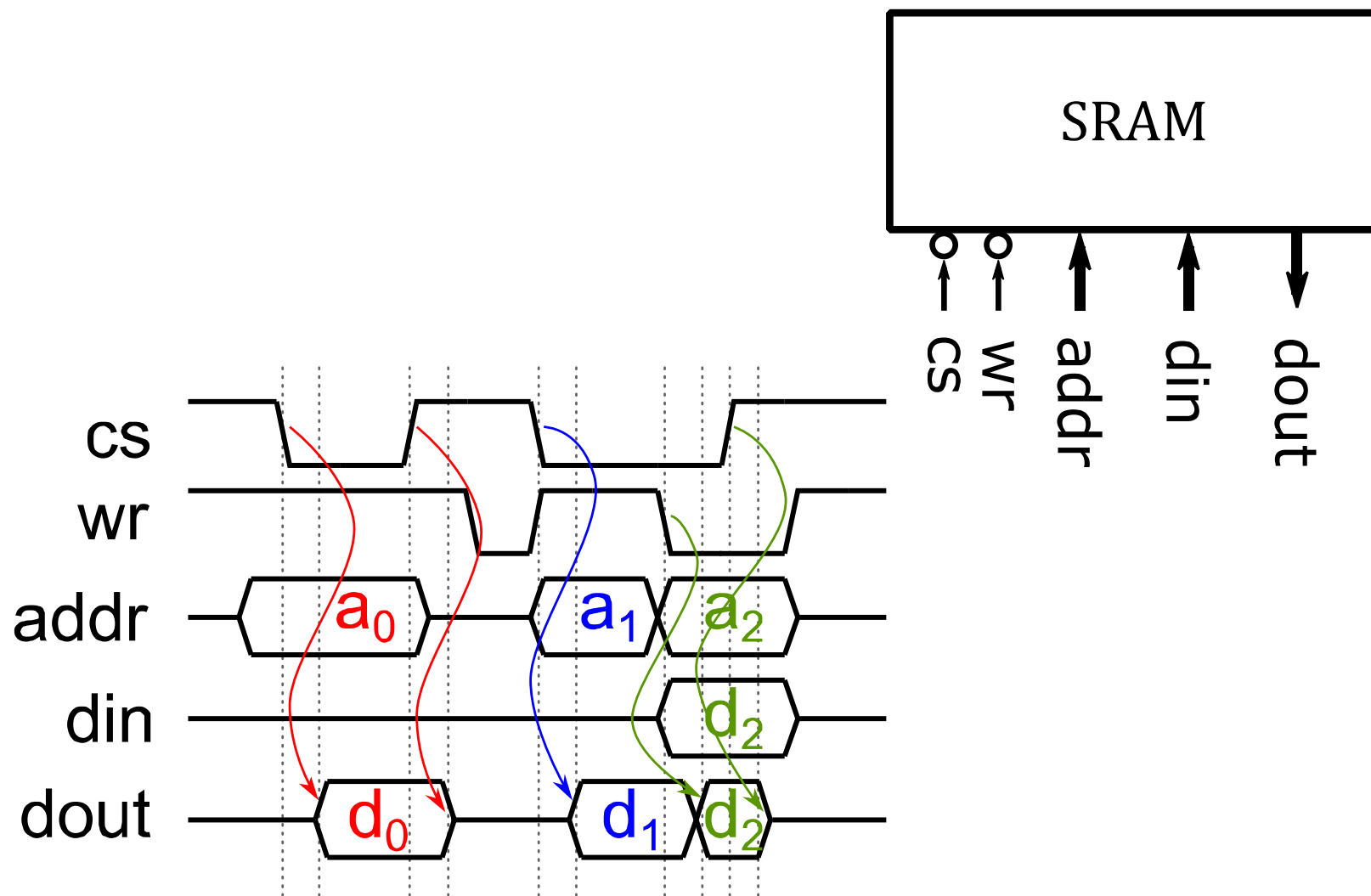
静态随机存储器——读操作



静态随机存储器——写操作



静态随机存储器读写时序



静态随机存储器总结

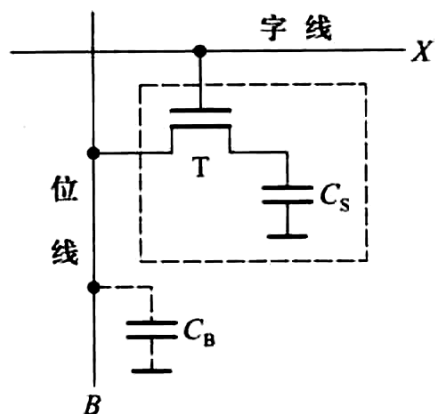
○构造

- 数模混合电路
- 由反门正反馈环保存状态
- 高密度存储矩阵
- 不需要时钟，但往往配合时钟使用
- 同一时刻只能读写一个数据

○使用

- 作为大规模集成电路的高速数据存储
 - 处理器的Register File
 - 一级缓存
 - 缓冲存储

动态随机存储器——存储单元



1T MOS存储单元

读: C_S 对 C_B 充电

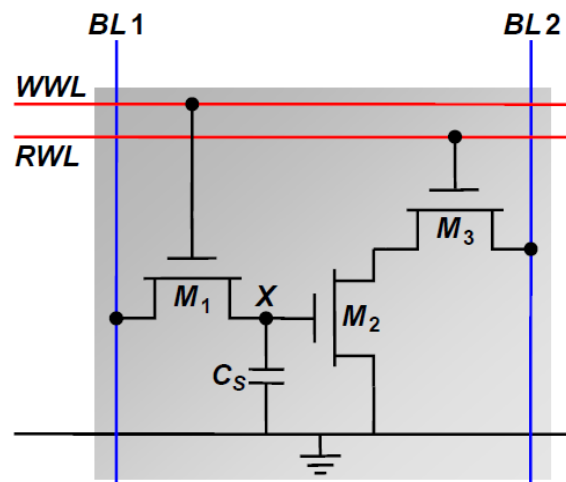
$$v_B = \frac{C_S}{C_S + C_B} v_{C_S}$$

读出的信号微弱, 需要信号放大器。

读操作是毁灭性的

写: 对 C_S 充电

刷新: 循环地将读出的值写回 C_S



3T MOS存储单元

读: M_3 接通,

根据 V_X 看BL2电压是否下降。

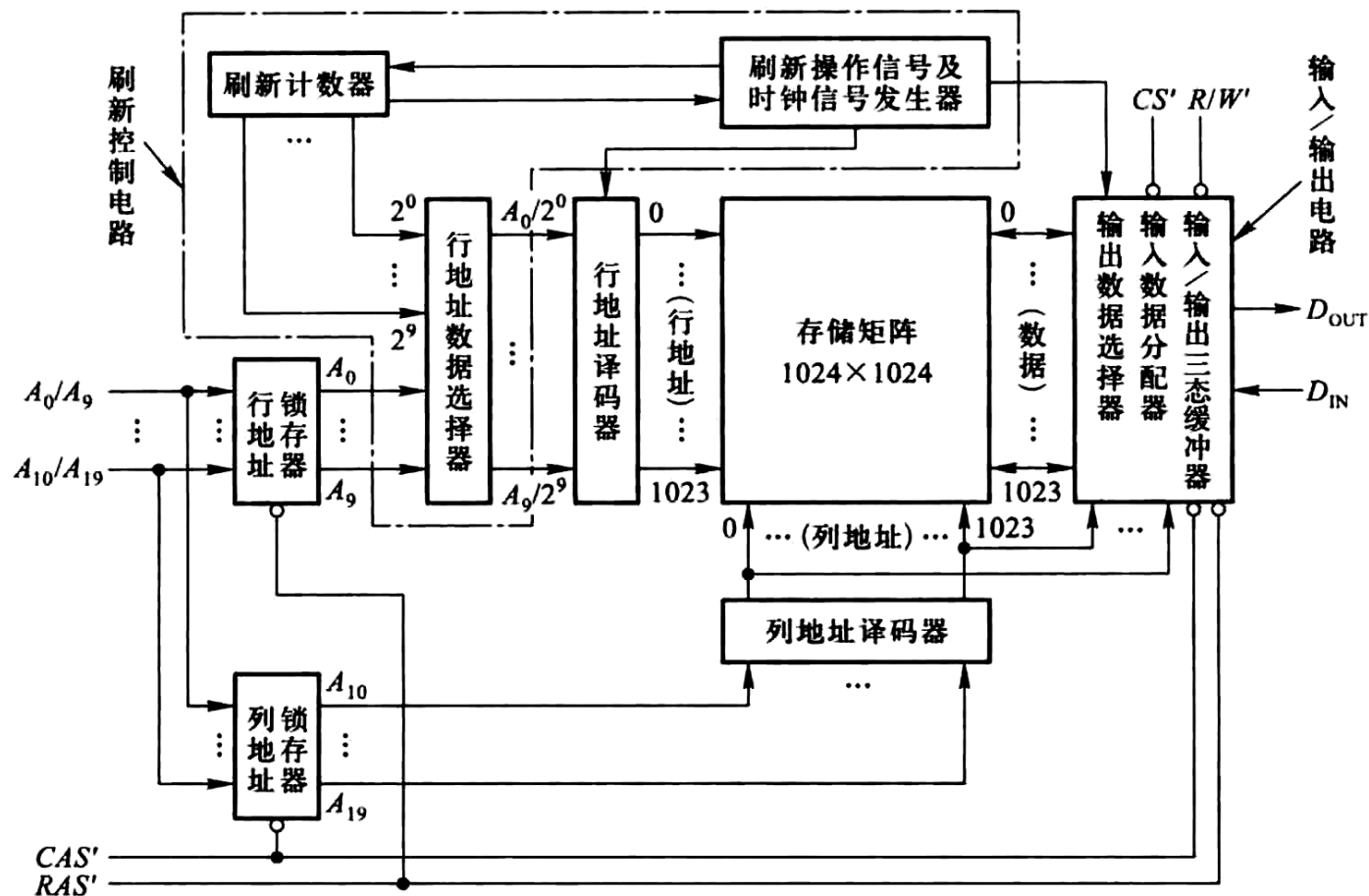
读操作不损坏 V_X

写: M_1 接通, 对 C_S 充电

刷新: 循环地将读出的值写回 C_S

为了更小的面积, 现在大量使用1T结构。

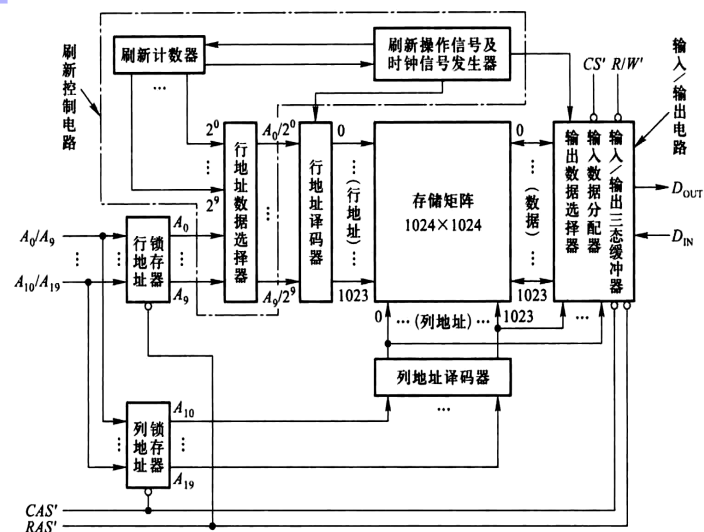
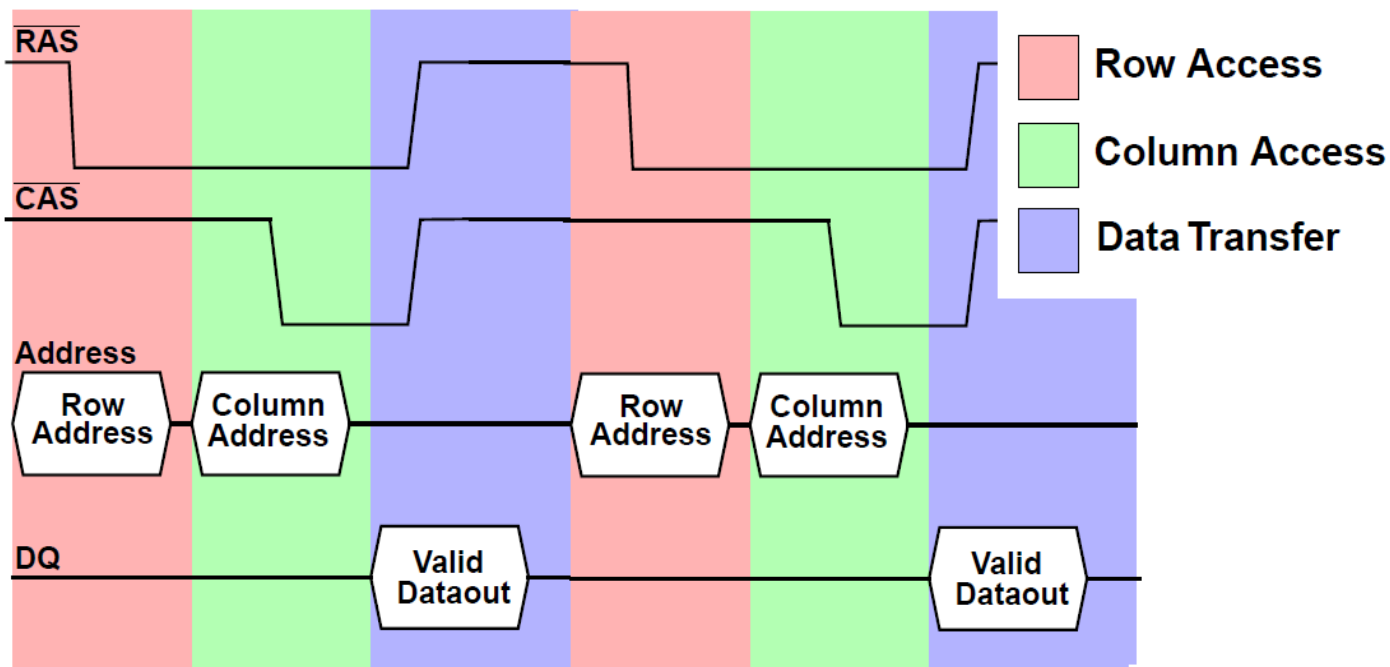
动态随机存储器——整体结构



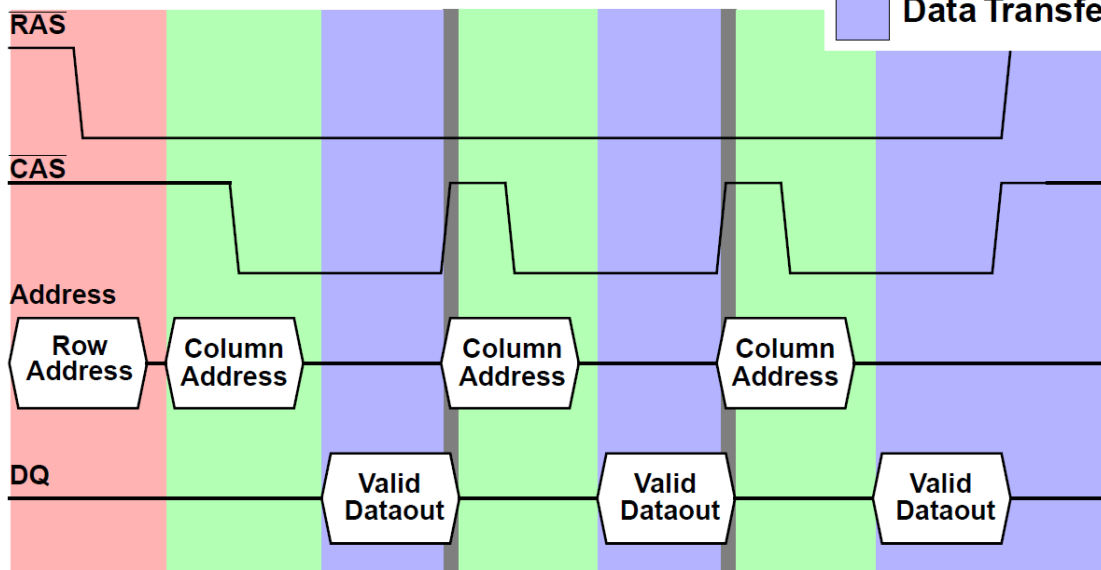
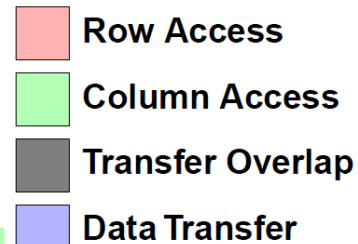
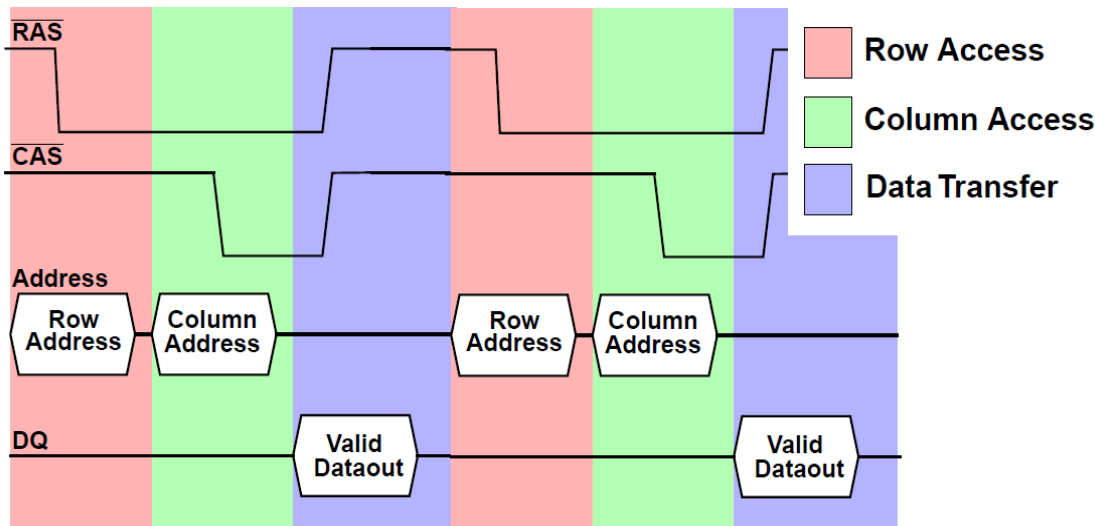
- 存储矩阵：先行后列（管脚复用）
- 刷新控制：10ms周期性，刷新时不可读写
- 信号放大：读出地信号微弱，需要放大

控制时序复杂！

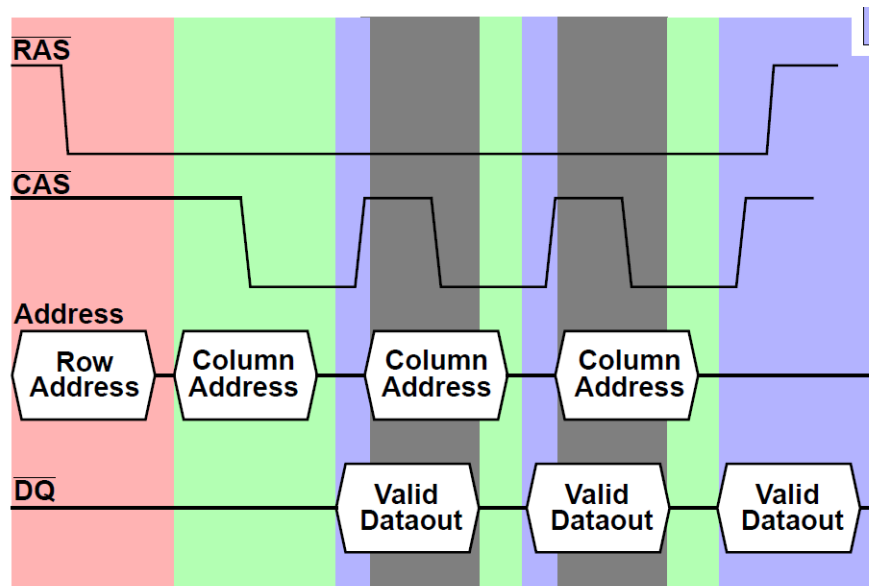
动态随机存储器——时序



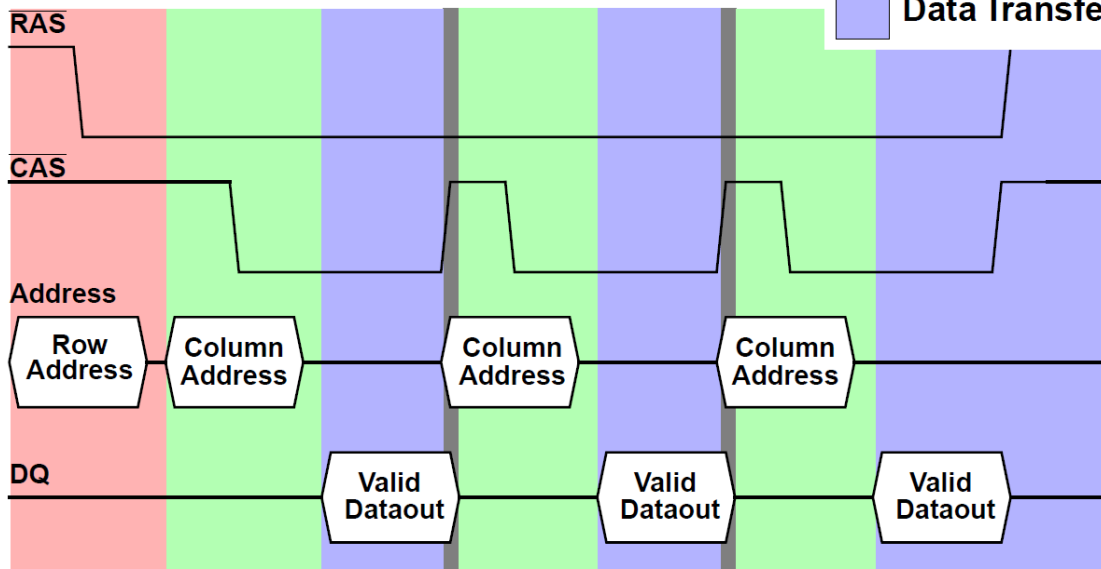
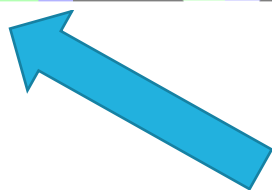
动态随机存储器——时序改进



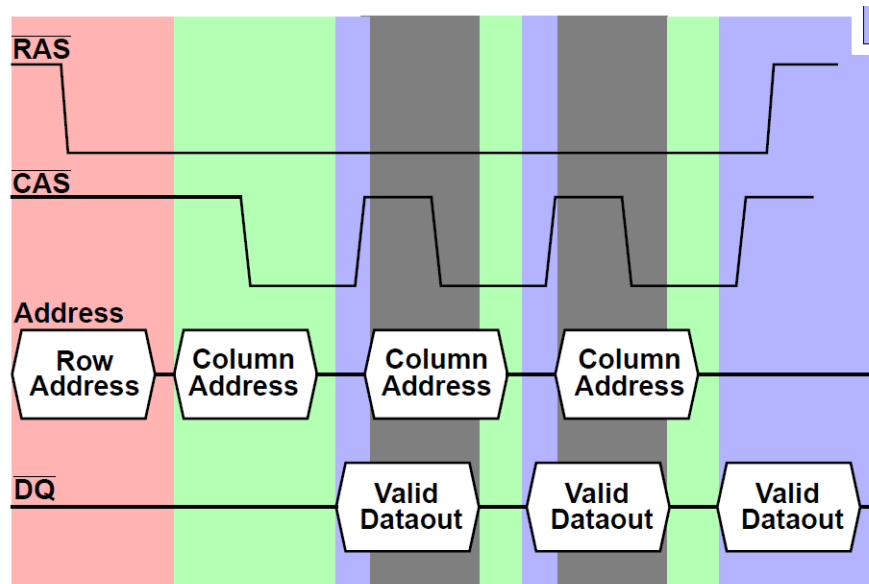
动态随机存储器——时序改进



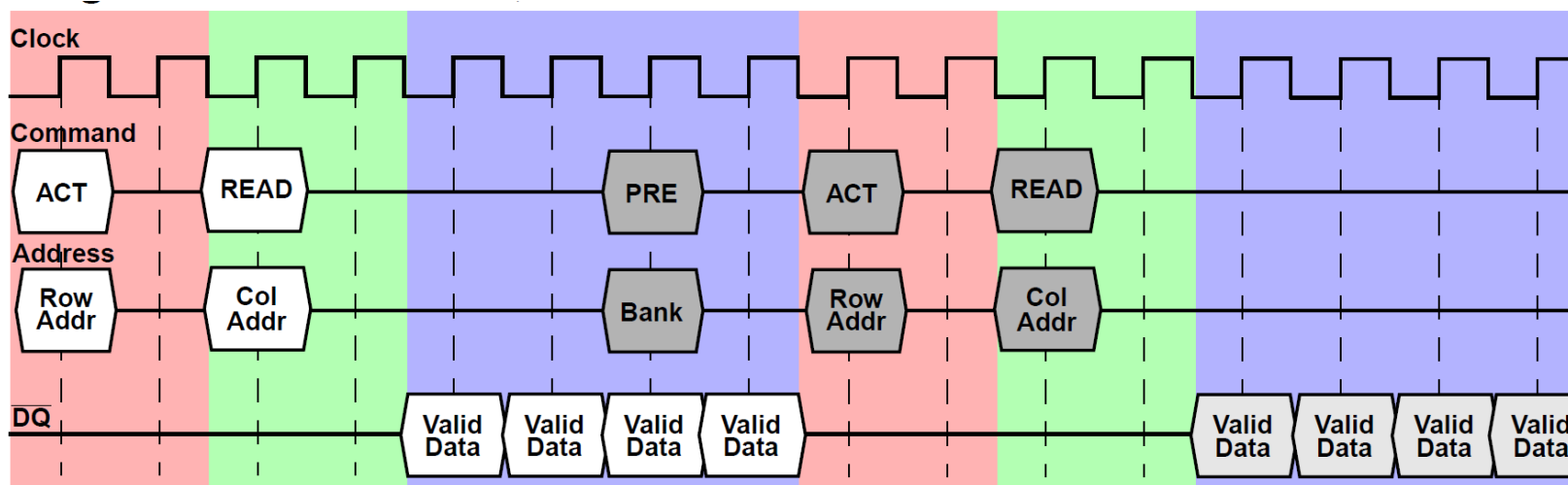
- Row Access
- Column Access
- Transfer Overlap
- Data Transfer



动态随机存储器——时序改进



引入时钟，命令，数据和控制的并行处理，数据簇发。



动态随机存储器——控制器

- DRAM Controller: 动态随机存储器控制器
 - 由于DRAM的时序复杂，DRAM一般需要一个专用控制器
 - 专用控制器一边是DRAM的接口
 - 另一边是总线接口（PCIe, AMBA）
- 控制器的功能
 - 从总线上接收内存读写指令
 - 在内部调度多个并发的读写指令以获得最短平均延时
 - 按照DRAM的时序向DRAM发出细节的请求
 - 定时调度DRAM的刷新
 - 在上电时初始化DRAM的配置

动态随机存储器总结

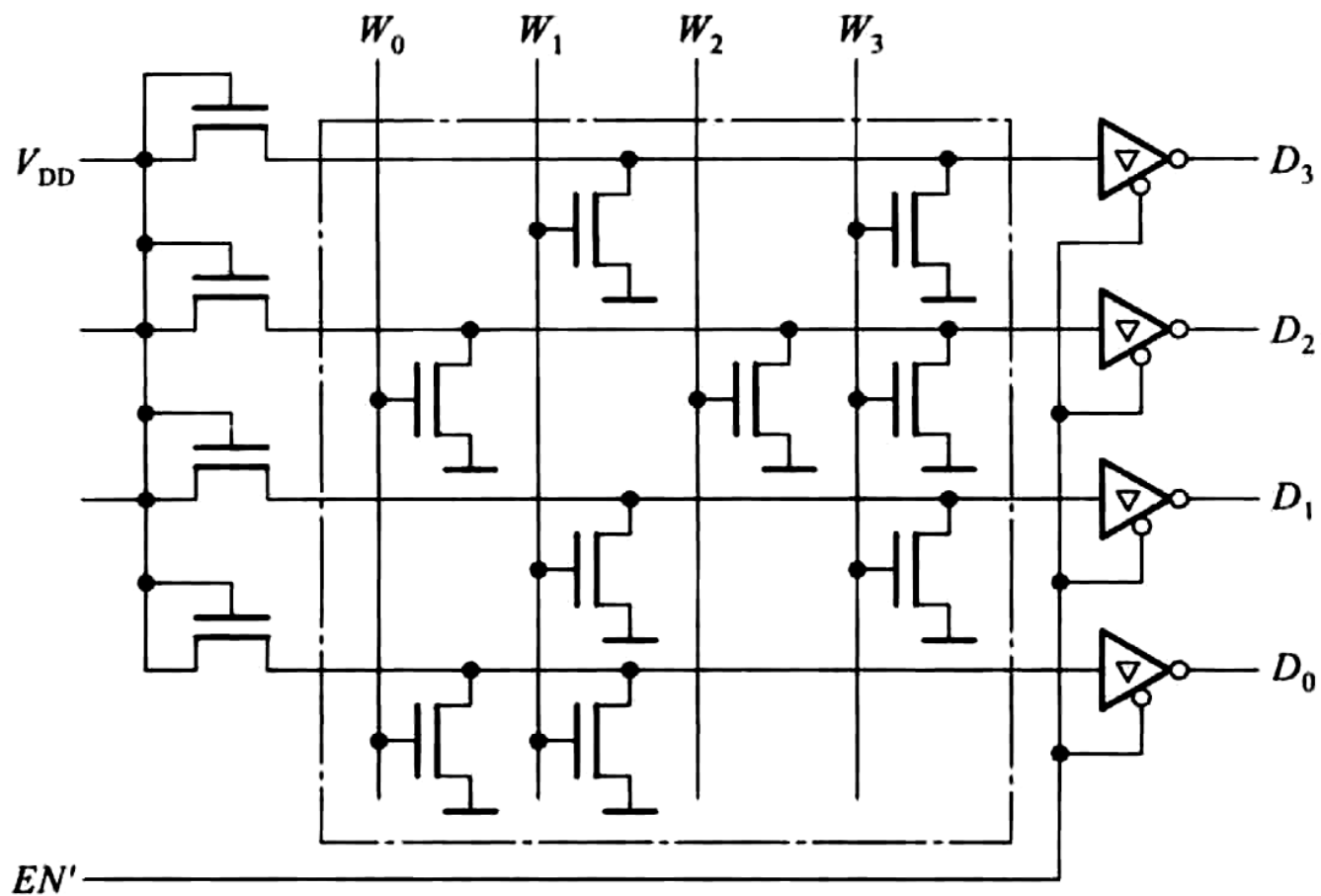
○ DRAM

- 动态随机存储使用电容存储状态
- 相比6T的SRAM，具有更高的存储密度

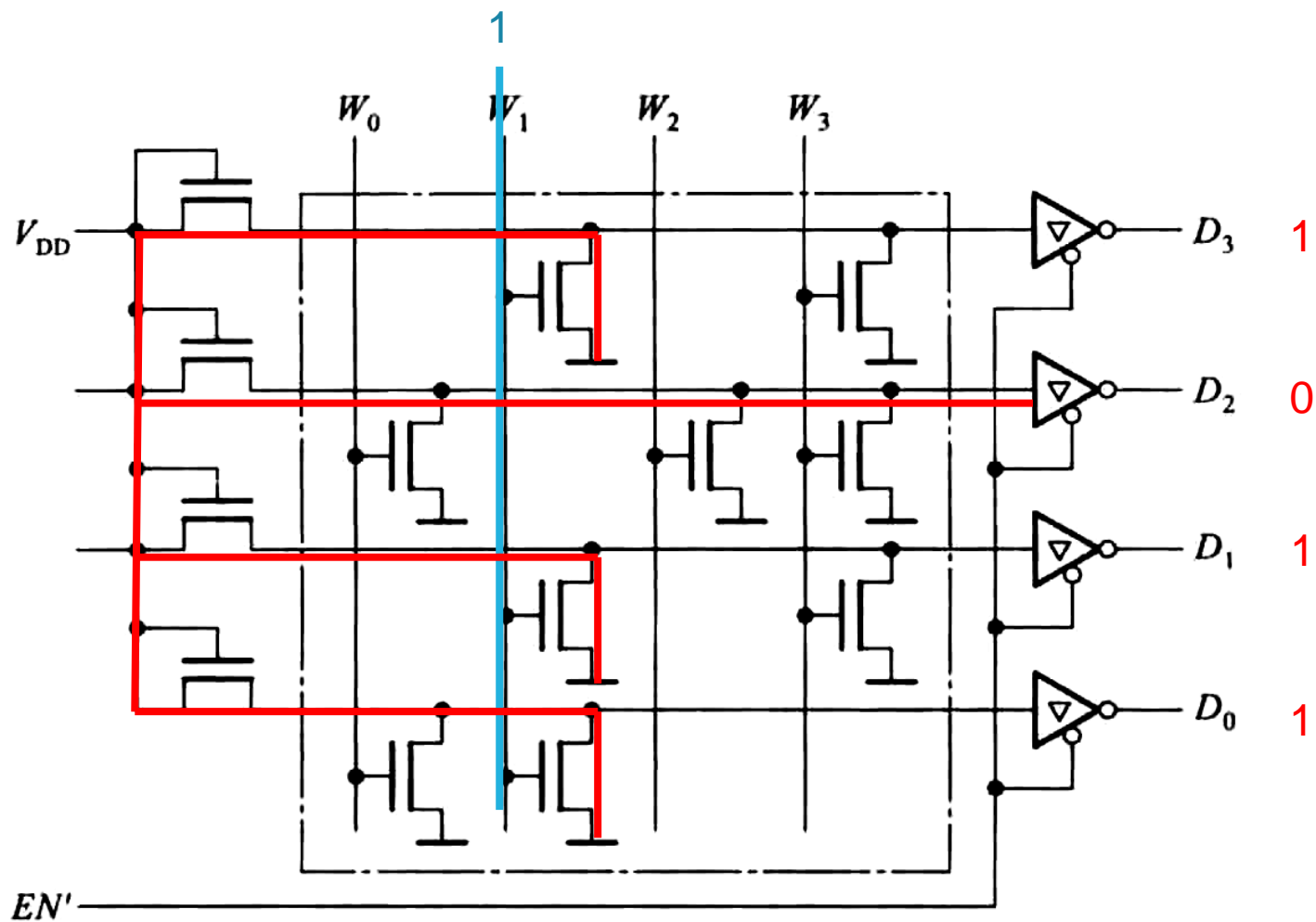
- 对电容的状态读取可能导致状态的丢失
- 电容本身的漏电也会导致状态丢失
- 需要定期刷新

- 时序复杂
- 需要专门的控制器调度访问次序和保证时序

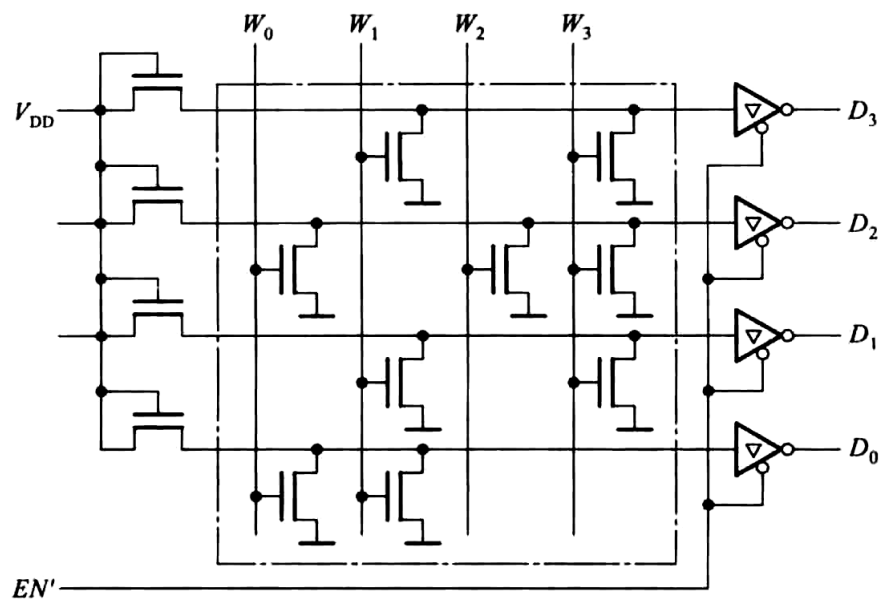
MOS管构成的只读存储器ROM



MOS管构成的只读存储器ROM



用ROM构成任意逻辑



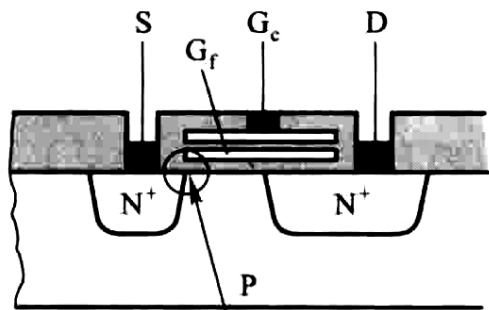
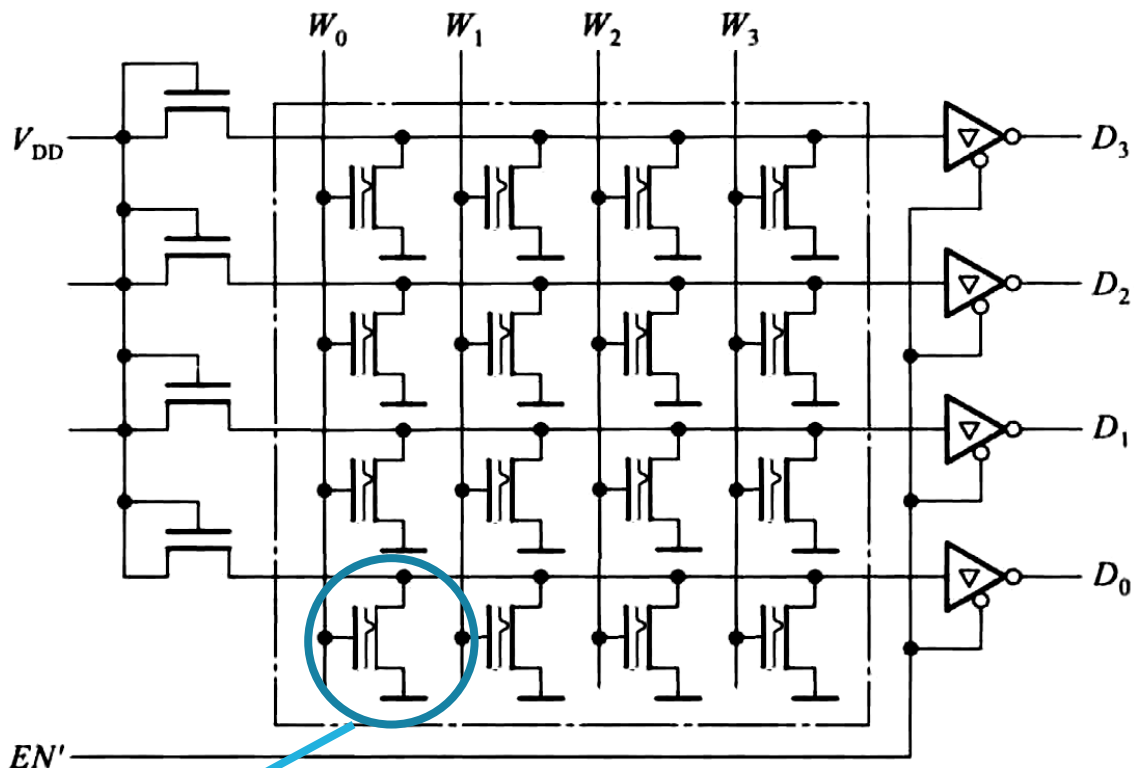
$$D = F(A)$$



地址		数 据			
A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	1	1	1	1	0

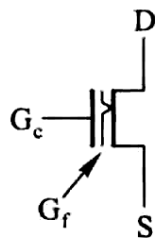
E²PROM: 可擦写ROM

当 G_f 充有负电荷:
 G_e 为高电平时管子不导通。
 当 G_f 没有电荷时:
 G_e 为高电平时管子导通。



隧道区

浮栅MOS管

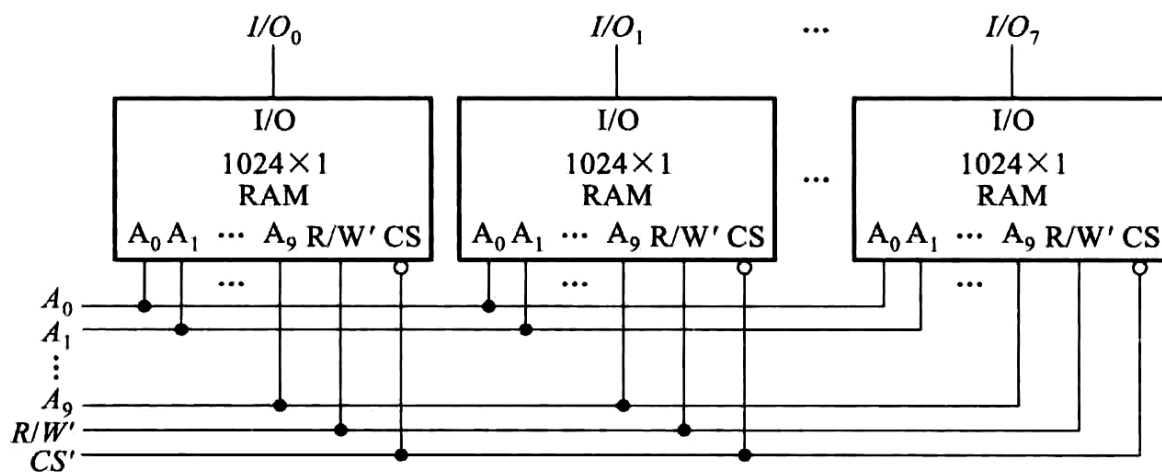


关于 G_f 的充放电过程, 自学教材第五章, 241页。

- 不需要擦写的ROM
 - 直接由MOS管构成的存储矩阵
 - 面积小，存储密度很高
- 需要擦写的ROM (PROM)
 - E²PROM：利用浮栅MOS管构成的存储矩阵
 - 通过给浮栅极充电来存储信息
 - 充放电次数有限
 - Flash闪存
- 内存其实可以用来构成任意逻辑
 - 地址为输入
 - 读取的数据为输出

内存的扩展方式——按位扩展

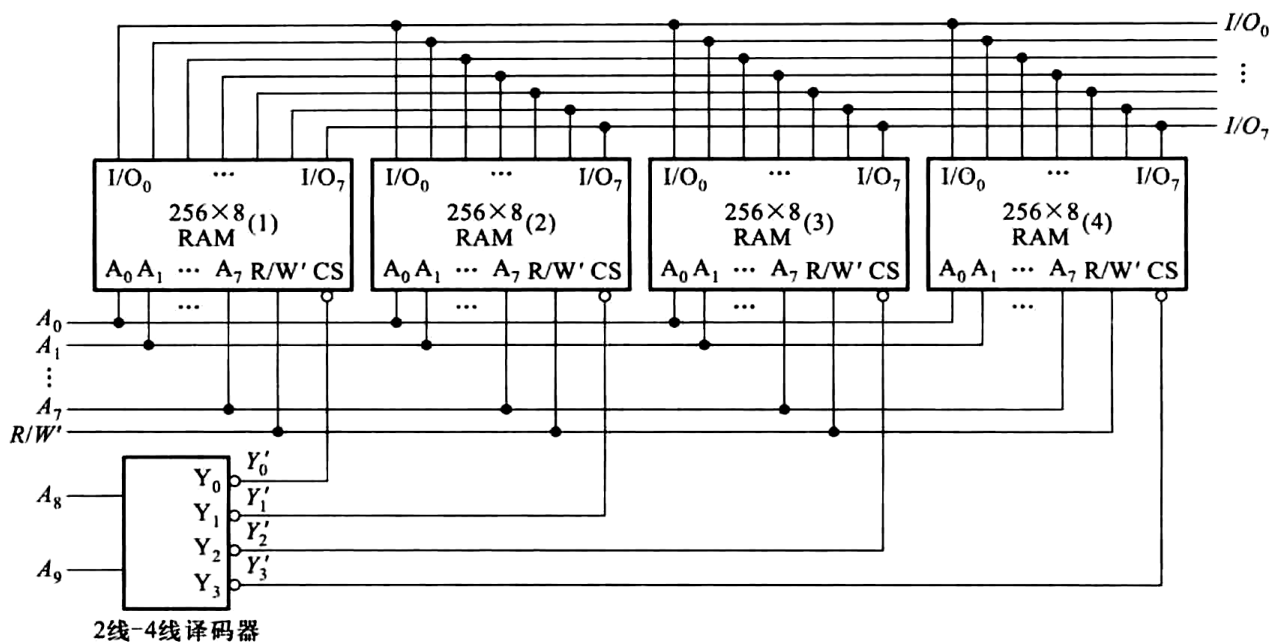
- 单个内存模块的位宽为 W ，目标位宽为 M ， $M > W$



8个1024x1的内存模块构成一个1024x8的内存模块

内存的扩展方式——按字扩展

- 单个内存模块的地址宽度为A，目标地址宽度为B， $B > A$



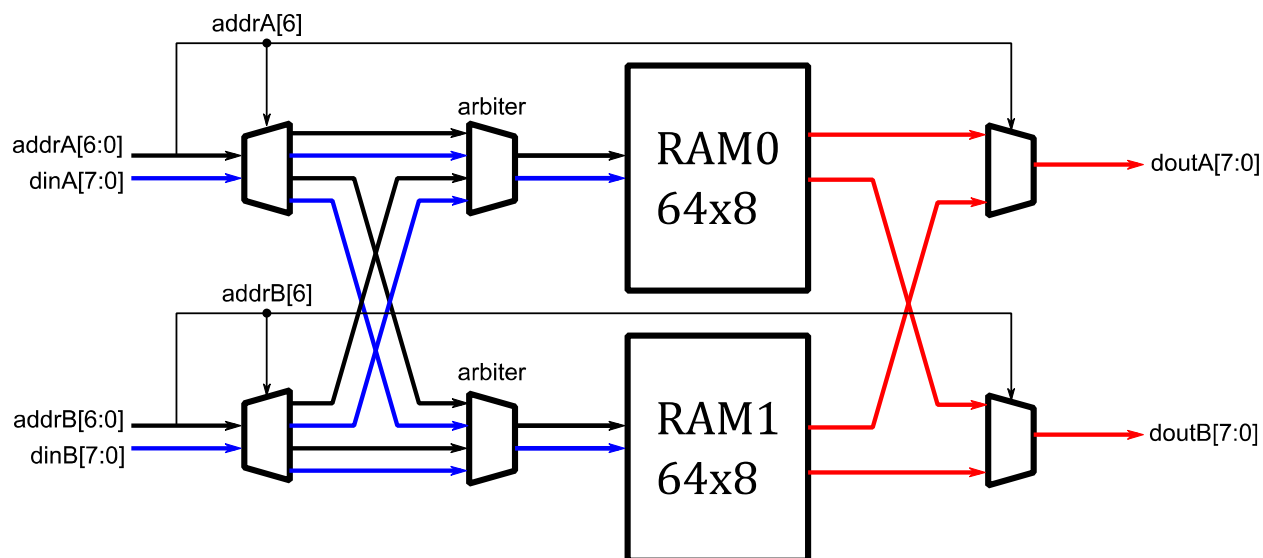
4个 256×8 的内存模块构成一个 1024×8 的内存模块

256×8 ，地址宽度为8位

1024×8 ，地址宽度为10位

内存的扩展方式——阔端口数

○ 扩展内存端口数量



将一个128x8的RAM拆成两个64x8的RAM bank，由地址addr[6]确定bank的选取。

当端口A和B访问的地址 $addrA[6] \neq addrB[6]$ 的时候，两个端口可以同时读写内存。

总结：考试范围（概念）

- **各种内存模块**
 - **静态存储器 SRAM**
 - **动态存储器 DRAM**
 - 需要刷新
 - 时序复杂，需要控制器
 - **只读存储器 ROM**
 - 使用ROM构造任意逻辑电路
 - **可擦写只读存储器 PROM**

- **内存模块的扩展**
 - **按位扩展**
 - **按字扩展**
 - **扩展端口**

任何问题?

课堂习题 (提问方式)

- 假设一个系统工作在50MHz，系统中有多个模块需要同时访问同一个内存模块，该内存模块可以工作在100MHz。

请问：有什么方法可以让该内存模块高效地被系统使用？