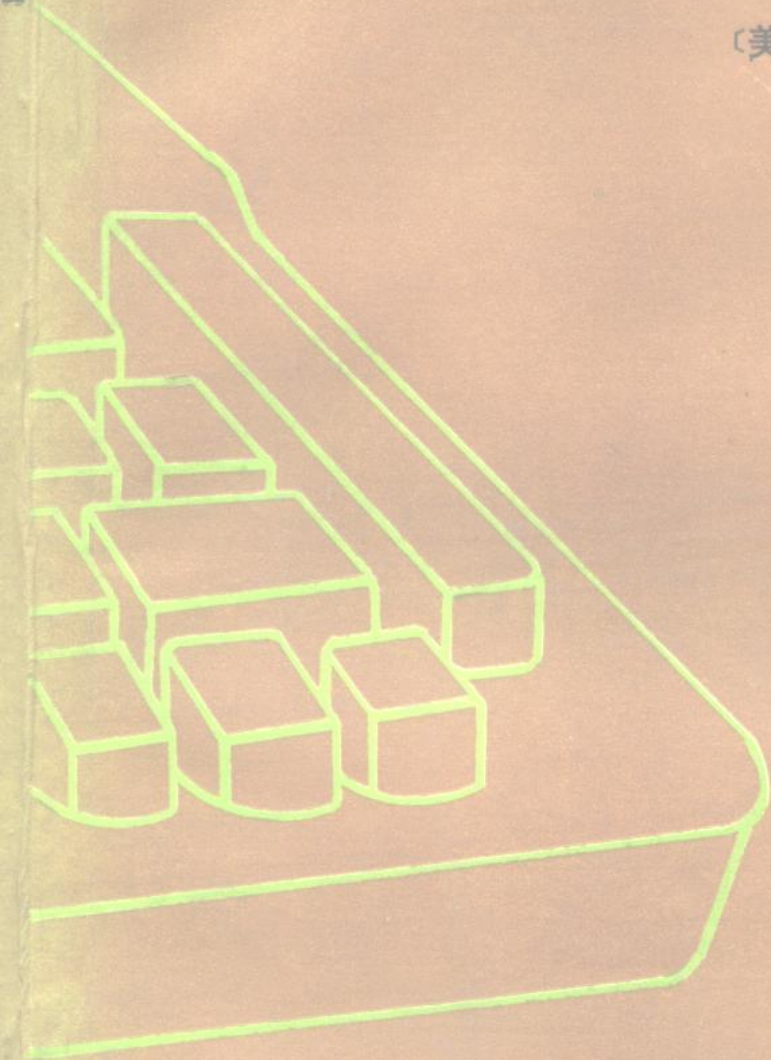


高档微处理器

〔美〕 G·本森·格雷 著



3.876

156

机械工业出版社

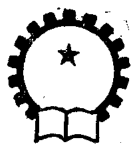
高档微处理器

〔美〕 G·本森·格雷 著

王自强 胡崇岳

郑清明 王永鸿

吴昊江 译



机械工业出版社

内 容 简 介

JS/32/03

本书为IEEE“自学丛书”中的一本，它系统介绍了微处理器及微型计算机系统的软硬件结构原理、开发工具、系统设计方法及典型产品等。内容包括：微处理器回顾、高档微处理器结构、16及32位微处理器、外部支援器件、软件、系统设计及开发系统。

本书对高档微处理器作了全面的分析，编排方法新颖，叙述深入浅出，教学辅助环节丰富，是一本很好的高档微型计算机入门读物及自学教材。

本书可作为各行业工程技术人员、科学工作者、管理干部、广大职工的继续教育自学教材，也可供大中专院校师生阅读参考。

ADVANCED MICROPROCESSORS

G. Benson Gray

THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS

1985

* * *

高档微处理器

〔美〕 G. 本森·格雷 著

王自强 胡崇岳 吴昊江 译
郑清明 王永鸿

责任编辑：刘思源 责任校对：宁秀娥
封面设计：刘代 版式设计：霍永明
责任印制：卢子祥

机械工业出版社出版（北京阜成门外百万庄南里一号）

（北京市书刊出版业营业许可证出字第117号）

北京市密云县印刷厂印刷

新华书店北京发行所发行·新华书店经售

开本 787×1092¹/₁₆·印张12¹/₂·字数303千字

1989年11月北京第一版·1989年11月北京第一次印刷

印数：0,001—2,030·定价：10.20元

ISBN 7-111-01366-2/TP·85

1100100

译 者 的 话

从1971年4位微处理器诞生算起,微处理器问世仅短短十几年,而产品已更新若干代了,发展速度异常迅猛。今天,32位微处理器作为高档微型计算机的核心部件,已广泛用于科学计算、数据处理、实时控制、CAD/CAM等领域,高档微型机的性能与小型机相比已不相上下。

微型机在科研生产中的应用已越来越普及。随着技术的飞速发展,广大科技、管理人员迫切 need 加强微型机方面的知识。这本“高档处理器”自学教材是电气电子工程师学会(IEEE)于1985年赠予中国电气电子工程学术团体的,IEEE作为世界上最大的工程技术团体,在继续教育方面已有多年成功的经验。作为尝试,我们把这本书翻译介绍给广大读者,希望能对我国科技人员的知识更新、继续教育做一点贡献。

本书在题材的选编、内容及形式的安排方面有独到之处,每课都配有词汇表、自我测验、小结。书后配有参考文献、测验答案及结业考试题。本书内容丰富,具有相当的广度和深度,对微处理器领域的最新技术发展都有介绍,是一本很适时、很成功的教材。

王自强翻译了本书第二、四、五课;胡崇岳翻译了第一课;郑清明翻译了第三课;王永鸿翻译了第八课;吴昊江翻译了第六、七课;王自强对全书译稿进行了整理和校订,并翻译了录音带。

IEEE北京分会、中国电工技术学会及学部委员丁舜年同志对本书的翻译出版给予了关怀与指导,在此特表谢意。

限于译者水平,错误及不妥处敬请读者批评指正。

译 者

1987年12月3日

《高档微处理器》教材编写组简介

IEEE〈电气电子工程师学会〉“高档微处理器”自学教材是由Longman Crown 出版公司计划与开发部副主席小路易斯·E·弗伦泽尔领导的小组编制的。这个以弗吉尼亚州雷斯顿市为基地的出版社专营高等技术继续教育计划和计算机软件培训方面的书刊。

本教材主要作者是俄勒冈州波特兰市(Portland)市 G·本森·格雷 (G·Benson Gray) 博士,他在西德波恩大学取得学士、硕士和博士学位,曾在欧洲IBM 公司任计算机系统工程师,在Amdahl 公司任计算机系统设计师。

弗吉尼亚州麦克莱恩市 (McLean) Snyder Associates 公司的托马斯·戈德史密斯 (Thomas Goldsmith) 负责教材编辑。录音带由俄勒冈州莱克奥斯威尔 (Lake Oswego) Instructional Research Associates 公司的罗伯特·米勒 (Robert R. Miller) 提供。审稿由弗吉尼亚工学院电机系教授C·E·纽纳利 (C·E·Nunnally) 博士担任。

插图未经审定,由Longman Crown 公司小路易斯·弗伦泽尔 (Louis E. Frenzel) 提供。

目 录

译者的话

《高档微处理器》教材编写组简介

如何使用本教材.....	1
序言.....	3
课程说明.....	4
第一课 微处理器的回顾.....	5
第二课 高档微处理器的结构.....	29
第三课 16位微处理器.....	46
第四课 32位微处理器.....	86
第五课 外部支援器件.....	109
第六课 软件.....	134
第七课 用16位和32位微处理器进行系统设计.....	151
第八课 开发系统.....	165
参考文献.....	182
自我测验答案.....	183
结业考试.....	187

如何使用本教材

IEEE“高档微处理器”课程由五个主要部分组成:

1.学习指南——即本书,是本课程的核心。它包括自学课文,此为学习的基础;还包括学习的目标,描述你将学到什么,学完此课后能做什么;最后还有自我测验,能帮助你检查自己所理解的程度和掌握的情况。

2.参考文献——IEEE出版社发行的《高档微处理器参考资料》一书(编号PC 01602)[⊖],收集了有关的文章和论文,是对自学课文的重要补充。

3.录音磁带[⊕]——标准的C-60盒式磁带,包括一系列总结讨论,对每一课都有5~8min的小结。用来帮助复习每一课的要点,这一口头讨论可帮助你记忆。

4.结业考试——这一综合的选择题方式的考试,不仅用于衡量学习成果,也是一个重要的学习工具。考题检查你掌握的理论实践知识,让你把它们运用于实际世界。答完考卷并寄往IEEE评分后,你会收到结业证书和继续教育成绩学分(CEAUS)。

5.词汇表——书中所用术语的词汇表放在每课后面。课文中已被定义过的词汇,一般在词汇表中就不再定义。没在课文中定义的特殊术语可在词汇表中查到。

这五部分内容将帮助你从所提供的信息中获取微处理器的基础知识。下面略述你在学习本课程时应遵守的步骤:

开始:

1.阅读前言——将此课程作一透视,概述它的要求及重要性。

2.读课程说明——“学前要求”会告诉你,为学习本课应具备的基础。课程目标将勾出本课的轮廓,指出你将学到些什么。

3.开始第一课——以后每课也都遵循如下步骤。

为学习每课:

①浏览本课目标——告诉你本课将学些什么。

②读课文——学习每课最好一气呵成,这将保证你思维连续,减少对复习的需求。

③回答自我测验题——课后的问题将给出你需要复习的范围。

④放录音磁带[⊕]——听本课的录音小结,它会辅助你复习,并为下一课作好准备。

磁带内容如下:

第一面

※第一课——微处理器的回顾

※第二课——高档微处理器的结构

⊖ 该书的主要部分已列入本书的参考文献中——译者注。

⊕ 需要录音磁带者可向北京197信箱计算中心联系。

⊕ 小结内容已译成文字附在每课之后——译者注。

※第三课——16位微处理器

※第四课——32位微处理器

第二面

※第五课——外部支援器件

※第六课——软件

※第七课——用16位及32位微处理器进行设计

※第八课——开发系统

4.重复该程序——按照以上步骤学习每一课，不要在每课学习间中断太多时间，保持一稳定的学习周期，可确保所学知识的连续性。

最后：

1.结业考试——这是本课程一个不可缺少的重要部分，考试将总结和复习全部课程内容，并证明你已达到课程目标。请遵循考卷的要求。

2.答卷寄回 IEEE——这是一个可选步骤，如果你想要一个合格证和CEAUS学分，应将完成的答卷寄给 IEEE。IEEE 将对你的试卷评分，并将答卷连同一份标准答案寄回给你。如果你取得了合格分（70%）或更高，你将被授予合格证书和CEAUS学分。

序 言

现在,商品微处理器已问世十几年了。大多数电子工程师都知道并正在应用这些多用途器件,他们把这种“一块芯片上的计算机”看作是一种能组成较大较完善系统和装置的集成电路部件。

正如大多数电子器件一样,十几年来微处理器已经经历了重大变化。新的半导体工艺已使微处理器体积更小,速度更快,功耗更低。新的超大规模集成电路(VLSI)技术也向我们提供了优良的外围接口电路,从而大大减少了设计时间、芯片数量和成本。结构的开发产生了新的内部设计,它们具有改进了的指令系统,并有更多的内部寄存器和逻辑电路,从而大大增强了计算功能。最重要的是软件开发已进一步增强了微处理器的通用性,并使之更容易应用和设计。

事实上,变化已相当巨大,新型16位和32位系列微处理器已诞生。这必然开辟了新的应用领域并增强了微处理器的功能。作为一个工程师、经理或教师,这些发展正冲击着你们,这门课程正是想让你赶上时代,便于你们利用这些精良的新器件。

本课程计划讲授高档微处理器;包括新型16位和32位微处理器的详细结构、最新的器件的说明、它们的特点、性能的比较以及软件、应用和设计方面信息。

在本课程中要回答工程师和管理人员的一些普通问题:

1. 16位和32位器件的重要性是什么?它们能做哪些8位器件做不了的事?
 2. 8位与16/32位中央处理单元(CPU)的基本结构差别是什么?哪些性能和规格最重要?
 3. 设计系统用到的最流行的16位、32位微处理器是哪些?最常见的是哪一种?
 4. 在使用16/32位微处理器时,应该知道的芯片技术和软件是什么?
- 本课程将使你接触到这一新技术,必能使你得益非浅。

课 程 说 明

学前要求:

具备数字电路及微处理器的基础知识, 最好有 4 位或 8 位微处理器使用经验。

课程目标

学完本课程后, 你将能:

1. 说出新型 16 位/32 位微处理器与老一代 8 位器件不同的结构特点, 及其优良性能。
2. 列举 4 种 16 位商品微处理器, 并指出两种使用最广泛的机型。
3. 列举 4 种 32 位商品微处理器, 并指出两种使用最广泛的机型。
4. 对比 16 位与 32 位微处理器的性能和特点, 讨论各种选择方案。
5. 对于给定的应用目标, 列出选择 16 位或 32 位微处理器及用它们进行设计的步骤。
6. 举出 4 个必须采用 16 位或 32 位微处理器的实际应用例子。
7. 讨论在选择 16 位或 32 位微处理器、及用其进行设计时涉及的三种主要软件。

第一课 微处理器回顾

导 言

第一课的目的是回顾微处理器的构造和作用。我们选择的是最流行的、应用最广的8位微处理器8080A以及相继发展起来的8085和Z80为例来说明典型器件的主要性能、特点和功能。目的是补充你的基础知识，并以此作为学习后面几课的起点。这一课的重点是正确了解各组成部分的相互联系，并为学习后面几课作好必要的准备。

如果你觉得自己已经熟练掌握了微处理器的基本原理，也可以跳过这一课。不过，你也许应该先读一读“课程目标”，并按本课末尾所附的“自我测验”检查自己的知识。如果你能正确回答8个以上的问题，就可以学习第二课了。

本 课 目 标

学完这课后你将能：

1. 说出微型计算机的四个主要部分。
2. 列举典型微处理器的主要组件，并说明其作用。
3. 描述微处理器的运行过程。
4. 说明中断的必要性、中断功能及中断操作。
5. 定义并说明存储器直接存取过程。
6. 描述8080A、8085及Z80微处理器的结构并对比它们的性能。

微处理器的应用

微处理器是一种单片集成电路，它能完成“中央处理单元（CPU）——任何计算机的心脏”的全部基本功能。微处理器（CPU）再加上存储器（通常包括RAM和ROM）和输入/输出电路（I/O）就构成一台完整的微型计算机。那么，这种微型计算机（micros）用在什么地方？事实上，计算机和电子领域内到处都在使用微型计算机，由于微型计算机的结构小巧、功能强大而又具有多种的用途，使它在许多领域获得了普遍的应用。

微型计算机，基本上可分为通用的和专用的两大类。通用微型计算机是大家最清楚、接触最多的。通用数字计算机的核心是微处理器。通常，计算机由CRT、与用户联络的键盘、外围设备（如打印机或鼠标）、辅助存储设备（如软盘和硬盘）组成。称为微型计算机、个人计算机或工作站的计算机都是通用的，它们都能按照程序工作，完成用户所需要的功能，软件能使计算机满足用户的要求。

专用微型计算机具有固定功能，只完成一种任务，而且软件通常固化在ROM中，微型计算机在该软件的控制下完成一项单一的、特定的任务。

通常，专用微型计算机也可以装入大的系统或一整套设备中，如装在测试设备、工业控制机或武器系统中。

专用微型计算机一般可取代以往由较大、较昂贵的、分立逻辑元件组成的数字电路。

在这一课中，我们把注意力集中到CPU或微处理器的基本功能上。在以后的课文中要扩展到：阐明新型16位和32位微处理器，如何使通用和专用微型计算机系统增加新的功能和用途。

计算机的功能

一台典型的数字计算机包括：

1. 一个中央处理单元（CPU）或称处理器
2. 一个存储器
3. 输入/输出（I/O）端口

参阅图1-1

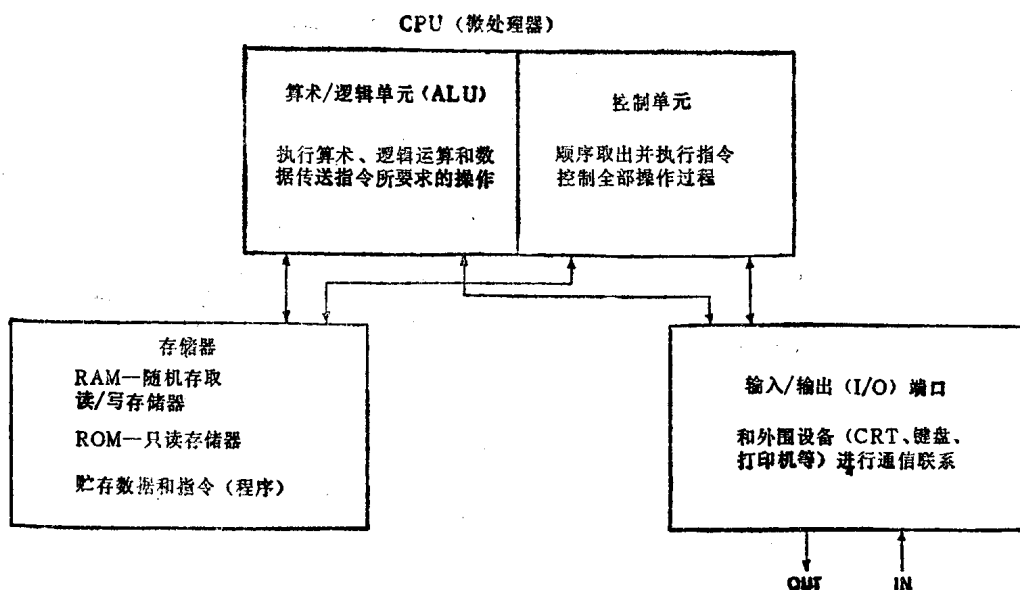


图1-1 数字计算机结构框图

存储器是用来存储指令和数据的地方。指令是一组指挥CPU活动的编码信息，数据是一种被CPU处理的编码信息。一组逻辑上相关的储存在存储器中的指令称为程序。CPU以确定的逻辑顺序从存储器中“读出”每一条指令，并用它启动处理过程。如果程序序列有条理并合乎逻辑，经处理就会产生明白有用的结果。

存储器也可以用来存储被操作的数据以及指挥操作的指令。运行时要取出指令还是数据是由程序决定的，编制程序应能做到，在需要一条指令时，CPU不会读入非指令的字。CPU能够快速存取存储器中的任何数据，但是，通常存储器不可能大到足以存入应用所需要的全

部数据。这个问题可以解决，只要为计算机提供一个或多个输入端口就行。CPU可以给这些端口编址，输入从这些端口进入的数据。由于加上了输入端口，使计算机能高速、大量地接收来自外部设备（如软盘或终端设备）的信息。

计算机还需要一个或多个输出端口，这些端口使CPU能把它的处理结果传送给外部世界。输出信息可以接到操作人员使用的显示器上；接到产生“硬拷贝”的外围设备，例如打印机上；接到外部存储设备，如软盘上；或构成过程控制信号来指挥另一个系统——如自动生产线的运行。像输入端口一样，输出端口也是可编址的。输入和输出端口一起就使处理器能和外部世界通信了。

CPU或处理器协调整个系统，控制各部件的工作。CPU必须能从存储器中取出指令、对其二进制信息进行译码并执行之。在指令执行过程中，必要时它必须能对存储器和I/O端口进行访问。此外，CPU应能识别和响应某些外部控制信号，例如“中断”和“等待”请求信号。在CPU内部能完成这些功能的部件将在下面描述。

微型计算机或个人计算机包括上文描述的所有基本部件。而CPU是一种微处理器，是一种能够完成算术/逻辑运算和控制功能的单片集成电路。

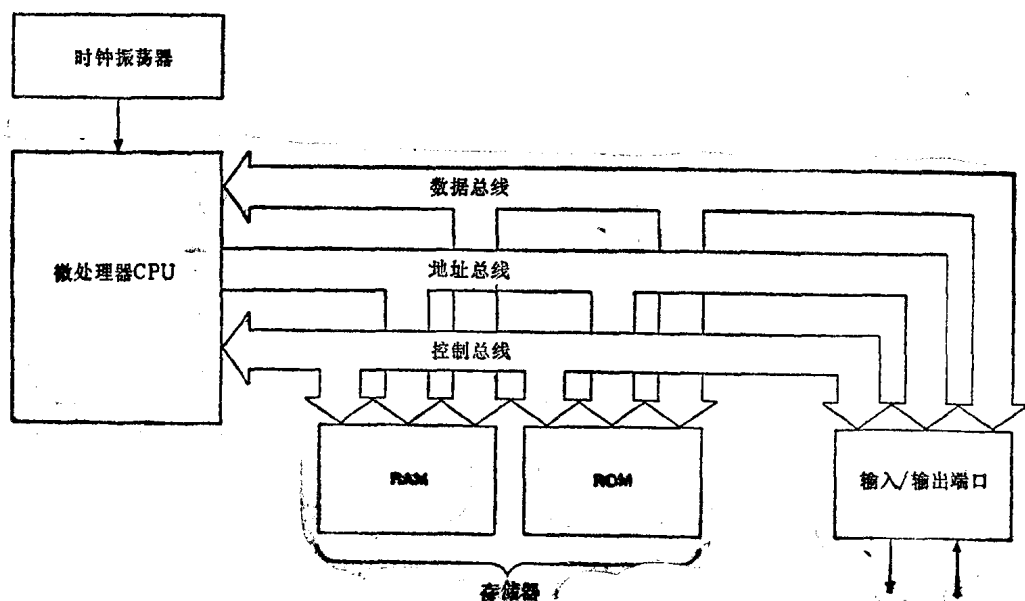


图1-2 微型计算机结构框图

图1-2示出较详细的典型的微型计算机结构框图，CPU或微处理器与存储器及I/O电路之间是通过一种称为“总线”的内部连接线进行通信联系的。数据总线是一组并行线，各种指令和数据都可以通过它输入CPU或从CPU输出。地址总线也是一组并行线，它把地址送往存储器和I/O电路，以便选通一个特定的存储单元或外围设备。

多路控制线通常合称为控制总线，用来传送启动操作的时控脉冲或某些特定状态信号。

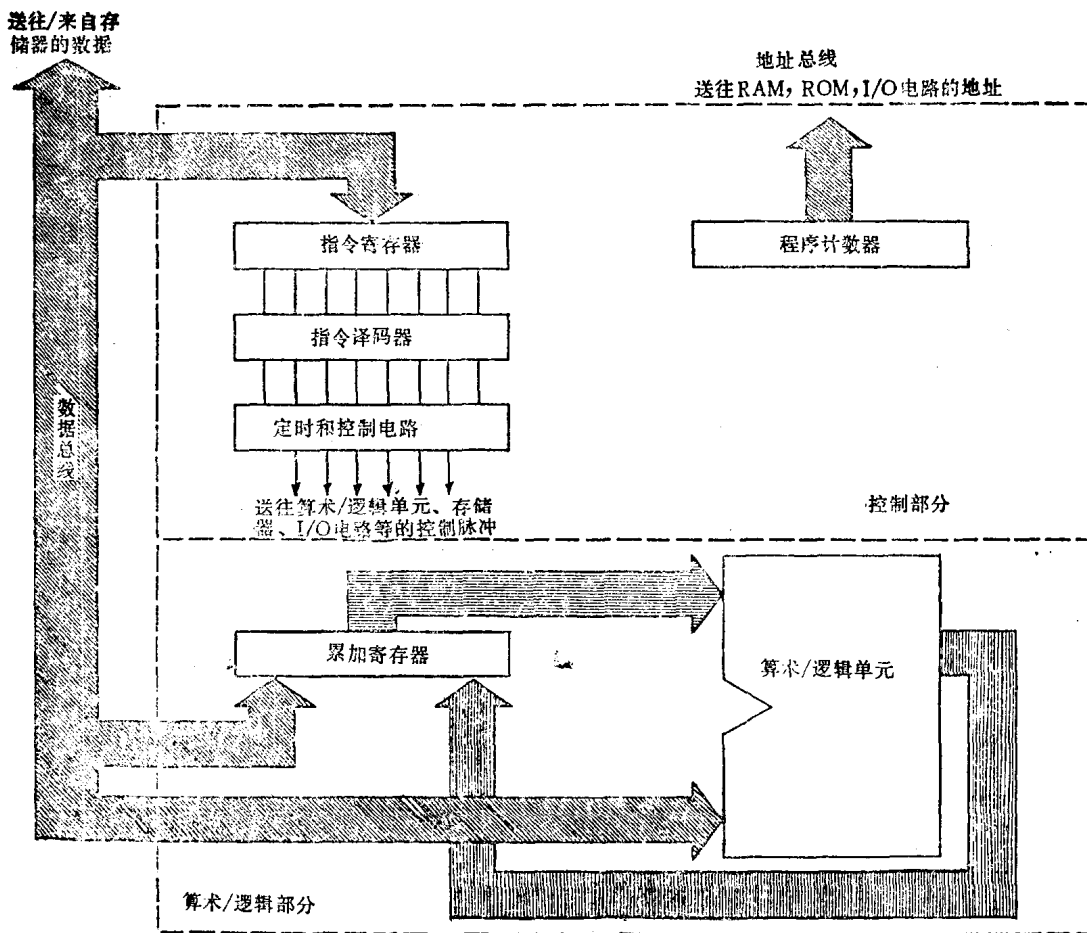


图1-3 微处理器（CPU）的基本结构框图

中央处理单元的结构

参阅图1-3

典型的中央处理单元（CPU）或处理器由下列互相联系的功能部件组成：

寄存器——用于二进制字（数据或指令）的固定位长（8位或16位）临时存储单元。

算术/逻辑单元（ALU）——用来完成加、减、乘、除及布尔运算和其它功能的电路。

控制电路——用来取出和执行指令及监测控制全部操作的逻辑单元。

累加器

累加器通常用来存储被ALU处理的操作数之一。一条典型的指令可以指挥ALU把某个寄存器中的内容加到累加器的内容上，然后把结果再存入累加器中。一般说来，累加器既是源（操作数）寄存器，又是目的（结果）寄存器。

通常，CPU包含若干个称为通用寄存器的辅助累加器，利用它们存储操作数或中间数据。有了这些通用寄存器，中间结果就不必在存储器和累加器之间来回传送了，因而提高了

处理速度和效率。

程序计数器

组成程序的指令是顺序地存入系统的存储器中的。为了确定下一步做什么，中央处理器要访问存储器。这意味着处理器必须知道下一条指令存在哪一个单元中。为区分出不同的单元，存储器的每一个单元都编上号，这个辨别存储单元的号数就称为该单元的地址。

处理器包含一个计数器。这个计数器中存着即将被取出并执行的下一条指令的地址，这个寄存器称为程序计数器。处理器每取出一个指令字节，就使程序计数器的内容加1进行更新。这样，程序计数器总是指向下一条指令。

因此，程序员按相邻地址号储存指令时，应使先执行的指令存入地址号较低的单元中，只有当指令是从存储器某一区跳转或转移到另一区时，程序员才可以违反这个“按顺序”的规律。

跳转指令内含下一条要执行的指令的地址。一般情况下，这个地址不是“按顺序”的下一个地址。只要跳转指令指出正确的地址，下一条指令就可以存在存储器的任何单元内。执行跳转指令时，处理器要把程序计数器的内容更换为跳转指令内包含的地址。这样，就保证了程序在逻辑上的连续性。

子程序

调用是一种特殊的程序转移，当事先存入的程序访问子程序时就需要调用。在这种转移中，处理器必须“记住”出现转移时程序计数器中的地址，有了这个地址，处理器在执行完子程序的最后一条指令时就可以返回，再继续执行主程序。

子程序是程序中的程序。通常，它是整个主程序执行过程中需要频繁使用的一组指令。例如，计算变量的平方、正弦或对数的例行程序就是一种函数子程序。又如，向特定的外围设备输入或输出数据的程序也是一种子程序。

为了保证运行时按一定规则返回主程序，处理器有一套处理子程序的特殊方法：当处理器接到调用指令时，就使程序计数器的内容递增并将计数器的内容存入称为“堆栈”的专用存储区内（参阅堆栈操作）。

附录：堆栈操作

大多数微处理器的结构中都含有堆栈，堆栈是为子程序操作而设置的RAM或读/写存储器的一部分。堆栈没有固定的大小，划归堆栈使用的存储单元的数目由堆栈本身的需要来决定。在某些应用中可以少到8~12个单元，而在另外一些较复杂的操作中，堆栈可以占用好几百个单元。

参阅图1-4，堆栈是一个后进先出（LIFO）的存储器。存在堆栈内的数据可按顺序写入和取出。这样，最后存入堆栈的项将成为最先被取出的项，而最先存入的项则最后被取出。

用来确定堆栈限度和识别堆栈中特定字存储单元的地址寄存器称为堆栈指针。在大多数8位微处理器中，这是一个16位寄存器，它可以指向65536个不同存储单元中的任何一个。这意味着，在大多数微处理器中，堆栈可以位于RAM的最大寻址范围内的任何一处。通常，堆栈指针寄存器在程序控制下装入为此目的的特定指令，用以设置堆栈范围。一旦堆栈指针

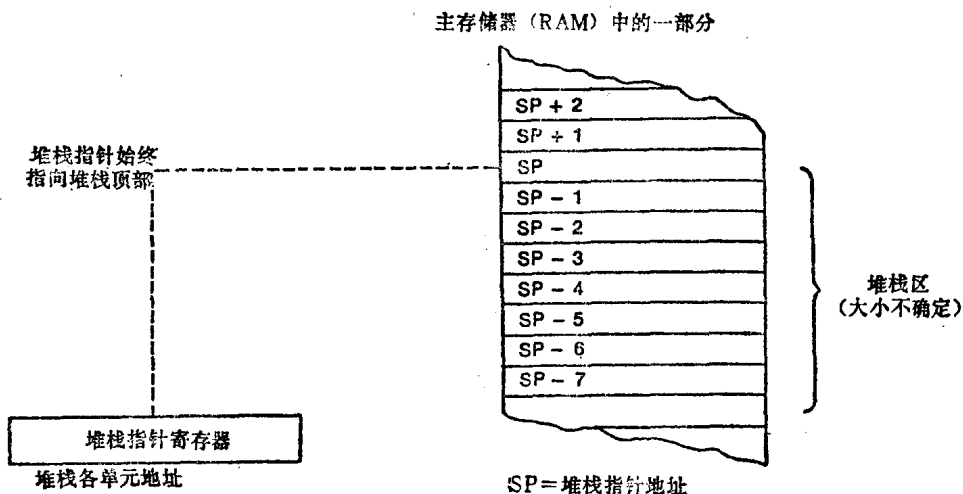


图1-4 堆栈

预置完毕，它将随着顺序访问存储单元（存数或取数）而加1或减1。

把数据存入堆栈的过程称为“推入”。一般认为，写入堆栈的每一个字是从堆栈上方（栈顶）推入的，每当推入一个新的字，以前存入的字就被逐渐推向栈底。

从堆栈读出数据的过程称为“弹出”。最后存入的字首先从堆栈读出，这种“读”操作就称为弹出或上托。每当从堆栈顶部弹出一个字，顺序排列的下一个字这时就移到了栈顶，为下一次读出作好准备。

术语“推入”和“弹出”没有真正表示出堆栈的确切作用，而是使数据存入堆栈和从堆栈读出的顺序更形象化。堆栈中的数据实际上并不象术语推入和弹出所表示的那样向下或向上移动。存入堆栈的全部数据都可以随机地存取，并且可以用其他指令来改变或进行检测。但是，数据的储存或检索是顺序进行的，因为堆栈指针不是加1就是减1。

堆栈指针总是包含着最后存入堆栈的那一项的地址。由于数据是从堆栈上方推入的，每推入一个数据，堆栈指针就减1，因此，堆栈指针指示的最低地址就是最后存入的数据字的地址。当数据被弹出堆栈，堆栈指针就加1，但是，不管计数方向如何，堆栈指针始终指向堆栈顶部。

这样，堆栈就把子程序执行完毕以后需要继续执行的指令的地址（断点）保存起来了。然后，处理器把调用指令中规定的子程序入口地址装入程序计数器。因而，下一条被取出的指令将是子程序的第一步。子程序总是以返回指令作为结束。返回指令无需规定地址，当处理器取出返回指令时，它就用堆栈顶部的断点地址取代程序计数器当前的内容。这样，处理器就从断点处立即恢复执行原来的主程序。

子程序的执行过程可参阅图1-5。

子程序通常可以嵌套。也就是说，一个子程序有时要调用第二个子程序，第二个子程序又可以调用第三个，等等。只要存储器有足够的容量储存必要的返回地址，CPU有相应的逻

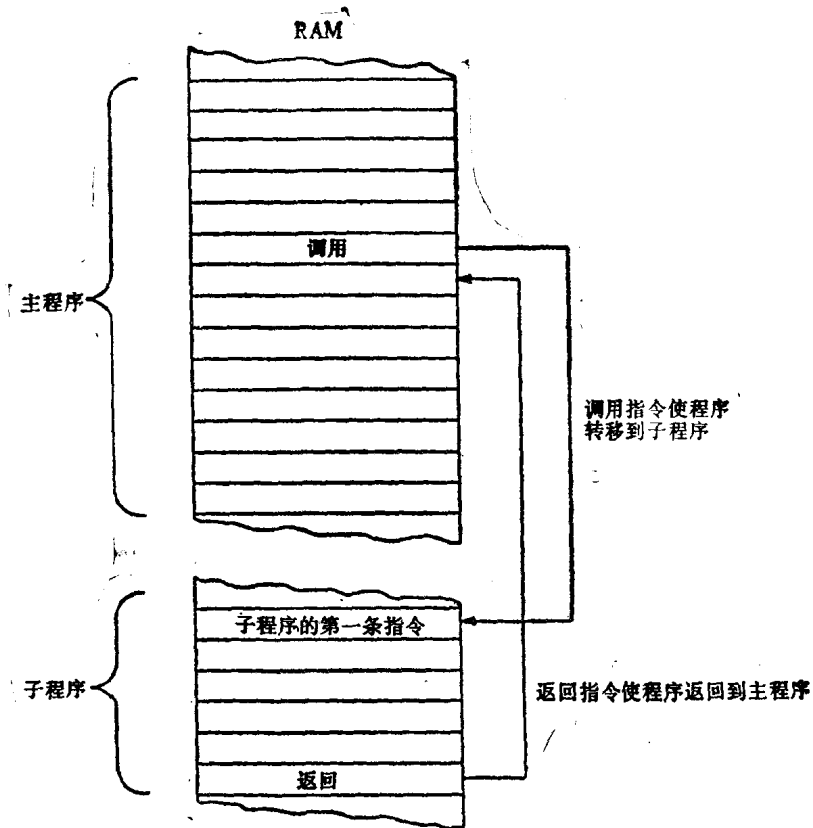


图1-5 子程序的调用与返回

辑装置，嵌套多少层都是完全可以的。换句话说，嵌套的最大深度取决于堆栈本身的深度，如果堆栈能够储存三个返回地址，那么就可以容纳三层嵌套。

不同的处理器有设置堆栈的不同方法。在某些处理器中，储存返回地址的装置已成为处理器本身不可分割的一部分。另外一些处理器利用外部存储器的保留区作为堆栈，同时设置一个POINTER寄存器，用来存放最后存入的数据字的地址。用RAM提供的堆栈实际上允许无限个子程序嵌套。此外，如果处理器能够提供一些指令使累加器和其他通用寄存器的内容按照堆栈指针中的地址被推入堆栈或由堆栈弹出，那末，实现多级中断处理就是可能的。

处理器的状态（亦即所有寄存器的内容）在中断被接受时都能保存在堆栈中，当中断服务完成后又能恢复。这种在任何时候都能保存处理器状态的功能即使中断服务程序本身被中断时也能实现。

指令寄存器和译码器

每一种计算机都以字长作为机器的特征。字长通常取决于内部存储单元和数据总线的容量。例如，某台计算机的寄存器和总线可以储存和传送8位数据，它的特征字长就是8位，称之为8位处理器。一般地，8位处理器处理8位二进制字是最有效的。因此，与8位处理器组合的存储器在每一个可寻址的存储单元中都能存储8位数据。数据和指令都是作为8位二进制数存入存储器的，或者是作为8位的整数倍，如16位、24位、32位等被存储。通常，