



内附 AMI BIOS 源代码及完整工具程序

BIOS

研发技术剖析

陈文钦 著

- BIOS 基本概论与设置画面
- 最新硬件产业标准的讨论研究
- 知名厂商 AMI 美商安迈独家提供的 BIOS 源代码，从目录规划、内部模块到执行流程均有详细介绍
- BIOS 掌握的系统资源与诊断机制
- BIOS 移植的关键技术与相关源代码浏览
- BIOS 画面选项、默认值与特定图案的设计
- AMI BIOS 的烧写、组态、设置语言等工具程序的完整介绍
- 最精彩的硬件平台移植实战记录，让您有章可循，功力得到更大提高



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



BIOS 研发技术剖析

陈文钦 著

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2001-2275 号

本书繁体字版由旗标出版股份有限公司出版, 版权归旗标出版股份有限公司所有。本书简体字中文版由旗标出版股份有限公司授权清华大学出版社出版。专有出版权属清华大学出版社所有, 未经本书原版出版者和本书出版者的书面许可, 任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部。

内 容 简 介

本书是全球第一本针对新一代计算机平台的BIOS核心技术做深入剖析的技术性书籍, 内附完整的BIOS源代码以及相关的技术文献, 完备程度超过二十年前那本IBM BIOS Bible, 将这个业界研发技术缺口至少拉近了十八、九年, 所以本书的出现, 绝对可以说是难得的创举!

本书不仅着重于BIOS的结构剖析, 还指导读者如何制作及修改BIOS, 更难能可贵的是本书附赠了最重要的源程序代码, 让读者能实际动手做BIOS。本书主要针对具有基本汇编语言基础, 熟悉DOS操作指令且从事软件开发、编程的读者编写而成。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: BIOS 研发技术剖析

作 者: 陈文钦 著

责任编辑: 陈仕云

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 **印张:** 31.5 **字数:** 463 千字

版 次: 2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

书 号: ISBN 7-900635-75-0

印 数: 0001~5000

定 价: 49.00 元(附光盘)

JSB 11/03

从二十年前个人计算机 PC 的诞生, BIOS 就一直是最低层软件 / 硬件之间互动的最深层核心,也是每一代计算机平台开发时必须掌握的关键性技术。BIOS 的编写更是一门独门工夫,除了以最难写的汇编语言(Assembly Language)编写之外, BIOS 的开发必须同时具备支持最新的产业硬件规格,兼顾过去规格 (或者说包袱) 的兼容性,以及在成本要求下的程序代码最小化的极端要求,至今只有不到四五家软件公司,能够完全掌握这样关键性的技术。

对于 BIOS 的讨论,绝大多数仅止于 BIOS 设置值的介绍与讨论,顶多是小小开机画面 / 图案的小修改,以及将 BIOS 烧写到 ROM 的技术讨论;但是若要论及系统内部,去研究 BIOS 内部的执行流程、支持规格与数据结构等,那么市面上除了二十年前那一本 IBM8086/8088 BIOS——业界号称 BIOS 蓝皮书的中译本外,就没有任何书籍对此作过相关讨论。

因为这种深入系统核心一等的资料,甚至说程序代码开发的关键性技术,向来是只有 IBM、COMPAQ、宏碁、华硕等系统开发厂商才能接触到的机密资料,不是一般人士就能够一窥究竟的。

本书是全球第一本针对新一代计算机平台的 BIOS 核心技术做深入剖析的技术性书籍,而且本书肯定是全球第一本,内附完整 BIOS 源代码以及相关技术文献的书籍,完备程度超过二十年前那本 IBM BIOS Bible,将这个业界研发技术缺口至少拉近了十八、九年,所以本书的出现,绝对可以说是难得的创举,笔者绝对敢如此夸口!

本书的初步雏形,起自于某种因缘际会,本人与 BIOS 领导厂商——美商安迈(AMI,American Megatrends Inc.)公司高层人员的接触,该公司鉴于对这种深入系统核心的研究资料在市场上极为欠缺,对于计算机 / 微电脑

序

Preface

ROM 开发人才的训练与培养绝对是一项缺憾，因此有意将视为商业机密的关键性资料，寻求一个有汇编语言开发背景、对硬件规格了解的专业作者出书。

而笔者近年来较少从事软件的研发，而专注在媒体文章与专栏的发表，以及硬件组装书籍的写作，但十年前也曾经参与防毒软件的研发工作，对于汇编语言与 BIOS 内部流程、功能调用等知之甚详，亦或是兴趣、机缘或宿命，于是双方展开密集而漫长的技术合作。

笔者也晓得在当今软件写作的书籍，一窝蜂朝向网页设计，或较为入门的窗口语言写作，从事这种相对冷僻、艰深的书籍创作，真是一项吃力不讨好的工作，但笔者自许为技术入门的带领者，而 AMI 肯将视为商业机密的源代码关键技术完全公开，可见该公司直追 Linus Torvalds 开放 Linux 源代码的开阔胸襟，笔者能够何其荣幸的与该公司充分合作，协力完成本书，为 BIOS ROM 人才的培训贡献一点心力；而本人跟 AMI 高层共有的一个梦，一个让计算机人才掌握计算机系统核心级软件技术的美梦下，笔者与 AMI 愿意协力完成，当然也希望读者们能牵成，给予我们支持、鼓励！

本书定位在有基本汇编语言基础，还记得 DOS 操作指令且仍从事软件开发、编程的读者为主。但即便定位如此，本书仍然力求在章节规划、图表以及用字上力求平易与通顺，希望通过这些由浅而深的章节说明，让各位读者对 AMI BIOS 源代码的内部规划及移植的关键模块 / 源代码等都有了充分的了解；而许多从未在杂志、网络上现身的 AMI BIOS 的工具程序，其功能之强大，在本书详实的操作步骤介绍之下，即使是一般组装玩家，都可以自行尝试而游刃有余！

为了让读者能进一步地体验这种深入 BIOS 移植与开发的乐趣，我们特地准备了由 AMI 公司所授权的 BIOS 源代码光盘，这份 BIOS 源代码主要是

针对硅统科技 SiS530 芯片组结构下所研发的版本，可以移植到各种以 SiS530 整合芯片组(Socket7 结构，插 Intel Pentium MMX、AMD K6-2、K6-III 以及 Cyrix 6x86、6x86MX 等处理器)的主板。读者要花的成本除了本书之外，只要去买一张 SiS530 芯片组结构的主板、Socket 7 结构的 CPU，一切研发、移植的乐趣，只要自己亲自动手做过一遍，你就能体会！

本书所附的源代码光盘，也许不像 OEM/ 主板合作厂商所付费购买的 BIOS 源代码那样齐备，这是受限于 AMI 美国原厂审核、授权的附赠光盘内容所导致，毕竟有些 ASM 的细节仍然太过敏感，基于避免竞争对手参考或其他市场因素的考虑所致，不得略做些细微的“润饰”。但无论如何，本书所附的源代码光盘内容，绝对可以生成一份可完整烧写、执行的 SiS530 版本的 AMI BIOS，对于读者的实作绝对不会有任何影响，这点是毋庸置疑的，也是笔者跟 AMI 公司交涉过程中所坚持的原则。

最后，笔者也要在此感谢 AMI Taiwan 的 Bird，有了本书的原始构想，以及总经理 S.J Lin 先生的鼎力支持，也感谢吴存展(Den Wu)先生，这位因担任长期接触窗口而日久生情(友情)的新挚友；以及 Brown、Hugh、David、Chung、Pailin 等在写作技术上的全力支持，让我这个患有数年软件开发痴呆的老菜鸟，能够快速归位进入状况；硅统科技(SiS) 产品经理李志村先生，授权 SiS530 技术文献以及相关 Chipset 之 Porting Source，让本书落实一个有真实平台可实作、DIY 的梦想，而不仅仅只是纸上谈兵而已；最后旗标技术编辑志明的全程进度跟踪，在此也一一铭谢！

陈文钦

2001 年 2 月初

关于光盘

说明与需求

本光盘主要内容包含 AMI (American Megatrends Inc.)授权给本书读者参考的 BIOS 源代码(Source Code), 版本为 6.27.10 (简称 627.10)。

由于本 Open Source BIOS 源代码是针对采用 SiS530 整合芯片组为结构的主板而设计的, 所以当你准备钻研此源代码, 或编译自己的 BIOS 监控程序之前, 需要自行准备一些工具才行。在硬件部分, 请准备一片 SiS530 (不限品牌) 的主板, 与足够的 Flash 工具 (如烧写 BIOS 用的 FlashROM、拔 IC 专用的 IC 夹等等), 以作为研究的材料。

在软件部分, 还需要下列编译 / 工具程序文件:

文件名称	建议版本	用途说明
NMAKE.EXE	1.40	程序开发维护工具(保护模式可执行文件)
NMAKER.EXE	1.30	程序开发维护工具(保护模式可执行文件)
MASM.EXE	6.11	Macro Assembler 6.11 主编译程序
ML.EXE	6.11	Macro Assembler 6.11 后勤编译程序兼连结程序 (Linker)
LIB.EXE	3.20	Library Manager 链接库删减 / 编修工具程序
LINK.EXE	5.31	目标码(Obj) / 链接库连接器(Segmented Linker)

这些文件隶属于 Microsoft Assembler 6.11, 或者在 Visual Studio 6.0、Visual C++、MSDN(Microsoft Development Network) 等微软程序开发套件中可以找到。为了维护智能财产权, 这些编译程序请读者自行备妥。至于详情, 请看第 4 章的说明。

关于光盘

文件安装

安装方式非常简单，假设你光驱代号为 R:，只要在 DOS 下 (或者在 Windows 95/98 的 MS-DOS 窗口下)，输入：

```
XCOPY R:\627.10 C:\627.10 /S
```

就可以将所有 BIOS Source 拷到 C:\627.10。当然也可以在 Windows 资源管理器中用拖曳的方式将文件复制到 C:，但请记得事后用 ATTRIB -R C:\627.10 /S 的指令将所有文件的只读属性去除，以免编译时发生错误！

环境设置

大多数程序语言编译器都需要设置编译环境，本 BIOS 也不例外。这里的 BIOS 程序代码几乎是以汇编语言(Assembly)来设计的，须通过 MASM 6.11 来编译。编译时请记得设置以下的环境变量，假设 MASM 6.11 是安装到 C:\MASM611，则需要作以下的设定：

```
set include=c:\masm611\include\  
set helpfiles=c:\masm611\help\*.hlp  
set init=c:\masm611\init\  
set asmex=c:\masm611\samples\  
set tmp=c:\masm611\tmp  
set demo=c:\masm611\demo
```

在 627.10\EXE 目录下有个简单的批处理文件 Setpath.bat，只要执行过一次，就会设置好上述相关的路径，你也可以在 AUTOEXEC.BAT 加入执行 setpath.bat 的叙述。

其他

至于其他事项、内容，请参照本书说明。

关于光盘

目录结构

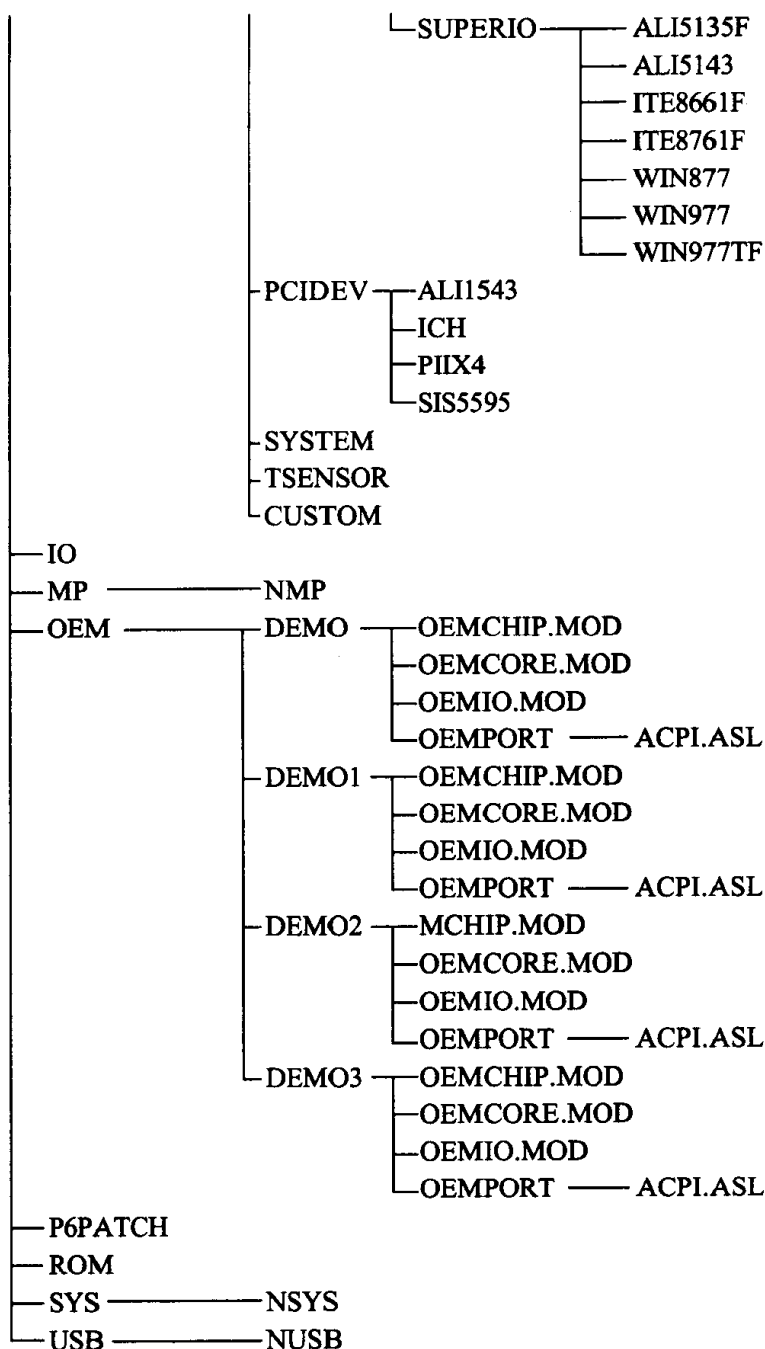
本光盘的目录结构如下:

```

627.10 — ADD-ON.ROM
      — ASM1
      — ASM2
      — BOOTINI
      — CPUCORE
      — EXE
      — INCLUDE
      — IO.112 — ALI5105
                  — ALI5113
                  — ALI5119
                  — ALI5123
                  — ALI5135
                  — ALI513X
                  — IT8661
                  — IT8671
                  — IT8679
                  — IT8680
                  — SYSTEM
                  — UMC8663B
                  — UMC8669
                  — WIN627
                  — WIN787
                  — WIN787IF
                  — WIN877
                  — WIN877AF
                  — WIN977AF
      — OBJ
      — P6UCODE
      — SIS530 — ACPI
                  — CORE.MOD
                  — CPU
                  — DIM — NDIM
                  — GPC — NGPC
                  — INC
                  — INCLUDE.ASL — OEMIO — AUDIO — ESS1869
                                      — ESS1878
                                      — ESS1887
                                      — ESS1888

```

关于光盘



以上各目录的用途，在本书第 4 章都有详细的介绍，请自行参考本书内的说明。

第 1 章 BIOS 基本概论与设置

- 1-1 BIOS 的定义与解释 2
- 1-2 何谓 BIOS 设置、如何更改 CMOS 设置 8
- 1-3 BIOS 市场风云录 15
- 1-4 简介美国安迈(AMI,American Megatrends Inc.)公司 21

第 2 章 硬件基本概论与规格导读

- 2-1 计算机处理器、主板、芯片组的介绍 30
- 2-2 主板零组件、总线等特定规格介绍 44
- 2-3 跟电源管理有关的规格与术语解释 58

第 3 章 BIOS 开发与硬件之间的互动关系

- 3-1 CPU 历代演变与 BIOS 的关系 70
- 3-2 x86CPU 内存结构与 ROM 大小的演进 74
- 3-3 BIOS 的诊断码 (Post Code) 简介 87
- 3-4 BIOS 所要掌握的系统资源 96
- 3-5 CMOS 的索引数据 103

第 4 章 AMI 源代码目录结构与文件维护

- 4-1 AMI BIOS 的源代码版本缘由 114
- 4-2 AMI BIOS 的源代码目录设计 115
- 4-3 编译 BIOS 所需要的编译 / 链接程序 125
- 4-4 BIOS 的编译 / 维护作业的相关文件之一 (IO.MAK) 129
- 4-5 BIOS 的编译 / 维护作业的相关文件之二
——供 OEM 厂商自定规格 / 功能等设置值的维护
文件 OEM.MAK 138

目录 Contents

第5章 BIOS 的内部模块与执行流程介绍

- 5-1 BIOS 内部模块的说明 160
- 5-2 BIOS 执行流程图 165
- 5-3 BIOS 程序源代码浏览(一)
 - BootBlock(激活区块) 的相关模块.....180
- 5-4 BIOS 程序源代码浏览(二)
 - Flashpart(快速存取存储器) 相关程序代码分析.....202
- 5-5 BIOS 程序源代码浏览(三)
 - 其他在 POST 阶段参考到的程序代码分析.....212

第6章 BIOS 平台移植的关键技术——

特定平台修改与关键模块 / 源代码的介绍

- 6-1 掌握芯片组、外围 I/O 芯片的相关规格 226
- 6-2 移植到特定芯片组要修改的源代码 240
- 6-3 内存容量检测 / 计算 247
- 6-4 PCI IRQ 号码配置与绕线设置 279

第7章 BIOS 画面选项、默认值设计

- 7-1 AMI BIOS 的开机设置画面与选项简介 294
- 7-2 以 AMIBCP 调整开机默认值 / 隐蔽项目 301
- 7-3 AMI BIOS 的 Setup Script (.SSP) 语言的介绍 306
- 7-4 直接取用源代码文件群的.SSP 来修改 338

第 8 章 AMI BIOS 工具程序——ROM 升级与编辑工具

- 8-1 AMI BIOS 基本工具程序列表..... 342
- 8-2 先从烧写 BIOS 开始
——强大的 AMIFLASH ROM 烧写程序..... 346
- 8-3 BIOS 组态魔术师——AMIBCP.EXE 之一..... 369
- 8-4 BIOS 组态魔术师——AMIBCP.EXE 之二..... 380

第 9 章 AMI BIOS 工具程序——研发 / 维护工具

- 9-1 BIOS 组态魔术师——AMIBCP.EXE 之三..... 388
- 9-2 AMIMM.EXE BIOS 模块操作 / 修改程序..... 398
- 9-3 AMIMM.EXE 的描述 / 操作语言介绍..... 400
- 9-4 RU.EXE 硬件系统寄存器检视程序..... 407
- 9-5 其他研发工程师 / 高等玩家的工具程序..... 417

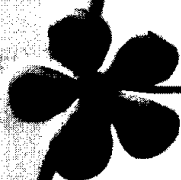
第 10 章 硬件移植实战 DIY

- 10-1 简介本书所示范的主板与其他配备..... 431
- 10-2 建立必要的目录、相关文件与路径..... 437
- 10-3 开始进行 Porting 作业..... 441
- 10-4 修改开机的设置界面、商标图案与全屏幕画面..... 456
- 10-5 BIOS 开发的错误解决与明日应用..... 476

- 附录 A 483

CHAPTER

1



BIOS基本概论与设置

1-1 BIOS 的定义与解释

大约在二十年(1981)前,当时被信息界称为蓝色巨人的 IBM,在研究第一部个人计算机——IBM PC 时,他们的工程师把一些开机时的硬件启动 / 检测码 (Initial Code),从软盘或硬盘加载到操作系统,将开机程序的前导程序代码,以及一些最基本的外围 I/O 处理的子程序码(如屏幕显示、磁盘驱动器驱动、摇杆控制等),通通挤入在一块大约 32KB 大小的 PROM (Programmable ROM,可编程只读存储器)中。这个程序代码就叫做 BIOS (Basic Input/Output System),如图 1-1 所示计算机主板中的 BIOS 芯片。

当你的计算机从开机的一瞬间,硬件特性就由 CPU 从主板的 BIOS 芯片内取得程序代码,用 BIOS 内部的程序代码获得控制权并且发挥作用,从 CPU 内外部的检测设置、激活 DRAM 以及针对芯片组与各种外围设备作初始化设置之后,最后驱动软盘或硬盘,直到把操作系统(如 DOS, Win98/NT, Linux)加载成功, BIOS 的开机引导工作就此告一段落,转向从事幕后的支持、协调工作,并帮助操作系统或应用程序,来处理外围设备沟通的细部操作。

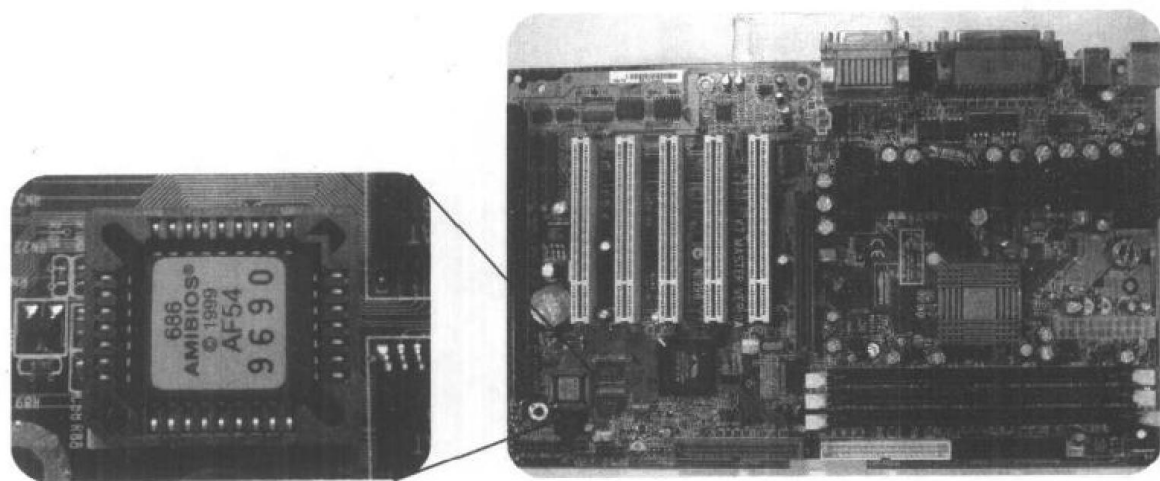


图1-1 唤醒整个计算机系统的 BIOS 芯片

事实上除了主板上有 BIOS 之外，另外像显卡、SCSI 控制卡、磁盘附加卡或高级网卡也都有自己的 BIOS 芯片。这些设计在主板上、负责整个主板运行的 BIOS，我们一般称之为 System BIOS 或 Mainboard BIOS；显卡上的 BIOS 多半以 Video BIOS 称呼，SCSI 控制卡的 BIOS 则以 SCSI BIOS 称呼。

由于一般人所接触的计算机系统，最起码有一组 Mainboard BIOS，而且有些主板已经整合了显卡、SCSI 接口，因此 Mainboard BIOS 已经涵盖了以往 Video BIOS 与 SCSI BIOS 等监控程序。为了方便本书的阅读，如果没有特别声明的话，BIOS 就是指主板上的 BIOS。

如果要更清晰地定义什么是 BIOS, 在一家 BIOS 研发厂商的技术培训文件中, 对于 BIOS 的叙述, 他们是这么说明的:

A Firmware program (一种属于 ROM 的程序代码)，存储在 ROM、EPROM 或 FlashROM 内存中，提供最基本的硬件初始化 (initialize)、外围控制的必要程序代码。通常是用汇编语言 (Assembly Language) 编写的。

BIOS 平时设置为只读的状态,除了特殊工具程序或特殊方式之外,任何应用程序都没办法对它进行修改或删除。以前的修改方式比较复杂,如果是 Mask-ROM 或 PROM,几乎得重新换一块空白的 ROM 芯片,然后烧写。但到了 EPROM 时代,可以用紫外线照射其窗口一段时间,在重新擦除其内部的所有数据之后,再用 EPROM 烧写器将 BIOS 数据烧写到 EPROM 中。

当 Windows 95 流行之后, 由于 Plug and Play (PnP) 与 ACPI 以及 DMI 协议的需要, 主板中需要有一个可以记录外围设备资源的异动情况, 却又不会被其他程序所破坏的内存区域, 而 FlashROM 对于 ROM 程序代码的更新比 EPROM 还要方便。所以到后来, 从主板 BIOS, 以至于显卡、各外围接口卡存储 BIOS 的内存芯片, 几乎都是 Flash ROM 的天下。

一般在设计计算机或外围设备时, 存储 BIOS 程序代码所使用的只读存储器 (ROM) 种类, 大致可以分为下列几种:

① PROM (Programmable ROM) 可编程存储器

PROM 出厂时内部每一字节的数据都是 \$FF (也就是每个位都为 1), 且从未烧写过, 可供客户用烧写器以特殊电压将数据写入 (把某个位写入 0, 就相当于把代表 1 的线路给烧断)。但是也因为这种特性, PROM 只能写入一次数据, 日后要再次修正数据时, 必须重新购买一块新的空白 PROM 来烧写。

② Mask ROM

它是由客户指定的数量较大的, 由内存生产厂家依客户的指定内容, 在大量生产的过程中直接将数据写入, 出厂之后就已经有特定的程序 / 数据码, 内容也无法自行修改或烧写。通常在大量生产下, MaskROM 的容量比较大, 因此成本也比较低, 这也就是早期许多汉卡字形、甚至大型电子游戏机喜欢采用 MaskROM 的原因。

③ EPROM (Erasable PROM)

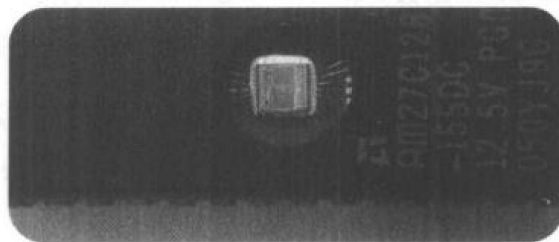


图1-2 EPROM 有个窗口

接下页