

## 5.2.2 动态随机存取存储器DRAM

### ■ DRAM概述

- 一个动态RAM的基本存储单元，它由一个MOS管T1和位于其栅极上的分布电容CS构成。当栅极电容C上充有电荷时，表示该存储单元保存信息“1”，反之，当栅极电容上没有电荷时，表示该单元保存信息“0”
- 刷新：动态RAM存储单元实质上是依靠T1管栅极电容的充放电原来保存信息的。时间一长，电容上锁保存的电荷就会泄露，造成了信息的丢失。因此，在动态RAM的使用过程中，必须及时地想保证“1”的那些存储单元补充电荷，以维持信息的存在。这一过程，就称为动态存储器的刷新操作。

## ■ DRAM技术和工艺的发展

- 存储器芯片的容量和速度在不断地提高，此外存储器的组织结构和访问方式也在发生变化
- FPM（快页模式） DRAM
- EDO DRAM
- SDRAM（同步DRAM）
- DDR（双数据率） SDRAM
- RDRAM（Rambus DRAM）
- 容量也由以前的1M到现在的256M， 512M甚至更高速度发展

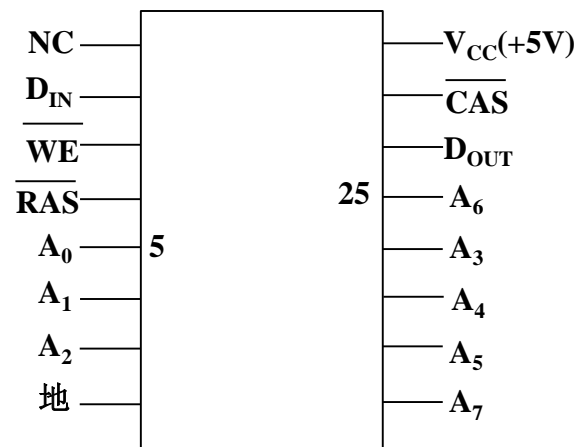
## ■ 1. 2164A的引脚功能

- Intel2164A是一种 $64K \times 1$ 的动态RAM存储器芯片，其基本存储单元采用单管存储电路，其他的典型芯片有Intel 21256/21464等。

### ■ 引脚含义

具有16个引脚的双列直插式集成电路芯片。

- $A_0 \sim A_7$ ：地址输入线
- $D_{IN}$ 和 $D_{OUT}$ ：芯片的数据输入、输出线
- $\overline{WE}$ ：写允许信号，当为低电平时，执行写操作
- $\overline{RAS}$ ：行地址锁存信号，当为低电平时，表明芯片当前接收的是行地址
- $\overline{CAS}$ ：列地址锁存信号，当为低电平时，表明芯片当前接收的是列地址



## ■ 2. DRAM的工作过程

### ➤ 数据读出

- ◆ 在对Intel 2164A的读操作过程中，它要接收来自CPU的地址信号，经译码选中相应的存储单元后，把其中保存的一位信息通过DOUT数据输出引脚送至系统数据总线。
- ◆ 时序图如图

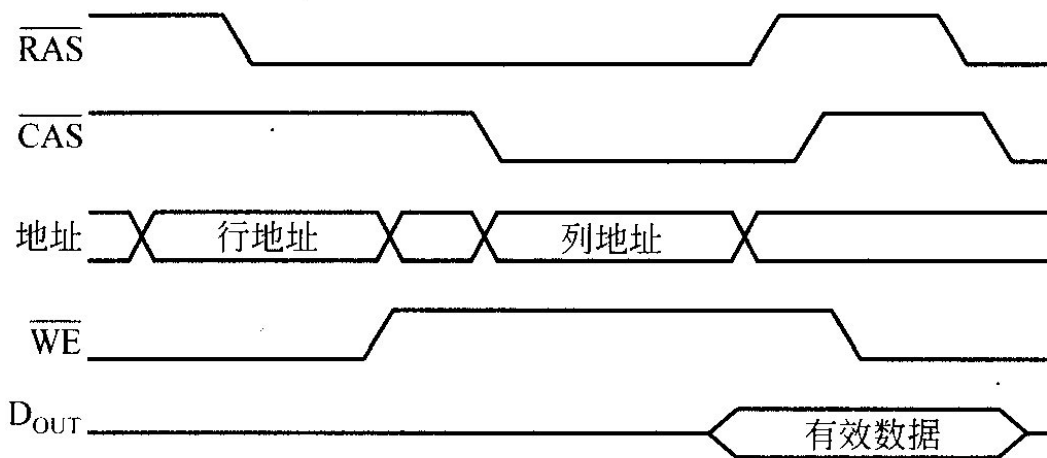


图 5-15 DRAM 的数据读出时序图

➤ 数据写入

◆ 时序图如图

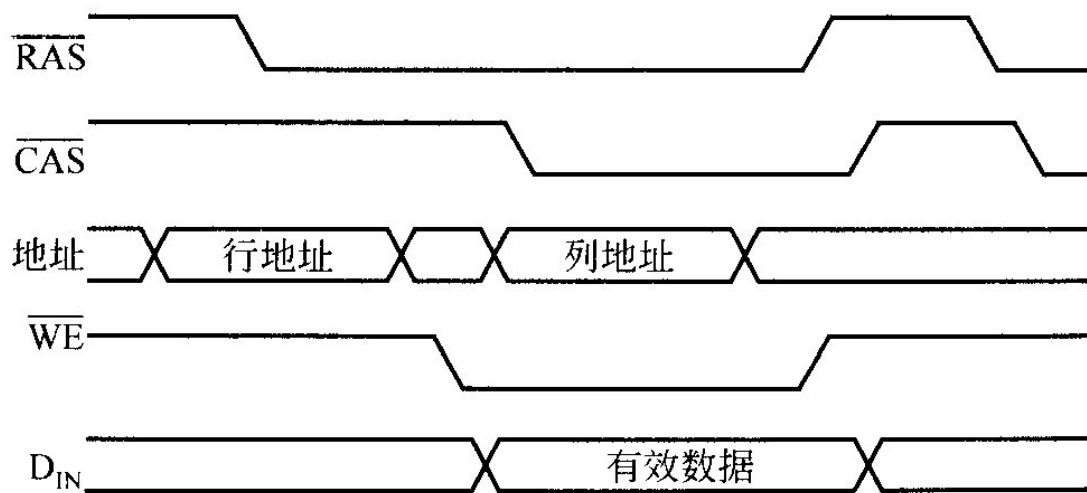


图 5-16 DRAM 的数据写入时序图

## ➤ 刷新

- ◆ 刷新：将动态存储器中存放的每一位信息读出并重新写入的过程
- ◆ 方法：使列地址锁存信号无效，只送上行地址并使行地址锁存有效，实现整行的单元刷新。
- ◆ 时序图

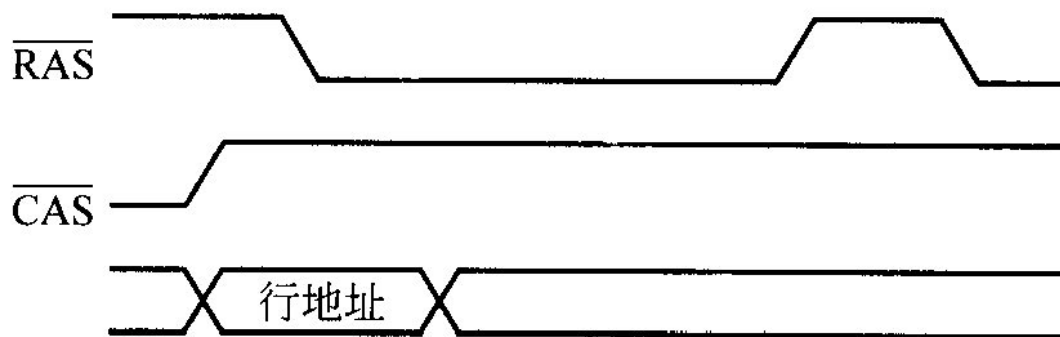


图 5-17 DRAM 芯片的刷新时序图

### ■ 3. DRAM在系统中的连接

- 芯片上的每个单元中只存放1位二进制码，每字节数据分别存放在8片芯片中
- 系统的每一次访存操作需同时访问8片2164A芯片，该8片芯片必须具有完全相同的地址
- 芯片的地址选择是按行、列分时传送，由系统的低8位送出行地址，高8位送出列地址
- 结论：
  - 每8片2164A构成一个存储体（单独一片则无意义）
  - 每个存储体内的所有芯片具有相同的地址（片内地址），应同时被选中，仅有数据信号由各片分别引出。

## 5.2.3 存储器扩展技术

### ■ 存储器扩展

- 存储器芯片的存储容量等于：  
    单元数 × 每单元的位数
- 用多片存储芯片构成一个需要的内存空间
- 任一时刻仅有一片（或一组）被选中



## ■ 存储器扩展方法

- 位扩展：构成内存的存储器芯片的字长小于内存单元的字长时，需进行位扩展——扩展字长
- 字扩展：芯片每个单元中的字长满足，但单元数不满足时，需进行字扩展——扩展容量
- 字位扩展：字长和单元数均不满足时，需同时进行位扩展和字扩展——既扩展字长也扩展容量

## ■ 1. 位扩展

- 例：用2片4K\*4b的芯片构成4KB的存储器
- 例：用8片2164A芯片构成64KB存储器
- 位扩展方法
  - ◆ 将每片的地址线、控制线并联，数据线分别引出至数据总线的不同位上
- 位扩展特点：
  - ◆ 存储器的单元数不变，位数增加

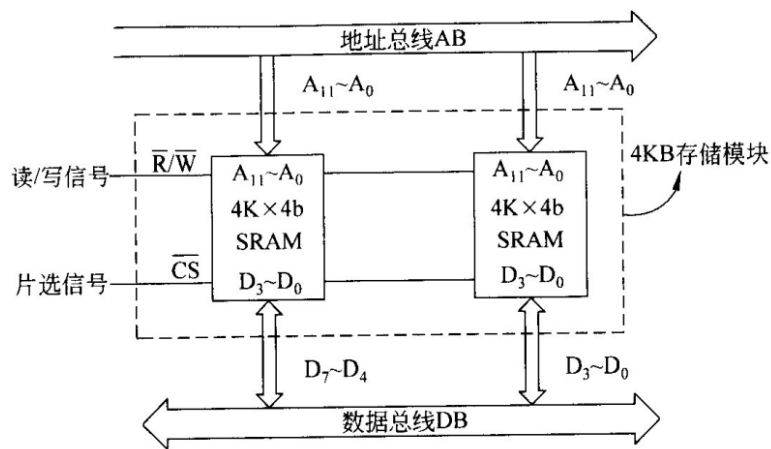


图 5-19 用 4K×4b 的 SRAM 芯片进行位扩展以构成容量为 4KB 的存储器

## ■ 2. 字扩展

### ➤ 扩展原则

- ◆ 每个芯片的地址线、数据线、读/写信号线并联
- ◆ 片选端分别引出，以使每个芯片有不同的地址范围。

### ➤ 扩展特点

- ◆ 位数不变，容量增加

### ➤ 例：用2片64K\*8b的SRAM芯片构成容量为128KB的存储器

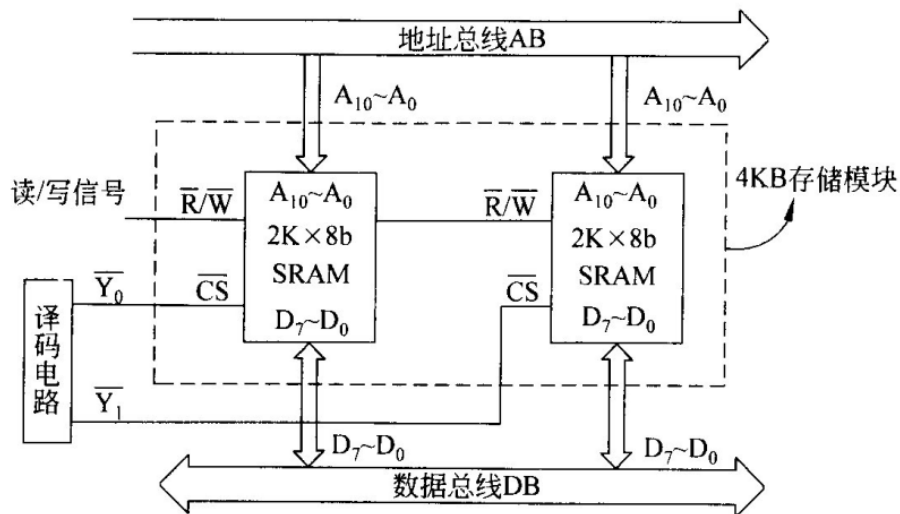


图 5-21 字扩展连接示意图

### ■ 3. 字位扩展

- 若已有存储芯片的容量为 $L \times K$ ，要构成容量为 $M \times N$ 的存储器，需要的芯片数位：

$$(M / L) \times (N / K)$$

- 例：用2164A芯片构成容量为128KB的内存
- 设计过程：
  - ◆ 根据内存容量及芯片容量确定所需存储芯片数
  - ◆ 进行位扩展，构成内存模块，以满足字长要求
  - ◆ 对内存模块进行字扩展，以满足容量要求

## 5.3 只读存储器ROM

---

- EPROM（紫外线擦出）
- EEPROM（电擦除）

## 5.3.1 EPROM

- EPROM又被称为可擦除编程的ROM，其信息可以重复写入及清除。
- 清除方式：用一定波长的紫外光照射浮动栅，使电子以光电流的形式释放掉，这时原来存储的信息也就不存在了。
- 写入方式：在源极和漏极间加上较高负电荷和编程脉冲，电子穿过绝缘层注入到浮动栅，当高电压撤去后，由于浮动栅被绝缘层所包围，注入电子在室温、无光照下可以长期保存在浮栅中，这就实现信息的存储。
- 特点
  - 可多次编程写入
  - 掉电后内容不丢失
  - 内容的擦除需用紫外线擦除器



## ■ 1. 2764的引线及功能

- 是一种 $8K \times 8\text{bit}$ 的EPROM芯片，双列直插式封装，28个引脚，其最基本的存储单元，就是采用如上所述的带有浮动栅的MOS管，其它的典型芯片有Intel 2732/2716/27128等。

## ■ 引脚含义

- $A_0 \sim A_7$ ：地址输入线
- $D_0 \sim D_7$ ：双向数据线
- $/\text{CE}$ ：片选信号
- $/\text{OE}$ ：输出允许信号
- $/\text{PGM}$ ：编程脉冲输入端
- $V_{\text{PP}}$ ：编程电压输入端

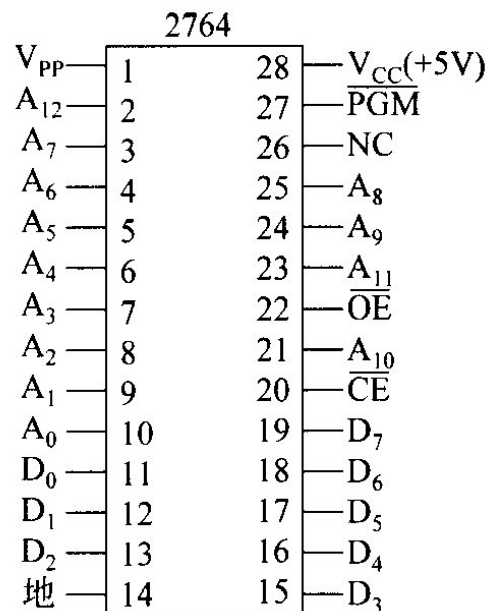


图 5-24 EPROM 2764 引线图

## 5.3.1 EPROM

### ■ 2. 2764的工作方式

#### ➤ 1) 数据读出

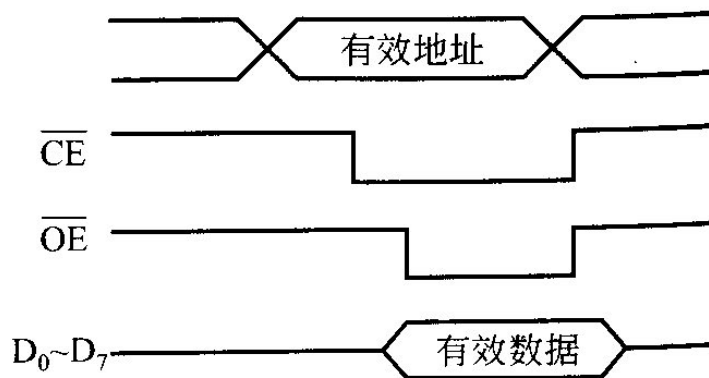


图 5-25 2764 读出过程的时序图

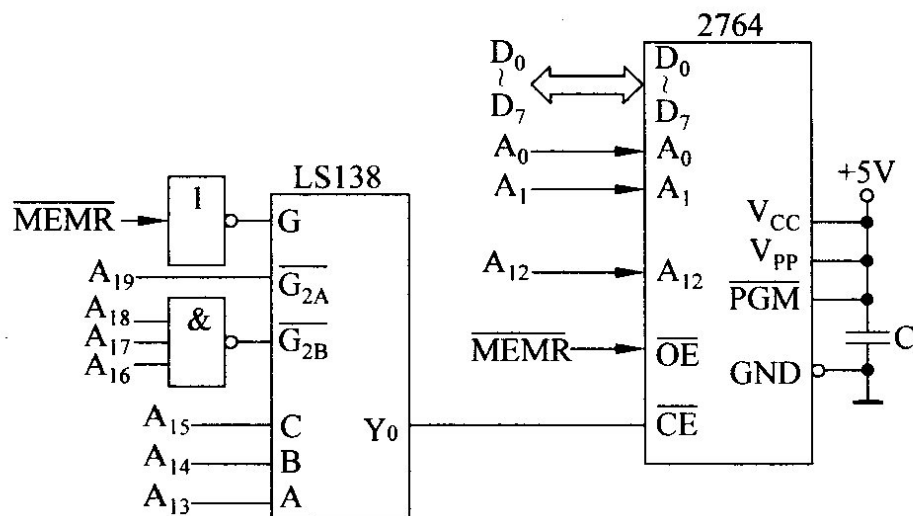


图 5-26 2764 与 8088 系统的连接图



## ➤ 2) 编程写入

◆ 标准编程方式：每出现一个编程负脉冲就写入一个字节数据

- ✓ VPP端加上高电压
- ✓ 给出地址
- ✓ 片选信号，/OE无效
- ✓ 在数据线上给出要写入的数据
- ✓ 稳定后，/PGM加上50+/-5ms的负脉冲，就将一字节的数据写入相应的地址单元中。

## ◆ 快速编程方式

- ✓ 编程脉冲比标准脉冲要窄
- ✓ 校验有错时，仅重写此单元，写完后再校验，直到正确；或是连续10次依然有错，认为芯片已损坏。

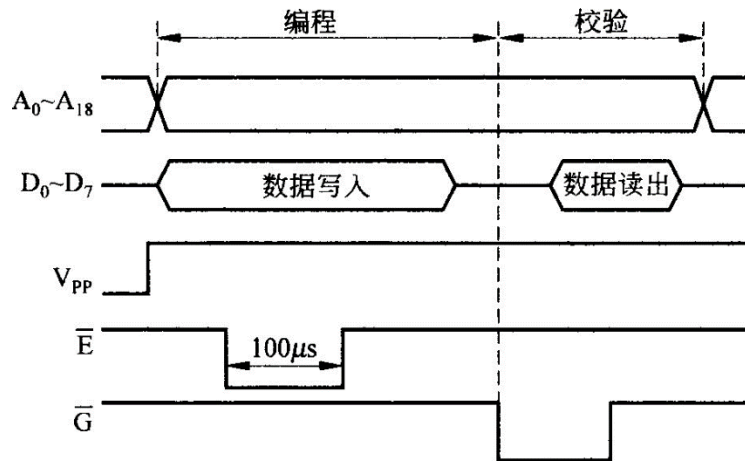


图 5-27 EPROM 27C040 的编程时序图

### ➤ 3) 擦除

- ◆ 放到专门的擦除器上进行紫外线光照射，擦除内容
- ◆ 每个存储单元的内容都是FFH

## 5.3.2 EEPROM（电擦除可编程只读存储器）

---

- 工作原理与EPROM相似，但是擦除数据方式不同，EEPROM编程与擦除所用的电流是极小的
- 特点
  - 可在线编程写入
  - 掉电后内容不丢失
  - 电可擦除

## ■ 1. 98C64A的引线

➤ 8K × 8bit芯片

### ■ 引脚含义

- $A_0 \sim A_{12}$ : 地址线
- $D_0 \sim D_7$ : 数据线
- $\overline{CE}$ : 片选信号
- $\overline{OE}$ : 输出允许信号
- $\overline{WE}$ : 写允许信号
- $\overline{READY}/\overline{BUSY}$ : 状态输出端

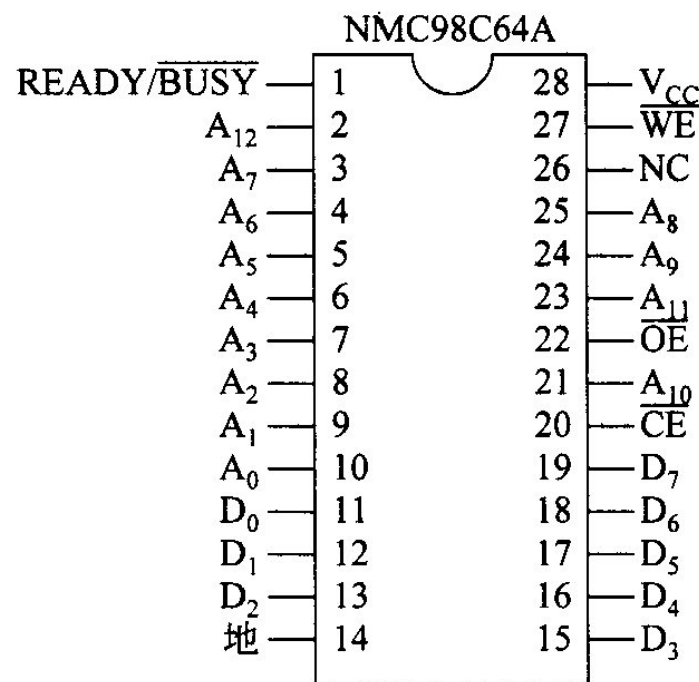


图 5-29 NMC98C64A 引线图

## ■ 2. 98C64A的工作方式

### ➤ 数据读出

➤ 编程写入 { 字节写入： 每一次BUSY正脉冲写入一个字节  
自动页写入： 每一次BUSY正脉冲写入一页  
(1~32字节)

➤ 擦除 { 字节擦除： 一次擦除一个字节  
片擦除： 一次擦除整片



- 

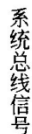


图 5-31

## START:

**MOV AX, 3E00H****MOV DS, AX**

**MOV SI, 0000H**

## MOV CX, 8192

## AGAIN:

**MOV AL, 66H**

**MOV [SI], AL**

**CALL T120us**

INC SI

## LOOP AGAIN

# HLT

---

<b>START:</b>	<b>MOV AX, 3E00H</b>
	<b>MOV DS, AX</b>
	<b>MOV SI, 0000H</b>
	<b>MOV CX, 8192</b>
	<b>MOV BL, 66H</b>
<b>AGAIN:</b>	<b>MOV DX, 20E0H</b>
<b>WAIT:</b>	<b>IN AL, DX</b>
	<b>TEST AL, 01H</b>
	<b>JZ WAIT</b>
	<b>MOV [SI], BL</b>
	<b>INC SI</b>
	<b>LOOP AGAIN</b>
	<b>HLT</b>



## 5.3.3 闪存FLASH

- 具有EEPROM的特点，又可在计算机内进行擦除和编程，它的读取时间与DRAM相似，而写时间与磁盘驱动器相当。
- 快擦型存储器可替代EEPROM，在某些应用场合还可取代SRAM，尤其是对于需要配备电池后援的SRAM系统，快擦型存储器还可应用于计算机的外部设备中。
- 特点
  - 编程速度快
  - 掉电后内容不丢失



## ■ 1. 引线及结构

➤ 512K × 8bit芯片，有19根地址线和8根数据线

➤ 引脚含义

- $A_0 \sim A_{18}$ ：地址线
- $DQ_0 \sim DQ_7$ ：数据线
- $\overline{G}$ ：输出允许信号
- $\overline{E}$ ：芯片写允许信号

## ■ 2. 工作方式

- 数据读出
  - 读单元内容
  - 读内部状态寄存器内容
  - 读芯片的厂家及期间标记

- 编程写入： 数据写入，写软件保护

- 擦除
  - 字节擦除，块擦除，片擦除
  - 擦除挂起