第 3 章　BIOS 和 DOS 系统功能调用

3.1　重点与难点讲解

系统功能调用是程序设计的一个重要方面，DOS 利用类型码 21H 号中断提供给用户近百个系统功能，也就是近百个独立的中断服务程序。而驻留在 ROM 中的 BIOS 基本输入输出系统为用户提供了更基本的不依赖于操作系统的功能调用。这些系统功能调用为用户程序和系统程序提供了主要外设的控制功能，计算机系统软件就是利用这些基本的设备驱动程序，实现各种功能操作。这些中断子程序的入口地址已由系统置入中断向量表中，在汇编语言程序中可以用中断指令直接调用，程序员不必深入了解有关设备的接口电路，只需要遵循 DOS 或 BIOS 规定的调用方法就可以直接调用了。

某些系统功能调用，既能选用 DOS 中断，也能选用 BIOS 中断实现同样的功能。DOS 系统功能调用对硬件的依赖性很少，建议读者尽可能使用 DOS 功能。

BIOS 和 DOS 系统功能都是通过内部中断调用，调用方法非常方便、简单，只需要按以下步骤就可以完成 BIOS 或 DOS 功能调用。

(1) 设置入口参数：将入口参数送到指定的寄存器中。

(2) 将系统功能号送入 AH 寄存器中。

(3) 按中断类型码 n 调用 BIOS 或 DOS 中断，即执行中断指令 INT n 完成中断调用。

(4) 检查出口参数，分析调用结果是否正确。

有的子程序不需要入口参数，但大部分需要将入口参数送入指定的位置或指定的寄存器中。系统将中断类型码 21H 分配给 DOS，程序员只须给出以上 3 个方面的信息，执行 INT 21H 指令，DOS 就会自动转入相应的中断服务子程序去执行。而系统分配给 BIOS 的中断类型码 n 主要有 10H、13H、16H、17H 等，通过执行中断指令 INT n 可以直接调用。出口参数就是该子程序执行完以后的结果，一般存放在指定的寄存器中，有些子程序调用结束时会在屏幕上看到结果。

3.2　例 题 精 选

例 3.1　在显示器当前光标的位置输出显示一个字符。

解答：

```
MOV  AH,02H              ;调用号装 AH 寄存器
MOV  DL,'A'              ;要显示输出的字符的 ACSII 码装 DL 寄存器
INT  21H
```

例 3.2　向标准打印设备打印输出一个字符。

解答：

```
MOV  DL,'A'          ;入口参数是要打印的字符 A
MOV  AH,05H
INT 21H
```

例 3.3 从键盘输入字符送 AL 寄存器中。

解答：

```
MOV  AH,07H
INT  21H
```

例 3.4 在显示器上输出显示一字符串。

解答：

```
MOV  BX,SEG STRING
MOV  DS,BX           ;DS←输出字符串的段地址
LEA  DX,STRING       ;DX←输出字符串偏移地址
MOV  AH,09H          ;9 号功能调用
INT  21H             ;将内存中的字符串输出显示
```

例 3.5 从键盘接收字符串存到指定内存的输入缓冲区，直到输入是回车符为止。

解答：

```
MOV  BX,SEG BUF
MOV  DS,BX           ;DS←输入缓冲区段地址
LEA  DX,BUF          ;DX←输入缓冲区偏移地址
MOV  AH,0AH          ;0A 号功能调用
INT  21H             ;从键盘输入字符串存到指定内存的输入缓冲区中
```

例 3.6 设置中断向量。将中断向量表中的内容设置为新的中断子程序的入口地址。

解答：

```
MOV  BX,SEG INTHAND
MOV  DS,BX           ;DS←中断子程序的段地址
LEA  DX,INTHAND      ;DX←中断子程序的偏移地址
MOV  AH,25H          ;25H 号功能调用
MOV  AL,n            ;中断类型码 n
INT  21H
```

例 3.7 获取指定中断号的中断向量。

解答：

```
LEA  SI,OLDVECTOR
MOV  AH,35H          ;35H 号功能调用
MOV  AL,n            ;中断类型码 n
INT  21H             ;中断向量在 ES:BX 中
MOV  WORD PTR [SI],BX
MOV  WORD PTR [SI+2],ES
```

DOS 的 35H 号功能调用后，ES＝中断子程序的段地址，BX＝中断子程序的偏移地址。

例 3.8 设置光标开始行为 6 结束行为 7。

解答：

```
MOV  AH,1
MOV  CX,0607H
INT  10H
```

例 3.9 把光标设置到当前页的第 10 行第 20 列。

解答：

```
MOV  AH,2
MOV  DH,10
MOV  DL,20
MOV  BH,0
INT  10H
```

例 3.10 在光标位置显示字符。

解答：

```
MOV  AH,0AH
MOV  AL,'A'
MOV  BH,0
MOV  CX,1
INT  10H
```

例 3.11 判断键盘是否有输入。

解答：

```
    MOV  AH,1
    INT  16H
    JZ   NO
    MOV  BX,AX
         ⋮
NO:      ⋮
```

BIOS 的 16H 号功能调用后，如果没有键输入，ZF＝1；如果有键输入，ZF＝0，并且 AL＝字符码，AH＝扫描码。

例 3.12 等待用户按键。

解答：

```
MOV  AH,10H
INT  16H
MOV  BX,AX
```

按键后，AL＝字符的 ASCII 码，AH＝键盘扫描码。

例 3.13 利用 DOS 调用的 0AH 号功能调用，从键盘输入 80 个字符的字符串存入以 BUF 为首址的缓冲区中。

解答：

```
    DATA   SEGMENT
    BUF    DB 80                    ;定义缓冲区长度
           DB ?                     ;保留为填入实际输入的字符个数
           DB 80 DUP(?)             ;定义 80 个字节的存储空间,用以存放输入的字符
    DATA   ENDS
    CODE   SEGMENT
    ASSUME CS: CODE,DS: DATA
START:     MOV  AX,DATA
           MOV  DS,AX               ;DS←输入缓冲区段地址
           LEA  DX,BUF              ;DX←输入缓冲区偏移地址
           MOV  AH,0AH              ;0AH 号功能调用
           INT  21H                 ;DOS 调用
           MOV  AH,4CH              ;用 4CH 号功能调用返回 DOS
           INT  21H
CODE       ENDS
           END  START
```

如果从键盘输入了 ABCD 和回车，则 BUF 缓冲区中依次存放：50H，04H，41H，42H，43H，44H，0DH，00H，00H，…，00H，00H。

例 3.14 将一字符串信息输出到屏幕上显示。

解答：

(1) 用 DOS 调用的 02 号系统功能调用完成：

```
STRING     DB 'Hello,everyone!'     ;定义一字符串
LEN        EQU    $-STRING          ;定义一常量=字符串的长度
           ⋮
           MOV  CX,LEN              ;CX 存字符串的长度
           LEA  BX,STRING           ;BX 指向串的首地址
           MOV  AH,02H              ;调用号存 AH 中
NEXT:      MOV  DL,[BX]             ;输出显示的字符送 DL 中
           INT  21H                 ;02H 号调用一次,显示一个字符
           INC  BX                  ;修改指针,指向下一个字符的地址
           LOOP NEXT                ;重复执行 02H 系统功能调用,直到 CX=0 退出循环
             ⋮
```

(2) 用 DOS 调用的 09 号系统功能调用完成：

```
STRING     DB 'Hello,everyone!',0DH,0AH,'$'
             ⋮                      ;定义一字符串,以$ 结尾,光标自动换行
           MOV  BX,SEG STRING       ;
           MOV  DS,BX               ;字符串的段地址送 DS
           LEA  DX,STRING           ;字符串的偏移地址送 DX,使 DS: DX=串首地址
           MOV  AH,09H              ;9 号功能调用
           INT  21H                 ;将内存中的字符串输出显示
             ⋮
```

例 3.15 利用 DOS 系统功能调用,实现人机对话。

解答:

```
BUF       DB 100                    ;定义缓冲区长度
          DB ?                      ;保留为填入实际输入的字符个数
          DB 100 DUP(?)             ;定义 100 个字节的存储空间,用以存放输入的字符
STR       DB 'What is your name?',0AH,0DH,'$'        ;定义一字符串以$ 结束
            ⋮
          LEA  DX,STR               ;字符串的偏移地址送 DX,使 DS: DX 指向 STR 首地址
          MOV  AH,09H               ;9 号功能调用
          INT  21H                  ;显示提问: What is your name?
          LEA  DX,BUF               ;DS: DX 指向缓冲区 BUF 首地址
          MOV  AH,0AH
          INT  21H                  ;0AH 号调用,接收键盘的回答信息,以回车结束
          MOV  DL,0AH               ;换行的 ASCII 码是 0AH
          MOV  AL,02H
          INT  21H                  ;实现换行
          LEA  DX,BUF+2             ;显示输入的回答信息
          MOV  AH,09H               ;
          INT  21H                  ;
            ⋮
```

例 3.16 从键盘输入一个字符并显示在屏幕上。

解答:

(1) 用 BIOS 功能调用实现:

```
MOV  AH,0                 ;功能号送 AH 中
INT  16H                  ;BIOS 调用: 从键盘读字符,输入字符的 ACSII 码→AL(出口参数)
MOV  BX,0                 ;设置入口参数,字符的 ACSII 码在 AL 中
MOV  AH,0EH               ;功能号送 AH 中
INT  10H                  ;BIOS 调用: 显示 AL 中的字符
```

(2) 用 DOS 功能调用实现:

```
MOV  AH,1                 ;功能号送 AH 中
INT  21H                  ;键盘输入并回显,输入字符的 ACSII 码在 AL 中
```

例 3.17 检查程序执行过程中是否有键盘输入,按任意键退出。

解答:

```
L:  ⋮
    MOV  AH,0BH           ;功能号送 AH 中
    INT  21H              ;读键盘状态,AL=FFH 有输入,AL=0 无输入
    INC  AL               ;
    JNZ  L                ;无键盘输入,继续
    RET                   ;有输入,返回 DOS
```

例 3.18 设置光标位置,将光标置在第 8 行 30 列。

解答：

```
MOV  BH,0                ;BH=页号,通常取 0 页
MOV  DH,8                ;DH=行号,取值 0~24
MOV  DL,30               ;DL=列号
MOV  AH,2                ;功能号
INT  10H                 ;BIOS 调用
```

例 3.19 置光标开始行是 10,结束行是 9,并把它设置到第 5 行第 6 列。

解答：

```
MOV  CH,10               ;光标开始行
MOV  CL,9                ;光标结束行
MOV  AH,1                ;功能号 1: 显现光标
INT  10H                 ;BIOS 调用
MOV  DH,4                ;5 行
MOV  DL,5                ;6 列
MOV  BH,0                ;0 页
MOV  AH,2                ;功能号 2: 设置光标位置
INT  10H                 ;BIOS 调用
```

例 3.20 在当前光标处显示字符：用绿色清屏,然后在第 20 行 30 列显示 8 个白底黑字 R。

解答：

```
MOV  AL,0                ;屏幕初始化
MOV  BH,20H              ;背景绿色
MOV  AH,6                ;功能 6 号: 屏幕初始化
MOV  CX,0                ;窗口左上角行列号
MOV  DH,24               ;窗口右下角行号
MOV  DL,79               ;窗口右下角列号
INT  10H                 ;清屏幕
MOV  AH,2                ;功能 2 号: 置光标位置
MOV  BH,0                ;页号
MOV  DH,20               ;
MOV  DL,30               ;光标设置在第 20 行 30 列
INT  10H                 ;
MOV  AL,'R'              ;(AL)=显示字符的 ACSII 码
MOV  CX,8                ;(CX)=字符重复显示的次数
MOV  BH,0                ;页号
MOV  BL,70H              ;显示属性: 白底黑色
MOV  AH,9                ;功能号
INT  10H                 ;显示 8 个字母'R'
```

第 4 章　存储器组织结构

4.1　重点与难点讲解

4.1.1　有关存储器的概念

存储器是用来存储计算机工作时必需的程序和数据。处理器实际运行的大部分周期是用于对存储器的读写或访问，所以，存储器的性能很大程度上决定了计算机性能的优劣。存储器是计算机系统中重要的组成部分，存储器的存取速度、存储容量、功耗以及可靠性是计算机技术发展的一个重要标志。

当前的各种计算机系统中，为了提高速度，增加存储容量，降低成本，广泛采用分级结构来组织存储器系统：位于 CPU 内部的寄存器、高速缓冲存储器(cache)、主存储器、外存储器，它们的存取速度依次递减，而存储容量则依次递增，如图 4.1 所示。

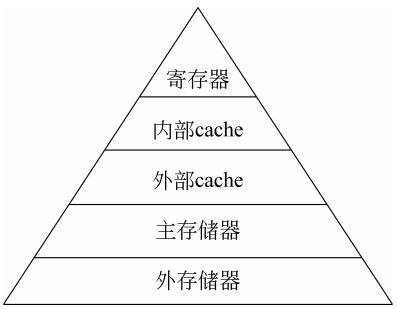

图 4.1　存储器系统的层次结构

1. 寄存器

寄存器是最高一级的存储单元，它包含在微处理器中，CPU 对其内部寄存器的访问速度最快，但由于受芯片面积和集成度的限制，寄存器的数量是有限的。

2. 高速缓冲存储器(cache)

高速缓冲存储器是指可以与 CPU 进行高速数据交换的存储器，其存取速度与 CPU 相匹配，因此速度很快。它一般只装载当前用得最多的程序或数据，使 CPU 能以最快的速度工作。一级缓存(L1 Cache)是 CPU 的第一层高速缓存，内置的 L1 高速缓存的容量和结构对 CPU 的性能影响较大，不过高速缓冲存储器均由静态 RAM 组成，结构较复杂，在 CPU 芯片面积不能太大的情况下，L1 级高速缓存的容量不可能做得太大。一般 L1 缓存的容量通常在 32～256KB。二级缓存(L2 Cache)是 CPU 的第二层高速缓存，分内部和外部两种芯片。CPU 内部的二级缓存运行速度与主频相同，而外部的二级缓存则只有主频的一半。L2 高速缓存容量也会影响 CPU 的性能，原则是越大越好，现在家庭用 CPU 的 L2 高速缓存最大的是 512KB，而服务器和工作站上用 CPU 的 L2 高速缓存容量更高达 1～3MB。总之，缓存(cache)的大小是 CPU 的重要指标之一，其结构与大小对 CPU 速度的影响非常大。简单地讲，缓存就是用来存储一些常用或即将用到的数据或指令，当需要这些数据或指令的时候直接从缓存中读取，这样比到内存甚至硬盘中读取要快得多，能够大幅度提升 CPU 的处理速度。

3. 主存储器

主存储器用于存放当前运行的程序和数据。由于 CPU 的寻址大部分落在高速缓冲存储器上，主存储器可以采用速度稍慢而容量比较大的存储器芯片，其速度对系统性能的影响

不会太大。主存储器除了大量的 DRAM 外，还需要一定的 ROM，ROM 主要用于解决初始化的一系列操作，如设备检查、接口电路的初始化以及启动操作系统等工作，一般说来，ROM 的存取时间比较长，但这种少量慢速的 ROM 存储器只在开机时运行，对系统性能影响不大。

4. 外存储器

外存储器也称海量存储器，其容量很大，而存取速度比主存储器要慢得多，大量用于后备存储器，存储各种程序和数据。在高档计算机系统中，外存储器还广泛用作虚拟存储器的硬件支持。

4.1.2 半导体存储器的性能指标

随机读写存储器(Random Access Memory，RAM)和只读存储器(Read Only Memory，ROM)是计算机系统中不可缺少的两大类半导体存储器芯片。

其中，RAM 是一种随机读写存储器，但一旦掉电，所存的信息就会丢失，只能作为二进制信息的缓冲存储器。主要用于：

(1) 存放当前正在执行的程序和数据，如用户的程序、程序的中间运行结果以及掉电时无须保存的输入输出数据和参数。

(2) 作为输入输出数据缓冲存储器，如显示输出缓冲存储器、打印输出缓冲存储器、键盘输入缓冲存储器。

(3) 作为中断服务程序和子程序中保护 CPU 现场信息的堆栈。

(4) 在后备电源及掉电保护电路的支持下，作为存放系统配置参数和状态参数的存储器。如计算机中的 CMOS 电路，用于保存系统当前设置的各种参数，包括硬盘类型、显示方式、键盘参数、内存大小、cache 状态、cache 范围、总线时钟选择等。

而 ROM 具有掉电后信息不会丢失的特点(也称非易失性)，弥补了读写存储器(RAM)性能上的不足，正是这一特性，ROM 用来存放固定不变的程序或重要参数，计算机系统中就是用 ROM 存放引导程序、基本输入输出程序(BIOS)、固定的系统表格等。ROM 分为掩膜 ROM、只能写入一次的可编程只读存储器(PROM)、紫外线擦除可编程只读存储器(EPROM)、电擦除可编程只读存储器(EEPROM)、闪烁存储器(Flash M)等。

下面是半导体存储器的主要性能指标。

1. 存储容量

存储器可以容纳的二进制信息量称为存储容量。一般是以存储器所能存储的字数乘以字长来表示：

$$\text{存储容量} = \text{字数} \times \text{字长}$$

如果一个存储器能存 4096 个字，字长为 16 位，则存储容量为 4096×16 位。

位(b)是二进制数的最基本单位，也是存储器存储信息的最小单位，8 位二进制数称为一个字节(B)，计算机系统中，存储器是按字节进行编址的，所以，常用字节数表示存储容量，如一个 8 位机的内存容量为 64KB，就是 64Kb×8，一个 16 位的内存容量是 1MB，或 1Mb×8。

2. 存取速度

存储器的存取速度常用存取时间或存取周期来描述，存取时间是指访问（读写）一次存储器所需要的时间。存取周期是指连续两次访问存储器所需要的最小时间，存储周期等于存取时间加上存储器的恢复时间。半导体存储器的存取时间、存取周期通常以纳秒（ns）为单位。存取时间或存取周期越小，速度越快。目前计算机内存读写时间一般在 10 纳秒以内，而高速缓冲存储器（cache）的存取速度更快。

3. 可靠性

存储器的可靠性用平均无故障时间 MTBF 来衡量。MTBF 可以理解为两次故障之间的平均时间间隔。MTBF 越长，表示可靠性越高，即保持正确工作能力越强。

4. 性能价格比

性能主要包括存储器容量、存储周期和可靠性三项内容。性能价格比是一个综合性指标，常以每位价格来描述，对于不同的存储器有不同的要求。对于外存储器，要求容量极大，而对缓冲存储器则要求速度非常快，容量不一定大。因此，性能/价格比是评价整个存储器系统很重要的指标。

4.1.3 半导体存储器芯片的主要外部引脚

存储器芯片的外部引脚主要包括：

（1）地址线——地址线的多少决定了该芯片有多少个存储单元，如 SRAM 的 6264 芯片，有 13 条地址线，表明该芯片可寻址 $2^{13}=8192$ 个存储单元，在与系统连接时，这 13 条地址线通常接到系统地址总线的低 13 位上，以便 CPU 能够寻址芯片上的各个单元。

（2）数据线——数据线通常是双向的，数据线的多少决定了芯片上每个存储单元的二进制位数。如 6264 芯片有 8 条数据线，表明 6264 芯片内的每个存储单元中可以存储 8 位二进制信息。系统连接时，芯片的数据线与系统的数据总线相连，当 CPU 访问该芯片上的某个存储单元时，输出或输入的数据都是通过数据线传送的。

（3）控制信号——控制信号主要包括读写控制、编程控制以及片选信号，片选有效时，允许对该芯片进行访问操作，该控制端一般与系统的高位地址线相关联，连接时有多种处理方法：全译码、部分译码、线选法等。

在计算机系统中，CPU 对主存储器的访问（读写操作）是通过执行指令完成的，指令中已经包含了将要访问的存储单元的地址，通过地址总线输出地址信号，然后发出相应的读或写的控制信号，最后在数据总线上进行数据交换，所以，存储器与 CPU 的连接主要涉及地址总线、数据总线和控制总线的连接。数据总线的连接是由 CPU 数据总线的位数和存储器芯片的数据位数决定的。地址总线的连接需要根据存储器系统的容量和存储器地址的范围共同决定。而控制总线的连接必须由 CPU 和存储器芯片对控制信号的要求来决定，主要有存储器请求信号和读写控制信号的连接，CPU 根据所执行的指令的要求，按一定的时序向存储器芯片的控制端发出这些控制信号，使存储器系统处于某种工作状态。

4.2 例题精选

例 4.1 什么是存储器的分级结构？存储器采用分级结构的目的是什么？通常有哪几级存储器？性能上有什么特点？

解答：当前的各种计算机系统中，为了提高速度，增加存储容量，降低成本，广泛采用分级结构组织存储器系统：位于 CPU 内部的寄存器、高速缓存器(cache)、主存储器、外存储器，它们的存取速度依次递减，而存储容量则依次递增。

例 4.2 在多级存储体系中，采用 cache-主存结构的作用主要解决什么问题？

解答：高速缓冲存储器(cache)是指可以与 CPU 进行高速数据交换的存储器，其存取速度与 CPU 相匹配，因此速度很快。它一般只装载当前用得最多的程序或数据，使 CPU 能以最快的速度工作。所以，采用 cache-主存结构的作用主要解决 CPU 与内存之间的速度匹配。

例 4.3 和外存储器相比，内存储器的特点是什么？

解答：内存储器的特点是容量小、速度快、成本低。

例 4.4 EPROM 是指什么存储器？

解答：EPROM 是指可擦除可编程只读存储器。

例 4.5 外存储器是大容量的存储器，常用来存储各种程序和数据，还可以有什么作用？

解答：外存储器还广泛用于构建一个虚拟存储器的硬件支持。虚拟地址空间的大小由微处理器内部的存储器管理部件 MMU 决定，并由外存储器支持。

例 4.6 什么是虚拟存储器？

解答：虚拟存储器是通过硬件和软件的管理来扩大用户的可用存储空间的一种技术，它将内存和外存统一编址，形成一个比内存空间大得多的存储空间，称为虚拟存储空间，虚拟地址空间的大小由微处理器内部的存储器管理部件 MMU 决定，并由外部存储器支持。

例 4.7 虚拟存储器中地址映射方式有哪些？

解答：实现虚拟存储器的关键是快速地实现虚拟地址向物理地址的转换，人们把这种地址变换称为地址映像。通常采用的地址映像方式有段式、页式、段页式，因此，就形成了常见的段式虚拟存储器、页式虚拟存储器、段页式虚拟存储器。

例 4.8 半导体存储器主要分为哪两类？它们的主要区别是什么？

解答：半导体存储器主要分为随机读写存储器(RAM)和只读存储器(ROM)。

RAM 是一种随机读写存储器，但一旦掉电，所存的信息就会丢失，只能作为二进制信息的缓冲存储器。

ROM 具有掉电后信息不会丢失的特点，ROM 用来存放固定不变的程序或重要参数。

例 4.9 半导体存储器的主要性能指标有哪些？

解答：半导体存储器的主要性能指标有存储容量、存取速度、可靠性、性能价格比等。

例 4.10 已知一个 SRAM 芯片的容量是 8Kb×8，该芯片有一个片选信号引脚和一个读写控制引脚，问该芯片至少有多少个引脚？地址线多少条？数据线多少条？其他还有什么信号线？

解答：15 个引脚。13 条地址线。8 条数据线。1 条输出允许引脚。

例 4.11 已知一个 DRAM 芯片外部引脚信号中有 4 条数据线、7 条地址线，计算它的容量。

解答：如果是采用全译码方式，它的容量是 128×4 位。

如果是采用行、列地址分时复用技术，7 条地址线分 2 次传送 14 位地址，它的容量是 16K×4 位。

例 4.12 32M×8 的 DRAM 芯片，其外部数据线和地址线为多少条？

解答：25 条地址线，8 条数据线。

例 4.13 已知某 SRAM 芯片有 8 条数据线，14 条地址线，要求用该芯片构成一个具有 0000H～BFFFH 寻址空间的内存。这样的芯片应该选几片？说明各芯片的地址分配。

解答：该存储器容量是 BFFFH－0000H＋1＝C000H＝48KB。

每片芯片的容量是 214B＝16KB，所以，应选的芯片数是：48KB/16KB＝3 片。

第一片芯片的地址范围：0000H～3FFFH。

第二片芯片的地址范围：4000H～7FFFH。

第三片芯片的地址范围：8000H～BFFFH。

例 4.14 DRAM 为什么需要定时刷新？

解答：动态 RAM 是以电容为基本存储单元，需要刷新电路对存储单元进行定时刷新，否则信息会丢失。

例 4.15 半导体存储器常用的实现片选的控制方式有哪些？各有什么特点？

解答：常用的实现片选的 3 种控制方法是全译码法、部分译码法、线选法。

例 4.16 主存储器与 CPU 连接时应该考虑哪些问题？

解答：

(1) 考虑 CPU 总线的负载能力，接入缓冲器或总线驱动器可以增加总线的驱动能力。

(2) CPU 与存储器的速度匹配。

(3) 存储器与 CPU 信号线的连接，包括数据线的连接、控制线的连接、地址线的连接。

例 4.17 某存储器系统的起始地址为 A000H。使用容量为 2K×8 的 SRAM 芯片 6116，构成 8K×8 位的存储器系统模块。

由于一个芯片的容量是 2K×8，可以访问 2048 个存储单元，每个存储单元的数据线为 8 位，所以，每个芯片有 11 条地址线 A_0～A_{10} 和 8 条数据线 D_0～D_7，要构成 8K×8 位的存储器系统，需要用 4 片这样的芯片。根据芯片容量的要求，4 片 6116 芯片的始末地址分别是：A000H～A7FFH、A800H～AFFFH、B000H～B7FFH、B800H～BFFFH。

这样，该存储器系统的地址范围是 A000H～BFFFH。CPU 使用 16 位地址线访问该存储器系统。

由于只扩展字，因此 4 个芯片的数据线直接并联在一起，连接到系统的数据总线上，读写控制线也并联，芯片的地址线 A_0～A_{10} 直接连到地址总线的对应位上，地址总线的高位 A_{11}～A_{15} 接译码器的输入，译码器的输出用作各个芯片的片选信号，这就是字扩展，采用全译码法。其系统连接图如图 4.2 所示。

该存储器系统对地址信息采用分段译码，地址总线中的低 11 位地址线 A_0～A_{10} 与 4 片 6116 芯片的地址线 A_0～A_{10} 相连，地址总线的高位地址 A_{11}～A_{19} 通过译码器产生“片选”信

号。在对存储器某单元进行读操作时，地址信号通过分段译码选中了某芯片中的某个单元，随后，存储器读命令有效，将该单元的信息读到数据总线上。而对存储单元进行写操作时，要写入的信息放在数据总线上，指定的存储单元的地址放在地址线上，通过分段译码，选中了某一芯片的某个单元，随后的存储器写命令有效，将数据线上的信息写入指定的存储单元中。

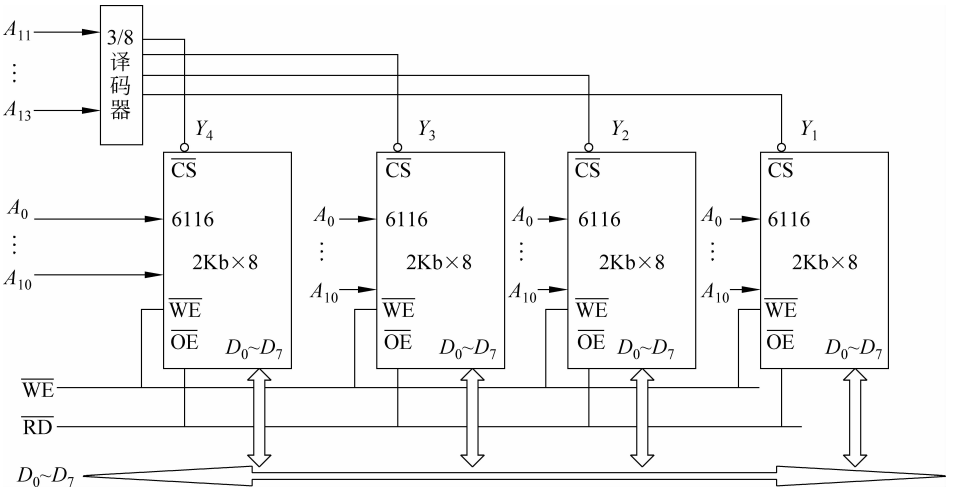

图 4.2　存储器系统连接图

例 4.18　用 64K×1 位 DRAM 构成 256KB 的存储器系统。问：需要多少这样的 DRAM 芯片？设系统地址线有 20 位，采用全译码方式，则片内地址多少位？需要多少位地址作为片外地址译码？

解答：一个 256KB 的存储器系统，表明有 256 个存储单元，每个存储单元的数据是 8 位。而 64Kb×1 的 DRAM 芯片内只有 64×1024 个存储单元，每个存储单元只有 1 位，使用这样的芯片就需要同时进行字扩展和位扩展。

所以，该系统需要的 64Kb×1 的 DRAM 芯片数＝位扩展数×字扩展数。

本题中需要芯片数＝8×4＝32(片)。

设系统地址线有 20 位，采用全译码，片内地址 16 位(A_0～A_{15})，片外地址 4 位(A_{16}～A_{20})。先进行位扩展，用 8 片 64Kb×1 的 DRAM 芯片构成一个 64Kb×8 的芯片组；再作字扩展，用 4 个 64Kb×8 的芯片组构成一个 256KB 的存储器系统。其系统连接图如图 4.3 所示。

例 4.19　用 64Kb×1DRAM 构成 256KB 的存储器系统。

① 需要多少个 RAM 芯片？

② 设系统地址线有 20 位，采用全译码，则需要多少位地址作为片外地址译码？片内地址多少位？

解答：一个 256KB 的存储器系统，表明有 256 个存储单元，每个单元的数据是 8 位。而使用 64Kb×1 的 DRAM 芯片，就需要同时进行字扩展和位扩展。

需要芯片数＝位扩充数×字节扩充数，本题中

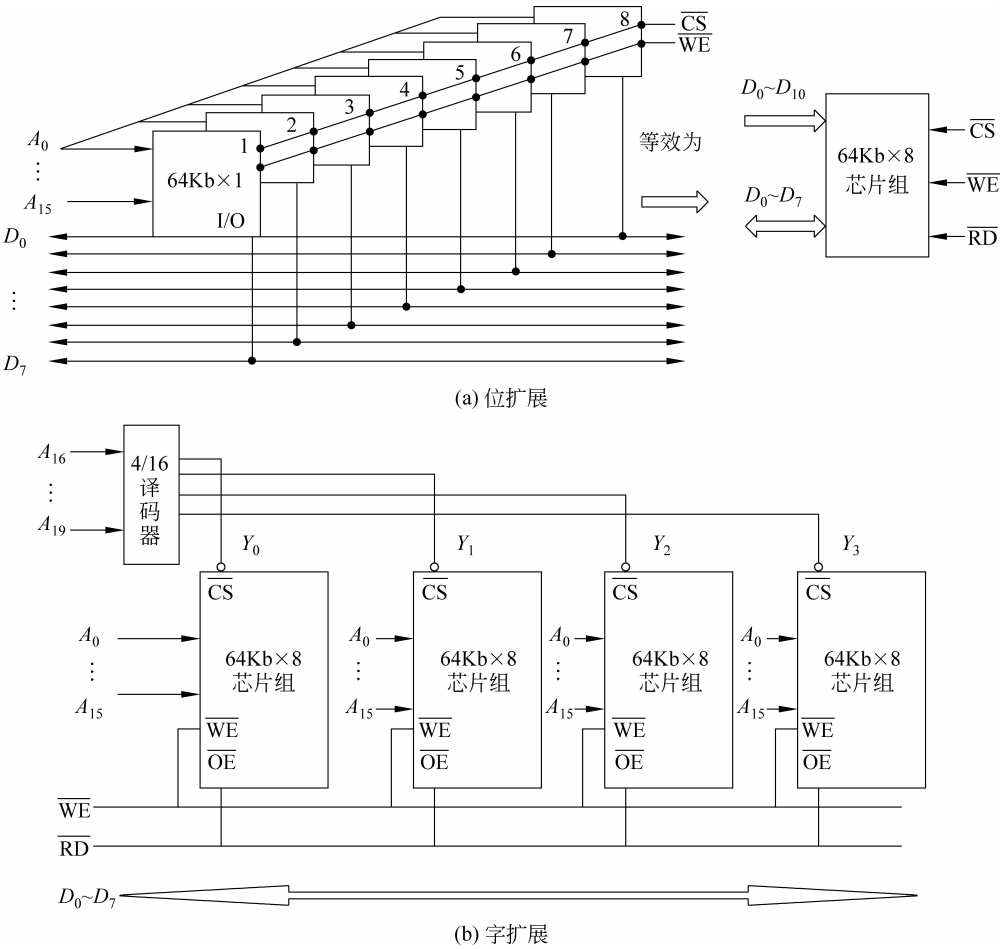

图 4.3　用 64Kb×1DRAM 构成 256KB 的存储器系统

位扩充数 = 8/1 = 8

字节扩充数 = 256/64 = 4

所以需要芯片数 = 8 × 4 = 32(片)

片内地址 16 位,片外地址 = 20 − 16 = 4 位

例 4.20　采用全译码方式设计一个 24KB 的存储器系统,其中,16KB 是 RAM,选用两片 6264 芯片;8KB 是 ROM,选用两片 2732 芯片。地址范围是 00000H～05FFFH。

解答:根据设计要求,CPU 的 8 条数据线刚好与存储器芯片的 8 条数据线相连。而控制线的连接必须根据具体 CPU 和存储器芯片对控制信号的要求来决定,包括存储器请求信号、存储器读写信号的连接等。

其次是地址线的连接:CPU 的 A_0～A_{12} 直接与 6264 芯片的 13 条地址线相连。而 2732 只有 12 条地址线,CPU 的 A_0～A_{11} 直接与 2732 芯片的 12 条地址线相连。这样,芯片地址线就连好了。

片选信号通过一片 138 译码器产生,译码器的 3 条地址线 C、B、A 分别与 A_{15}、A_{14}、A_{13} 相连。译码器的输出 $\overline{Y}_0$、$\overline{Y}_1$ 分别接两片 6264 芯片的片选,$\overline{Y}_2$ 和 A_{12} 经二次译码后输出给两

片 2732 的片选，如图 4.4 所示。

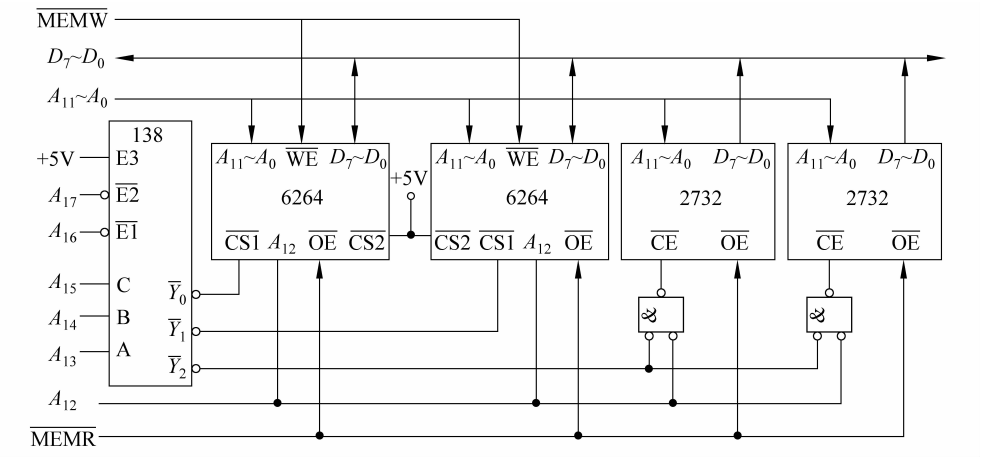

图 4.4　存储器系统的设计

按上述的连接方式，各芯片的地址范围如图 4.5 所示。

芯片	A_{19}~A_{16}	A_{15}~A_{13}	A_{12}	A_{11}~A_0	一个可用地址
6264-1	XX 00	0 0 0	全0~全1	全0~全1	00000H~01FFFH
6264-2	XX 00	0 0 1	全0~全1	全0~全1	02000H~03FFFH
2732-1	XX 00	0 1 0	0	全0~全1	04000H~04FFFH
2732-2	XX 00	0 1 0	1	全0~全1	05000H~05FFFH

3/8译码器译码

通过与门组合
产生译码输出信号

图 4.5　各芯片的地址范围