

### 3.2.2 DOC(Disk on Chip)

#### 3.2.2.1 DOC 简介

DOC(Disk on Chip) 是 M-Systems 公司的系列半导体存储设备的一种, 是世界上第一种超大容量的固态 flash 盘。它在单个芯片上集成了磁盘控制器和存储设备, 特别适合机顶盒、掌上电脑、车载 PC、瘦服务器和移动终端等瘦客户机和 Internet 设备使用。此类设备无一例外的都在存储设备的重量、体积、功耗和存储容量方面有苛刻的要求。DOC 提供一种低代价的传统硬盘和软盘的替代方式, 能够胜任一个系统的外存储器的工作, 提供系统引导、读和写等功能。

#### 3.2.2.2 DOC 硬件

DOC 系列采用标准 32 针 DIP (Dual Inline Package 双列直插) 封装方式, 有 2M 到 288M 各种规格, 它的后续产品 DOC 2000 以及 DOC Millennium 与之完全兼容。

在 DOC 中内置了 M-Systems 标准的 TrueFFS 驱动程序。在 PC 环境下 (确切的说, 是在 BIOS 的支持下) 能够象标准的磁盘一样操作, 在没有软/硬盘的情况下进行操作系统引导。另外, 在有软/硬盘的情况下, 也可以定制启动设备。DOC 的引脚示意图如 3-2 所示。

在图 3-2 中我们特意将 DOC 的内部结构表示了出来。在它里面有两块芯片, 其中一块是提供 TrueFFS 的驱动程序以及扩展 BIOS 的程序代码, 它可以保证 DOC 象普通的硬盘一样操作。

另一块芯片是普通的 flash memory, 它是存储数据的介质。在系统上电以后, DOC 将一些信息自动放到扩展 BIOS 空间 (通常是 0C8000H 到 0EFFFFH 这个地址范围) 中的一段 8K 的存储区域。这块存储区域就像硬盘的 I/O 地址, 在它的特定位写入数据, 它就可以让 DOC 执行相应的操作。从图 3-5 可以很清楚地了解 DOC 的工作方式。

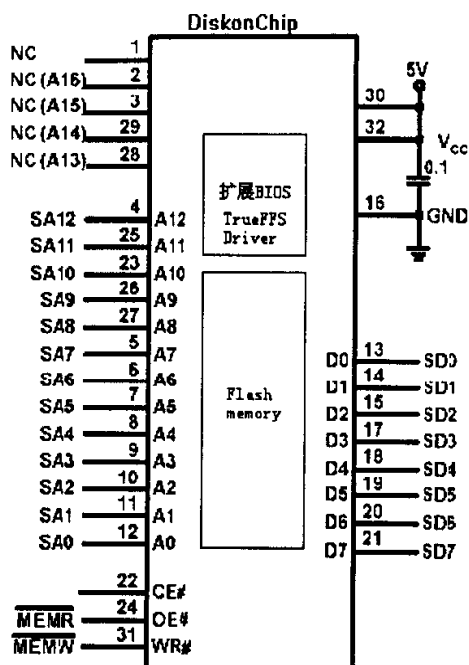


图 3-4 DOC 的引脚示意图

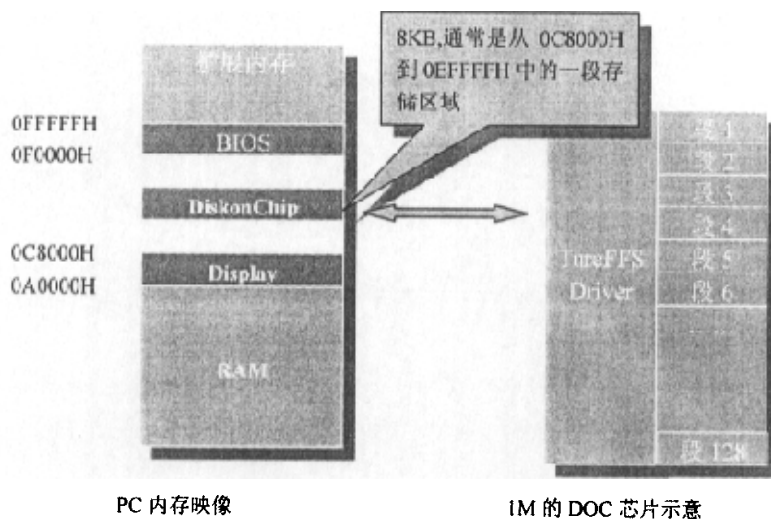


图 3-5 DiskonChip 的工作方式

图 3-5 中给除了一个具有 1M 存储空间的 DOC 工作方式的示意图。首先，TrueFFS 将它分成了 128 个段，每一个段的容量为 8k，然后在系统上电时将控制

信息和一部分数据信息复制到主存中为 DOC 预留的区域中。DOC 通过数据窗口的方式来管理主存中的数据和 DOC 中 flash memory 的数据交换,具体的做法是:将本身的 BIOS 作为系统扩展的 BIOS 挂入 INT13,用特殊的功能号来区分硬盘与 DOC,如果发生中断时需要的段在 DOC 上而不在主存中,则将它调入,并修改主存中相应的控制信息。要指出的是,DOC 上预先设计好的 TrueFFS 代码均为 MS-DOS 这样系统设计的,如果一个操作系统不使用系统 BIOS,该扩展的 BIOS 将没有任何意义。这种情况下需要自己去实现上述所有功能,这种做法是被 Linux 等系统广泛采用的,也是我们使用的方法。

### 3.2.2.3 DOC 的软件

在实现 DOC 的驱动程序的时候,也有象我们在硬盘驱动程序中提到的一些功能模块,驱动程序库中实现的函数包含在下面结构中:

```
private struct driver doc_dr_tab = {
    doc_name,      /*返回当前设备名*/
    doc_open,      /*安装设备*/
    doc_close,     /*释放设备*/
    doc_prepare,   /*准备 I/O 参数*/
    doc_rdwt,      /*执行 I/O 操作*/
    doc_geometry,  /*返回(伪)物理参数*/
};
```

以上的各个模块的功能在前面已经进行了详细的描述,这里要解释的是在函数中被广泛使用的两个十分重要的宏:

```
#define ReadDOC(adr, reg)*(((unsigned char *)adr) + DoC_##reg)
```

```
#define WriteDOC(d, adr, reg)*(((unsigned char *)adr) + DoC_##reg) = d)
```

所有对 DiskonChip 的读写都采用了这两个宏,其中 DoC\_##reg 这种表示方式是用 reg 所表示的字符(串)紧接在 DoC\_之后,例如:

```
WriteDOC(buf[di], docptr, 2k_CSDN_IO);
```

其含义就是:

```
DoC_2k_CSDN_IO+ docptr = buf[di] ://把 buf 中一个特定的字节写到 DOC 的
```

制定位置中去。

有了这两个宏以后，可以说 DOC 的大部分问题已经解决了，剩下的问题就是一些概念上的东西了，比如，向什么地址写入它的控制字、如何分段、读写等待延时等特性，这些是极其繁琐和枯燥的过程。针对 DOC，有数十行的此类定义，我们将它们全部放在了一个叫做 doc2000.h 的头文件中。

### 3.2.3 RAMDisk

#### 3.2.3.1 RAM 盘的硬件与设计思想

RAM 盘的设计思想很简单。块设备具备两个操作命令的存储介质：写数据块和读数据块。通常这些数据块存储与旋转存储设备上，例如硬盘和软盘。RAM 盘则简单的多，它使用预先分配的主存来存储数据。RAM 盘具有快速存取的优点（没有寻道和旋转延时），适用于存储需要频繁存取的程序和数据。

图 3-6 给出了 RAM 盘的实现思想。根据为 RAM 盘分配内存的大小，RAM 盘被分为  $n$  块，每块的大小和实际磁盘块的大小相同。当它的驱动程序收到一条读写一个数据块的请求时，它只计算被请求的块在 RAM 盘存储区的位置，并读出或写入该块，而不读写软盘或硬盘。数据的传输通过调用一个汇编语言过程来实现，该过程以硬件所能实现的最高速度把数据拷贝到用户程序或者从用户程序拷出。

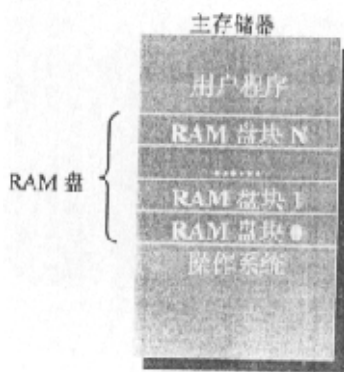


图 3-6 RAM 盘的实现方式

一个 RAM 盘驱动程序可以支持将存储器中若干区域当作 RAM 盘来使用，