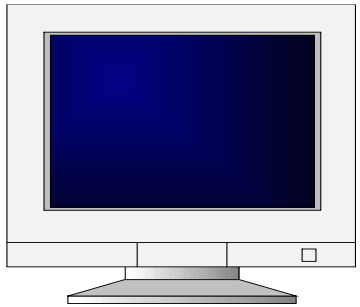
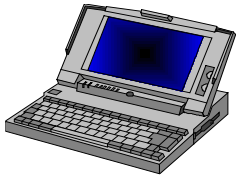
第5章

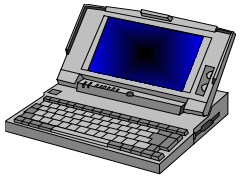
存储器系统





主要内容：

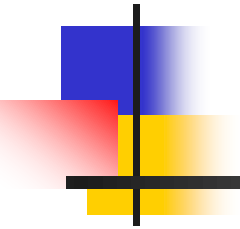
- 存储器系统
- 半导体存储器的分类及其特点
- 半导体存储芯片的外部特性及其与系统的连接
- 存储器接口设计（存储器扩展技术）
- 高速缓存



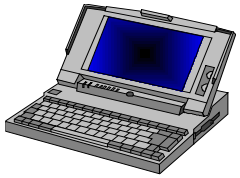
§5.1 概 述

主要内容：

- 存储器系统
- 半导体存储器的分类及特点
- 两类半导体存储器的主要区别



一、存储器系统

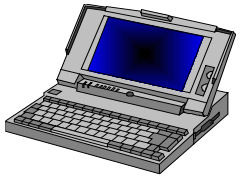


微机中的存储器系统

- 微型机中的存储器总体上包括：内存和外存
- 内存和外存在工作速度、容量、价格、制造材料等各方面都不相同。

	内 存	外 存
速度	快	慢
容量	小	大
单位容量价格	高	低
制造材料	半导体	磁性材料

存储器的分级体系结构



微机系统中存储器采用分级体系结构的根本目的是为了协调速度、容量、成本三者之间的矛盾。

简单的二级结构：

主　　存　　＋　　辅　　存

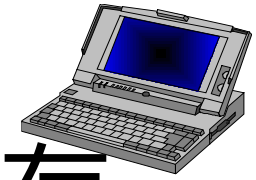
一般为半导体存储器，也称为短期存储器；

解决读写速度问题；

包括磁盘（中期存储器）、磁带、光盘（长期存储）等；

解决存储容量问题；

四级结构



寄存器 + Cache + 主存 + 辅存

CPU内部高速电子线路(如触发器)

一级：在CPU内部
二级：在CPU外部
一般为静态随机存储器SRAM。

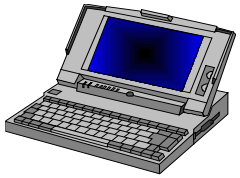
磁盘、磁带、光盘等

只有主存（内存）占用CPU的地址空间

一般用动态随机存储器DRAM存放临时数据，而用闪速存储器FLASH存放固化的程序和数据（即固件fireware）

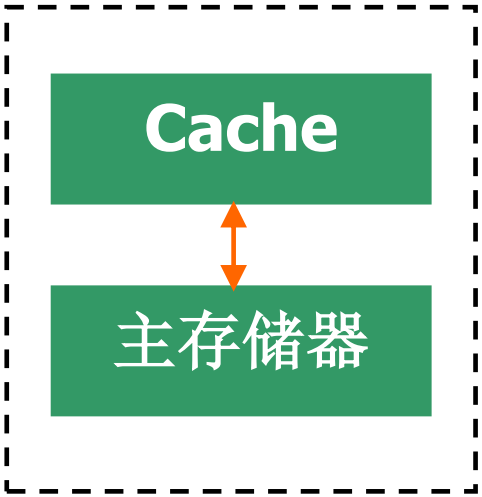
其中：Cache-主存结构解决高速度与低成本的矛盾；

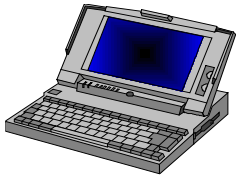
主存-辅存结构利用虚拟存储器解决大容量与低成本的矛盾；



Cache存储系统

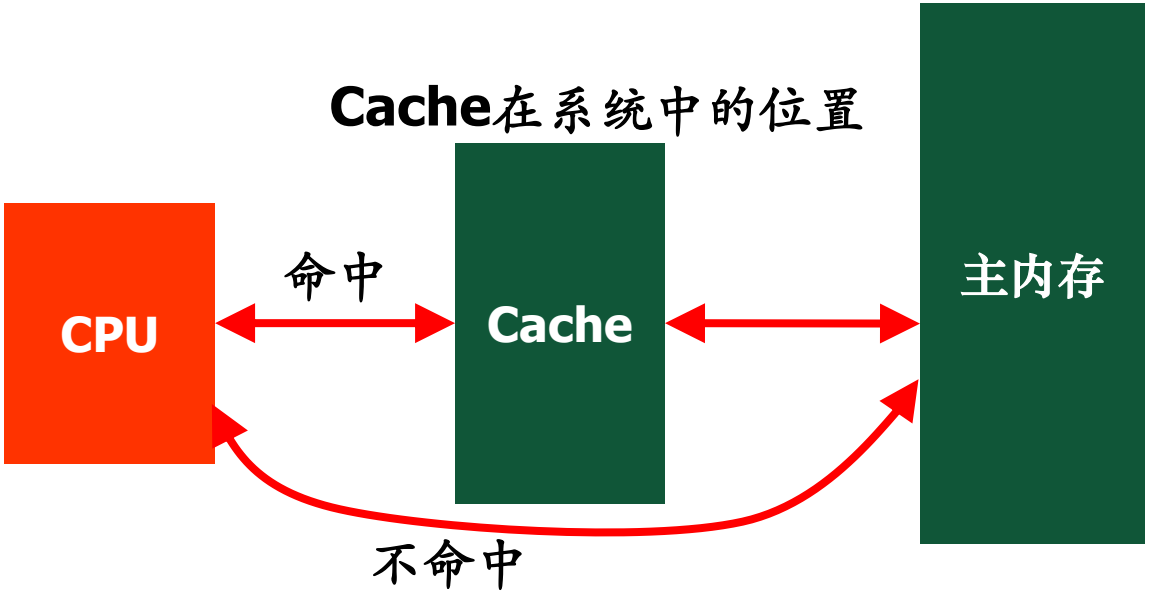
- Cache存储系统由高速缓冲存储器（Cache）和主内存构成，由硬件系统负责管理。
  - 对程序员是透明的
- 主要设计目标：
  - 提高CPU访问内存的存取速度。





Cache

- Cache
  - 高速缓存存储器

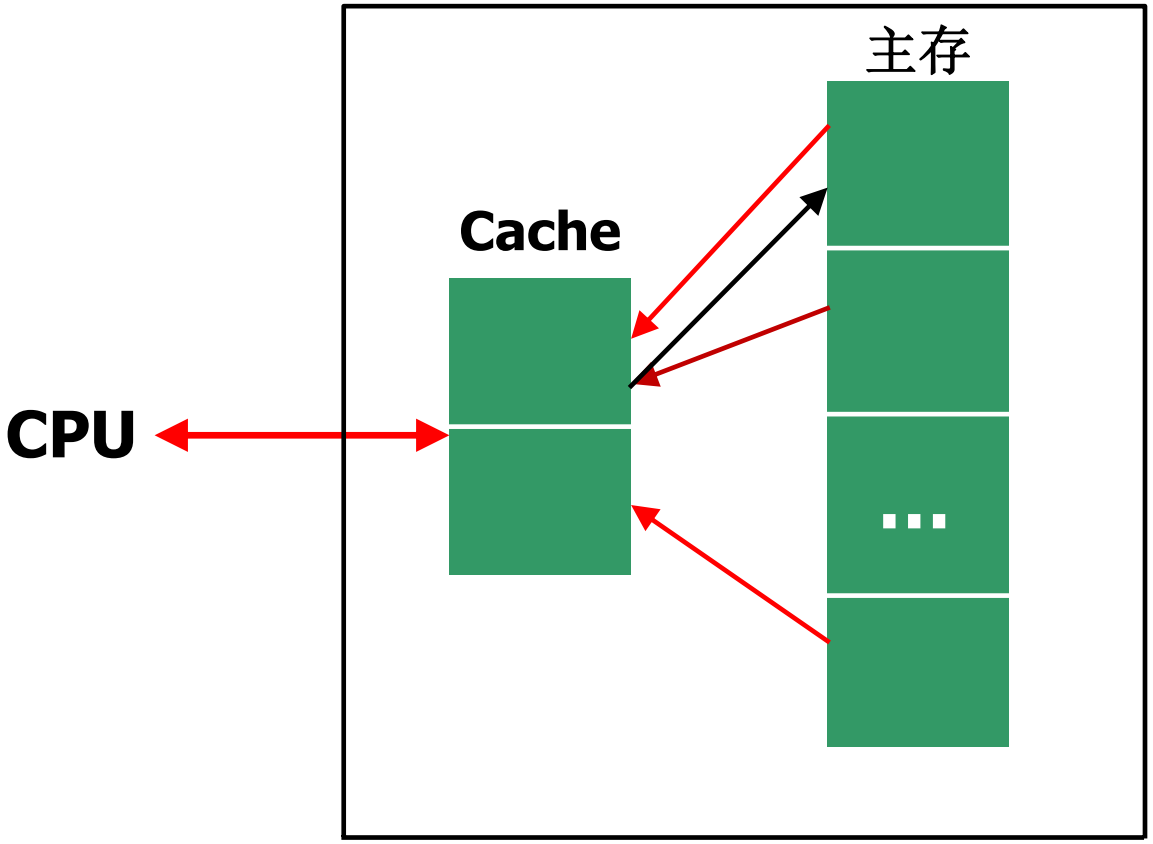


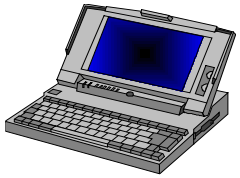
当命中率足够高时，整个Cache存储系统的：

1）访问速度接近与Cache的存取速度；

2）存储容量接近与主存的容量；

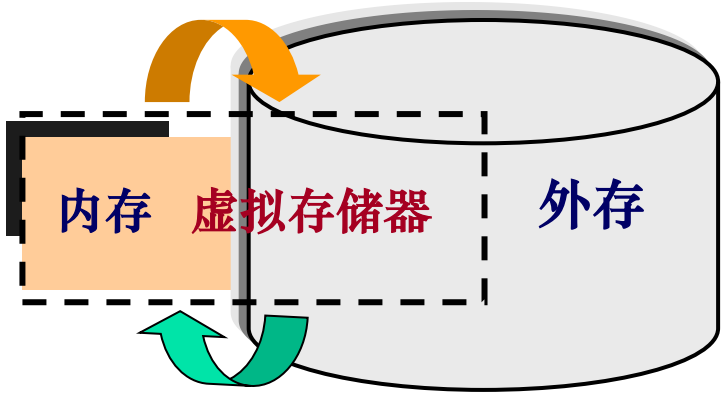
3）价格接近与主存的价格。

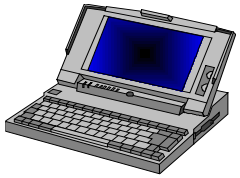




虚拟存储器系统

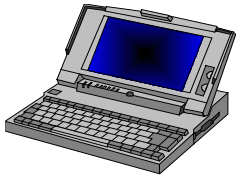
- 虚拟存储器系统由主内存和部分硬磁盘构成，主要由操作系统管理。
  - 对应用程序员是透明的
- 主要设计目标：
  - 扩大存储容量



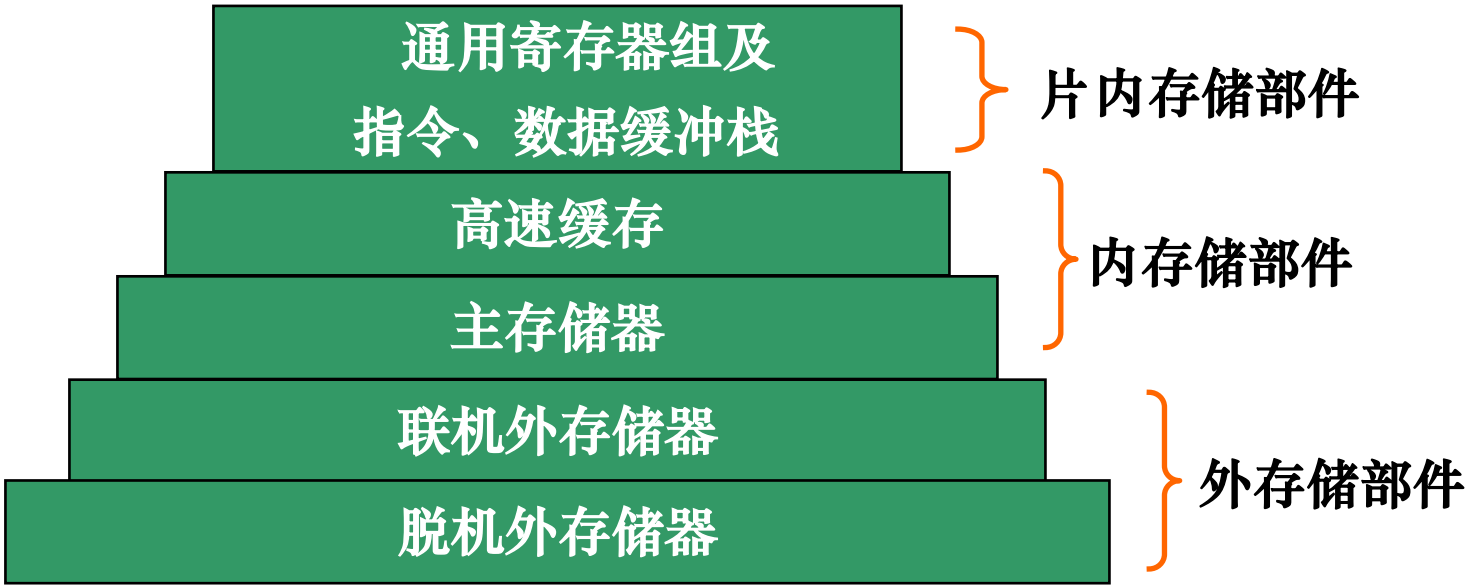


3. 主要性能指标

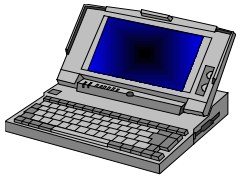
- 存储容量（S） （字节、千字节、兆字节等）
- 存取时间（T） （与系统命中率有关）
  - 命中率（H）
  - $T=H*T_1+(1-H)*T_2$
- 单位容量价格（C）



4. 微机中的存储器


通用寄存器组及
指令、数据缓冲栈
高速缓存
主存储器
联机外存储器
脱机外存储器
片内存储部件
内存储部件
外存储部件

存储器的分类及选用

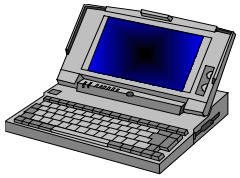


按存储介质分类

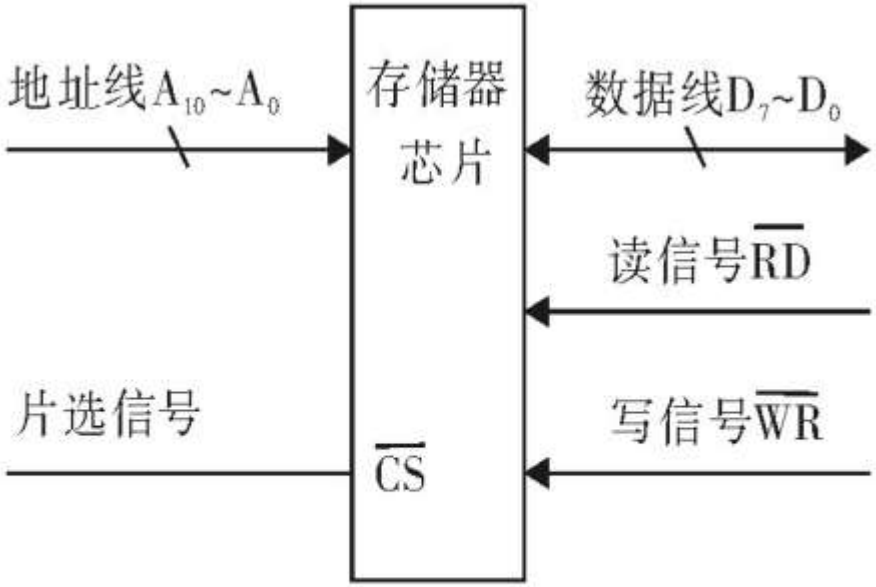
- 半导体存储器（按读写功能分类）
  - 读写存储器RAM（按器件原理分类）
    - 双极型：存取速度快，但集成度低，一般用于大型计算机或高速微机中；
    - MOS型（按存储原理分类）
      - 静态SRAM
        - Multi-SRAM
        - NV-SRAM
        - FIFO
        - Cache

        速度较快，集成度较低，一般用于对速度要求高、而容量不大的场合。
      - 动态DRAM：集成度高但存取速度较低，一般用于需要较大容量的场合。
  - 只读存储器ROM
    - 掩膜ROM
    - 一次性可编程PROM
    - 紫外线可擦除EPROM
    - 电可擦除EEPROM
    - 可编程只读存储器FLASH
- 磁介质存储器
- 光存储器

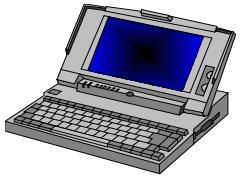
5. 存储原理与地址译码



存储器芯片逻辑图



存储器芯片逻辑图



① 存储体

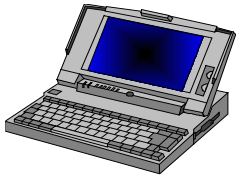
- 存储器芯片的主要部分，用来存储信息

② 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

③ 片选和读写控制逻辑

- 选中存储芯片，控制读写操作



① 存储体

- 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：

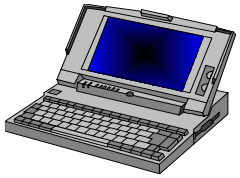
芯片的存储容量＝$2^M \times N$　(bit,位)

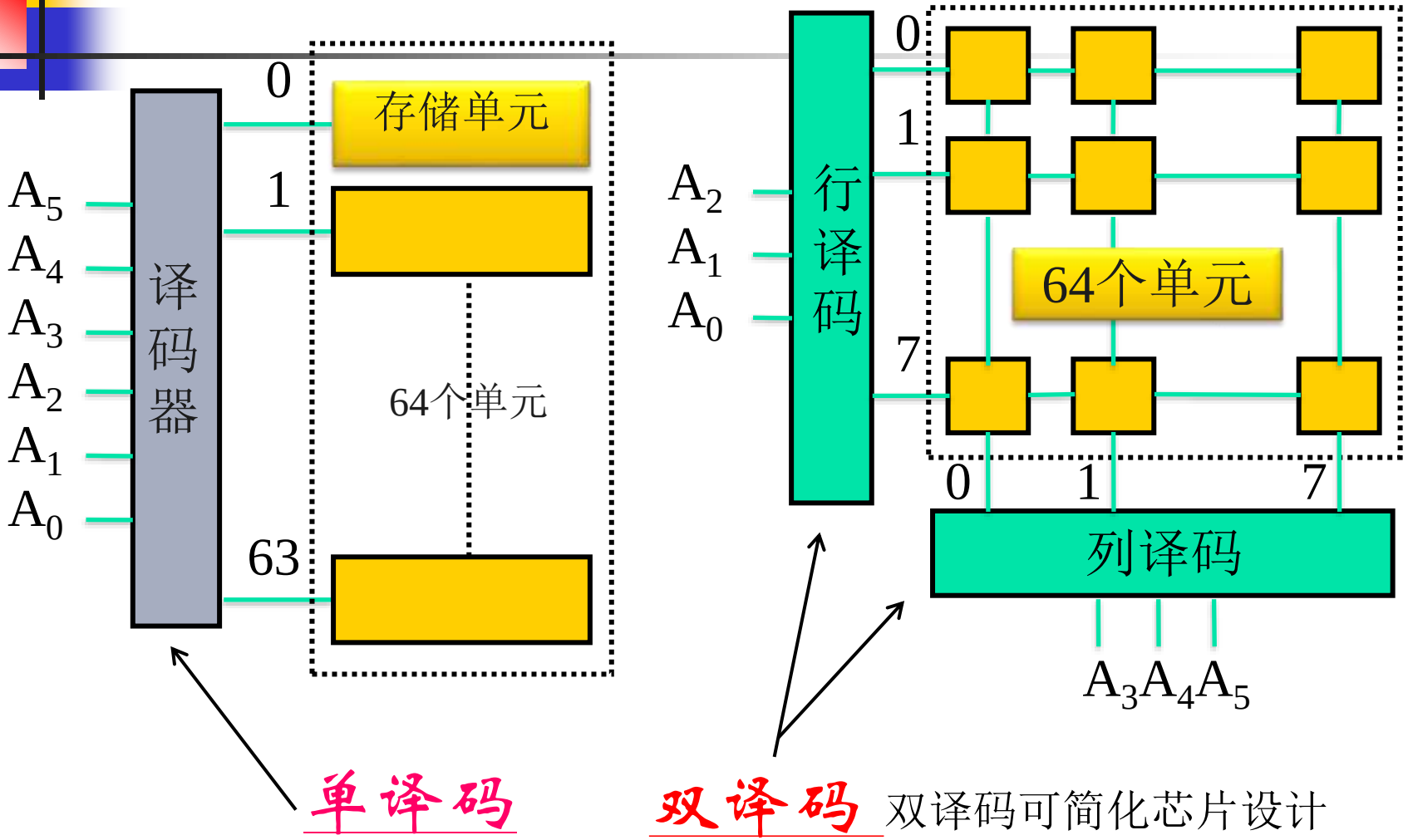
＝存储单元数×存储单元的位数

M：芯片的地址线根数

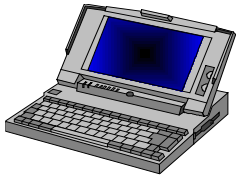
N：芯片的数据线根数

② 地址译码电路





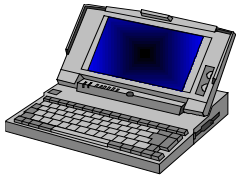
双译码可简化芯片设计



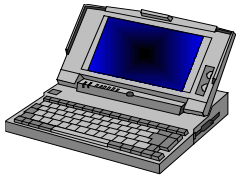
③ 片选和读写控制逻辑

- 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*（WR*和RD*）
  - 控制写操作。有效时，数据进入芯片中
  - 该控制端对应系统的写控制线

存储器芯片的工作方式



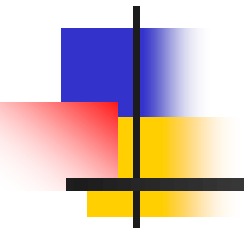
$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	操　　作
1	×	×	无操作
0	0	1	RAM→CPU操作
0	1	0	CPU→RAM操作
0	0	0	非法
0	1	1	无操作



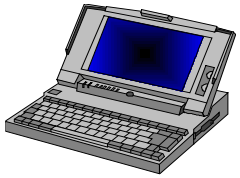
§5.2 随机存取存储器

掌握：

- SRAM与DRAM的主要特点
- 几种常用存储器芯片及其与系统的连接
- 存储器扩展技术



一、静态存储器SRAM

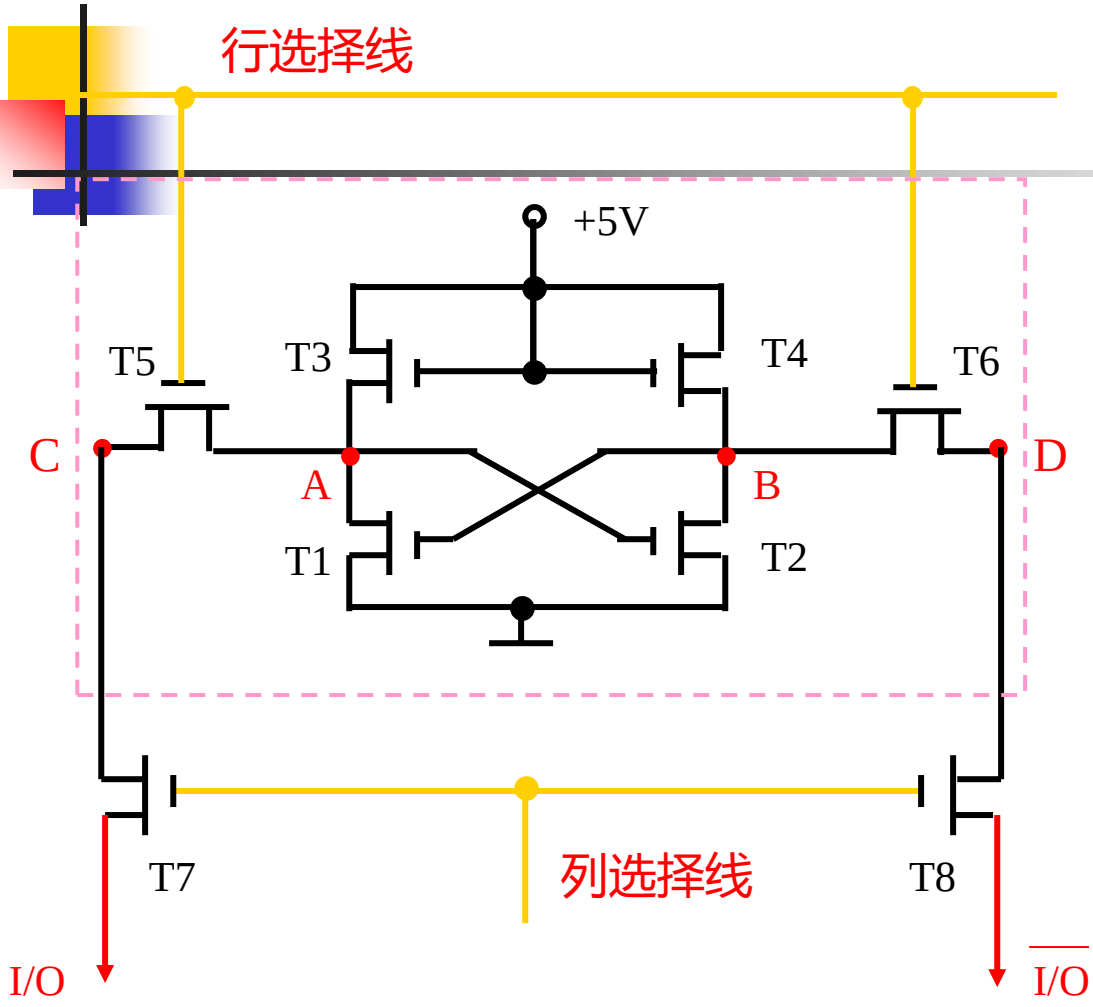


1. SRAM的特点

- 存储元由双稳电路构成，存储信息稳定。

p202

静态RAM的六管基本存储单元



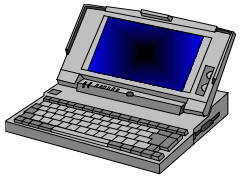
1. T1和T2组成一个双稳态触发器，用于保存数据。T3和T4为负载管。

2. 如A点为数据D，则B点为数据$\overline{D}$。

3. 行选择线有效（高电 平）时，A 、B处的数据信息通过门控管T5和T6送至C、D点。

4. 列选择线有效（高电 平）时，C 、D处的数据信息通过门控管T7和T8送至芯片的数据引脚I/O。

集成度低，但速度快，价格高，常用做Cache。



2. 典型SRAM芯片

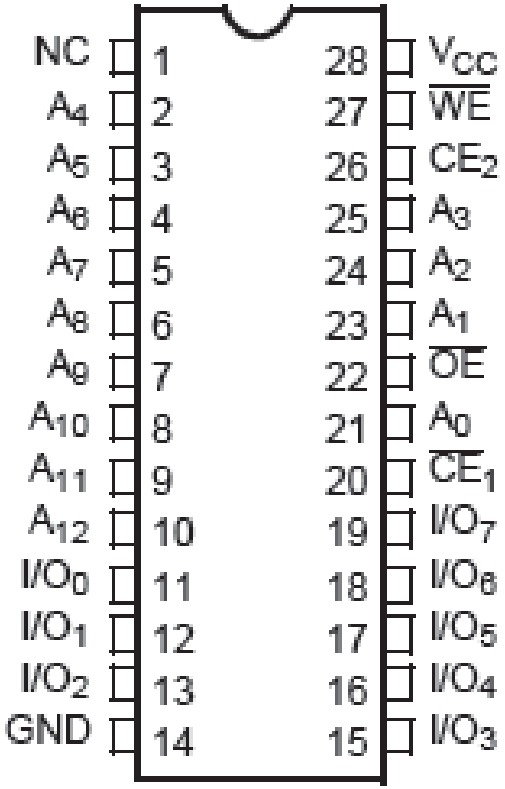
掌握：

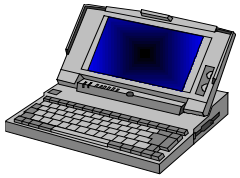
- 主要引脚功能
- 工作时序
- 与系统的连接使用

典型SRAM芯片

SRAM:6264

- 容量：
  - 8K ×8b
- 主要引线：
  - 地址线：A_0------A_{12}；
  - 数据线：D_0------D_7；
  - 输出允许信号：$\overline{OE}$；
  - 写允许信号：$\overline{WE}$；
  - 选片信号：$\overline{CE_1}$，CE_2。

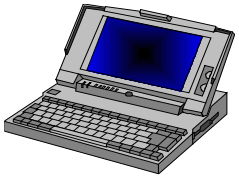




6264的工作过程

- 读操作
- 写操作

工作时序

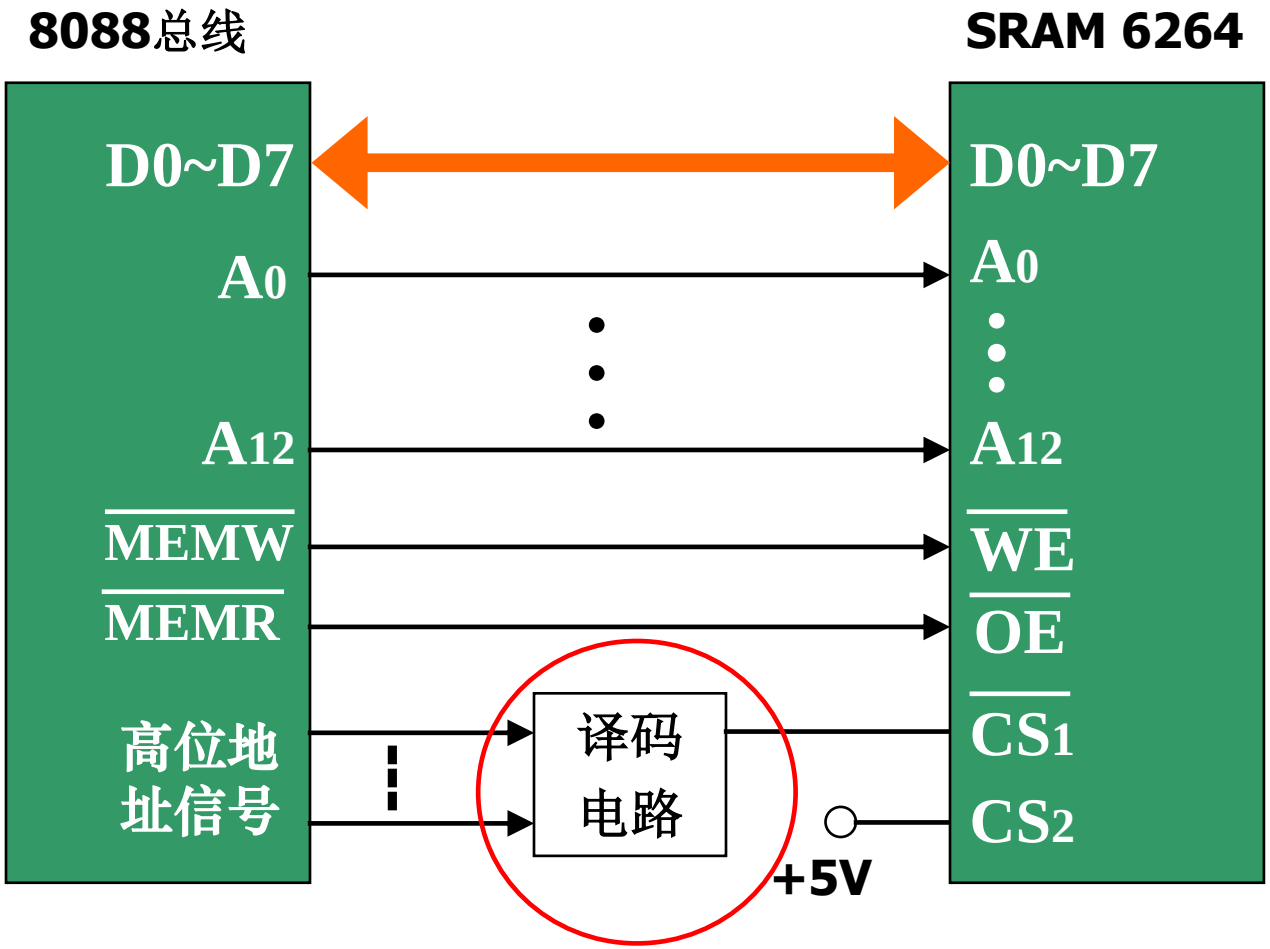


3. 6264芯片与系统的连接

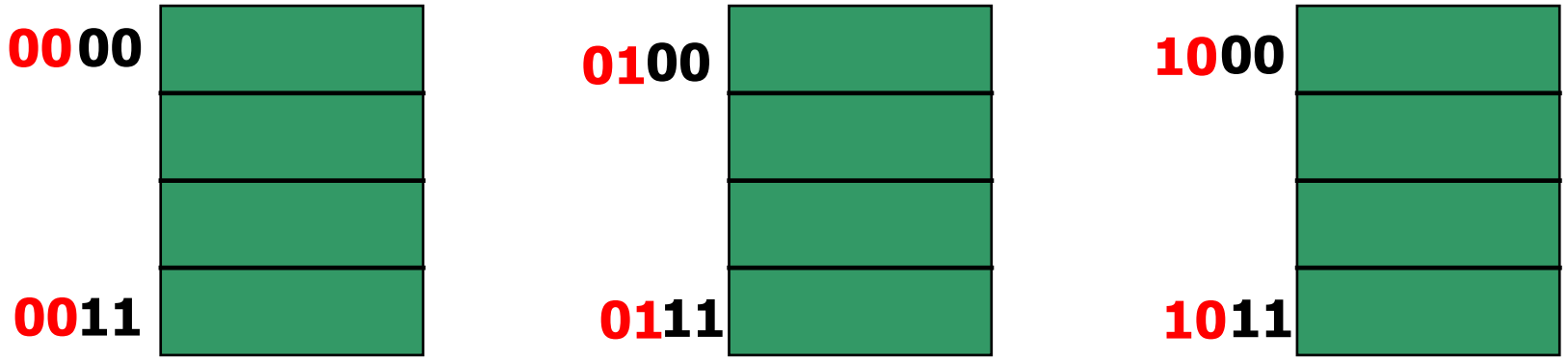
- 存储器芯片与系统的连接分为两部分：
  - 确定要访问的存储芯片
    - 系统中可能存在多片存储器芯片，要访问的单元只能存在于某一片芯片上。
  - 找到芯片后，寻找该芯片上要访问的单元。
    - 6264芯片上有8K个单元，每个单元在该芯片上有惟一的13位地址码。
    - 每片6264芯片上第一个单元在该芯片上的地址：0
    - 每片6264芯片上最后一个单元在该芯片上的地址：8191

用芯片的13位地址码A0-A12寻址片内的每个单元

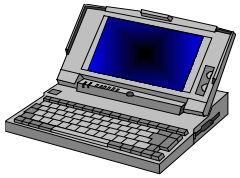
6264与系统的连接框架图



4. 存储器编址



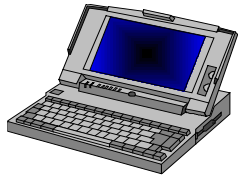
6264芯片的编址

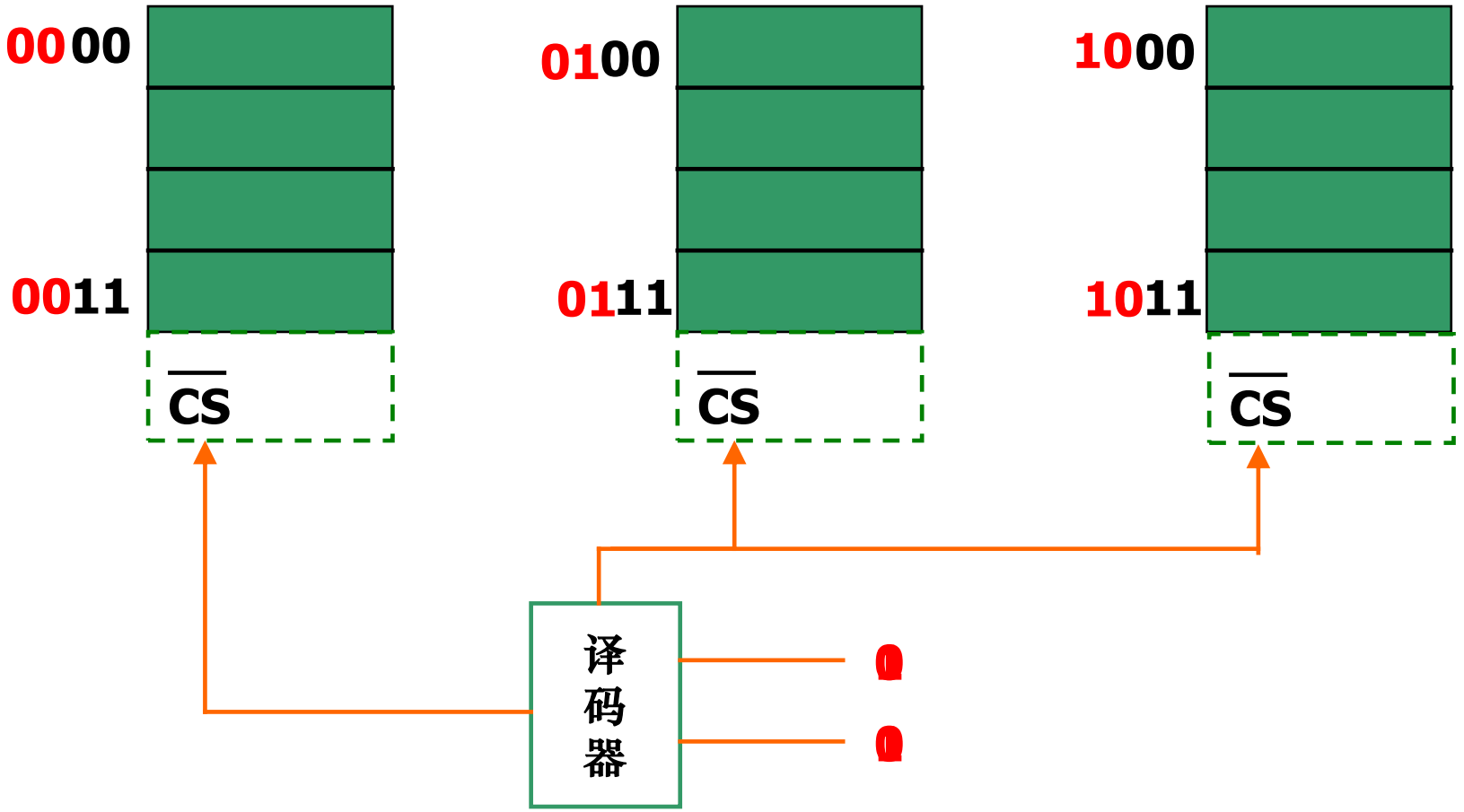


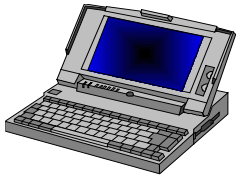
芯片上所有的单元具有相同的高位地址（片选地址）

从片首地址到片尾地址，构成芯片在内存空间中占有的地址范围

存储器编址

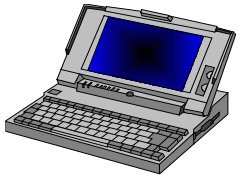






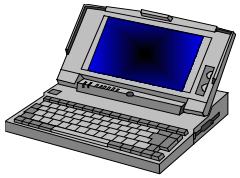
5. 译码电路

- 将输入的一组高位地址信号通过变换，产生一个有效的输出信号，用于选中某一个存储器芯片，从而确定了该存储器芯片在内存中的地址范围。
- 将输入的一组二进制编码变换为一个特定的输出信号。



译码方式

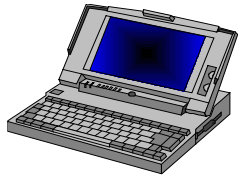
- 全地址译码
- 部分地址译码
- 线译码(极少使用)

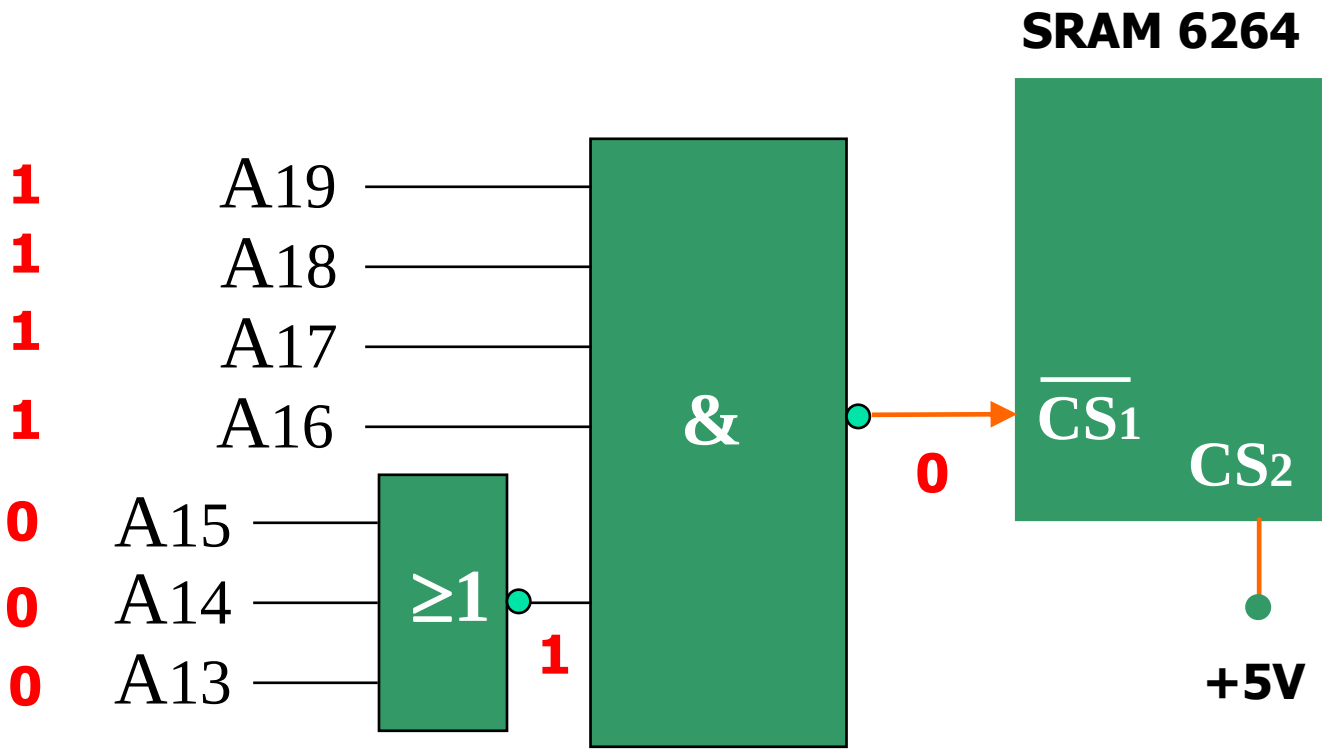


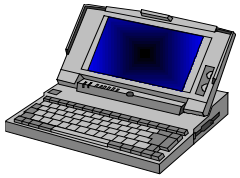
全地址译码

- 用全部的高位地址信号作为译码信号，使得存储器芯片的每一个单元都占据一个唯一的内存地址。

全地址译码例




SRAM 6264
1
1
1
1
A19
A18
A17
A16
0
0
0
A15
A14
A13
≥1
1
&
0
$\overline{CS_1}$
CS_2
+5V

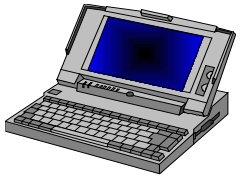


6264芯片全地址译码例

A19							A12													A0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	片首地址

A19							A12													A0
1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	片尾地址

该6264芯片的地址范围 = F0000H~F1FFFH



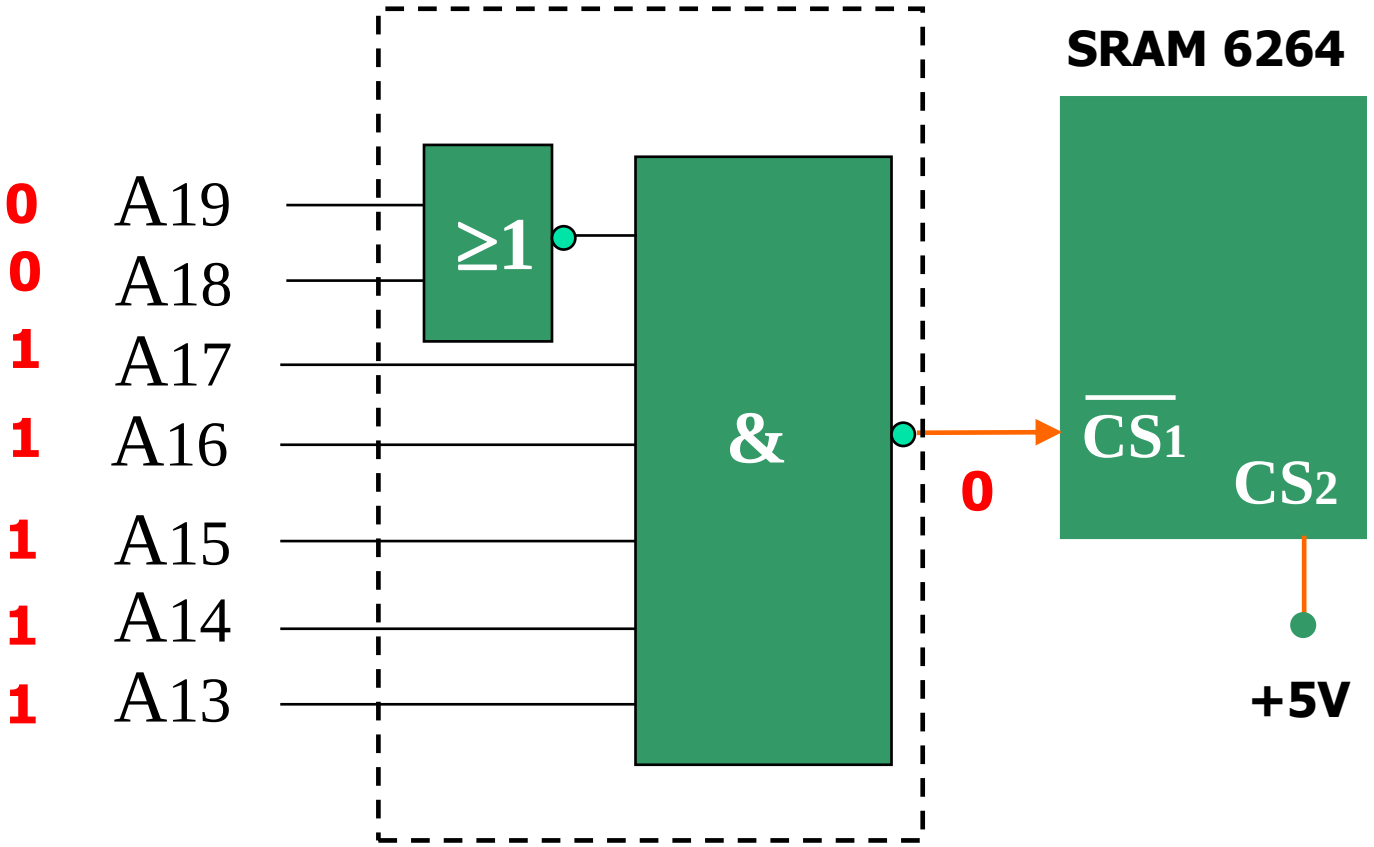
全地址译码例

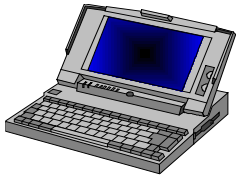
- 例：
  - 已知某SRAM 6264芯片在内存中的地址为：3E000H～3FFFFH。试画出将该芯片连接到系统的译码电路。
- 设计步骤：
  - 写出地址范围的二进制表示；
  - 确定各高位地址状态；
  - 设计译码器。

A19							A12												A0	
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	片首地址
A19							A12												A0	
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	片尾地址

全地址译码例

高位地址：0011111

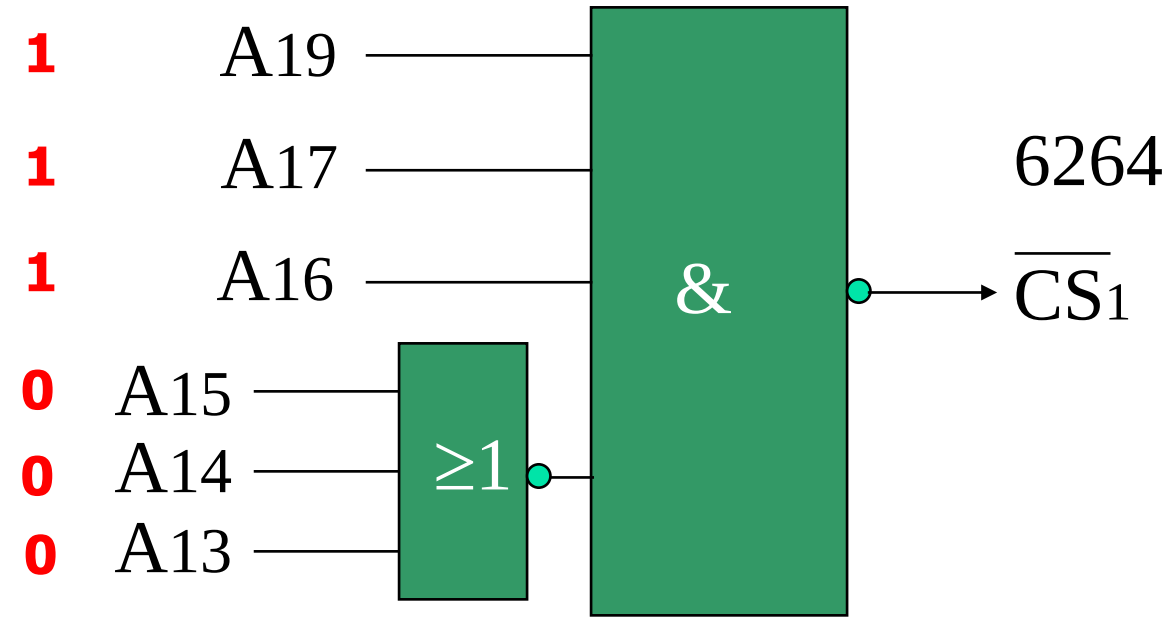




部分地址译码

- 用部分高位地址信号（而不是全部）作为译码信号，使得被选中存储器芯片占有几组不同的地址范围。

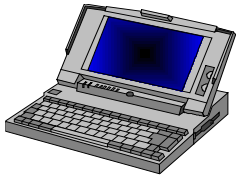
部分地址译码例



高位地址： 1×11000 ——— 1011000，1111000

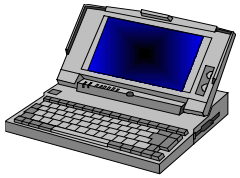
两组地址： B0000H —— B1FFFH

F0000H —— F1FFFH



译码电路小结

- 当仅需要将一片存储器芯片连接到系统中时，只需要简单的译码电路。
  - 根据设定地址确定电路设计，由基本逻辑门实现
- 当系统中的存储器由多片存储器芯片构成时，译码电路较复杂，需要借助专用译码器。

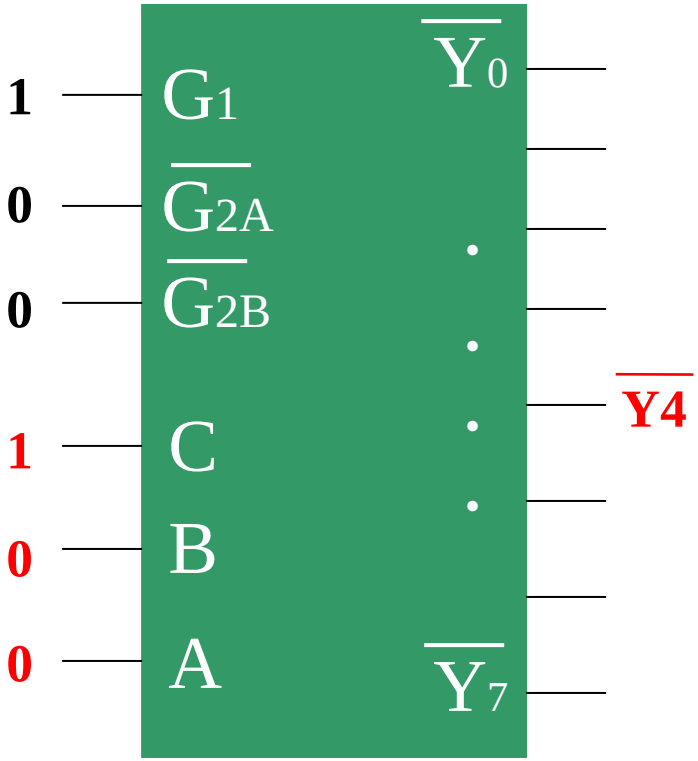


74LS138译码器

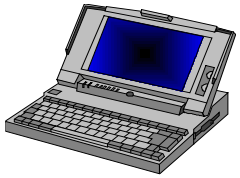
- 3输入8输出的专用译码器
- 可以同时控制8片芯片，并使在任一时刻，其所连接的8片芯片只有1片被选中。

74LS138译码器

- 主要引脚及功能

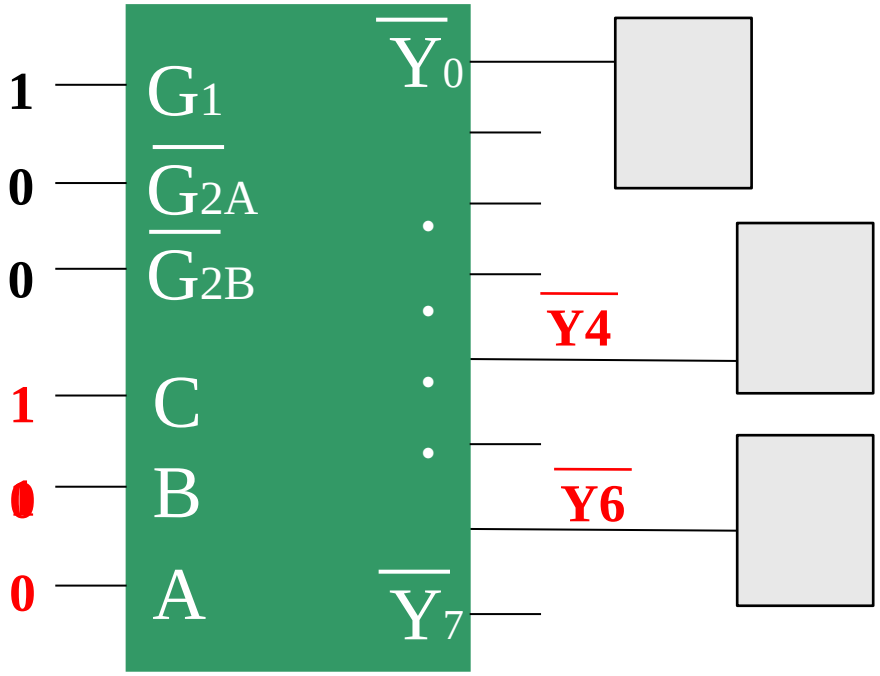


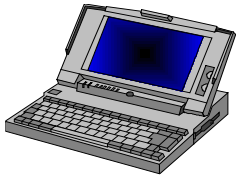
使能端			输入端			输出端							
G_1	$\#G_{2A}$	$\#G_{2B}$	C	B	A	$\#Y_0$	$\#Y_1$	$\#Y_2$	$\#Y_3$	$\#Y_4$	$\#Y_5$	$\#Y_6$	$\#Y_7$
×	1	1	×	×	×	1	1	1	1	1	1	1	1
0	×	×	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0



74LS138译码器

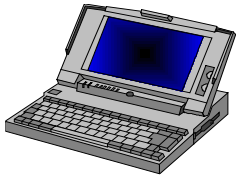
- 译码器主要功能：
  - 根据输入的不同编码组合，确保其控制的各电路（芯片）在**任一时刻只有一路（1个芯片）处于工作状态**。





SRAM存储器接口设计例

- 将SRAM 6264芯片与系统连接，使其地址范围为：38000H~39FFFH。
- 使用74LS138译码器构成译码电路。



SRAM存储器接口设计例

- 由题知地址范围：

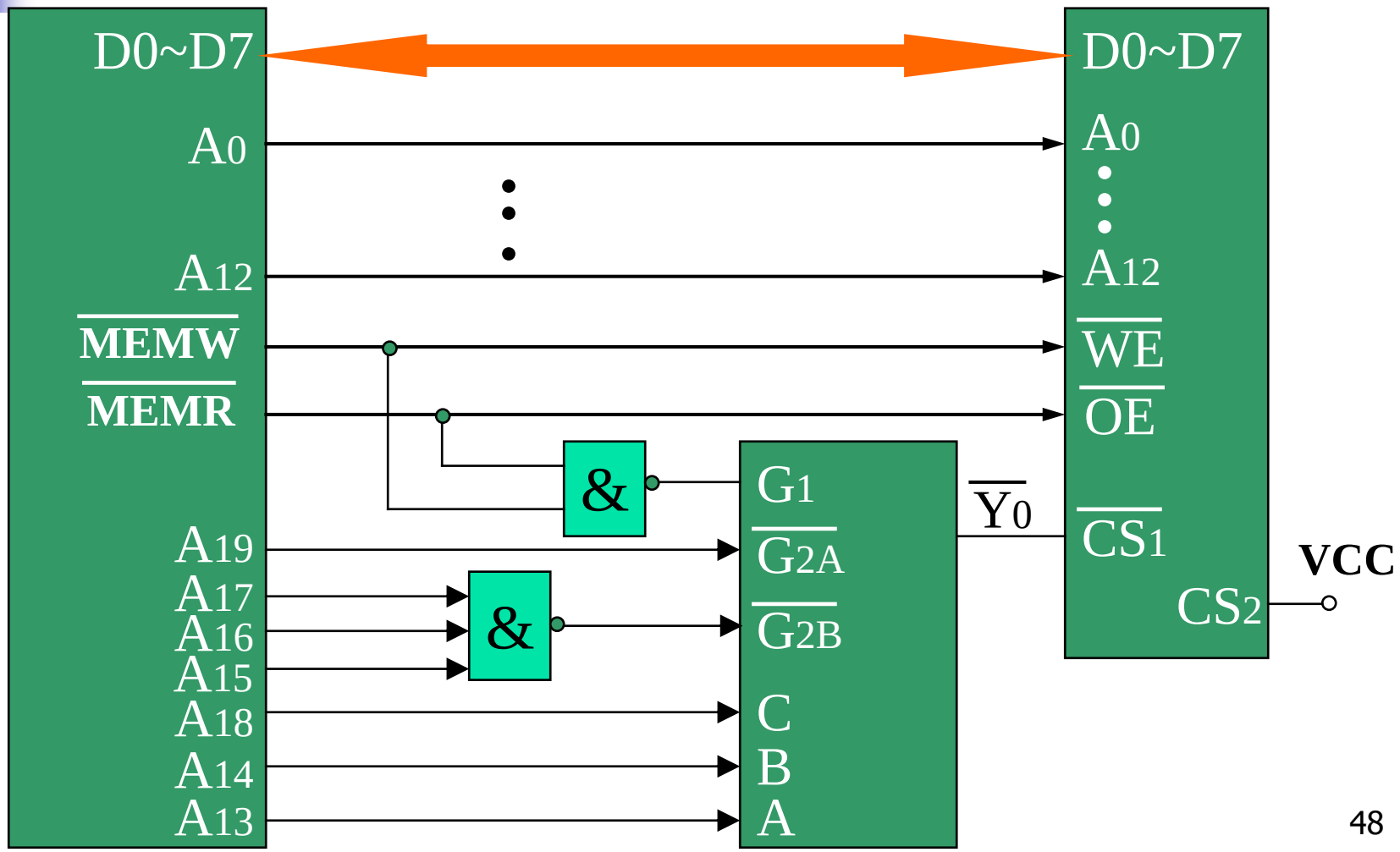
A19 A12 A0

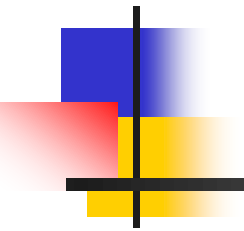
0 0 1 1 1 0 0 0 … … … 0

0 0 1 1 1 0 0 1 … … … 1

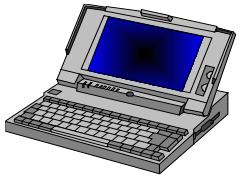
高位地址

SRAM存储器接口设计例





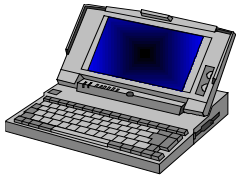
二、动态随机存储器DRAM

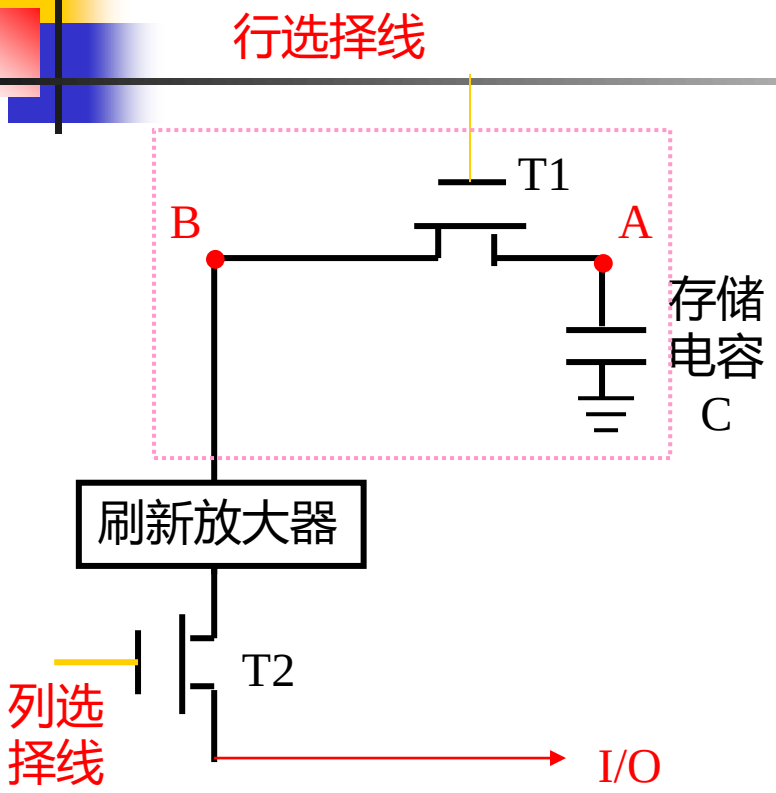


1. DRAM的特点

- 存储元主要由电容构成；
- 主要特点：
  - 需要定时刷新。

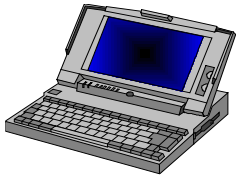
动态RAM的单管基本存储单元





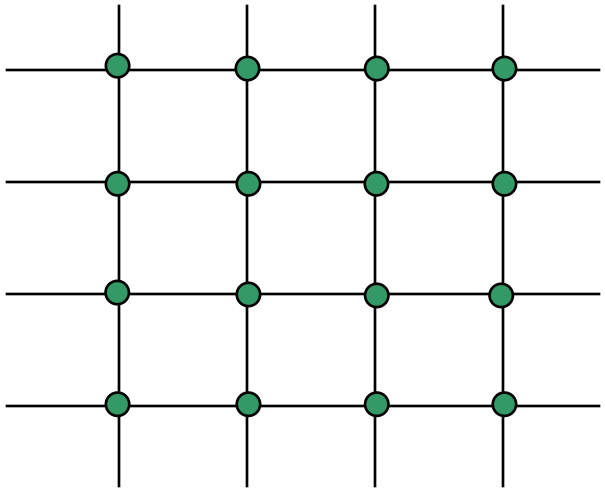
1. 电容上存有电荷时，表示存储数据A为逻辑1；
2. 行选择线有效时，数据通过T1送至B处；
3. 列选择线有效时，数据通过T2送至芯片的数据引脚I/O；
4. 为防止存储电容C放电导致数据丢失，必须定时进行刷新；
5. 动态刷新时行选择线有效，而列选择线无效。（刷新是逐行进行的。）

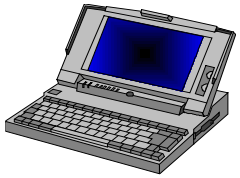
集成度高，但速度较慢，价格低，一般用作主存。



2. 典型DRAM芯片2164A

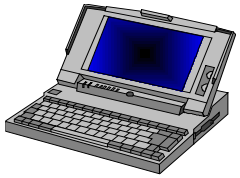
- 2164A：64K×1bit
- 采用行地址和列地址来确定一个单元；
- 行列地址分时传送，共用一组地址信号线；
- 地址信号线的数量仅为同等容量SRAM芯片的一半。





主要引线

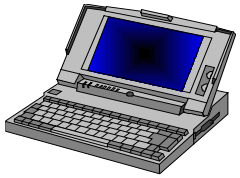
- $\overline{\text{RAS}}$：行地址选通信号。用于锁存行地址；
- $\overline{\text{CAS}}$：列地址选通信号。
  - 地址总线上先送上行地址，后送上列地址，它们分别在#RAS和#CAS有效期间被锁存在锁存器中。
- $\overline{\text{WE}}$：写允许信号
  - WE=0 → 数据写入
  - WE=1 → 数据读出
- DIN：数据输入
- DOUT：数据输出



工作原理

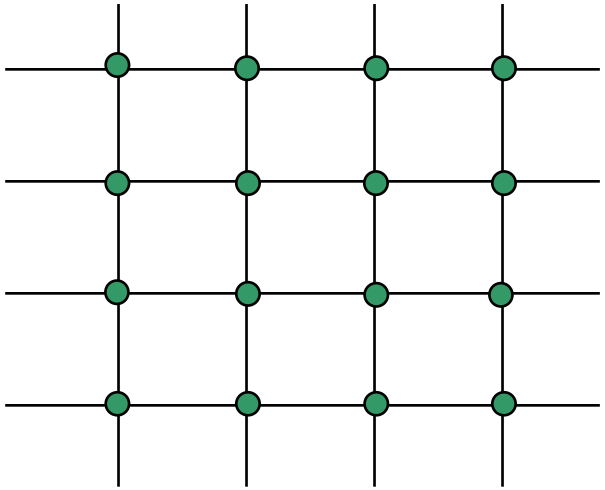
- 数据读出
- 数据写入
- 刷新

工作时序

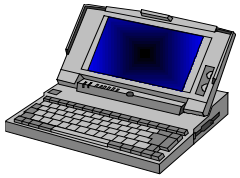


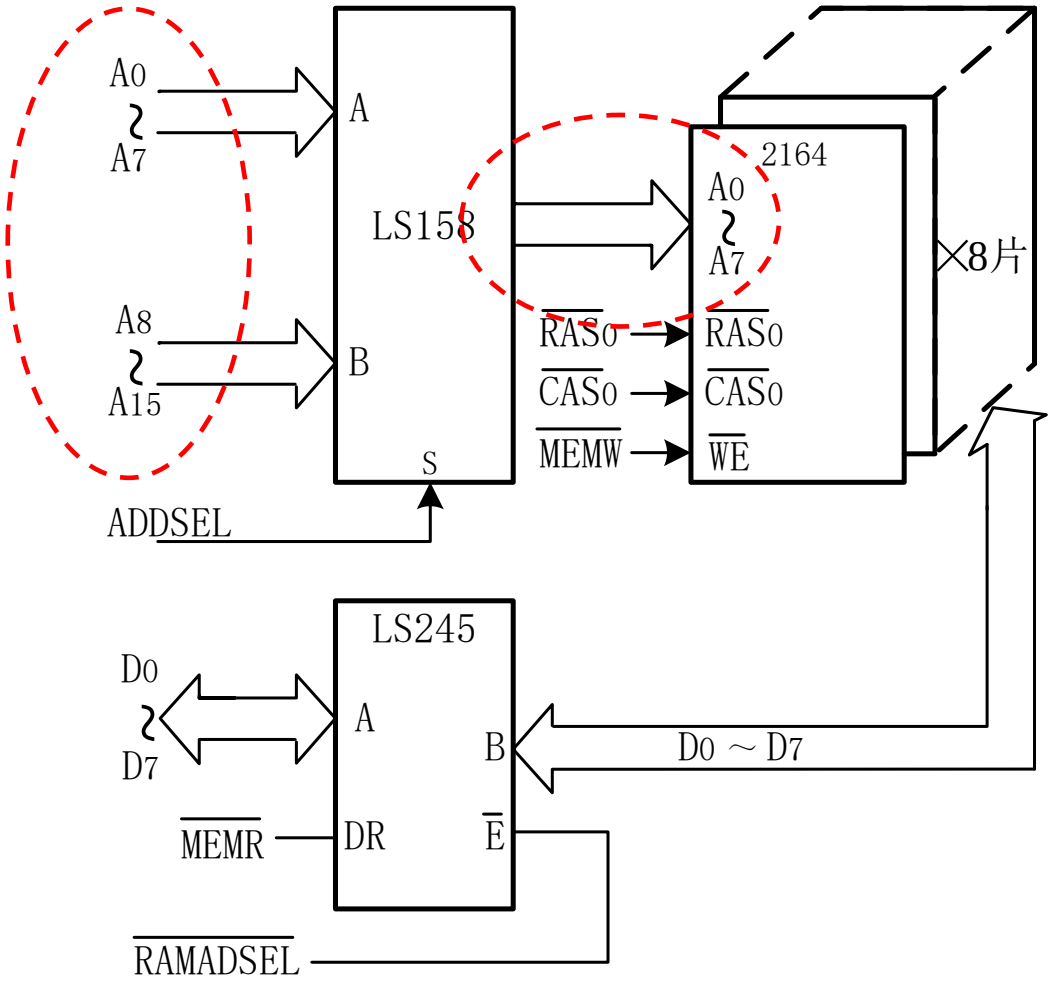
刷新

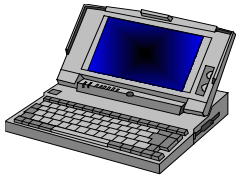
- 将存放于每位中的信息读出再照原样写入原单元的过程---------刷新



3. 2164A在系统中的连接





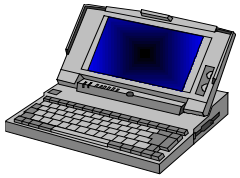


2164A在系统中的连接

- DRAM 2164A与系统连接的几点说明：
  - 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。

三、存储器扩展技术

（内存储器设计）



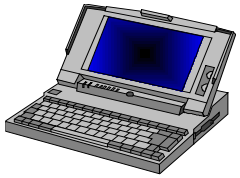
1. 存储器扩展

- 用多片存储芯片构成一个需要的内存空间；
- 各存储器芯片在整个内存中占据不同的地址范围；
- 任一时刻仅有一片（或一组）被选中。
- **存储器芯片的存储容量等于：**

单元数×每单元的位数

字节数　　字长

存储器容量扩展



存储体、地址译码、
数据缓冲和读写控制

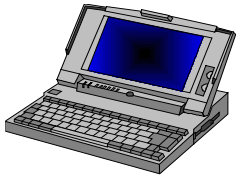
存储芯片 → 存储模块 → 存储体

进行位扩展 以实现按字节编址的结构

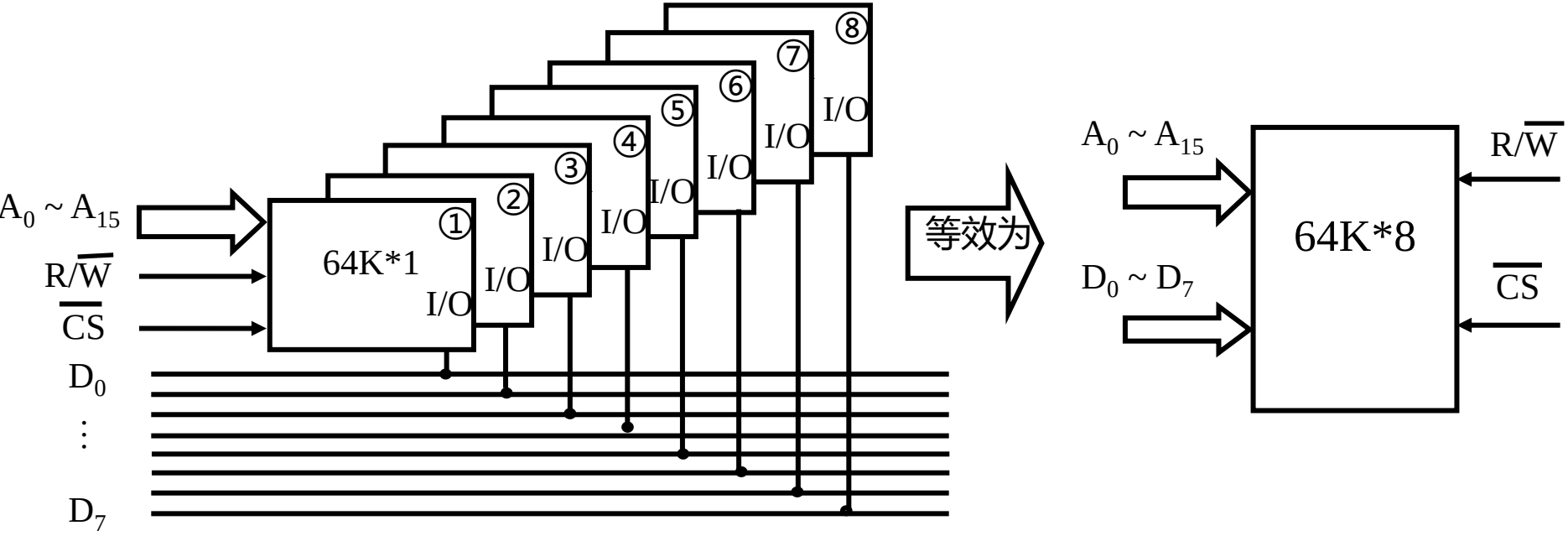
进行字扩展 以满足总容量的要求

- 位扩展：因每个字的位数不够而扩展数据输出线的数目；
- 字扩展：因总的字数不够而扩展地址输入线的数目，所以也称为地址扩展；

存储芯片的位扩展

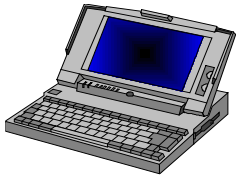


用64K×1bit的芯片扩展实现64KB存储器

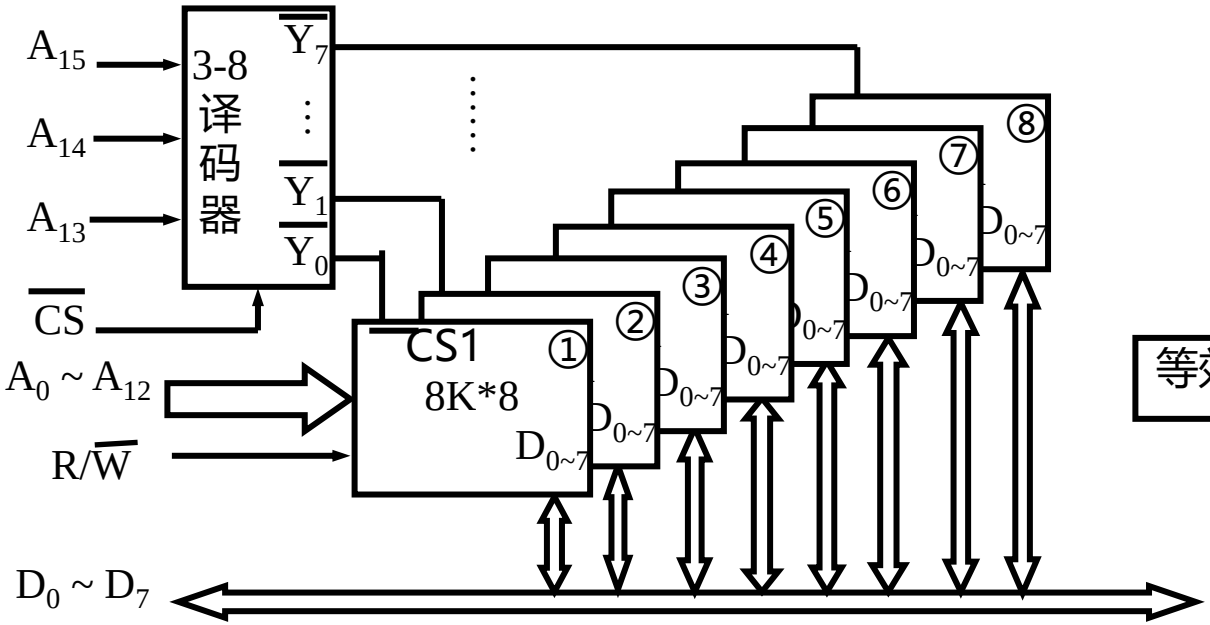


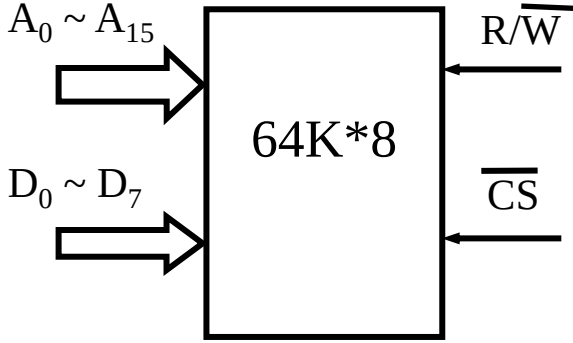
进行位扩展时，模块中所有芯片的地址线和控制线互连形成整个模块的地址线和控制线，而各芯片的数据线并列（位线扩展）形成整个模块的数据线（8bit宽度）。

存储芯片的字扩展



用8K×8bit的芯片扩展实现64KB存储器

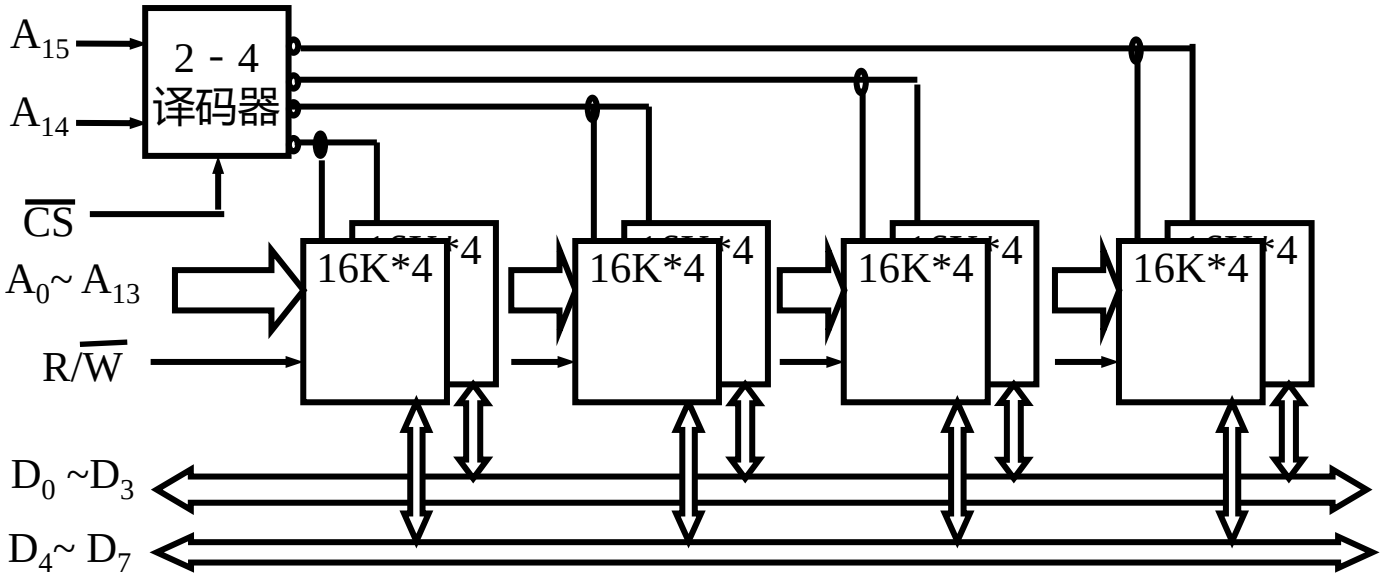


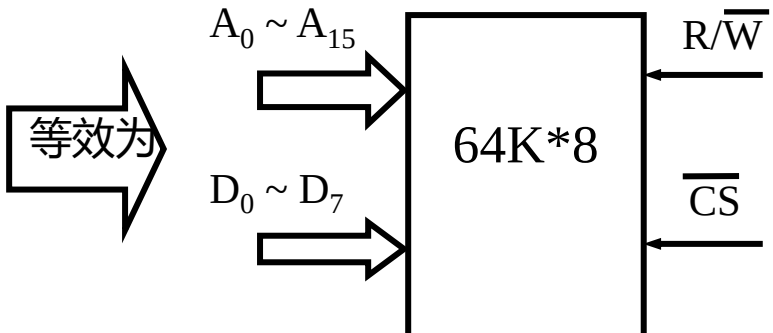


进行字扩展时，模块中所有芯片的地址线、控制线和数据线互连形成整个模块的低位地址线、控制线和数据线， CPU的高位地址线（扩展的字线）被用来译码以形成对各个芯片的选择线 —— 片选线 。

存储芯片的字、位同时扩展

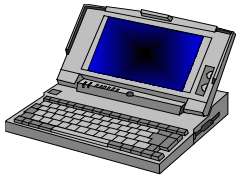
用16K×4bit的芯片扩展实现64KB存储器





首先对芯片分组进行位扩展，以实现按字节编址；

其次设计个芯片组的片选进行字扩展，以满足容量要求；



例

设某系统地址总线宽度为20bit，数据总线宽度为8bit。现采用8K×8芯片实现32KB扩展存储器，要求其地址从0C0000H 开始，试画出该扩展存储器与系统三总线的连接方式。

扩展存储器共需要8K×8的存储芯片数量

N = (32K×8)/ (8K×8)=4×1片

数据线：不要位扩展，芯片数据线互连后与系统数据线连接

读写控制线：所有芯片的读/写线分别互连后与系统相连

低位地址线：8K容量的存储芯片需要13根地址线进行字选，所有芯片地址线互连后与系统的低13位地址线（A0~A12）连接；

高位地址线：剩余的7根系统地址线（A13 ~ A19）可用于产生所需的4根片选线；

用全译码法实现扩展存储器的片选设计

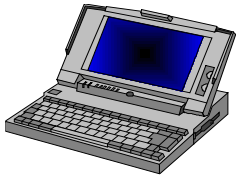
全译码方式下，系统的每一条地址线都应该参与译码。设该扩展存储器占用0C0000H开始的一段连续地址空间，则可用下表表示系统地址信号与各芯片所占地址空间的关系：

芯 片	A_{19}~A_{15}	A_{14}	A_{13}	A_{12}~A_0	地址空间（顺序方式）
①	11000	0	0	111111111111 ～ 000000000000	0C1FFFH～0C0000H
②	11000	0	1		0C3FFFH～0C2000H
③	11000	1	0		0C5FFFH～0C4000H
④	11000	1	1		0C7FFFH～0C6000H

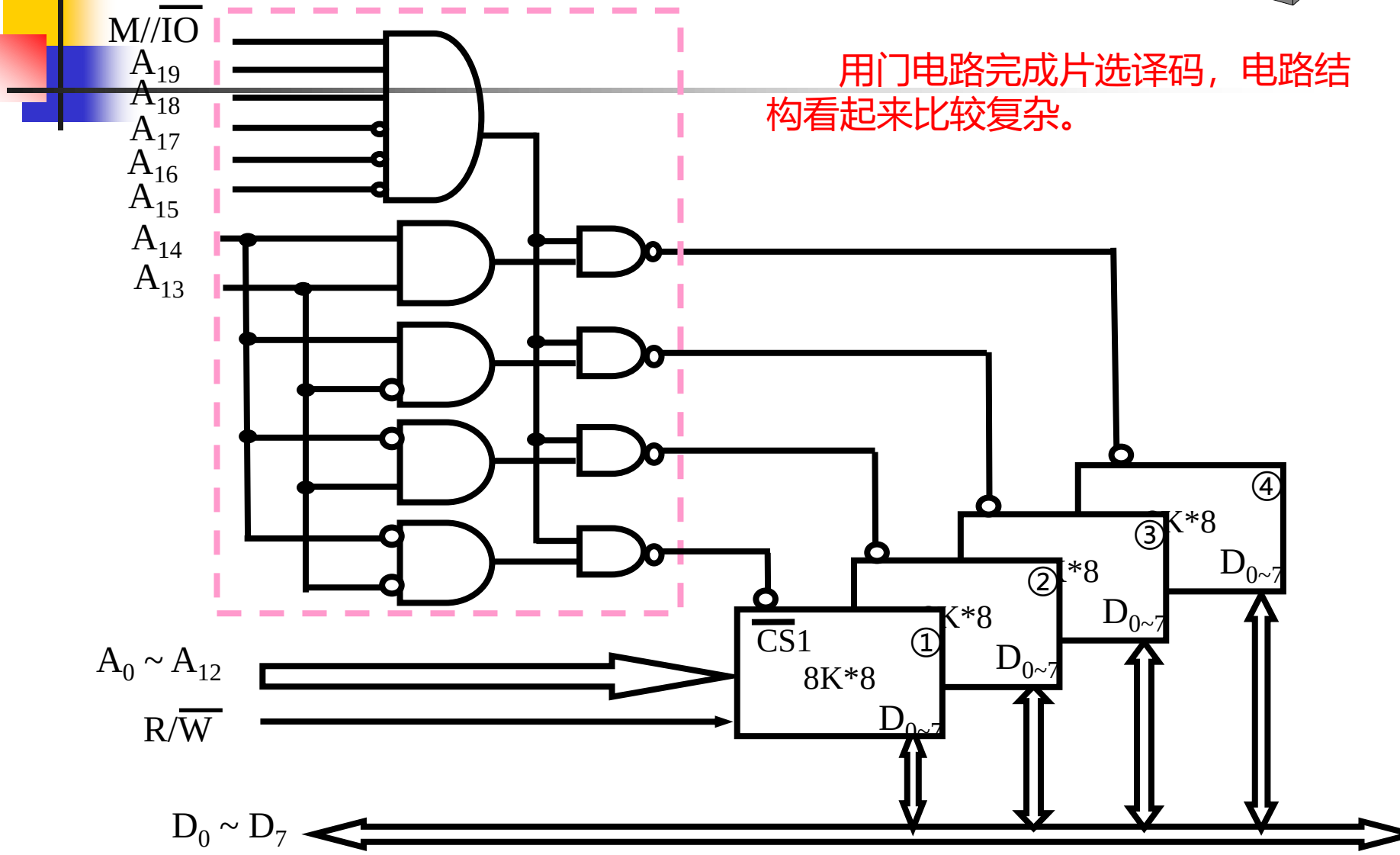
从该表中可以看出：

- 低位地址线A_{12}~A_0应直接接在存储芯片上，寻址片内8K单元；
- 次高位地址线A_{14}、A_{13}译码后产生片选信号区分4个存储芯片；
- 最高位地址线A_{19}~A_{15}及控制信号M/(/IO)可用作片选信号有效的使能控制；

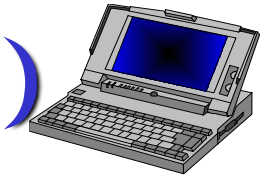
符合要求的全译码电路（一）

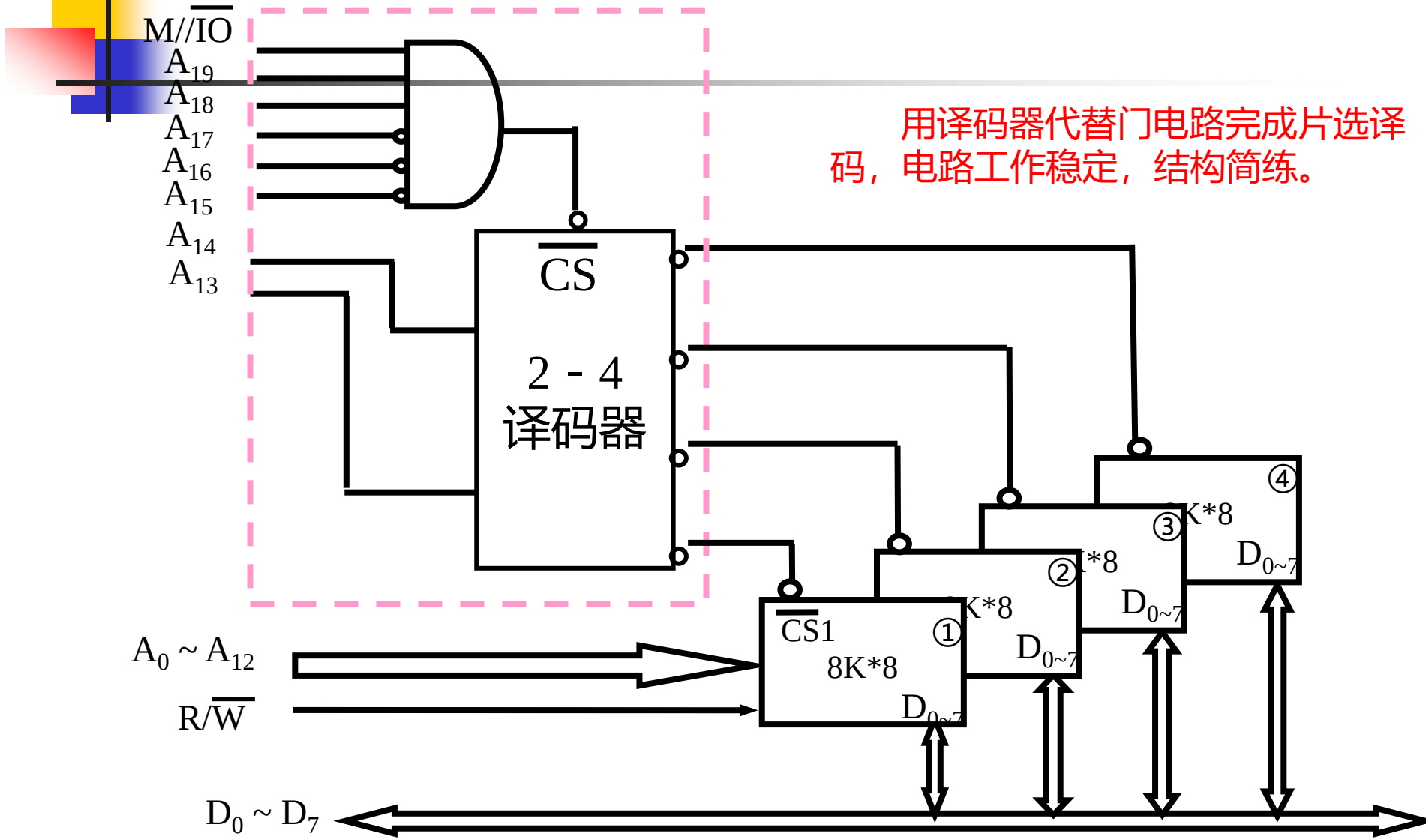


用门电路完成片选译码，电路结构看起来比较复杂。



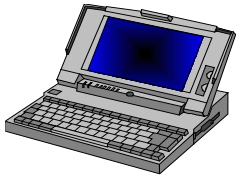
符合要求的全译码电路（二）





用译码器代替门电路完成片选译码，电路工作稳定，结构简练。

用部分译码法实现扩展存储器的片选设计



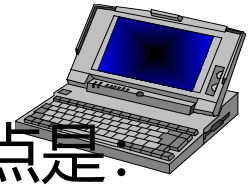
与全译码方式的唯一区别是：系统最高段地址信号（A_{19}～A_{15}）不参与片选译码，即这几位地址信号可以为任何值。

芯片	A_{19}～A_{15}	A_{14}	A_{13}	A_{12} ～ A_0	地址空间（顺序方式）
①	00000 …… 11000 …… 11111	00		1111111111111 ～ 0000000000000	001FFFH～000000H …… 0C1FFFH～0C0000H …… 0F9FFFH～0F8000H
②	11000	01			0C3FFFH～0C2000H
③	11000	10			0C5FFFH～0C4000H
④	11000	11			0C7FFFH～0C6000H

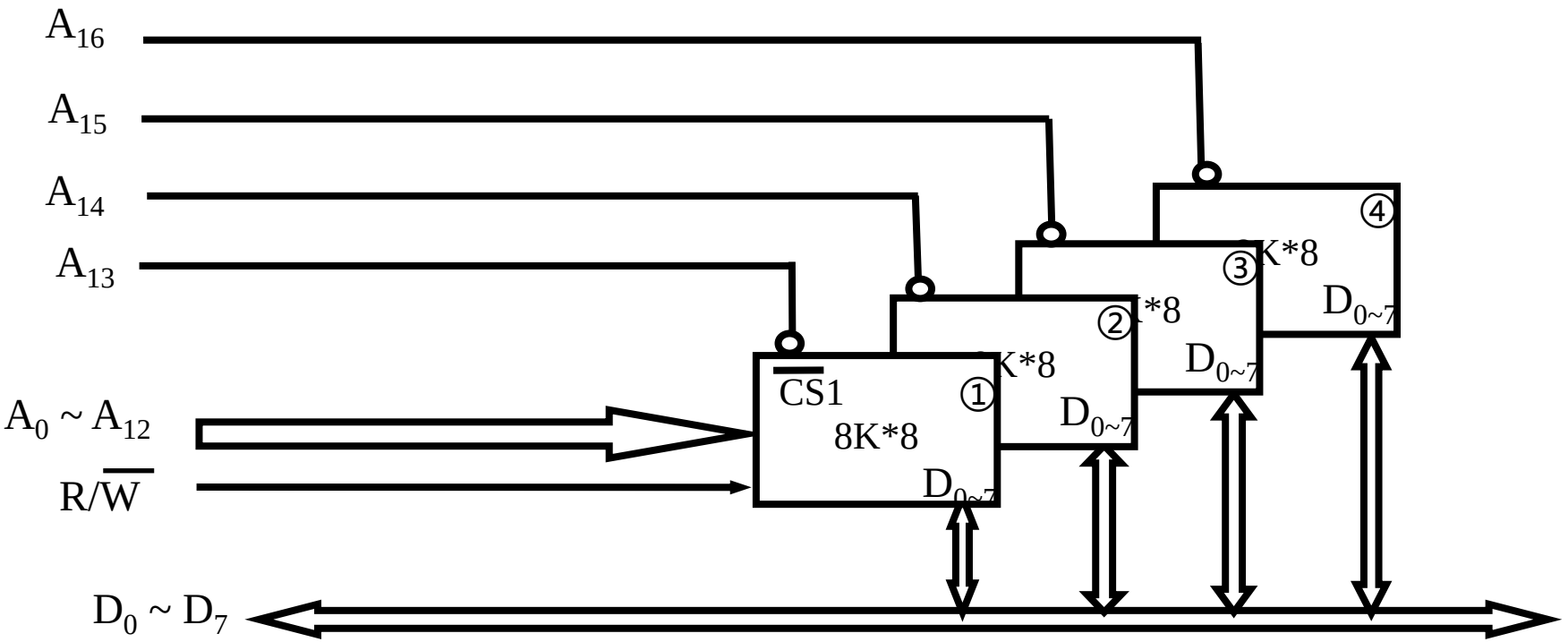
共占用2^5组地址

造成地址空间的重叠

用线译码法实现扩展存储器的片选设计

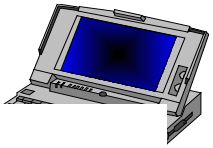


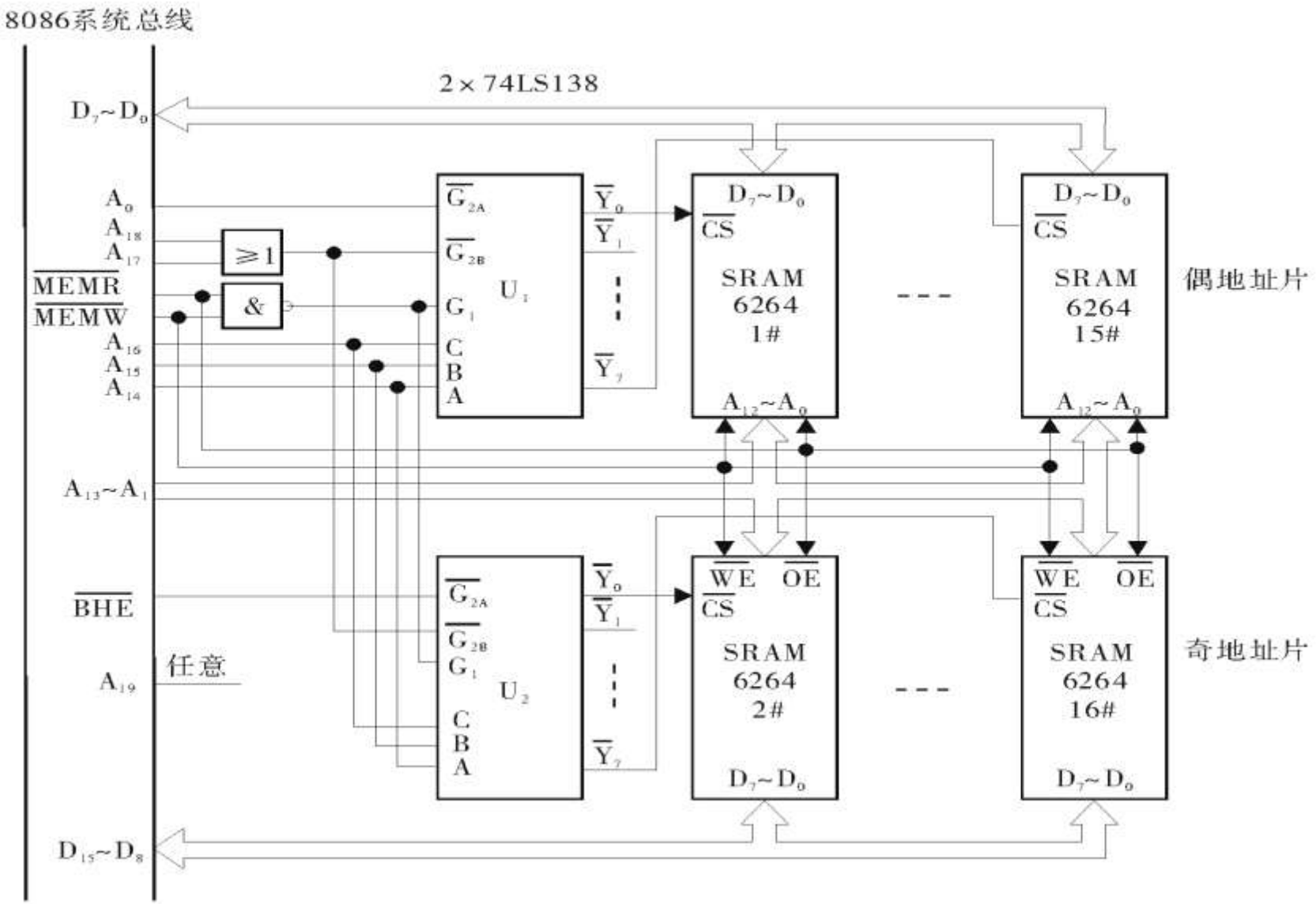
- 4个片选信号必须使用4根地址线，电路结构简单，缺点是：
  - 系统必须保证A_{16} ~ A_{13}不能同时为有效低电平；
  - 同部分译码法一样，因为最高段地址信号（A_{19} ~ A_{15}）不参与译码，也存在地址重叠问题；

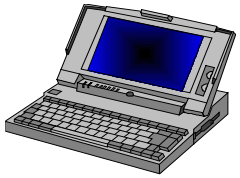


思考：试写出各芯片占用的地址空间。

例：8086系统与存储器的连接

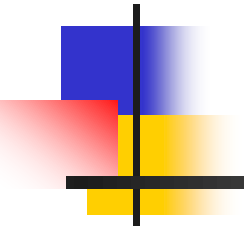




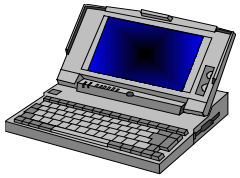


§5.3 只读存储器（ROM）

- EPROM（紫外线擦除）
- EEPROM（电擦除）

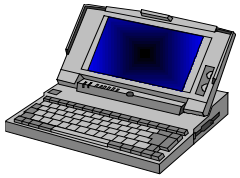


一、EPROM



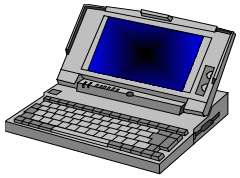
1. 特点

- 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。



2. EPROM 2764

- 8K×8bit芯片
- 地址信号：A0 —— A12
- 数据信号：D0 —— D7
- 输出信号：$\overline{OE}$
- 片选信号：$\overline{CE}$
- 编程脉冲输入：PGM
- 其引脚与SRAM 6264完全兼容.



2764的工作方式

- 数据读出

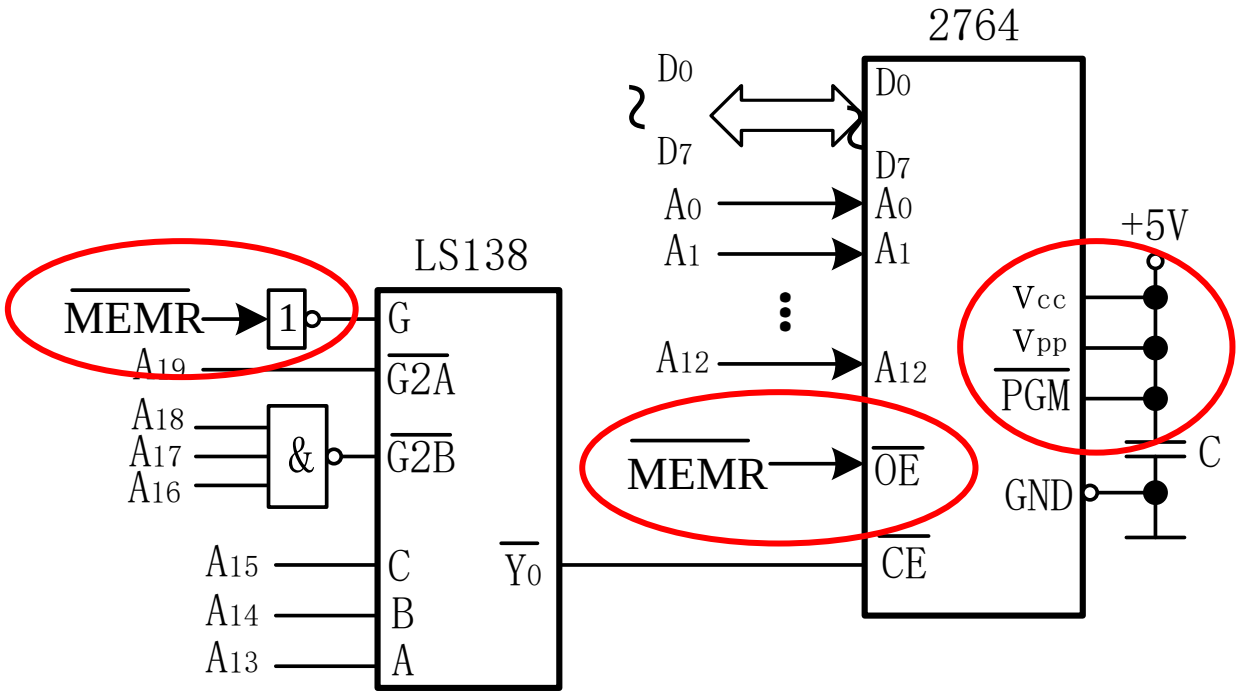
EPROM芯片因其较高的稳定性，常用作程序存储器，存放相应的控制程序。

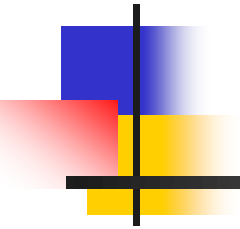
RAM芯片则因其便利性，常用作数据存储器，存放操作的数据。

- 紫外光擦除

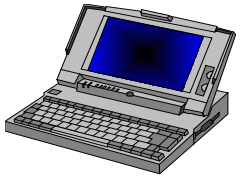
EPROM 2764的应用

- 2764与系统的连接



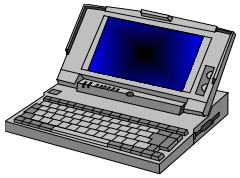


二、EEPROM



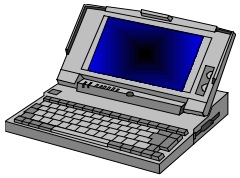
1. 特点

- 可在线编程写入；
- 掉电后内容不丢失；
- 电可擦除。



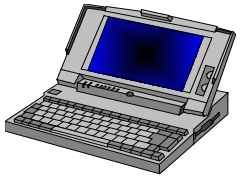
2. 工作方式

- 数据读出
- 编程写入
  - 字节写入：每一次BUSY正脉冲写入1B
  - 自动页写入：每一次BUSY正脉冲写入 32B
- 擦除
  - 字节擦除：一次擦除一个字节
  - 片擦除：一次擦除整片



3. 典型EEPROM芯片98C64A

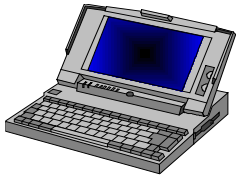
- 8K×8bit芯片；
- 13根地址线（A0 —— A12）；
- 8位数据线（D0 —— D7）；
- 输出允许信号（#OE）；
- 写允许信号（#WE）；
- 选片信号（#CE）；
- 状态输出端（READY / #BUSY）。



4. EEPROM的应用

- 可通过程序实现对芯片的读写；
- 仅当READY /#BUSY=1时才能进行“写”操作
- “写”操作的方法：
  - 根据参数定时写入
  - 通过判断READY / #BUSY端的状态进行写入
    - 仅当该端为高电平时才可写入下一个字节。

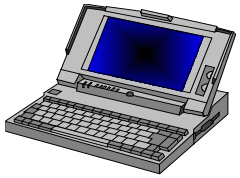
P221例



四、闪速EEPROM

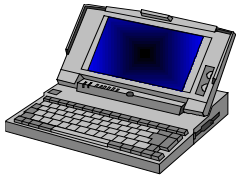
特点：

- 通过向内部控制寄存器写入命令的方法来控制芯片的工作方式。
- 与SRAM的区别：
  - 在进行写入和擦除操作时需要12V编程电压
- 与普通EEPROM的区别：
  - 通过读状态寄存器的内容确定是否可继续写入
  - 提高写命令字的方式控制其处于何种工作方式。



工作方式

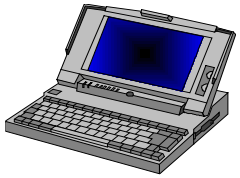
- 数据读出
  - 读单元内容
  - 读内部状态寄存器内容
  - 读芯片的厂家及器件标记
- 编程写入：数据写入，写软件保护
- 擦　　除
  - 字节擦除，块擦除，片擦除
  - 擦除挂起



§5.4 高速缓存（Cache）

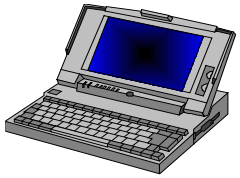
了解：

- Cache的基本概念；
- 基本工作原理；
- 命中率；
- Cache的分级体系结构



Cache的基本概念

- 设置Cache的理由：
  - CPU与主存之间在执行速度上存在较大差异；
  - 高速存储器芯片的价格较高；
- 设置Cache的条件：
  - 程序的局部性原理
    - 时间局部性：
      - 最近的访问项可能在不久的将来再次被访问
    - 空间局部性：
      - 一个进程所访问的各项，其地址彼此很接近



Cache的工作原理

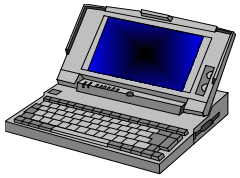
- 将主存和Cache都划分为固定大小的区，由硬件管理

① 将主存一个区中的内容及其地址映像到Cache；

② CPU访问内存时，首先访问Cache，若找到则“命中”；

③ 否则再访问主存，并同时查询Cache中是否有空闲区；

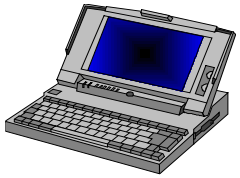
④ 若Cache中有空闲区，则将主存对应区中内容整体调入；若无空闲区，则采用某种替换算法，将Cache某区内容调回主存，腾出空间。



掌握：

- 基本概念：
  - 不同半导体存储器的特点及应用场合
  - Cache的基本概念
- 系统设计：
  - 存储器芯片与系统的连接
  - 译码电路及其他控制信号
  - 存储器扩展技术

} 能够设计出所需要的内存储器



作业

- P236
- 5.7,5.8,5.9,5.12,5.13