

## 12. 半导体存储器



[xgsun@fudan.edu.cn](mailto:xgsun@fudan.edu.cn)

孙晓光

2024-11-17



1

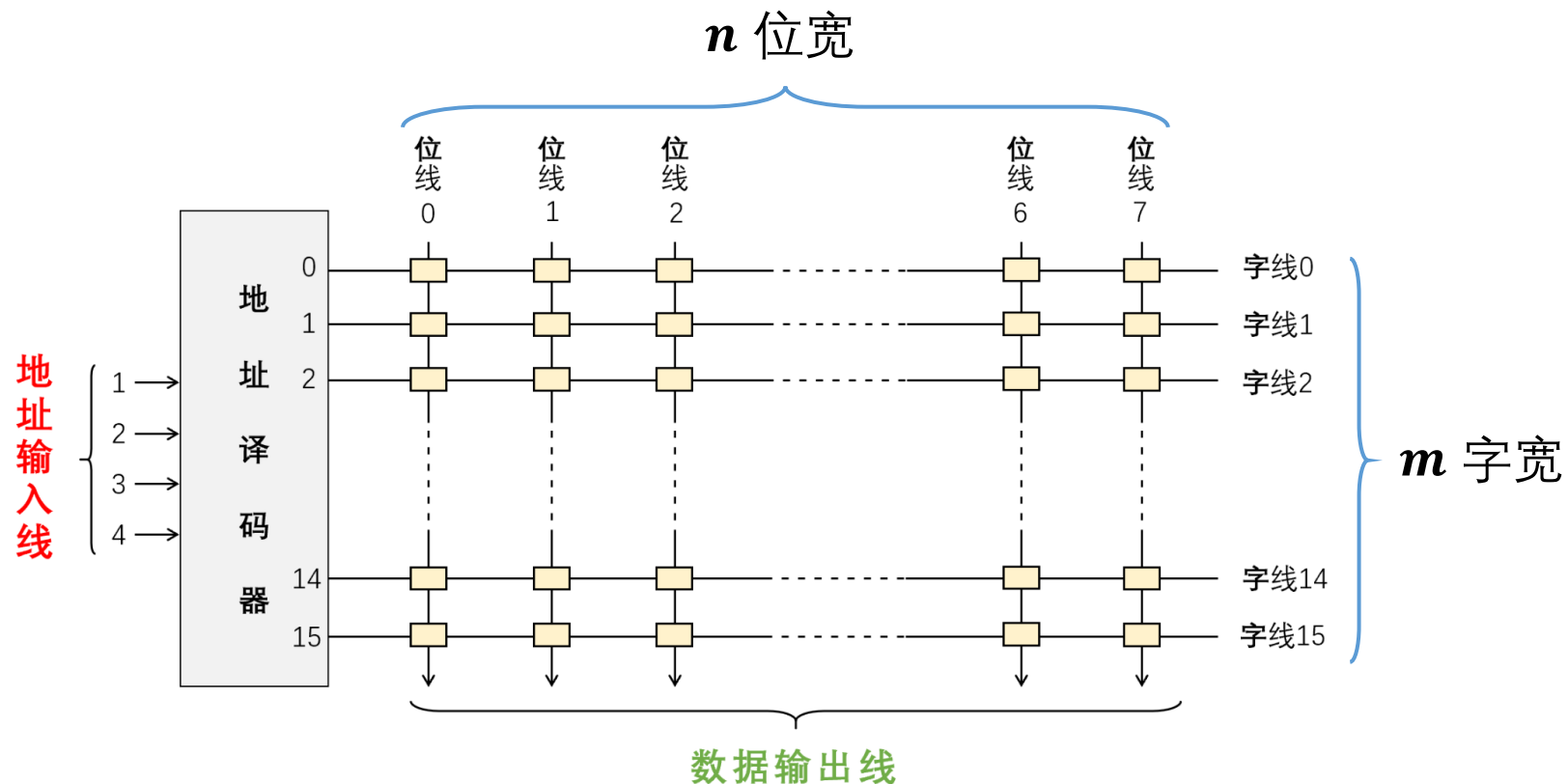
# 简介

# register 寄存器、存储器 memory

- **寄存器**：由锁存器或触发器组成，一次能够并行存储  $n$  位 比特数据的逻辑部件。  
(一维)

字宽 位宽 (二维)

- **存储器**：能够存储  $m \times n$  个二进制比特数的逻辑电路。



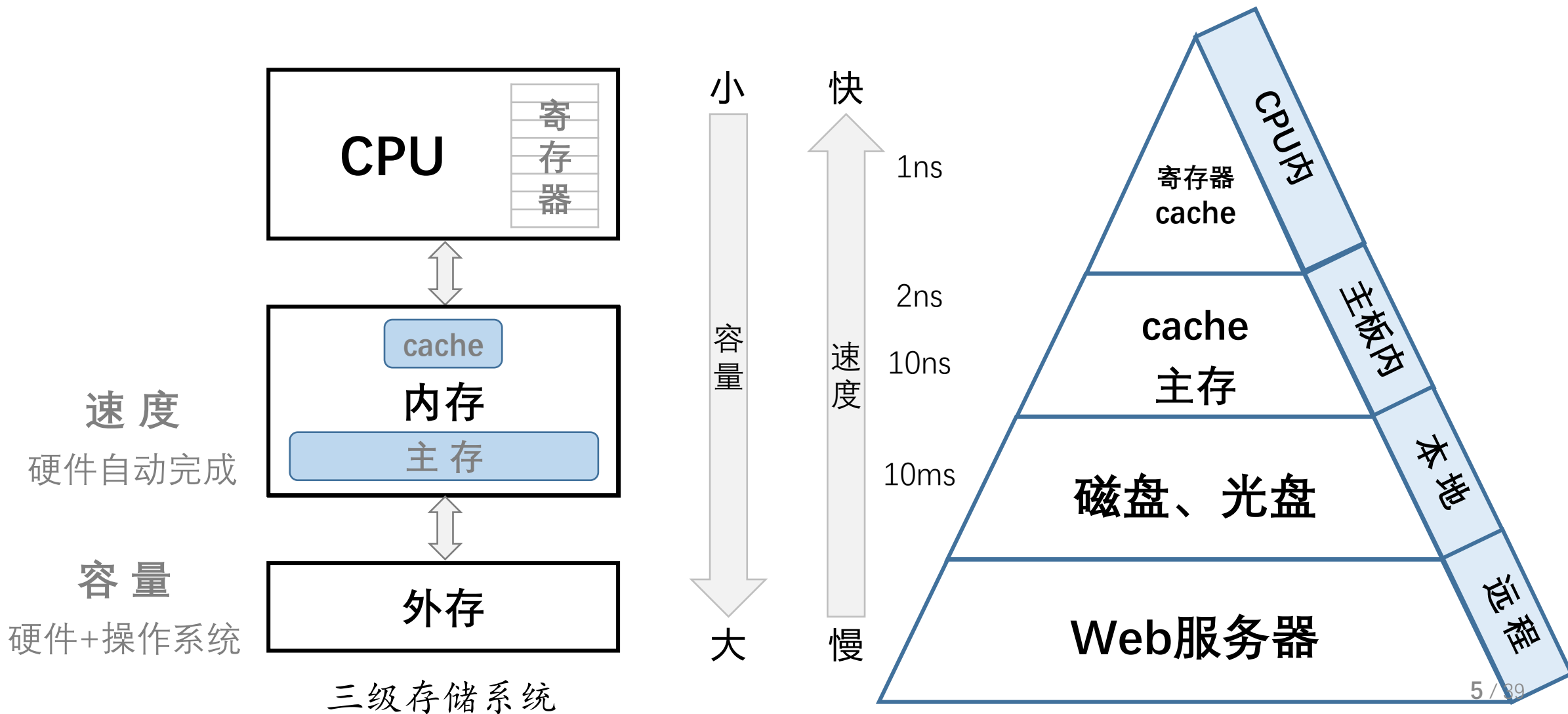
# 存储器的分类



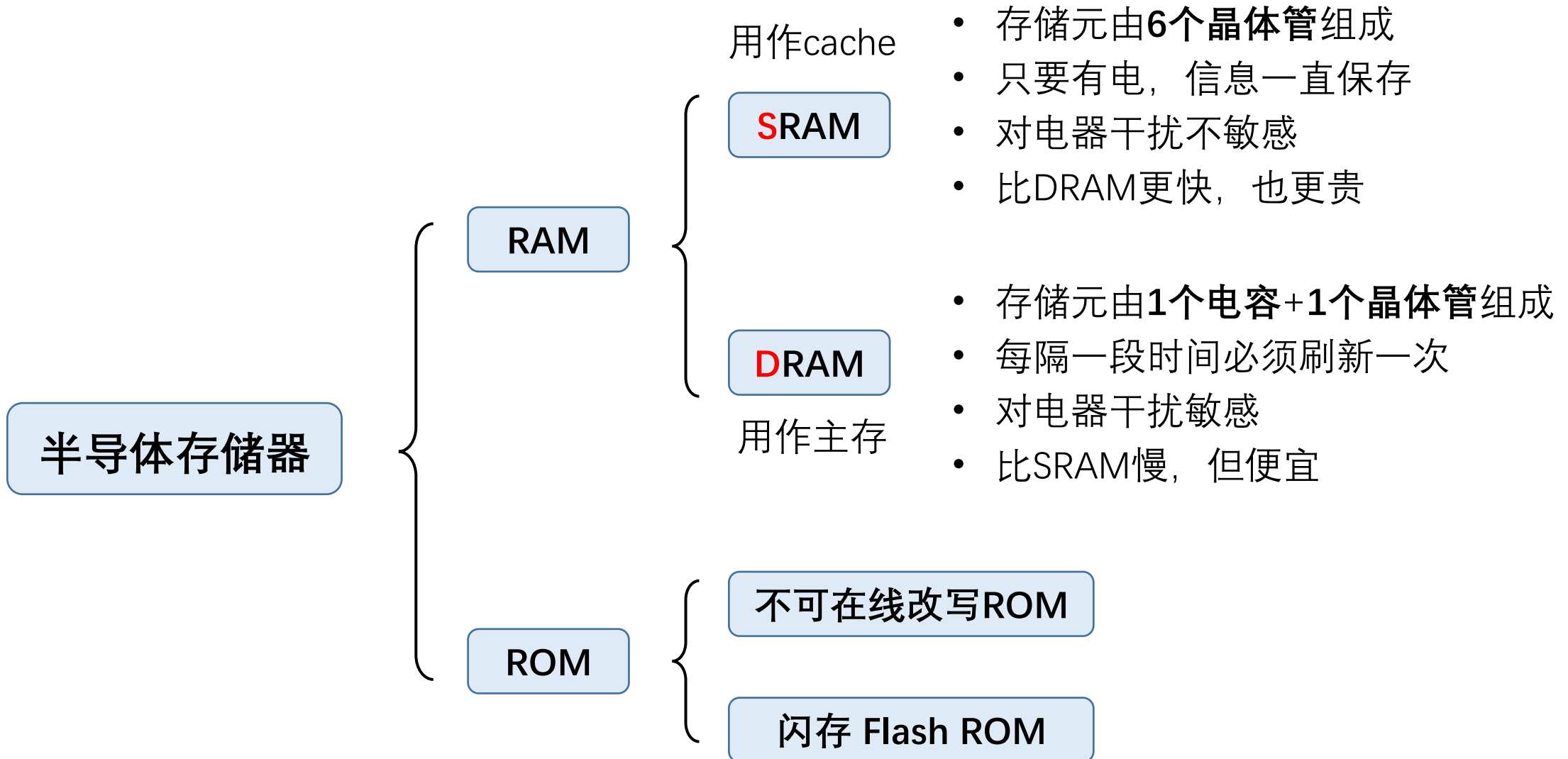
存储介质	半导体器件	TTL、MOS
	磁性材料	磁盘、磁带
	光存储器	光盘
存取方式	顺序存取	磁带
	随机存取	ROM、RAM。存取时间与存储单元物理位置无关
	直接存取	磁盘。沿磁道方向顺序读取，垂直半径方向随机存取
读写功能	只读	ROM (Read-Only Memory)
	读写	RAM (Random Access Memory)
信息易失性	易失	RAM。断电后信息消失
	非易失	ROM、硬盘、光盘、U盘…。 断电后仍能保存信息
与CPU耦合程度	内部存储器	cache、主存
	外部存储器	硬盘、光盘 ...

# 多级存储系统

一个具有不同容量、成本和访问时间的存储设备的层次结构。



# 半导体存储器的分类



# 存储器的技术指标

## 容量

- 存储容量：存储器中可存储的信息比特数 (bit / Byte)。

存储字数 × 存储字长

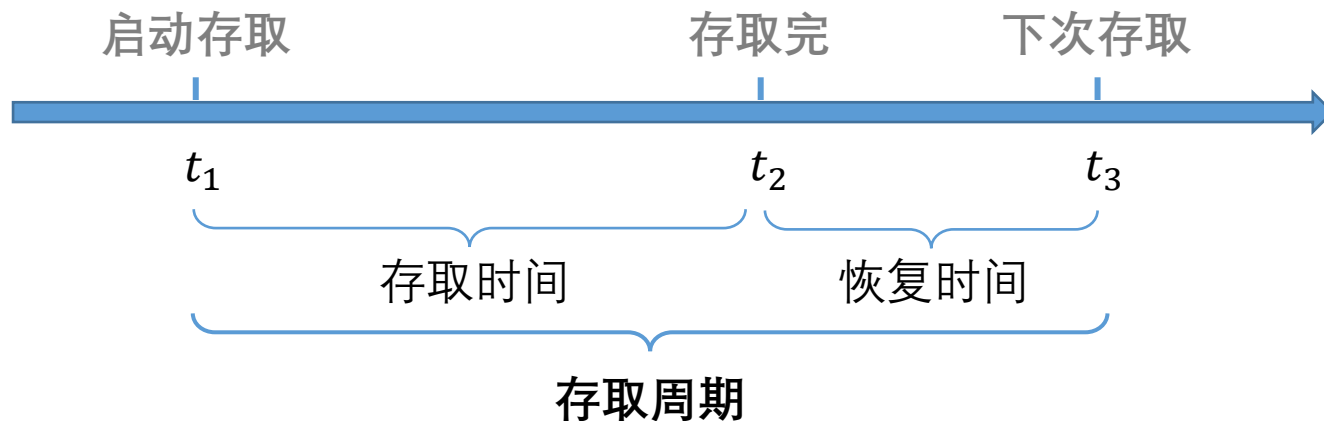
如：1024 x 8 bit

(房间数) (每个房间的人数)



## 速度

- 存取时间 / 访问时间：从存储器接收到读/写命令开始，  
到信息被读出或写入完成所需的时间。
- 存取周期 / 存储周期：在存储器连续读写过程中，一次完整的存取操作所需的时间。



2

ROM



# 只读存储器 ROM

## 掩膜 ROM

由厂家一次写入，无法修改

## 一次性编程ROM (PROM)

由用户用专门的设备一次性写入，之后无法修改

## 紫外线擦除PROM (EPROM)

用紫外线擦除，修改次数有限，写入时间长



## 电可擦除 PROM (EEPROM)

用电擦除，速度快，一般可以擦除1百万次左右

## 闪存 (Flash memory)

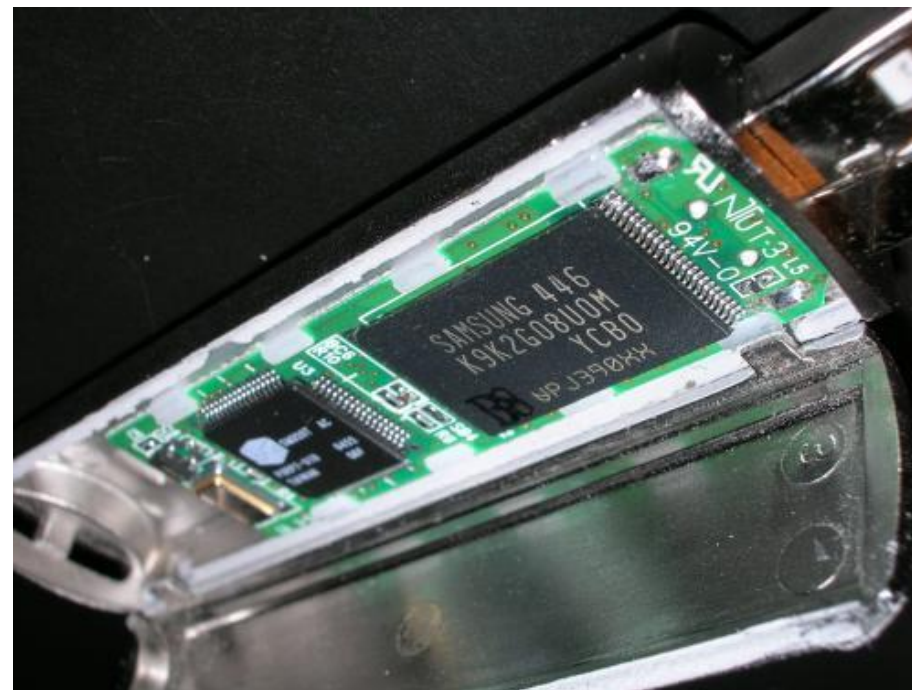
高密度、非易失。如 U盘

## 固态硬盘 (SSD)

闪存芯片 + 控制单元

# FLASH 存储器 (闪存)

- 高密度非易失性的读/写存储器。
- 既有RAM优点，又有ROM优点。
- FLASH存储元由单个**MOS晶体管**组成。
- FLASH三种基本操作：
  - 读出操作
  - 写入操作(编程)
  - 擦除操作



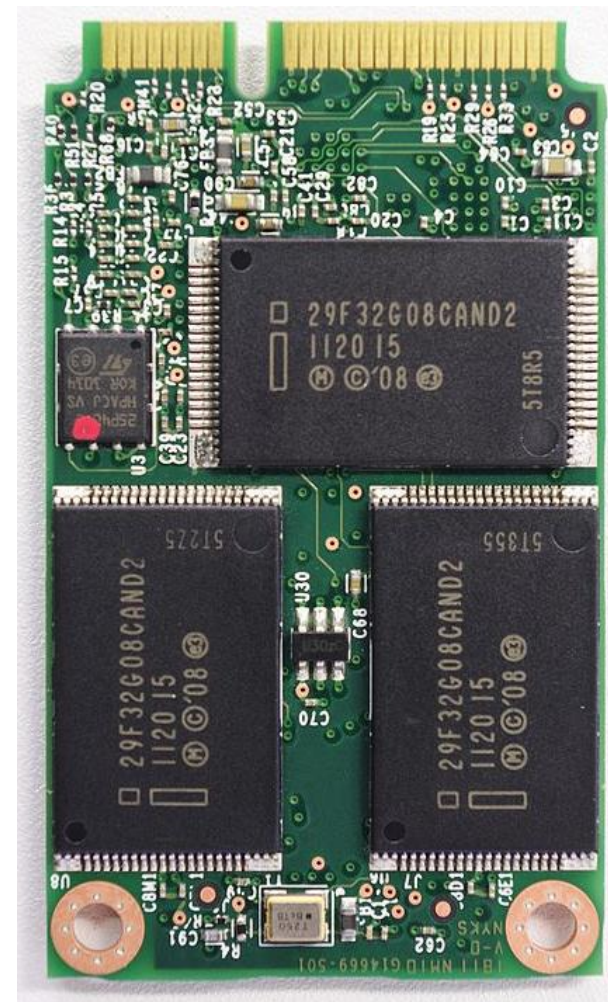
用于数据处理系统的快闪电子式外存储方法及其装置

- 1999年朗科 Netac 提交“**U盘**”专利 ~2019到期
- 核心技术: **FLASH** (1988 Intel) + **USB** (1996)
- 躺着挣钱: 每年专利获利超千万

## 与机械硬盘相比

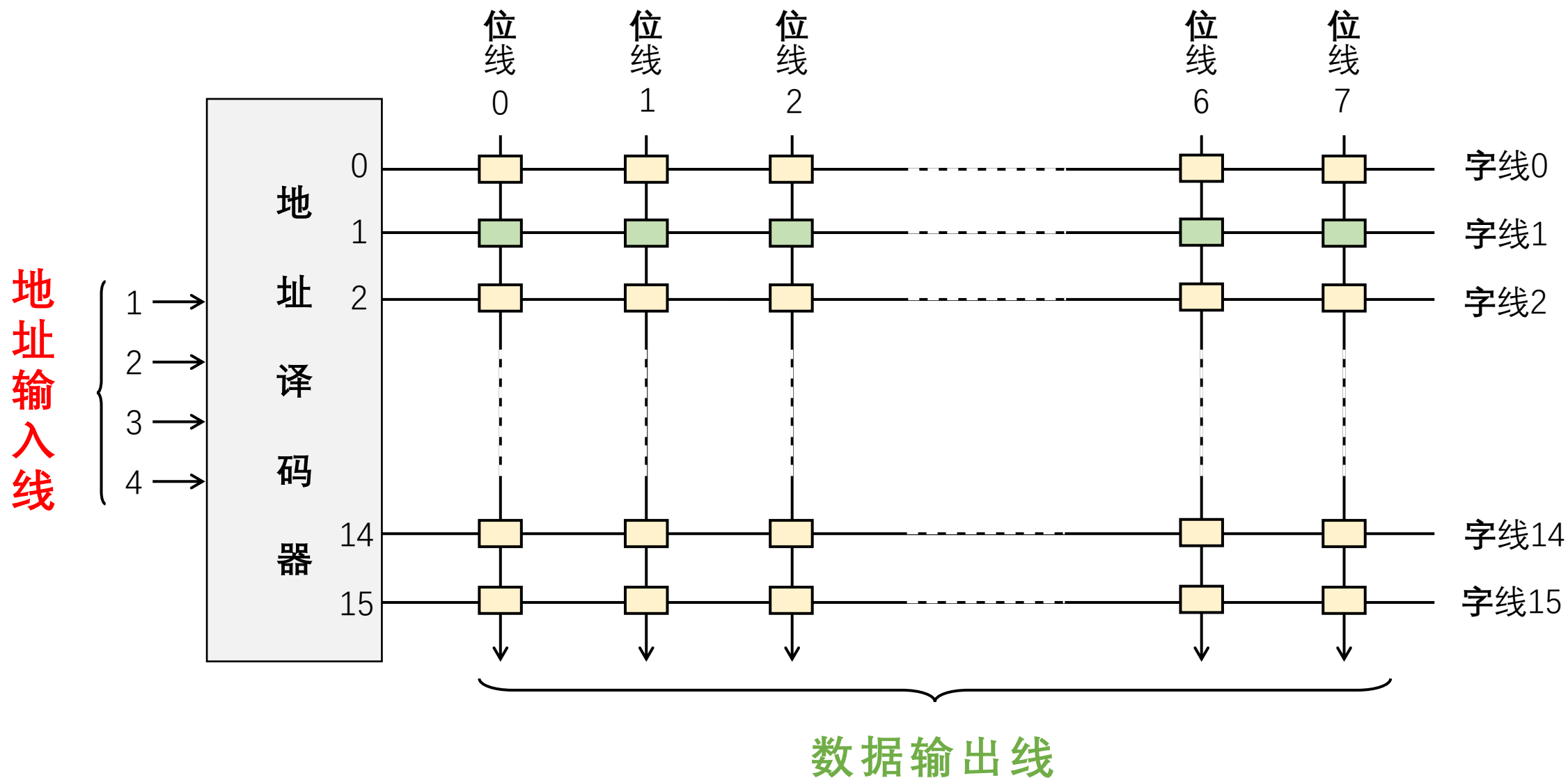
**优点：** 没有移动部件，速度快、能耗低、更结实。

**缺点：** 价格贵、损坏后难以修复、写入次数有限、  
静置时数据消失。



mini PCI-E SSD

# ROM 结构示意图

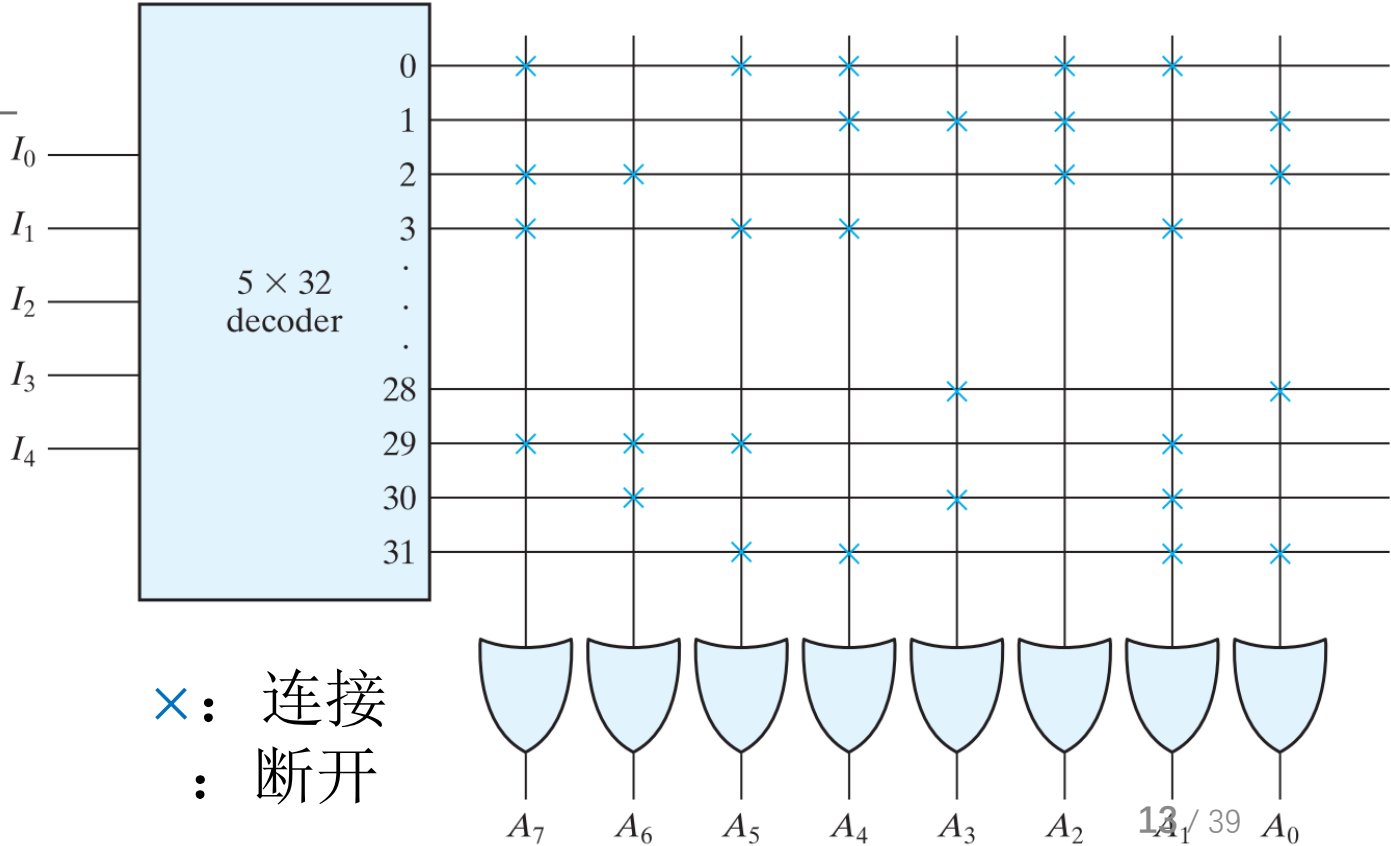


# ROM 存储的内容

ROM Truth Table (Partial)

Inputs					Outputs							
$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮						⋮				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

1: 连接, 0: 断开



# ROM设计： 电路输入是3位二进制， 输出是输入数值的平方

Inputs			Outputs						Decimal
A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

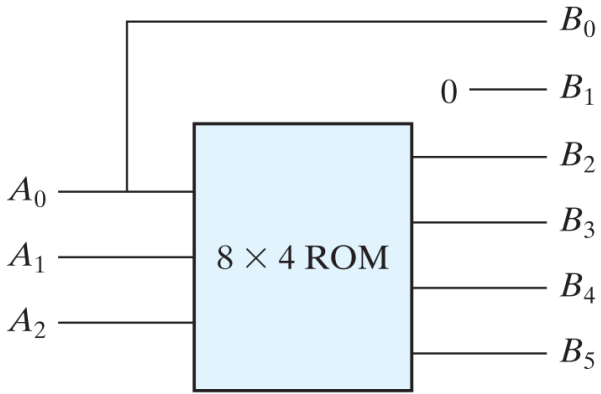


A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table

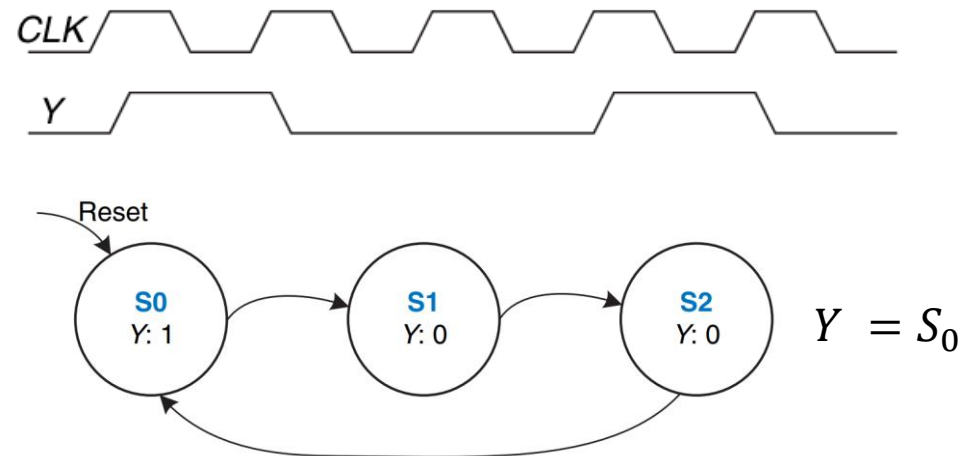
用ROM实现组合电路:

- ① 写出原始电路真值表
- ② 化简为ROM的真值表
- ③ 确定ROM容量



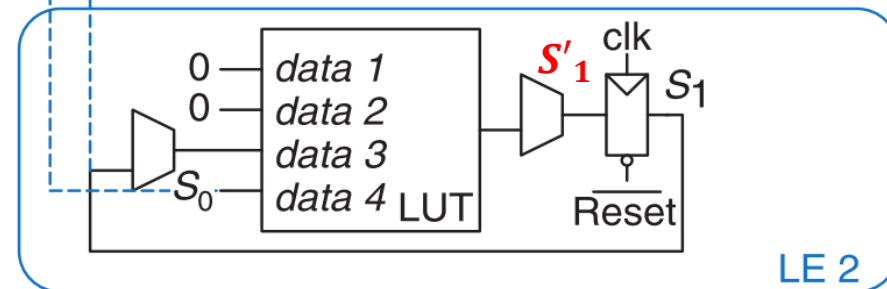
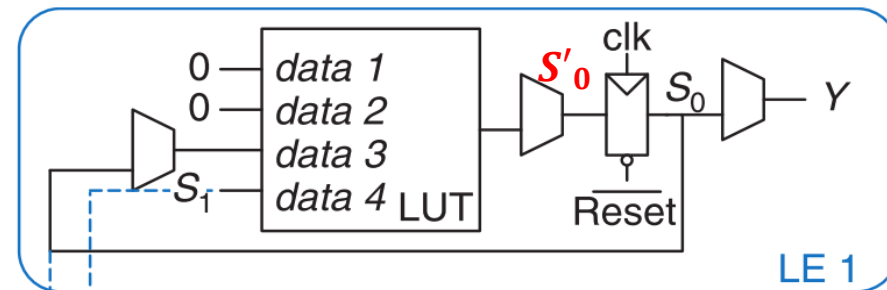
# 【例3.6】使用LE实现：3分频计数器

当前状态		下一状态	
$S_1$	$S_0$	$S'_1$	$S'_0$
0	0	0	1
0	1	1	0
1	0	0	0
1	1	0	0



		( $S_1$ )	( $S_0$ )	( $S'_1$ )
data 1	data 2	data 3	data 4	LUT output
X	X	0	0	0
X	X	0	1	1
X	X	1	0	0
X	X	1	1	0

		( $S_0$ )	( $S_1$ )	( $S'_0$ )
data 1	data 2	data 3	data 4	LUT output
X	X	0	0	1
X	X	0	1	0
X	X	1	0	0
X	X	1	1	0



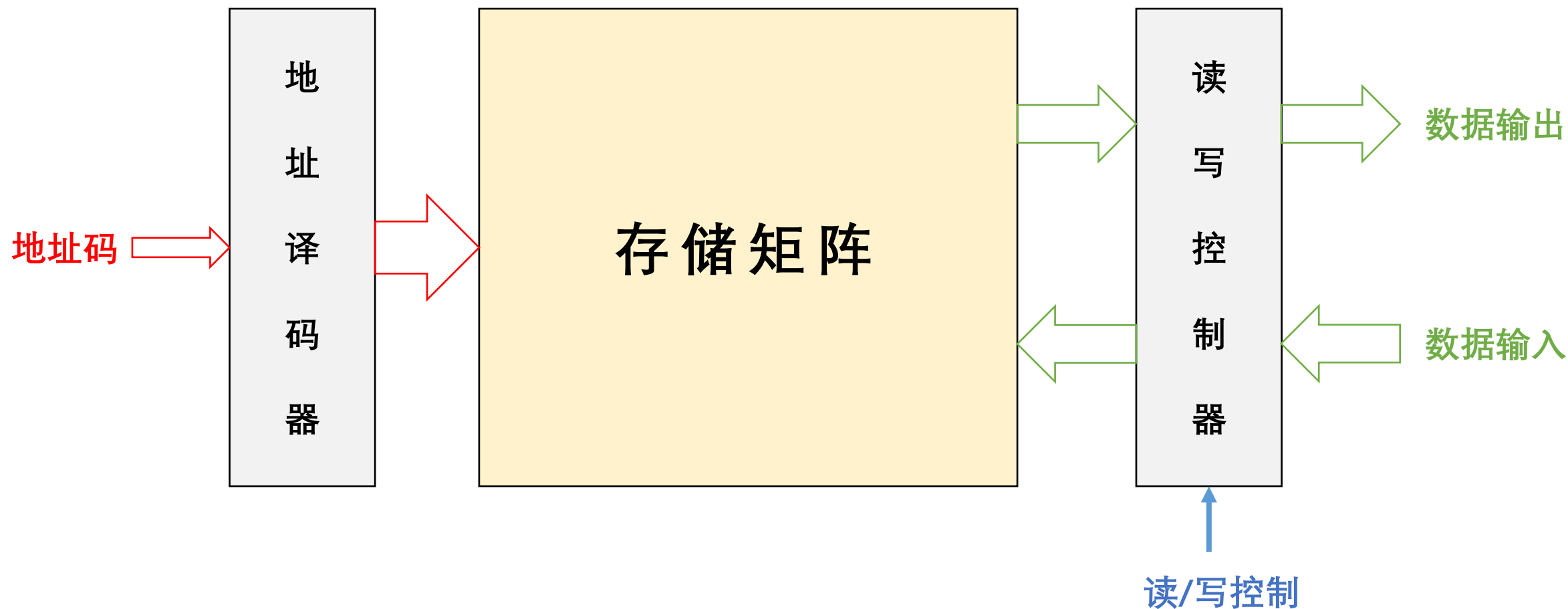
3

RAM

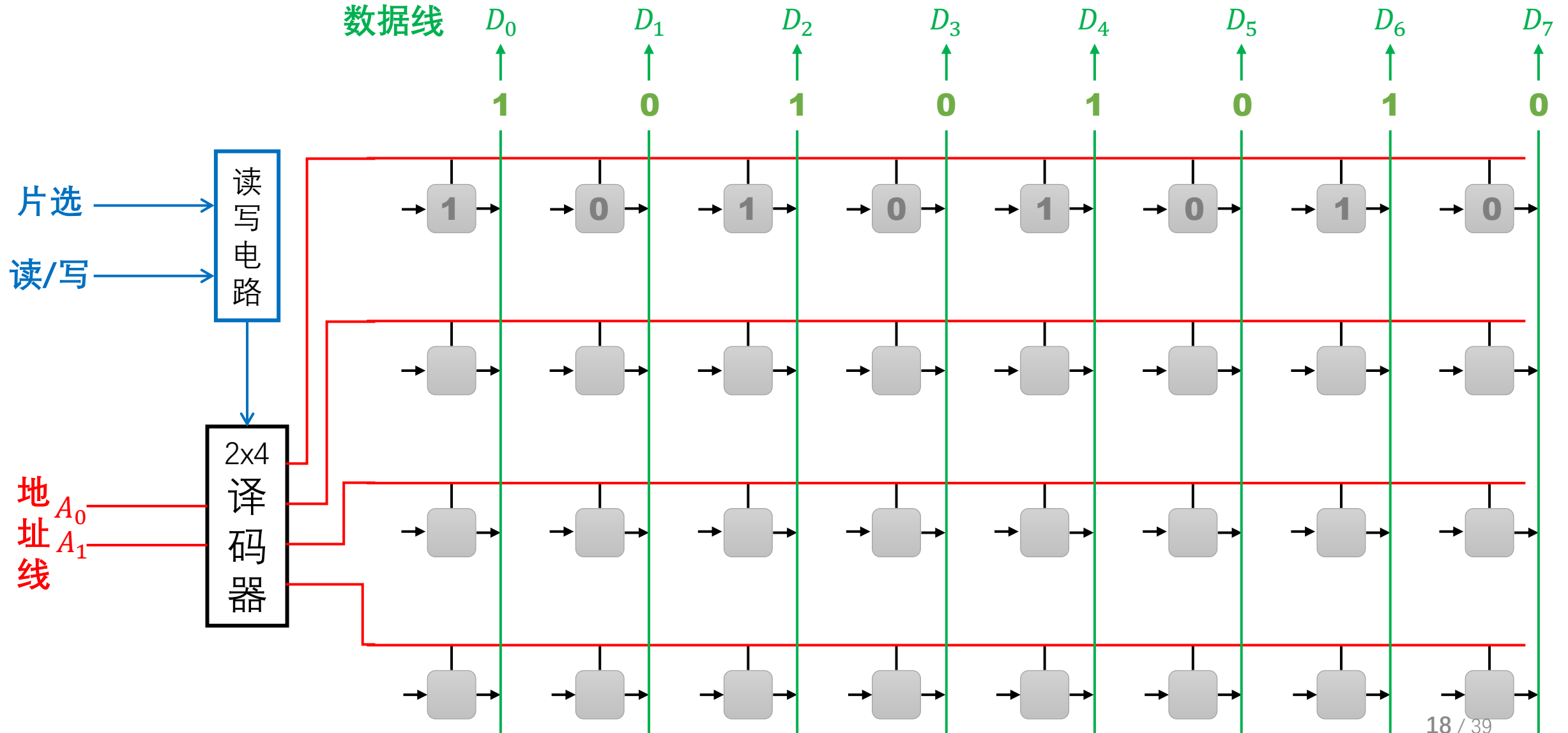


# 随机读写存储器 RAM

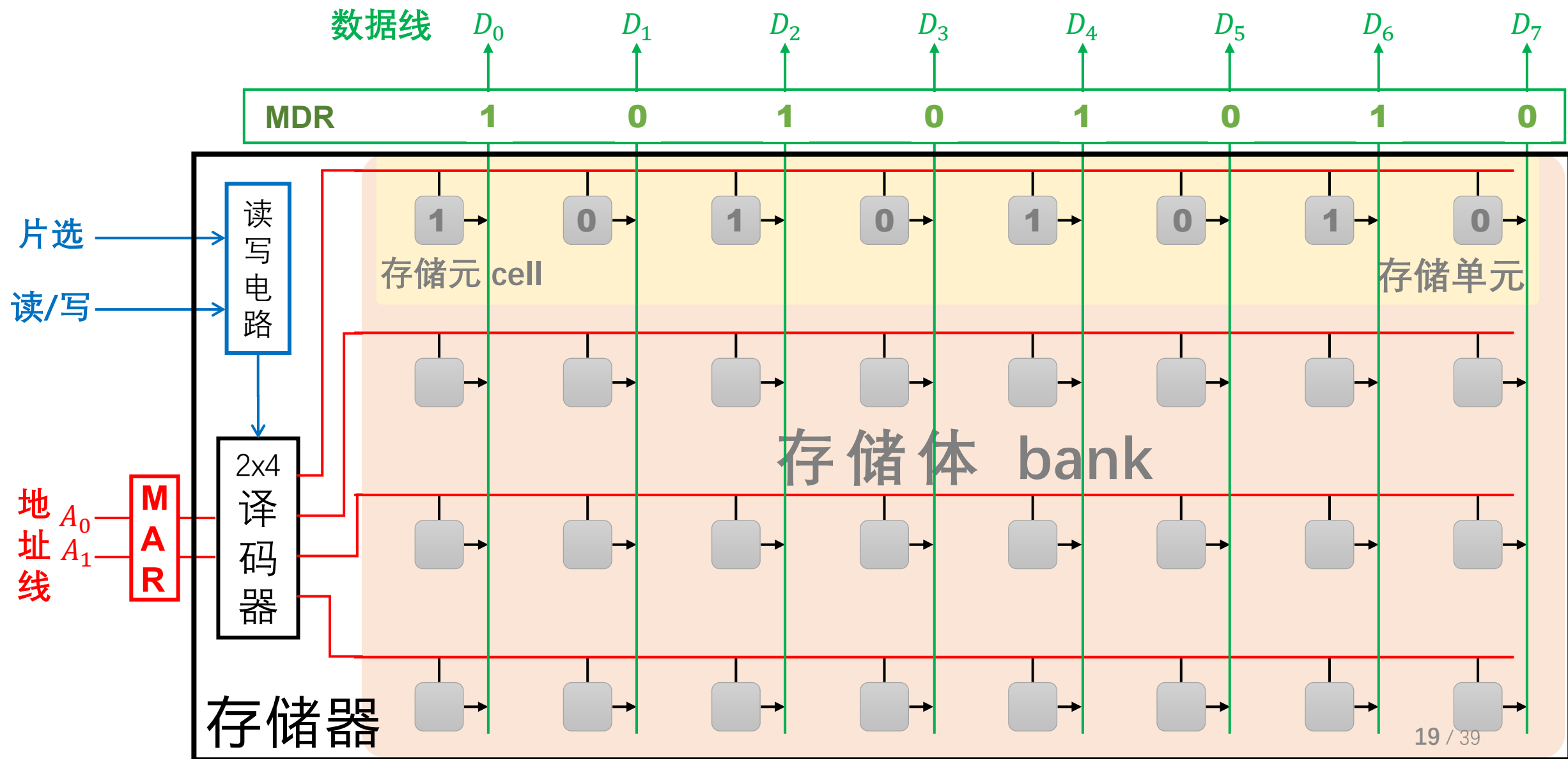
从任何单元存取信息的时间都是相同的



# 存储元阵列 (读)



# 存储器



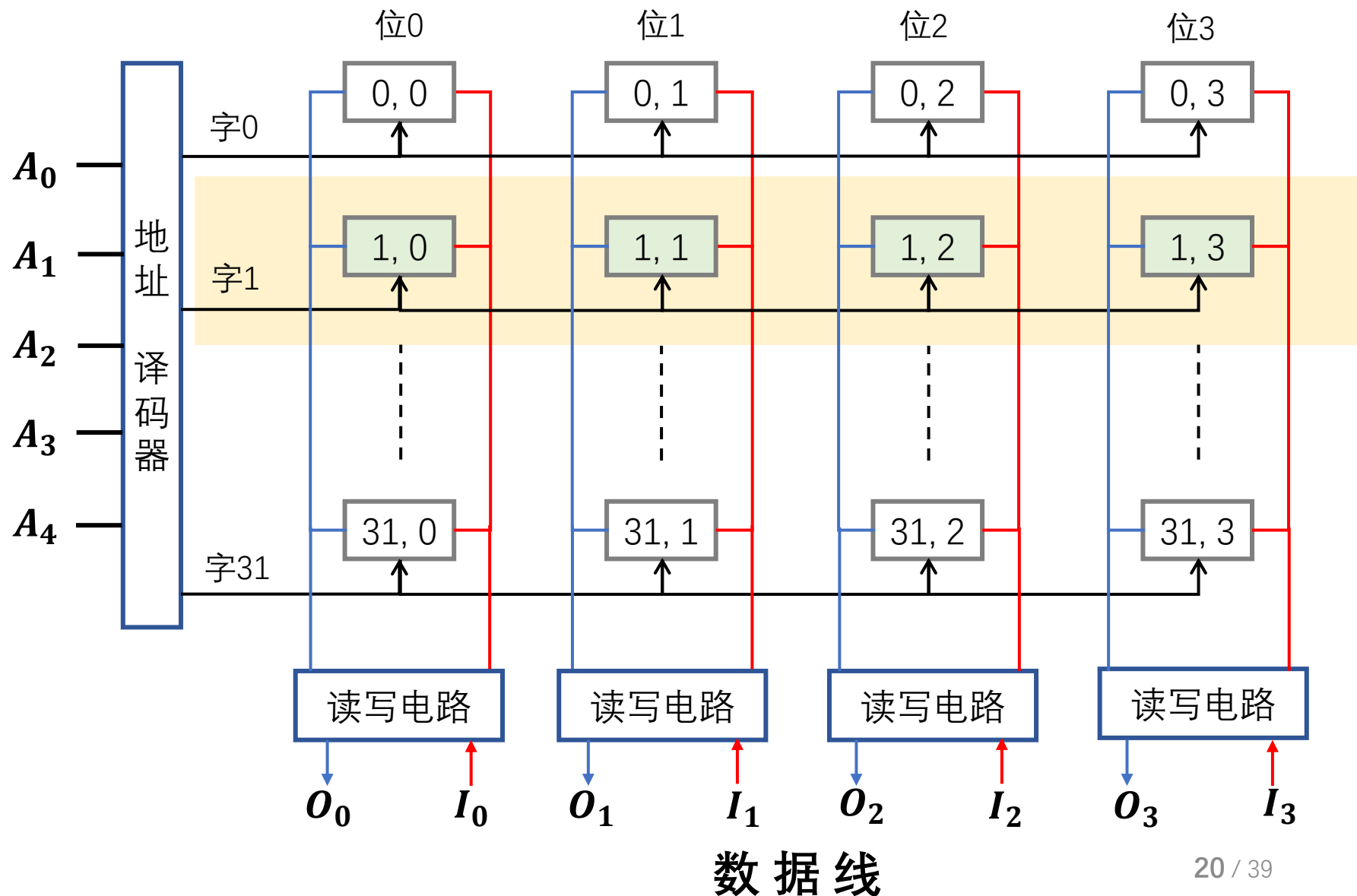
# 地址译码：单译码结构

$32 \times 4$  位

(字宽  $\times$  位宽)

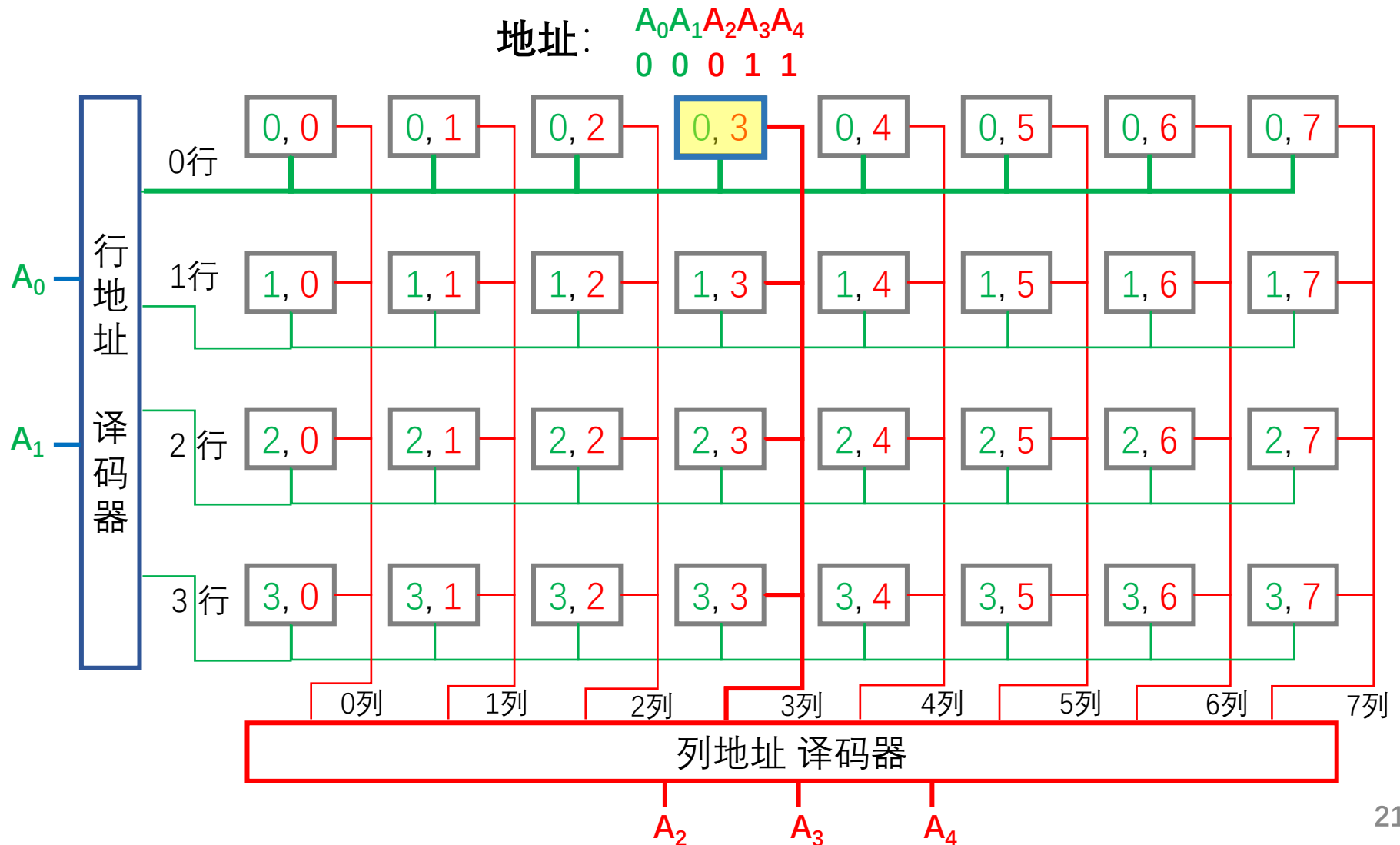
$$32 = 2^5$$

地址线



# 地址译码：双译码结构

地址复用可减少IC封装中的引脚个数



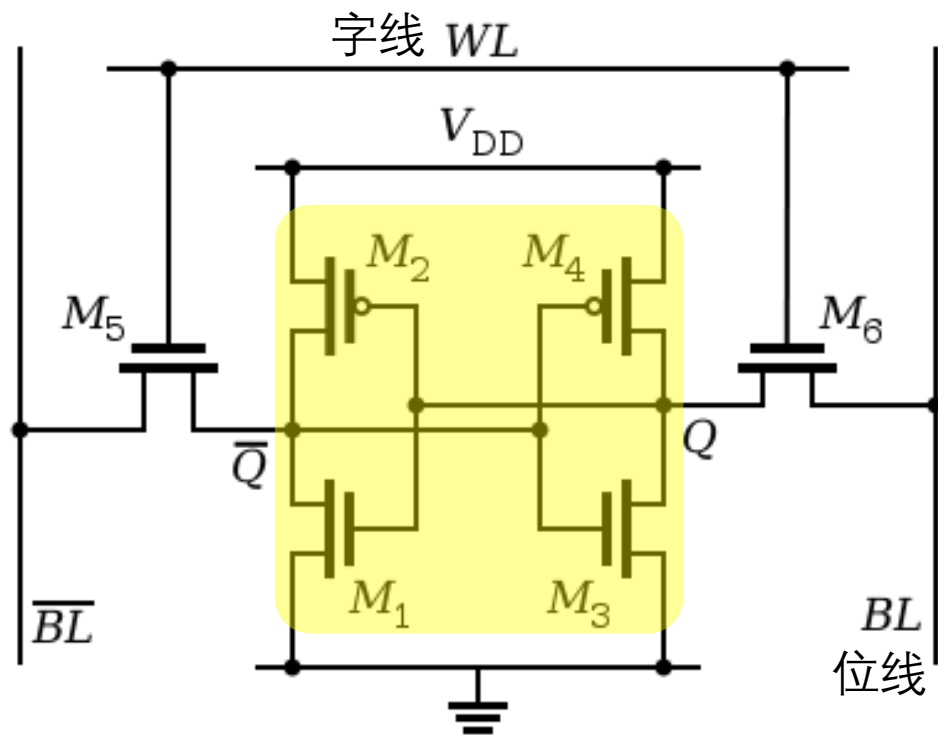
4

SRAM



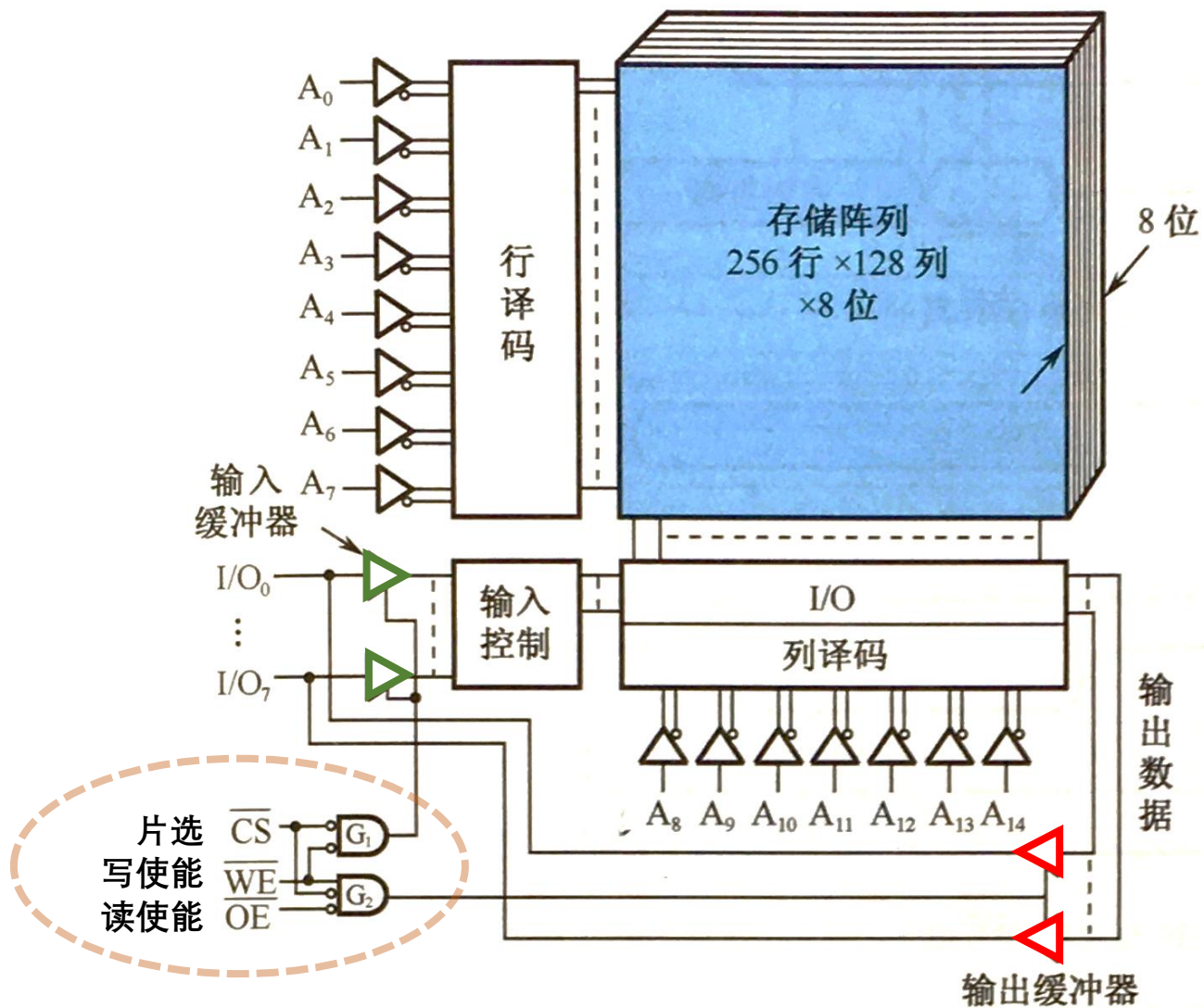
任天堂FC游戏机 (灰机)的SRAM  
2K X 8 bit

# SRAM 存储元

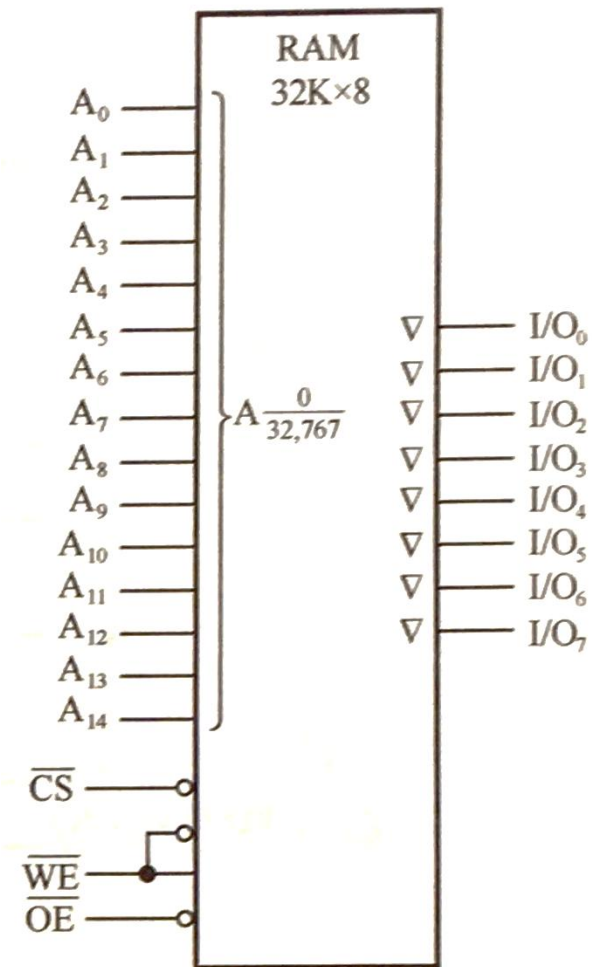


- 有0、1两个稳定状态。  
由M1, M2, M3, M4组成两个交叉耦合CMOS反相器;  
M5、M6: 两个控制开关。
- 读出操作: (假设存储元中原来保存 1)  
置WL=1, M6导通,  $BL=1$ ; M5导通,  $\overline{BL}=0$
- 写入操作: (若要写入0)  
置WL=1, 置 $BL=0$  且  $\overline{BL}=1$ , 0写入存储元中。
- 除6管的, 还有8管、10管……  
用于多端口的读写访问。

# SRAM基本逻辑结构



(a) 结构图

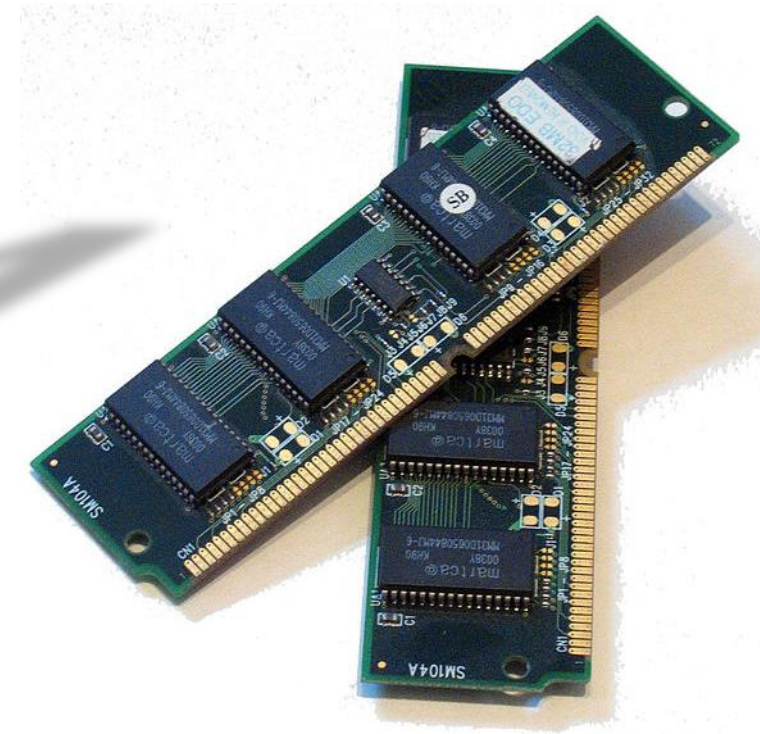


(b) 逻辑图

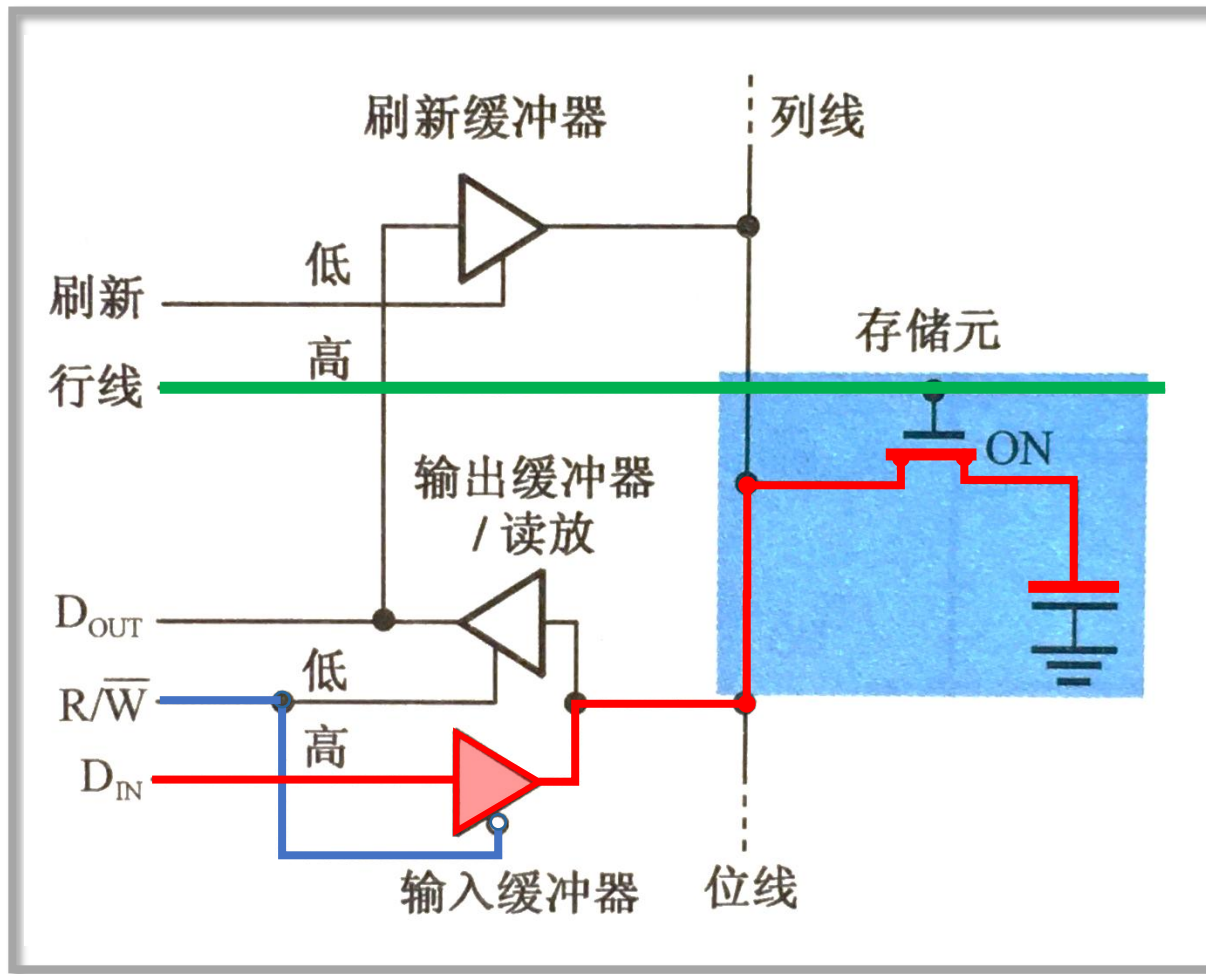


5

DRAM



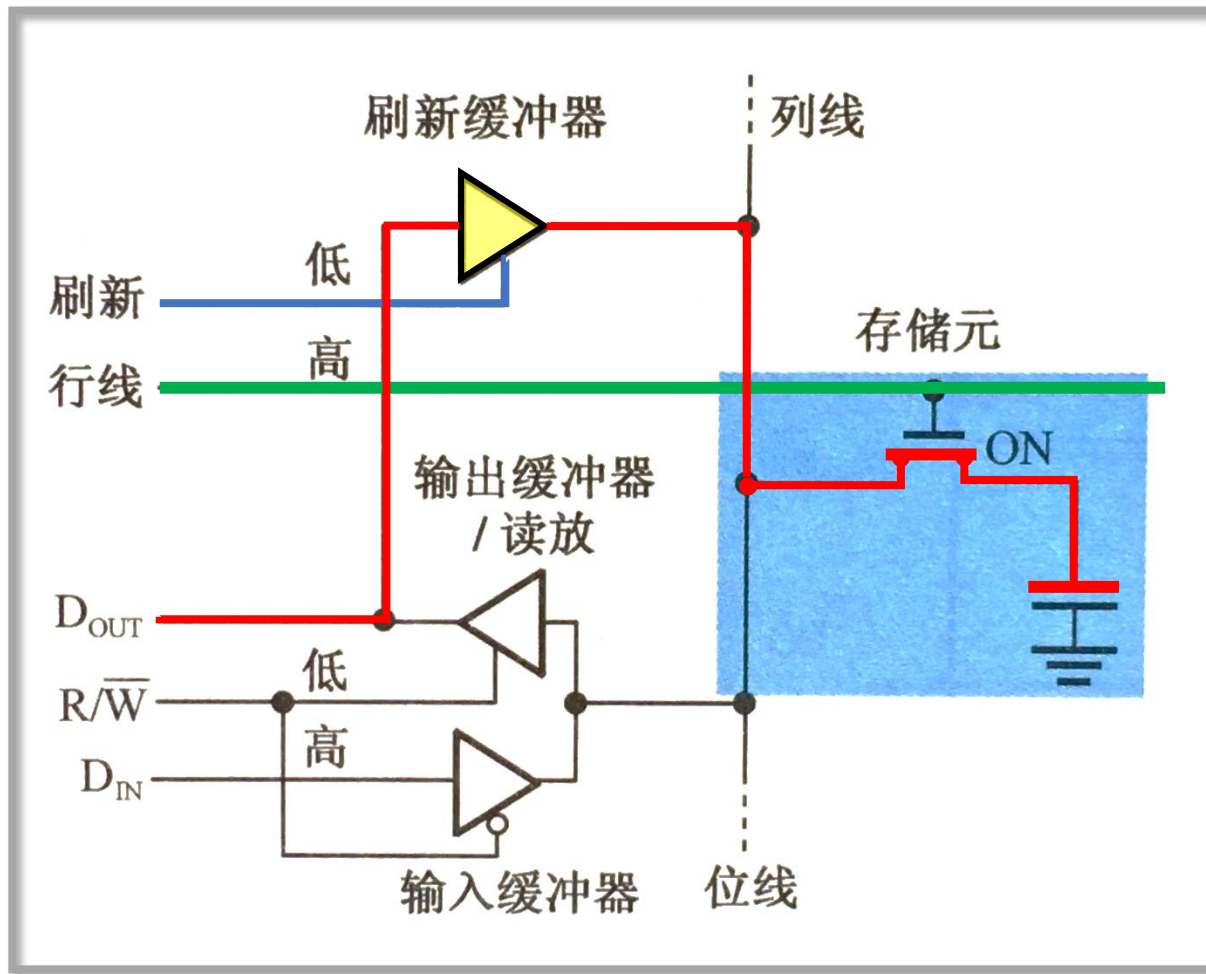
# DRAM 存储元



MOS晶体管 + 电容器 组成的单管记忆电路  
(开关) (电量高=1)  
(电量低=0)

- **写1**: 仅输入缓冲器打开, 行线导通CMOS,  $D_{in}=1$ , 给电容器充电。
- **写0**: 仅输入缓冲器打开, 行线导通CMOS,  $D_{in}=0$ , 电容器放电。
- **读1**: 仅输出缓冲器打开, 行线导通CMOS,  $D_{out}=1$ 。电容电量减少。

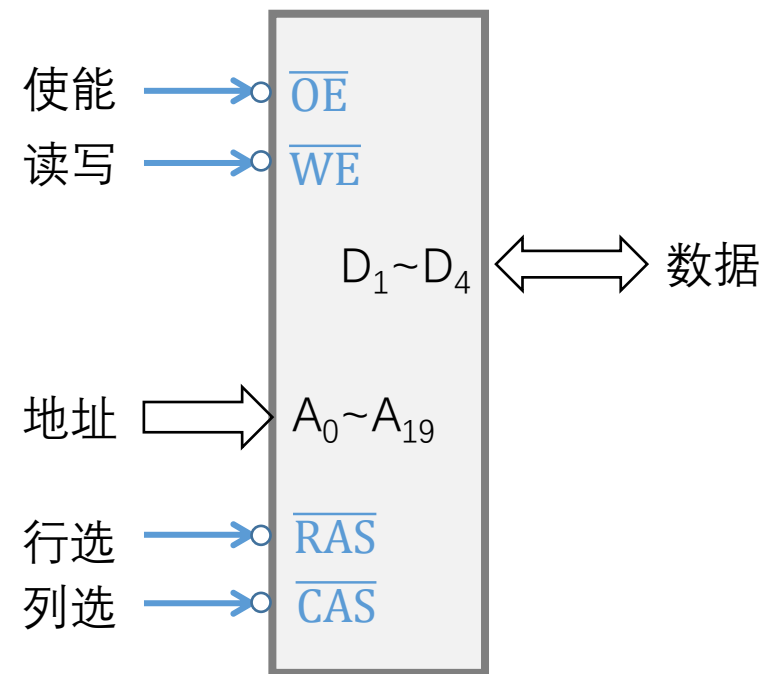
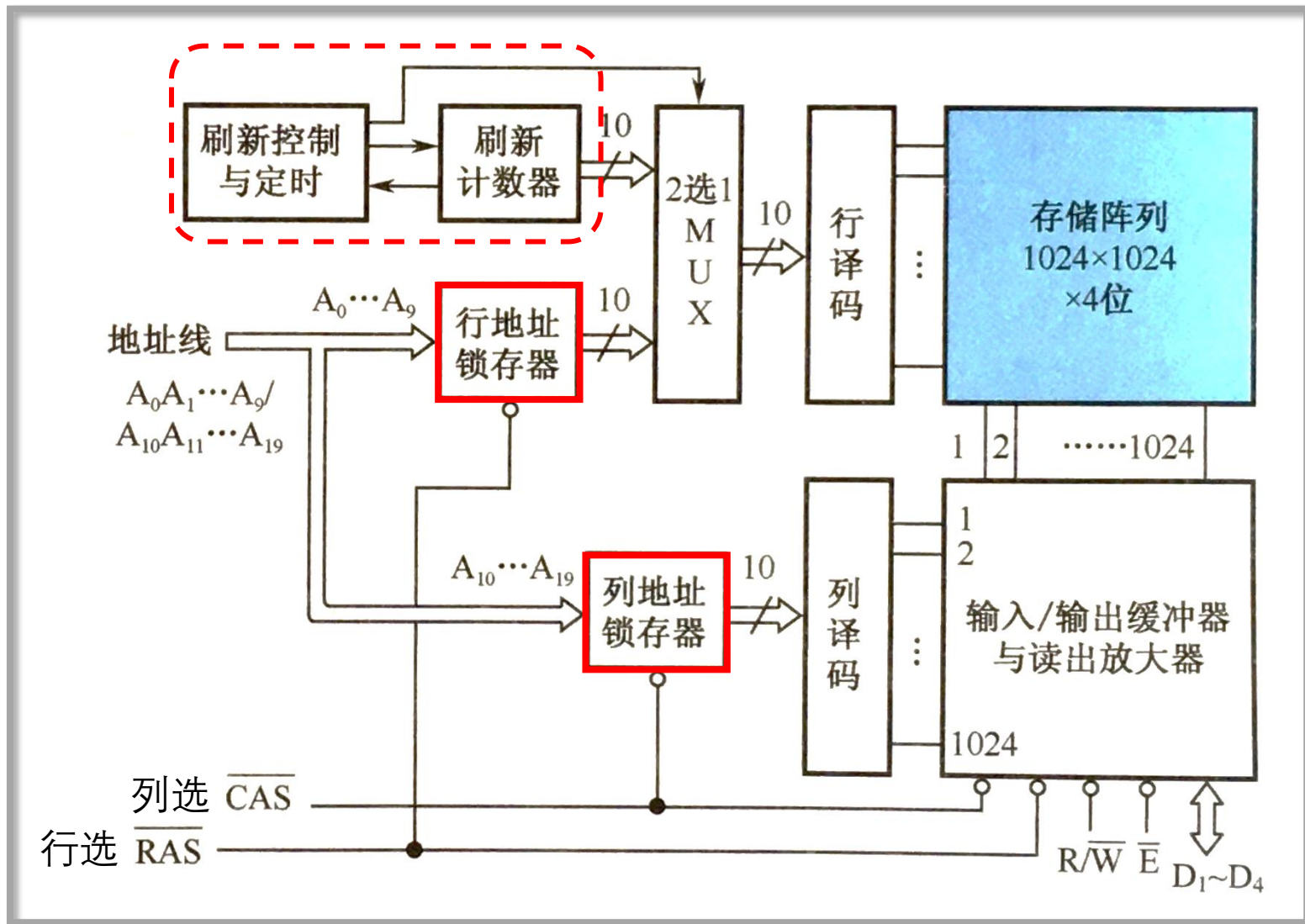
# DRAM 存储元



MOS晶体管 + 电容器 组成的单管记忆电路  
(开关) (电量高=1)  
(电量低=0)

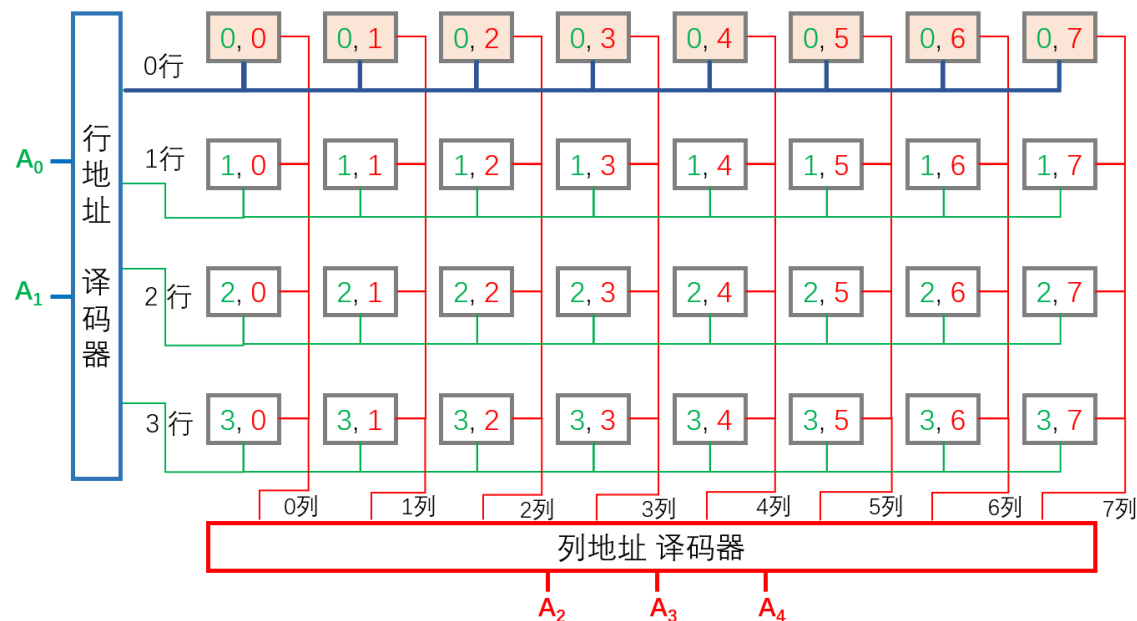
- **写1:** 仅输入缓冲器打开, 行线导通CMOS,  $D_{in}=1$ , 给电容器充电。
- **写0:** 仅输入缓冲器打开, 行线导通CMOS,  $D_{in}=0$ , 电容器放电。
- **读1:** 仅输出缓冲器打开, 行线导通CMOS,  $D_{out}=1$ 。电容电量减少。
- **刷新:** 仅刷新缓冲器打开, 行线导通CMOS,  $D_{out}=1$  电流再流回到电容上。

# DRAM芯片的逻辑结构

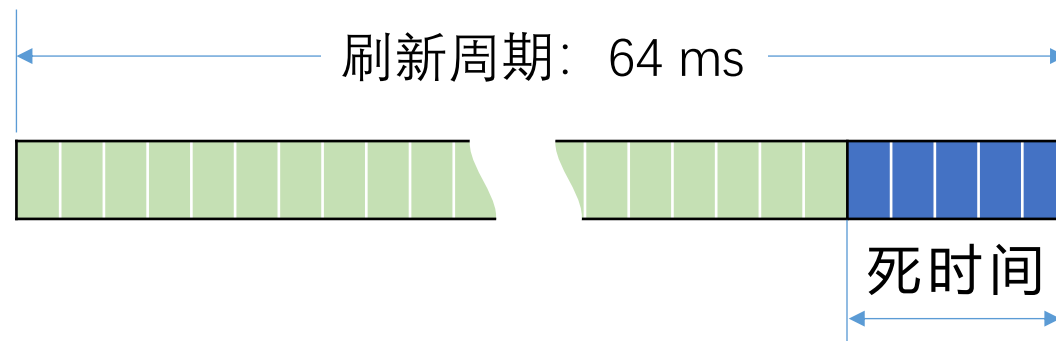


# DRAM 的刷新

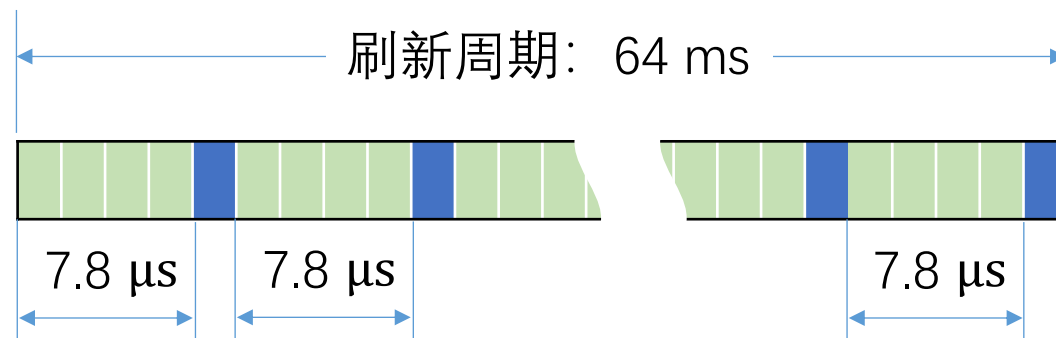
- 每次读操作后都要刷新
- 电容放电，需定期刷新



## 集中式刷新

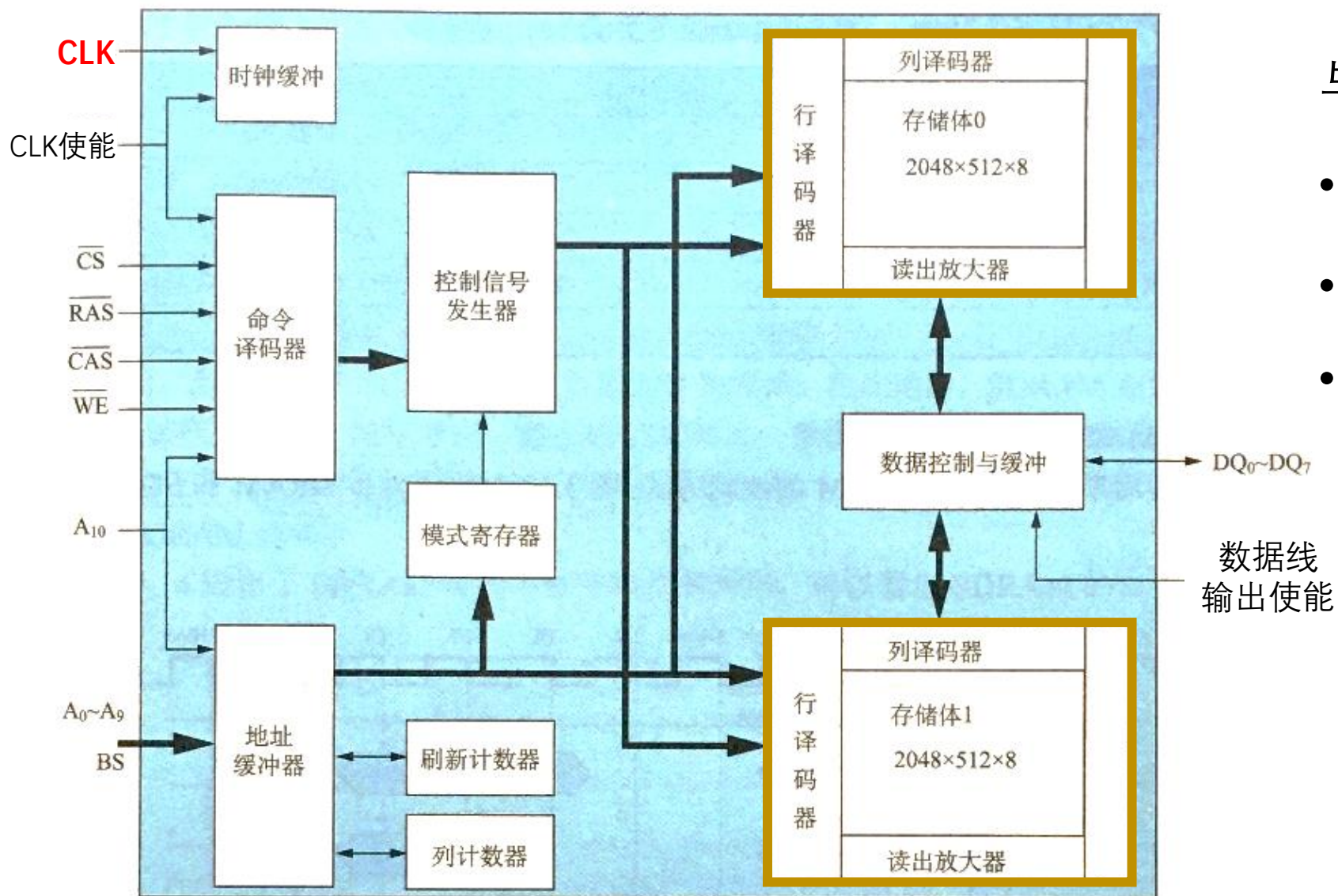


## 分散式刷新





# 同步DRAM (SDRAM)

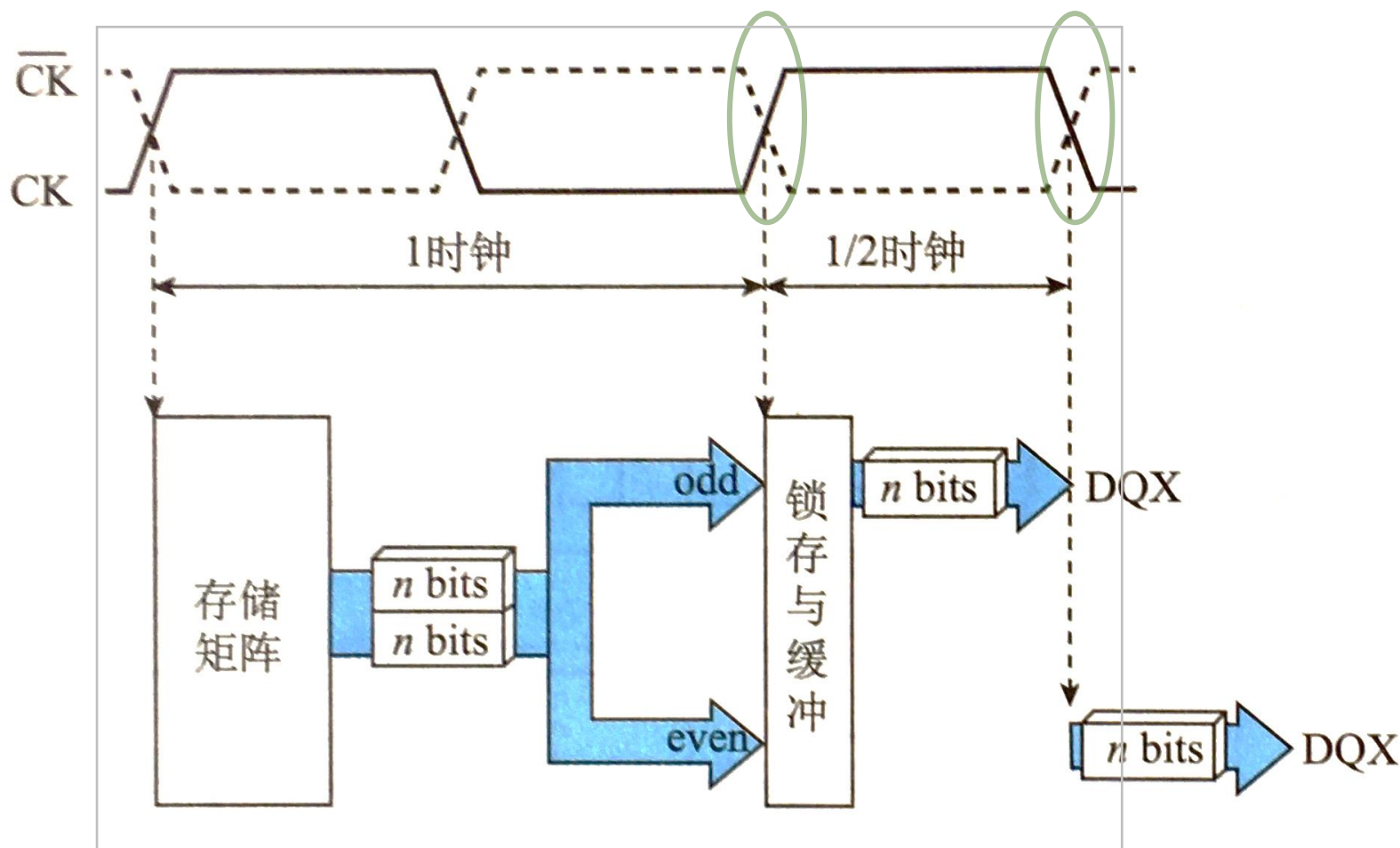


与传统异步DRAM相比:

- 增加**系统时钟 CLK**
- 有多个独立的**存储体**
- 将控制信号组合为命令

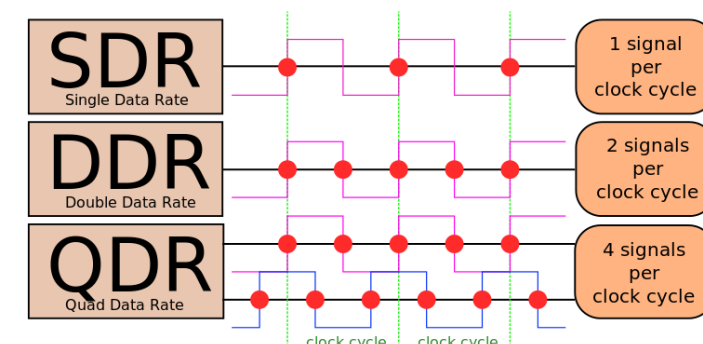


差分时钟



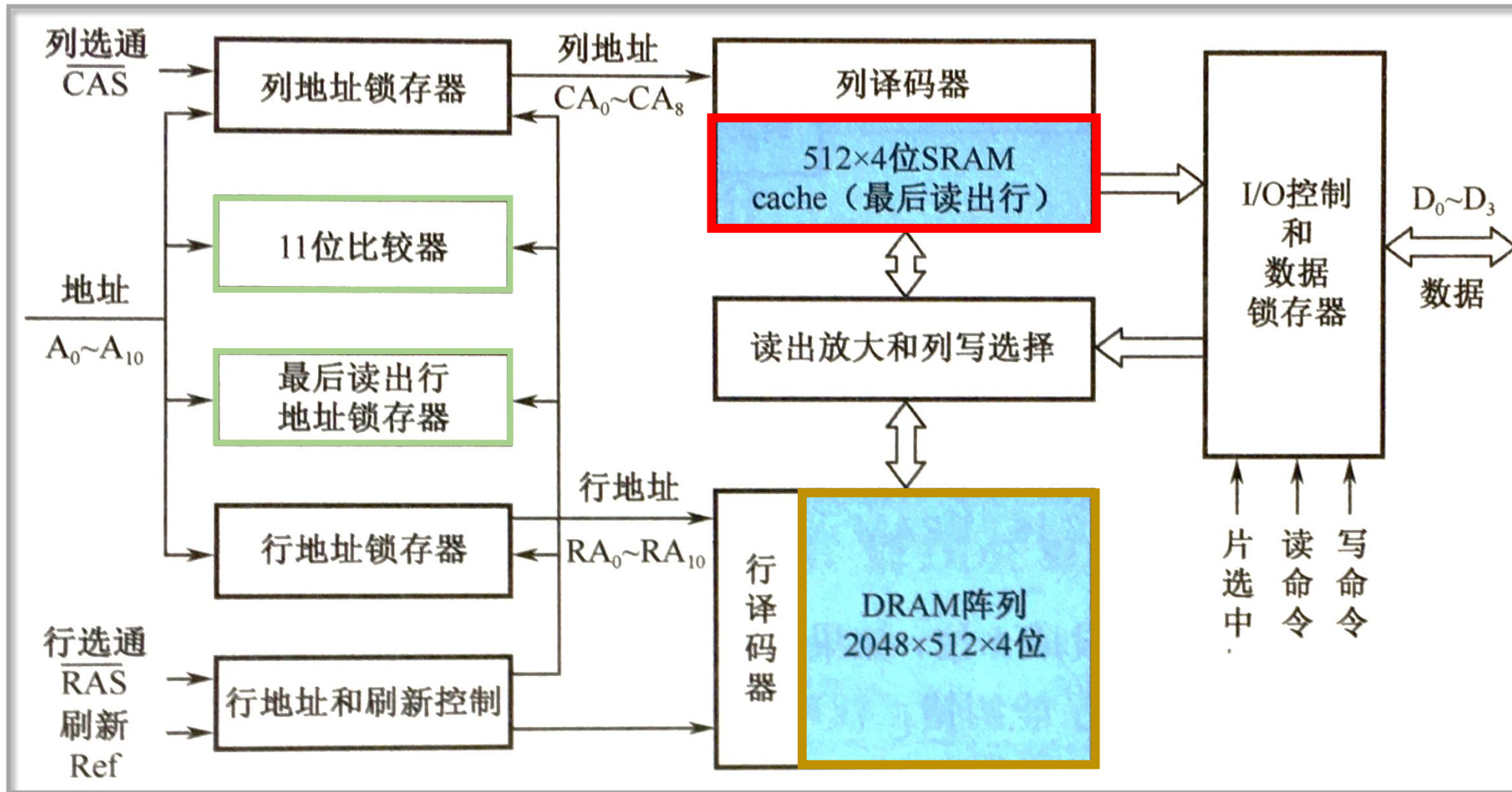
DDR与SDR相比:

- 时钟上下沿都能传数据
- $CK + \overline{CK}$ : 差分时钟
- DDR2、DDR3、DDR4...



# (Cached) CDRAM

1M×4位 CDRAM (内含1片512×4位 cache)



- 首次读:  
DRAM→SRAM
- 下次读:  
比较SRAM?

## 优点

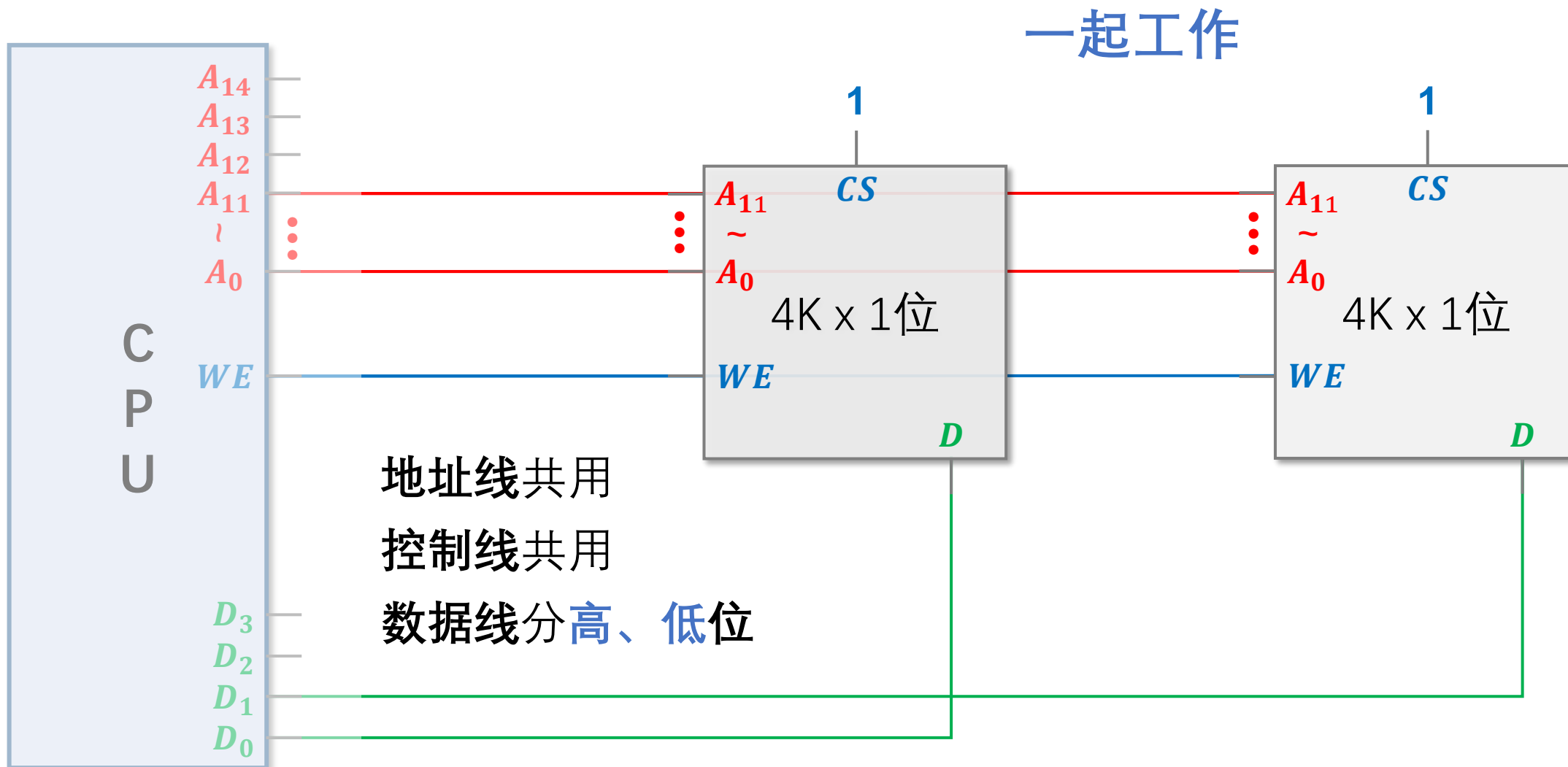
- 突发速度快
- 读, 写并行
- 读, 刷新并行



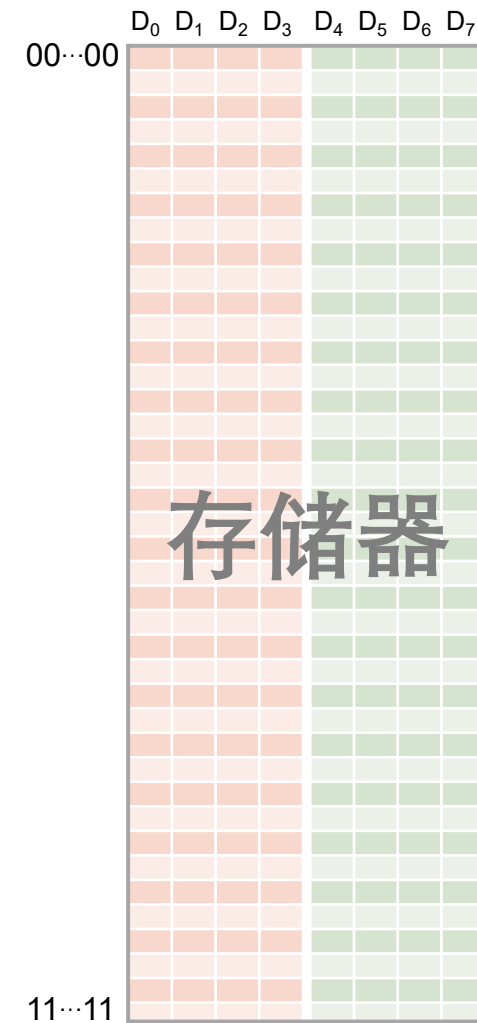
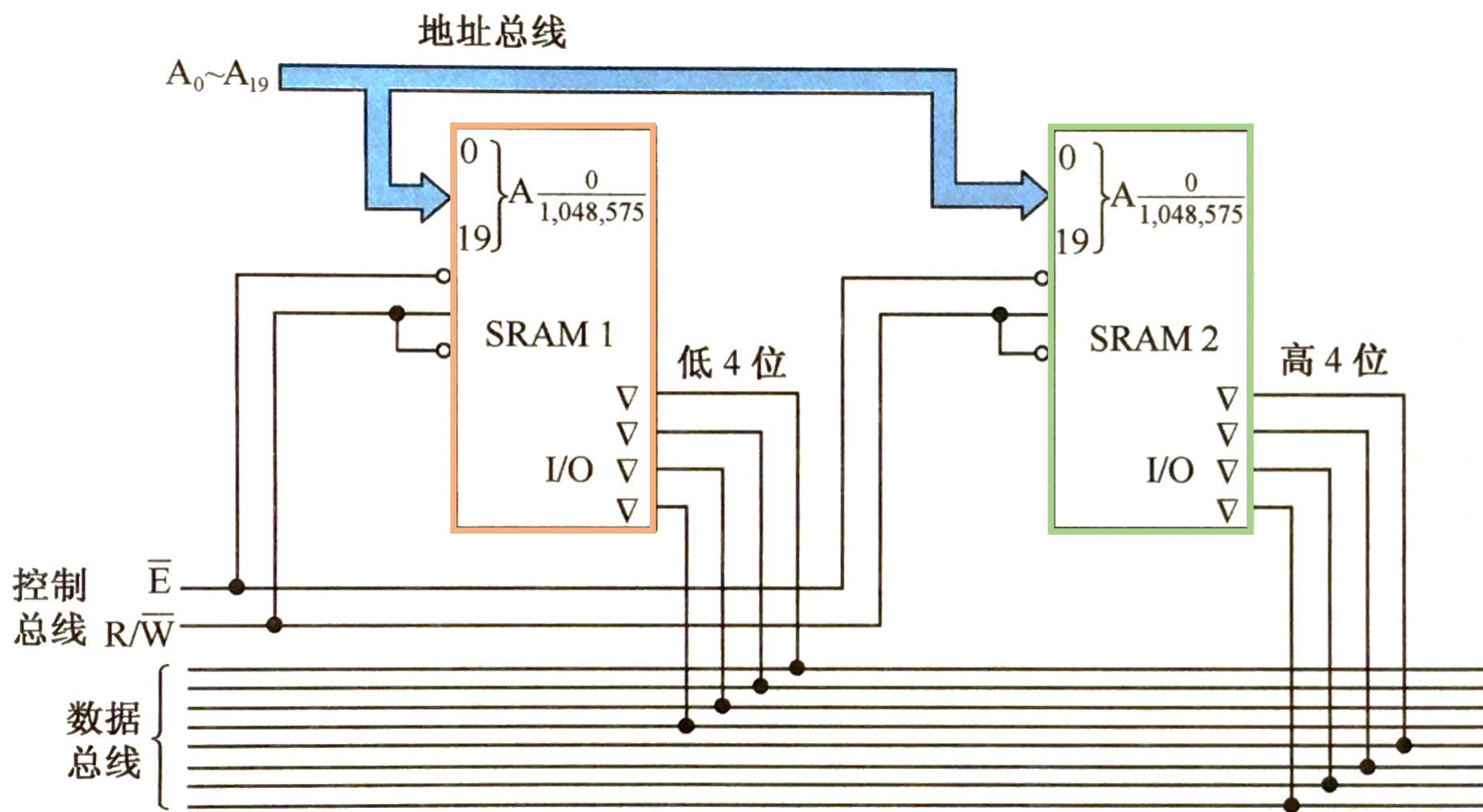
6

# 容量扩展

# 存储器位扩展

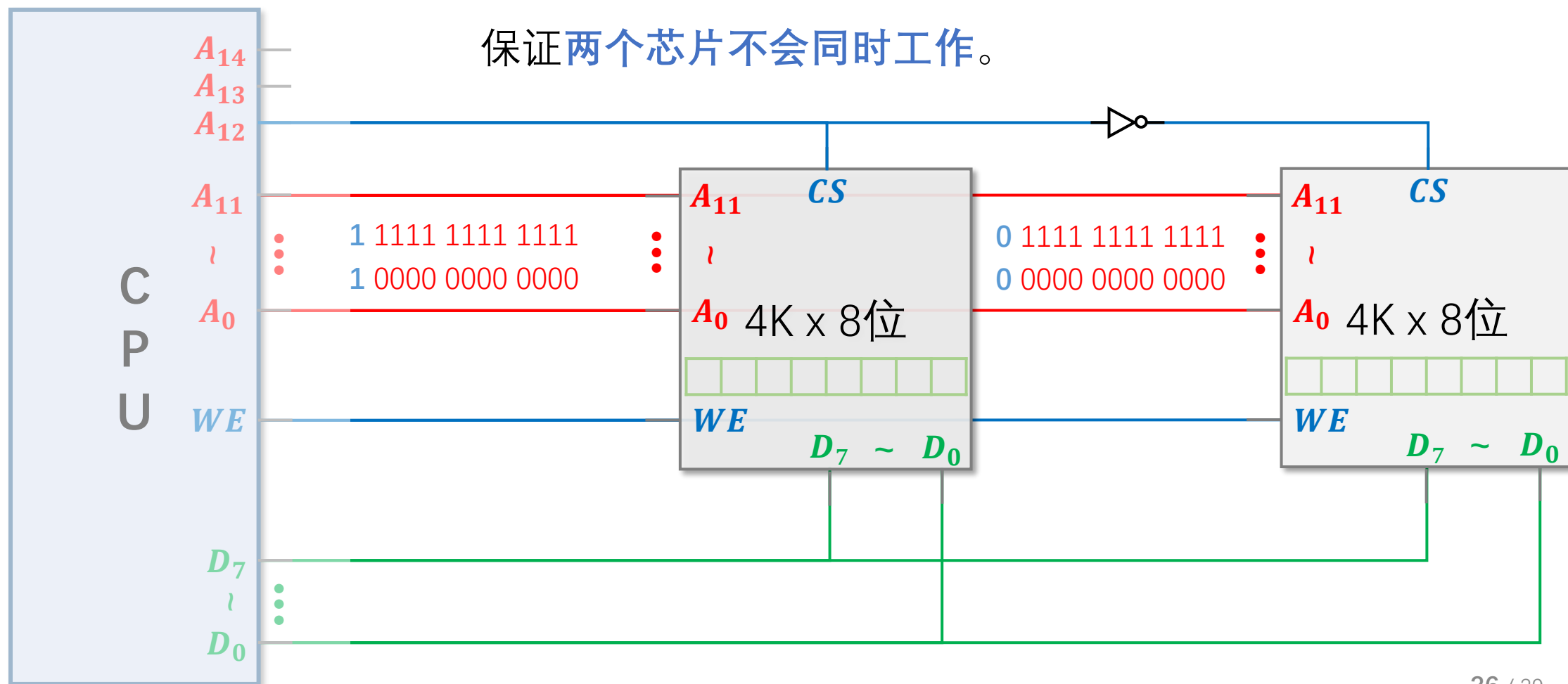


# 【例1】2片1M×4位 → 1M×8位

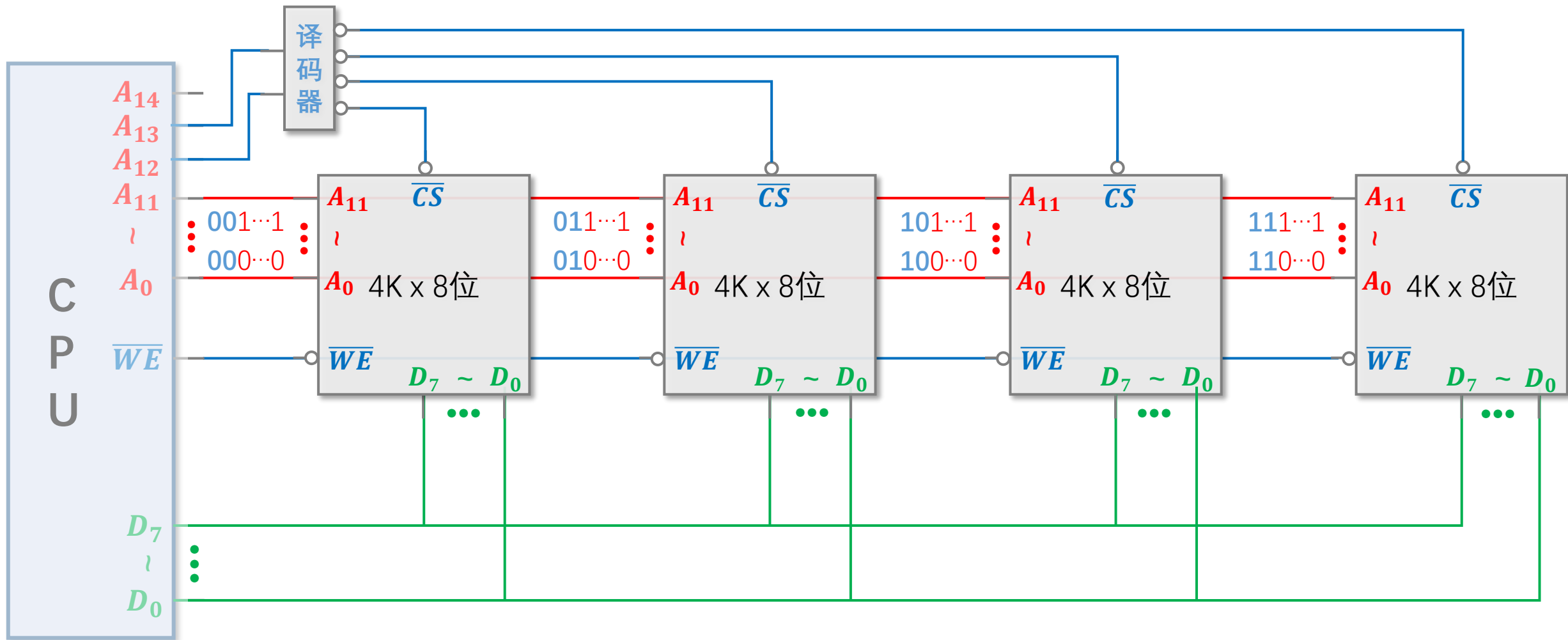


# 存储器字扩展

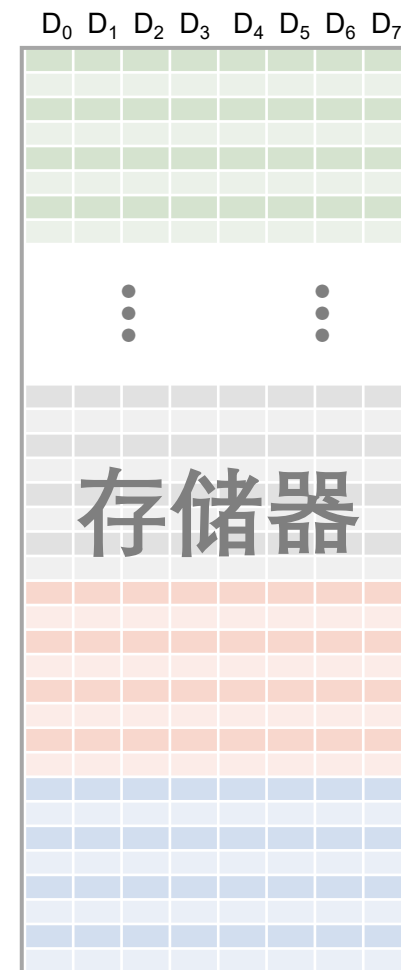
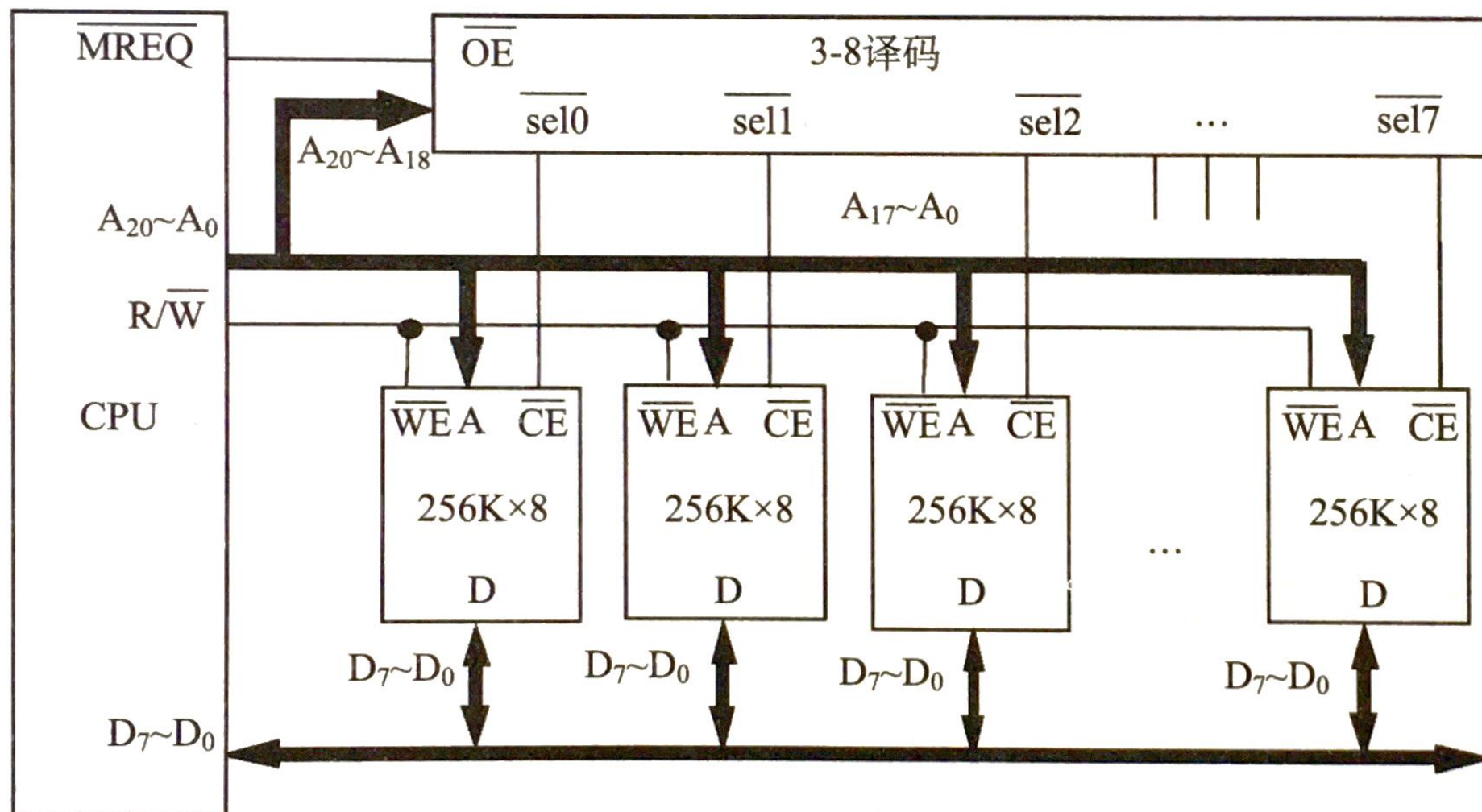
- 数据线共用;
- 地址线中的最高位 ( $A_{12}$ ) 取反后连到第2个DRAM的片选控制线, 保证两个芯片不会同时工作。



# 存储器字扩展 + 译码器



# 【例2】8片 $256\text{K} \times 8\text{位}$ $\rightarrow$ $2048\text{K} \times 8\text{位}$



# 存储器字 + 位扩展

