计算机逻辑结构

山东省电子学会出版

一九八一年十二月

计算机逻辑结构

树数　编译

申晓明　李晔　校对

山东省电子学会出版

1981年12月

序　言

《计算机逻辑结构》是一本介绍计算机结构的教材，它用深入浅出的方式从计算机系统结构、传统计算机级、微程序级、操作系统计算机级、汇编语言级、到多级计算机，系统地叙述了第三代电子计算机的发展过程和设计思想，随着电子计算机科学的迅速发展，以往只教学生们就某个机型编写程序的时代已经过去了，必须介绍更多有关计算机结构的新知识，这些知识仅在几年以前还属于研究阶段，例如，分段虚拟存贮，平行处理，微程序，自行虚拟化计算机等问题，本书试图满足这些要求，当然，更新的研究成果又在不断出现了。但本书仍然是一本较好的基础教材。

学习本书的唯一先决条件是：对计算机科学应有一点入门知识或实践经验，同时也应该懂一点使用FORTRAN，COBOL，或PL等高级语言编程序的知识。它可以作为大学教材使用，也可以作为广大科技工作者，设计人员学习电子计算机的参考读物。

译者文笔流畅，增加了读者学习兴趣，这本书的出版，为四个现代化建设中最关键的一门科学，电子计算机，提供了学习工具，它是值得欢迎和推荐的。

王运丰

1981年8月10日于北京

目　录

第一章 概 述

数字计算机是利用赋于它的执行指令代替人解决问题的一种机器。描述如何解决相应问题的指令序列称之为程序。每条指令由电子电路直接执行。通常将指令分为三类：

运算型指令：如两数相加；

逻辑型指令：如判断一个数是否为全0；

传送型指令：把信息从存贮器的一部分传到另一部分。

计算机由基本指令组成语言，利用这种语言进行人机通信。这样的语言称之为机器语言。

在设计一种新的计算机过程中，设计人员首先要决定在机器语言中包括那些指令。在满足要求的情况下，指令力求尽量简单。机器语言越简单，硬件造价越低廉，维护也越方便。但是要用简单的机器语言解决实际问题需要做大量的复杂而繁琐的工作。为了解决这个矛盾，人们设计了一种新的语言。利用这种语言解决问题比用机器语言更方便。机器语言称为L_1，而这种新的语言称为L_2。

用语言L_2写的程序有不同的执行方法：第一种方法是把程序逐条变成L_1指令序列，而后由计算机执行L_1程序，这种技术称为“翻译”；另一种方法是用L_1写出程序，把L_2程序做为它的输入信息，每条L_2指令都用一组等价的L_1指令序列来执行，边翻译边执行，这种方法称为解释法，用L_1写的程序称为解释程序。

翻译法和解释法是十分相似的，其相似之处是：L_2指令都要用L_1指令序列来执行；其不同之处是：在翻译法中，全部L_2程序都首先转换成L_1程序，这样L_2程序就不使用了，而后执行新的L_1程序就行了。在解释法中，每条L_2指令被检查和译码后就要直接执行。这两种方法都得到广泛的应用。

为了方便起见，我们可以不考虑翻译和解释的过程而假设存在一种虚拟机器，它的机器语言是L_2。如果这种机器可以廉价创造成功，就不需要再有L_1，可以把执行L_1程序的机器取消。人们可以写出由计算机直接执行的L_2程序。如果直接执行这种程序因耗费太大而无法实现，就可以通过翻译法或解释法由直接执行L_1程序的机器来完成。虚拟机器是一种假想的而又能实现假想效果的机器。

在实际应用中发现L_2语言与L_1语言相比並没有太大的差别。用L_2语言仍然达不到方便地表达程序人员意图的要求，这就促使人们进行新的探索，创造一种新的语言，它比起语言L_2来更接近于人，而不接近机器。这第三种语言我们称为L_3。人们可以用语言L_3写程序，如同机器语言是L_3的计算机真正存在一样。L_3程序可以翻译成L_2程序，或者由翻译员写成L_2后执行之。

这样我们就发明了一个语言系列，每一种语言都比它先前的语言对人更方便更合适，每一种语言又以它先前的语言做为基础。按语言系列就形成了计算机的分层结构。如图1-1所示，底部的语言最简单，顶部的语言最复杂。

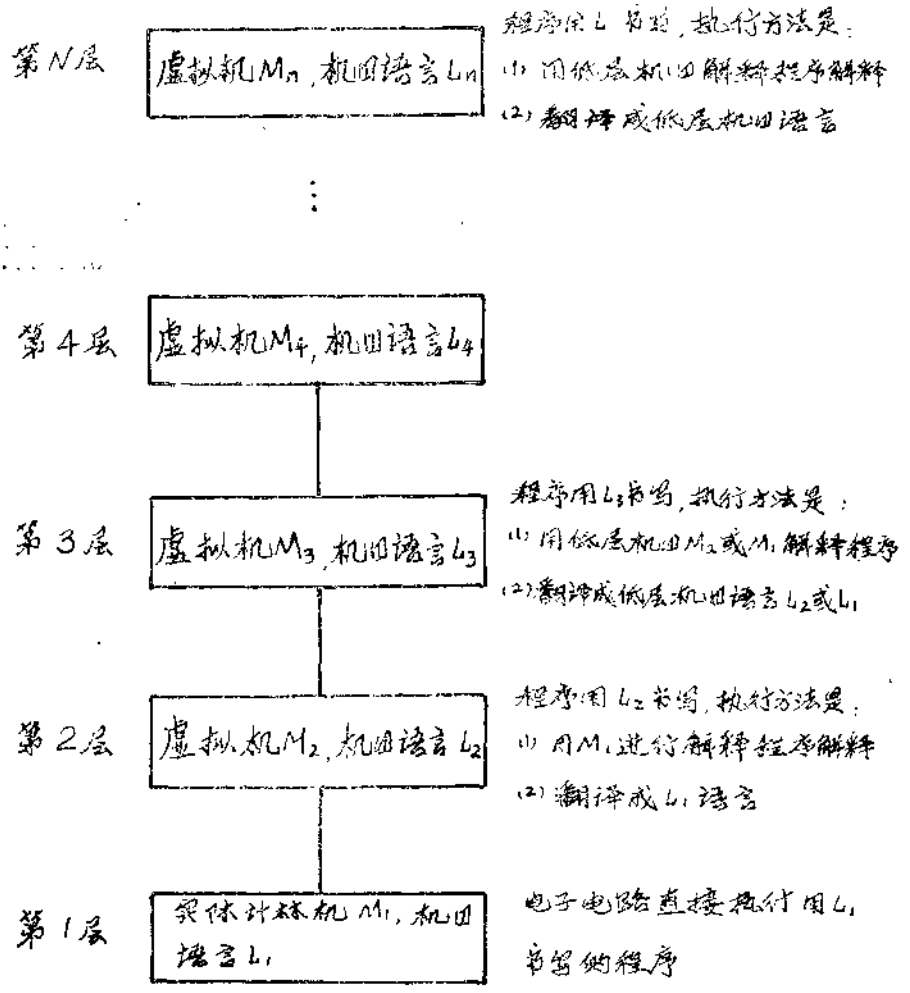

图 1-1 多层机

1.1 语言，层次和虚拟计算机

语言和虚拟机器之间存在着重要的关系。每种机器都对应于一定的机器语言，这些语言由全部机器指令组成，机器将执行这些指令。反之，语言也决定机器，机器必须能执行用某种语言写成的全部程序。然而，一些由语言决定的机器太复杂，甚至不能用电子线路直接构成。尽管如此，我们仍然假定这种机器的存在，这种机器就是虚拟计算机。以PL/I，ALGOL68或COBOL为机器语言的机器是十分复杂的，或许在最近几年内也会变成现实。

N层计算机可以认为由N种不同的虚拟机器所组成，每层机器都有自己的机器语言。为了描叙机器的性能，我们将反复用到“层次”和虚拟机器的术语。只有用L_1语言写成的程序才能直接由电子线路所执行，而并不需要使用翻译和解释方式。用L_2、L_3……L_N

写成的程序必须用解释或翻译的方法在低层机器上运行。

用N层虚拟机器编写程序的人员可以不管低层的解释或翻译过程。无论低层机器用何种方式来执行，在它们看来都达到了执行程序的同样效果。

但是对于计算机设计者或研究人员来说仅了解N层计算机的最高层是远远不够的，必须清楚计算机的实际工作过程，分析每层虚拟机器的功能和特点。本书的重点就是研究建造多层计算机的概念和技术。用分层结构的方法来研究计算机，这对了解计算机的原理和工作过程是十分有利的。用这种方法从事设计也会创造出性能良好的机器。

1.2 现代多层计算机

现代计算机由两层或多层组成。如图1-2所示的五层机器系统並非罕见。市场上销售的许多计算机的第一层就指令系统和总体结构来说有很多共同之处。严格说来，两种机器语言不同的计算机，其第一层也不一样，但它们有许多共性存在，虽然它们已经明确定义了，但仍能使我们抽象出共同的特征进行讨论。例如，各种机器一般都有20～30种以上的指令，大多数指令是传送型指令，少数是测试型指令，这层机器称为“微程序级”，解释程序称为“微程序”，这层机器是第一层，它是建造其它各层的基础。这层指令由计算机电子线路直接执行。

第一层机器有一种或多种翻译程序。每种翻译程序都规定了它自己的第二层语言（虚拟机器的机器语言）。第二层机器也有许多共同之处。不同厂家生产的计算机，第二层机器的相同之处多于它们之间的不同之处。在本写中我们称机器的第二层为常规机器层。

计算机生产厂家，为了推销机器，都要出版一本“机器语言参考手册”，这种手册所讲的就是第二层虚拟机器，而不是第一层实际机器。这里所谈到的机器指令系统是由微程序用解释的方式来执行的，而不是由指令化的硬件来执行的。如果生产厂家对于他们的一种机器提供了能解释两类第二层语言的能力，就需要提供两种机器语言参考手册，分别对两种解释程序进行说明。

应当注意，有些机器，特别是老的机器没有微程序层，在这些机器中常规机器指令直接由电子线路来执行，不需要进行解释，因此，常规机器层是第一层而不是第二层。尽管如此，我们仍然把常规机器层看作是第二层而不管上述的特例。

第三层通常是混合层。第三层语言中所用到的大多数指令在第二层语言中也用到（没有任何理由规定一种指令仅能被一层语言应用而不能被另一层语言应用）。此外，在这一层中还有一些新的指令，不同的存贮器结构，能够进行两道或多道程序並行操作，以及其它各种特征。第三层语言比第二层或第一层有更多的变异性。

用第二层解释程序来执行的第三层语言所增加的新设备，因历史的原因，这种新设备称为操作系统。操作系统是用户与计算机之间的接口。用户通过操作系统使用计算机，因此，操作系统是方便用户提高计算机利用率，加快计算机响应速度的一种软件。其主要功能是管理中央处理机、内存、外部设备和信息，控制作业的运行，以及处理中断等。此外，各种子系统（编译程序、编辑程序、装配程序）和应用程序皆在操作系统控制下运行。在第三层中，与第二层相同的指令直接由微程序来执行，而不通过操作系统。换

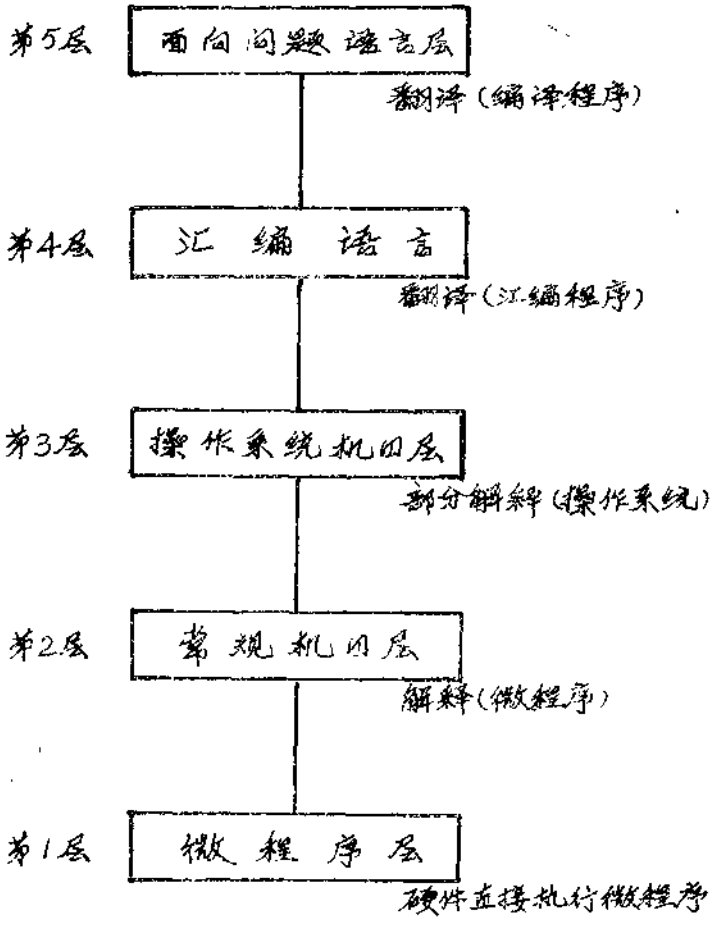

图1—2 现代计算机的五层结构。各层都由它下面的一层支持，括号内是支持程序的名称。

句话说，在第三层语言中，有些由操作系统解释程序执行，有些由微程序直接执行。我们称第三层为“操作系统机器层”。

第三层和第四层之间有根本性变化。最低的三层机器通常不是程序人员解决问题的午台。程序人员需要的是以解释程序和翻译程序为基础的更高层。解释程序和翻译程序是系统程序人员书写的，它们各具特点並能完成新的虚拟层。程序人员真正用来解决问题的是第四层或更高层的机器。

第四层机器中有方法上的变化，这种方法可以对高层计算机有所协助。虽然也有例外，但一般说来都通过翻译程序来执行，而第二层和第三层机器总是用解释方法来实现的。第四层和第五层或更高层则时常要由翻译来支持。

此外，第一层、第二层、第三层同第四层及更高层之间的区别还在于它们的语言性质，前者的语言是数字的，程序由一长串数码组成，面向机器而不面向人；后者的语言是文句格式，面向人而不面向机器。

第四层机器是汇编语言，实际上是一种低级的符号语言，更确切地说，这层机器为程序人员提供了写程序的方法，而不是作为虚拟机器而独立存在。汇编语言首先要翻译成第一、二或第三层机器语言，而后再由适当的虚拟机器或实际机器来执行。完成翻译的程序就称为“汇编程序”。汇编语言曾经是很重要的，但随着时光的流逝，其重要性正在被高级语言所取代。

解决应用问题的语言组成了第五层机器。这类语言有很多名称，包括：“高级语言”和“面向问题语言”。有几百种不同的文字语言。其中有名的语言是：ALGOL60，ALGOL68，APL，BASIC，COBOL，FORTRAN，LISP，PASCAL，RPG和SNOBOL4。程序人员用这种语言写的程序都要用所谓“编译程序”来翻译成第三层或第四层语言，在少数情况下也用解释方法。

第六层或更高层的语言是为了某些特殊应用而设计的，在应用中包含有大量的信息处理。我们可以想象，在管理、教育、计算机设计等领域中，是需要有这种虚拟机器的，这些高级的虚拟机器目前正在研制阶段。

1.3 多层计算机的历史发展述评

为了对多层计算机进行深入了解，有必要对它的历史发展概况做一简要回顾。

1940年第一台数字计算机仅有一级，同现在的常规计算机相类似。机器语言是数字形式，程序直接用机器语言书写，用开关输送到计算机内。机器指令直接由机器电路执行，这些电路（由电子管电路组成）复杂，难于设计制造。在1954年M·V·Wilkes提出了建立两层计算机结构的设想。机器设有固定不变的解释程序，用来逐条解释常规语言程序。在50年代初期出现了少量的两层计算机，到60年代这种机器就开始增多了，发展到1970年，常规机器语言用微程序解释的方式就得到了广泛的应用。

汇编语言和编译语言在50年代初就开始研制了。当时主要是为了简化编制程序的过程。程序人员编制程序、输入程序，修改程序所花费的时间是大量的，机器经常处于停机待用状态。到1960年，企图减少机器停机时间和提高操作员的工作自动化的努力，有了初步成效。称为操作系统的程序长期在机器中保存。程序人员提供的控制信息同程序一起送进机器并执行。图1-3给出了早期操作系统FMS(FORTRAN监视系统为IBM709所采用）的工作情况。

操作系统读*JOB卡片，它上面的信息是用于内务整理的，而后读FORTRAN卡片，它上面的指令是用来从磁带上读FORTRAN编译程序的。当编译程序读写之后，又回到操作系统并读DATA*卡片，按照DATA卡片上的数据进行程序翻译。

从上述这个例子中我们可以看出，操作系统使操作员工作自动化了，操作系统正是由此而得名的。这也是研制虚拟机器的第一步。我们可以认为FORTRAN卡片是虚拟编译程序指令，DATA卡片是虚拟执行程序指令。当然，每层仅有两条指令的机器是不多的，但这确是虚拟机器产生的启蒙点。

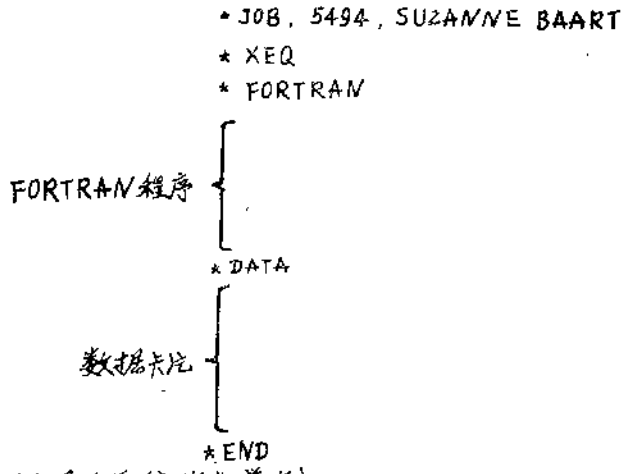

图1—3. FMS操作系统作业举例

到后来，操作系统发展得更为复杂完善。在常规机器层上增添了新的指令和性能，最后发展成一个新的机器层。在这个新的机器层中，有些指令是和常规机器层一样的，也有一些，特别是I/O指令确是完全不一样的。这些新的指令统称为“操作系统”，或“访管”，这个术语一直沿用到现在。

对操作系统也通过其它途径进行了研制。卡片读出机和行式打印机通称为“分批系统”，是为几个用户分别输入程序和输出结果的。通过终端设备可以成批地向计算机输入数据和程序，在计算机上处理后，计算的结果成批地送回终端。

到60年代初期，操作系统发展到能使程序人员直接同计算机通信。在这些系统中，遥控终端通过电话线直接联到中央计算机。程序人员可以直接给出程序並得到计算结果，他可以在办公室里、库房里，或任何其它设有这种终端的地方同计算机通信。在这种系统中，允许多个用户同时使用一台计算机设备，每个用户通过控制台或终端采用问答方式控制程序的运行，系统把处理机时间轮流地分配给各联机作业，每个作业只利用极短的一个时间片，如果在时间片结束之前计算机还未完成，该程序就被时钟中断，等待下一轮再计算，此时处理机让给另一联机作业使用。这样，各用户的每次要求都能快速响应，给每个用户的印象是他自己独占该计算机。按这种方式工作的系统称为“分时系统”。

本书对操作系统论述的重点是具有第三层机器特点的解释指令，而不是第二层指令，对分时系统也不予赘述。虽然在今后的讨论中不再多加说明，读者也应清楚，操作系统实际上比第三层程序的解释部分有更为丰富的内容。

1.4 硬件、软件，多层计算机

用计算机的机器语言书写的程序可以由计算机的电子线路直接执行，而不需要进行解释或翻译。电子线路、存贮器和输入/输出设备统称为计算机的硬件。硬件是有形的，可触知的实体组成，如集成电路，印制板，电缆线、电源，存贮设备，卡片读出机，行

式打印机和其它终端设备。硬件不是抽象的思维和指令。硬件用什么样的元件组成及其工作原理属于电子工程师的工作范畴。

软件是相对硬件而言的，它是计算机系统中用户共同使用的一组程序及有关资料，是计算机系统日常工作中不可缺少的组成部分，或为了扩大计算机功能，提高计算机效率所需要的程序。例如：库程序，汇编程序，编辑程序，诊断程序，控制程序，专用程序包，操作系统，数据管理系统，各种维护和使用手册，程序说明和框图等。表示程序的媒介有穿孔卡片、磁带、相片和其它材料。软件的本质是组成程序的指令，而不是记录程序的物质本身。

本书的中心思想是反复强调硬件和软件在逻辑上是等价的。

软件的任何一种操作功能都可以直接用硬件来实现，而硬件的功能同样也可以用软件来实现。一种功能到底用硬件来实现还是用软件来实现，判断的依据是：造价、工作速度、存贮容量、可靠性和灵活性。没有一个固定的法规或一成不变的原则，决定X必须用硬件，y必须用软件，这主要由设计者按照不同的情况来决定取舍。

在早期计算机中，硬件和软件的分界线是很鲜明的。硬件执行少量的简单指令，如加法、转移。如果需要做乘法，程序人员就需要编写乘法子程序，后来这些操作都可以用乘法器来直接执行，也即由硬件完成。程序人员为了提高运算速度，往往希望直接用速度更高的硬设备。

多层计算机和微程序设计的发展，又出现了硬件软化的趋势。在古老的计算机中加法无可怀疑地要用硬件来执行。但是在微程序计算机中，加法指令要由第一层的微程序来解释。在微程序的解释过程中包括：取指令，判断它的类型，取数据，进行加法运算，最后存放结果。这是从硬件变到微程序的一个例子。再强调一次，必须用硬件或必须用软件的规则是不存在的。

在研制多层计算机的过程中，设计者必须决定在每一层机器上安排什么样的任务。实际上这是很早就提出来的老问题。但现在我们所感兴趣的是原来常规机器程序在现代计算机中哪些用硬件执行，哪些用软件执行。这些指令有：

1．整数乘法和除法指令；
2．浮点算术指令（见附录B）；
3．双字长精度指令；
4．转子或返主指令；
5．高速循环指令；
6．计数指令；
7．处理字符串指令；
8．高速处理阵列（变址和间接寻址）；
9．动态变化存贮功能（再定址设备）；
10．时钟定时程序；
11．I/O中断系统；
12．停止主程序启动开关量输入。

这些讨论表明，硬件和软件的界限是任意的，而且是经常变化的。今天用软件，明