

RENESAS M16C/62

嵌入式微控制器程序设计

黄一夫 黄立 邱邦能 编著

RENESAS M16C/62

嵌入式微控制器程序设计

黄一夫 黄立 邱邦能 编著

华中科技大学出版社

图书在版编目(CIP)数据

RENESS M16C/62 嵌入式微控制器程序设计/黄一夫 黄立 邱邦能 编著
武汉:华中科技大学出版社,2004 年 10 月
ISBN 7-5609-3194-4

I. R…
II. ①黄… ②黄… ③邱…
III. 控制器-程序设计
IV. TP332.3

RENESAS M16C/62
嵌入式微控制器程序设计

黄一夫 黄立 邱邦能 编著

责任编辑:叶见欣

封面设计:潘 群

责任校对:吴 晗

责任监印:张正林

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉正佳文化发展有限公司

印 刷:湖北新华印务有限公司

开本:787×1092 1/16

印张:11.75

字数:266 000

版次:2004 年 10 月第 1 版

印次:2004 年 10 月第 1 次印刷

定价:22.80 元

ISBN 7-5609-3194-4/TP·528

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

瑞萨(RENESAS)公司生产的嵌入式微控制器 M16C/62, 是近年来获得广泛应用的一种功能增强型 16 位新型微控制器, 它广泛应用于国民经济各部门和日常生活中。

本书以 M16C/62 学习板为基础, 系统地介绍了 M16C/62 的程序设计技术。本书共分三篇分别介绍了 M16C/62 的程序设计基础知识, 集成开发环境和程序设计技术, 并举出范例对用 M16C/62 组成的控制系统加以说明。

本书采用大量图形加以说明, 便于理解和学习。

本书可作为高等院校相关专业学习嵌入式微控制器程序设计课程的教材, 也可作为从事微控制器研究、开发和应用的技术人员的培训教材和参考书。

序

瑞萨科技深信,学习半导体在电子产业领域的应用,特别是成为电子控制枢纽的微控制器 MCU 的应用,并不断积累经验对我们是极其重要的。

特别值得介绍的是 M16C,该产品消耗电力低,在高温状态下也能够运行,具有对抗噪声的特性,卓越的 ROM 程序效率,管脚(Pin)互换性,丰富的位处理指令及拥有适用于实时控制的指令群,是瑞萨科技富有代表性的高性能多功能的自信产品。

希望有机会阅览本书的广大读者,能进一步积累更接近实际的经验,瑞萨科技期待着为中国的电子产业的发展贡献一份光和热。

在此,请允许我感谢华中科技大学黄一夫教授等编著了本书,并感谢众多人士,为本书的编辑制作所提供的积极协助。另外,对出版本书的华中科技大学出版社也表示深深的谢意。

瑞萨科技股份有限公司 通用产品统括本部 MCU 事业部
MCU 产品技术部 部长

松井秀夫

2004 年 8 月

前 言

瑞萨 (RENESAS) 公司是 2003 年 4 月由著名半导体厂商——日本日立制作所、日本三菱电机公司的半导体部门合并而成的一家新的半导体公司。

瑞萨公司生产的嵌入式微控制器 M16C/62, 是由原日本三菱电机公司半导体部门开发, 近年来获得广泛应用的一种功能增强型 16 位新型微控制器。

由于嵌入式微控制器 M16C/62 具有高速处理、内置多种功能模块、功耗低、极强的抗干扰能力以及很高的编程效率和性能价格比等特点, 它广泛应用于工业、交通、通信、商业、银行、服务行业、医疗、学校与行政办公等各个领域。

本书以嵌入式微控制器 M16C/62 学习板为基础, 系统地介绍了用嵌入式微控制器 M16C/62 进行程序设计的有关基础、集成开发环境和程序设计技术。

本书共分三篇。

第 I 篇介绍嵌入式微控制器 M16C/62 学习板的基本组成, 程序设计步骤, 控制对象的特性, 用汇编语言和 C 语言编程的基础知识等, 并举出几个练习来对程序设计进行说明。本篇是学习用嵌入式微控制器 M16C/62 进行程序设计的基础。

第 II 篇介绍嵌入式微控制器 M16C/62 的集成开发环境 TM (TOOL Manager)。本篇分别叙述了集成开发环境 TM 的启动, 各工具 (编译、编辑、调试、闪存写入) 的登录、新项目制作、调试程序的启动、程序写入等过程。

第 III 篇介绍嵌入式微控制器 M16C/62 学习板的程序设计技术。本篇在对启动程序、预处理、基本程序、中断程序等叙述的基础上利用前两篇的基础对液晶显示器模块、开关电路等范例的程序设计进行了说明, 并列出了程序清单, 供学习者学习。

本书有 CD-ROM, 在其中装有用户手册、工具管理器 TM、编译程序、调试程序、闪存写入程序、数据单、产品目录等, 供学习者进行程序设计时参考。需要者, 请与作者联系。联系电子邮件地址为: gter11@hotmail.com。

本书可作为高等院校相关专业学习嵌入式微控制器程序设计的教材, 也可作为从事微控制器设计、开发和应用的专业技术人员的培训教材和参考书。

本书由黄一夫、黄立、邱邦能编写。

在编写过程中, 日本专门教育研究所、日本工业大学特聘教师汤本兵吾先生提供了大量参考资料; 日本工业大学情报工学科片山滋友教授提出了许多宝贵意见; 华中科技大学水电学院研究生姜哲在紧张的学习和研究中抽出时间, 为本书的图文移植和文稿整理做了大量的工作; 该院青年教师任波、研究生李嘉、邓雪华、陈鹏、冀旭钢、张峰等也参加了本书的图文移植的工作; 华中科技大学副校长王秉教授自始至终关心和支持本书的出版; 华中科技大学出版社总编辑张峰教授在人力、物力方面为本书的出版提供条件。在此一并表示感谢。

由于作者水平有限, 对于新型嵌入式微控制器的理解不够深刻, 加之时间仓促, 书中有不当之处, 欢迎读者和专家批评指正。

作者

2004 年 7 月

目 录

第 I 篇 程序设计基础

第 1 章 微机概述	(3)
1.1 微机的基本组成	(3)
1.2 微机的基本操作	(5)
1.3 微机中的数据和语言	(6)
第 2 章 M16C/62 学习板	(10)
2.1 CPU	(10)
2.2 内置周边功能	(11)
2.3 存储器配置	(12)
2.4 复位	(12)
2.5 M16C/62 的基本操作	(13)
第 3 章 使用 M16C/62 学习板开发程序	(14)
3.1 M16C/62 学习板的开发步骤	(14)
3.2 存储器配置	(15)
第 4 章 M16C/62 学习板的程序设计	(18)
4.1 范例 A 的规格	(18)
4.2 子程序	(20)
4.3 汇编语言的基础知识	(21)
4.4 C 语言程序的基础知识	(30)
4.5 存储器配置的基础知识	(34)
第 5 章 I/O 控制的练习	(36)
5.1 LED	(36)
5.2 开关	(38)

第 II 篇 集成开发环境 TM

第 6 章 集成开发环境 TM	(45)
6.1 运行环境	(45)
6.2 TM 的安装	(45)
第 7 章 集成开发环境 TM 的启动	(46)
7.1 TM 的启动	(46)
7.2 项目栏各按钮	(46)
7.3 项目栏的整理	(47)

第 8 章 工具的登录	(49)
8.1 调试器(kd30)的登录	(49)
8.2 编辑器的登录	(51)
8.3 闪存 ROM 写入工具(Flashstart)的登录	(54)
第 9 章 make	(58)
9.1 make 的思考方法	(58)
9.2 make 的编写方法	(59)
第 10 章 制作项目	(60)
10.1 项目	(60)
10.2 制作基本项目	(61)
10.3 项目编辑	(63)
10.4 文件的添加	(66)
10.5 命令	(69)
10.6 摩托罗拉 S 格式的文件的制	(70)
10.7 任选项的指定	(72)
10.8 依赖关系的更新	(75)
第 11 章 build	(76)
11.1 build 的种类	(76)
11.2 build 的启动	(76)
11.3 build 内编译程序发生错误(Error)时的处理	(77)
第 12 章 调试器的启动	(78)
12.1 调试器的启动	(78)
12.2 下载程序	(79)
12.3 程序运行	(80)
第 13 章 写入程序	(82)

第Ⅲ篇 程序设计

第 14 章 C 语言的基础知识	(91)
14.1 C 语言	(91)
14.2 嵌入用 C 语言	(91)
14.3 数据类型	(92)
14.4 运算符	(93)
14.5 控制语句	(94)
14.6 函数	(96)
14.7 数组	(98)
14.8 结构体	(101)
14.9 共用体	(103)
第 15 章 用 M16C/62 学习板使用 C 语言进行开发之前的工作	(104)
15.1 启动程序	(106)

15.2 预处理器的处理 (117)

第 16 章 使用 M16C/62 学习板进行程序设计 (124)

16.1 基本程序 (124)

16.2 开关电路 (130)

16.3 中断程序 (137)

16.4 LCD 模块的控制 (152)

附录 1 M16C/62 学习板电路图 (172)

附录 2 M16C/62 学习板中 CPU 板电路图 (173)

第 I 篇 程序设计基础

第 1 章 微机概述

在使用 M16C/62 开发嵌入式系统之前,先对微机进行简单的说明。微机是微型计算机的简称,使用微机的目的在于进行输入/输出的控制。为了进行控制,微机系统应具有硬件(H/W)和软件(S/W)两个要素,如图 1-1 所示。

硬件是指微机的机器部分,包括连接的输入/输出装置,如开关、LED(发光二极管)、电阻、电子元件、电源、地线等。

软件是指使微机工作的程序。微机读入程序,依照命令进行工作。输入/输出控制程序应在微机系统中预先置入。

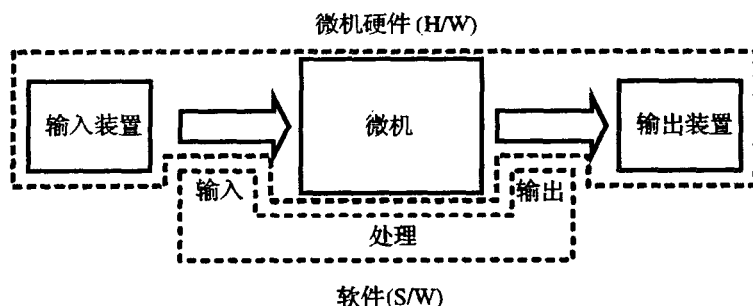
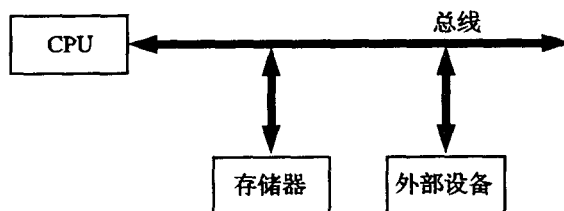


图 1-1

1.1 微机的基本组成

微机通常由 CPU(Central Processing Unit,称为中央处理器)、存储器(记忆装置)、外部设备组成。称为总线(Bus)的信号线群将它们连接起来,如图 1-2 所示。



< 微机的结构 >

图 1-2

外部设备通常由称为外围接口(Peripheral Interface)的电路单元通过总线与微机相连。外部设备因为可分为输入设备和输出设备,所以外部接口也分为输入接口和输出接口,二者一起称为 I/O 接口,如图 1-3 所示。

CPU、存储器、I/O 接口是构成微机的三要素。

CPU 是微机的大脑部分。能对写入存储器里的程序顺次读出,运算后进行 I/O 控制,其

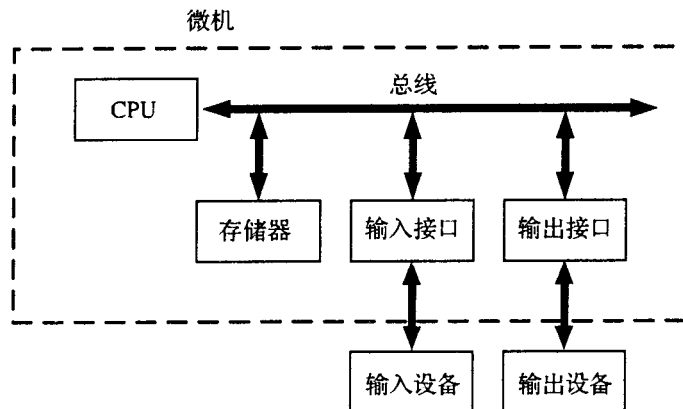


图 1-3

中设有执行过程中必要的寄存器和计数器。

存储器(记忆装置)是将程序和数据以数字形式写入、保存的部件。存储器可分为 ROM (Read Only Memory) 和 RAM(Random Access Memory) 两类。

ROM 是专供 CPU 读出的存储器。因为即使断了电源,ROM 中的数据也能保存,所以通常称为只读存储器,用于预先写入程序。ROM 有工厂出厂时写入程序的掩膜 ROM;多次可更换内容的 EPROM;电气可擦写的 EEPROM;以程序块为单位写入的闪存等。

RAM 是能读能写的存储器,断电时,数据不能保存,因此程序不能预先写入,通常称为随机存储器,用于保存程序执行过程中的数据和程序状态。RAM 有静态读/写存储器 SRAM;动态读/写存储器 DRAM;嵌入式存储器等。

I/O 接口是将微机与外部设备相连接的电路。

将微机的组成要素装在一块印刷板上的称为单板机,装在一个芯片(IC 集成电路)上的称为单片机,如图 1-4 所示。

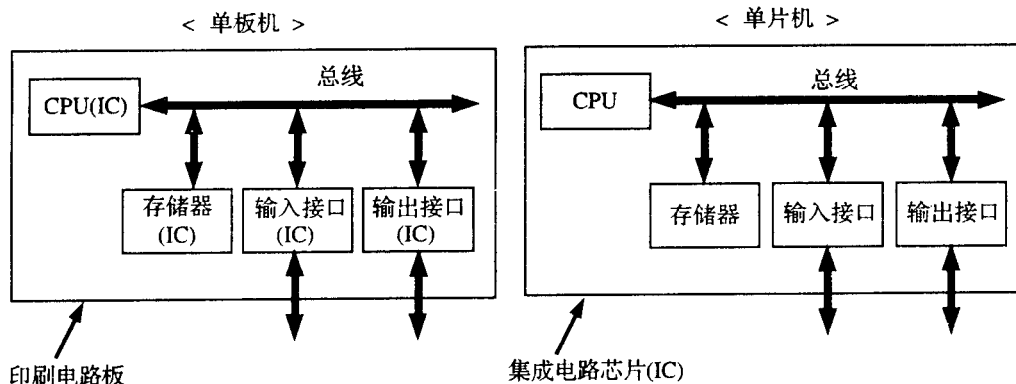


图 1-4

微机的各组成要素由总线相连,进行信息的传输。总线按功能可分为地址总线、数据总线和控制总线等。

地址总线:它传输 CPU 发出的地址,以便选中 CPU 所寻找的存储单元或外部设备。

数据总线:由于数据在 CPU 与存储器和 CPU 与 I/O 接口之间的传输是双向的,数据总线为双向总线。

控制总线:它传输各种控制信息。这些控制信号有的是由 CPU 到存储器或 I/O 接口的,如读信号 RD(Read 信号),写信号 WR(Write 信号),地址锁存允许信号 ALE(Address Latch Enable),中断响应信号 INTA(Interrupt Acknowledge)等,有的是由 I/O 接口到 CPU 的,如可屏蔽中断请求信号 INTR,准备就绪信号 READY 等,如图 1-5 所示。

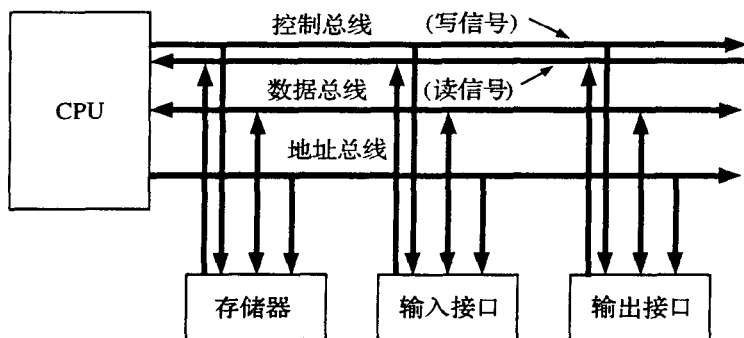


图 1-5

微机中除了上述基本构成要素外,尚有时钟发生电路和复位电路,如图 1-6 所示。

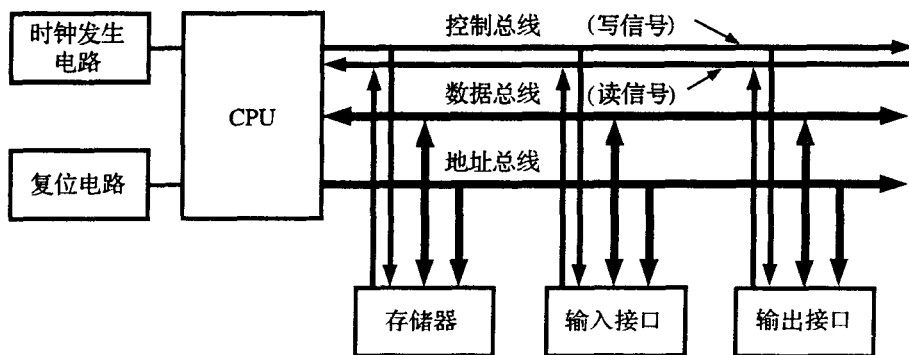


图 1-6

时钟发生电路:以一定的周期产生“H”和“L”信号作为操作的基准。

复位电路:使微机获得起始设定的状态。

1.2 微机的基本操作

微机将写入存储器的程序,依次读入加以执行。操作的内容是在解除复位后周而复始地按读命令、分析命令、执行命令的顺序进行,直到完成全部指令操作为止。

其基本操作步骤如下:

- ① 得到复位信号后,解除复位;
- ② 把程序计数器的值,输出到地址总线;
- ③ 从控制总线读 RD 信号;
- ④ 把写入程序的命令,从数据总线输入到 CPU;

- ⑤ 读命令；
- ⑥ 执行命令。

其基本操作如图 1-7 所示。

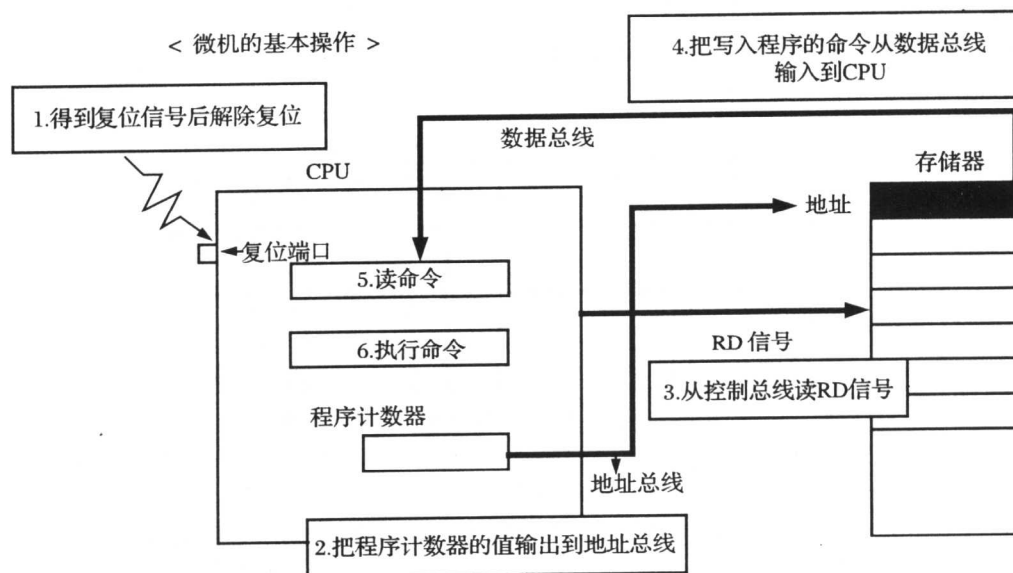


图 1-7

其中：

- 复位 在一定的时间后,复位被解除,微机开始操作。
- 读命令(Fetch) CPU 将程序计数器的内容从地址总线读出,发出读信号后,从数据总线上输入命令。
- 分析命令(Decode) 在 CPU 内部,分析命令。
- 执行命令(Executor) 执行分析了的命令。

1.3 微机中的数据和语言

微机中所有的信息是以“H”和“L”两个信号进行表达的。换言之,微机能理解的信息只有两类。这两类信息使用多条信号线,利用存储器和 I/O 进行处理。将“H”和“L”分别以“1”和“0”加以置换,则容易理解。“1”和“0”表示的单位,称为位(bit)。位是微机所能表示出的最小数据单元。因为微机采用二进制数,将若干个二进制数进行组合,就可表示各种数据。位表示指数的值,如 2 位表示 2^2 ,有 4 种信息,4 位则表示 2^4 ,即 16 种信息,8 位则表示 2^8 ,可表示 256 种信息,如图 1-8 所示。

这种方法可表现微机的程序、数据等所有的信息。

1.3.1 表示数的数据

我们日常使用的数字,是用从 0 到 9 的数来表示的。达到 10 后则进 1 位,这称之为十进制。微机使用的数,仅为“1”和“0”,达到 2,则进 1 位,这称之为二进制。但是二进制数表示数时,书写太长,这时可采用十六进制,数字从 0 到 9 后用英文字母的 A、B、C、D、E、F 进行表示,

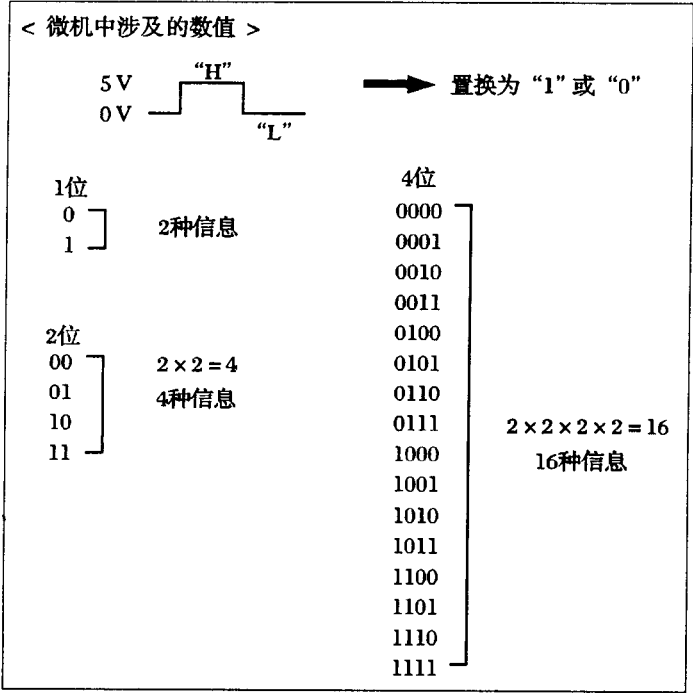


图 1-8

如表 1-1 所示。

表 1-1

二进制	十六进制	十进制
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

为了区别二进制数、十六进制数、十进制数，常使用下述符号来表示不同的进制数。表示二进制数的符号为“B”或“%”，表示十六进制数的符号为“H”或“0x”。

例如：01010101B、%01010101 均为二进制数；

55H、0x55 均为十六进制数。

人们习惯用十进制数，但微机中常常用 4 位二进制数表示十进制数，简称为 BCD 码。

较常用的有 8421 BCD 码等,如表 1-2 所示。

表 1-2 BCD 码

BCD 码	十六进制数	十进制数
00000000	0	0
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00010000	A	10

1.3.2 程序使用的数据(语言)

在写程序时若使用的是机器语言,则即使把二进制数转换成十六进制数也仍然难以理解,为此考虑将命令用助记符(Mnemonic)进行置换。用助记符置换后的语言,称为汇编语言。

汇编语言的问题在于:使用的 CPU 不同,助记符也不同,因此程序没有互换性。不同 CPU 均能共同使用的是程序语言,其中之一就是 C 语言。C 语言可用比较简单的记述方式来编程序,能记述程序的内容。

图 1-9 所示的是使用各种语言的记述方法和特点。

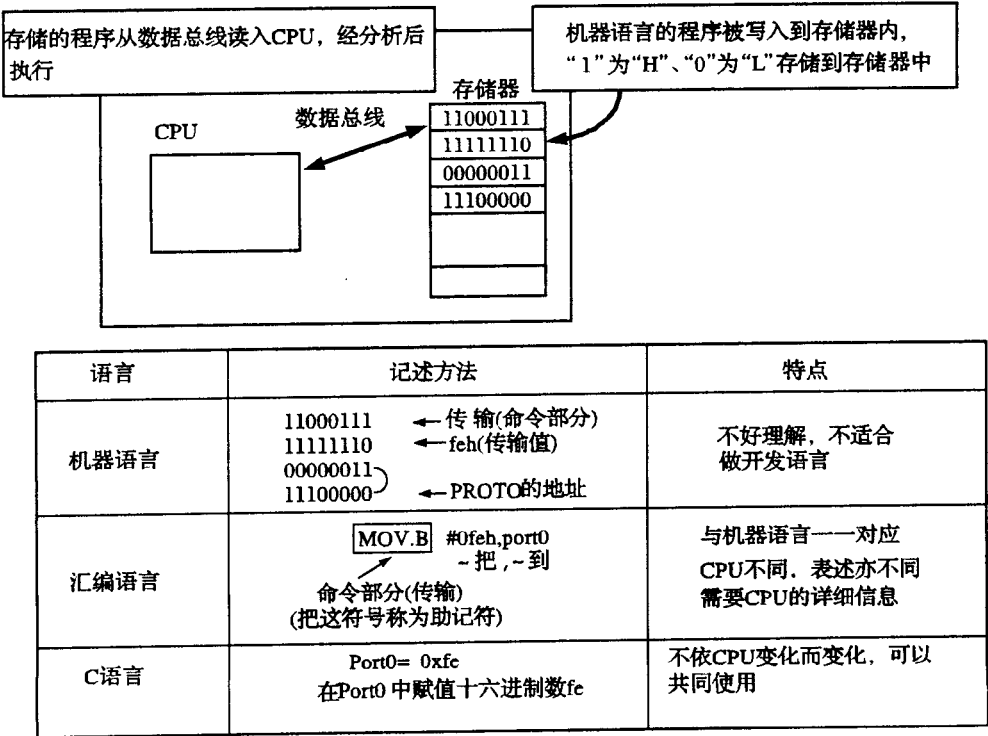


图 1-9