



## • 课前练习:

- 1: 十进制数10, -100转换为16进制 (字长16位)。
- 2: 定义两个变量 `int8_t m=100`, `n=-100`;他们在计算机中是以\_\_\_\_\_进制的形式存储的, 具体存储的数据为: `m`:\_\_\_\_\_, `n`:\_\_\_\_\_。
- 3: 定义一个字符串: `char hello[] = "Hello"`;则存储器`hello`地址处存放的数据是\_\_\_\_\_, `hello+1`处存放的数据是\_\_\_\_\_, `hello+2`处存放的数据是\_\_\_\_\_, `hello+3`处存放的数据是\_\_\_\_\_, `hello+4`处存放的数据是\_\_\_\_\_, `hello+5`处存放的数据是\_\_\_\_\_。(为方便书写, 可用16进制填写)。
- 4: 定义一个变量 `int16_t k=1230`。把这个变量, 从一个嵌入式系统传到另一个嵌入式系统, 实际上传递的数据是\_\_\_\_\_。(为方便书写, 可用16进制填写)。



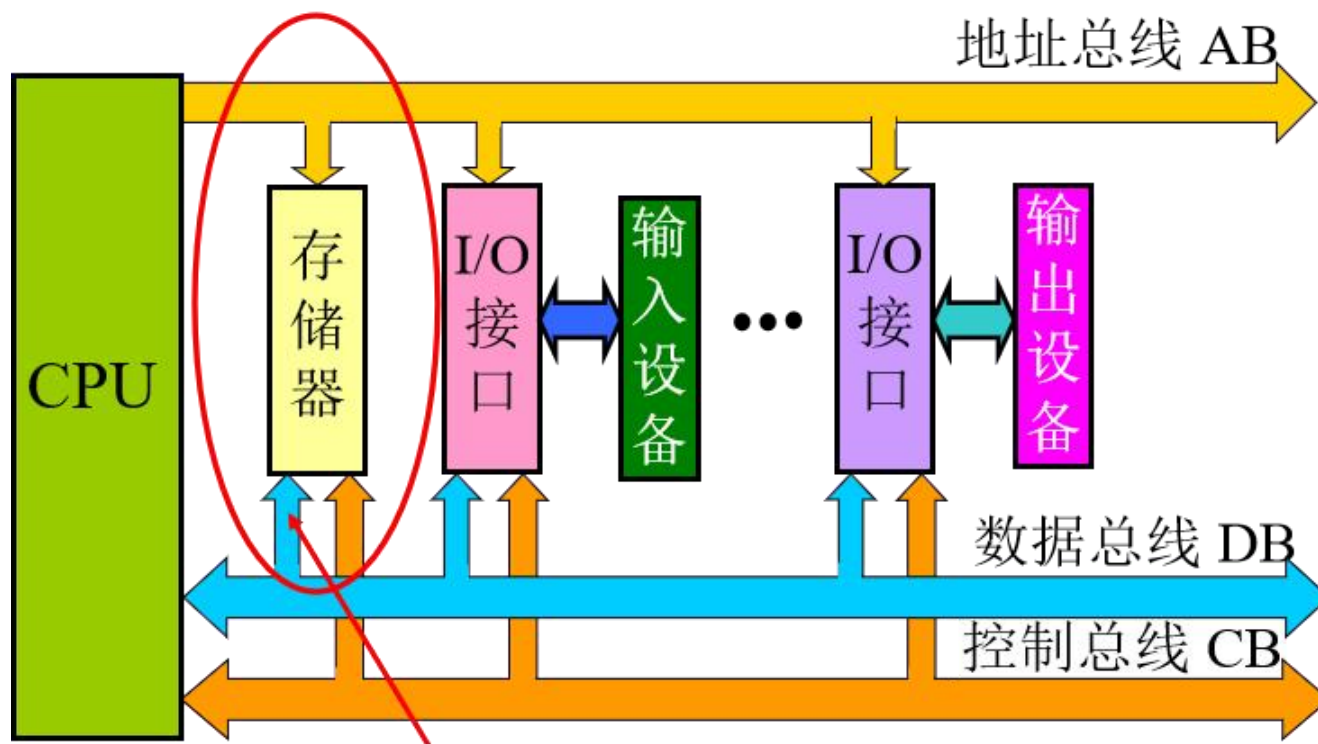
# 微机系统与接口 ——存储器

王江峰 副研究员  
电气工程学院



## 1 存储器的基本概念

- 存储器是计算机及嵌入式系统中的记忆设备





## 2 存储器的分类

- 按存储介质分类
  - 半导体存储器（由半导体器件组成）
  - 磁带、光盘等（磁性材料作为记录介质、应用激光在记录介质上进行读/写）
- 按存取方式分类（RAM: Random Access Memory 随机存取存储器；ROM: Read Only Memory 只读存取器）





## 2 存储器的分类

- SRAM: Static RAM 静态随机存储器; DRAM: Dynamic RAM 动态随机存取器; NVRAM: Non-volatile RAM 非易失随机存取器

	组成单元	速度	集成度	应用
<b>SRAM</b>	触发器	快	低	小容量系统
<b>DRAM</b>	极间电容	慢	高	大容量系统
<b>NVRAM</b>	铁电材料, 磁阻材料, 液晶	慢	低	小容量非易失



## 2 存储器的分类

- 掩膜ROM

- 信息制作在芯片中，不可更改

- PROM

- 允许一次编程，此后不可更改

- EPROM

- 用紫外光擦除，擦除后可编程；并允许用户多次擦除和编程

- EEPROM (E<sup>2</sup>PROM)

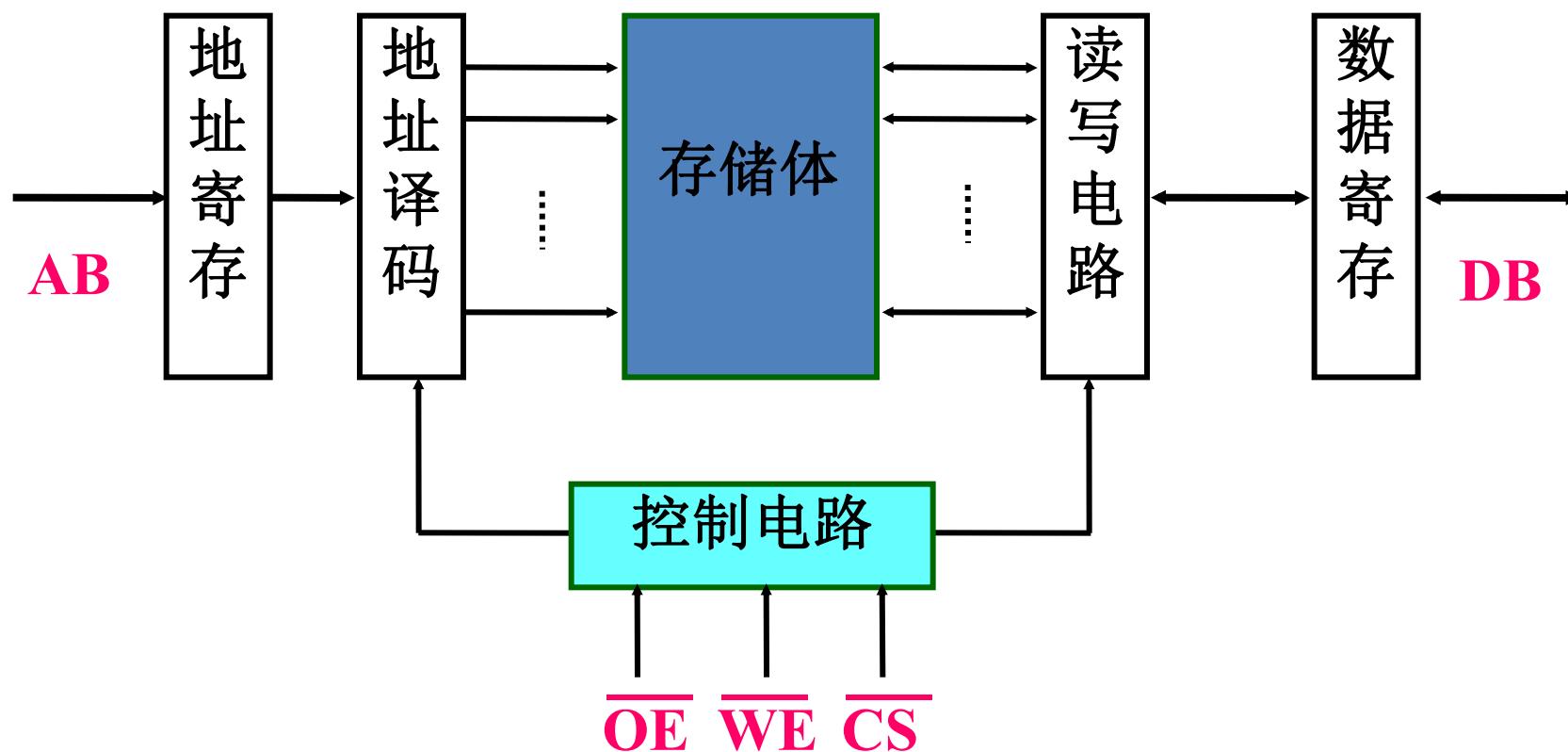
- 采用加电方法在线进行擦除和编程，也可多次擦写

- Flash Memory (闪存)

- 能够快速擦写，但只能按块 (Block) 擦除



### 3 半导体存储器芯片的组成





### 3 半导体存储器芯片的组成

#### 一、存储体

- 存储器芯片的主要部分，用来存储信息

#### 二、地址译码电路

- 根据输入的地址编码来选中芯片内某个特定的存储单元

#### 三、片选和读写控制逻辑

- 选中存储芯片，控制读写操作

#### 四、数据缓冲

- 存储器数据与总线的接口





### 3 半导体存储器芯片的组成

#### ● 存储体

- 一个基本存储单元，只能存储1位数据
- 每个存储单元具有一个唯一的地址，可存储1位（位片结构）或多位（字片结构）二进制数据
- 存储容量与地址、数据线个数有关：

芯片的存储容量

$$= 2^M \times N \text{ bits}$$

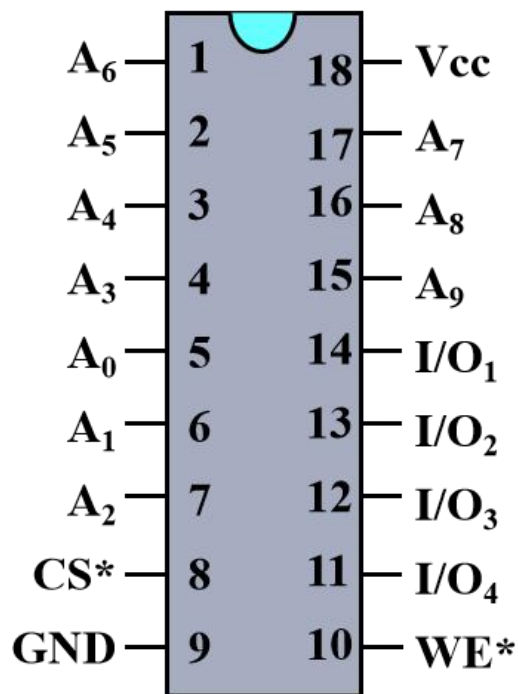
= 存储单元数 × 存储单元的位数

M：芯片的地址线根数

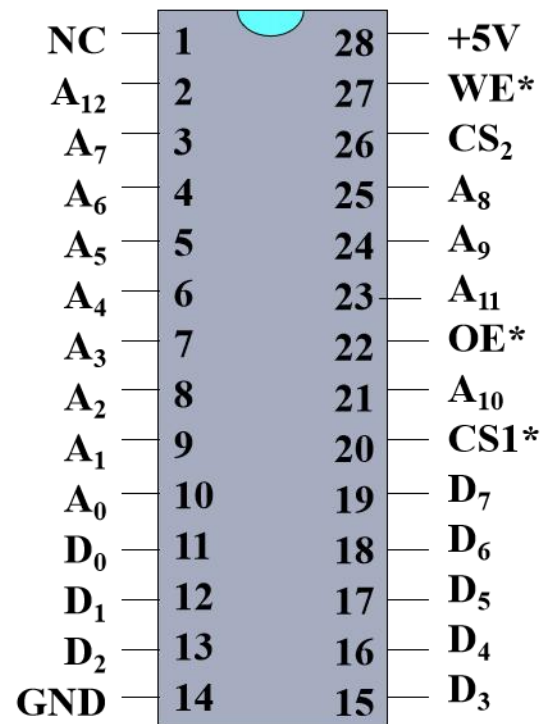
N：芯片的数据线根数



### 3 半导体存储器芯片的组成



SRAM芯片2114



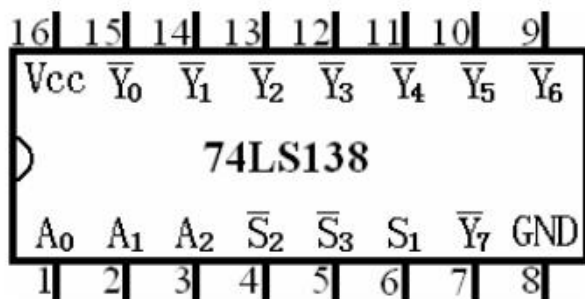
SRAM芯片6264



### 3 半导体存储器芯片的组成

#### ● 译码电路

- 根据地址，选中其唯一对应的存储单元，如典型的74LS138译码电路



3线-8线译码器（74LS138）功能表

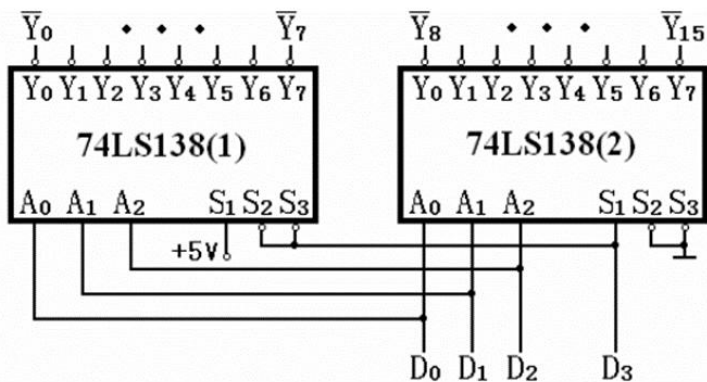
输入			输出							
$S_1$	$\overline{S_2} + \overline{S_3}$	$A_2 A_1 A_0$	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$
0	x	x x x	1	1	1	1	1	1	1	1
x	1	x x x	1	1	1	1	1	1	1	1
1	0	0 0 0	0	1	1	1	1	1	1	1
1	0	0 0 1	1	0	1	1	1	1	1	1
1	0	0 1 0	1	1	0	1	1	1	1	1
1	0	0 1 1	1	1	1	0	1	1	1	1
1	0	1 0 0	1	1	1	1	0	1	1	1
1	0	1 0 1	1	1	1	1	1	0	1	1
1	0	1 1 0	1	1	1	1	1	1	0	1
1	0	1 1 1	1	1	1	1	1	1	1	0



### 3 半导体存储器芯片的组成

#### ● 译码电路

##### ■ 级联使用

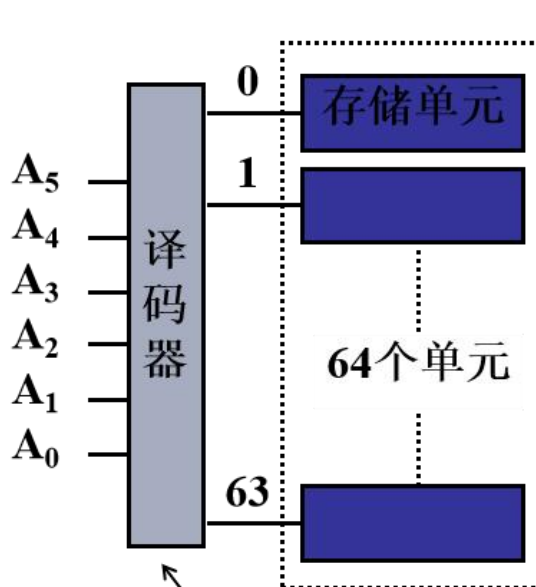


$D_3$	$D_2$	$D_1$	$D_0$	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$	$\bar{Y}_4$	$\bar{Y}_5$	$\bar{Y}_6$	$\bar{Y}_7$	$\bar{Y}_8$	$\bar{Y}_9$	$\bar{Y}_{10}$	$\bar{Y}_{11}$	$\bar{Y}_{12}$	$\bar{Y}_{13}$	$\bar{Y}_{14}$	$\bar{Y}_{15}$
0	0	0	0																
0	0	0	1																
0	0	1	0																
0	0	1	1																
0	1	0	0																
0	1	0	1																
0	1	1	0																
0	1	1	1																
1	0	0	0																
1	0	0	1																
1	0	1	0																
1	0	1	1																
1	1	0	0																
1	1	0	1																
1	1	1	0																
1	1	1	1																

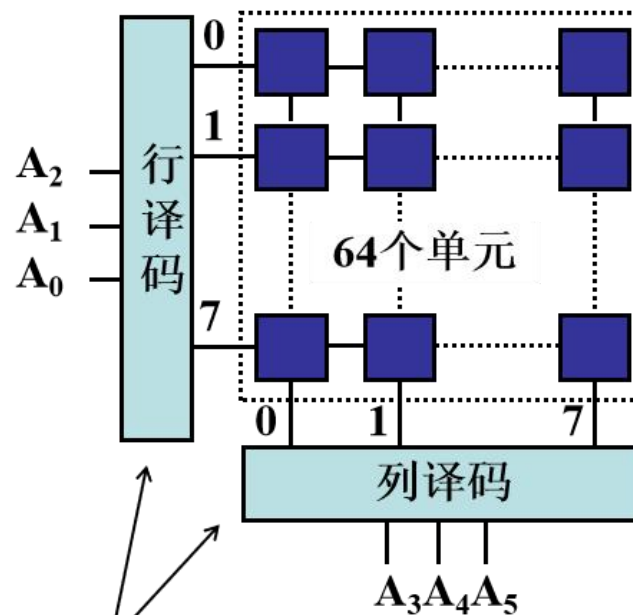


### 3 半导体存储器芯片的组成

- 存储器内部的译码电路



单译码



双译码

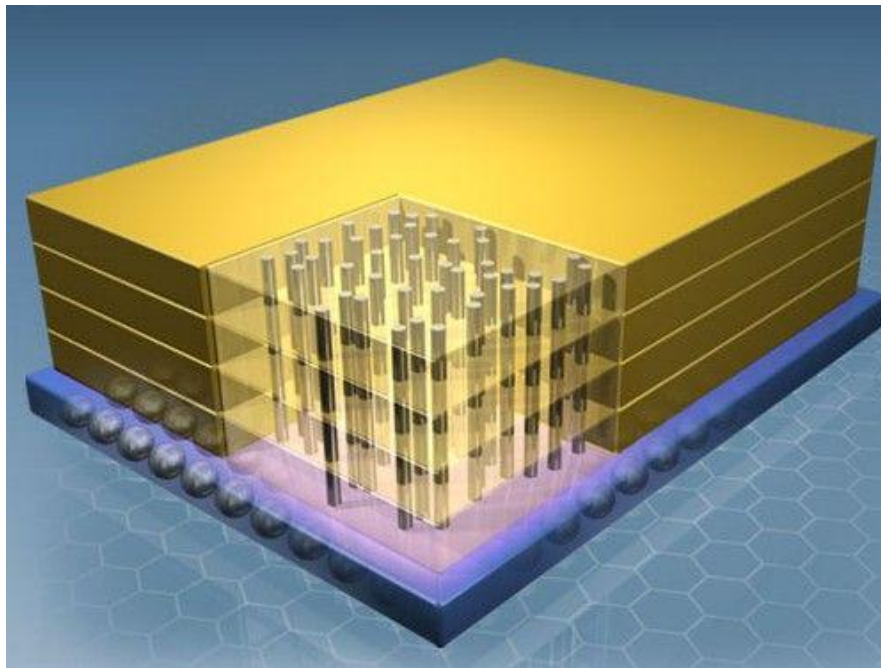
可简化芯片设计  
主要采用的译码结构



### 3 半导体存储器芯片的组成

#### ● 3D存储技术

- 普通的存储芯片多为平面结构，数据只能前后左右移动，而3D存储芯片可实现数据在三维空间中的存储和传递，将大幅提高存储设备的存储能力





### 3 半导体存储器芯片的组成

- 片选端CS\*或CE\*
  - 有效时，可以对该芯片进行读写操作
- 输出OE\*
  - 控制读操作。有效时，芯片内数据输出
  - 该控制端对应系统的读控制线
- 写WE\*
  - 写操作。有效时，数据进入芯片中
  - 该控制端对应系统的写控制线

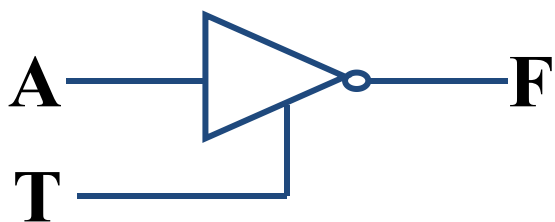




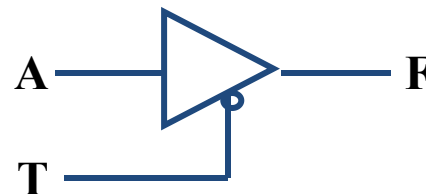
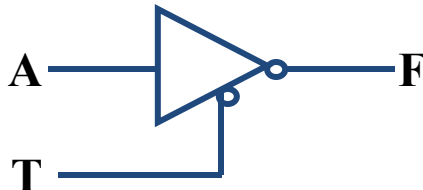
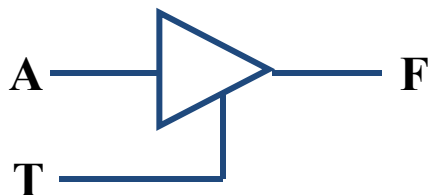
### 3 半导体存储器芯片的组成

#### ● 数据缓冲

- 存储器数据与总线的接口，本存储器工作时，既要**输入**，又要**输出**。本存储器不工作时，要呈现**高阻态**，不影响其他存储器工作
- 三态门



T为低平时：  
输出为高阻抗（三态）  
T为高电平时：  
输出为输入的反相



○ 表示反相或低电平有效

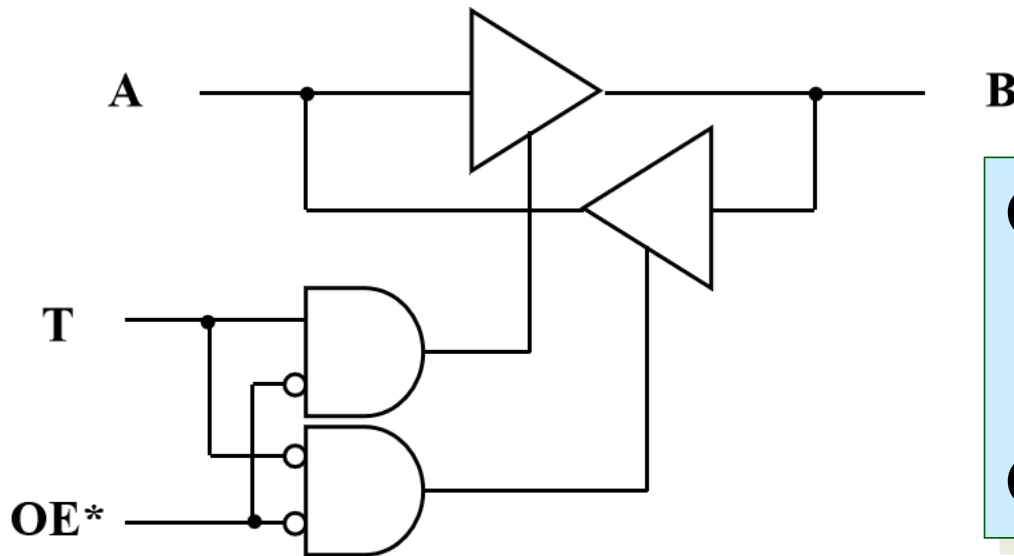




### 3 半导体存储器芯片的组成

#### ● 数据缓冲

##### ■ 双向缓冲



**$OE^*=0$ ，导通**

**$T=1 \quad A \rightarrow B$**

**$T=0 \quad A \leftarrow B$**

**$OE^*=1$ ，不导通**

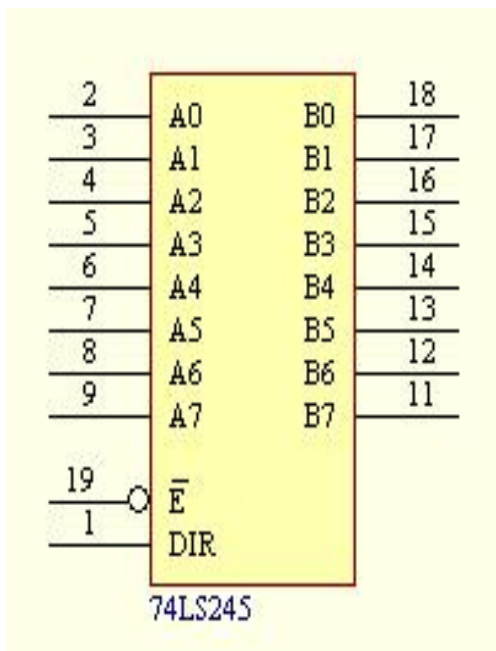
具有双向导通和三态的特性



### 3 半导体存储器芯片的组成

#### ● 数据缓冲

##### ■ 双向缓冲



74LS245

#### 8位双向缓冲器

- 控制端连接在一起，低电平有效
- 可以双向导通
- 输出与输入同相

**$E^*=0$ ，导通**

**$DIR=1$   $A \rightarrow B$**

**$DIR=0$   $A \leftarrow B$**

**$E^*=1$ ，不导通**



## 4 SRAM的工作原理

### ● SRAM

- SRAM的基本存储单元是触发器电路
- 每个基本存储单元存储二进制数一位
- 许多个基本存储单元形成行列存储矩阵
- SRAM一般采用“字结构”存储矩阵：
  - 每个存储单元存放多位（4、8、16等）
  - 每个存储单元具有一个地址



## 4 SRAM 的工作原理

### ● 保持 (Standby)

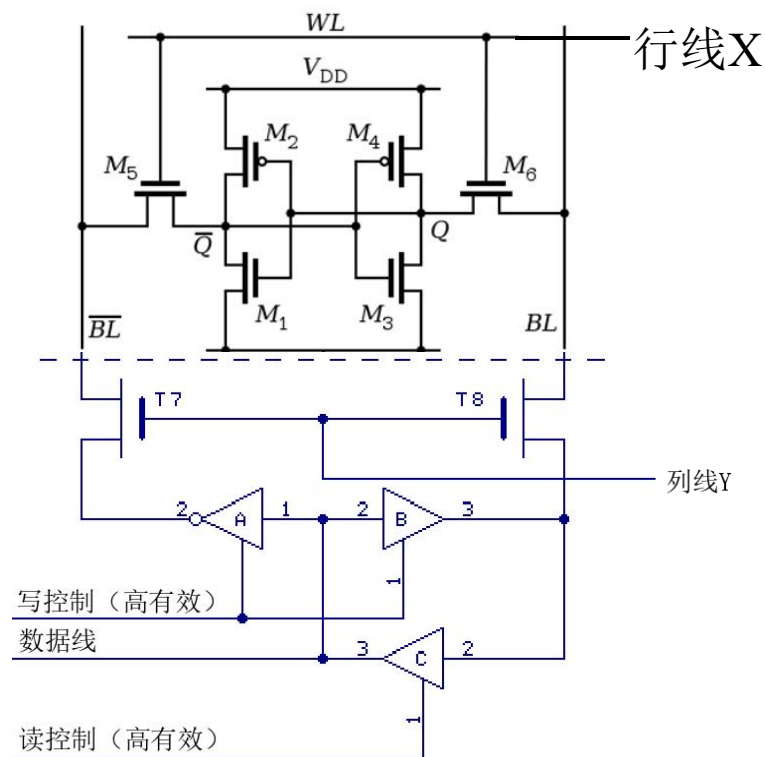
- 如果字线 (行线, Word Line) 没有被选为高电平, 那么作为控制用的  $M_5$  与  $M_6$  两个晶体管处于断路, 把基本单元与位线隔离。由  $M_1 - M_4$  组成的两个反相器继续保持其状态, 只要保持与高、低电平的连接。数据线呈 **高阻态**

### ● 读 (Reading)

- 行线和列线被选中, 读使能。写无效。  $M_5$ ,  $M_6$ ,  $T_7$ ,  $T_8$  导通,  $Q$  中保存的值传送的数据线上

### ● 写 (Writing)

- 行线和列线被选中, 写使能。读无效。  $M_5$ ,  $M_6$ ,  $T_7$ ,  $T_8$  导通, 数据线上为需要写入的数据, 且具有较强的驱动能力, 强制存储单元改变状态





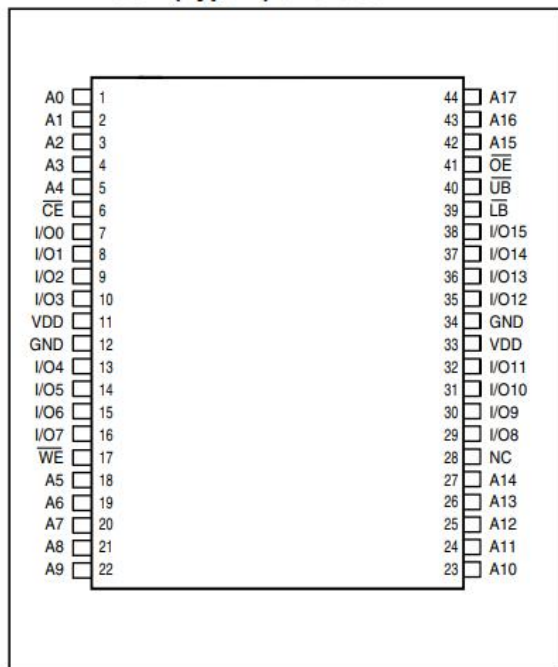
## 4 SRAM 的工作原理

### ● IS61LV25616AL

- 共有 4,194,304-bit 个存储单元  $262,144 \times 16$  bits

#### PIN CONFIGURATIONS

44-Pin TSOP (Type II) and SOJ



#### PIN DESCRIPTIONS

A0-A17	Address Inputs
I/O0-I/O15	Data Inputs/Outputs
$\overline{CE}$	Chip Enable Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
$\overline{LB}$	Lower-byte Control (I/O0-I/O7)
$\overline{UB}$	Upper-byte Control (I/O8-I/O15)
NC	No Connection
VDD	Power
GND	Ground





## 4 SRAM的工作原理

### ● IS61LV25616AL

#### TRUTH TABLE

Mode						I/O PIN		V <sub>DD</sub> Current
	$\overline{WE}$	$\overline{CE}$	$\overline{OE}$	$\overline{LB}$	$\overline{UB}$	I/O0-I/O7	I/O8-I/O15	
Not Selected	X	H	X	X	X	High-Z	High-Z	I <sub>SB1</sub> , I <sub>SB2</sub>
Output Disabled	H	L	H	X	X	High-Z	High-Z	I <sub>CC</sub>
	X	L	X	H	H	High-Z	High-Z	
Read	H	L	L	L	H	DOUT	High-Z	I <sub>CC</sub>
	H	L	L	H	L	High-Z	DOUT	
	H	L	L	L	L	DOUT	DOUT	
Write	L	L	X	L	H	DIN	High-Z	I <sub>CC</sub>
	L	L	X	H	L	High-Z	DIN	
	L	L	X	L	L	DIN	DIN	



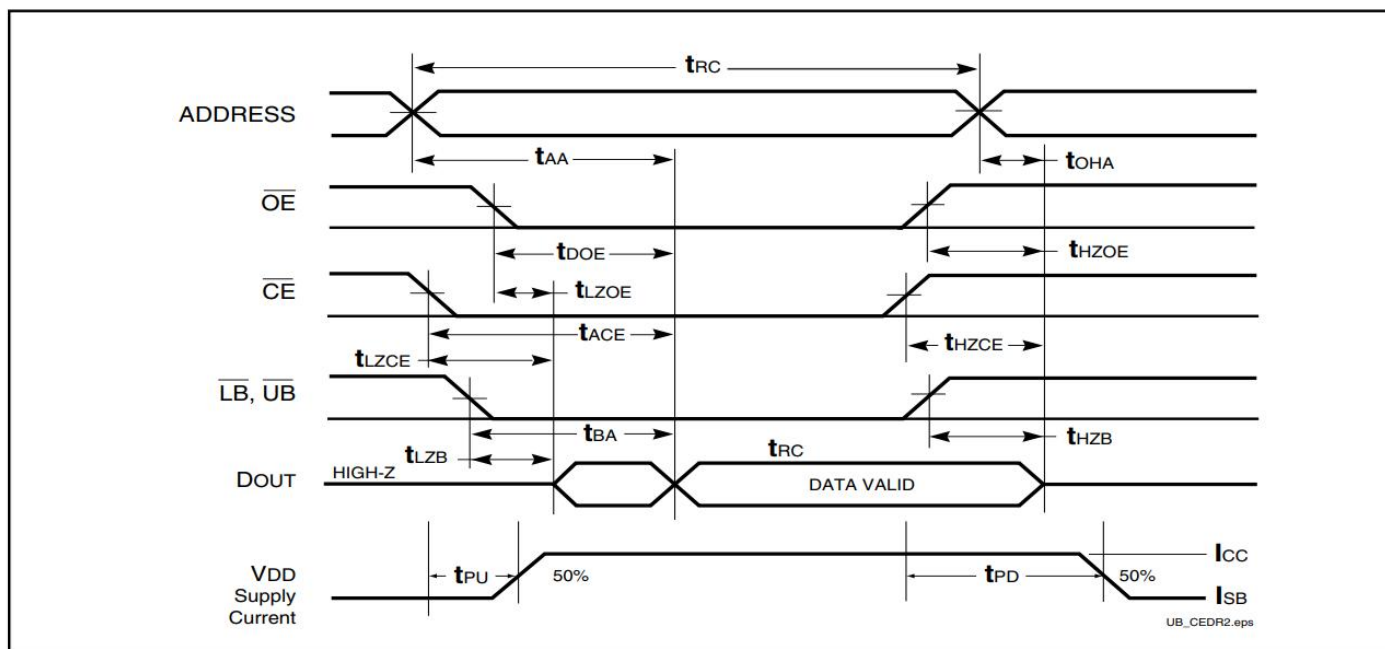
## 4 SRAM 的工作原理

### ● IS61LV25616AL

#### ■ 读周期

$\overline{\text{WE}}$  为高

READ CYCLE NO. 2<sup>(1,3)</sup>





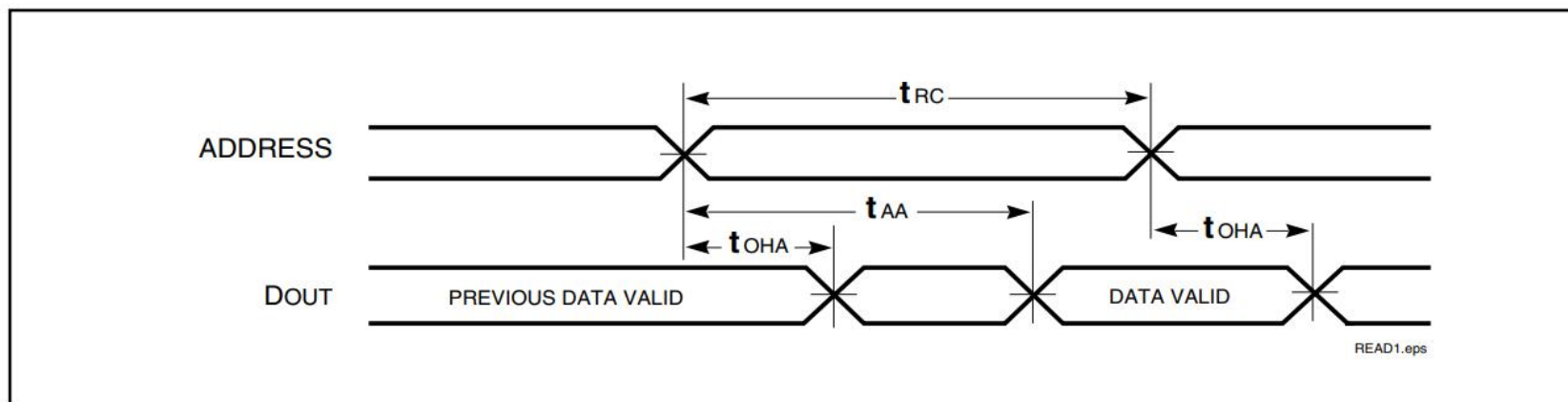


## 4 SRAM 的工作原理

### ● IS61LV25616AL

#### ■ 读周期

**READ CYCLE NO. 1<sup>(1,2)</sup>** (Address Controlled) ( $\overline{CE} = \overline{OE} = V_{IL}$ ,  $\overline{UB}$  or  $\overline{LB} = V_{IL}$ )





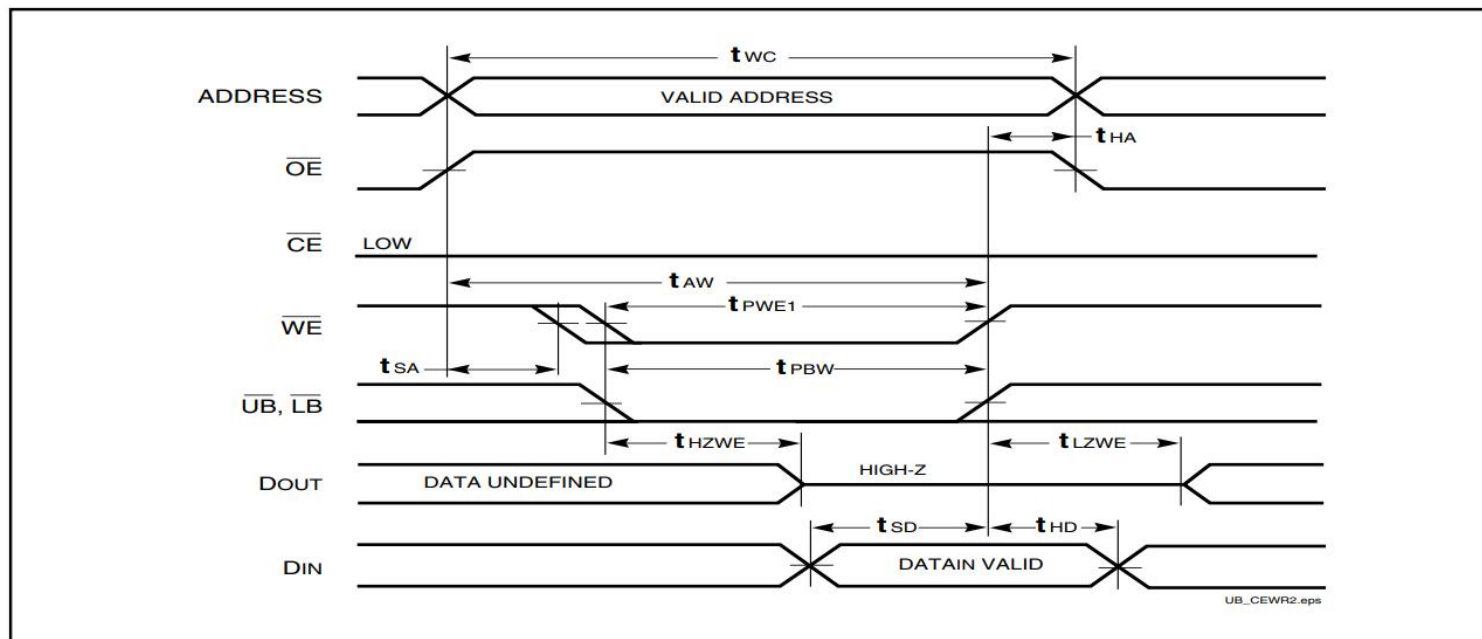


## 4 SRAM 的工作原理

### ● IS61LV25616AL

#### ■ 写周期

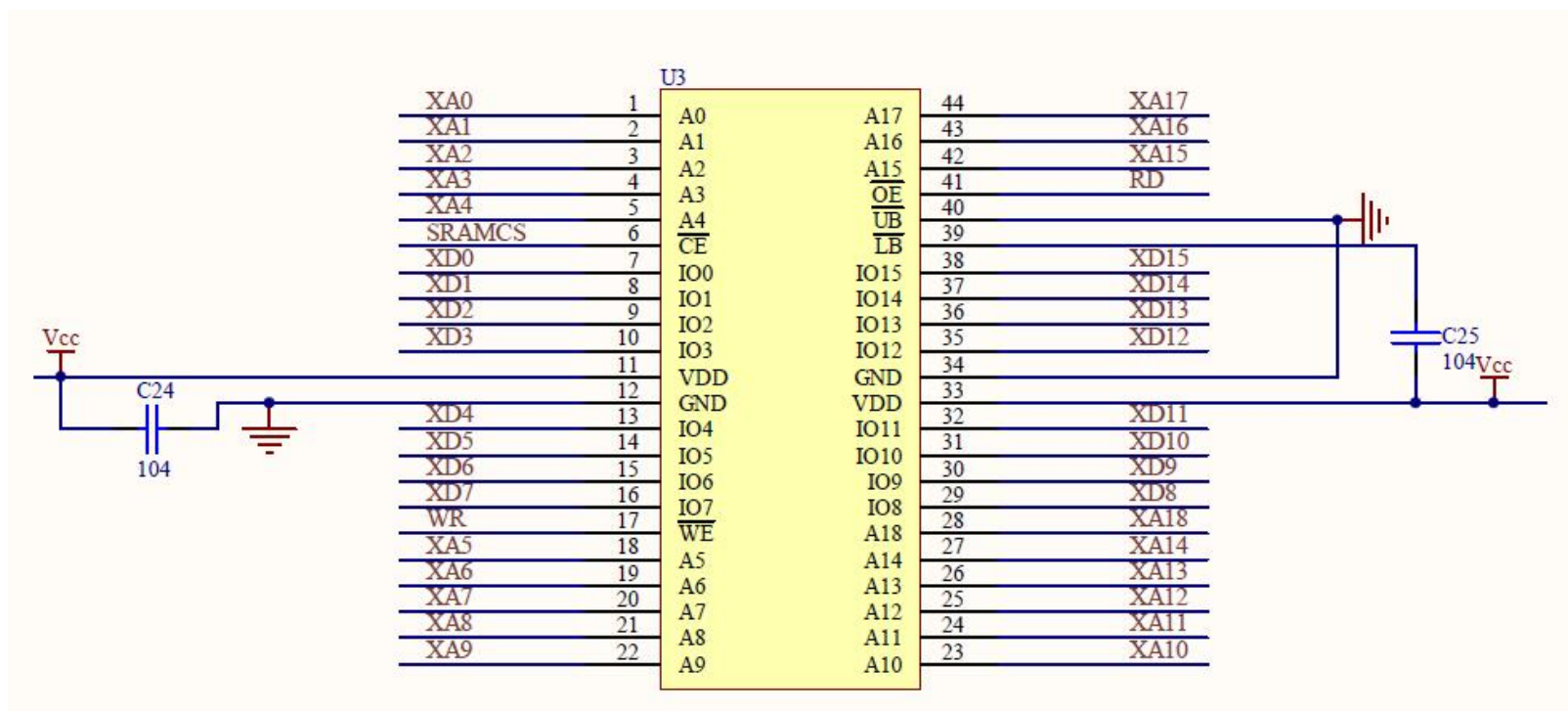
**WRITE CYCLE NO. 2** ( $\overline{WE}$  Controlled.  $\overline{OE}$  is HIGH During Write Cycle) <sup>(1,2)</sup>





## 4 SRAM 的工作原理

### ● 与CPU的连接





## 4 DRAM 的工作原理

- DRAM (Dynamic Random Access Memory)，即动态随机存储器，也就是我们常说的计算机内存，在现代计算机系统和SOC系统中有很重要的作用

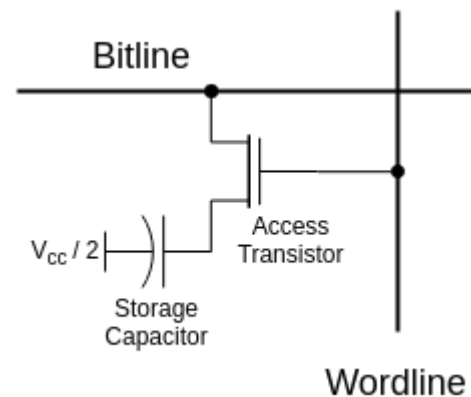
存储电容(Storage Capacitor),它通过存储在其中的电荷的多和少,或者说电容两端电压差的高和低,来表示逻辑上的 1 和 0。存储电容的 Common 端接在  $V_{cc}/2$ 。

- 当存储电容存储的信息为 1 时,另一端电压为  $V_{cc}$ ,此时其所存储的电荷
- $Q = +V_{cc}/2 / C$
- 当存储电容存储的信息为 0 时,另一端电压为 0,此时其所存储的电荷
- $Q = -V_{cc}/2 / C$

访问晶体管(Access Transistor)，它的导通和截止,决定了允许或禁止对存储电容所存储的信息的读取和改写。

字线(Wordline)，它决定了访问晶体管的导通或者截止。

位线(Bitline)，它是外界访问存储电容的唯一通道,当访问晶体管导通后,外界可以通过位线对存储电容进行读取或者写入操作。





## 4 DRAM的工作原理

### ● 数据读写原理

- 读数据时，**字线**设为逻辑高电平，打开**访问晶体管**，然后读取位线上的状态
- 写数据时，先把要写入的电平状态设定到**位线**上，然后打开**访问晶体管**，通过位线改变**存储电容**内部的状态

### ● 问题

- 外界的逻辑电平与 Storage Capacitor 的电平不匹配

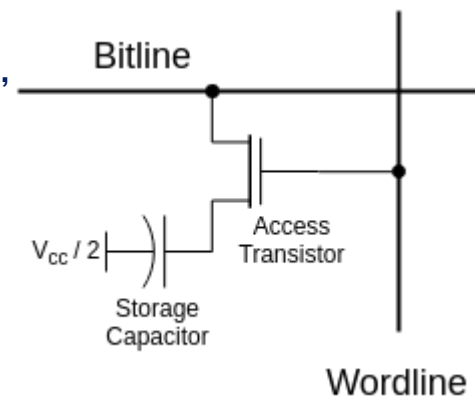
——由于位线的电容值比存储电容要大的多（通常为 10 倍以上），当访问晶体管导通后，如果存储电容存储的信息为 1 时，位线电压变化非常小。外界电路无法直接通过位线来读取存储电容所存储的信息

- 进行一次读取操作后，存储电容存储的电荷会变化

——在进行一次读取操作的过程中，访问晶体管导通后，由于位线存储电容端的电压不一致，会导致存储电容中存储的电荷量被改变。最终可能会导致在下一次读取操作过程中，无法正确的判断存储电容内存储的信息

- 由于存储电容的物理特性，即使不进行读写操作，其所存储的电荷都会慢慢变少

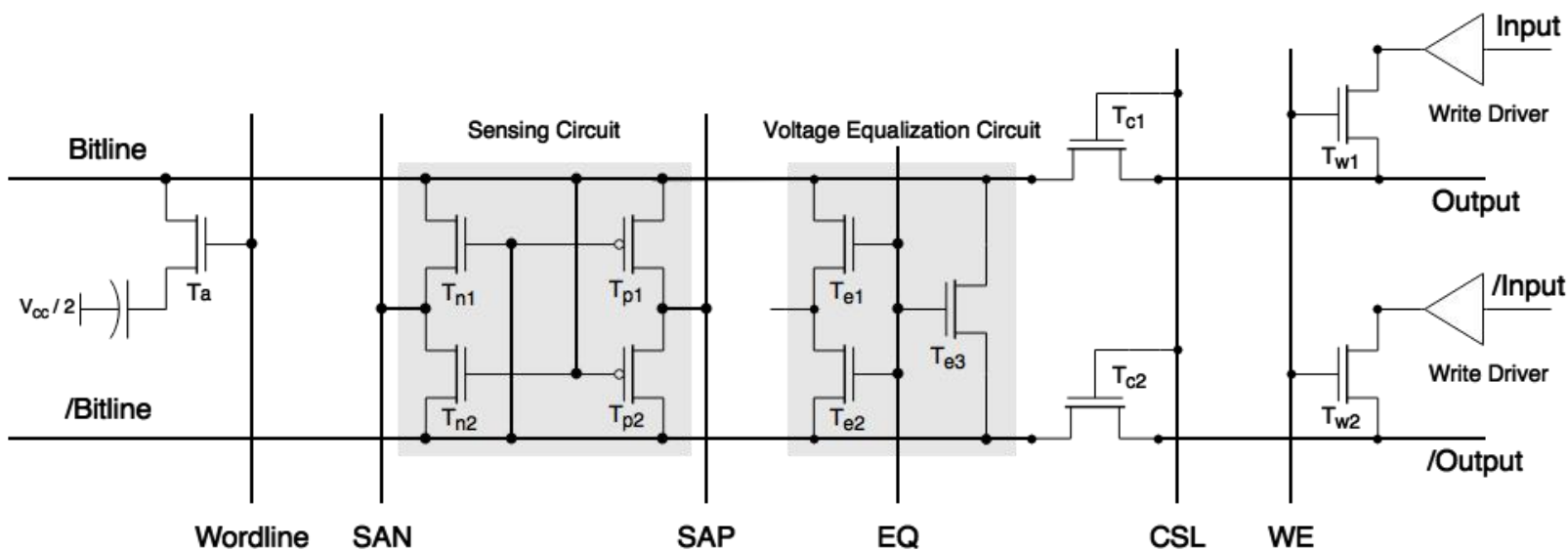
——这个特性要求 DRAM 在没有读写操作时，也要主动对存储电容进行电荷恢复的操作





## 4 DRAM的工作原理

### ● DRAM读写电路



- 差分读取放大器（Differential Sense Amplifier）包含 读出电路（Sensing Circuit）和电压均衡电路（Voltage Equalization Circuit）两个主要部分。它主要的功能：
  - 将存储电容存储的信息转换为逻辑 1 或者 0 所对应的电压，并且呈现到位线上
  - 同时，在完成一次读取操作后，通过位线将存储电容中的电荷恢复到读取之前的状态



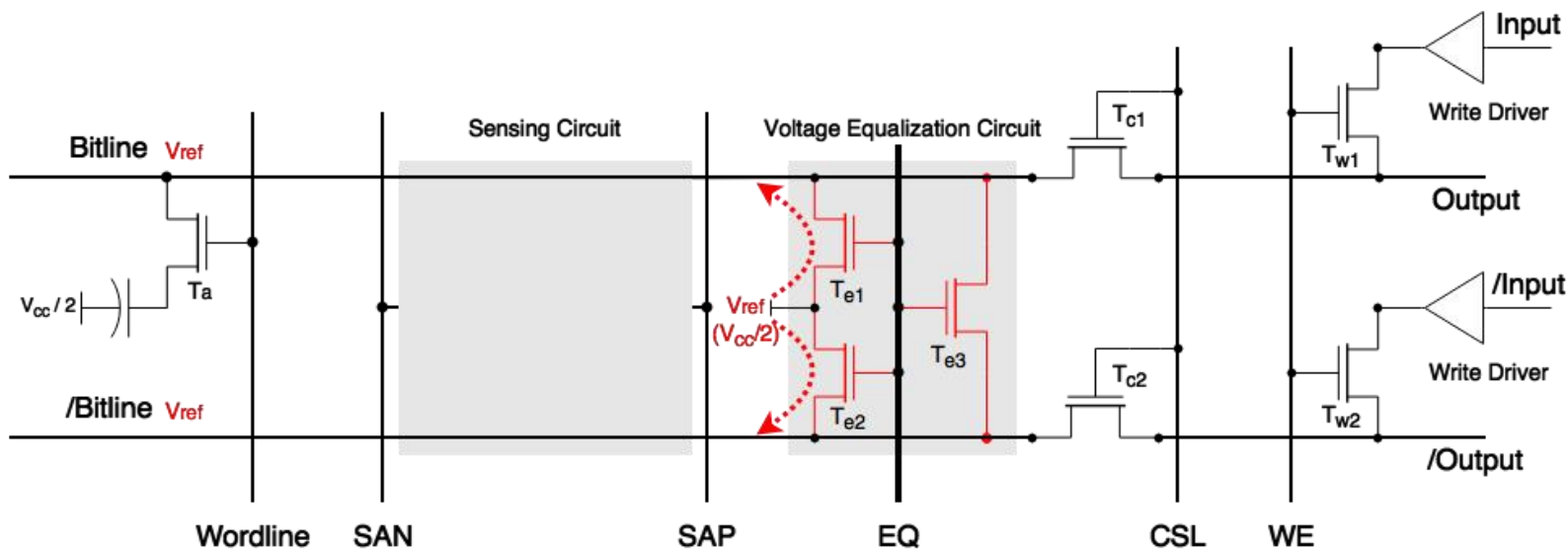
## 4 DRAM 的工作原理

### ● DRAM读写电路

■ 一个完整的 Read Operation 包含了, Precharge、Access、Sense、Restore 四个阶段

■ Precharge:

——在这个阶段, 首先会通过控制 EQ 信号, 让  $T_{e1}$ 、 $T_{e2}$ 、 $T_{e3}$  晶体管处于导通状态, 将 Bitline 和 /Bitline 线上的电压稳定在  $V_{ref}$  上,  $V_{ref} = V_{cc}/2$ 。然后进入到下一个阶段



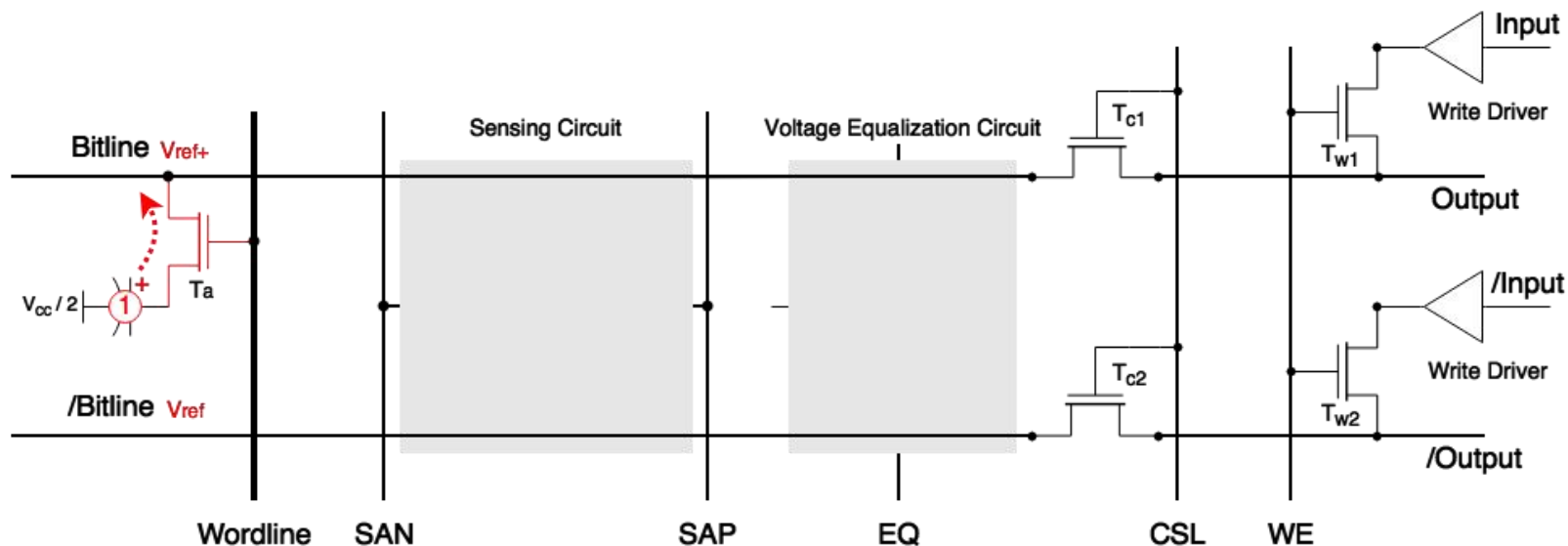


## 4 DRAM的工作原理

### ● DRAM读写电路

#### ■ Access:

——经过 Precharge 阶段，Bitline 和 /Bitline 线上的电压已经稳定在  $V_{ref}$  上了，此时，通过控制 Wordline 信号，将  $T_a$  晶体管导通。Storage Capacitor 中存储正电荷会流向 Bitline，继而将 Bitline 的电压拉升到  $V_{ref+}$ 。然后进入到下一个阶段





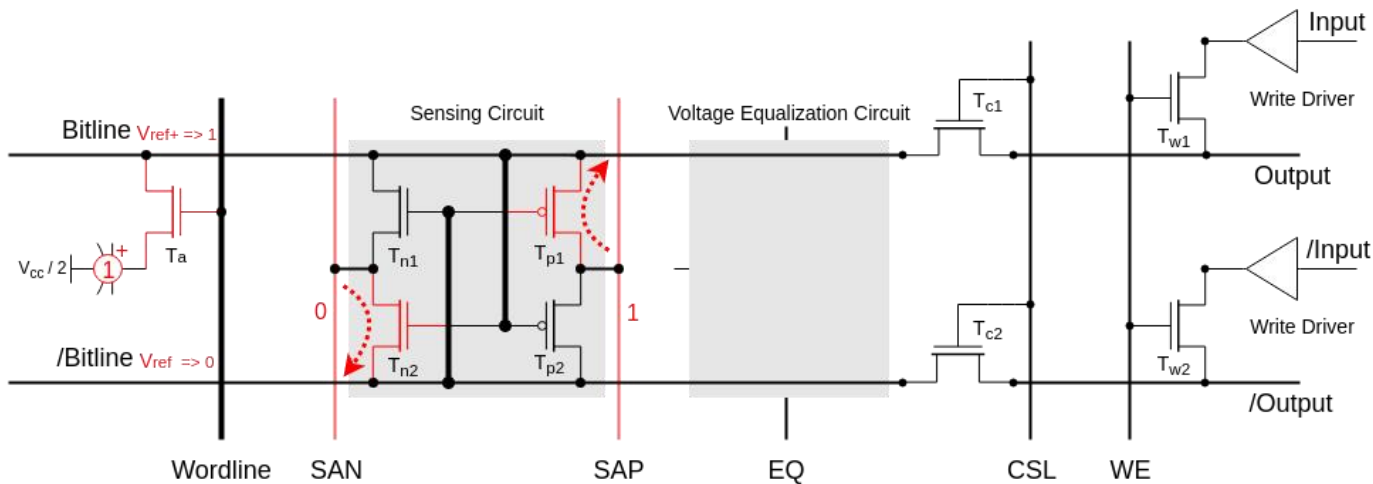


## 4 DRAM 的工作原理

### ● DRAM 读写电路

#### ■ Sense:

- 由于在 Access 阶段，Bitline 的电压被拉升到  $V_{ref+}$ ， $T_{n2}$  会比  $T_{n1}$  更具导通性， $T_{p1}$  则会比  $T_{p2}$  更具导通性
- 此时，SAN (Sense-Amplifier N-Fet Control) 会被设定为逻辑 0 的电压，SAP (Sense-Amplifier P-Fet Control) 则会被设定为逻辑 1 的电压，即  $V_{cc}$ 。由于  $T_{n2}$  会比  $T_{n1}$  更具导通性，/Bitline 上的电压会更快被 SAN 拉到逻辑 0 电压，同理，Bitline 上的电压也会更快被 SAP 拉到逻辑 1 电压。接着  $T_{p1}$  和  $T_{n2}$  进入导通状态， $T_{p2}$  和  $T_{n1}$  进入截止状态
- 最后，Bitline 和 /Bitline 的电压都进入稳定状态，正确的呈现了 Storage Capacitor 所存储的信息 Bit





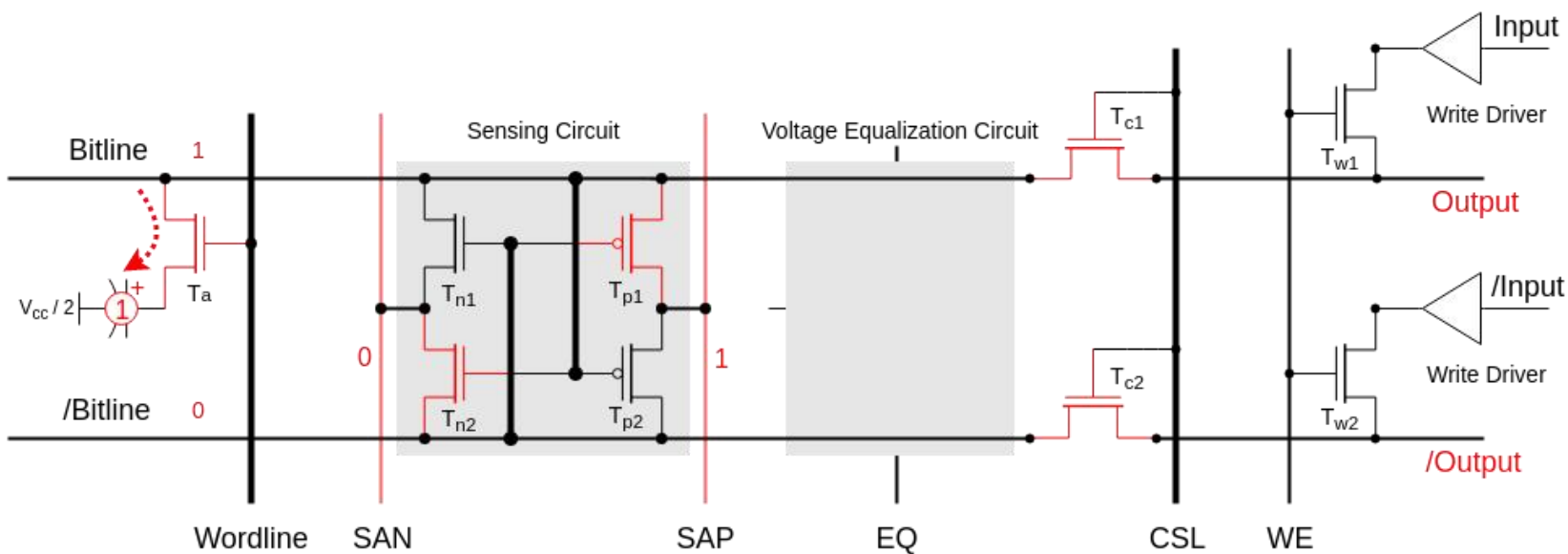


## 4 DRAM 的工作原理

### ● DRAM 读写电路

#### ■ Restore:

——在完成 Sense 阶段的操作后，Bitline 线处于稳定的逻辑 1 电压  $V_{cc}$ ，此时 Bitline 会对 Storage Capacitor 进行充电。经过特定的时间后，Storage Capacitor 的电荷就可以恢复到读取操作前的状态

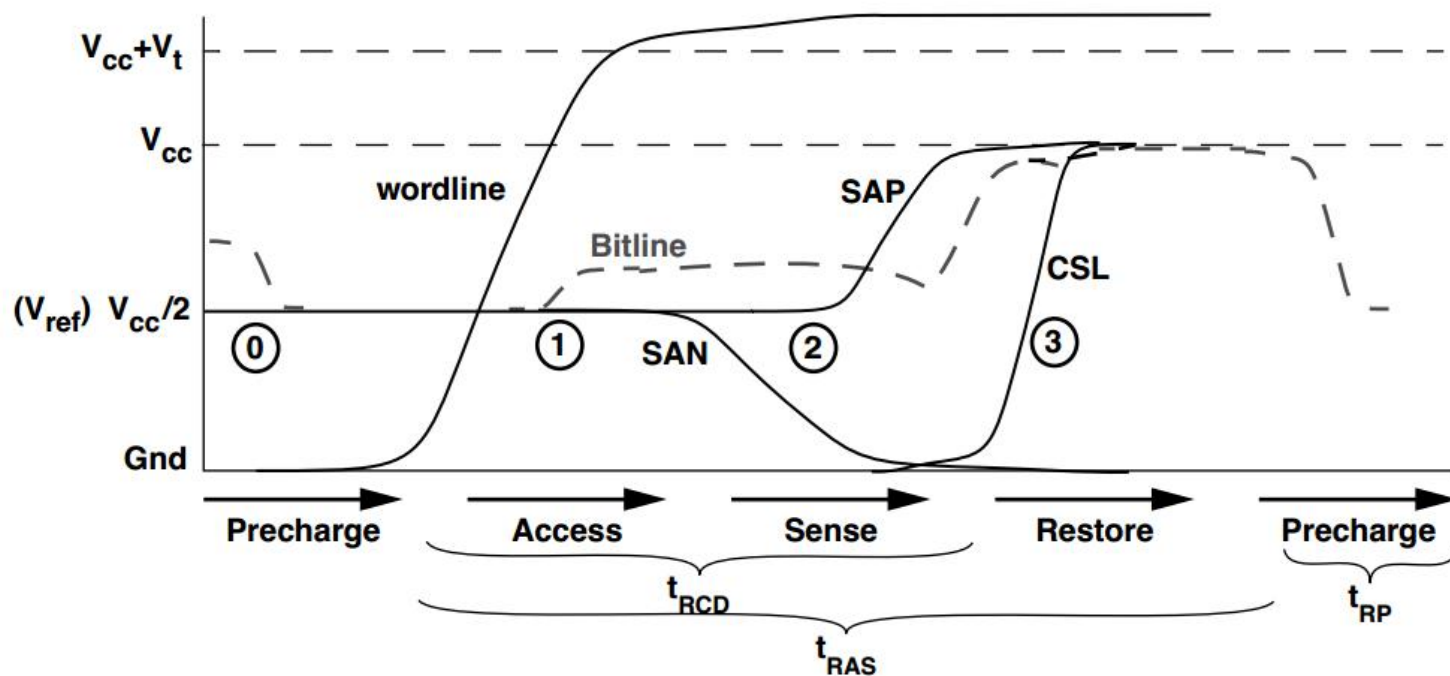




## 4 DRAM的工作原理

### ● DRAM读写电路

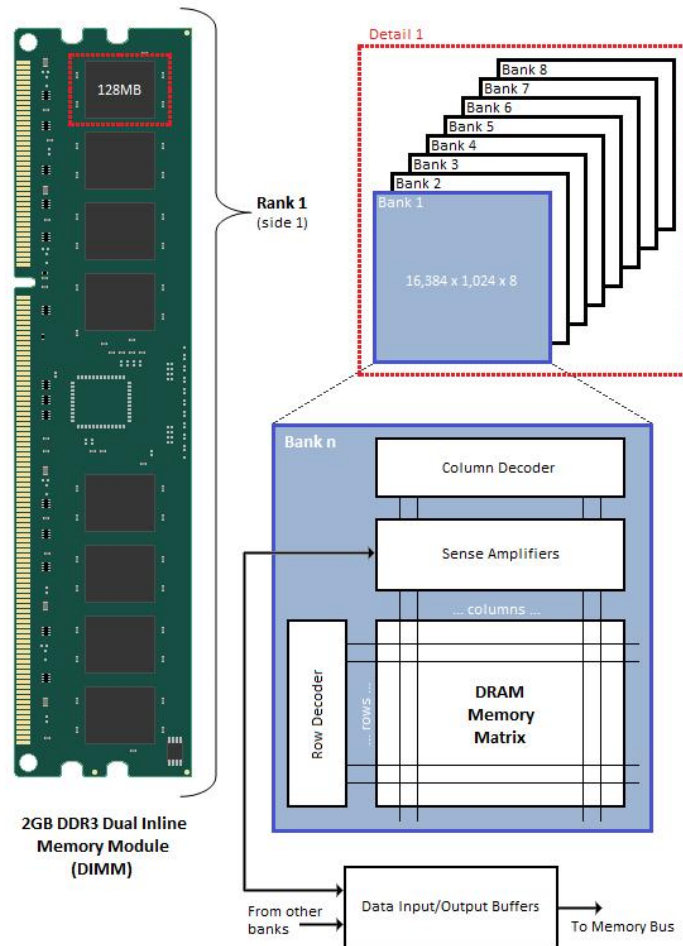
#### ■ 时序图





## 4 DRAM的工作原理

### ● DRAM存储矩阵的结构





## 4 DRAM 的工作原理

### ● 集中刷新

- 在刷新周期内，对全部存储单元集中一段时间进行刷新（逐行进行），此时必须停止读写操作

### ● 异步刷新

- 在刷新周期内对所有行各刷新一次，即每隔（刷新周期/总行数） $\mu\text{s}$ ，刷新一次。能充分利用刷新周期，提高刷新频率



## 4 DRAM的工作原理

### ● DRAM的特点

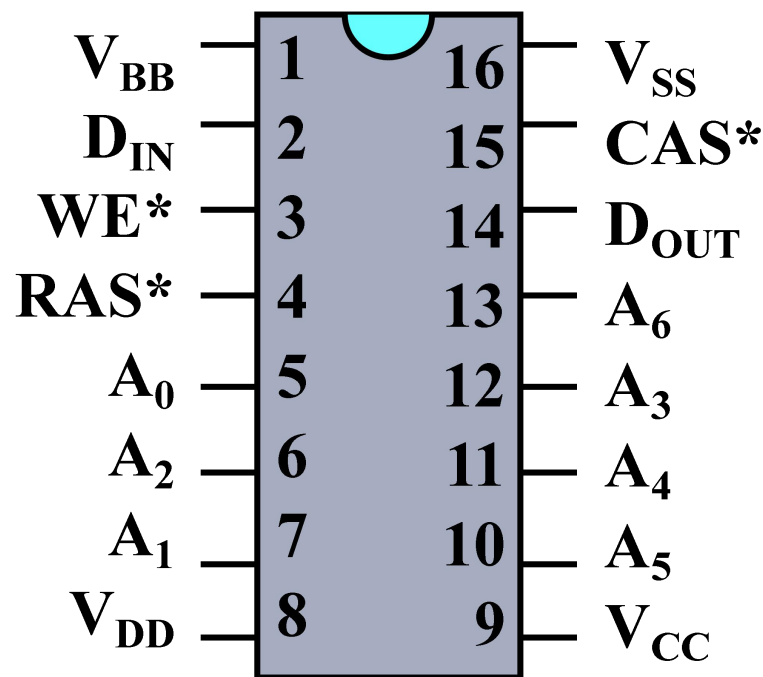
- 与SRAM相比，DRAM所用的mos管少，占硅面积小，因而功耗小，集成度高
- 但是因为采用了电容存储电荷的原理来寄存信息，会发生漏电现象，所以要保持状态不变，需要定时刷新，因为读操作会使得状态发生改变，故需要读后再生。且速度比SRAM慢。
- 但是由于其功耗小，集成度高，被广泛应用于计算机中
- 需要有专门的控制器与CPU连接



## 4 DRAM 的工作原理

### ● DRAM4116

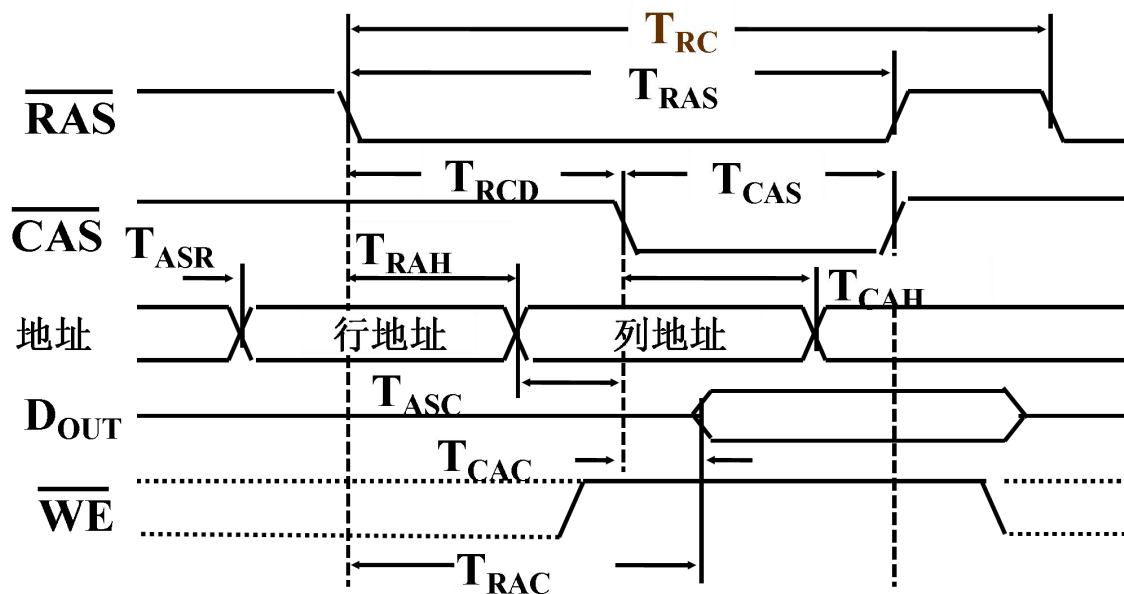
- 存储容量为  $16K \times 1$
- 16个引脚：
  - 7根地址线  $A_6 \sim A_0$
  - 1根数据输入线  $D_{IN}$
  - 1根数据输出线  $D_{OUT}$
  - 行地址选通  $RAS^*$
  - 列地址选通  $CAS^*$
  - 读写控制  $WE^*$





## 4 DRAM的工作原理

### ● DRAM4116——DRAM读



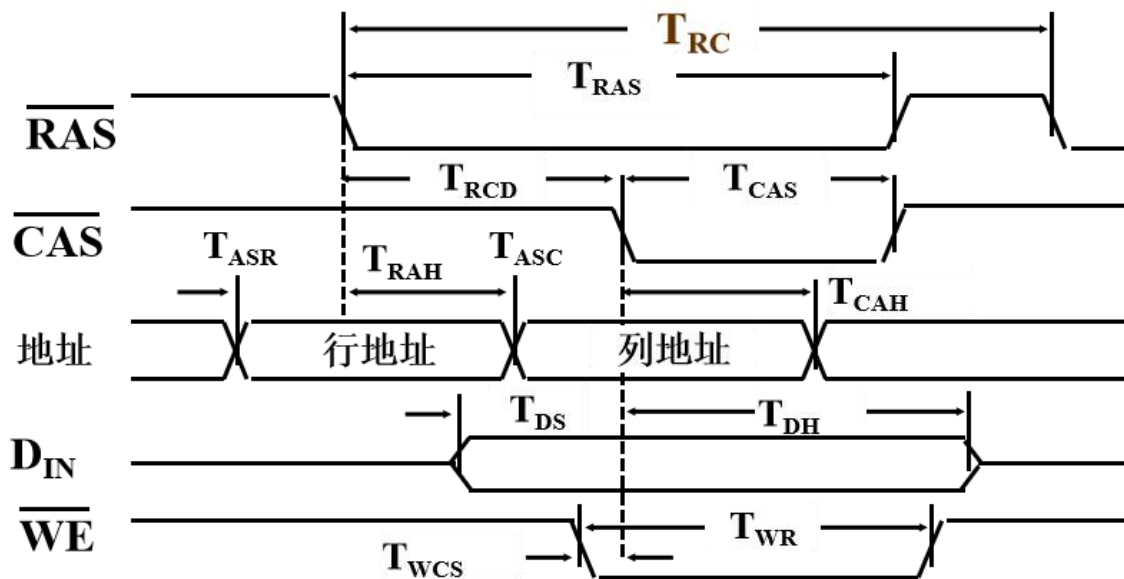
存储地址需要分两批传送

- 行地址选通信号**RAS\***有效, 开始传送行地址
- 随后, 列地址选通信号**CAS\***有效, 传送列地址, **CAS\***相当于片选信号
- 读写信号**WE\***读有效
- 数据从**D<sub>OUT</sub>**引脚输出



## 4 DRAM的工作原理

### ● DRAM4116——DRAM写



存储地址需要分两批传送

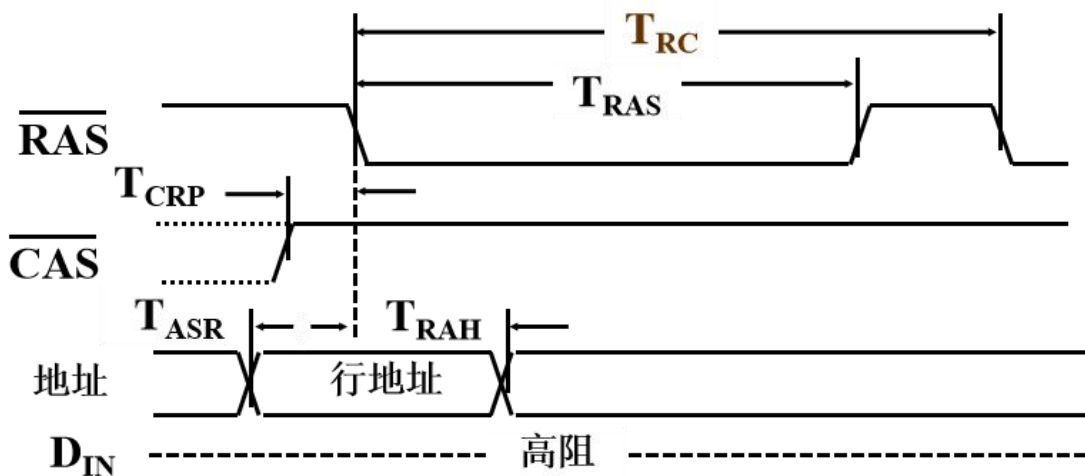
- 行地址选通信号**RAS\***有效，开始传送行地址
- 随后，列地址选通信号**CAS\***有效，传送列地址
- 读写信号**WE\***写有效
- 数据从**D<sub>IN</sub>**引脚进入存储单元





## 4 DRAM的工作原理

### ● DRAM4116——DRAM刷新



采用“仅行地址有效”方法刷新

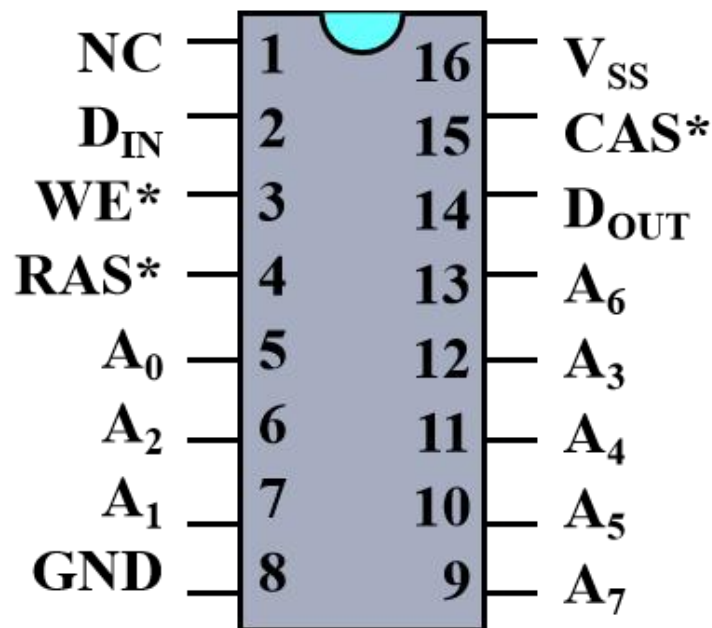
- 行地址选通**RAS\***有效，传送行地址
- 列地址选通**CAS\***无效，没有列地址
- 芯片内部实现一行存储单元的刷新
- 没有数据输入输出
- 存储系统中所有芯片同时进行刷新
- **DRAM**必须每隔固定时间就刷新



## 4 DRAM的工作原理

### ● DRAM2164

- 存储容量为 $64K \times 1$
- 16个引脚：
  - 8根地址线 $A_7 \sim A_0$
  - 1根数据输入线 $D_{IN}$
  - 1根数据输出线 $D_{OUT}$
  - 行地址选通 $RAS^*$
  - 列地址选通 $CAS^*$
  - 读写控制 $WE^*$





## 5 EPROM 的介绍

### ● 紫外线擦除可编程ROM (EPROM) 介绍

- 顶部开有一个圆形的石英窗口，用于紫外线透过擦除原有信息
- 一般使用专门的编程器（烧写器）进行编程
- 编程后，应该贴上不透光封条
- 出厂未编程前，每个基本存储单元都是信息1
- 编程就是将某些单元写入信息0

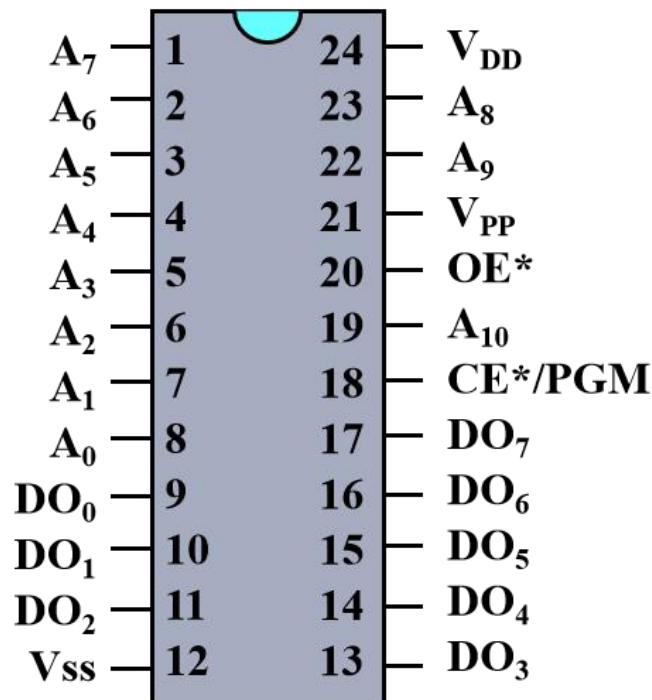




## 5 EPROM的介绍

### ● EPROM芯片2716

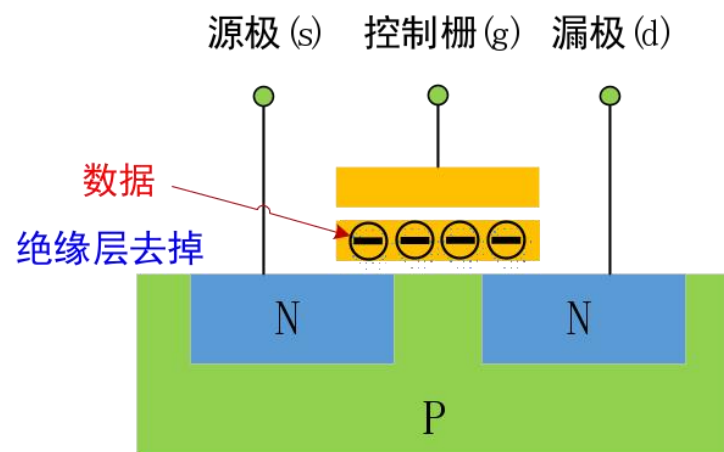
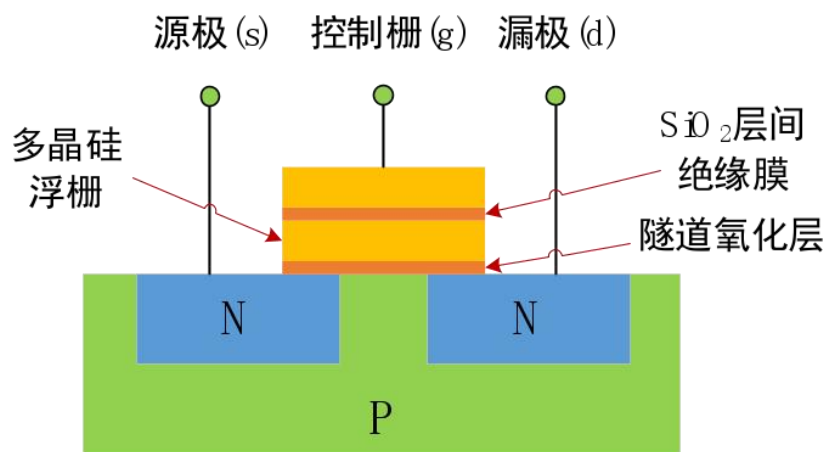
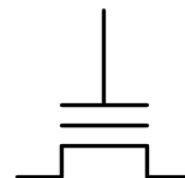
- 存储容量为 $2K \times 8$
- 24个引脚：
  - 11根地址线 $A_{10} \sim A_0$
  - 8根数据线 $DO_7 \sim DO_0$
  - 片选/编程 $CE^*/PGM$
  - 读写 $OE^*$
  - 编程电压 $V_{PP}$





## 6 EEPROM

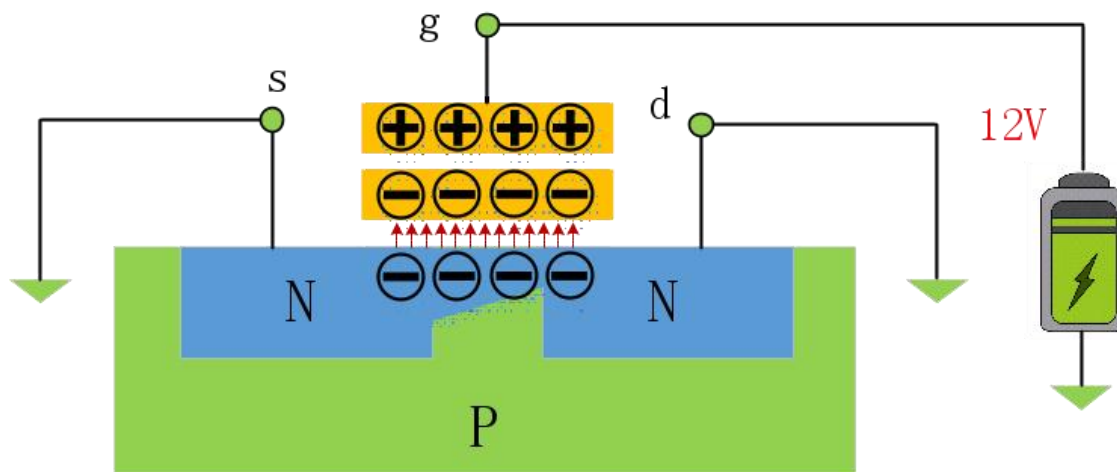
- EEPROM (Electrically Erasable Programmable read only memory)
  - 即电可擦可编程只读存储器，是一种掉电后数据不丢失（不挥发）存储芯片
- EEPROM使用浮栅场效应管(Floating Gate FET)作为基本存储单元来存储数据





## 6 EEPROM

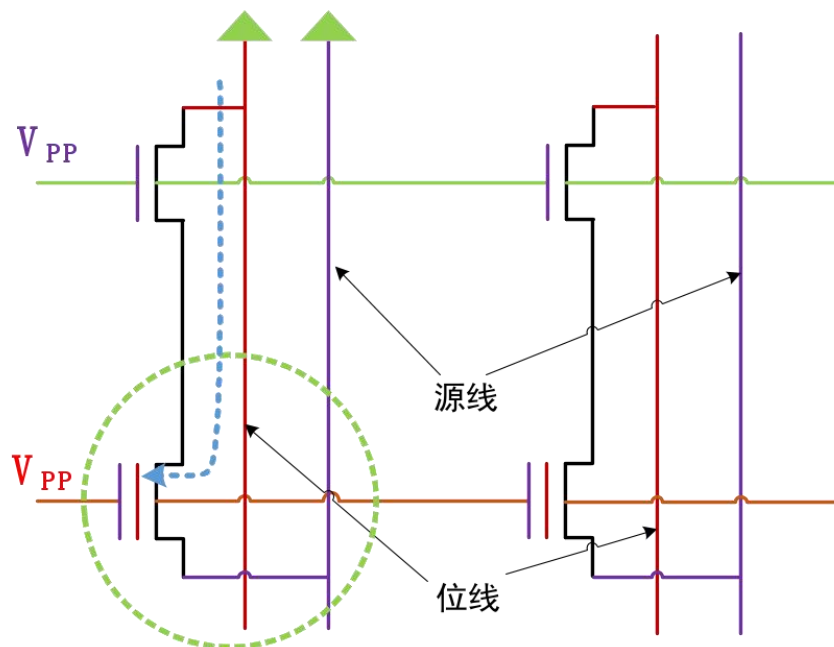
- EEPROM (Electrically Erasable Programmable read only memory)
  - 通常利用F-N隧道效应 (Fowler-Nordheim tunneling) 或热电子注入对EEPROM存储单元进行“擦除”或“写入”操作
- 对EEPROM存储单元进行“擦除”操作，就是将电子注入到浮栅中的过程





## 6 EEPROM

- 对EEPROM存储单元进行“擦除”操作，就是将电子注入到浮栅中的过程
  - 进行存储单元的“擦除”操作时，源线（Sources Lines, SL）与位线（Bit Lines, BL）均为低电平，而控制栅线（CL）为高电平，当对应存储单元的选通管打开时（Word Line, WL= $V_{PP}$ ），如下图所示

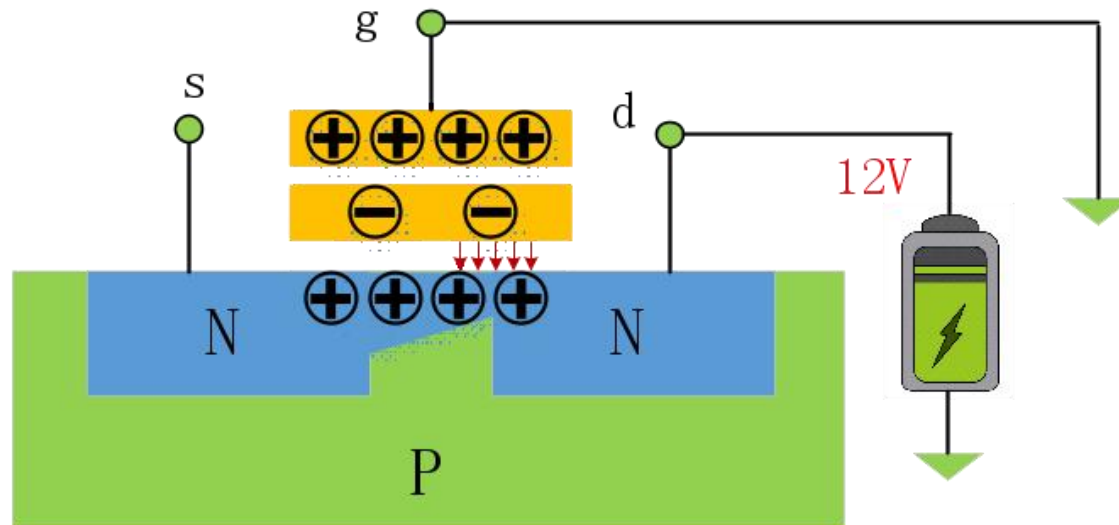






## 6 EEPROM

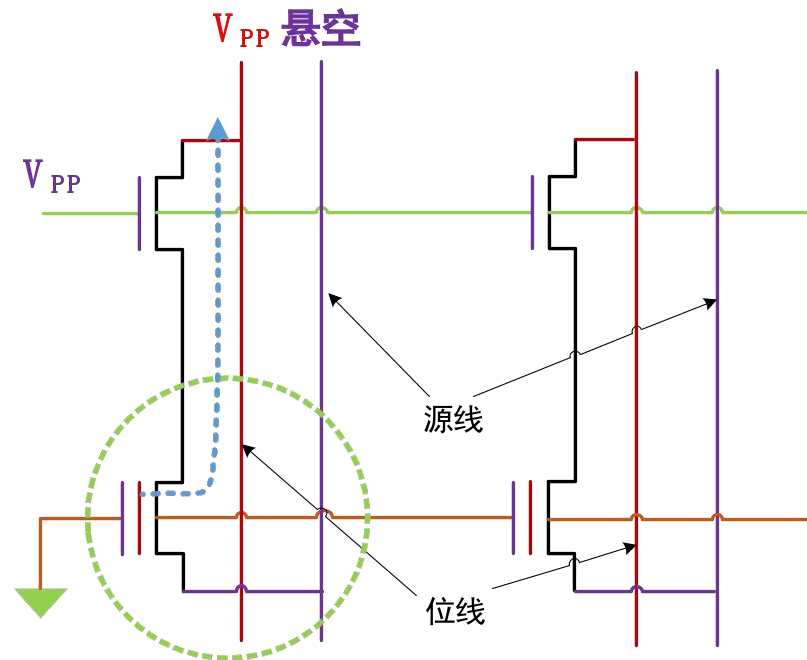
- 对EEPROM存储单元进行“写入”操作，就是将浮栅中电子释放的过程





## 6 EEPROM

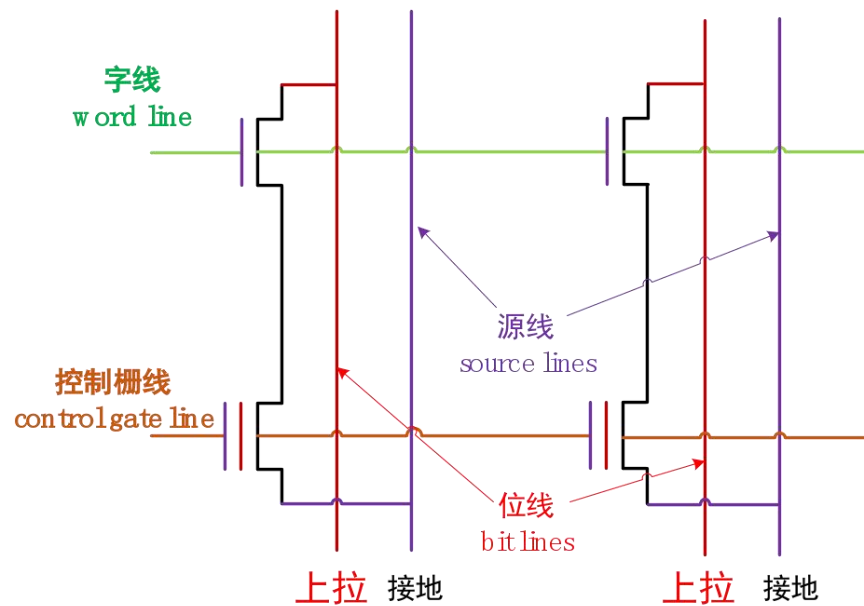
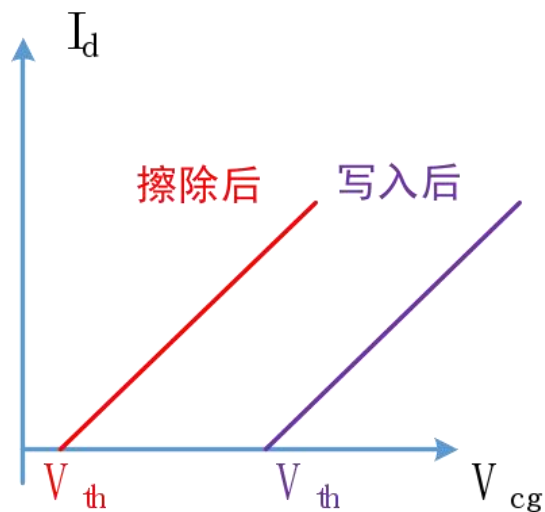
- 对EEPROM存储单元进行“写入”操作，就是将浮栅中电子释放的过程
  - 进行存储单元的“写入”操作时，位线为高电平，而源线悬空且CL为低电平，当对应存储单元的选通管打开时（ $SG=V_{PP}$ ），如下图所示





## 6 EEPROM

- 读取时位线被上拉，给控制栅极加一个中间电压，如果浮动栅极有电荷，位线读出1，否则位线读出0

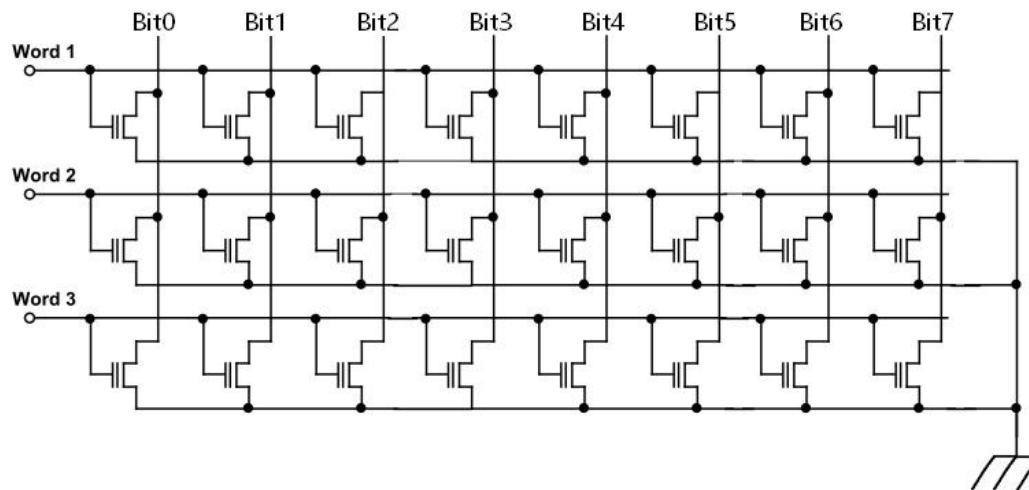




## 6 Flash

### ● 快闪存储器 (Flash Memory)

- 全名叫Flash EEPROM Memory，是一种电子式可清除程序化只读存储器的形式，允许在操作中被多次擦或写的存储器
- Flash又分为NOR Flash和NAND Flash两种
- NOR Flash每个Bit Line下的基本存储单元是并联的，当某个Word Line被选中后，就可以实现对该Word的读取，也就是可以实现位读取（即Random access），且具有较高的读取速率

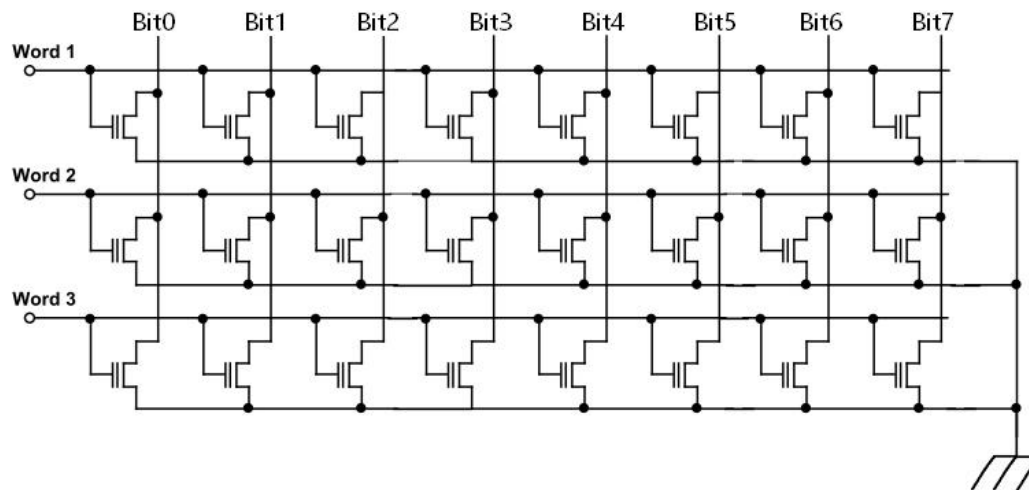




## 6 Flash

### ● NOR Flash

- NOR FLASH中的N是NOT，含义是Floating Gate中有电荷时，读出‘0’，无电荷时读出‘1’，是一种‘非’的逻辑；OR的含义是同一个Bit Line下的各个基本存储单元是并联的，是一种‘或’的逻辑，这就是NOR 的由来





## 6 Flash

### ● NOR Flash

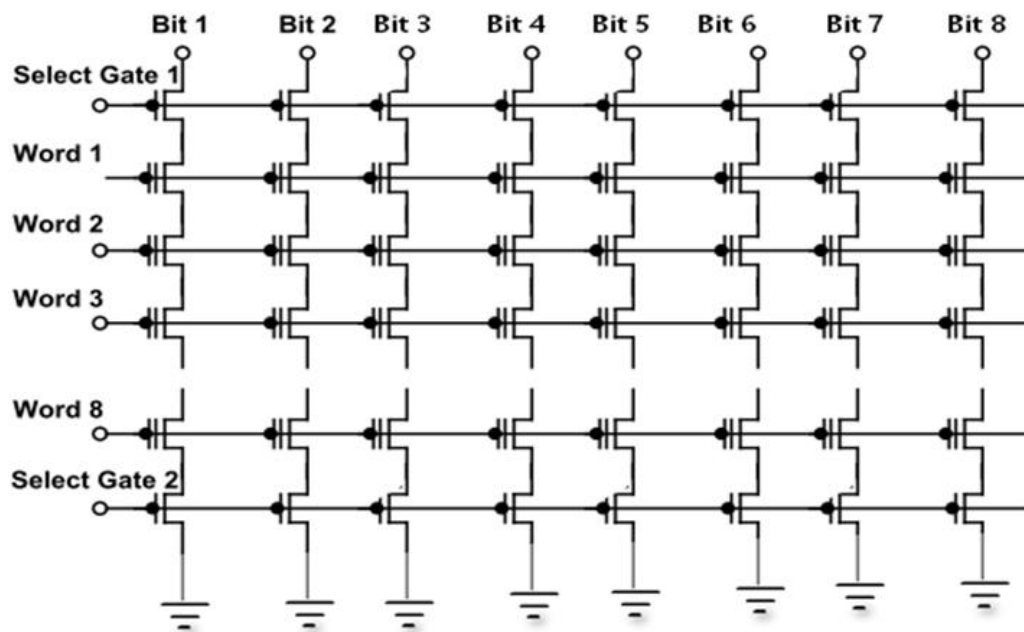
- 基本存储单元的并联结构决定了金属导线占用很大的面积，因此NOR FLASH的存储密度较低，无法适用于需要大容量存储的应用场合，即适用于code-storage，不适用于data-storage
- 基本存储单元的并联结构决定了NOR FLASH具有存储单元可独立寻址且读取效率高的特性，因此适用于code-storage，且程序可以直接在NOR 中运行（即具有RAM的特性）
- NOR FLASH写入采用了热电子注入方式，效率较低，因此NOR写入速率较低，不适用于频繁擦除/写入场合



## 6 Flash

### ● NAND Flash

- NAND Flash每个Bit Line下的基本存储单元是串联的，NAND读取数据的单位是Page
- NAND无法实现位读取（即Random access），程序代码也就无法在NAND上运行







## 6 Flash

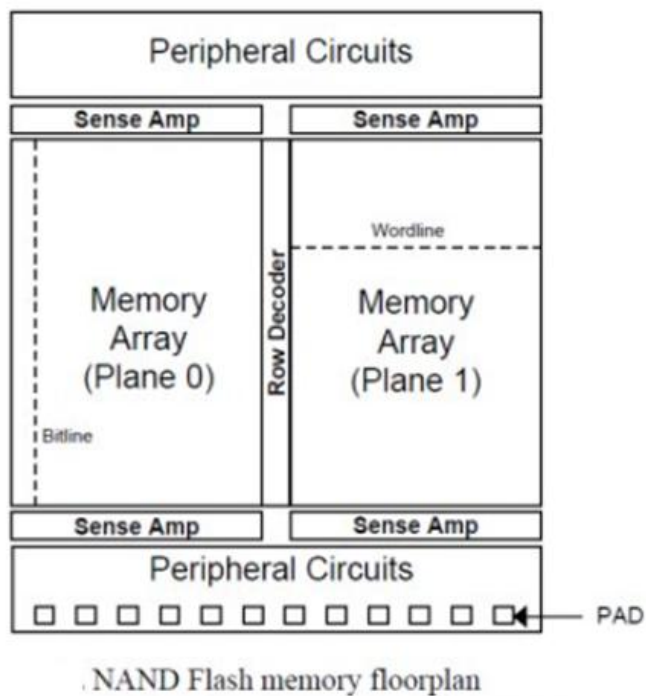
### ● NAND Flash

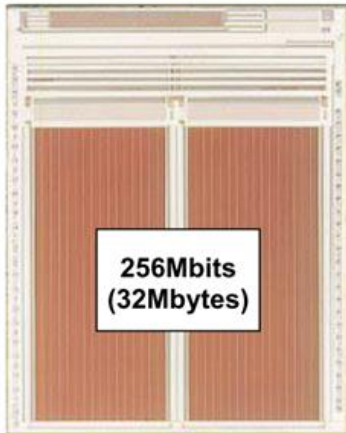

- 基本存储单元的串联结构减少了金属导线占用的面积，Die的利用率很高，因此NAND FLASH存储密度高，适用于需要大容量存储的应用场合，即适用于data-storage
- 基本存储单元的串联结构决定了NAND FLASH无法进行位读取，也就无法实现存储单元的独立寻址，因此程序不可以直接在NAND 中运行，因此NAND是以Page为读取单位和写入单位，以Block为擦除单位
- NAND FLASH写入采用F-N隧道效应方式，效率较高，因此NAND擦除/写入速率很高，适用于频繁擦除/写入场合。同时NAND是以Page为单位进行读取的，因此读取速率也不算低（稍低于NOR）



## 6 Flash

### ● NAND Flash与NOR Flash对比



32MBytes (256Mbits) DiskOnChip Millennium Plus (NAND-based)	16MBytes (126Mbits) NOR Flash
 256Mbits (32Mbytes)	 128Mbits (16Mbytes)
244% better than NOR	



## 7 TM4C控制器中的存储器

- 256 KB bit-banded SRAM
  - 可以实现位操作，避免读取-修改-写入操作
- internal ROM
  - Boot Loader and vector table
  - 存储外设驱动程序
  - 存储加密标准表
  - 存储冗余校验功能
- 1024 KB Flash memory
  - 四个 256 bits预取寄存器
- 6 KB EEPROM
  - 单个存储单元50万次写入



## 7 TM4C控制器中的存储器

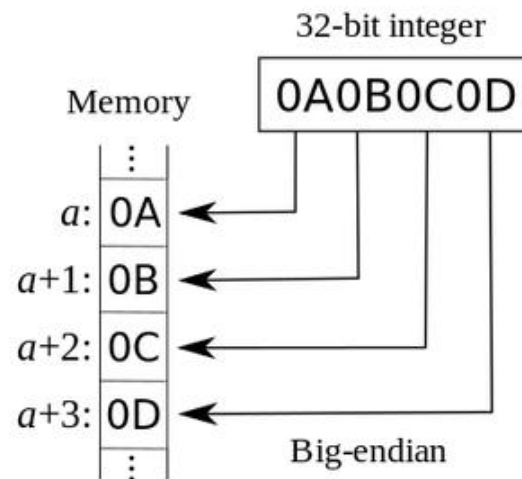
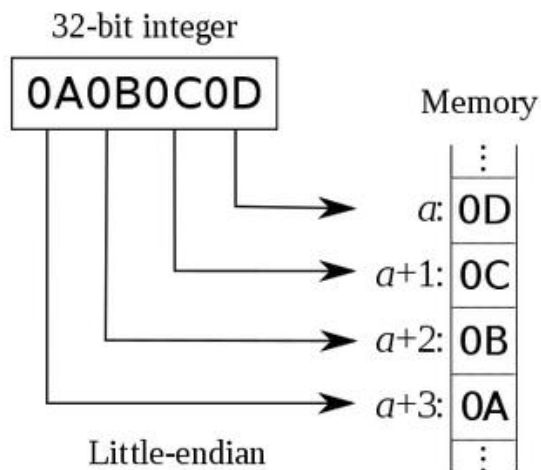
### ● 地址编排

Start	End	Description	For details, see page ...
<b>Memory</b>			
0x0000.0000	0x0007.FFFF	On-chip Flash	621
0x0008.0000	0x01FF.FFFF	Reserved	-
0x0200.0000	0x02FF.FFFF	On-chip ROM (16 MB)	602
0x0300.0000	0x1FFF.FFFF	Reserved	-
0x2000.0000	0x2006.FFFF	Bit-banded on-chip SRAM	602
0x2007.0000	0x21FF.FFFF	Reserved	-
0x2200.0000	0x2234.FFFF	Bit-band alias of bit-banded on-chip SRAM starting at 0x2000.0000	602
0x2235.0000	0x3FFF.FFFF	Reserved	-
<b>Peripherals</b>			
0x4000.0000	0x4000.0FFF	Watchdog timer 0	1028
0x4000.1000	0x4000.1FFF	Watchdog timer 1	1028
0x4000.2000	0x4000.3FFF	Reserved	-



## 8 字节顺序

- 字节顺序，又称端序或尾序（Endianness），在计算机科学领域中，指存储器中或在数字通信链路中，组成多字节的字的字节的排列顺序
  - Little-Endian（小端模式）就是低位字节排放在内存的低地址端，高位字节排放在内存的高地址端
  - Big-Endian（大端模式）就是高位字节排放在内存的低地址端，低位字节排放在内存的高地址端





8

## 字节顺序

- 字节顺序，又称端序或尾序（Endianness），在计算机科学领域中，指存储器中或在数字通信链路中，组成多字节的字的字节的排列顺序
  - Little-Endian（小端模式）就是低位字节排放在内存的低地址端，高位字节排放在内存的高地址端
  - Big-Endian（大端模式）就是高位字节排放在内存的低地址端，低位字节排放在内存的高地址端

`uint32_t a=100000, &a=0x20000010`, 那么，存储变量**a**是如何存储的？存储变量**a**所用的存储单元的地址分别是哪些？这些存储单元里各存储了什么？（存储器以字节为单位编址）

`a=0x186A0`



## 9 半导体存储器与CPU的连接

- 存储芯片的数据线、地址线、片选端、读写控制线？

- 存储芯片的数据线

- 若芯片的数据线正好与CPU数据线数量相同

- 一次可从芯片中访问到8/16/32位数据

- 全部数据线与系统的数据总线相连

- 若芯片的数据线与CPU数据线数量不同

- 一次不能从一个芯片中访问到全部数据

- 利用多个芯片扩充数据位

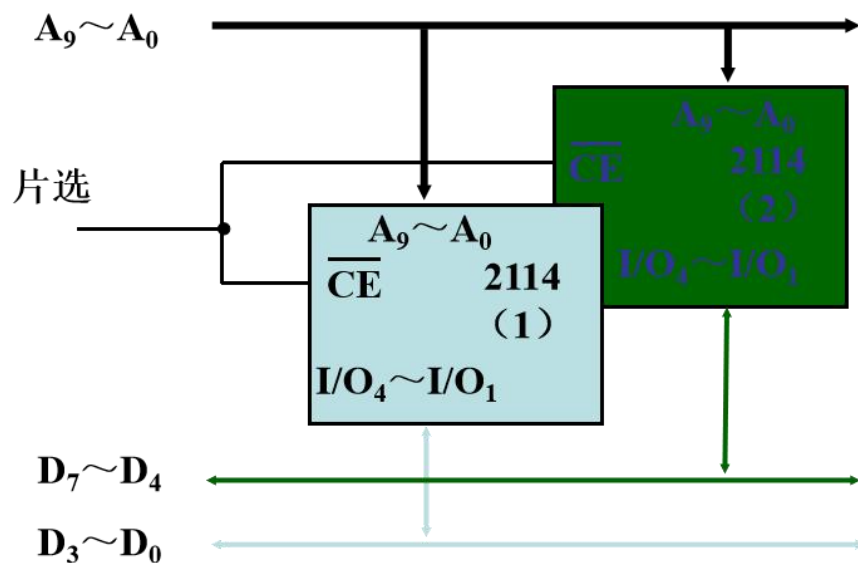
- 这个扩充方式简称“位扩充”





## 9 半导体存储器与CPU的连接

### ● 存储芯片的数据线



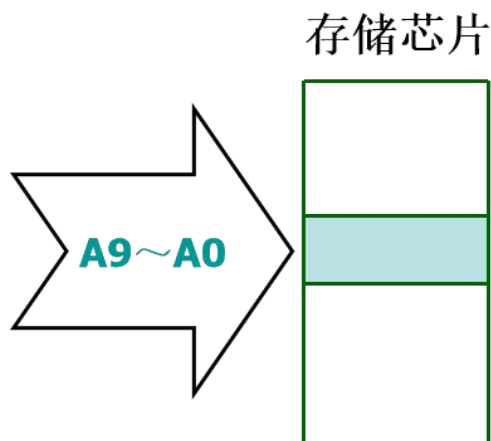
- 多个位扩充的存储芯片的数据线连接于系统数据总线的不同位数
- 其它连接都一样
- 这些芯片应被看作是一个整体
- 常被称为“芯片组”



## 9 半导体存储器与CPU的连接

### ● 存储芯片的地址线

- 芯片的地址线通常应全部与系统的低位地址总线相连
- 寻址时，这部分地址的译码是在存储芯片内完成的，我们称为“片内译码”



$A_9 \sim A_0$		范围（16进制）
00...00	全0	000H
00...01	⋮	001H
00...10		002H
...		...
11...01		3FDH
11...10		3FEH
11...11	全1	3FFH



## 9 半导体存储器与CPU的连接

### ● 片选端译码

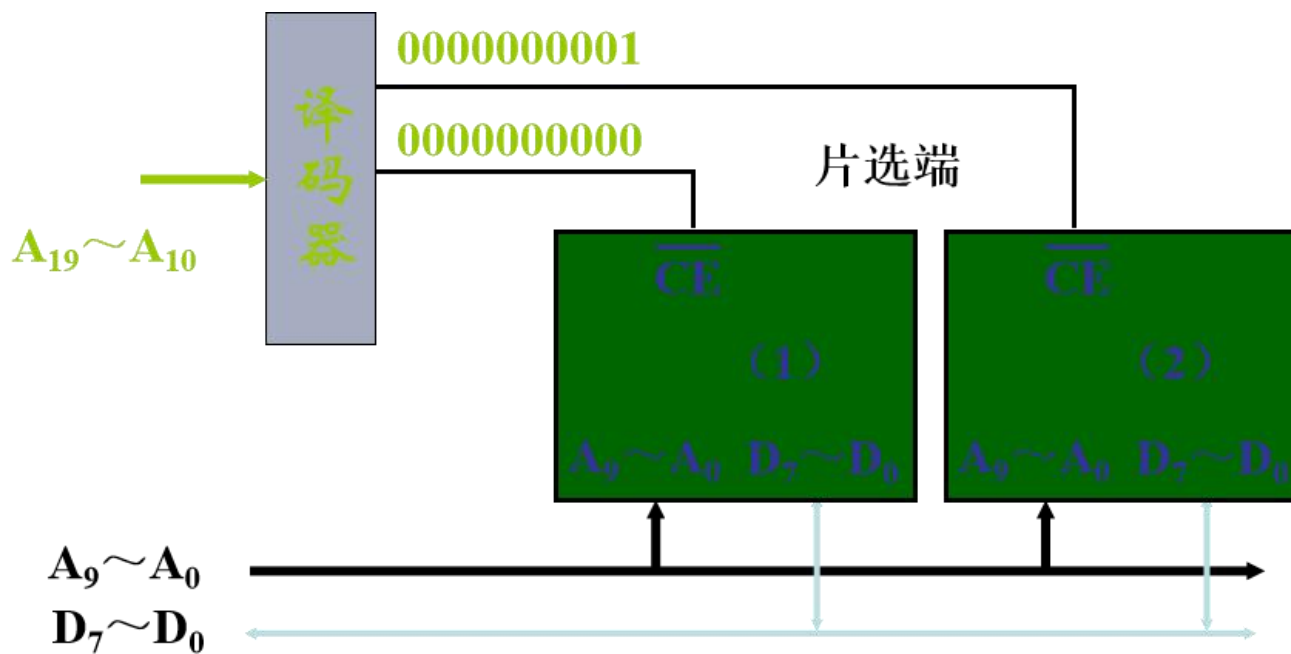
- 存储系统常需利用多个存储芯片扩充容量，也就是扩充了存储器地址范围
- 进行“地址扩充”，需要利用存储芯片的片选端对多个存储芯片（组）进行寻址
- 这个寻址方法，主要通过将存储芯片的片选端与系统的高位地址线相关联来实现
- 这种扩充简称为“地址扩充”或“字扩充”



## 9 半导体存储器与CPU的连接

### ● 片选端译码

#### ■ “地址扩充”或“字扩充”

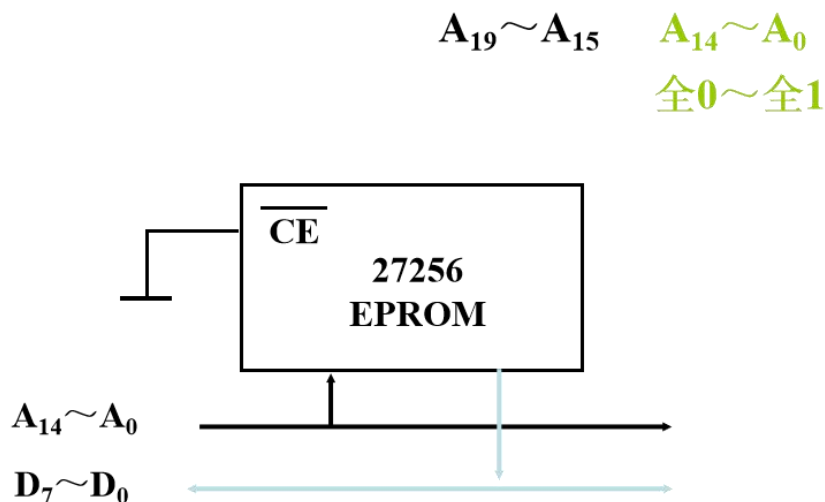




## 9 半导体存储器与CPU的连接

### ● 片选端译码

#### ■ “地址扩充”或“字扩充”



- 令芯片（组）的片选端常有效
- 不与系统的高位地址线发生联系
- 芯片（组）总处在被选中的状态
- 虽简单易行、但无法再进行地址扩充，会出现“地址重复”



## 9 半导体存储器与CPU的连接

### ● 片选端译码

#### ■ 地址重复

——一个存储单元具有多个存储地址的现象

——原因：有些高位地址线没有用、可任意

使用地址：出现地址重复时，常选取其中既好用、又不冲突的一个“可用地址”

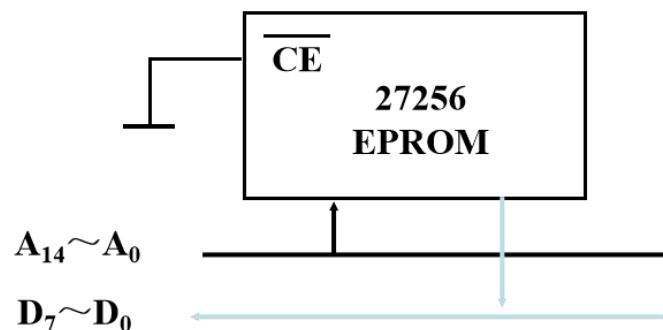
——例如：00000H~07FFFH

——选取的原则：高位地址全为0的地址

$A_{19} \sim A_{15}$

$A_{14} \sim A_0$

全0~全1





## 9 半导体存储器与CPU的连接

### ● 译码和译码器

- 译码：将某个特定的“编码输入”翻译为唯一“有效输出”的过程
- 译码电路可以使用门电路组合逻辑
- 译码电路更多的是采用集成译码器
  - 常用的2:4译码器：74LS139
  - 常用的3:8译码器：74LS138
  - 常用的4:16译码器：74LS154





## 9 半导体存储器与CPU的连接

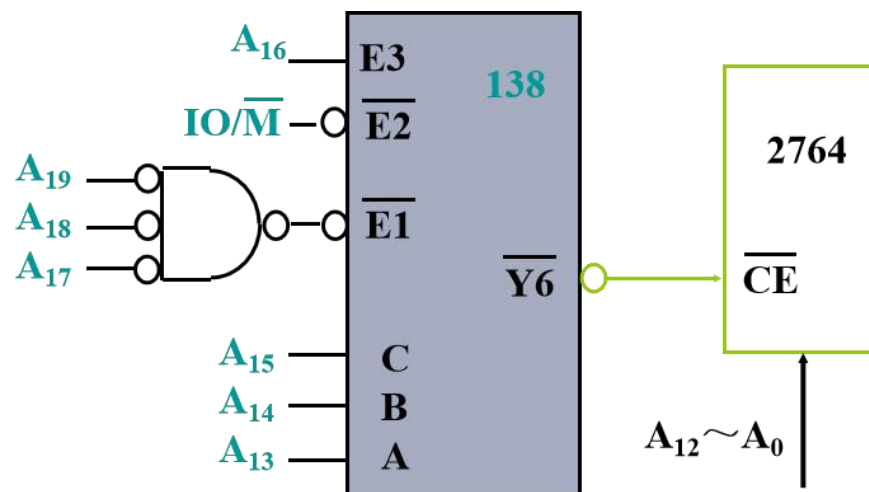
### ● 全译码

- 所有的系统地址线均参与对存储单元的译码寻址
- 包括低位地址线对芯片内各存储单元的译码寻址（片内译码），高位地址线对存储芯片的译码寻址（片选译码）
- 采用全译码，每个存储单元的地址都是唯一的，不存在地址重复
- 译码电路可能比较复杂、连线也较多



## 9 半导体存储器与CPU的连接

### ● 全译码



$A_{19}A_{18}A_{17}A_{16}A_{15}A_{14}A_{13}$	$A_{12} \sim A_0$	地址范围
0 0 0 1 1 1 0	全0	1C000H
⋮	⋮	⋮
	全1	1DFFFH



## 9 半导体存储器与CPU的连接

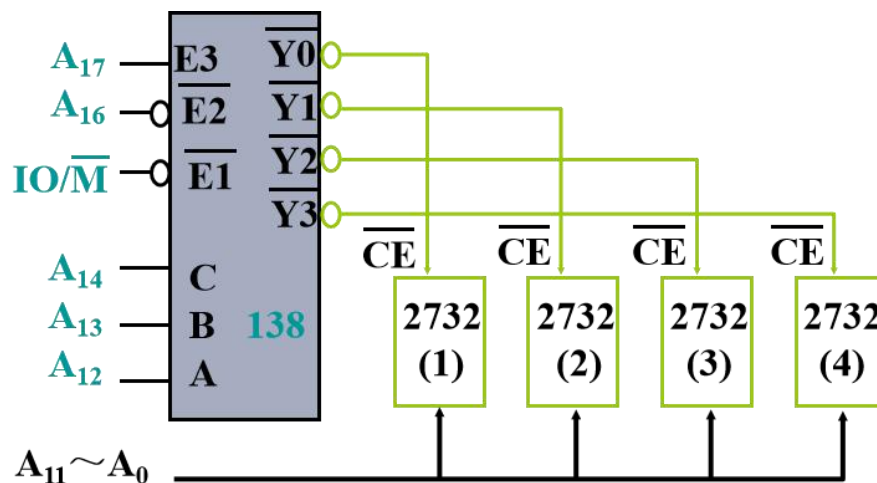
### ● 部分译码

- 只有部分（高位）地址线参与对存储芯片的译码
- 每个存储单元将对应多个地址（地址重复），需要选取一个可用地址
- 可简化译码电路的设计
- 但系统的部分地址空间将被浪费



## 9 半导体存储器与CPU的连接

### ● 部分译码



	$A_{19} \sim A_{15}$	$A_{14} \sim A_{12}$	$A_{11} \sim A_0$	一个可用地址
1	$\times \times 10 \times$	000	全0~全1	20000H~20FFFH
2	$\times \times 10 \times$	001	全0~全1	21000H~21FFFH
3	$\times \times 10 \times$	010	全0~全1	22000H~22FFFH
4	$\times \times 10 \times$	011	全0~全1	23000H~23FFFH



## 9 半导体存储器与CPU的连接

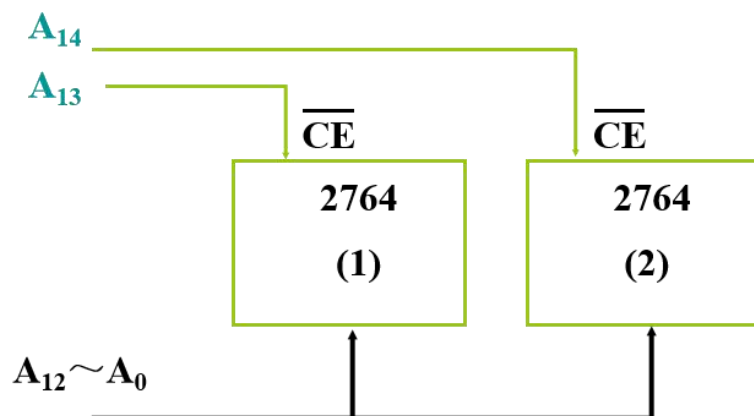
### ● 线选译码

- 只用少数几根高位地址线进行芯片的译码，且每根负责选中一个芯片（组）
- 虽构成简单，但地址空间严重浪费
- 必然会出现地址重复
- 一个存储地址会对应多个存储单元
- 多个存储单元共用的存储地址不应使用



## 9 半导体存储器与CPU的连接

### ● 线选译码



$A_{14} A_{13} = 00$  的情况不能出现  
00000H~01FFFH的地址不可使用

	$A_{19} \sim A_{15}$	$A_{14} A_{13}$	$A_{12} \sim A_0$	一个可用地址
1	××××××	1 0	全0~全1	04000H~05FFFH
2	××××××	0 1	全0~全1	02000H~03FFFH



## 9 半导体存储器与CPU的连接

### ● 片选端译码小结

- 存储芯片的片选控制端可以被看作是一根最高位地址线
- 地址空间的选择和高位地址的译码选择（与系统的高位地址线相关联）
- 对一些存储芯片通过片选无效可关闭内部的输出驱动机制，起到降低功耗的作用



## 9 半导体存储器与CPU的连接

### ● 存储芯片的读写控制线

#### ■ 芯片OE\*与系统的读命令线相连

——当芯片被选中、且读命令有效时，存储芯片将开放并驱动数据到总线

#### ■ 芯片WE\*与系统的写命令线相连

——当芯片被选中、且写命令有效时，允许总线数据写入存储芯片





## 9 半导体存储器与CPU的连接

### ● 存储芯片与CPU总线连接的两个重要问题

#### ■ CPU的总线负载能力

——CPU能否带动总线上包括存储器在内的连接器件

#### ■ 存储芯片与CPU总线时序的配合

——CPU能否与存储器的存取速度相配合



## 9 半导体存储器与CPU的连接

### ● CPU的总线负载能力

- CPU的总线驱动能力有限
- 单向传送的地址和控制总线，可采用三态锁存器和三态单向驱动器等来加以锁存和驱动
- 双向传送的数据总线，可以采用三态双向驱动器来加以驱动



## 9 半导体存储器与CPU的连接

### ● 存储芯片与CPU总线时序的配合

- 分析存储器的存取速度是否满足CPU总线时序的要求
- 如果不能满足：
  - 考虑更换芯片
  - 总线周期中插入等待状态 $T_w$