

接回拨术。

第5章 存储器技术

主编: 王让定 朱莹

宁波大学信息学院









本章主要内容录



存储器概述



半导体随机 存取存储器



高速缓冲 存储技术



虚拟存储 技术



存储器概述

PART 01



存储器的发展及分类

1. 存储系统的发展

- 50年代中期,主要使用磁芯存储器作为主存。
- 60年代中期以后,半导体存储器已取代磁芯存储器
- 1968年, IBM-360/85最早采用了高速缓冲——主存储器的存储层次。
- 1970年,美国RCA公司研究成功虚拟存储器系统。
- IBM公司于1972年在IBM370系统上全面采用了虚拟存储技术。

- 2. 存储器分类
- (1) 按功能分类
- 1) 主存储器
- 2) 外部存储器
- 3) 高速缓冲存储器

(2) 按存取方式分类

- 1) 随机存取存储器RAM(Random Access Memory)
- 2) 只读存储器ROM (Read Only Memory)
- 3) 顺序存取存储器SAM(Sequential Access Memory)
- 4) 直接存取存储器DAM(Direct Access Memory)

(3) 按存储介质分类

- 1) 磁芯存储器
- 2) 半导体存储器
- 3) 磁表面存储器
- 4) 光存储器

(4) 按信息的可保存性分类

断电后信息即消失的存储器,称为非永久记忆的存储器。断电后仍能保存信息的存储器,称为永久性记忆的存储器。磁性材料做成的存储器是永久性存储器,半导体读写存储器RAM是非永久性存储器。

(5) 按串、并行存取方式分类

目前使用的半导体存储器大多为并行存取方式,但也有以串行存取方式工作的存储器,如电耦合器件(CCD)、串行移位寄存器和镍延迟线构成的存储器等。



存储器的主要技术指标

- 1. 速度
- 2. 存储容量
- 3. 存储带宽
- 4. 存储器的可靠性

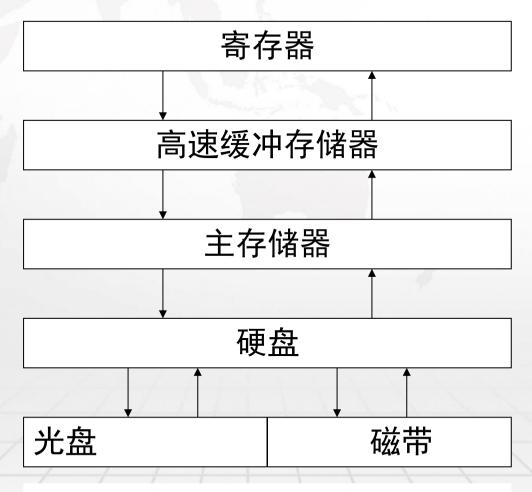


存储器的基本结构框架

	存储	单元	
地址码	单元位数(bits)		
0	1		m
1	1	···. ···	m
J-2			
J-1	1	\\\	m
/ J/ /	1		m



存储系统的层次结构



存储器层次结构图

实际上,存储器的层次结构主要体现在缓存──主存和主存──辅存这两个存储层次上,如图6.3所示。

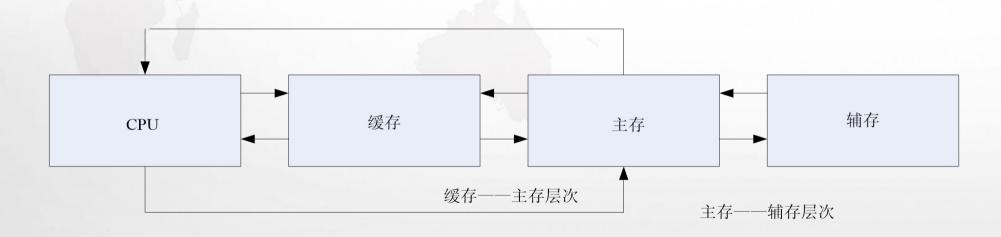


图6.3 缓存——主存层次和主存——辅存层次



半导体随机存取存 储器

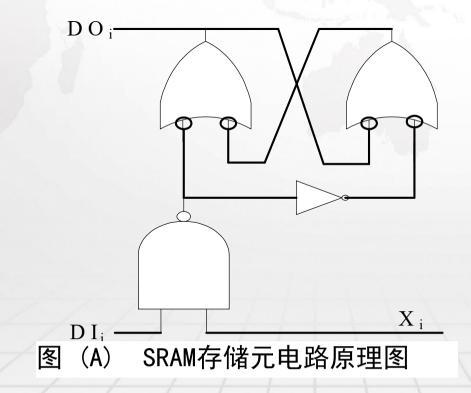
PART 02

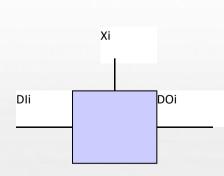
半导体随机存取存储器是构成计算机主存储器、高速 缓冲存储器的核心部件。根据信息的存储机理,可分 为静态随机存取存储器(SRAM)和动态随机存取存储 器(DRAM)。 SRAM的优点是速度非常快,只要电源 不切断,内容就不会消失,一般高速缓冲存储器 (Cache memory) 用SRAM组成。DRAM的特点是成本低, 集成度很高,存储容量比SRAM大很多,通常用作计算 机的主存储器。



SRAM存储器

1. 基本存储元及存储阵列





图(B) 逻辑结构原理框图

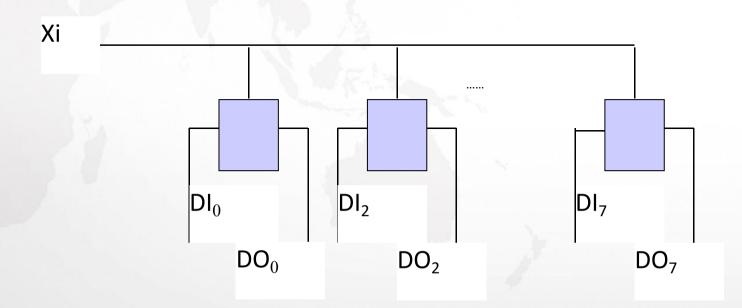
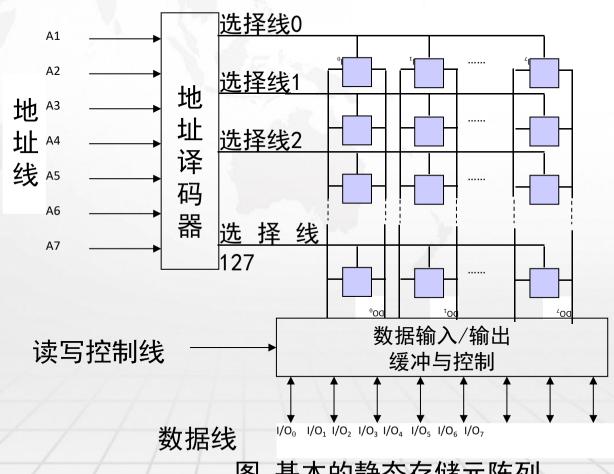


图 SRAM存储单结构框图

存储器芯片的逻辑结构



基本的静态存储元阵列

- SRAM存储器芯片的容量(capacity) = Z×B bits, 其中:
- Z表示该芯片有多少个存储单元。一般Z=2ⁿ, n为正整数, 其数值n表示该存储 器芯片有多少条(即n条)地址线, 地址线通常用An-1An-2···A1A0表示;
- B表示该芯片某存储单元的位数,即数据宽度,B的取值一般为1,2,4,8,…,数据宽度常用I/0B-1I/0B-2…I/01I/00或DB-1DB-2…D1D0表示。
- Z×B bits 又被称为存储器的组织方式(Organization)
- 为了便于组织更大容量的SRAM存储器,其芯片都采用双译码形式,即采用x向和y向的两级译码结构。我们可将x向译码称行译码,y向译码称列译码。

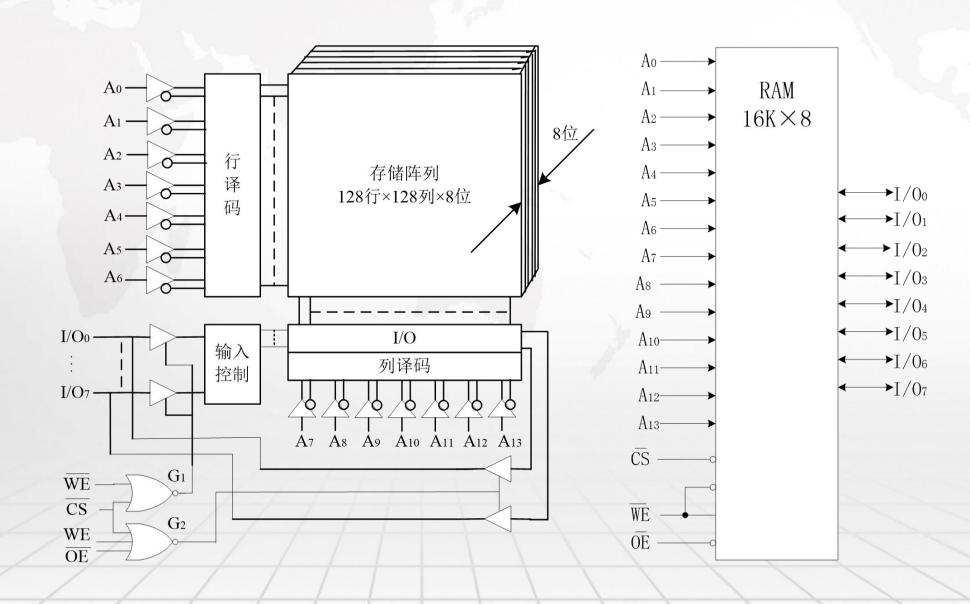
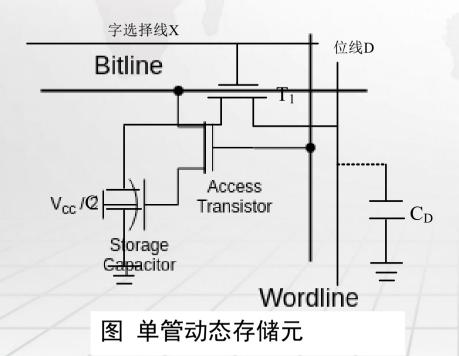


图 16K×8位SRAM结构图(A)和逻辑图(B)



1. 基本存储元

当字选线(或行线)为高电平时,该存储元被选中。



✓写入时,若写入"1",位线D为高电平,对电容C进行充电;写入"0",位线D为低电平,电容C上的电荷经位线进行泄放。

✓读出时,若原存为"1",C上有电荷,经T₁管在位线上产生读电流,完成读"1"操作;若原存为"0",C上无电荷,在位线上不产生读电流,完成读"0"操作。

注:信息是存储在一个很小的电容C上, 也只能保留2ms左右的时间,所以必须 要定时地进行刷新(或再生)操作。

2. DRAM芯片的逻辑结构

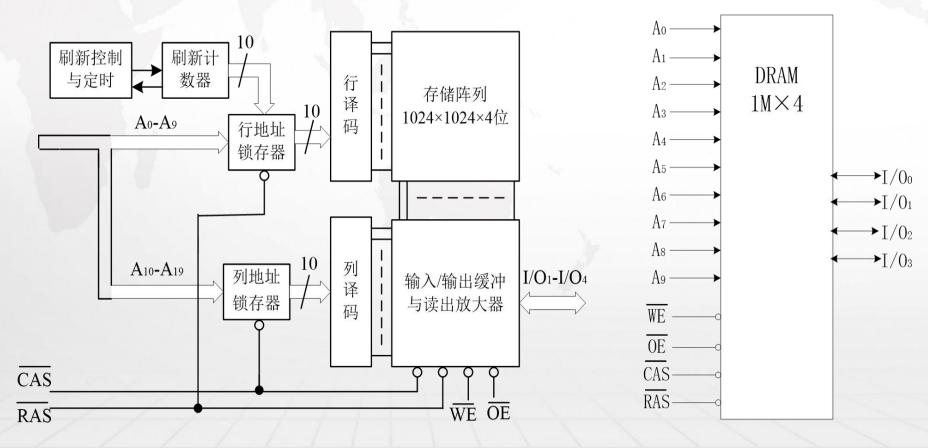
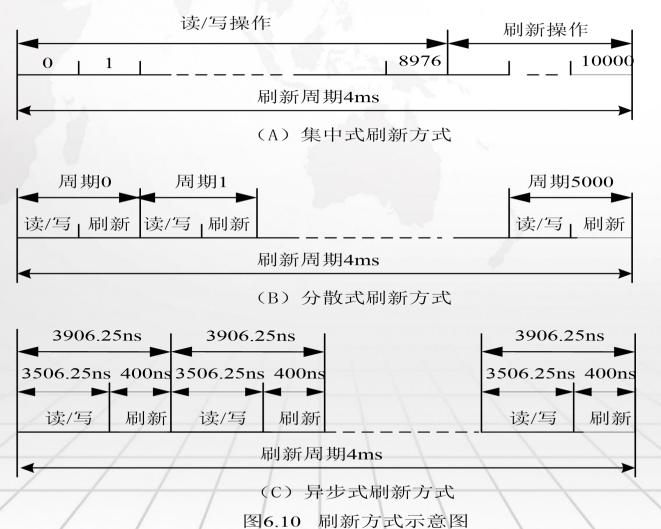


图 1M×4位DRAM结构图(A)和逻辑图(B)

3. DRAM的刷新

常见的刷新操作有集中式、分散式和异步式三种。



只读存储器ROM

1. 掩膜式只读存储器MROM(MASK ROM)

掩膜式只读存储器MROM的存储内容固定,是由生产厂家规模 化生产的产品。

2. 可编程只读存储器

可编程ROM有PROM、EPROM和E2PROM三种。

3. 闪速存储器(Flash Memory)

高密度、非易失的读写存储器,高密度意味着它具有很大的 存储容量。



FPM DRAM

CDRAM

SDRAM

RDRAM

DDR SDRAM等新型存储器芯片。



主存容量的扩展

- 使用多个芯片组成为存储器的技术称为存储器的扩展技术。
- 如果只是扩展存储器存储元的个数,即扩展每个单元的数据位数,称为"位扩展";
- 如果只是扩展存储器存储单元的个数,称为"字扩展";
- 如果两者都要扩展,称为"字位扩展"。
- 主存储器同CPU连接时,要完成地址线、数据线和控制线的连接,还要涉及芯片间的片选译码等。

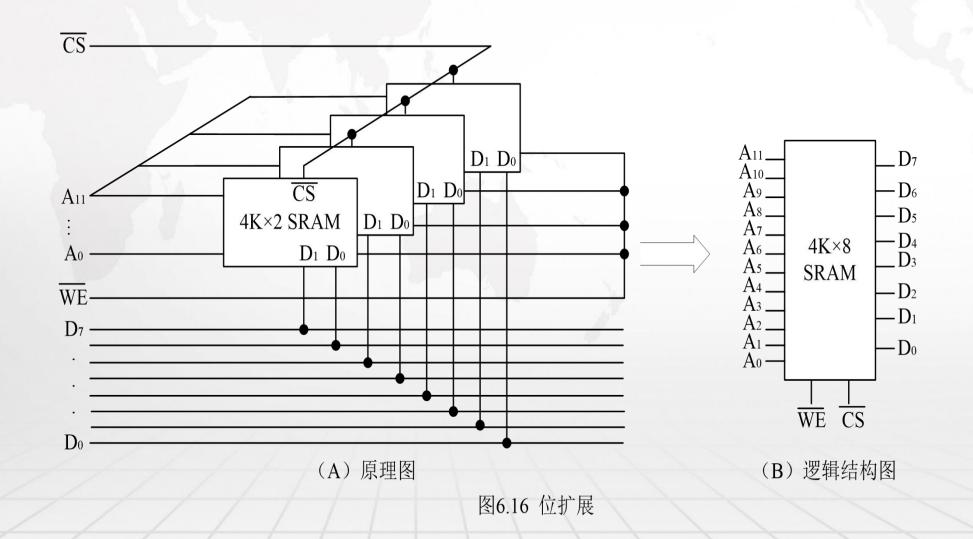
1. 位扩展

- 当所选用的存储芯片的每个单元的数据位不能满足存储器所需的位数时,就要进行位扩展。
- 位扩展的连接方式是将所有芯片的地址线、片选线、读/写线对应连接,数据线分别引出。

例:用 $4K \times 2$ 位的SRAM存储器芯片构成 $4K \times 8$ 位的存储器,所需芯片数为($4K \times 8$)/($4K \times 2$)=4(片)。

由于 $4K \times 2$ 位的芯片有12条地址线 $A_{11}^{}A_0$ 、2条数据线 $D_1^{}D_0$ 、1个 片选信号 和1个读/写控制信号 ,因此由 $4K \times 2$ 位的SRAM存储器芯片构成的 $4K \times 8$ 位存储器有12条地址线 $A_{11}^{}A_0$ 、8条数据线 $D_7^{}D_0$ 、1个片选信号 和1个读/写控制信号 。

连接方式如图6.16所示。经过位扩展后,我们可以把经图(A)扩展后的存储器等效为一个4K×8位的存储器模块,如图6.16(B)所示。



2. 字扩展

- 字扩展就是进行存储器容量的扩展,在进行位扩展时,存储器容量未增加,仅仅是扩展了一个存储单元的数据位。
 从另一个角度讲,即在位扩展后,存储器的地址线未增加,只增加了数据线。
- 字扩展的连接方式是将所有芯片的地址线、数据线、读/ 写线对应连接,每个芯片的片选线要用译码器将高位地址 译码后分别连接。

例:用 $4K \times 8$ 位的SRAM存储器芯片构成 $16K \times 8$ 位的存储器,所需芯片数为($16K \times 8$)/($4K \times 8$)=4(片)。

从所要设计的 $16K \times 8$ 位的存储器分析,它有14条地址线 $A_{13}^{}A_{0}$ 、8条数据线 $D_{7}^{}D_{0}$,而我们现在所用的 $4K \times 8$ 位的 SRAM存储器芯片的地址线是12条,为了构成 $16K \times 8$ 位的存储器,需要4个 $4K \times 8$ 位的SRAM存储器芯片。

在进行字扩展时,用12条地址线A₁₁~A₀分别与每个4K×8位的SRAM存储器芯片连接,用两条高位地址线A₁₃、A₁₂经过一个2-4译码器产生的译码信号分别与每个4K×8位的SRAM存储器芯片的片选信号连接,如图6.17所示。

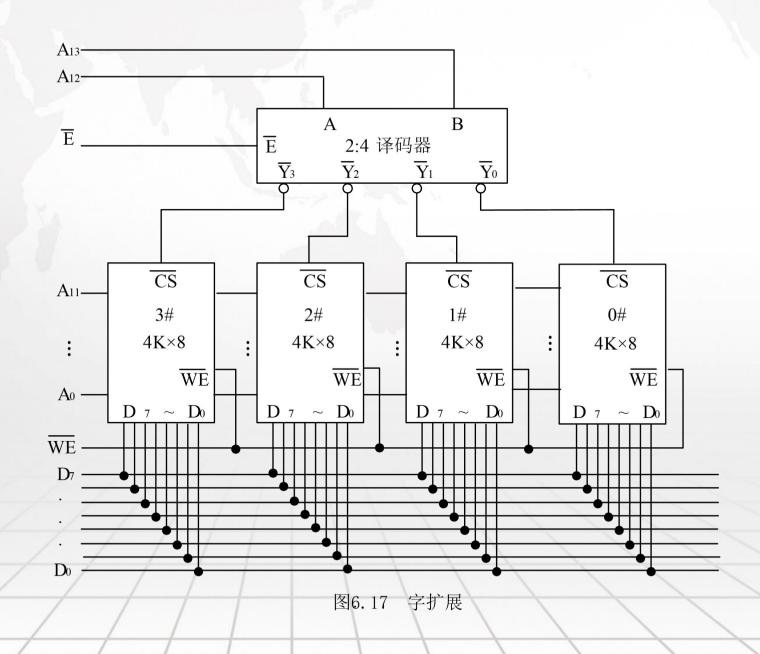


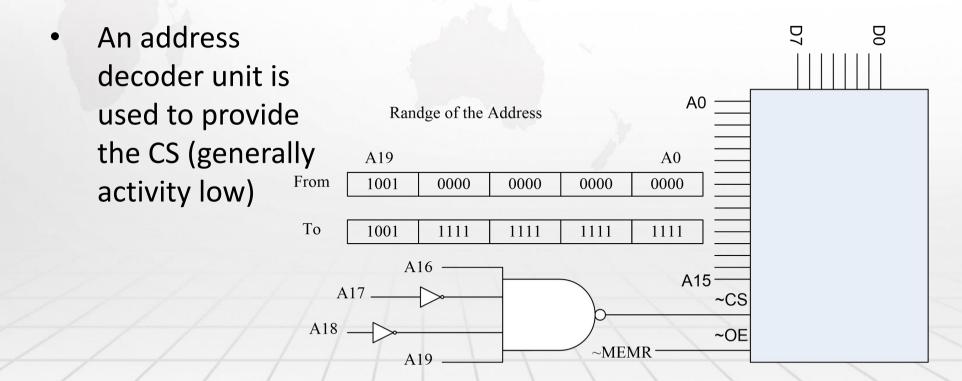
表6.2 地址空间分配

芯片或 模块号	译码器地址 A 13 A 12	芯片自身地址空间 A11~A0	芯片在所扩展的存储器 中的地址空间
0	00	0000 0000 0000 : 1111 1111 1111	0000H : 0FFFH
1	01	0000 0000 0000 : 1111 1111 1111	1000H : 1FFFH
2	10	0000 0000 0000 : 1111 1111 1111	2000H : 2FFFH
3	11/	0000 0000 0000 : 1111 1111 1111	3000H : 3FFFH

3. 字位扩展

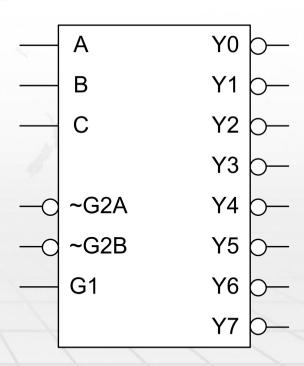
当构成一个容量较大的存储器时,往往需要在字数(容量) 方面和位数方面同时进行扩展,即位扩展和字扩展的结 合,这种扩展方式称为字位扩展。

Section 10.2 Memory Address Decoding



Section 10.2 Memory Address Decoding

- Using the 74LS138 as decoder
 - 74LS138 is a 3-to-8 decoder
 - ABC are address inputs
 - G1, G2A, G2B is control pin, and can be linked with both address bus and control bus



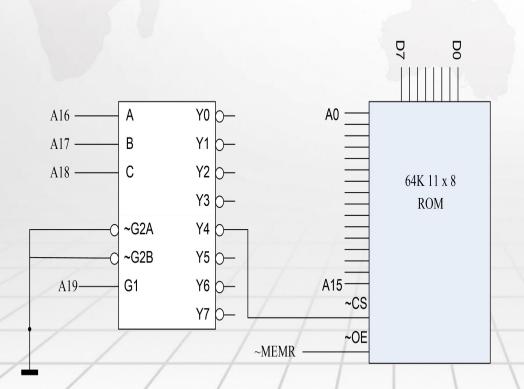
Section10.2 Memory Address Decoding

74LS138 Function Table

Inputs		
Enable	Select	Outputs
G1 G2	СВА	Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7
ХН	XXX	ннннннн
L X	XXX	ннннннн
H L	LLL	LHHHHHHH
H L	LLH	нцннннн
H L	LHL	H H L H H H H
H L	LHH	ннн L н н н н
H L	HLL	нннн L н н н
H L	HLH	нннньнн
H L	HHL	ннннньн
H/L	ННН	ннннннь

Memory Address Decoding

Calculate the address below: (Figure 10-12)

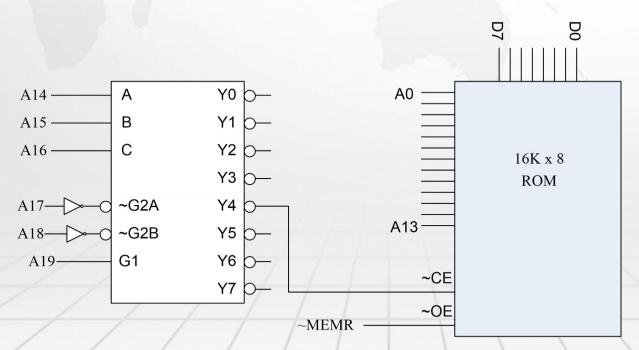


Solution:

- Enable 74LS138 Encoder
 - A19 = 1
- Enable the output pin
 - A18A17A16=100
- Address range
 - A15~A0 ranges from all low to all high. And that would be 2¹⁶ (=65536) bytes

Section 10.2 Memory Address Decoding

Calculate the address below: (Example 10-7)



Solution:

- 1) Enable 138 A19~A17=111B
- 2) Enable Y4A16~A14=100B
- 1) Calculate Range
 A13~A0=All low to All
 high
 That is 16K bytes

Memory Address Decoding

- How to calculate address ranges of memory?
 - 1. write down all the address bus pins involved;
 - 2. assign values to the AB pins so that to activate the ~CS pin of the memory IC (truth table of the decoder circuit may be required);
 - 3. assign '0' to the AB pins of the memory IC to get the lower range; assign '1' to them to get the upper range.

Address Decoding by Fill in an Address Table

1. Draw a table with three major columns: Column I for **Chip Inside Address** (Address Busses connected with chip address bus pins); column II for **Decoding Address** (Address Busses connected with 74LS138 ABC input pins); column II for **Decoder Active Address** (Address Busses connected with 74LS138 chip enable pins: G1 and G2A ,G2B)

Address Table in step I

74LS138 Active	74LS138 input	ROM 64Kx8	The state of the s
~G2A ~G2B G1	B C A	A15~A0)
Decoder Active Address	Decoding Address	Chip Inside Address	9 1
		The second secon	

2. Fill in address bus pin's numbers into these columns. The numbers in the table must be consecutive, for example: A1, A2, A3(even not been used in Address Decoding), A4, A5,…

Address Table in step II

74LS	138 Act	ive	74L	S138 ir	nput	ROM 64Kx8															
~G2A	~G2B	G1	С	В	Α	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
GND	GND	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
													11 1/2								
																					75.

3. Fill in the table to Active 74LS138

Address Table in step III

74LS	138 Act	ive	74L	S138 ir	nput	-	ROM 64Kx8										_				
~G2A	~G2B	G1	С	В	Α	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
GND	GND	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	АЗ	A2	A1	A0
0	0	1		/	/	/				1											
		/																			

Address Decoding by Fill in an Address Table

4. Fill in the table to enable 74LS138 Y4 output

Address Table in step IV

74LS	138 Act	ive	74L	S138 ir	put		1	40					ROM (64Kx8				1	47	200	
~G2A	~G2B	G1	С	В	Α	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
GND	GND	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
0	0	1	1	0	0	Ú.		11/2 1	Per											A .	
				3					at By		F 1										

5. Fill in the table to find internal address upper range and lower range

Address Table in step V

74L9	6138 Act	ive	74L	S138 ir	put	ROM 64Kx8															
~G2A	~G2B	G1	С	В	Α	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
GND	GND	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A 5	A4	А3	A2	A1	A0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

6. Append another column to fill in the result

Address Table in step V

74LS	6138 Act	ive	74L	S138 i	nput		1 1	1		1 1		/	ROM 6	64Kx8	/						
~G2A	~G2B	G1	С	В	Α	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	А3	A2	A1	A0
GND	GND	<mark>A19</mark>	A18	<mark>A17</mark>	<mark>A16</mark>	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	1	0	0	0	0/	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	/1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Memory Address Decoding

- What is the relationship between your instruction, the Address decoder, and the memory?
 - 1. When the MOV instruction is executed, BIU will calculate the physical address and put them onto Address Bus, then enable ALE control signal. (timing diagram of Read from Memory)
 - 2. ALE will force the address latcher 74LS373 to latch address signals;

Memory Address Decoding

- 3.lower part of the address signal will be send to the target memory chip; higher part will be received by the address decoder. Then, chip selection signal CS will be released by the decoder.
- 4. when CS is received by the target memory chip, the lower part address signal received before will be decoded inside the chip to select the location your instruction identified. (timing diagram of memory read timing for SRAM).

例 CPU具有16条地址线(A₁₅~A₀),16条双向数据线(D₁₅~D₀),控制总线中与主存有关的信号有(允许访存,低电平有效),(读写控制,低电平为写,高电平为读)。主存按字编址,其地址空间分配如下:0~1FFFH为系统程序区,由EPROM芯片组成,从2000H起共24K地址空间为用户程序区,最后(最大)4K地址空间为系统程序工作区。现有如下芯片:

EPROM: 4K×8位(仅有一个端),8K×8位。

SRAM: 16K×1位, 2K×8位, 4K×8位, 8K×8位。

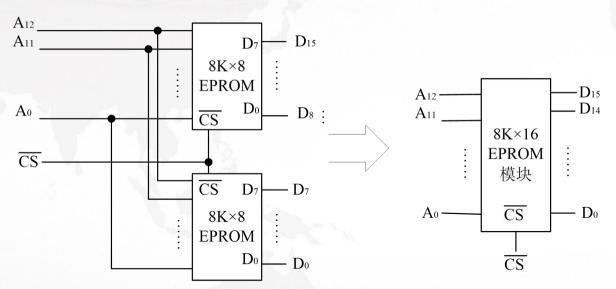
- (1) 请选择适当的芯片,按要求设计主存储器;
- (2) 可选用3:8译码器,画出主存储器与总线逻辑连接图。

解:

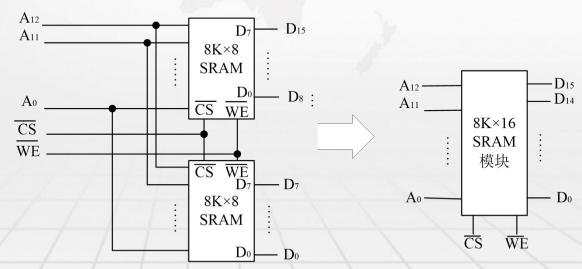
1)根据题意,主存空间为0000H[~]0FFFH,共64KB,其空间分配及各存储空间设计时可选用的存储芯片如下表:

地址空间	空间功能	可选用的存储类型
0000H~1FFFH	系统程序区	EPROM
2000H~7FFFH	用户程序区	SRAM
8000H~7FFFH	保留	
OFOOOH~OFFFFH	系统程序工作	SRAM
	X	

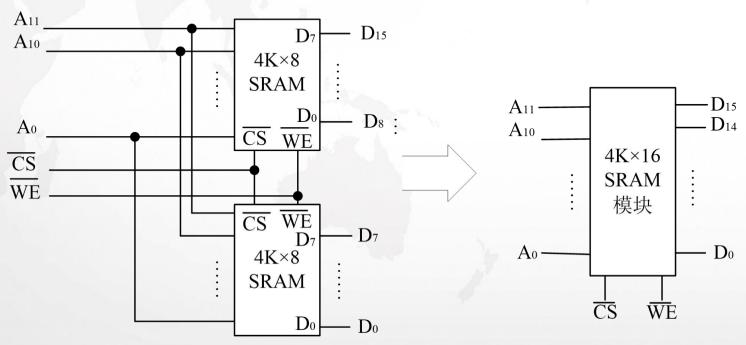
- 2)在各存储空间的设计中,首先根据所选用芯片进行位扩展,即2片8K×8位EPROM芯片扩展成8K×16位EPROM模块;2片8K×8位SRAM芯片扩展成8K×16位的SRAM模块;2片4K×8位SRAM芯片扩展成4K×16位的SRAM模块。各模块的位扩展图如图6.18所示。



(A) 位扩展后的8K×16位EPROM逻辑连接及模块



(B) 位扩展后的8K×16位SRAM逻辑连接及模块



(C) 位扩展后的4K×16位SRAM逻辑连接及模块 图6.18 位扩展逻辑及扩展模块

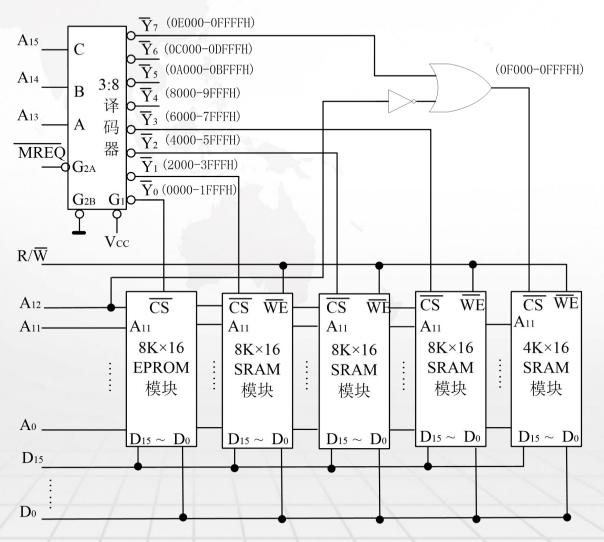


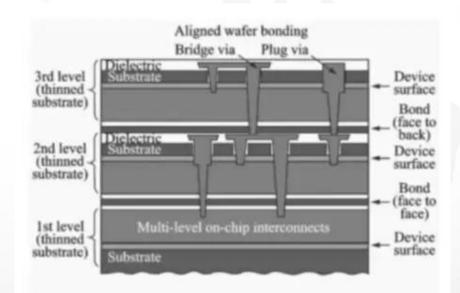
图6.19 主存储器扩展与总线逻辑连接

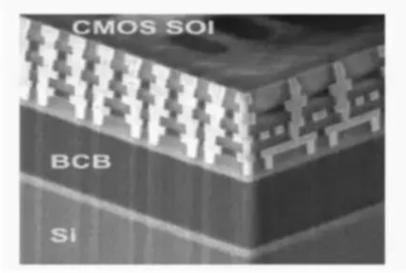


存储器封装技术发展——芯片堆叠技术

1. 什么是芯片堆叠

- 上世纪初,BGA(球栅阵列封装)代替了外引脚封装,以2D平面布局替代传统边界引线方式,扩展芯片1/0引出点数量,缩短引出信号传递路径长度;但时至今日,BGA封装也遇到了天花板;
- 一种芯片封装的发展方向是将多芯片(芯片核心)之间的2D平面布线连接改为3D立体直连,片与片之间层叠布置,依靠TSV(硅通孔,Through Silicon Via)布置金属连接线,构建片间连接;
- 优点: 封装密度极大增强, 片间传输通道带宽增加, 距离变短;
- 可以通过混合封装实现SOC的3D增强版,做到单颗粒主机形态;





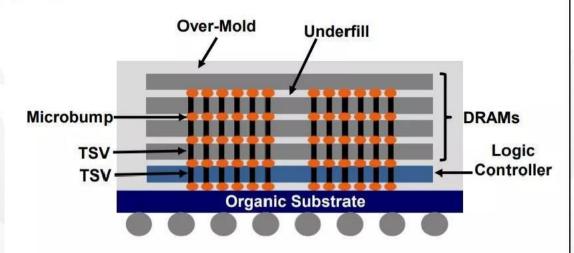


存储器封装技术发展——芯片堆叠技术

2. 航天科技集团公司九院第771研究所

基于TSV技术的产品具有全局互 连短、延迟小、灵活性高、高密 度集成等优点,是先进封装的核 心技术, 也是实现三维集成的最 优途径。近期, 我所通过集智攻 关、成功实现了TSV硅转接板、 等壁增长TSV、超大面阵凸点三 种封装形式产品的研制及小批量 风险可控生产;

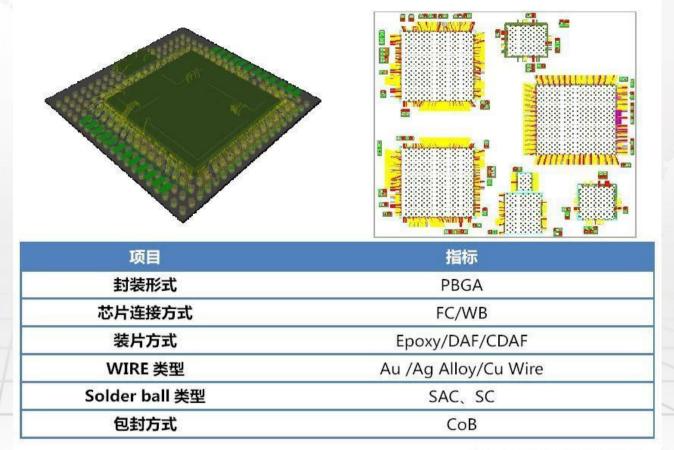
3D-IC



项目	指标	
晶圆尺寸	12″ /8″	
最小线宽	1µm	
最小孔径	3µm	
单层最小厚度	100μm	
植球尺寸	60-760μm	
深宽比	10:1	
最大层数	8层	

- (2021-9-20)我国首款宇航级存储控制器芯片Bifort正式发布。据介绍,该 芯片完全为航天而设计,适用于宇航任务中高可靠、大容量数据的存储应用, 为航天器智能信息系统提供解决方案。攻克了芯片堆叠架构中抗辐射加固技术的应用研究难题。

 System in Package
- 芯片堆叠技术的另一 个应用方向: 异构芯 片混合堆叠
- SOC 的 3维 实现方式: System in Package
- 难点在于模拟、数字 电路的混合堆叠





高速缓冲存储技术

PART 03

1. 程序的局部性

Cache就是利用程序的局部性原理,把程序中正在使用的部分存放在一个容量较小的缓冲存储器(Cache)中,使CPU的访问操作绝大部分时间针对Cache进行,从而使程序的执行速度大大提高。

2. Cache的工作原理

- CPU与Cache之间的数据交换以字为单位,而Cache与主存之间的数据交换以块为单位,一个块由若干个字或字节组成,大小相等,常将Cache的块称为Cache行。在一个时间段内,Cache的某行中存放着主存某块的全部信息,即Cache的某一行是主存某块的副本(或叫映像)。
- 当CPU访问某一存储单元(按字访问)的内容时,通过地址总 线向主存和Cache同时发出访问请求,若访问的内容在Cache 中,表示命中,此时,终止内存访问;若访问的内容不在 Cache中,表示未命中,此时继续访问内存,并将含有所访问 内存单元的相应内存块调入Cache的某行。

Cache的管理

1. Cache的映射方式

常用的Cache的地址映射有全相联方式、直接方式和组相联方式

- 全相联映射方式是主存中一个块可以映射到Cache中的任意一行
- 直接映射方式也是一种多对一的映射关系,与全相联映射方式 的区别在于一个主存块只能映射到cache的一个特定行。直接映 射方式的优点是硬件简单,成本低。缺点是每个主存块只有一 个固定的行位置可存放,容易产生冲突。
- 组相联映射方式是把主存按Cache的容量分区,主存中的各区和 Cache再按同样大小划分成相等的组,组内再划分为块。

2. 替换策略

- 选择替换策略的主要目的是获得较高的Cache命中率, 换句话说,要使得所访问的块不在Cache中的次数为 最小,使Cache中尽可能地保存最新数据。
- 目前常用的替换策略有最不经常使用(LFU)策略, 最近最少使用策略和随机替换策略。

- 1) 最不经常使用(LFU)策略
- 将一段时间内被访问次数最少的Cache行数据替换出去的 方法称为LFU算法。
- 2)最近最少使用(LRU)策略 LRU算法是将近期内长久未被访问的行替换出去。
- 3) 随机替换策略
- 随机替换策略就是随机地选取Cache的一行换出,这种策略硬件易于实现,且速度快于LFU和LRU算法,其缺点是可能换出的行随后马上又要使用,从而降低Cache命中率和工作效率。

2. 写操作策略

- 1) 写回法
- 2) 全写法
- 3)写一次法



虚拟存储技术

PART 04

32位微处理器存储器系统简介

80486充分重视了对多任务操作系统的支持性。主要体

现在两方面: 一是从硬件上为任务之间的切换提供了

良好的条件; 二是支持容量极大的虚拟存储器, 并且

为了管理如此大的存储空间,采用片内两级管理。

1. 虚拟存储技术

- 虚拟存储技术的最终体现是建立一个虚拟存储器。虚拟存储器是相对物理存储器而言的,物理存储器指由地址总线直接访问的存储空间,其地址称物理地址。
- 虚拟存储器机制由主存储器、辅助存储器和管理部件共同组建。通过管理软件,达到主存和辅存密切配合,使整个存储系统具有接近主存的速度和接近辅存的容量。这种技术不断改进完善,就形成了虚拟存储系统。
- 按照对主存的划分方式,虚拟存储器有段式虚拟存储器和 页式虚拟存储器两类。

2. 片内两级存储管理

- MMU和CPU做在同一个芯片中,并使MMU能管理大容量的虚拟 存储器。这种片内的MMU免去了通常片外MMU带来的种种延迟, 而且,程序员可以使用的存储空间即逻辑地址空间大大超过 物理地址空间,所以,极大地减少了存储空间出现的故障率, 减轻了操作系统的负担。
- 在两级存储管理中,段的大小可以选择,因此,可以随数据结构和代码模块的大小而确定,使用起来很灵活,另外,对每一段还可赋予属性和保护信息,从而可以有效地防止在多任务环境下各个模块对存储器的越权访问。

本 章 小 结

- 在介绍存储器基本概念(包括存储器的分类、存储器的分级 结构以及存储器的主要性能指标等)的基础上,以如何构建 存储器为主线,介绍了构成存储器的基本单位,即存储元。 由此介绍了如何构建存储单元的方法,以及由存储单元构成 存储阵列进而构成存储器的方法,紧接着介绍了存储器容量 扩展的方法。
- 对广泛使用SRAM和DRAM而言,存储器扩展的思路和方法是相同的,前者速度比后者快,但集成度不如后者高,另外,DRAM存储器需要刷新。

- 只读存储器和快闪存储器可在断电后也能保存原写入的数据。快闪存储器具有低功耗、高可靠性以及移动性等特点,是一种全新的存储器结构。
- Cache仅仅是一种高速缓冲存储器,是为提高计算机 系统性能所采用的一种硬件解决方案。主存与Cache 的地址映射有三种方案,其中组相联方式是全相联和 直接映射的折衷方案,得到了广发应用。
- 本章最后简单地介绍了80X86存储器的组织和虚拟存储管理技术。虚拟存储技术可大大提高存储系统的效率,增强系统的安全稳定性。