第4章 存储器

大 纲



- (一)存储器的分类
- (二)存储器的层次化结构
- (三)半导体随机存取存储器
 - 1. SRAM存储器的工作原理
 - 2.DRAM存储器的工作原理
- (四)只读存储器
- (五)主存储器与CPU的连接
- (六)双口RAM和多模块存储器
- (七)高速缓冲存储器(Cache)
 - 1. 程序访问的局部性原理
 - 2. Cache的基本工作原理

- 3. Cache和主存之间的映射方 式
- 4. Cache中主存块的替换算法
- 5. Cache写策略

(八) 虚拟存储器

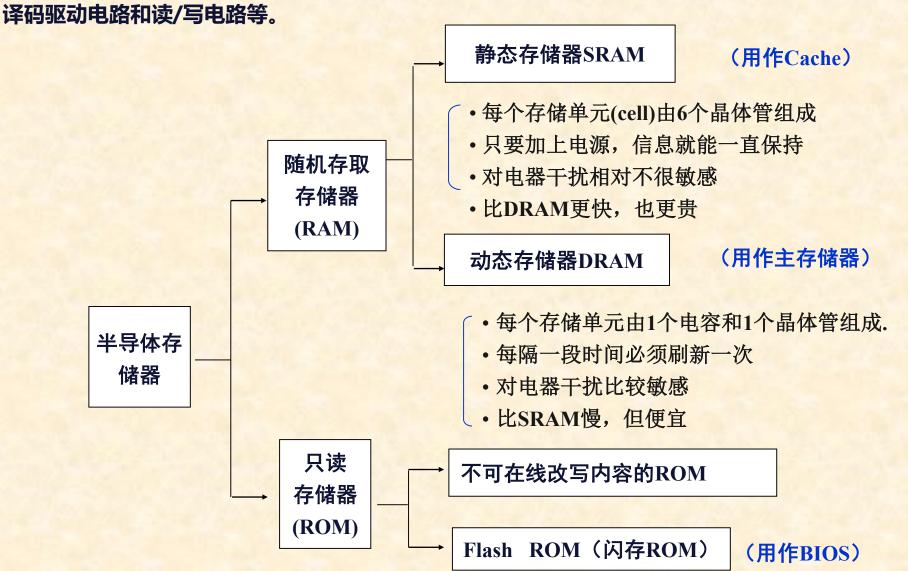
- 1. 虚拟存储器的基本概念
- 2. 页式虚拟存储器
- 3. 段式虚拟存储器
- 4. 段页式虚拟存储器
- 5. TLB (快表)

Contents



内存芯片的多种类型

半导体存储器芯片采用超大规模集成电路制造工艺,在一个芯片内部集成具有记忆功能的存储矩阵、 译码驱动电路和读/写电路等



五、存储器与 CPU 的连接

- 1. 存储器容量的扩展:由于单个芯片的存储器容量有限,为了组成较大容量的存储器,需要多个芯片进行组合。
 - 位扩展(增加存储字长): 由 $mK \times n_1$ 的存储器芯片组成 $mK \times n_2$ 的存储器,需(n_2/n_1)片 $mK \times n_1$ 的存储器芯片。
 - 字扩展(增加存储字数量): 由 $m_1K\times n$ 的存储器芯片 $m_2K\times n$ 的存储器,需(m_2/m_1)片 $m_1K\times n$ 的存储器芯片。
 - 字位同时扩展: $\operatorname{hm}_1 K \times n_1$ 的存储器芯片组成 $\operatorname{m}_2 K \times n_2$ 的存储器,需 $(\operatorname{m}_2/\operatorname{m}_1) \times (\operatorname{n}_2/\operatorname{n}_1)$ 片 $\operatorname{m}_1 K \times \operatorname{n}_1$ 的存储器芯片。

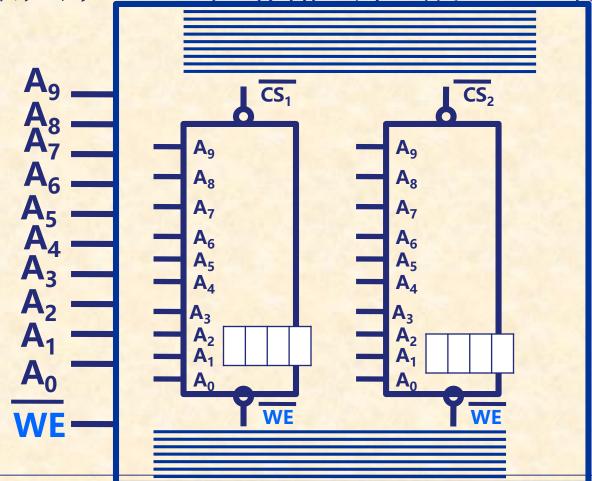
五、存储器与 CPU 的连接



(1) 位扩展(增加存储字长)

10根地址线

用 2片 1K×4位 存储芯片组成 1K×8位 的存储器



8根数据线

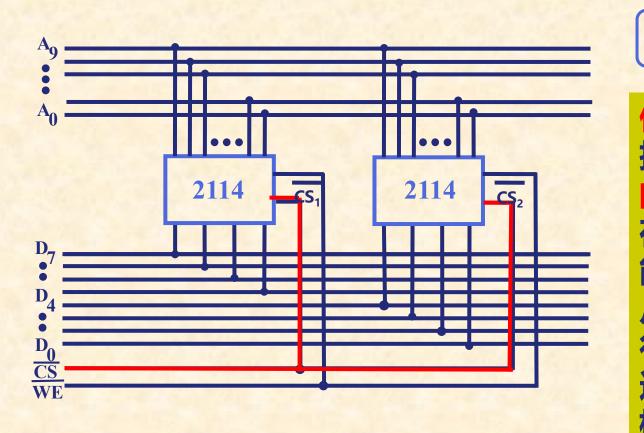
五、存储器与 CPU 的连接

4.2

(1) 位扩展(增加存储字长)

10根地址线

用 2片 1K×4位 存储芯片组成 1K×8位 的存储器



8根数据线

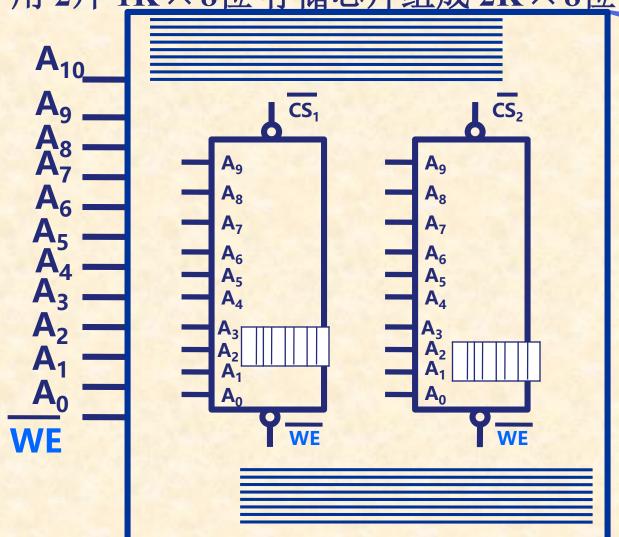


- 1、一块存储芯片的存储容量为: 1K×8位,问存储器需要_____根地址线,需要____根数据线。
- 2、一块存储芯片地址范围为: 4000H~7FFFH, 问该存储器的容量是_____, 与此容量对应的地址线需要 根。

(2) 字扩展(增加存储字的数量)

11根地址线

用 2片 1K×8位 存储芯片组成 2K×8位 的存储器



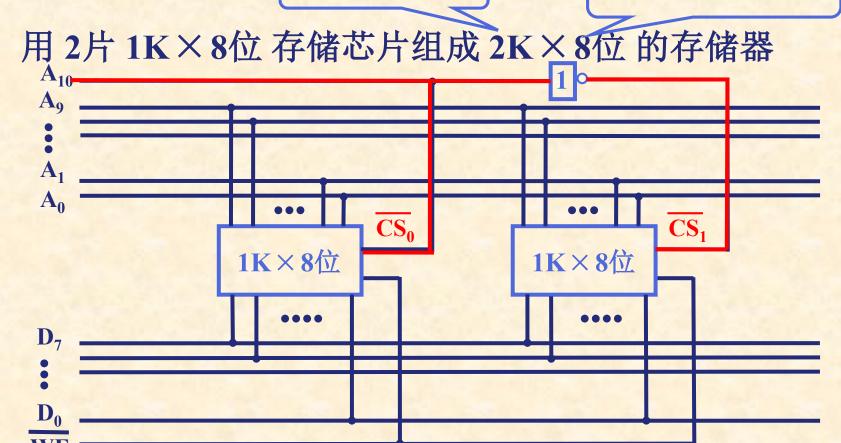
8根数据线

4.2

(2) 字扩展(增加存储字的数量)

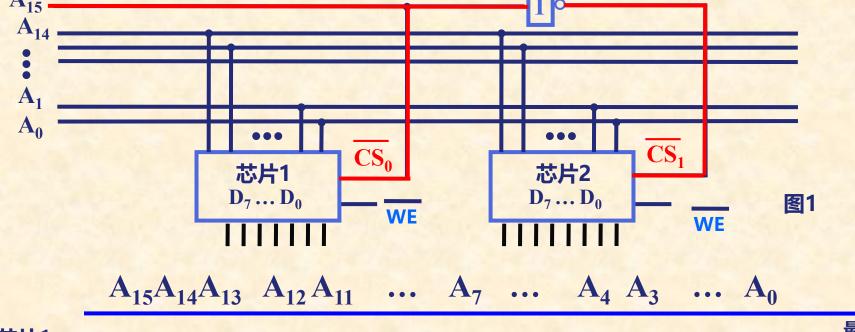
11根地址线

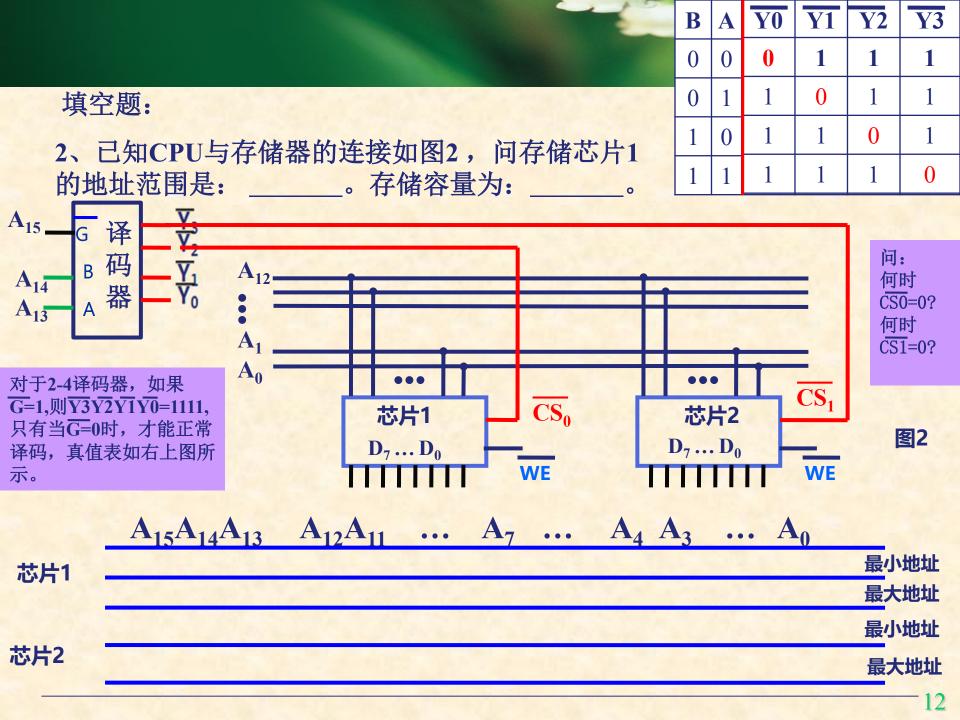
8根数据线



字扩展:参与字扩展的多块芯片分时工作,每块芯片的地址范围不同,通常用未直接连到芯片上的地址的高位经过非门或者译码器等来连接不同芯片的CS引脚,从而保证参与字扩展的芯片的分时工作且地址范围不同。

填空题: 1、已知CPU与存储器的连接如图1,问存储芯片1的地址范围 是: ____。存储容量为: ____。存储芯片2的地址范围 是: 。存储容量为: 。 说明:只有当对应的片选信号CS为低电平时,对应的芯片才选中,地址才能进入该芯片。



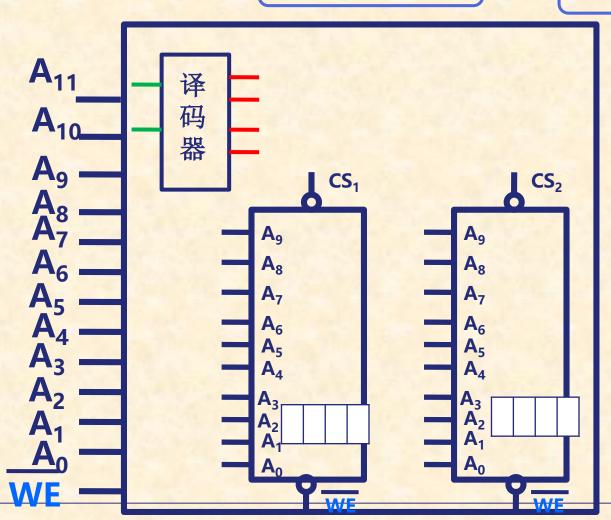


(3)字、位扩展

用 8片 1K×4位 存储芯片组成 4K×8位 的存储器

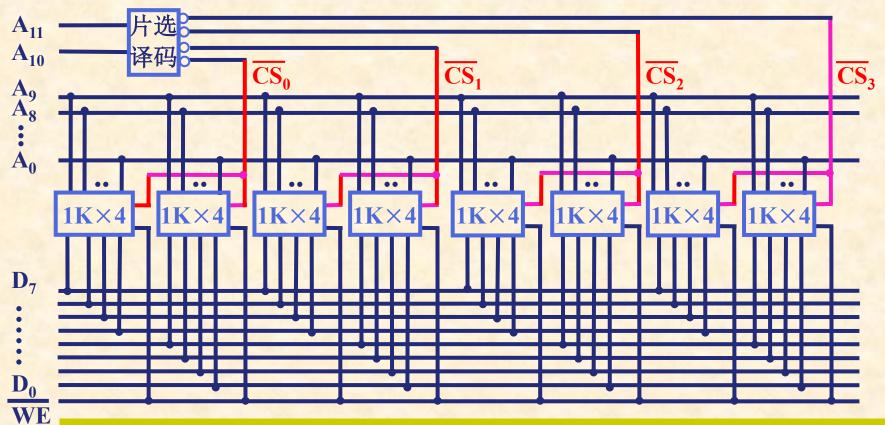
12根地址线

8根数据线



(3)字、位扩展

用 8片 1K×4位 存储芯片组成 4K×8位 的存储器



字位扩展:参与位扩展的多块芯片的CS引脚连接在一起。参与字扩展的多块芯片:通常用未直接连到芯片上的地址的高位经过非门或者译码器等来连接不同芯片的CS引脚,从而保证参与字扩展的芯片的地址范围不同。

储

器

2. 存储器与 CPU 的连接

- (1)地址线的连接:内部地址线、芯片选择线 CPU地址线往往比存储器地址线多,所以将CPU 地址线低位与存储芯片地址相连,CPU地址线高 位或做存储芯片扩充,或做他用。
 - (2) 数据线的连接:数据线对应连接
 - (3) 读/写线的连接: 对应连接
 - (4) 合理选用芯片,片选信号的连接
 - (5) 其他 时序、负载

例1 设CPU有16根地址线、8根数据线,并用MREQ作为访存控制信号(低电平有效),用WR作为读/写控制信号(高电平为读,低电平为写)。现有下列存储芯片:1 K×4位RAM、4 K×8位RAM、8 K×8位RAM、2 K×8位ROM、4 K×8位ROM、4 K×8位ROM、CPU与存储器连接图,要求

- ①主存地址空间分配 6000H~67FFH为系统程序区 6800H~6BFFH为用户程序区
- ②合理选用上述存储芯片,说明各选几片
- ③详细画出存储芯片的片选逻辑图

- 解答: (1) CPU有16根地址线、8根数据线
 - (2) 主存地址空间分配: 6000H~67FFH为系统程序区 6800H~6BFFH为用户程序区

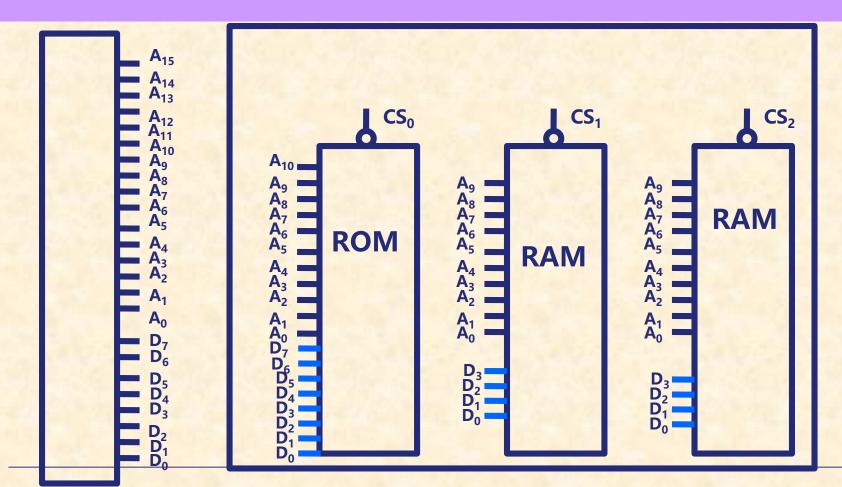
选用芯片时要求容量正好就可以了。如果容量选大了算浪费,不符合要求。

- (3) 系统程序区 空间地址的最后一个为:
- 67FFH-6000H=7FFH。7FFH=(0111,1111,11111)₂ 上面的二进数中有11个"1",可以由地址线A0~A10实现 (7FFH=2¹¹-1=2K-1) 选用一片2K×8的ROM芯片
 - (4) 用户程序区空间地址的最后一个为:
- 6BFFH-6800H=3FFH。 3FFH=(0011,1111,11111)₂ 上面的二进数中有10个"1",可以由地址线A0[~]A9实现 (3FFH=2¹⁰-1=1K-1) 选用二片1K×4的RAM芯片

解答: (1) CPU有16根地址线、8根数据线

(2) 主存地址空间分配: 6000H~67FFH为系统程序区需要11根地址线。6800H~6BFFH为用户程序区需要10根地址线。

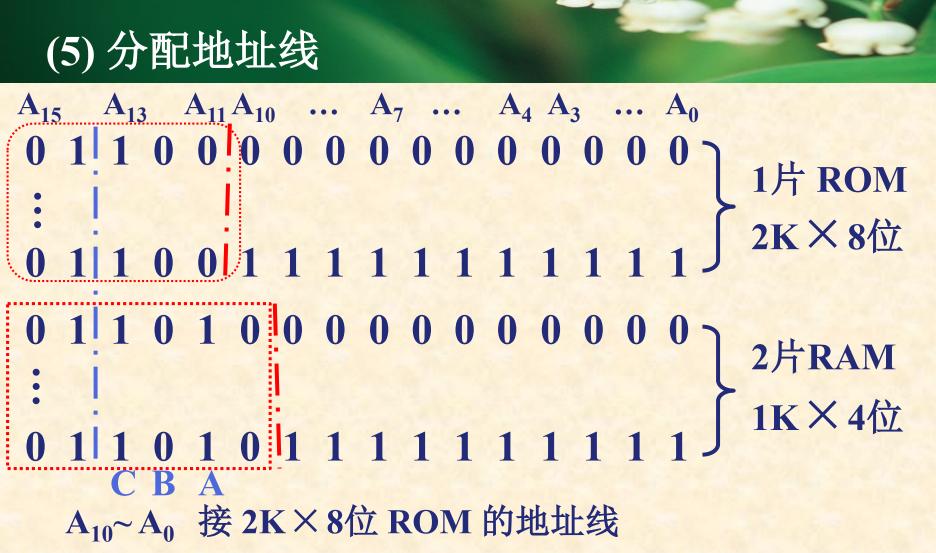
根据容量确定芯片的数量。根据每个芯片的地址范围来确定片选信号CS的连接。





(4) 确定芯片的数量及类型

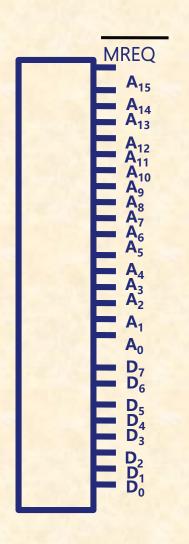
2片1K×4位

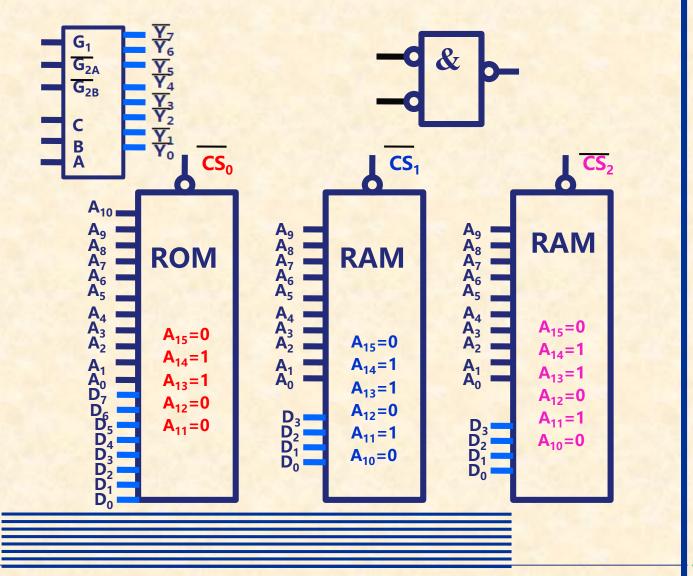


A₀~A₀ 接 1K×4位 RAM 的地址线

(6) 确定片选信号

字位扩展:参与位扩展的多块芯片的CS引脚连接在一起。参与字扩展的多块芯片:根据参与字扩展的芯片的地址范围,从而确定不同的高位地址与CS信号的关系。通常用用未直接连到芯片上的地址的高位经过非门或者译码器等来连接不同芯片的CS引脚。





字位扩展:参与位 例 4.1 CPU 与存储器的连接图 根据参与字 扩展的芯片的地址 范围,从而确定不 $\frac{G_1}{G_{2A}}$ **A**₁₄ 同的高位地址与CS $\overline{\mathbf{Y}}_{5}$ A₁₅ & 信号的关系。通常 $\overline{G_{2B}}$ A₁₃ 上的地址的高位经 **MREQ** $\overline{\mathbf{Y}_4}$ A₁₂ B 等来连接不同芯片 A_{11} 的CS引脚。 A_{10} A₉ : A₀ CS₂ CS₀ CS₁ 2K×8位 PD/Progr 1K×4位 1K×4位 **ROM** RAM **RAM** WR \mathbf{D}_7 \mathbf{D}_4 $\mathbf{D_3}$ \mathbf{D}_0 WR

例 4.2

例4.2 假设同前,要求最小 8K为系统程序区,相邻 16K为用户程序区。最大 4K为系统程序区工作区。

解答: ①主存地址空间分配: 8K为系统程序区 16K为用户程序区 4K为系统程序区工作区

- ②系统程序区: 8K=2¹³。需要13根地址线,可以由地址线 A0~A12实现。由ROM构成。
- ③用户程序区: 16K=2 × 8K; 8K=2¹³。需要13根地址线,可以由地址线A0[~]A12实现。由RAM构成。
- 系统程序工作区: 4K=2¹²。需要12根地址线,可以由地址 线A0[~]A11实现。由RAM构成。

例 4.2

根据容量确定芯片的数量。根据每个芯片的地址范围来确定片选信号CS的连接。

选用芯片时要求容量正好就可以了。如果容量选大了算浪费,不符合要求。

- (1) 写出对应的二进制地址码
- (2) 确定芯片的数量及类型(根据容量选择芯片的数量)

1片8K×8位 ROM 2片8K×8位 RAM

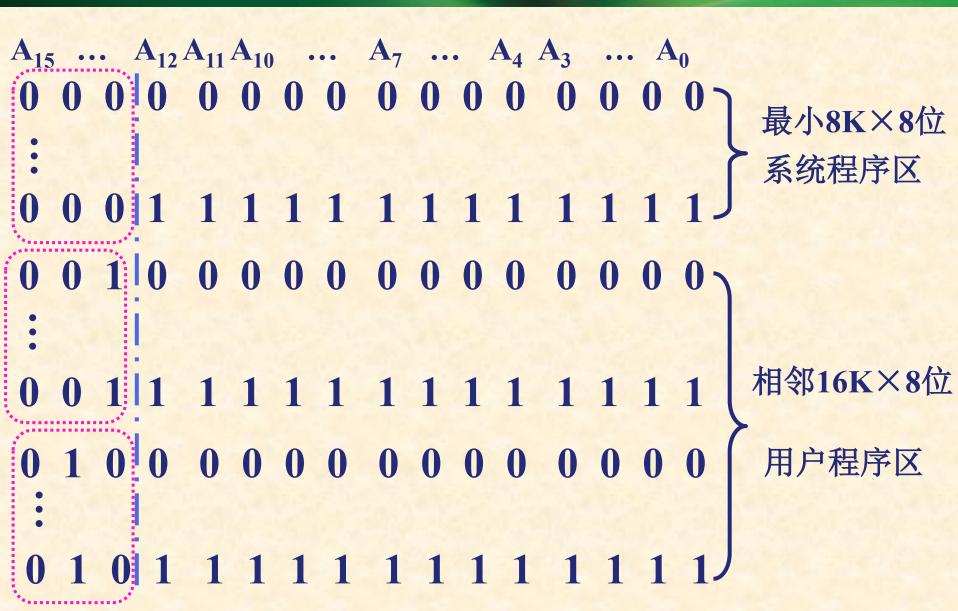
1片 4K × 8位 RAM 每个芯片需要一个CS片选信号。分别称为: CS0、CS1、CS2、CS3

(3) 分配地址线

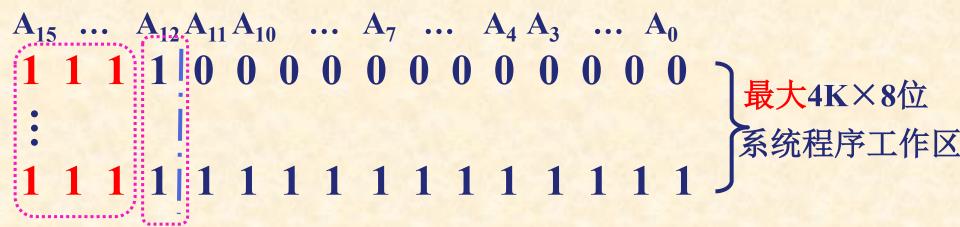
A₁₂~A₀ 接 ROM 和 8K × 8位 RAM 的地址线 A₁₁~A₀ 接 4K × 8位 RAM 的地址线

(4) 确定片选信号 用地址高位来确定连接CS0、CS1、CS2、CS3的方法。

(1) 写出对应的二进制地址码

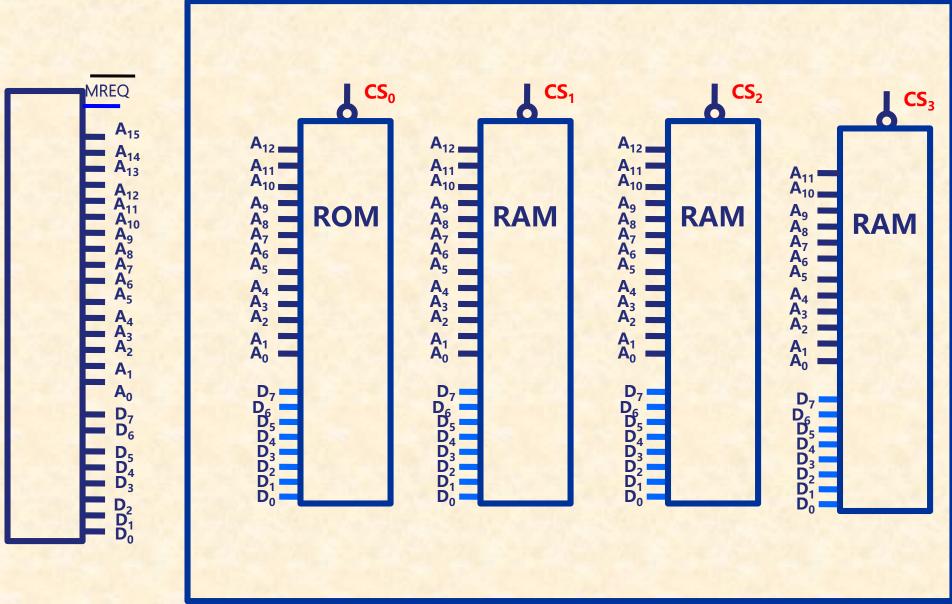


(1) 写出对应的二进制地址码

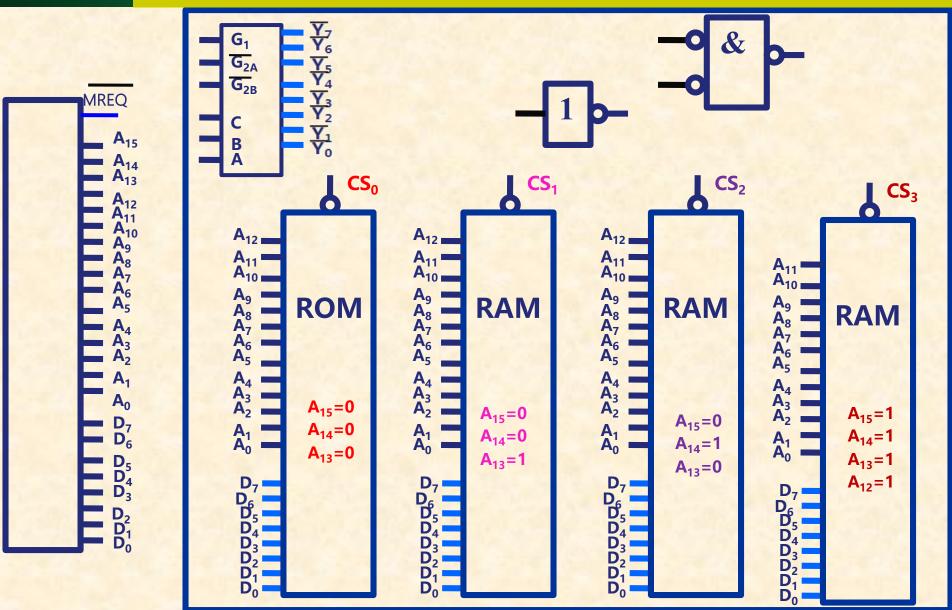


(2) 确定芯片的数量及类型

- 最小8KB系统程序区: 1片8K×8位ROM;
- 与其相邻的16KB用户程序区:选择2片8K×8位RAM;
- 最大4KB系统程序工作区:选择1片4K×8位RAM。

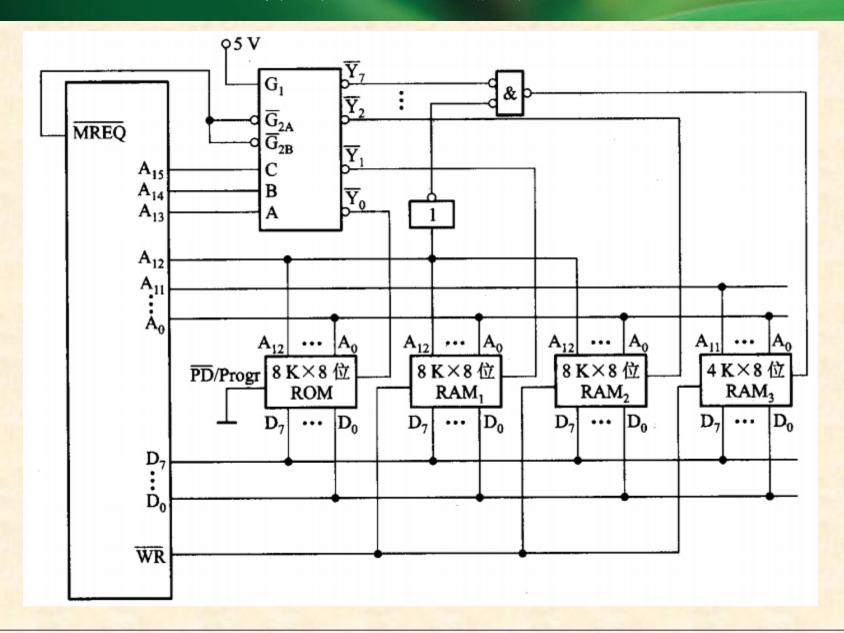


字位扩展:参与位扩展的多块芯片的CS引脚连接在一起。参与字扩展的多块芯片:根据参与字扩展的芯片的地址范围,从而确定不同的高位地址与CS信号的关系。通常用未直接连到芯片上的地址的高位经过非门或者译码器等来连接不同芯片的CS引脚。



2.8

例4.2 CPU 与存储器的连接图



例3: CPU和主存的连接

CPU地址线A15~A0,数据线D7~D0,WR为读/写信号,MREQ为访存请求信号。0000H~3FFFH为BIOS区,4000H~FFFFH为用户程序区。用8K×4位ROM芯片和16K×8位RAM芯片构成该存储器,要求说明地址译码方案,并将ROM芯片、RAM芯片与CPU连接。

解:因为0000H~3FFFH为BIOS,故ROM区高两位总是00,低14位为全译码。

ROM区大小为: 2¹⁴×8位=16K×8位=16KB, ROM芯片数为:

 $16K \times 8位 / 8K \times 4位 = 2 \times 2 = 4$,字方向扩展2倍,位方向扩展2倍。

ROM芯片内地址位数为13位,连到CPU低13位地址线A12~A0。

因为4000H~FFFFH为用户程序区,故RAM区高两位是01、10、11, 低14位为全译码。RAM区大小为: 3×2¹⁴×8位=3×16K×8位=48KB。

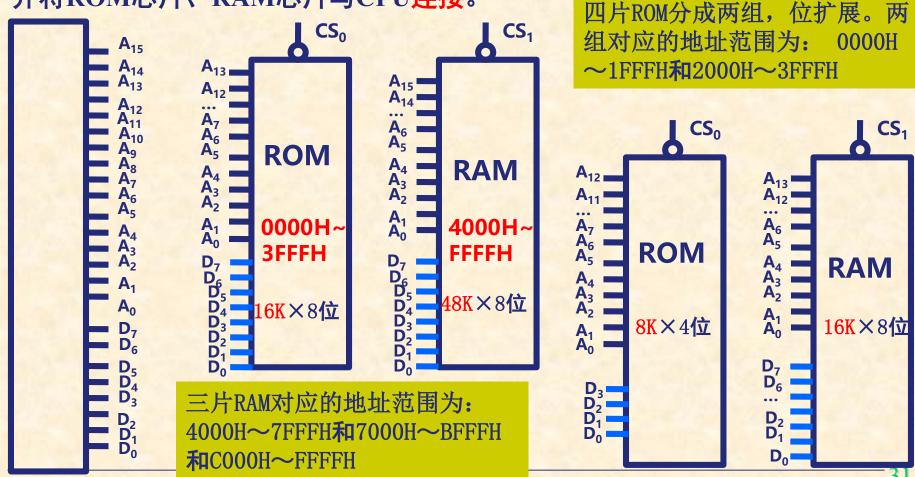
RAM芯片数为:

 $48K \times 8位 / 16K \times 8位 = 3 \times 1 = 3$,字方向上扩展3倍,位方向上不扩展。 RAM芯片内地址位数为14位,连到CPU低14位地址线A13 \sim A0。

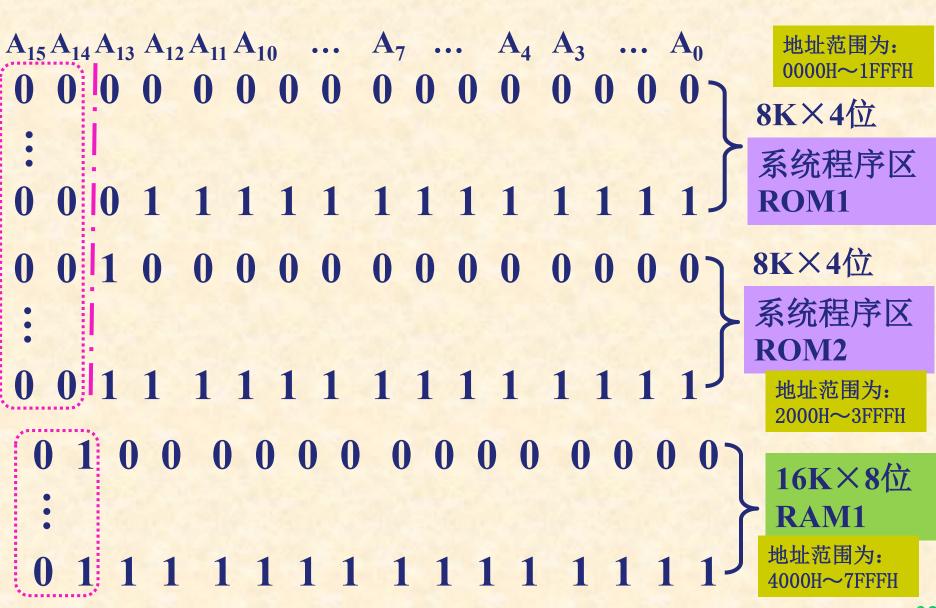
例3: CPU和主存的连接

CPU地址线A15~A0,数据线D7~D0,WR为读/写信号,MREQ为访存请求信号。0000H~3FFFH为BIOS区,4000H~FFFFH为用户程序区。用8K×4位ROM芯片和16K×8位RAM芯片构成该存储器,要求说明地址译码方案,

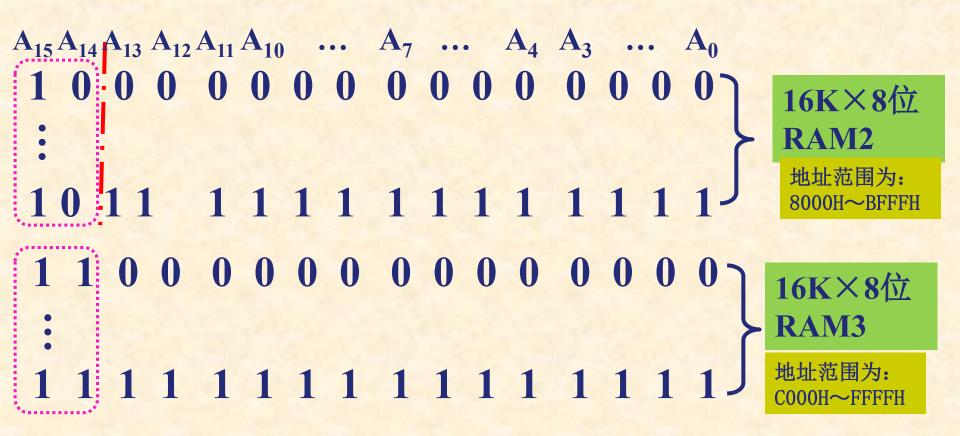
并将ROM芯片、RAM芯片与CPU连接。



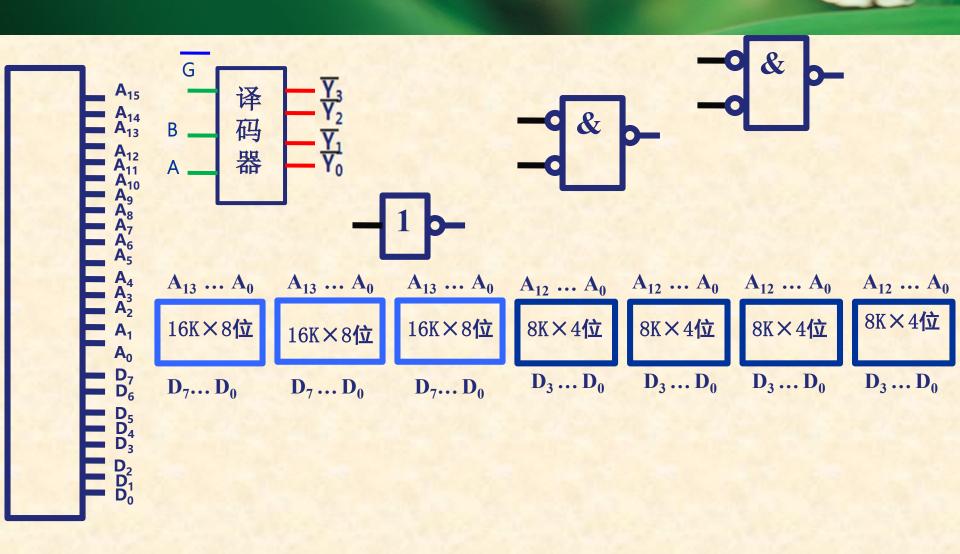
(1) 写出对应的二进制地址码



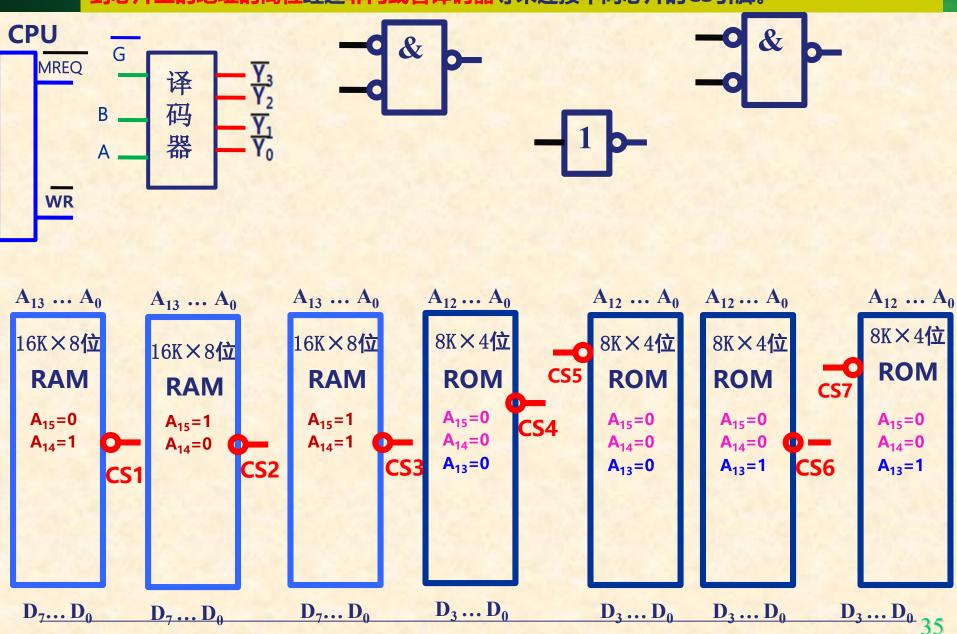
(1) 写出对应的二进制地址码

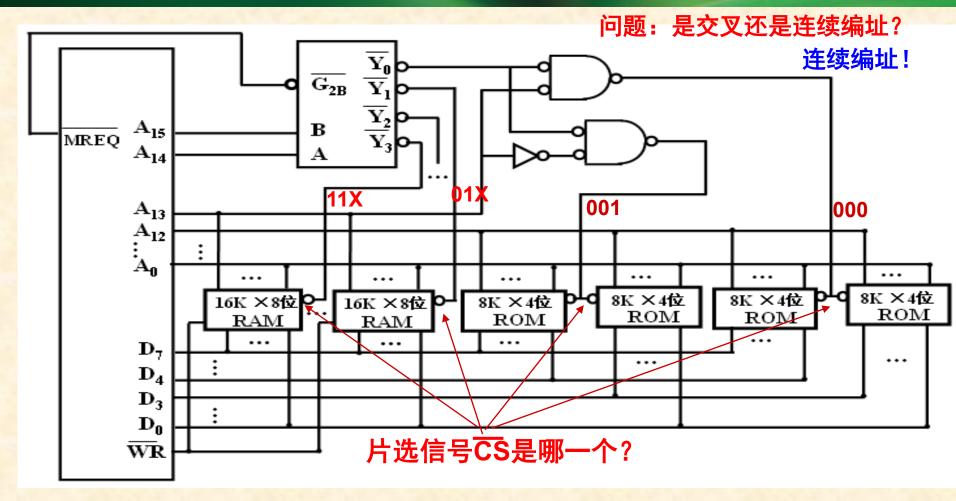


根据容量确定芯片的数量。根据每个芯片的地址范围来确定片选信号CS的连接。



字位扩展:参与位扩展的多块芯片的CS引脚连接在一起。参与字扩展的多块芯片:根据参与字扩展的芯片的地址范围,从而确定不同的高位地址与CS信号的关系。通常用未直接连到芯片上的地址的高位经过非门或者译码器等来连接不同芯片的CS引脚。





问题:为什么WR不连到ROM芯片上? ROM芯片只读不写

问题: MREQ信号的作用是什么? 有效(低电平)时,表示选中主存读写。

存储器接口

(1)8位存储器接口

如果数据总线为8位,则存储器只能按字节编址

(2) 16位存储器接口

16位存储器系统由2个存储体组成,存储体选择通过选择信号BHE实现。

如果要传送一个16位数,则2个存储器被选中如果要传送一个8位数,则1个存储器被选中

(3) 32位存储器接口

32位存储器系统由4个存储体组成,存储体选择通过选择信号BE3~BE0实现。

如果要传送一个32位数,则4个存储器被选中如果要传送一个16位数,则2个存储器被选中如果要传送一个8位数,则1个存储器被选中

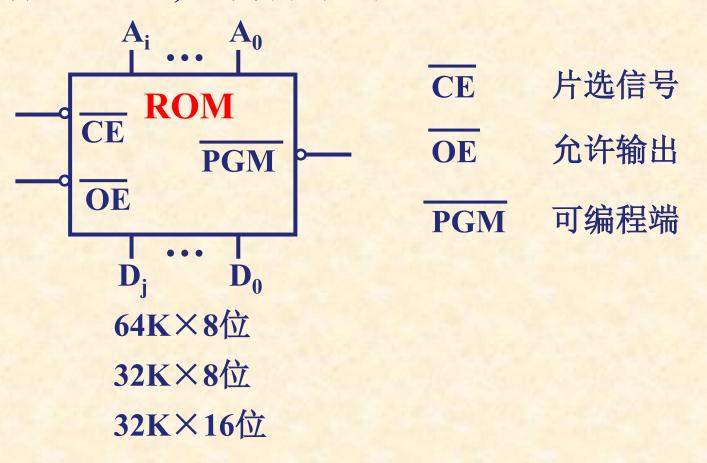
存储器接口

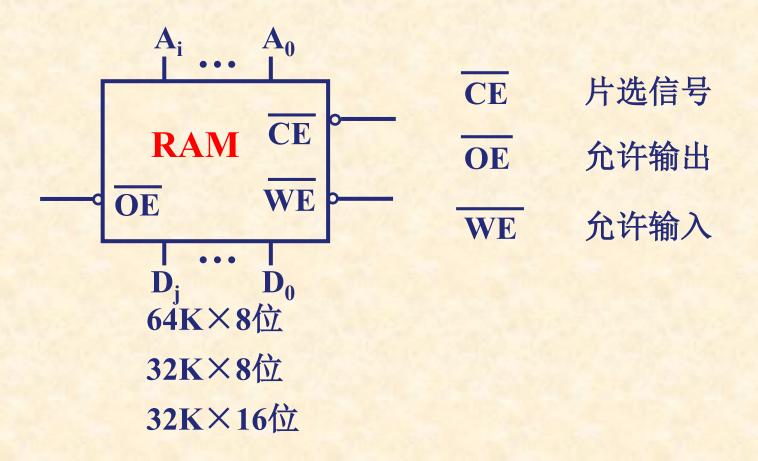
(4) 64位存储器接口

64位存储器系统由8个存储体组成,存储体选择通过选择信号BE7~BE0实现。

如果要传送一个64位数,则8个存储器被选中如果要传送一个32位数,则4个存储器被选中如果要传送一个16位数,则2个存储器被选中如果要传送一个8位数,则1个存储器被选中

例 4.4 (了解) 设 CPU 有 20 根地址线,16 根数据线。并用 IO/M 作访存控制信号。RD 为读命令,WR 为写命令。现有 PROM, 外特性如下:



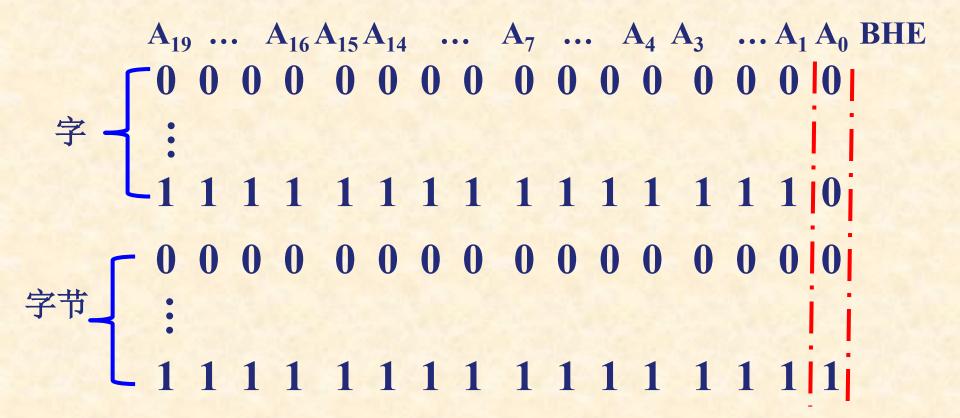


用 138 译码器及其他门电路(门电路自定)

- (1) CPU按字节访问和按字访问的地址范围各多大?
- (2) CPU按字节访问时需分奇偶体,且最大64KB为 系统程序区,与其相邻的64KB为用户程序区,写出 每片存储芯片对应的二进制地址码。
- (3) 画出CPU与存储芯片连接图。

BHE	A0	访问形式
0	0	字
0	1	奇字节
1	0	偶字节
1	1	不访问

(1) 写出对应的二进制地址码

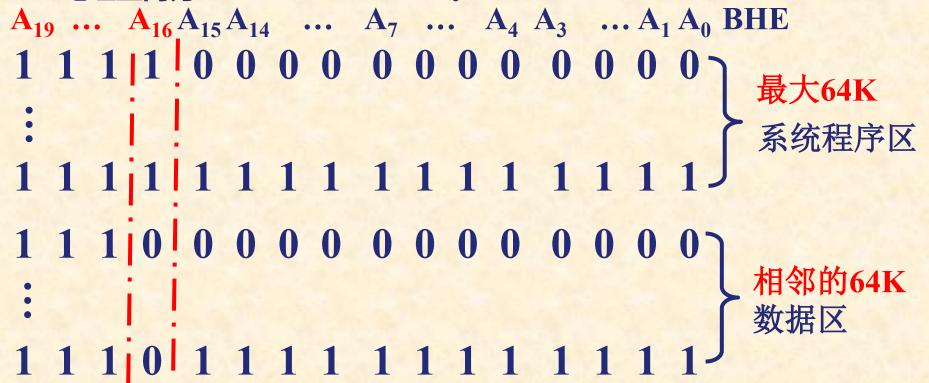


根据上页的表格可知:按字进行访问时,A0必须为0,这时只有A19...A1的值可以从00...00变动到11...11,故有地址空间为2¹⁹=512KB。按字节进行访问时,A0可以为0也可以为1,这时有A19...A0的值可以从00...00变动到11...11,故有地址空间为2²⁰=1MB。

(2) 按字节访问时:

 $64K=64 \times 2^{10}=2^{16}$

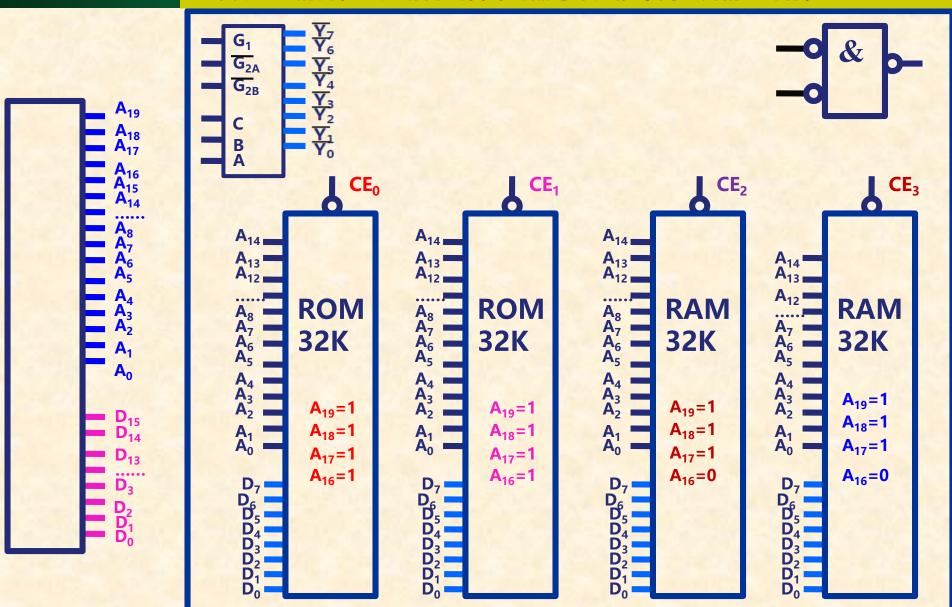
这个地址空间需要16根地址线与之对应,用A0~A15对应。最大的64K,对应的地址空间为: F0000H~FFFFFH。相邻的64K,对应的地址空间为: E0000H~EFFFFH。

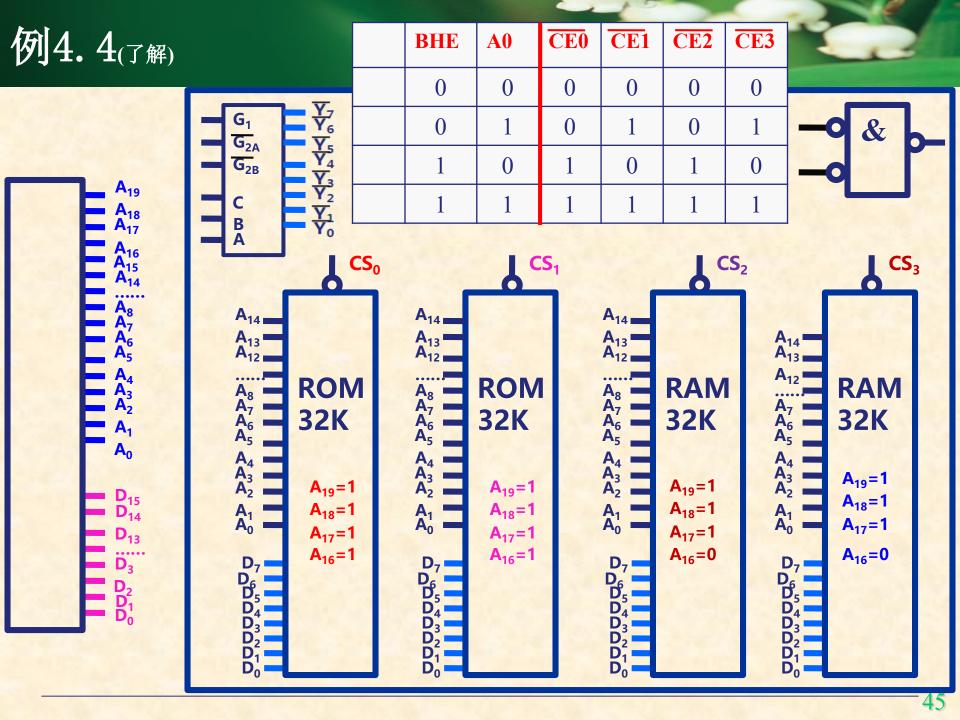


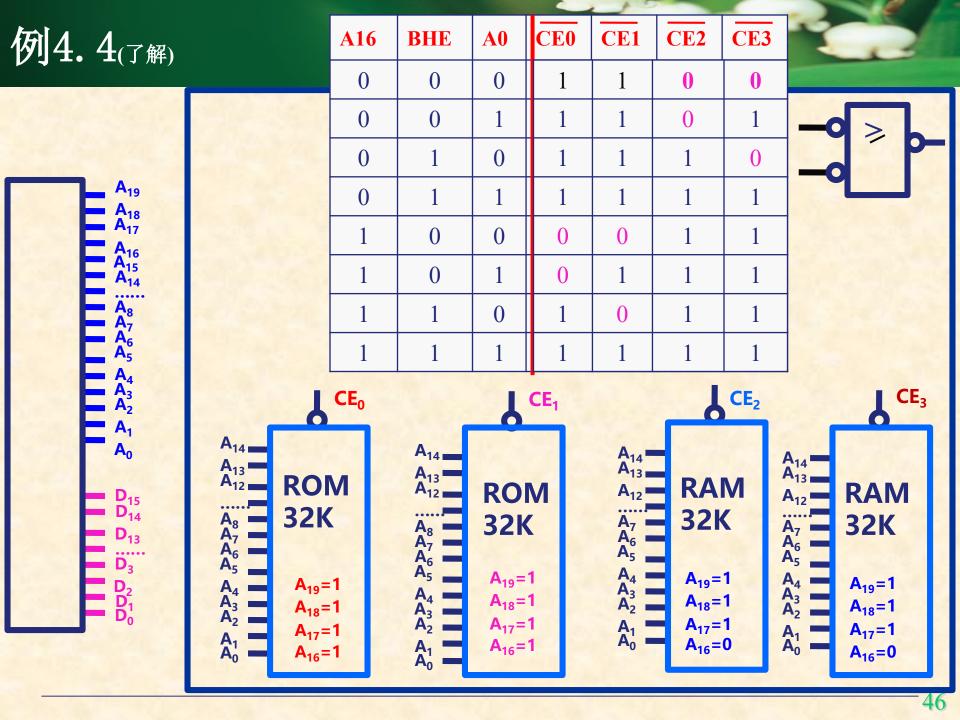
题目要求可以按字访问,也可以按字节访问。所以不能直接选16位的存储芯片,只能选8位数据线的存储芯片。为了奇字节和偶字节能分开,需要对应到2个芯片里面。故采用2个32K ×8的ROM,2个32K ×8 的RAM。如下页图:

例4.4(了解)

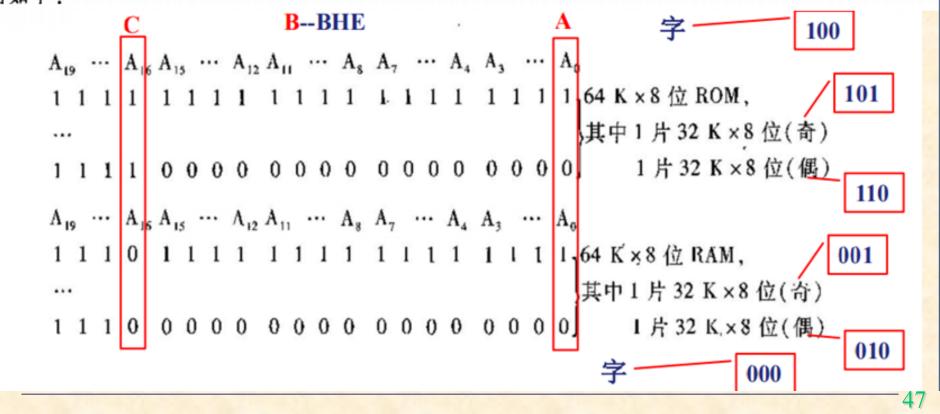
字位扩展:参与位扩展的多块芯片的CE引脚连接在一起。参与字扩展的多块芯片: 根据参与字扩展的芯片的地址范围,从而确定不同的高位地址与CS信号的关系。 通常用<mark>地址的高位</mark>经过<mark>非门或者译码器</mark>等来连接不同芯片的CS引脚。

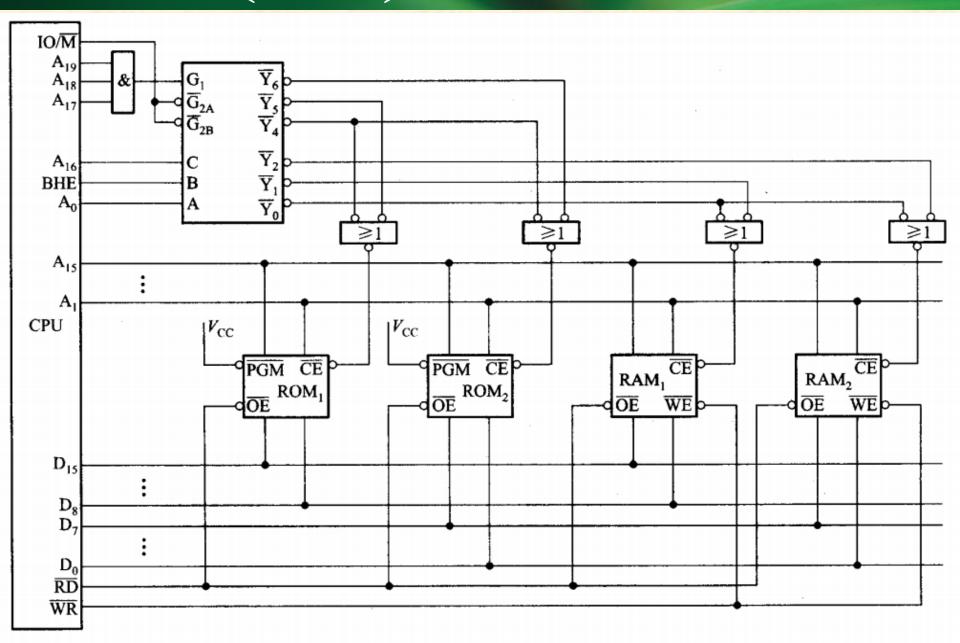






- 解:(1) CPU 按字节访问的地址范围为 1 M, CPU 按字访问的地址范围是 512 K。
- (2)由于 CPU 按字节访存时需区分奇偶体,并且还可以按字访问,因此如果选择 64 K×8位的芯片,按字节访问时体现不出奇偶分体;如果选择 32 K×16位的芯片,虽然能按字访问,但不能满足以字节为最小单位。故一律选择 32 K×8位的存储芯片,其中系统程序区 64 KB 选择 2 片32 K×8位 RAM。它们对应的二进制地址范围如下:





六、存储器的校验

- 数据校验码: 是一种常用的带有发现某些错误或带有自动纠错能力的数据编码系统。
- 数据校验码检错纠错方法: "冗余校验"。

n位(信息位) k位(校验位)

码字和码距

- 由若干位代码组成的一个字叫"码字"
- 两个码字中具有不同代码位的个数叫这两个码字间的"距离"
- 码制中各码字间最小距离为"码距",它就是这个码制的距离。

问题: "8421"码的码距是几?

- 2 (0010) 和3 (0011) 间距离为1, "8421"码制的码距为1。
- 数据校验中"码字"指数据位和校验位按某种规律排列得到的代码
- 常用的数据校验码有:

奇偶校验码、海明校验码、循环冗余校验码 (CRC)。

码距与检错、纠错能力的关系

编码的纠错、检错能力与编码的最小距离有关

$$L-1=D+C(D\geq C)$$

L:编码的最小距离

D: 检测错误的位数

C: 纠正错误的位数

L = 3

具有一位 检错能力

具有一位 纠错能力

码距与检错、纠错能力的关系 (当b≤4)

- ① 若码距L为奇数,则能发现L-1位错,或能纠正(L-1)/2位错。
- ② 若码距L为偶数,则能发现L/2位错,并能纠正(L/2-1)位错。

海明码是具有一位纠错能力的编码

奇偶校验码

- 奇校验:每个码字(包括校验位)中1的个数为奇数
- 偶校验:每个码字(包括校验位)中1的个数为偶数

奇偶校验码的特点

- 问题: 奇偶校验码的码距是几? 为什么?
 - 码距b=2。

在奇偶校验码中, 若两个数中有奇数位不同,则它们相应的校验位就不同;若有偶数位不同,则虽校验位相同,但至少有两位数据位不同。因而任意两个码字之间至少有两位不同。

特点

- 根据码距和纠/检错能力的关系,它只能发现奇数位出错,不能发现偶数位出错,而且也不能确定发生错误的位置,不具有纠错能力。
- 开销小,适用于校验一字节长的代码,故常被用于存储器读写检查 或按字节传输过程中的数据校验

因为一字节长的代码发生错误时,1位出错的概率较大,两位以上出错则很少,所以可用奇偶校验。

海明校验码

- 海明码:是一种能够纠正一位错误或者检测两位错误并纠正一位错误的线性编码系统。
 - 1950年由贝尔实验室的Richarb Hamming首先提出
 - 主要用于存储器校验
- 基本思想: 分组校验 (多重奇偶校验)
 - 将有效信息按某种规律分为若干组,每组安排一个校验位进行奇偶校验。
 - 在数据位组中加入几个校验位,增加了数据代码间的码距。
 - 当某一位发生变化时会引起校验结果发生变化,不同代码位上的 错误会得出不同的校验结果。

各检测位 C; 所承担的检测小组为

C₁ 检测的 g₁ 小组包含第 1, 3, 5, 7, 9, 11,

C2 检测的 g2 小组包含第 2, 3, 6, 7, 10, 11, …

C₄ 检测的 g₃ 小组包含第 4, 5, 6, 7, 12, 13,…

C₈ 检测的 g₄ 小组包含第 8, 9, 10, 11, 12, 13, 14, 15, 24,…

 g_i 小组独占第 2^{i-1} 位

 g_i 和 g_j 小组共同占第 $2^{i-1}+2^{j-1}$ 位

 g_i 、 g_i 和 g_l 小组共同占第 $2^{i-1}+2^{j-1}+2^{l-1}$ 位

海明校验码

1、原理

- (1) 特点: 能检测出两位同时出错、亦能检测出一位出错并能自动纠错。
- (2) 实现原理:

在n个数据位之外加上k个校验位,从而形成一个n十k位的新码字,当某一位出错后,就会引起相关的几个校验位的值发生变化,从而达到检错、纠错的目的。

n位(信息位) k位(校验位)

K个校验位可以形成2^k种不同组合,每种组合对应一种出错的状态或者完全正确那种状态。

数据位n与校验位k的对应关系:

n	k(最小)
1	2
2~4	3
5~11	4
12~26	5
27~57	6

2^{k-1}≥n+k (一位出错并纠错并发现两位错) 码距为4

n	k(最小)
1~4	4
5~11	5
12~26	6
27~57	7
58~120	8

求信息码的海明校验码(续)

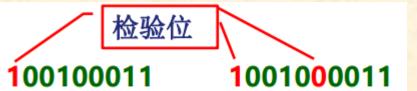
- 每个校验位C_i放在海明码中位号为2ⁱ⁻¹的位置
- 海明码的每一位码H_i(包括数据位和校验位)由 多个校验位进行校验:

如果数据位为 $b_4b_3b_2b_1$,校验位 $C_1C_2C_4$,则有:

汉明码排序如下:

二进制序号	1	2	3	4	5	6	7
	C ₁						

偶校验 00100011



海明码的分组是非划分方式,组间有交叉。



第1组: 1 3 5 7

第2组: 2 3 6 7

第3组: 4 5 6 7

1: 001 2: 010 4: 100

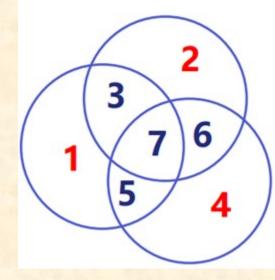
分组看"1"

3: 011 3: 011 5: 101

5: 101 6: 110 6: 110

7: 111 7: 111 7: 111







给出错误位置

例4.4

求 0101 按 "偶校验" 配置的汉明码

根据
$$2^k \ge n + k + 1$$

得k=3

汉明码排序如下:

二进制序号	1	2	3	4	5	6	7
名称	C ₁	C ₂	0	C ₄	1	0	1

: 0101 的汉明码为 0100101

练习1 按配偶原则配置 0011 的汉明码

解:
$$n=4$$
 根据 $2^k \ge n+k+1$ 取 $k=3$

二进制序号	1	2	3	4	5	6	7
名称	\mathbf{C}_1	\mathbb{C}_2	0	C ₄	0	1	1
	1	0		0			
$C_1 = 3 \oplus 5$	5 ⊕ 7	= 1					
$C_2 = 3 \oplus 6$	5⊕7	= 0					
$C_4 = 5 \oplus 6$	5⊕7	= 0					

:. 0011 的汉明码为 1000011

3. 汉明码的纠错过程

形成新的检测位 P_i ,其位数与增添的检测位有关,如增添 3 位 (k=3),新的检测位为 P_4 P_2 P_1 。 以 k=3 为例, P_i 的取值为

$$P_{1} = \overset{\mathbf{C}_{1}}{1} \oplus 3 \oplus 5 \oplus 7$$

$$P_{2} = \overset{\mathbf{C}_{2}}{2} \oplus 3 \oplus 6 \oplus 7$$

$$P_{4} = \overset{\mathbf{C}_{4}}{4} \oplus 5 \oplus 6 \oplus 7$$

对于按"偶校验"配置的汉明码不出错时 $P_1=0$, $P_2=0$, $P_4=0$

例4.5

己知接收到的汉明码为0100111

(按配偶原则配置) 试问要求传送的信息是什么?解: 纠错过程如下

$$P_1=1 \oplus 3 \oplus 5 \oplus 7=0$$
 无错

$$P_2=2 \oplus 3 \oplus 6 \oplus 7=1$$
 有错

$$P_4P_2P_1 = 110$$

第6位出错,可纠正为0100101, 故要求传送的信息为0101。

练习2

写出按偶校验配置的汉明码

$$P_4 = 4 \oplus 5 \oplus 6 \oplus 7 = 1$$

$$P_2 = 2 \oplus 3 \oplus 6 \oplus 7 = 0$$

$$\mathbf{P_1} = \mathbf{1} \oplus \mathbf{3} \oplus \mathbf{5} \oplus \mathbf{7} = \mathbf{0}$$

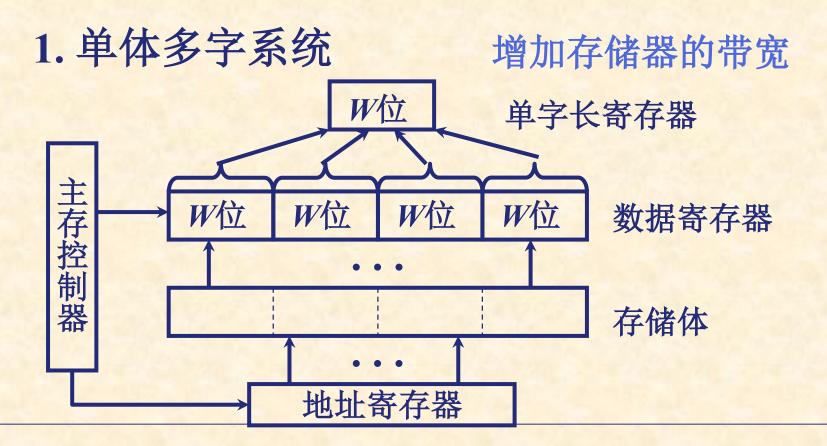
∴ P₄P₂P₁ = 100 第 4 位错,可不纠

练习3 按配奇原则配置 0011 的汉明码

配奇的汉明码为 0101011

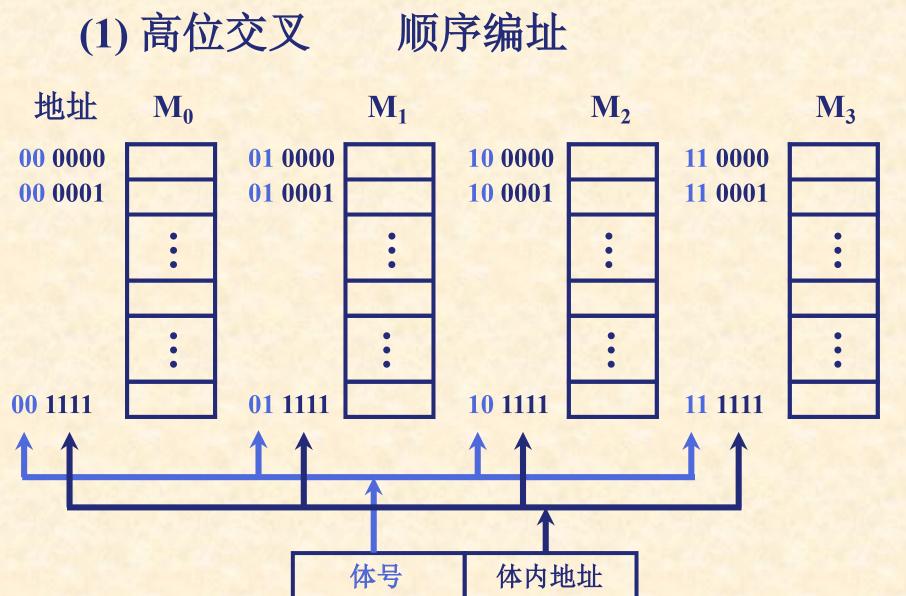
七、提高访存速度的措施

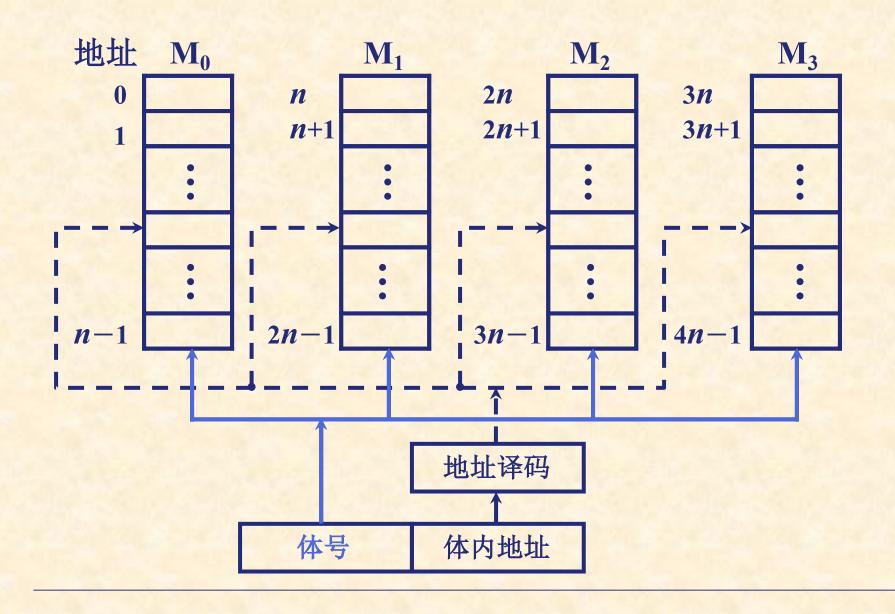
- 采用高速器件
- ·采用层次结构 Cache -主存
- 调整主存结构



2. 多体并行系统

4.2

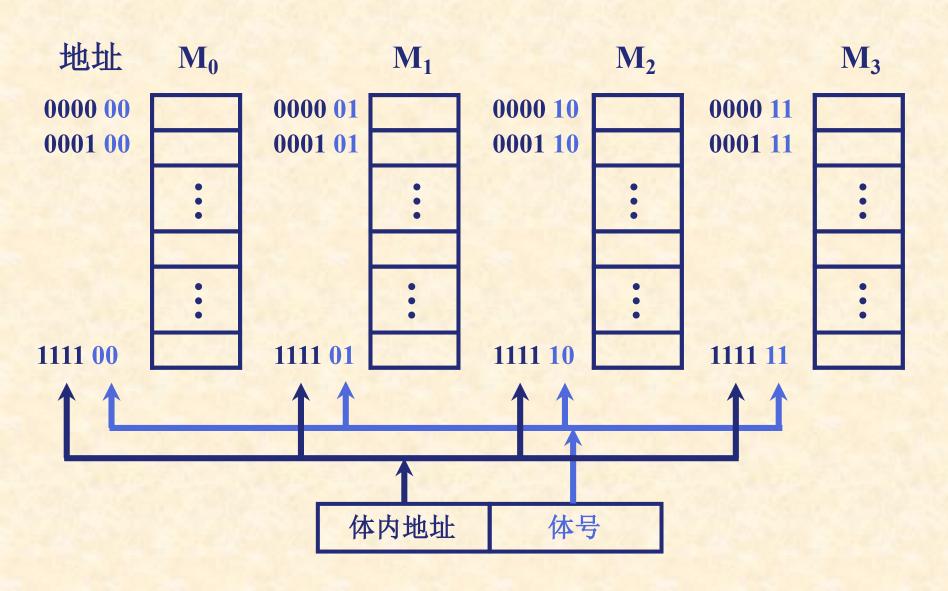




(2) 低位交叉

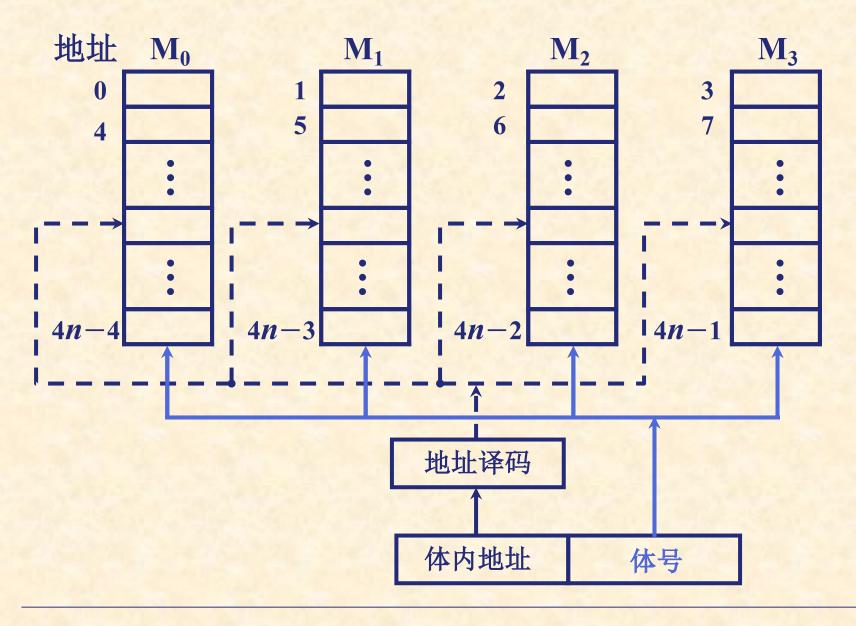
各个体轮流编址



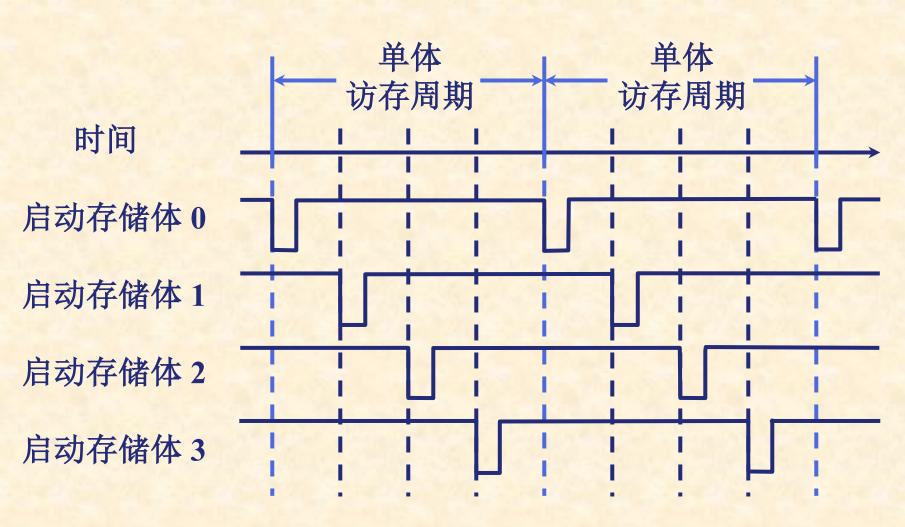


(2) 低位交叉 各个体轮流编址

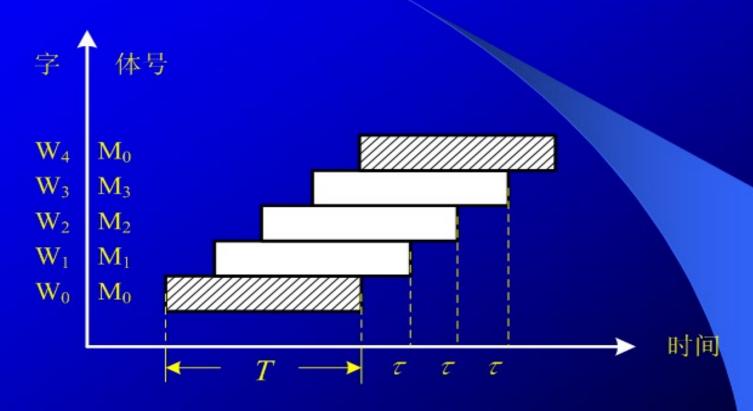




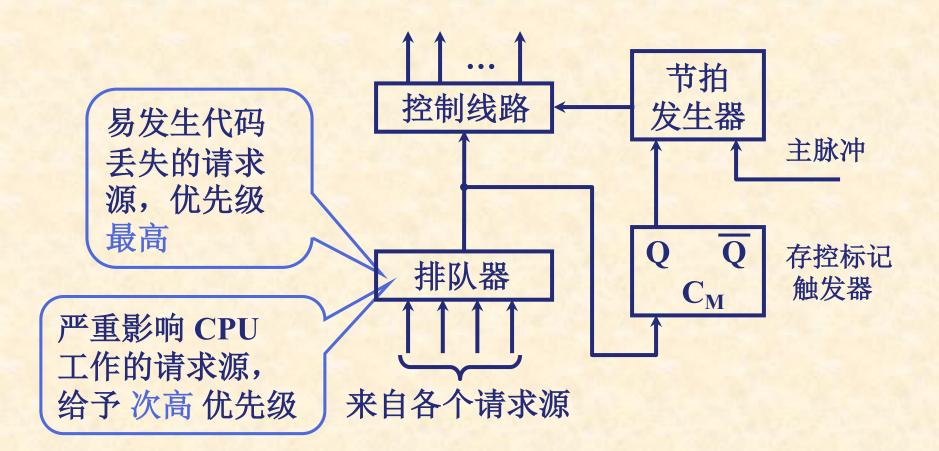
在不改变存取周期的前提下,增加存储器的带宽



设四体低位交叉存储器,存取周期为T,总线传输周期为 τ ,为实现流水线方式存取,应满足 $T=4\tau$ 。



连续读取 4 个字所需的时间为 $T+(4-1)\tau$



3.高性能存储芯片

(1) SDRAM (同步 DRAM)

在系统时钟的控制下进行读出和写入 CPU 无须等待

(2) RDRAM

由 Rambus 开发,主要解决存储器带宽问题

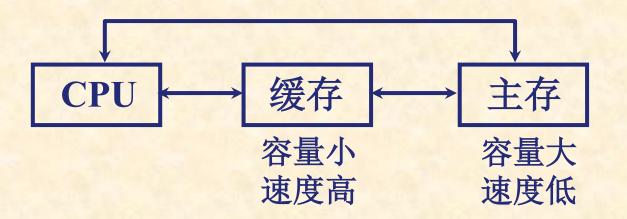
(3) 带 Cache 的 DRAM

在 DRAM 的芯片内 集成 了一个由 SRAM 组成的 Cache, 有利于 猝发式读取

4.3 高速缓冲存储器

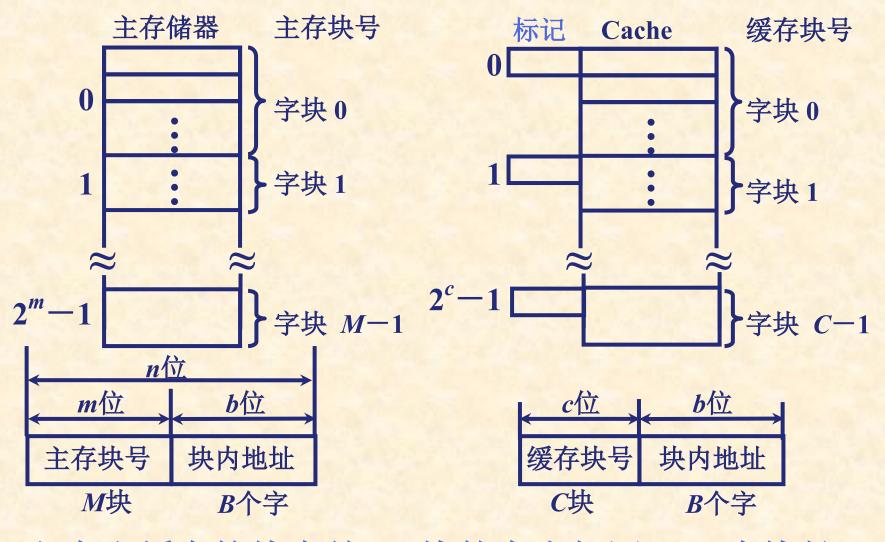
一、概述

1. 问题的提出 避免 CPU "空等"现象 CPU 和主存(DRAM)的速度差异



程序访问的局部性原理

(1) 主存和缓存的编址



主存和缓存按块存储

块的大小相同

B为块长

缓存共有 C 块 主存共有 M 块 M >> C

命中 主存块 调入 缓存

主存块与缓存块 建立 了对应关系

用标记记录与某缓存块建立了对应关系的主存块号

未命中 主存块 未调入 缓存

主存块与缓存块 未建立 对应关系

(3) Cache 的命中率

CPU 欲访问的信息在 Cache 中的 比率

命中率与 Cache 的 容量与 块长 有关

一般每块可取 4~8 个字

块长取一个存取周期内从主存调出的信息长度

CRAY_1 16体交叉 块长取 16 个存储字 IBM 370/168 4体交叉 块长取 4 个存储字 (64位×4=256位)

(4) Cache –主存系统的效率

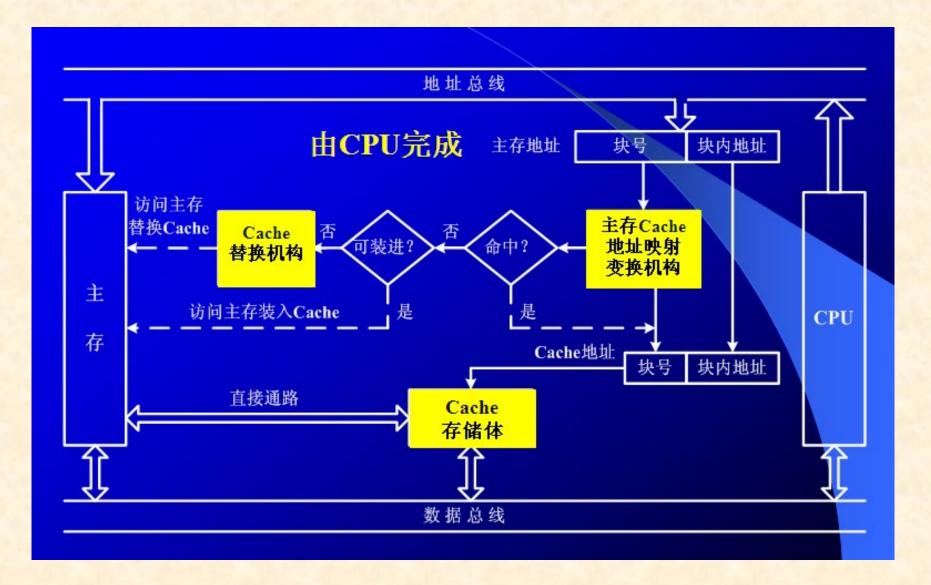
效率e与命中率有关

$$e = \frac{$$
 访问 Cache 的时间 \times 100%

设 Cache 命中率 为 h, 访问 Cache 的时间为 t_c , 访问 主存 的时间为 t_m

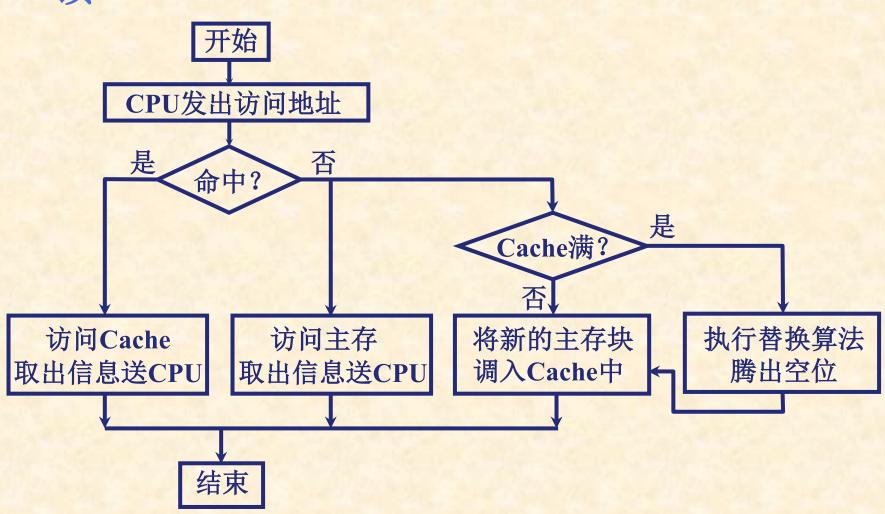
则
$$e = \frac{t_c}{h \times t_c + (1-h) \times t_m} \times 100\%$$

3. Cache 的基本结构



4. Cache 的 读写 操作





4. Cache 的 读写 操作

写 Cache 和主存的一致性

- •写直达法(Write through) 写操作时数据既写入Cache又写入主存 写操作时间就是访问主存的时间,读操作时不 涉及对主存的写操作,更新策略比较容易实现
- 写回法 (Write back)

写操作时只把数据写入 Cache 而不写入主存 当 Cache 数据被替换出去时才写回主存 写操作时间就是访问 Cache 的时间, 读操作 Cache 失效发生数据替换时, 被替换的块需写回主存,增加了 Cache 的复杂性

- (1) 增加 Cache 的级数 片载(片内) Cache 片外 Cache
- (2) 统一缓存和分立缓存 指令 Cache 数据 Cache

与主存结构有关

与指令执行的控制方式有关 是否流水

Pentium

8K 指令 Cache 8K 数据 Cache

PowerPC620 32K 指令 Cache 32K 数据 Cache

例题

例:求信息码01101110的海明码,使之能纠正1位错误,并画出海明校验逻辑电路。

解: <1> 确定校验位位数:

设r为校验位位数,则码字的位数应满足不等式:

$$2^{r}-1 \ge k+r$$

设r=3,则 $2^3-1=7$, k+r=8+3=11,不满足设r=4,则 $2^4-1=15$, k+r=8+4=12,满足所以,r最小取4。

例 题 (续)

例:求信息码01101110的海明码,使之能纠正1位错误,并画出海明校验逻辑电路。

<2> 确定校验位位置:校验位C_i位于位号为2ⁱ⁻¹的位置 因此, C₁、C₂、C₃、C₄位于2⁰、2¹、2²、2³的位置

1 2 3 4 5 6 7 8 9 10 11 12 C_1 C_2 b_7 C_3 b_6 b_5 b_4 C_4 b_3 b_2 b_1 b_0

<3>分组:有4个校验位,将12位分成4组, 被校验位的位号=校验位位号之和。

4-7-3-	1	2	3	4	5	6	7	8	9	10	11	12
	C1	C2	b7	C3	b6	b5	b4	C4	b3	b2	b1	b0
			0		1	1	0		1	1	1	0
第一组(c1)1	1		1		V		V		V		V	
第二组(c2)2		V	V			1	V			V	1	
第三组(c3)4				1	V	1	V					1
第四组(c4)8								V	V	1	V	V

例 题 (续)

	1	2	3	4	5	6	7	8	9	10	11	12
170	C1	C2	b7	C3	b6	b5	b4	C4	b3	b2	b1	b0
	1	1	0	0	1	1	0	1	1	1	1	0
第一组(c1)1	1		V		1		V		V		1	
第二组(c2)2		1	V			1	V		- 7	√	1	
第三组(c3)4				1	1	1	1					1
第四组(c4)8								1	1	1	1	√

<4>校验位的形成:

 C_1 =第一组中所有位(除 C_1 外)求异或= b_7 ⊕ b_6 ⊕ b_4 ⊕ b_3 ⊕ b_1 =1 C_2 =第二组中所有位(除 C_2 外)求异或= b_7 ⊕ b_5 ⊕ b_4 ⊕ b_2 ⊕ b_1 =1 C_3 =第三组中所有位(除 C_3 外)求异或= b_6 ⊕ b_5 ⊕ b_4 ⊕ b_0 =0 C_4 =第四组中所有位(除 C_4 外)求异或= b_3 ⊕ b_2 ⊕ b_1 ⊕ b_0 =1 \therefore 海明码为110011011110

<5> 校验原理

在数据输出存储器时分别求P₁,P₂,P₃,P₄

 $P_1 = C_1 \oplus 第一组中所有位求异或=C_1 \oplus b_7 \oplus b_6 \oplus b_4 \oplus b_3 \oplus b_1$

 $P_2=C_2$ 中第二组中所有位求异或= C_2 中 b_7 中 b_5 中 b_4 中 b_2 中 b_1

 $P_3=C_3$ 田第三组中所有位求异或= C_3 田 b_6 田 b_5 田 b_4 田 b_0

 $P_4 = C_4 \oplus$ 第四组中所有位求异或= $C_4 \oplus b_3 \oplus b_2 \oplus b_1 \oplus b_0$

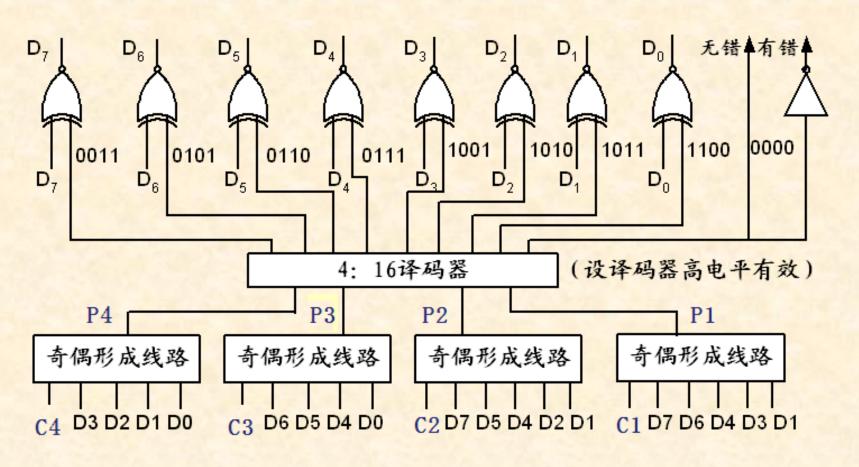
当P₄P₃P₂P₁=0000时,则从存储器中取出的数据无错。

否则,P₄P₃P₂P₁的二进制编码即为出错位号。

例如: $P_4P_3P_2P_1=1001$, 1 2 3 4 5 6 7 8 9 10 11 12 C_1 C_2 b_7 C_3 b_6 b_5 b_4 C_4 b_3 b_2 b_1 b_0

说明第9位出错,即b3错,将其取反,即可纠错。

• 纠正一位出错位的海明校验逻辑电路



例 题 (续)

• 进一步思考,本题改为求能够检测两位错误并纠正一位错误的海明校验逻辑电路。

<1>确定校验位位数:

设r为校验位位数,则码字的位数应满足不等式:

$$2^{r-1} \ge k+r$$

设r=4,则 $2^{4-1}=8$, k+r=8+4=12,不满足设r=5,则 $2^{5-1}=16$,k+r=8+5=13,满足所以,r最小取5。

- 进一步思考,本题改为求能够检测两位错误并 纠正一位错误的海明校验逻辑电路。
 - <2> 确定校验位位置:校验位C_i位于位号为2ⁱ⁻¹的位置 因此, C₁、C₂、C₃、C₄、C₅位于2⁰、2¹、2²、2³、 2⁴的位置, 但由于共有13个数, 所以最大位号为13

1												
C_1	C_2	b ₇	C_3	b_6	b ₅	b_4	C_4	b_3	b_2	b_1	b_0	C_5

<3>分组:有5个校验位,将13位分成5组,被校验位的位号=校验位位号之和,C5为总校验。

	1	2	3	4	5	6	7	8	9	10	11	12	13
	C1	C2	b 7	C3	b6	b5	b4	C4	b3	b2	b1	b0	C5
			0		1	1	0		1	1	1	0	
第一组(c1)1	1		1		1		1		1		1		
第二组(c2)2		1	1		Kain.	√	1			V	1		
第三组(c3)4		7		1	V	V	1					1	
第四组(c4)8								1	1	V	1	1	
第五组(c5)13	1	V	1	1	1	1	1	1	1	1	1	1	1

	1	2	3	4	5	6	7	8	9	10	11	12	13
	C1	C2	b7	C3	b6	b5	b4	C4	b 3	b2	b1	b0	C5
	1	1	0	0	1	1	0	1	1	1	1	0	0
第一组(c1)1	√		1		1		1		1		1		
第二组(c2)2		1	V			1	√			1	√		
第三组(c3)4	4			1	1	1	1	7				√	
第四组(c4)8								√	√	1	1	V	
第五组(c5)13	V	√	V	1	1	1	1	1	√	V	1	1	1

<4> 校验位的形成:

 C_1 =第一组中所有位(除 C_1 外)求异或= b_7 ⊕ b_6 ⊕ b_4 ⊕ b_3 ⊕ b_1 =1

 C_2 =第二组中所有位(除 C_2 外)求异或= b_7 ⊕ b_5 ⊕ b_4 ⊕ b_2 ⊕ b_1 =1

 C_3 =第三组中所有位(除 C_3 外)求异或= $b_6 \oplus b_5 \oplus b_4 \oplus b_0 = 0$

 C_4 =第四组中所有位(除 C_4 外)求异或= $b_3 \oplus b_2 \oplus b_1 \oplus b_0 = 1$

 C_5 =第五组中所有位(除 C_5 外)求异或= $C_1 \oplus C_2 \oplus b_7 \oplus ... \oplus b_0 = 0$

:. 海明码为1100110111100

<5> 校验原理

在存储字读出时分别求P₁,P₂,P₃,P₄,P₅

 $P_1 = C_1 \oplus 第一组中所有位求异或=C_1 \oplus b_7 \oplus b_6 \oplus b_4 \oplus b_3 \oplus b_1$

 $P_2=C_2$ 中第二组中所有位求异或= C_2 中 b_7 中 b_5 中 b_4 中 b_2 中 b_1

 $P_3 = C_3 \oplus 第三组中所有位求异或=C_3 \oplus b_6 \oplus b_5 \oplus b_4 \oplus b_0$

 $P_4 = C_4 \oplus 第四组中所有位求异或= C_4 \oplus b_3 \oplus b_2 \oplus b_1 \oplus b_0$

 $P_5 = C_5 \oplus$ 第五组中所有位求异或= $C_1 \oplus ... \oplus C_5 \oplus b_7 \oplus ...b_0$

P5=0,则无错或偶数个错

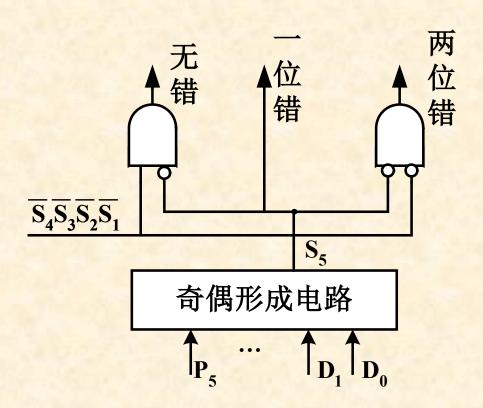
 P₄P₃P₂P₁=0000,
 无错

 P₄~P₁不全为零,两位错

 $P_5=1$,则有奇数个错

P₄~P₁中只有1个不为0,C_i≠0, 则C_i出错
 P₄P₃P₂P₁=0000, 则C₅出错
 P₄~P₁两位以上不为零,则数据位出错且 P₄P₃P₂P₁的二进制代码即出错位号

• 检测两位错误并自动纠正一位错误的附加电路



循环冗余码

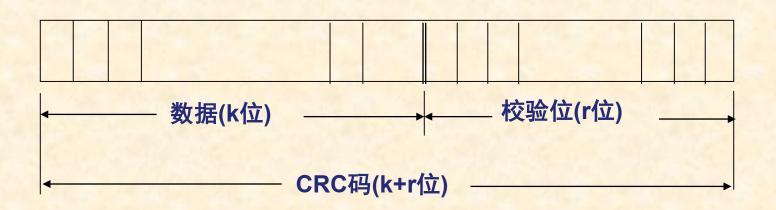
循环冗余校验码(Cyclic Redundancy Check),简称CRC码

- 具很强的检错、纠错能力。
- 用于大批量数据存储和传送中的数据校验。是目前磁表面存储器中应用最广泛的一种校验方法,也是多机通信中常用的校验方法
- 可以发现并纠正信息串行读写、存储或传送过程中出现的一位或多位错误。
- 为什么大批量数据不用奇偶校验?
 - 在每个字符后增加一位校验位会增加大量的额外开销;尤其在网络通信中,对传输的二进制比特流没有必要再分解成一个个字符,因而无法采用奇偶校验码。
- 通过某种数学运算来建立数据和校验位之间的约定关系。奇偶校验码和海明校验码都是以奇偶检测为手段的。

CRC码的编码方法

基本思想:

数据信息M(x)为一个K位的二进制数据,将M(x)左移r位后,用一个约定的"生成多项式"G(x)相除,G(x)是一个r+1位的二进制数,相除后得到的r位余数就是校验位。校验位拼接到M(x)后,形成一个K+r位的代码,称该代码为循环冗余校验(CRC)码,也称(k+r,k)码。



CRC码的编码方法

- 从k位信息位得到r位校验位的方法:
 - (1) 将待编码k位有效信息写成多项式M(x)

$$M(x)=C_{k-1}x^{k-1}+C_{k-2}x^{k-2}+...+C_1x+C_0$$

- (2) 将M(x)左移r位,得到M(x)·x^r。目的是空出r位,以便拼接r位校验位
- (3) 选取一个r+1位的生成多项式G(x)。对 $M(x) \cdot x^r$ 做模2除运算

$$\frac{M(x) \times x^r}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

(要产生r位余数, 所以除数应为r+1位)

(4) 将左移r位的待编码信息与余数R(x)做模2加,得到CRC码

$$M(x) \cdot x^r + R(x) = Q(x) \cdot G(x)$$

说明CRC码能被生成多项式G(x)整除(此结论是CRC译码纠错的依据) 以上说明CRC码能被生成多项式整除,这是译码与纠错得依据

CRC码的校验方法

一个CRC码一定能被生成多项式整除,当数据和校验位一起送到接受端后,只要将接受到的数据和校验位用约定好的同样的生成多项式除,如果能除尽,表明没有发生错误;若除不尽,则表明某些数据位发生了错误,余数将指明出错位所在的位置。

CRC码计算

例:信息码为1100,生成多项式G(x)=x3+x+1,求CRC码。

解:
$$M(x)=x^3+x^2$$
 $G(x)=1011$ $r=3$ $M(x) \cdot x^3=1100000$

$$\frac{M(x) \cdot x^3}{G(x)} = \frac{1100000}{1011} = 1110 + \frac{010}{1011}$$

CRC码为1100 010

循环冗余码举例

$$X^{3}\cdot M(x) \div G(x) = (x^{8} + x^{4} + x^{3}) \div (x^{3} + 1)$$

	100111
	100111
1001	100011000 1001
	1001
	0011
	0000
	0111
	0000
	1110
	1001
	1110
	1001
	1110
	1001
	111

(模2运算不考虑加法进位和减法借位,上商的原则是当部分余数首位是1时商取1,反之商取0。然后按模2相减原则求得最高位后面几位的余数。这样当被除数逐步除完时,最后的余数位数比除数少一位。这样得到的余数就是校验位,此例中最终的余数有3位。)

校验位为111, CRC码为100011 111。如果要校验CRC码,可将CRC码用同一个多项式相除,若余数为0,则说明无错;否则说明有错。例如,若在接收方的CRC码也为100011 111时,用同一个多项式相除后余数为0。若接收方CRC码不为100011 111时,余数则不为0。

表 4.5 对应 G(x)=1011 的 (7, 4) 循环的出错模式

序号	N _J	N ₂	N_3	N ₄	N ₅	N ₆	N ₇	余数	出错位
正确	1	1	0	0	0	1	0	000	无
	1	1	0	0	0	1	1	001	7
	1	1	0	0	0	0	0	010	6
错	1	1	0	0	1	1	0	100	5
	l	1	0	1	0	1	0	011	4
误	1	1	1	0	0	1	0	110	3
ĺ	1	0	0	0	0	1	0	111	2
	0	1	0	0	0	1	0	101	1

Thank You!

