

解释性计算机

B1700 / B1800系列的体系
结构和程序设计

埃利奥特·艾·奥盖尼克 著
詹姆斯·埃·海因兹

Interpreting Machines:
Architecture and
Programming of the
B1700/B1800 Series

Elliott I. Organick
James A. Hinds

NORTH-HOLLAND·NEW YORK

73.27/2

477

201

解释性计算机

B1700/B1800系列的体系结构
和程序设计

埃利奥特·艾·奥盖尼克^著
詹姆斯·埃·海因兹



中国外贸咨询与技术服务公司译

1116010

前 言

本书是美国著名电子计算机专家 Dr. Elliott Organick 教授及 Burroughs 公司工程师 Mr. Hinds 所著。详细讨论了可变微程序结构的原理及其优越性。Burroughs 公司是可变微程序的先导者。特别委托中国外贸咨询与技术服务公司翻译、印刷、并分送本书予各有关单位。希望能和中国的计算机工作者共享这项成就。

美国宝来计算机公司
中 国 办 事 处
北京友谊宾馆7222号

目 录

序 言	(1)
第一章 通用主机系统	(4)
1.1 B1700系列计算机的存储结构	(7)
1.2 B1726的存储器	(10)
第二章 B1700——一台解释机	(13)
2.1 指令译码	(16)
第三章 B1726微指令处理器的结构	(19)
3.1 取数据及形成地址	(19)
3.2 数据的检验和处理	(31)
3.2.1 运算功能或“功能部件”	(33)
3.2.2 算术运算的某些细节	(35)
3.3 指令译码	(37)
3.4 控 制	(41)
第四章 B1700的运算环境	(47)
4.1 宝来公司关于“代码文件”的概念	(50)
4.2 构造一个运算环境	(51)
4.3 一个完整的MIL程序的实现	(52)
4.4 MIL的说明部分	(57)
4.5 文字表	(65)
4.6 McMIL语言中的输入/输出	(66)

4.6.1 文件说明.....	(72)
4.7 装入程序(细节)	(77)
第五章 解释程序的结构	(82)
5.1 对故障和中断的检测和响应	(82)
5.2 主机环境	(87)
5.3 外壳的概念	(89)
5.4 用由顶到底的方法考查解释程序的结构	(94)
5.4.1 目标机存储器的表示法.....	(95)
5.4.2 ADD例行程序的MIL编码	(107)
第六章 数据处理用的MIL编码	(112)
6.1 24位功能部件的算术操作.....	(112)
6.2 VALIDATE·DECIMAL 验证十进制: 一个实用的例行程序的专例研究.....	(115)
6.3 BINARY·TO·FA 将二进制值转换成绝对地址送FA	(130)
6.4 ADDRESS·TO·BINARY 地址转换成二进制.....	(133)
6.5 EFFECTIVE·ADDRESS有效地址	(137)
6.6 ADD例行程序 加法例行程序.....	(146)
6.7 UNPACK·AND·WRITE 分开并写入	(156)
6.8 本章小结.....	(158)
第七章 错层控制存储器	(160)
7.1 对H存储器使用的控制.....	(162)
7.2 从H存储器或从G存储器取微指令.....	(165)
7.3 将列表数据嵌入MIL程序.....	(168)
7.4 从G存储器向H存储器传送微编码程序.....	(172)

7.5 将控制传递到另一个解释程序	(176)
7.6 小 结	(181)
附录A —— 简略的MIL参照指南	(183)
索 引	(183)
1. 表示法	(186)
2. 可执行MIL语句	(187)
3. 不可执行MIL语句	(223)
4. 特殊MIL表达式	(230)
附录B —— B1726参考资料摘录	(232)
1. B1726的寄存器摘要	(232)
2. IF 语句的可检测位	(238)
3. 微指令: 语法和语义	(240)
附录C —— McMIL和SMACK的用户手册	(279)
1. McMIL语句的语法	(280)
2. 为SMACK处理程序的运行而设置的 McMIL语句	(280)
3. 用于形成资料的McMIL语句	(281)
4. 用于形成印出格式的McMIL语句	(282)
5. 用于存储分配和定址的McMIL语句	(282)
6. 用做调试的McMIL语句	(283)
7. 与MCP接口用的 McMIL语句	(284)
8. 为与MCP通信的McMIL语句	(286)
附录D —— 装入程序入门	(292)
1. JCL命令序列	(293)
2. 装入程序卡片的语法	(294)

3. 举 例	(298)
--------------	-------

附录E——用McMIL编写的SAMOS解释

程序摘录	(299)
1. 输入到McMIL处理程序的SAMOS解释 程序的源程序	(299)
2. SAMOS解释程序的MIL汇编程序	(310)
3. 为SAMOS解释程序生成编码文件的装入程序的 源程序	(326)
4. 装入程序的输出程序	(326)
5. 将由SAMOS解释程序执行的一迭简单 的数据卡片	(327)
6. 当处理第五部分中的一迭数据卡片时, 由 SAMOS解释程序产生的输出	(327)
7. 对SAMOS解释程序简化版本的 一系列改进	(328)

附录F——SAMOS计算机..... (330)

1. SAMOS执行周期	(332)
2. 指令系统	(333)
3. 出错条件	(334)
4. SAMOS装入程序	(334)

序 言

宝来 (Burroughs) 公司的 B1700 系列计算机呈现了体系结构的一种新风格。人们称这些机器为可定义字段解释性机器。这些机器的正常执行方式是对其它机器 (虚拟机器或实际机器) 的解释。为了解释其它计算机系统而设计的一个系统应当具有灵活的存储器存取机构。这样, 在程序员的控制下, 可以对任意长度的位串进行存取和加工。B1700 系列的可定义字段特性支持了指令的有效解释和促进了存储器的有效使用。W.T.Wilner 先生 1972 年在一系列文章中综合概述了这些特性 (见 AFIPS Conference Proceedings, vol.41, Part1, "Design of the B1700" ("B1700 的设计") pp. 489—497; "B1700 Memory Utilization" ("B1700 存储器使用") pp. 579—586; 以及 *IEEE Computer Society* 的 COMPCON 72 "Microprogramming Environment of the Burroughs B1700" ("宝来 B1700 微程序设计环境") pp.103—106)。

对于计算机科学方面的教育和研究, 尤其软件工程的研究, 包括对新型计算机、专用计算机和数据库系统的设计和评价、以及对软件可移植性的研究, 象 B1700 及其后继产品这类革新性系统, 是有吸引力的实验室设备。

本书描述了宝来公司 B1700 系列的体系结构, 主要侧重于 B1726 计算机系统, 其内部结构, 以及如何对它编制程序来仿真其它计算机系统。对于某些计算机系统专家来说, 如果他们主要探讨一种计算机结构优于另一种结构的理由, 那末, 这本书可能只具有有限的吸引力。我们准备论述 B1700 的相对优缺点, 也不准备讨论诸如为什么解释是重要的, 什么时候宁可采用解释面不

采用当今流行的、传统的、基于编译程序的通用系统的方法等有趣问题。我们既不探究解释的历史，也不探究其潜在的前景（只是暗示多级解释是有前途的）。最后，我们也不是为了推荐B 1700体系结构的其它应用，如数据库计算，而只是想帮助那些已经动了心想深入到B 1700“内部”的人，以及想帮助那些想获得在微代码一级上实现解释程序这方面知识的人。

本书是由一套为较高水平的计算机科学方面的大学生所写的讲义发展而成的，这些大学生已经具备了传统计算机的系统结构和低级语言程序设计的预备知识。犹他（Utah）大学的学生们在软件实习课程中已经使用了这套讲义，该课程的主要目标是为一台十分简单的计算机（如PDP-11）产生一个微代码的仿真程序。对于那些想用B 1700进行研究工作的更高水平的学生，这套讲义也一直是自学的有用材料，它也可用来补充和代替现有的参考手册和文献资料。

在课文中，我们介绍和使用了程序设计语言McMIL。它是MIL语言的提高版本（MIL是一种微指令语言，宝来公司为这种语言提供了汇编程序）。MIL的超集合McMIL包含了一些语句类型，程序员可以用这类语句来简化MIL指令序列的生成，形成MIL解释程序与系统环境的正确接口。〔例如，实现中断处理，输入输出管理、文件系统服务和进程转接等〕。

本书共由七章和若干附录组成。前三章侧重于介绍作为解释机器的B 1700系列的体系结构，B 1700处理器的内部结构和它的（符号）微代码一级的机器语言。接着的三章指出了编写微代码一级的程序的路子。一台假想的计算机SAMOS（附录F给出了它的概要）的仿真程序是进行专题研究的一项重要工具。在四、五、六这三章中，透彻地说明了汇编语言MIL和对应的McMIL作了哪些提高。高级语言程序设计的方法论，这包括逐步分解、清晰的结构以及精细的资料说明等。在将相对抽象的术语所表达的问题，翻译成具体的McMIL程序的过程中用到了这些方法。附录

A、B、C的目的是作为MIL的参考手册，这三个附录分别介绍了实际计算机系统的寄存器，指令的语义，以及McMIL的扩充。

（附录D为安排一个解释程序的检测运行，提供了附带的参考材料，附录E给出了小型SAMOS解释程序，以及取样检测运行的程序清单。这个小型解释程序可以当作一组练习用，也可用作学习研究的工具，和作为某些有趣的修改和提高的出发点）。第七章对用作微程序处理器的B1726的控制结构的细节进行了考察。

这七章有意识地侧重于用来设计和实现解释的B1700系列的现存硬件，在很大程度上，与宝来所提供的支持软件无关。可以预期，另外一本侧重于讲述宝来公司软件结构和功能的书是会有用的。软件结构和功能是指操作系统（MCP）；关键的核心模块（称为GISMO），它的作用是输入输出设备的驱动程序、进程转接程序、输入输出缓冲区的收发程序、以及与MCP的接口程序等。这样一本书将有助于读者，在学会了编写和调试程序之后，进一步洞察支持这些程序执行的（系统所控制的）环境。

作者对犹他大学帮助过我们编写这本书的同事们、同学们和秘书朋友们的支持表示衷心感谢。我们还为得到了宝来公司的支持而感到莫大荣幸，宝来公司的许多人士帮助我们在犹他大学开展了这个设计项目，并促成了这本书——它是副产品之一，希望它是一个成功的产物。让我们对诸位深表感谢，尤其是R.R. Johnson, R.D. Merrell 和 R.S. Barton，没有他们的帮助和信任，这个项目是不可能成为现实的。这三位都是宝来公司工程结构部的成员，他们很早就推崇B1700，认为这是一个值得给予重大注意的系统，是一个用于计算机科学和工程研究方面的系统。本书的出版得到了宝来公司的许诺。

E.I. Organick

犹他州，盐湖城

J.A. Hinds

加利福尼亚州，戈尔他

第一章 通用主机系统

常规计算机系统(冯·诺依曼计算机系统)的一个重要特点是使用控制机构或处理器对由存储器取出的指令序列进行译码和加工(如图1.1)。处理器中一般至少具有两组寄存器,一组用于控制,另一组用于“信息处理”。第一组寄存器主要用于控制程序中的指令执行顺序并对每条指令进行译码以备正确执行。第二组寄存器与第一组几乎无关,它们用来完成译码后指令的执行。一般讲,执行是指对存储器取(或送)数据;或者对取自存储器的数据或对前面指令的执行结果进行检查和加工。

图1.1中的计算机示意图显然是不完整的,因为它还缺少与外部环境的联系。计算机与“外部”存储设备之间的信息交换通道由输入/输出(I/O)控制器及I/O设备提供。外部存储设备可以由多种介质构成(如磁带,磁盘,显示器,打印机等等)。在当前讨论中,暂且不考虑I/O与外存储器之间的传送。

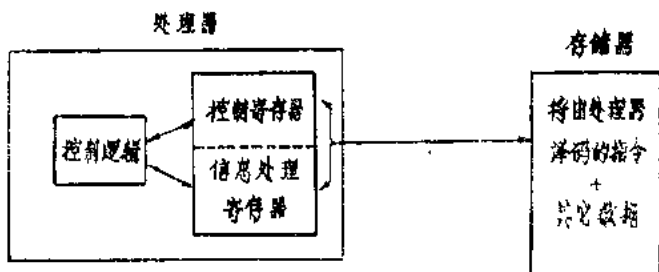


图1.1 典型的计算机体系结构

对计算机的每一条指令进行实际译码和处理是最基本的操作。程序设计者在通常情况下不能影响基本操作的执行方式。在早期的计算机中这些基本操作是由硬件完成的。在不少近期设计的计算机中，这些基本操作是由一串由硬件解释的微步进或微程序实现的。这些微程序常常做得使程序员无论如何也访问不到，因此，一个程序设计者可以使用的那些指令的解释仍然是基本的操作，也就是说，程序设计者无法影响机器的解释机构。

虽然这一类计算机的程序设计者不能改变机器的基本特性，但是作为一种辅助手段，他们可以编写一个模拟（或是仿真）程序来解释其它机器的程序。这种由程序提供的解释器的逻辑完全由用户控制。用户不仅可以改变译码机构的节拍，而且可以随意选择执行逻辑。

用户可获得的重新设计的机会是形形色色的。他可能想模拟一个与某一个基本计算机只有很小一点区别的机器，比如，只求在其指令系统中增加少量指令，或是对现有指令的解释做一些修改。而在另外一种极端情况下，用户可能想模拟一个具有完全不同的指令系统的机器。其指令格式与“主”机的相差很大，而且语义也不相同。例如，他可能想用PDP-9对PDP-15，SAMOS¹⁾机或FORTRAN机进行仿真。第一个机器（PDP-15）只不过是PDP-9本身的一个扩展机（只多几条新的指令）。第二个机器（SAMOS）的语义虽然与PDP-15有很大不同（具有十进制运算而不是二进制），但其指令的语法及语义功能却与PDP-9差不多。因此，SAMOS及PDP-9的指令格式均取固定长度，都只有几个固定的子字段，都用变址寄存器等等。另一方面，FORTRAN机器却具有可变的指令格式，可变数量的子字段，而且其语义的复杂程度要比PDP-9大得多。

图1.2给出了二级主/客系统的示意图，其中包括一个主机或H机，它具有解释另外一个机器系统——客机G的功能。图中显

1) 一种在某些计算机科学教科书的绪论中用到的假想机（见附录F）。

然在计算机结构中引入了递归概念²⁾。我们先从里向外进行研究。H处理器或主处理器由控制逻辑及一些存储器（寄存器组）构成。H机由H处理器及其指令存储器（H存储器）构成。而H机又具有另一台G机处理器的功能，因此H机实际上是G处理器。给H机加上“外”存储器就构成了一台新的机器（G机）。G机的外存储器在图1.2中没有表示出来，但其存在性是可以设想的（正如图1.1中描述的机器的外存储器一样）。就原理上讲，上述递归还可以扩大，因为G机又可以做为另一个GG机（客机的客机）的处理器并与GG机的程序存储器连接。

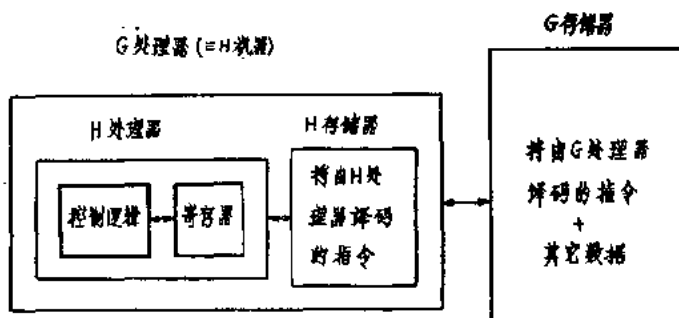


图1.2 二级主/客系统结构

在建立这类解释系统时总有实际的折衷考虑。如果主机的指令系统及寄存器与其客机的极不相同，则解释G语言指令的H语言的子程序可能相当长（而且要占用H机存储器的相当大空间）。不仅如此，在H机上用于解释G语言指令序列所花的时间可能远远超过在同一个H机上执行“等效的”H机指令序列所花的时间。G时间/H时间往往可以达到10到100。即使如此，建立在常规计算机上运行的解释程序还是广泛地受到重视。

由于任何一台机器原则上都可以编码成任何一台客机的主机，因而同一台主机在不同的时刻可以方便地起到几种不同客机中某台处理器的作用。一台H机的后备存储器中可以存放不同客

2) 处理器可以看作具有递归结构的观念是首先由Robert S. Barton提出的。

机的解释程序。这些解释程序可以根据某种调度规则调入或调出H存储器，这样H机在各别的时间片里就可以先按某个G处理器的方式工作，然后再按另一个G处理器的方式工作。根据“调入调出”所用技术，时间片可以是几天、几分、几秒（或是更小）。但是，不管一个解释程序处于工作状态的时间片是长还是短，现在应当容易地承认这样一个事实了：任何一台主机都可以作为一台通用主机，也就是一台可以模仿多种客机的主机。

虽然如此，但几乎所有已设计出的计算机系统都不是作为主机使用的，更不要说作为多种其它机器的通用主机了。B1700系列计算机确实曾经是为了起通用主机的作用而搞的一个系统。当我们研究这个计算机系列的时候，我们将看到它的各种特性是以何种方式来支持上述功能的。

由宝来公司生产的B1700系列计算机最近已扩充成高一档的型号，叫做B1800。在本书中除非专门提到某个型号外，我们将用‘B1700’这个词来泛指各种新增加的系统型号。这些系统的主处理器的机器语言(H语言)都用同一组基本的16位微指令来定义。不仅如此，这些系统还具有基本相同的内部逻辑结构，仅仅取微指令的机构有所不同。B1700系列机的解释程序所执行的程序和数据存放在一个存储器中，该存储器可看成是一个按序排列的字段（或位串）集合，每个字段的长度是可定义的，因此B1700也称做“可定义字段解释机”。

1.1 B1700系列计算机的存储结构

为了满足通用主机的要求，H机的处理器必须能够访问每台客机的解释程序的微程序。实现上述要求的一种办法是假设H处理器实际上访问几个H存储器，每个H存储器中存放一台客机的解释程序。当然，这样一来处理器就必须能由一个H存储器转接

到另一个，这样系统就能在几个工作的解释程序间多道工作。过去15年中存储技术和存储管理技术的发展提供了一些价格合理的途径来实现这类系统。在B1700系列中采用了三种相应的办法，每种办法用于这个系列中的一种型号。这些型号是B1710，B1720和B1800。

在B1710中采用了第一种办法（最简单，硬设备最便宜，速度最慢）。其中（见图1.3）主存储器被划分为几个区，某些区作为H存储器，另外一些区作为G存储器。在作为H存储器的区中存放构成G机解释程序的微程序。图中只画出了一个G存储器和一个H存储器，但在原理上和实践上主存储器足以容纳几个G和H存储器。

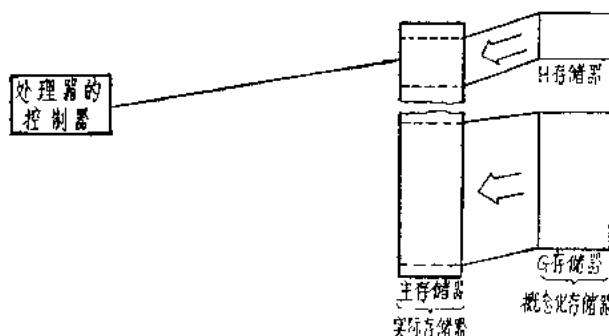


图1.3 B1710处理器访问H存储器的代码

在每个H存储器中存放构成一台G机解释程序的微程序。B1700处理器可以初始化成从主存储器的任何一个H存储器开始取出和执行微指令，并使用一个G存储器作为其工作区。在任何给定时刻，B1700处理器只访问主存中的一个H存储器和一个相应的G存储器。解释程序的转接意味着重新置位B1700的寄存器组，以便访问另外一对H存储器/G存储器。

B1800采用了类似的原理在主存储器中实现H存储器和G存

存储器，但通过采用“高速缓冲存储器”（见图1.4）的办法可以更快地取到微指令。高速缓冲存储器保存根据需要由主存调入的微指令块。若微指令在高速缓存中（正常情况如此），则访问这条微指令的时间要比访问一条先要由主存调入高速缓存的微指令所花的时间几乎快一个数量级。高速缓存的容量足够存放一个完整的解释程序，或者至少可以存放执行频率最高的那一部分程序。

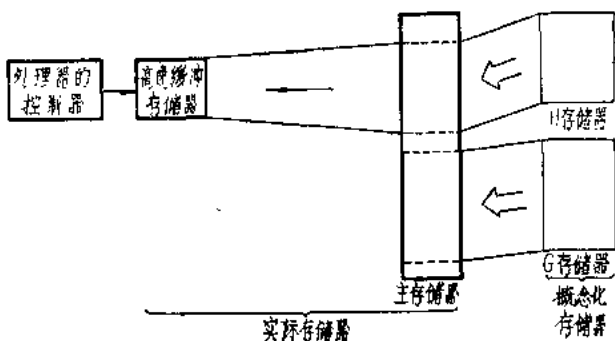


图1.4 B1800处理器访问H-存储器的代码

B1720采用了一种不太考究，但十分有效的加快取微指令的方法：在系统中加入了一个叫作高速控制存储器的第二个存储部件（见图1.5）。这个部件足够存放一个解释程序的最常用部分，如果空间允许的话还能存放多个解释程序的常用部分。依照快速控制存储器可用容量的大小，H存储器的一部分与快速控制存储器对应，另外的部分由主存对应。B1720处理器为此设置了专门的基寄存器，以便确定取下一条微指令时所需经过的访问通路。这一通路或者引向快速控制存储器（通路A），或者引向主存（通路B）。在其它条件相同的情况下，随着高速控制存储器或高速缓存的容量分别减少到零，B1720及B1800的性能会逐渐降到B1710的水平。本书第七章将对此进行详细讨论。

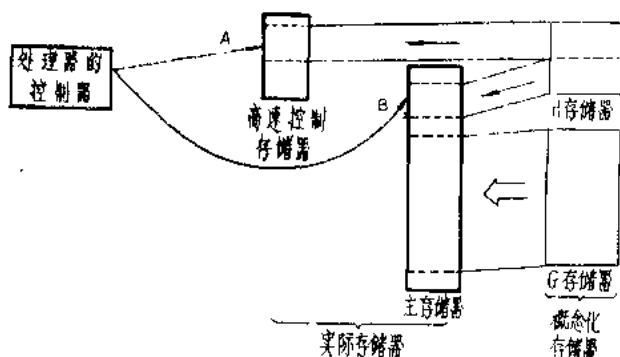


图1.5 B1720处理器访问H存储器的代码

B 1710, B 1720及 B 1800除上述区别外还有一些不同, 这些不同对本书要讨论的内容来讲都是不重要的。但是, 为了避免混淆, 我们将尽可能地指明所讨论的型号。由于作者在 Utah 大学主要接触的是 B 1720型号, 特别是它的变种 B 1726, 因此本书将详细讲解 B 1726。但是, 与此同时也要对有关型号进行广泛地讲解。当讨论到其它型号时, 我们将小心地说明。

1.2 B1726的存储器

我们现在以 B 1726机如何使用存储器以收到通用主机效果为例, 进一步加深我们的理解。宝来公司作为主存储器使用的 S 存储器 (字符串存储器) 一般容量至少大于 64K 字节 (2^{16} 位)。宝来公司作为微指令存储器 (M 存储器) 使用的快速控制存储器, 一般容量在 4K 到 8K 字节之间, 这一容量足够存放 2048 条 H 语言 16 位的微指令。

让我们首先假定 B 1726 只忙于处理一台 G 机的程序 (两台或更多台不同 G 机同时工作的一般情况将在后面讨论)。我们再进行