

第5章

存 储 器

无论简单还是复杂,每款计算机系统都有存储器。静态随机存储器存取速度快,容量小,造价高。动态随机存储器容量大,价格廉,但地址多路复用,还需要定时刷新。只读存储器保存永久驻留系统的程序,掉电后信息不丢失。闪存不仅掉电后信息不丢失,还可以在线擦除。不同种类的存储器构成微型计算机的存储系统。

5.1 存储器概述

计算机的存储器分为内存储器和外存储器两大类。任何程序和数据必须进驻内存储器后才能执行,因此,内存储器也称为主存储器。它比外存储器存取速度快,存储容量小。外存储器也叫辅助存储器,属于计算机的外部设备,如磁盘、光盘和 U 盘等,存储容量大,存取速度慢。

5.1.1 内存储器分类

内存储器主要由半导体材料构成,所以也称为半导体存储器。按照工作方式,内存储器可分为随机存储器(RAM)和只读存储器(ROM)两大类,如图 5-1 所示。随机存储器是一种易失性存储器(volatile memory),掉电后信息会丢失,常用来存放正在运行的程序和数据。只读存储器是一种非易失性存储器,信息一旦写入,就固定不变,掉电后信息也不会丢失。在使用过程中,只读存储器中的信息只能读出,一般不能修改,所以常用于保存固定不变且长期使用的程序和数据,如主板上的基本输入/输出系统程序 BIOS、外部设备的驱动程序等。

随机存储器分为静态随机存储器(Static RAM, SRAM)和动态随机存储器(Dynamic RAM, DRAM)。静态随机存储器以双稳态触发器为基本存储电路,保存的数据不需要刷新,只要通电就可以保存数据,所以常被称为静态存储器(static memory),与动态随机存储器比较,它的存取速度快、集成度低、功耗大。动态随机存储器以电容作为基本存储电路,但它存储的信息随电容上电荷的泄漏可能丢失,所以动态 RAM 必须定时刷新,它的存取速度相对较慢,但是集成度高、成本低。

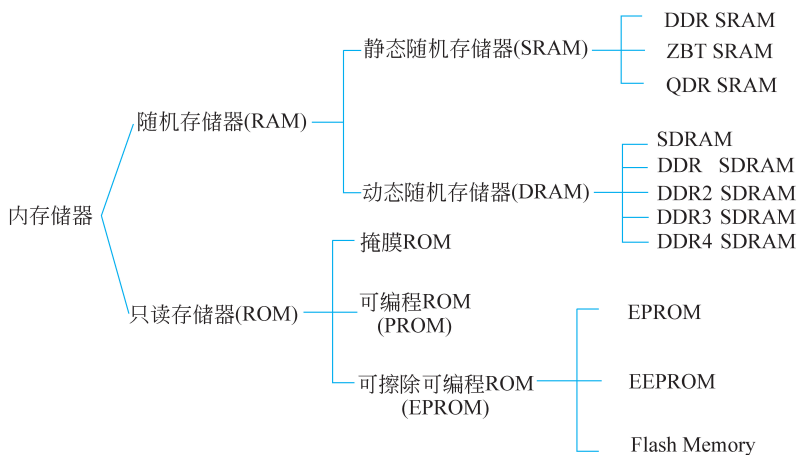


图 5-1 内存存储器分类

5.1.2 存储器件

计算机的存储器由一个或多个存储器件构成。存储器件通过计算机系统的地址总线、数据总线和控制总线与微处理器相连,实现程序和数据读/写操作。所以,所有的存储器件都有地址输入引脚、数据输出或者数据输入/输出引脚、控制读/写操作的控制引脚,还有从多片存储芯片中选定一个芯片的片选引脚,如图 5-2 所示。

1. 地址线

所有的存储器件都有地址输入引脚(地址线),用来接收 CPU 发过来的存储单元地址信息。地址线接收的地址信息用以选择存储芯片内部的存储单元。地址线都是从 A_0 开始标记,一直到 A_n 。地址线的根数与芯片内含有的存储单元个数有一个对应的关系。如含有 10 根地址线的芯片,地址线标记为 $A_0 \sim A_9$,存储单元应该有 $2^{10}=1024$ 个;含有 11 根地址线的芯片,地址线标记为 $A_0 \sim A_{10}$,存储容量为 $2^{11}=2048$ 个存储单元。反过来,一个存储容量为 4KB 的存储芯片,应该有 12 根地址线 $A_0 \sim A_{11}$;一个存储容量为 8KB 的存储芯片,应该有 13 根地址线 $A_0 \sim A_{12}$;一个存储容量为 1MB 的存储芯片,应该有 20 根地址线 $A_0 \sim A_{19}$ 。

如果存储芯片的地址线为 $A_0 \sim A_9$,则它的存储单元的地址范围为

$$00\ 0000\ 0000 \sim 11\ 1111\ 1111\text{B}, \quad \text{即 } 000\text{H} \sim 3\text{FFH}。$$

地址线为 $A_0 \sim A_{10}$ 根的芯片,地址范围为

$$000\ 0000\ 0000 \sim 111\ 1111\ 1111\text{B}, \quad \text{即 } 000\text{H} \sim 7\text{FFH}。$$

通常,用 3FFH 表示 1KB 的存储区域,用 7FFH 表示 2KB 的存储区域,用 FFFH 表示 4KB 的存储区域,用 FFFFH 表示 64KB 的存储区域,用 FFFFFFFH 表示 1MB 的存储区域。

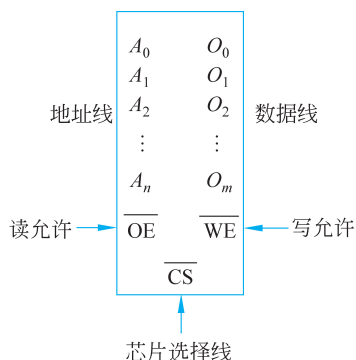


图 5-2 存储器件引脚示意图

2. 数据线

所有的存储器件都有数据线,负责数据的输出或者是数据的输入/输出。ROM 芯片的数据线只输出数据, RAM 芯片的数据线具有输入/输出功能。一般数据线的标识为 $O_0 \sim O_m$ 或者 $D_0 \sim D_m$ 。数据线的根数说明芯片内一个存储单元所能存放的二进制数的位数,如 HM6264 芯片含有 8 根数据线 $D_0 \sim D_7$,说明它的一个存储单元能存放 8 位二进制数。数据线的根数通常称为芯片的位宽,8 位宽的存储芯片通常称为字节宽存储器件。多数存储芯片是 8 位宽,也有 16 位宽、4 位宽、1 位宽的存储芯片。

3. 芯片选择线

任何单个的存储器件存储容量都是有限的。计算机系统的内存储器通常由多个存储器芯片构成。所以,所有存储器件都有至少一根芯片选择线,用来选中该器件,或者说激活这个器件。片选择线常被标识为片选 $\overline{CS}$ 、片使能 $\overline{CE}$ 或简称为选择 $\overline{S}$,这些符号上面的横杠,表示逻辑 0,也就是低电平有效。当在这个引脚上加一个低电平的时候,这个芯片被选中。如果存储芯片含有多根选择线,则只有在所有选择线都处于有效状态时,芯片才被激活,CPU 才可以对它进行读/写操作。

4. 控制线

每个存储器件都有控制数据输入/输出的控制线,通常标记为 $\overline{OE}$ 、 $\overline{WE}$,低电平有效。 $\overline{OE}$ 为读允许或输出使能。当 $\overline{OE}$ 为低电平时,存储单元存放的数据流出芯片,出现在系统数据总线上。 $\overline{WE}$ 为写允许。当 $\overline{WE}$ 为低电平时,存储器件从数据线上接收数据存储到芯片内部。ROM 存储器件因为只能读出数据不能写入数据,所以通常只有一个输出使能 $\overline{OE}$ 引脚,有些芯片将其标为 $\overline{G}$ (输出选通)。

有些 RAM 存储器件受到芯片引脚数量的限制,控制数据输入/输出的引脚只有一个,通常标记为 $R/\overline{W}$ 。当 $R/\overline{W}$ 引脚为高电平时,芯片输出数据,执行读操作;当它为低电平时,芯片接收数据,执行写操作。

如果 RAM 芯片有两个控制端 $\overline{WE}$ 和 $\overline{OE}$,则这两个引脚不能同时有效,也就是不能同时为低电平。当它们都为高电平时,都处于无效状态,这时芯片既不读出数据,也不写入数据,数据线引脚处于高阻状态。

5.1.3 存储器件的性能指标

衡量半导体存储器件性能的指标有很多,主要性能指标包括制造工艺、存储容量、供电电压、包装类型、适用的温度范围、存取时间、功耗、引脚配置、内部功能模块图、引脚名称及功能描述表、读/写时序图等。其中,存储容量和存取时间是使用芯片时必须关注的指标。

1. 存储容量

存储容量是指存储器件所能容纳二进制信息的总量,通常表示为存储单元个数 $\times$ 每个存储单元的位数。能存储 1 位二进制信息的电路称为存储元,多个存储元构成一个存储单

元,存储芯片由若干存储单元构成。例如,静态随机存储器芯片 62256 的容量为 $32\text{K} \times 8\text{b}$,说明它由 32K 个存储单元构成,每个存储单元由 8 个存储元构成,可以存储一字节的信息。动态随机存储器芯片 41256 的容量表示为 $256\text{K} \times 1\text{b}$,表示它由 256K 个存储单元构成,一个存储单元存储 1 位二进制信息。

2. 存取时间

存储芯片的存取速度通常用存取时间衡量。存取时间又称为访问时间或读/写时间,它是指从启动一次存储器操作(读或写)到完成该操作所需要的时间(t_{aA})。例如,读出时间是指从 CPU 向存储器发出有效地址开始,到将选中单元的内容送上数据总线为止所用的时间。写入时间是指从 CPU 向存储器发出有效地址开始,到信息写入被选中单元为止所用的时间。显然,存取时间越短,存取速度越快。

存取时间通常用 ns(纳秒)表示。芯片上的标识中“—7”表示 70ns,“—12”表示 120ns,依次类推。SRAM 的存取时间约为 60ns,最快可达到 1ns。高速缓冲存储器使用的 SRAM 的存取时间都在 10ns 以内。DRAM 的存取时间为 120~250ns。

一个系列的芯片会有一个基本的存取时间。例如,KM62256C 系列静态随机存储器芯片的存取时间最快是 55ns,最慢是 70ns。TMS4016 静态随机存储器芯片的最慢存取时间为 250ns,Intel 公司 27 系列可擦除只读存储器芯片 EPROM 的基本读取时间为 450ns。

5.2 随机存储器

随机存储器分为静态随机存储器(SRAM)和动态随机存储器(DRAM)。静态随机存储器以双稳态触发器为基本存储电路,保存的数据不需要刷新。与动态随机存储器比较,它的存取速度快、集成度低、功耗大。动态 RAM 以电容作为基本存储电路,每隔一段时间需要刷新一次,它的集成度高、成本低。

5.2.1 静态随机存储器

图 5-3 是静态随机存储器的功能结构图,由存储体和外围电路(行/列地址译码器、I/O 缓冲器和读/写控制电路等)组成。存储体由许多个存储元组成,这些存储元通常以矩阵的形式排列。存储体采用行、列地址单独译码的双译码方式。 $A_0 \sim A_i$ 为行地址输入端, $A_{i+1} \sim A_n$ 为列地址输入端。行地址译码器输出 2^{i+1} 根行选择线,每根行选择线选择一行;列地址译码器输出 2^{n-i} 根列选择线,每根列选择线选择一列。只有行、列均被选中的存储元,才能进行读或写的操作。I/O 缓冲器从系统数据总线接收数据或者把存储单元存储的数据输出到数据总线上。常见的静态随机存储器芯片都是 8 位宽的芯片,有 8 根数据线,一次可以输入/输出 1 字节。

静态随机存储器使用方便,在微型计算机领域有广泛的应用。常用的 SRAM 芯片有 6116($2\text{K} \times 8\text{b}$)、6232($4\text{K} \times 8\text{b}$)、6264($8\text{K} \times 8\text{b}$)、62128($16\text{K} \times 8\text{b}$)、62256($32\text{K} \times 8\text{b}$)和 62512($64\text{K} \times 8\text{b}$)等。下面以 SRAM 芯片 6264 为例,说明静态随机存储器的典型特性及工

作过程。

1. KM6264 SRAM 的引脚功能

KM6264 芯片是一个容量为 $8K \times 8b$ 的 CMOS 型 SRAM 芯片。它含有 8K 个存储单元,每个存储单元可以存储 8 位二进制信息。单一 +5V 供电,28 脚集成电路封装,额定功耗 200mW,典型存取时间为 200ns。KM6264 共有 28 根引线,包括 13 根地址线、8 根数据线和 4 根控制信号线,如图 5-4 所示。

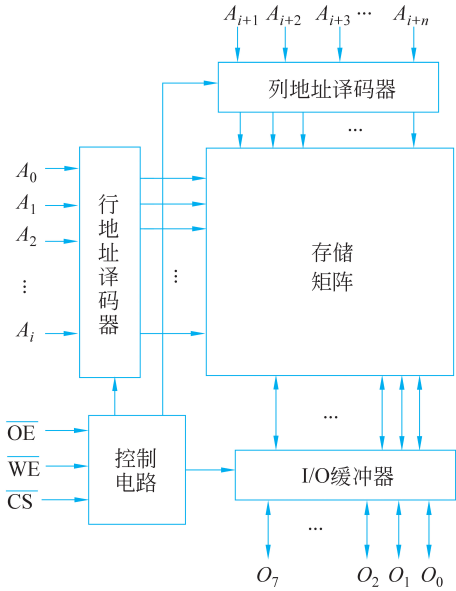


图 5-3 静态随机存储器的功能结构图

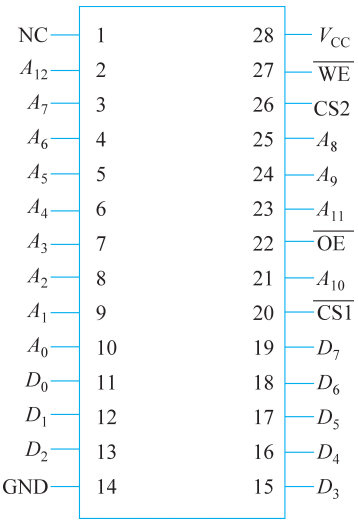


图 5-4 KM6264 SRAM 外部引线图

$A_0 \sim A_{12}$: 13 根地址信号线,用来选定 8K 个存储单元中的一个。这 13 根地址线通常与系统中地址总线的低 13 位($A_0 \sim A_{12}$)一对一地连接。

$D_0 \sim D_7$: 8 根数据线。每个存储单元可存储 8 位二进制数。这 8 根数据线通常与计算机系统的数据总线一对一地连接。

$\overline{CS1}$ 、 $\overline{CS2}$: 2 根片选信号线, $\overline{CS1}$ 低电平有效、 $\overline{CS2}$ 高电平有效。计算机系统中剩余的地址总线 $A_{13} \sim A_{19}$ 用来产生片选信号。

$\overline{OE}$: 输出允许,低电平有效,接系统读信号 $\overline{RD}$ 。

$\overline{WE}$: 写允许,低电平有效,接系统写信号 $\overline{WR}$ 。

V_{CC} : +5V 电源。

GND: 接地端。

NC: 空脚。

2. KM6264 的工作过程

(1) 写数据过程。CPU 把要写入的存储单元物理地址送到地址线上,其中 $A_0 \sim A_{12}$ 送到芯片的地址线 $A_0 \sim A_{12}$ 上, $A_{13} \sim A_{19}$ 经过地址译码电路生成片选信号使 $\overline{CS1} = 0$ 、 $\overline{CS2} = 1$;

接着 CPU 送出写控制信号 $\overline{\text{WR}}=0$, 并将数据发送到数据线 $D_0 \sim D_7$ 上, $\overline{\text{WR}}$ 信号使芯片的 $\overline{\text{WE}}$ 信号有效, 数据就写入了芯片指定的存储单元, 如图 5-5 所示。

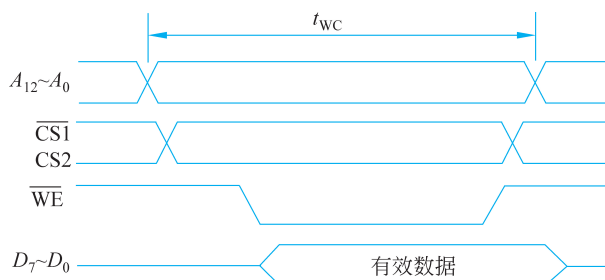


图 5-5 写周期时序图

(2) 读数据过程。CPU 把要读出的存储单元物理地址送到地址线上, 其中 $A_0 \sim A_{12}$ 送到芯片的地址线 $A_0 \sim A_{12}$ 上, 选定某个单元; $A_{13} \sim A_{19}$ 经过地址译码电路生成片选信号使 $\overline{\text{CS1}}=0$ 、 $\overline{\text{CS2}}=1$; 接着 CPU 送出 $\overline{\text{RD}}$ 信号, 它使芯片的输出允许信号 $\overline{\text{OE}}=0$, 指定单元的存储内容就出现在数据线 $D_0 \sim D_7$ 上, CPU 就从数据线捕获数据, 如图 5-6 所示。

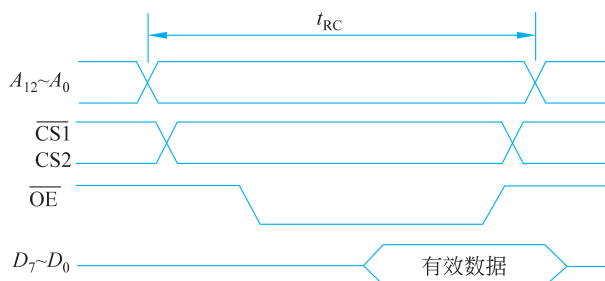


图 5-6 读周期时序图

CPU 的总线周期有固定的时序, 对存储器的读/写时间有要求。对存储器进行读操作时, 存储器必须在读信号 $\overline{\text{RD}}$ 有效期内将选中单元的内容送到数据总线上。进行写操作时, 存储器也必须在写信号 $\overline{\text{WR}}$ 有效期间将数据写入指定的存储单元, 否则就会出现读/写错误。

如果存储器的存取速度太慢, 不能满足 CPU 的要求, 就需要采取适当的措施解决这个问题。最简单的解决办法是降低 CPU 的时钟频率, 延长时钟周期 T_{CLK} , 但这样做会降低系统的整体运行速度。另一种方法是利用 CPU 上的 READY 信号, 在总线周期中插入一个或几个等待周期 T_w , 等待存储器操作完成。

3. HM6116 SRAM 芯片简介

如图 5-7 所示, HM6116 是容量为 $2\text{K} \times 8\text{b}$ 的高速静态 CMOS 随机存取存储器, 其基本参数如下。

(1) 高速: 存取时间为 100ns/120ns/150ns/200ns (分别以 6116-10、6116-12、6116-15、6116-20 为标志)。

(2) 低功耗: 运行时为 240mW, 空载时为 $2.0\mu\text{W}$ 。

(3) 与 TTL 兼容。

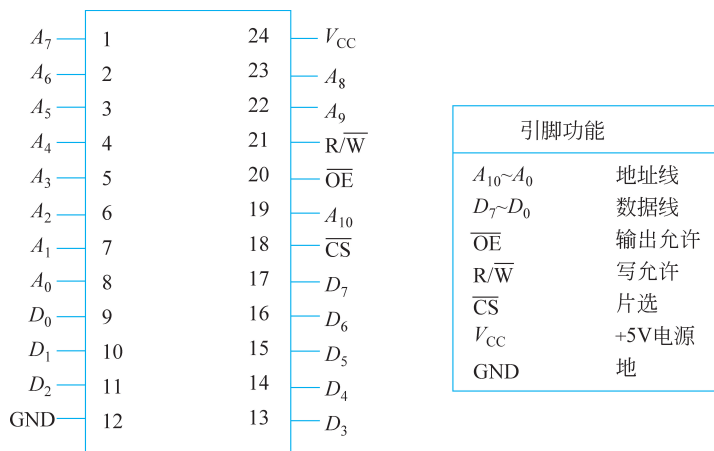


图 5-7 HM6116 SRAM 引脚图

(4) 引脚输出与 2716 芯片兼容。

(5) HM6116 有 11 根地址线 ($A_0 \sim A_{10}$), 8 根数据线 ($D_0 \sim D_7$), 1 根片选信号线 $\overline{CS}$, 2 根控制线: 读/写控制信号 $R/\overline{W}$ 和输出允许信号 $\overline{OE}$, 1 根电源线、1 根地线 GND。

4. KM62256 SRAM 芯片简介

图 5-8 描述了静态随机存储器 KM62256 的引脚图, 它的容量为 $32K \times 8b$, 28 脚集成电路封装, 典型的存取时间为 $85 \sim 150ns$ 。KM62256 的存取时间为 $55 \sim 70ns$ 。

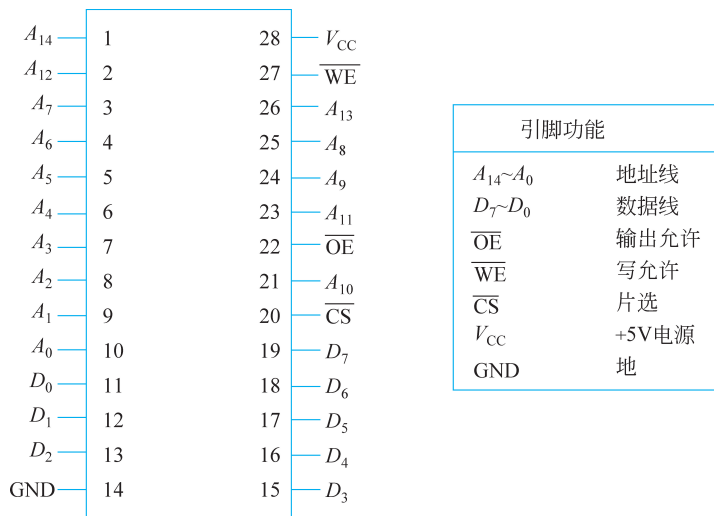


图 5-8 KM62256 SRAM 引脚图

8086/8088 CPU 工作在 5MHz 时钟下时, 总线周期中允许存储器或 I/O 端口存取数据的时间是 $460ns$ 。存储器的存取时间必须小于这个时间才能正常工作。以上的静态随机存储器芯片都可以直接与其相连, 不需要等待状态。

5.2.2 静态 RAM 芯片应用

在计算机系统中,系统总线是公共的数据通路,各种部件都挂接在系统总线上,总线的负载能力有限。在存储器芯片较少的系统中,存储器芯片可直接与总线相连;在存储器芯片较多的系统中,必须增加总线驱动能力,然后连接存储器芯片。KM6264 芯片的功耗很小(工作时为 15mW,未选中时仅 10 μ W),在简单的应用系统中,可直接和总线相连。

存储器芯片的应用就是将芯片正确接入计算机系统。根据 CPU 要求的地址范围,将芯片上的各种信号与计算机系统的地址线、数据线和控制线连接在一起,存储器芯片就接入了计算机系统。

(1) 数据线的连接。系统中所有的数据线都必须与芯片的数据线直接发生关联,双方都不能有剩余。如果芯片上的数据线和系统中的数据线的数量一致,则将它们一对一相连;如果芯片上的数据线少于系统数据线,如 2114(1K $\times$ 4b)只有 4 根数据线,则必须选用 2 个组成一组,构成数据线为 8 根的存储器芯片组,才可以与 8088 CPU 相连。如果芯片上的数据线多于系统数据线,说明选择的芯片不合适,必须更换。

(2) 控制信号线的连接。存储器只有两种操作:读和写。与读/写有关的控制信号通常只有两个:输出允许和写允许。它们应该分别与系统中的读/写控制信号线相连。

(3) 地址线的连接。一般存储芯片上地址线的数量比计算机系统中的地址线根数少,所以将芯片正确接入计算机系统,必须解决地址线不匹配的问题。芯片接入系统中时,芯片上的地址线和系统中的低位地址线一对一相连,使 CPU 可以选择芯片内的任一存储单元。系统中剩余的地址线在芯片中没有对应线,不能直接与芯片发生关联。

将一组输入信号转换为一个输出信号,称为译码。将系统中剩余的地址信号经过译码电路转换为一个输出信号,作为芯片的片选信号,称为地址译码。经过地址译码,系统中的全部地址线都与芯片产生了关联,使芯片中每个存储单元在系统地址空间中都有唯一的物理地址。地址译码是存储器芯片应用的核心和关键。

地址译码的方法有:全地址译码和部分地址译码。

1. 全地址译码

8088 系统中与存储器相关的信号线有 8 位数据总线 $D_0 \sim D_7$, 20 位地址总线 $A_0 \sim A_{19}$ 。8088 CPU 工作在最小模式下,与内存相关的控制信号有 $\overline{IO/\overline{M}}$ 、 $\overline{WR}$ 、 $\overline{RD}$ 3 根。在最大模式下,与内存相关的控制信号有两个: $\overline{MEMW}$ 、 $\overline{MEMR}$,这两个信号在 8086 系统中标为 $\overline{MRDC}$ 、 $\overline{MWTC}$ 。

全地址译码就是把系统中的全部地址线与芯片连接,其中高位地址线经过译码电路译码后作为芯片的片选信号;低位地址线与系统中的相应地址线一对一连接。全地址译码方式下,每个存储单元都有唯一的物理地址。

【例 5-1】 要求以全地址译码方式将 KM6264 SRAM 芯片接入计算机系统,地址范围为 F8000H~F9FFFH。

接线方法如下。

(1) 将 6264 芯片的数据线 $D_0 \sim D_7$ 与系统的 $D_0 \sim D_7$ 一对一连接,没有其他接法。

(2) 将 6264 芯片的地址线 $A_0 \sim A_{12}$ 与系统的 $A_0 \sim A_{12}$ 一对一连接, 没有其他接法。8088 系统中剩余的地址线 $A_{13} \sim A_{19}$ 需要通过译码电路, 作为片选信号与 6264 芯片连接。

(3) 将 6264 芯片的 $\overline{OE}$ 与系统的 $\overline{MEMR}$ 连接, 将 $\overline{WE}$ 与系统的 $\overline{MEMW}$ 连接。如果 8088 CPU 工作在最小模式下, 则将 6264 芯片的 $\overline{OE}$ 、 $\overline{WE}$ 分别与系统的 $\overline{RD}$ 、 $\overline{WR}$ 连接, 如图 5-9 所示。

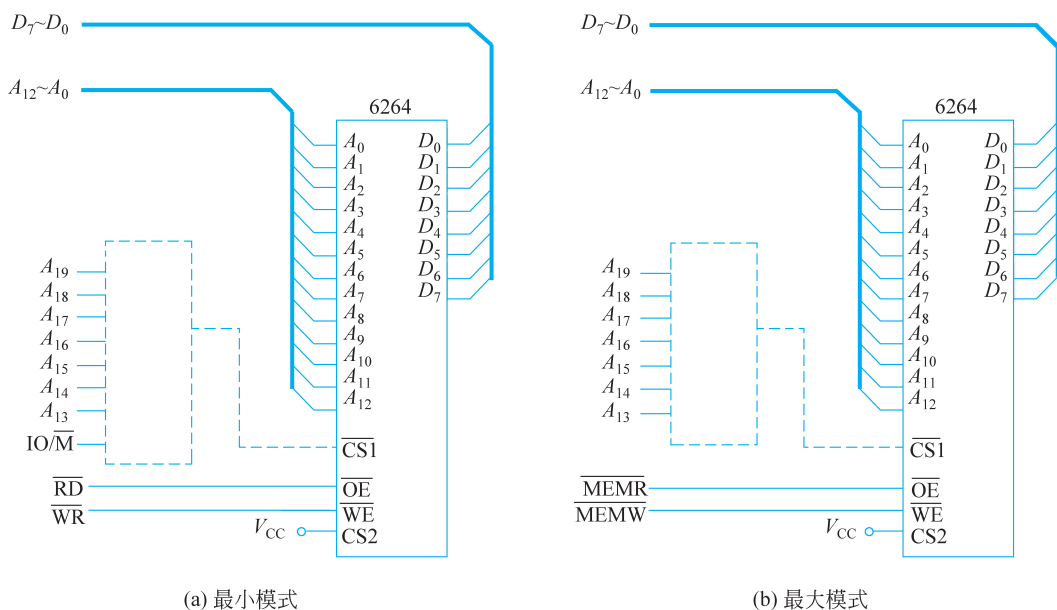


图 5-9 6264 芯片与系统总线的连接(未完成)

(4) 6264 芯片有 2 根片选信号线, 只要控制其中一根信号线就可以, 另一根接恒定高电平或低电平。通常将 CS2 接高电平, $\overline{CS1}$ 接控制信号。这种接法与 2764 芯片应用有关, 将在 5.3 节介绍。

图 5-9(a) 是 8088 CPU 工作在最小模式下的连接方法, 其中除地址线 $A_{13} \sim A_{19}$ 未连接外, 还有控制信号 $\overline{IO/M}$ 。 $\overline{IO/M}$ 信号在时序上与地址线基本相同, 所以在译码电路中可以把它当作一根地址线参与译码。图 5-9(b) 是 8088 CPU 工作在最大模式下的连接方法。

(5) 译码电路的设计有两种方法: 一种是利用基本的逻辑门电路搭建译码器; 另一种是利用专用的译码器芯片译码。

将芯片的地址范围以二进制形式表示, 如图 5-10 所示。其中, 虚线右边的 $A_0 \sim A_{12}$ 已经与芯片上的地址线一对一连接, 不用设计。左边的 $A_{13} \sim A_{19}$ 需要经过译码电路形成芯片的片选信号, 使得芯片在 8088 系统 1MB 的存储空间中定位在 F8000H~F9FFFH。通过对图 5-10 进行分析可以看到, 在 F8000H~F9FFFH 的地址范围内, A_{13} 和 A_{14} 始终为低电平, $A_{15} \sim A_{19}$ 始终为高电平, 译码电路的设计就要利用这个特点。

① 基本逻辑门电路译码。

先用基本逻辑门设计译码电路。 A_{13} 、 A_{14} 和 $\overline{IO/M}$ 是低电平信号, 可以各经过一个非门转为高电平; 再与 $A_{15} \sim A_{19}$ 一起经过一个 8 输入端的与非门 74LS30 产生低电平信号, 接到 6264 的片选端, 如图 5-11 所示。

地址总线	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	$A_{11} \sim A_8$	$A_7 \sim A_4$	$A_3 \sim A_0$
F8000H	1	1	1	1	1	0	0	0	0 0 0 0	0 0 0 0	0 0 0 0
F9FFFH	1	1	1	1	1	0	0	1	1 1 1 1	1 1 1 1	1 1 1 1

图 5-10 地址译码设计

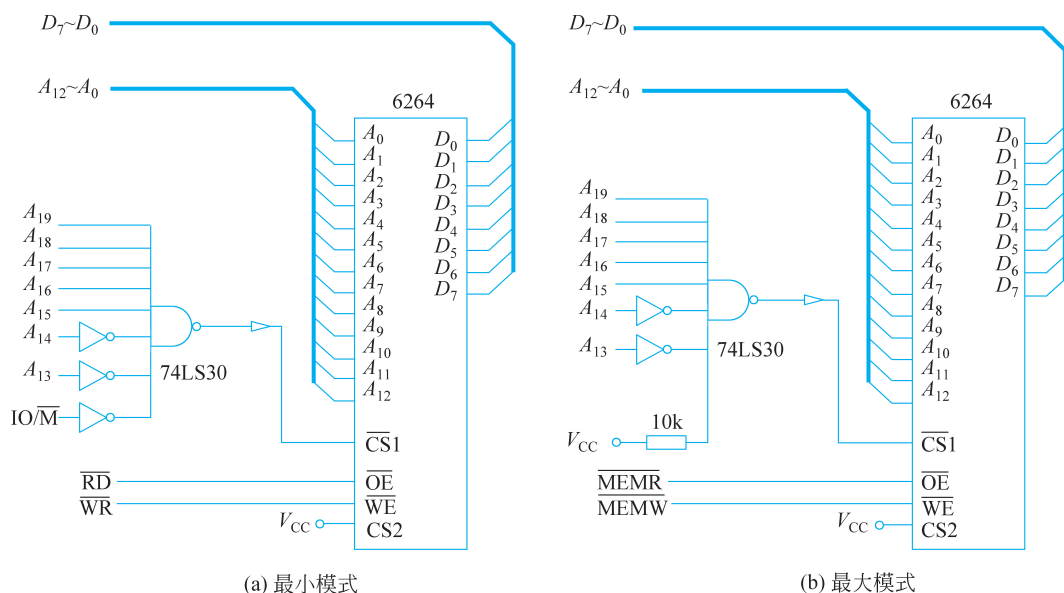


图 5-11 基本逻辑门电路译码

从图 5-11 可以看出,当 CPU 送出的内存单元地址在 F8000~F9FFFH 范围内时,6264 的片选端 $\overline{CS1}$ 就得到低电平,而 $CS2$ 恒定为高电平,如果此时读或写信号有效,CPU 就可以与 6264 芯片交换数据了。

74LS30 是 8 输入端的与非门,只有 8 个输入端都为逻辑 1 时,输出才为逻辑 0,因此图 5-11(b)中有一个引脚通过上拉电阻接到高电平上。

利用基本的逻辑门电路搭建译码器,设计灵活,可以设计出多种方案。图 5-12 是例 5-11 的另一种译码电路。

利用基本的逻辑门电路搭建译码器的缺点是,有多少个存储芯片,就要设计多少套译码电路,整体上看,设计复杂,不易修改。而利用专用的译码器芯片译码,方法简单,使用方便。

② 译码器芯片译码。

译码器有很多种,如 2-4 译码器、3-8 译码器、4-16 译码器等。最常用的译码器芯片 74LS138 是 3-8 译码器,74LS139 是双 2-4 译码器。74LS138 的引脚如图 5-13 所示,引脚 G_1 、 $\overline{G_{2A}}$ 、 $\overline{G_{2B}}$ 为控制端(使能端),即当 $G_1=1$ 并且 $\overline{G_{2A}}=\overline{G_{2B}}=0$ 时,译码器工作,否则译码器被禁止。 C 、 B 、 A 是译码输入端,负责输入 3 位二进制信息。 $Y_0 \sim Y_7$ 为 8 个译码输出端。译码器对 C 、 B 、 A 输入端的信号进行译码并通过 $Y_0 \sim Y_7$ 输出译码结果。74LS138 译码器的真值表如表 5-1 所示。译码器工作时,在任何情况下 8 个输出端只有一个为低电平,低电平有效。