

微型计算机软件技术

WEIXING JISUANJI RUANJIAN JISHU

冶金工业出版社

微型计算机软件技术

〔日〕 吉田征夫 著

王洪晏 译

冶金工业出版社

内 容 提 要

本书是介绍微型计算机软件技术的专著。书中系统地介绍了微型计算机编程技术的基本知识和技巧,探讨了微型计算机的标准程序设计。

全书共三篇十九章。第一篇微型计算机基础主要介绍学习编程技术的方法。第二篇指令及其功能介绍指令系统,详细说明各类指令的功能。第三篇程序设计技巧为全书核心部分,按照指令的类型,详细论述程序设计的方法和技巧,列举了大量的标准程序结构,并对微型计算机程序设计标准化问题提出见解。

本书可供工程技术人员系统学习编程技术用,对现场及实验室从事仪器仪表、自动控制技术的技术人员尤为适用,亦可供大专院校有关专业师生参考。

マイクロコンピュータ ソフトウェア技術

著者 吉田征夫

昭和54年1月30日 初版発行

* * *

微型计算机软件技术

(日) 吉田征夫 著

王洪晏 译

*

冶金工业出版社出版

(北京北阿裕大街嵩祝院北巷33号)

新华书店北京发行所发行

冶金工业出版社印刷厂印刷

*

787×1092 1/16 印张14 1/4插页1 字数336千字

1988年2月第一版 1988年2月第一次印刷

印数00,001~6,600册

ISBN7-5024-0247-0

TP·10 定价3.45元

译者的话

本书系根据日本マイテシク出版公司1979年出版、吉田征夫所著《マイクロコンピュータソフトウェア技術》一书翻译的。为了精简篇幅和避免与其它有关出版物的内容重复，将原书“第Ⅰ部”的“第2、3”两章删去。翻译本书的目的是为给广大科技人员、有关专业的师生有效地学习利用微型计算机提供参考资料。

在我国，随着微型机应用的普及，微型机技术的重点逐渐转向软件方面。广大用户对编程技术的普及和提高，要求越来越迫切。本书从内容到编写方法都适用于工程技术人员系统地自学编程技术，特别是对生产现场及研究室从事仪器仪表、自动控制技术的人员尤为适用。

在本书的翻译过程中，曾与安秋顺同志多次进行有益的讨论，并承蒙阅稿；李雨膏同志在百忙中帮助阅稿，提出许多修改意见，这里一并致谢。

译者水平有限，译文难免有不妥之处，望读者指正。

王 洪 晏

1983.10.27.

序

随着微型计算机的出现，计算机以及带有计算机的装置，可以相当便宜地买到了。智能元件或智能系统可以说已达到了惊人的普及程度。那么，与过去的计算机相比，微型机的使用是否变得更容易了呢？回答是否定的。微型机在使用上的难度与过去的计算机完全一样。也就是说，与过去一样，程序设计这件棘手的事依然要由用户来做。

微型计算机的程序设计，是学习微型机时首先碰到的难关。大多数学习者是手持微处理器使用说明书，看着其动作及指令系统表；然后，面对自己要设计的实际系统与使用说明书之间的差距，感到茫然不知所措。对于从事硬件工作的人来说，程序、软件等的概念确实会感到不易理解。为了某一目的，将指令语句编成程序是颇不易的，靠读程序例子也难以弄清其设计思想。

在微型机应用技术逐渐转移到软件方面的今天，不论何人在探讨汇编程序语言、高级语言的程序设计技术之前，都必须经过一番研究详细论述程序概念和编程技术的基础著述，这是理所当然的。

挚友吉田征夫编写的这本书正是一本能很好地满足上述愿望的好书。此书，第一篇、第二篇为微型机的基本概念和基本指令系统；第三篇则对第一、二篇所述知识的使用方法和技巧作了叙述。本书内容新颖，更令人高兴的是其记述方式、编写方法也颇有特色。

谨向将要从事微型机以及已经从事微型机工作的读者推荐此书。

日本INTEL公司 理学博士 鎌田信夫

1978年12月

前 言

随着集成电路技术的提高,微型机变得使用方便,价格低廉了。但这只是就硬件方面而言。而在软件方面,微型机动作的程序却日趋复杂。为此,要求有高超的编程技巧。

微型机的指令随机种而异,指令的功能也越来越带有浓厚的硬件色彩。这也是难以确立通用的程序设计方法的原因。

过去出版的大量微型机方面的书多限于一般性叙述,有关程序方面的内容,似乎都不过是一些对零星事例的论述。

对于初学者和有经验的广大读者来说,都需要有一本从中能够系统地学到程序技术的书。

从目前的情况来看,在8位机方面以8080A/8085A为首的两、三个机种已相继成为标准机种,确立程序的标准编制方法的时机已经成熟。以前对初学者来说,程序技术的学习处于一种摸索状态;尽管如此,大多还是通过实际接触微型机来学会编程技术的。现在,不仅需要实际接触微型机,而且还需要更加系统深入地学习编程的基础技术知识。

本书粗分为由下列三部分组成。

第一篇为微型计算机基础,是为初学者写的。

第二篇为微型计算机的指令篇,集录了8080A/8085A的指令,并对各个指令功能进行了说明。对于已具有相当水平的读者,可以跳过第一篇和第二篇,以从第三篇开始学习为好。

第三篇是本书的核心,内容是从两个方面来论述程序设计的技巧。一方面是与微型机的运算功能密切相关的基本程序设计技巧,另一方面是从处理方法的角度来看,程序如何构成为好。

过去把着眼点放在微型机的结构上去了。本来,所谓程序就是处理的记述。对如何解决程序设计中的关键性问题,过去做得不够。

本书中引入了高级语言编程技术的思考方法,进行处理的办法也分成了几类,这样,把指令组合起来编制程序时,就可以根据不同处理方法的标准程序设计技巧来进行。

对于上述这些,书中做了非常系统的解说,本书对初学程序技术的人来说,是部好教科书,对于有经验的人则能起到使知识条理化的作用,同时还可作为程序集供参考与使用。

如果掌握了本书的内容,就可以达到中等以上的微型机程序技术水平,能够自如地使用微型机。

本书在编写过程中得到了很多人的帮助。

本书能问世的直接支持者鎌田信夫氏、野尻孝氏在百忙中提供了许多宝贵意见。

对于本书的出版,给予许可并经常给以指导的以三菱重工名古屋制作所末永总技师为首的诸位领导给予了很大的支持。

本书第三篇的后半部分,承蒙在程序设计技术方面具有较深造诣的高桥正树帮助修改原稿,从而使本书更加充实。

此外,マイテック出版公司的高御堂富三夫、山上守对本书的出版也付出了辛劳。在本书出版之际,谨向他们深表谢意。

目 录

序	
前言	
第一篇 微型计算机基础	1
第一章 对程序的追求	2
一、随机逻辑向程序技术的转变	2
二、程序技术的学习方法	4
第二章 微型计算机的结构	5
一、微型计算机的基本结构	5
二、存储器	6
三、微处理器	8
四、输入输出口	12
第三章 指令的结构和执行	14
一、指令的形式	14
二、指令的机器码	14
三、指令在存储器中的分配	15
四、指令的执行	15
五、程序的执行	18
第二篇 指令及其功能	20
第一章 传送指令	22
第二章 运算指令	28
一、加法指令	29
二、减法指令	31
三、加1指令与减1指令	34
四、逻辑运算指令	35
五、比较运算指令	40
六、其他运算指令	41
第三章 循环移位指令和转移指令	44
一、循环移位指令	44
二、转移指令	46
第四章 堆栈操作指令和子程序调用指令	48
一、堆栈操作指令	48
二、子程序调用指令	50
第五章 输入输出指令、中断处理指令及其他指令	53
一、输入输出指令	56
二、中断处理指令	57
三、其他指令	58
第三篇 程序设计技巧	60

第一章 数据传送	62
一、立即数的传送	62
二、寄存器之间的数据传送	63
三、直接寻址型的数据传送	64
四、间接寻址型的数据传送	65
五、变址寻址型的数据传送	68
六、其他的寄存器之间的数据传送	70
第二章 基本运算的处理	72
一、寄存器和存储器内容的清除	72
二、加法运算	74
三、减法运算	77
四、逻辑运算	80
五、增量和减量	84
第三章 条件转移	87
一、条件转移的处理形式	87
二、条件转移指令	88
三、零的判断	90
四、数值大小的比较	92
第四章 数组和表的检索	103
一、一维数组	103
二、2 字节数据的数组	105
三、7 段代码变换	106
第五章 选择转移	110
一、选择转移处理	110
二、选择转移的结构	111
三、选择转移的方法	112
四、选择转移的通用化	114
五、选择转移过程中应注意的问题	115
第六章 循环处理	119
一、循环处理的方式	119
二、指定循环次数方式	121
三、反复循环方式	122
四、迭代循环方式	125
第七章 移位和循环移位	127
一、逻辑移位	127
二、算术移位	130
三、16 位数据的循环移位	132
四、译码	132
五、编码	135
六、位列的左右变换	138
第八章 子程序	140
一、子程序的结构	140

二、子程序的执行流程	144
三、几个子程序的例子	146
四、高质量子程序的编制方法	150
第九章 输入输出和中断处理	153
一、输入数据的原封不动输出	156
二、AND动作	157
三、OR动作	159
四、记忆动作	160
五、计数动作	162
六、对显示器的输出	164
七、传送带的控制	166
八、停表	170
第十章 算术运算	180
一、10进制加法运算	180
二、10进制减法运算	183
三、2进制数的乘法运算(不带正负号)	190
四、2进制数的除法运算(不带正负号)	197
第十一章 2进制-10进制的变换	205
一、10进制数变换为2进制数	205
二、2进制数变换为10进制数	208
附录	213
参考文献	218

第一篇 微型计算机基础

对于初学微型计算机者，要达到运用自如的程度，是要付出相当艰苦的劳动的。

首先，要从学习微型机中使用的二进制数开始，进而了解微型机的结构和功能，然后学习程序的编制方法及接口技术。

总之，要想达到熟练使用微型机的目的，首先必须牢固掌握微型机的基础知识。

学习微型机技术，就象登山一样，起初，虽然体力和精力充沛，但总觉得有点单调无味；但随着不断的攀登，环顾周围群山，美丽景色，便会逐渐感到别有一番情趣。当你攀上顶峰的时候，更会有豁然开朗之感。

请朝着这个行将打开的未知世界一步一步地努力前进吧。

第一章 对程序的追求

一、随机逻辑向程序技术的转变

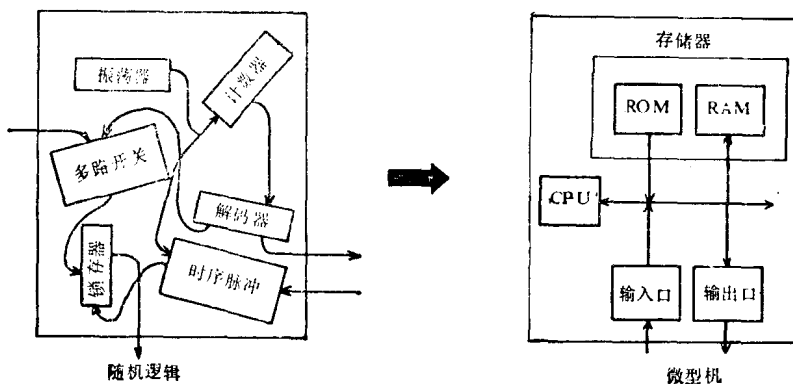
1971年,日本某计算器公司委托刚成立不久的INTEL公司制作计算器中使用的LSI(大规模集成电路),从此便对这种LSI的电路结构的设计和使用进行了新的大胆尝试。

在此以前,计算器中使用的LSI是由计数器、锁存器、译码器以及定时器等电子线路,经适当组合,即所谓随机逻辑组合而构成的。每当计算机的类型有改变时,电路也必须重新设计。

后来,出现了一种新的设计方法,即在算术运算单元里增设存储单元,以使用程序来进行控制。这种方法的基点是以改变程序的方法来适应不同类型的电算器。

微型机4004的原始设计就采用了这种方法。

随机逻辑向程序技术转变的示意图如下图所示。



如果说微型机的出现是从随机逻辑向程序发展的结果,那么,后来微型机应用工作的进展,也可以说成是经历了同样的过程。

现在看一看用微型机构成某一系统的情况。作为微型机的控制对象,可能是机械装置,或家用电器,也可能是管理系统。总而言之,不论对象是什么,都要着眼于各种繁杂的现象与微型机的结合。

过去是用IC(集成电路)或晶体管组成的系统来对繁杂现象进行控制的,这就是所谓随机逻辑控制。

用微型机对随机现象进行控制的系统结构,如图1-1-1所示。

用微型机的指令,把多种复杂现象的内容及其处理顺序编成程序并存放在存储器中。

指令与微型机之CPU的功能及作用、输入/输出、存储器容量以及信号线的数目等有密切关系,它代表了微型机的总体功能。通过指令系统可以看出一个微型机的能力。

程序设计就是利用指令把各种复杂的现象条理化,并写入存储器;最后,用2进制数表示看起来杂乱无章的各种现象,并以整齐的行列形式将这些2进制数存放于存储器中。

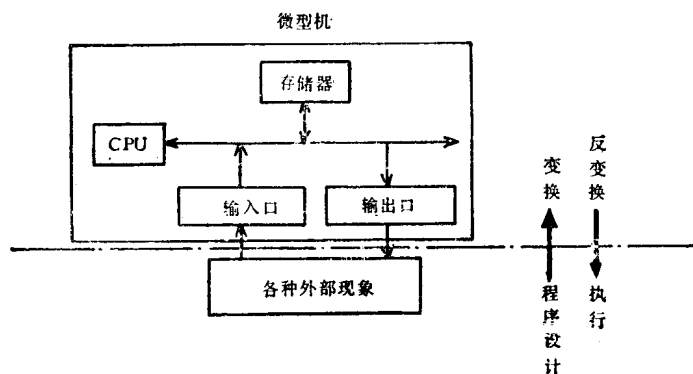
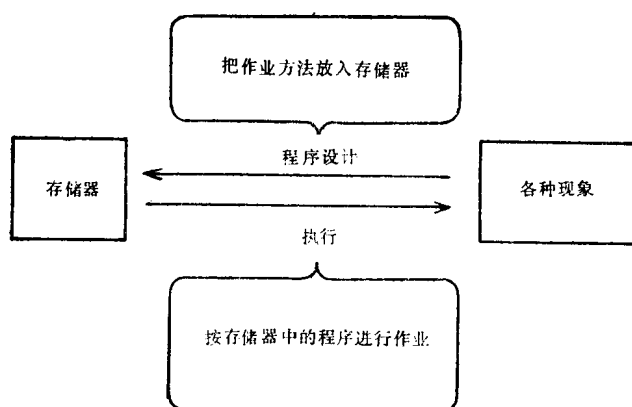


图 1-1-1 微型机控制系统

如果称把现象变换为存储器中的2进制数为程序设计，则将会有一个反变换，即：由CPU将存储器中以2进制数表示的指令一个个取出来，解读并加以执行，再通过输入输出，重现需要处理的繁杂现象，请参见下图。



程序设计对程序员来说是非常辛苦的工作，而计算机的CPU却能自动、高速地执行确定的程序。

下面，小结一下上述过程。

对随机逻辑技术来说，欲构成一个系统，必须根据控制对象的不同，对IC等不同的元部件进行配线。程序技术则不然，如图1-1-1所示，尽管被控制对象的内容有变化，而CPU、存储器、输入、输出等硬件部分却是完全固定的，只是存储器中的信息有变化。无论什么样的外部现象，只要变换一下存储器里的信息，便几乎都能进行处理。这一点是程序技术与随机逻辑技术的最大不同点。

上述硬件的固定性这一点，带来了产品可成批生产的效果；同时，随着半导体技术的进步，高密度集成化工艺取得进展，产品将处于高性能、廉价化的状态。

程序技术的渗透，不仅仅限于微型机本体方面，最近，已延伸到与外部现象直接结合的外围部件，出现了可程序的外围集成片，如可程序I/O（输入/输出）器件。

这种集成片备有各种各样的功能，可以根据使用者的需要，用程序进行选择。

此外，还有用程序技术来构成CPU功能的方法。这就是微程序设计，大型计算机中已经采用了这种构成法。

半导体技术也涉及到程序技术这一领域，位片式微处理器就是典型的代表。利用这种处理器，再结合程序技术，便可用规模不大的硬件，制做出具有任意功能的计算机。由于程序的规模十分庞大，在计算机的处理速度方面还不够理想。

不过，某些微处理器就是用上述方法构成的。向程序技术发展的这一趋势，在计算机及半导体产品的各个领域都在稳步发展。

二、程序技术的学习方法

由微型机构成系统时，由于其硬件已经定型化，只要将硬件组合起来即可。因此，重点应转移到系统及程序的开发等软件技术上来。

软件生产水平的提高已直接影响着系统的成本，因此，正在努力开展这方面的工作。一方面是开发系统和开发方法，另一方面则是改进程序设计者使用的程序设计方法和提高设计的基本技巧。

关于程序开发系统和开发方法，已有不少书刊、杂志作了详尽清楚的介绍，这方面的资料极易得到。

但是，有关程序编制方面的十分成熟的书似乎还没有见到。

初学者在学习程序技术时，盼望有能逐步系统地提高技术的参考书。

另外，关于供学习用的微型机机种问题，应选择当前市场上最多、最流行的机种。这对以后实际设计系统时，所遇到的对元件部件的购置、价格及程序的互换性等方面的问题的处理都有利。

基于上述想法，本书以当前世界上标准的8位机8080A/8085A为例，对程序的编制方法进行论述。

当前，市场上的微处理器的指令形式大致相同，因此，只要精通了一个机种，其他机种也就可以触类旁通了。

第二章 微型计算机的结构

一、微型计算机的基本结构

为了编制程序，有必要先了解一下微型计算机的基本结构和动作。

不过，微处理器（以下简称处理器），硬件各个信号的作用以及定时信号等方面的知识，并不是编制程序特别需要的，况且，对初学者来说，一开始就接触这种难懂的东西也是不合适的。

下面，详细看一看与程序编制有关的各个重要事项。

由微处理器8080A构成的微型计算机，如图1-2-1所示，它可以大致分为三个主要部分，即：

- (1) 微处理器；
- (2) 存储器；
- (3) 输入输出。

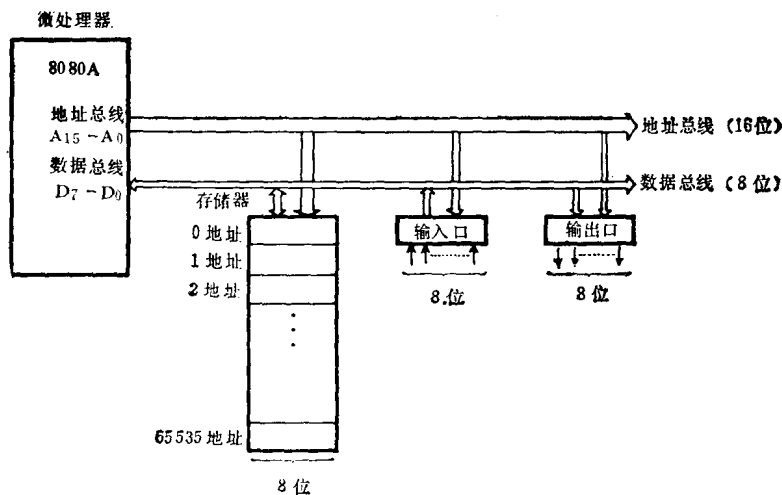


图 1-2-1 微型计算机的基本结构

上述三个主要部分，由被称为数据总线的8位（8根导线）信息通道连结起来；可以根据需要而通过数据总线相互传递信息，以便实现各自的功能。连结这三部分的数据总线是双向通道，从而使微处理器可以通过它进行数据的输入和输出。

地址总线为16位（16根导线），供微处理器选择存储器或输入输出地址用。

首先讨论一下微处理器和存储器的动作。存储器用于存放表达工作顺序和方法的程序、运算结果和数据。

为了实现微型处理器从存储器读取数据，由地址总线给存储器指定必要的地址。指定地址之后，存储器便将该地址处的数据送至数据总线，微处理器取入此数据，读取动作便完成了。从微处理器向存储器写入数据时，也同样要指定地址，然后再把数据写入存储器。

微型计算机与外部之间所进行的数据通信，要通过输入或输出来实现。输入和输出各自可以增设到最多为256个。地址总线中的一部分，可以用来指定必要的输入或输出地址。

图1-2-1所示为由微处理器8080A构成的微型计算机。由改进型微处理器8085A组成

的微型机与它无重大差异，只是地址总线的结构不同，但这仅仅是硬件问题，对程序的编制没有任何影响。图1-2-2示出了其结构。

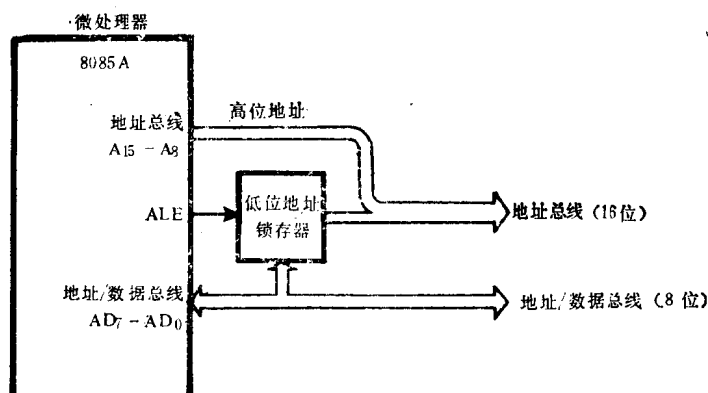


图 1-2-2 8085A的地址总线结构

8080A中一起引出16条地址线，而在8085A中则是分两组引出来的。地址的高8位从处理器引出；低8位地址/数据线上的数据可通过用特定时序进行锁存（锁存器是能暂时保存数据的电子线路）而获得。

8085A中的所谓地址/数据总线，具有数据总线和低位地址总线两者的功能，在特定的时序之下它将分别变成数据总线或低位地址总线。所谓地址/数据总线这一名称就是因为具有此两种功能而得到的。

在8085A中，并没有改变通常的功能就把地址总线的低8位，兼用作数据总线了，故与8080A相比就空出来了8只管脚。这些管脚可用以增设中断请求线、增设串行数据的输出线，起到有效地加强功能的作用。

二、存储器

存储器是存储指令和数据的场所，8080A和8085A的存储器之字长为8位（这里的指令，是指能被微处理器所执行的部分，可参阅第三章）。

存储器中分布有从0地址到65535地址的连续地址，由16条地址线来指定地址。

当处理器想从某个地址取指令或数据时，就将该地址从地址总线送出，该地址的8位存储器之内容则通过数据总线送入处理器。

从处理器向存储器的写入按照相反的方向进行，即：首先从处理器将8位的数据送至数据总线，而后该数据被写入指定地址的存储器。

微型计算机的存储器，通常是由只读存储器（ROM）和随机存储器（RAM）组成的。

（一）只读存储器（ROM: Read Only Memory）

ROM是专用于读出的存储器，处理器在工作中不能向它写入数据，ROM的特点是当电源切断之后，其中的信息保持不变。因此，在ROM中存放指令或内容不变的固定数据。由于程序是由指令组合而成的，用以表示处理器进行工作的顺序，而且在执行过程中内容不变，故程序通常存放在ROM中。

只要一接通电源，存放在ROM中的程序便可立即投入运行，而且只要存储器不坏，程序就永远不会消失。

如图1-2-3所示, ROM有几种类型。PROM (Programable Read Only Memory) 可由用户用写入装置自由地写入, 其中有一种可以消去再写的EPROM (Eraseable PROM), 使用非常方便, 故得到广泛利用。

(二) 随机存储器 (RAM: Random Access Memory)

RAM的特点是, 电源一切断, 其中的内容全部消失; 不过, 当电源投运之后, 可以自由地写入数据, 然后还可以读出来。

在暂时保存运算的中间结果或用做堆栈的场合, 一定要使用RAM。

这样, 在通常的系统中, 便集其各自的特长, 混合使用ROM和RAM组成存储结构 (图1-2-4)。

在开发程序时, 要如图1-2-5所示的那样变换RAM和ROM的结构, 这样, 对程序的开发便很方便。图1-2-5中左图所示为程序开发最初阶段的存储器结构。电源一投运, 通常便从0地址开始执行程序, 所以在靠近0地址的区域设置ROM或PROM。在这里存入所谓监控程序, 此程序的功能是通过纸带读取机或键盘将实验程序输入到RAM1 (也可称为装入), 以便修改已输入的程序。在RAM2中存放可变数据或指定堆栈, 这些是执行RAM1中的实验程序时所需要的。

上述这种存储器的结构与图1-2-4所示的结构不同, 每当断电再起动时必须由监督程序将实验程序装入RAM1, 不过RAM中实验程序的内容可以随便修改变更。经过反复修改, 第一阶段的程序一经修改完毕, 便在第二阶段中把已经完成的程序写入PROM (可消去再

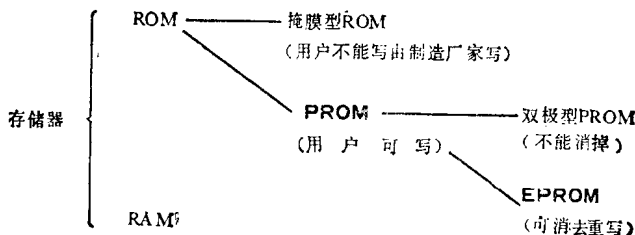


图 1-2-3 半导体存储器的种类

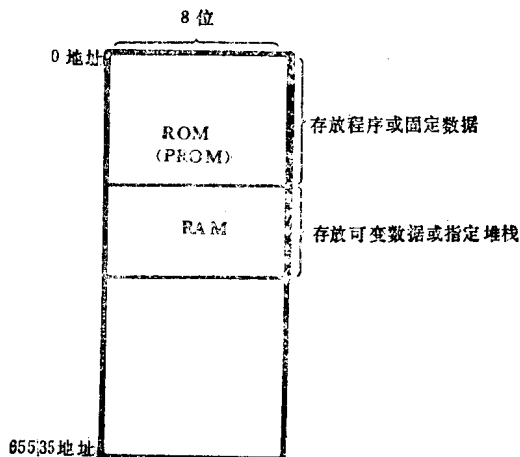


图 1-2-4 通常的存储器结构

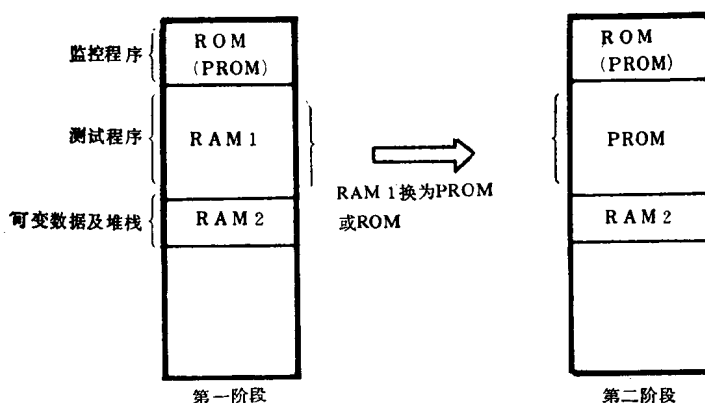


图 1-2-5 开发程序时的存储器结构

写的EPROM使用得更为广泛)，进行最后实验。在这一阶段不再需要用纸带读取机或键盘来装入实验程序了。

如上所述，用RAM、PROM输入实验程序，在机器上修改变更程序的内容称为“调试”（debugging）。调试在程序开发过程中是必不可少的。

三、微处理器

从程序员的角度看，可以认为微处理器8080A和8085A具有如图1-2-6所示的内部结构。微处理器的功能是进行运算并对输出进行控制，但这些都与程序员没有直接关系，可以简略地将这部分内容用图表示出来。

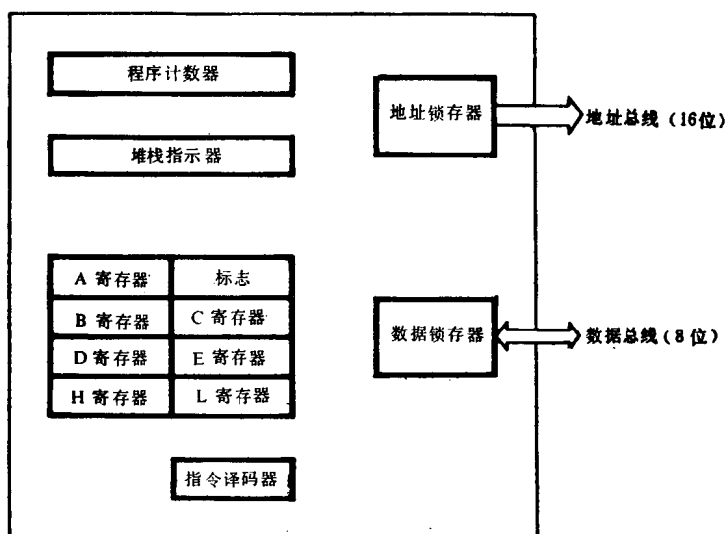


图 1-2-6 从软件角度看8080A、8085A的内部结构

（一）地址锁存器（address latch）

存储器地址或输入输出地址，在向外部输出时可暂存于地址锁存器中。这里用的是16位锁存器。程序计数器、堆栈指示器以及两个寄存器连结而组成的寄存器对等装置中的数据，可根据需要暂存于锁存器中，然后从地址总线送出去。

对8085A来说，地址锁存器的数据分为高位地址和低位地址，分别由不同的总线送往外部；如图1-2-2所示，两条地址总线在外部电路中又可以合在一起，形成16位的地址总