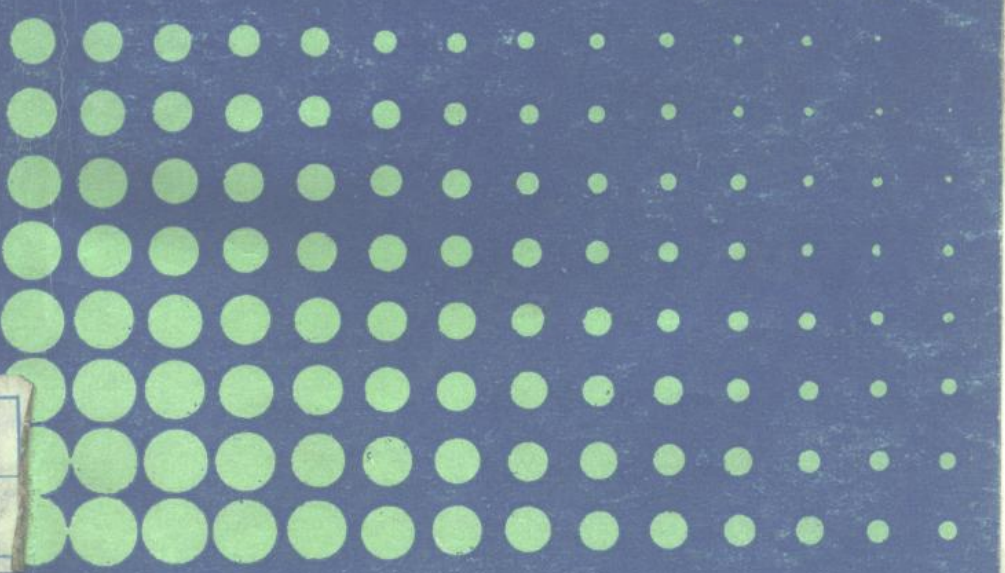


计算机 硬件/软件 系统结构

[美] 温·托伊 本杰明·齐 著

方存正 张汝杰 朱学林 译

陶增乐 审校



上海科学技术文献出版社

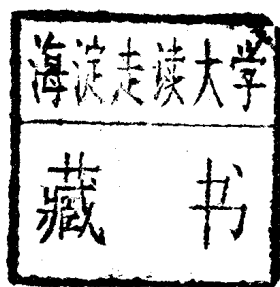
TP303
TY/1

计算机硬件/软件系统结构

[美]温·托伊 本杰明·齐 著

方存正 张汝杰 朱学林译

陶增乐 审校



上海科学技术文献出版社

119536

JS19/12

计算机硬件/软件系统结构

[美] 温·托伊 本杰明·齐 著

方存正 张汝杰 朱学林 译

陶增乐 审校

*

上海科学技术文献出版社出版发行

(上海市武康路2号)

全国新华书店经销

宜兴市第二印刷厂印刷

*

开本 787×1092 1/32 印张 15.5 字数 374,000

1990年9月第1版 1990年9月第1次印刷

印数: 1—2,900

ISBN 7-80513-618-1/T·166

定价: 7.90元

《科技新书目》222-351

译者序

60年代后,计算机系统结构领域的研究集中在对硬件和软件之间复杂性的最佳区分方面的权衡折衷的重新估计。即计算机系统结构提出了一个根本性的问题:硬件-软件的边界应当在哪儿?系统结构设计师面对着一个概念差距的严峻问题。这种概念差距的产生是因为,程序员考虑问题多以高级语言这个概念层次上进行,而硬件系统设计师却基于一个完全不同的概念层次来考虑。

以往出版的大量有关计算机系统结构的教材或专业书籍几乎都是偏重甚至局限于机器的硬件组织,而远没有从计算机硬件-软件这样一个系统的紧密结合、相互影响、协同工作的角度来考察和阐明问题。温·托伊和本杰明·齐的《计算机硬件/软件系统结构》一书正是我们所看到的第一本系统地从这方面总结了近期研究成果的教材。作者在贝尔实验室等公司长期从事实际工作,对系统结构设计有丰富的经验,故本书不仅资料详实,而且观点鲜明。每章末还列有大量的习题和参考资料,可供教学参考。

这里我们翻译本书,推荐给大学高年级学生和研究生,以及计算机硬件设计师和系统软件设计师。读者会从本书中获得解剖分析、消化理解和设计构造现代计算机系统的有益知识。

限于篇幅,我们略去了原书中的第九章“AT & T 3B20D处理机”及各章的参考文献。第九章以3B20D处理机为实例,覆盖了前面各章所讨论的系统结构的许多问题,包括了作者的许多

经验之谈。

本书第二、三章由方存正副教授译、第一、四、八章由张汝杰副教授译、第五、六、七章由朱学林讲师翻译。陶增乐教授及方存正副教授对全书作了认真细心的译校、审核。我们对陶增乐教授表示衷心感谢，译文中错误、不妥之处，敬请读者指正。

原 序

已出版的计算机系统结构的书籍，其大多数是为计算机工程师所用。书中多般讲授如何来设计算术逻辑单元、微程序控制器、直接存贮器存取机构以及存贮器系统，讲授这些硬件部件如何在计算机中组织、连接在一起。但计算机系统结构(Computer Architecture)这一术语已经发展成比单纯的硬件结构具有更广泛的含义。对于计算机系统设计师，他们的职责是设计一个计算机系统，使这个系统的操作原理和规格性能以最好的性能价格比来满足市场的需求。为了达到一个成功的设计，计算机系统设计师应当了解这系统的所有部分是如何工作并相互配合的。他们必须考虑现有的技术和它们的物理设计实现。计算机系统设计师需理解各种系统硬件和软件的配置，它们如何互相联系，它们相互之间以及与“外部世界”如何连接。

计算机系统设计师还必须担保系统的可维护性、容错能力和差错恢复策略。他们必须考虑有助于系统设计的所有问题的代价、局限、好处和可行性。例如，系统设计师必须确信，他们所提出的指标能够由现有的人员在可能的时间里实现，他们为解决种种难题所提出的各种出色的办法不会昂贵得令人怯步。而且，系统设计师还必须明白各种问题间的联系：为一个问题所作出的设计决定如何影响到其它的问题。无疑，开发一个完好的计算机系统结构并非是一个无足轻重的任务。可选方案很多，权衡难决，关系复杂。系统设计师很难能确准自己的选择恰是正确的。他们最好的判决只能基于经验、分析和仿真工具，并且最

重要的是,基于对整个系统将如何在一起工作的理解。

《计算机硬件/软件系统结构》是一本教科书。它将帮助计算机系统设计师在今天的开发环境中来设计系统。它用作为高年级大学生和研究生有关课程的课本,也可作为计算机系统设计专业人员的参考书。从本书中,计算机硬件设计人员和系统软件设计人员都可以对他们各自的设计如何与其对手的设计来接口连接,获得更好的理解。

为获取《计算机硬件/软件系统结构》一书所呈献的材料的所有好处,读者应具备一些有关程序设计语言、操作系统,以及计算机硬件结构的知识。许多课本都分别论述了这些领域。本书把这些各别领域的概念结合成一体。例如,本书阐述了程序设计语言中的概念如何影响计算机硬件设计和结构,操作系统诸功能如何与硬件相互影响,系统结构又如何影响到系统的维护。本书的重点不在于一个处理机是如何工作的,或者一个进程管理程序是做什么的,而在于它们是如何在一起工作来完成所预期的总的处理功能。

在本书写作中,我们广泛吸取了文献资料,并且从我们自己在AT&T(美国电报电话公司)贝尔实验室,身为计算机系统设计师在设计3B20D计算机,以及以前为高效电子交换系统设计处理机时所获得的经验中吸收了材料。本书从对计算机系统结构这一术语的讨论着手。计算机系统结构包括硬件部件和软件部件两者;两个部件的组合才构成了整个系统。第一章复习了在硬件设计中基本的逻辑元件,并渐次地述及了一些主要部件:控制器结构、算术逻辑单元、存储器组织,以及输入/输出结构。在器件、功能和系统几个级别上概述了技术对系统结构的影响。第一章指出了VLSI(超大规模集成电路)技术对计算机系统设计师的重要性,因为它不仅影响到成本,而且影响到计算

机系统的可靠性、性能,以及日益增进的功能要求。这一章从硬件的角度述及这些论题。

第二、第三和第四章论述高级语言和机器系统结构间的关系。高级语言是程序员用来描述和说明数据对象和算法的一个有力的表述媒体。高级语言的使用成了提高软件质量和可维护性,以及提高程序员生产率的一个不可缺少的组成部分。第二章讨论高级语言的主要成分:数据对象、类型、运算符、流程控制、异常、环境,以及把数据对象归组的主要程序单元:块、子程序、模块。这些概念经历了四个发展阶段。第二章在各种高级语言所引入和支持的概念方面说明了它们的关系。第三章述及编译过程和实现模型。这章一开始讨论把一个用高级语言所写的程序翻译为以一个机器指令系统表示的语义等价表述。基于一个描述高级语言概念的实现模型,编译程序生成在机器上运行的代码。书中仅涉及了编译在影响高级语言系统结构支持的那些方面。第四章讨论了一个实现模型如何映射为一个机器系统结构。这种映射是由指令系统和执行指令系统的硬件来支撑的。指令系统的设计是在设计机器系统结构以支持高级语言中的一个关键因素。它影响到硬件、固件及编译程序设计的复杂性和效率。

第五、第六和第七章论述操作系统和机器系统结构间的关系。如高级语言一样,操作系统是现代程序设计方法学的一个基本部分。它们管理硬件和软件资源,并在机器硬件层上面的几个抽象层次上产生一个用户友好的计算机系统环境。第五章确定操作系统的基本功能:管理系统资源。它对四种资源管理:存贮器、处理机、设备及信息,各给予了简要的描述。第五章还追溯了操作系统自50年代至今的历史发展。虽然各操作系统因其应用和硬件配置而异,但它们也有许多共同的功能。操作

系统通过对各软件模块相互间的保护、同步和提供通信，来对它们进行控制。第六章相当详细地讨论了保护、同步和通信的基本功能。其重点在于对操作系统的这些基本活动的系统结构支持。第七章考虑在同一机器系列先期产品关系上操作系统的兼容性和可移植性问题，以及讨论对各种操作系统功能的硬件支持。第七章还讨论了直接面向系统结构的操作系统的例子。

可靠性要求很大程度上因不同的应用而异。但达到高可靠性的主要因素在于故障检测、故障恢复、故障诊断，以及修理。第八章讨论这些因素，以及实现它们的各种技术。此外，软件故障和软件技术作为一个综合的系统方法包括于其中。

最后，第九章描述 AT & T 3B20D 处理机。这章涉及了在前面各章中讨论过的系统结构问题的许多方面。3B20D 处理机是高效应用中在系统结构内部具备容错特性系统的一个实例分析。

计算机系统结构是一个广阔的领域，系统结构设计师所作的决定是繁杂的。本书并非要为所提及的问题列出“正确的”结论，也做不到这一点，因为各种应用以及实现的可能实在太多，并且不断增长着。相反，本书提供了一个途径，在始终考虑到整个系统的情况下，怎样来看待影响到计算机系统结构的各个课题。我们确信，这种对计算机系统结构的综合方法必须在设计的一开始就实行，以保证在最少可能代价下开发出最佳的系统来。当你们面对开发计算机系统结构的挑战时，我们祝你们各位取得如我们一样的称心如意，并且去战胜这些挑战。

目 录

第一章 计算机结构概述

1.1 引言	(1)
1.2 数字逻辑	(3)
1.3 计算机的组成部分	(10)
1.4 控制结构	(11)
1.5 算术和逻辑结构	(20)
1.6 存储器组织	(36)
1.7 输入/输出结构	(40)
1.8 软件的一般分类	(43)
1.9 技术进步对系统结构的影响	(45)
1.10 小结	(59)
习题	(59)

第二章 高级语言的概念

2.1 引言	(63)
2.2 程序设计语言机制	(65)
2.3 数据对象	(66)
2.4 类型	(68)
2.5 运算符	(87)
2.6 流程控制	(92)
2.7 环境	(99)
2.8 异常	(123)
2.9 高级语言的发展	(127)
2.10 小结	(137)
习题	(139)

第三章 编译过程和实现模型

3.1 引言	(141)
--------------	---------

3.2	编译	(141)
3.3	实现模型	(157)
3.4	数据对象	(158)
3.5	类型	(159)
3.6	运算符	(182)
3.7	流程控制	(185)
3.8	环境	(189)
3.9	异常	(204)
3.10	存储分配	(206)
3.11	小结	(212)
	习题	(213)

第四章 指令系统设计

4.1	引言	(215)
4.2	从计算机系统结构设计者观点来设计指令系统	(216)
4.3	从编译程序编写者的观点看指令系统的设计	(219)
4.4	简化指令系统和复杂指令系统	(221)
4.5	指令系统举例	(225)
4.6	寻址方式	(236)
4.7	指令的格式和编码	(244)
4.8	指令系统设计方法学	(254)
4.9	指令混合比	(256)
4.10	高效率指令系统的探索	(267)
4.11	小结	(269)
	习题	(270)

第五章 操作系统和系统结构

5.1	引言	(273)
5.2	操作系统	(274)
5.3	操作系统功能	(276)
5.4	操作系统的发展	(280)

5.5	面向操作系统的系统结构	(282)
5.6	按应用分类操作系统	(283)
5.7	按配置分类操作系统	(288)
5.8	操作系统的共性	(290)
5.9	小结	(290)
	习题	(291)

第六章 操作系统: 概念和系统结构支持

6.1	引言	(293)
6.2	进程	(294)
6.3	信息包	(302)
6.4	功能抽象与数据抽象	(305)
6.5	保护结构	(305)
6.6	保护结构的系统结构支持	(319)
6.7	功能抽象和数据抽象的趋势	(333)
6.8	同步原语	(335)
6.9	同步概念的系统结构支持	(339)
6.10	通讯原语	(349)
6.11	通讯概念的系统结构支持	(351)
6.12	硬件资源的管理	(358)
6.13	处理机管理与系统结构支持	(358)
6.14	存储器管理与系统结构支持	(363)
6.15	小结	(375)
	习题	(377)

第七章 操作系统: 系统结构设计

7.1	引言	(379)
7.2	兼容性和可移植性	(379)
7.3	效率	(384)
7.4	性能与灵活性的权衡	(400)
7.5	对操作系统支持的探索	(402)

7.6 具有操作系统支持的系统结构	(404)
7.7 小结	(413)
习题	(414)

第八章 容错计算

8.1 引言	(416)
8.2 可靠性的费用	(417)
8.3 差错的分类	(418)
8.4 系统利用的效应	(425)
8.5 冗余技术	(427)
8.6 硬件故障检测技术	(434)
8.7 软件检测技术	(447)
8.8 故障恢复	(456)
8.9 故障诊断	(467)
8.10 高可靠性实时系统中的维护功能	(474)
8.11 技术上的影响	(479)
8.12 小结	(481)
习题	(483)

第一章 计算机结构概述

1.1 引言

计算机系统结构领域在过去几乎总是作为专门对计算机硬件结构的研究,但近几年来已扩展到涉及软件问题了。今日的计算机系统设计师在为用户设计最有用,且具有高性能价格比的系统时,对系统结构的软件部分的考虑,必须与硬件部分一样重要。在快速增长的超大规模集成电路(VLSI)的制造及其它技术领域,软件问题正日益起着重要作用。

总的系统结构(组合硬件和软件两个部分)可以看作是一组解释器或抽象的层次。(图 1.1)这就是,从功能来看,在系统内包含有若干个不同类型的结构。在最高层(第一层),该系统结构决定着本系统包括哪一些数据处理功能以及用户端的外部世界包括有什么。在系统与外部世界之间的通信是通过两个界面:语言和系统应用程序,例如实用程序、分类程序以及信息检索程序。系统结构在第一层上定义了这两组界面。

第二层表示由高级语言和操作系统来实现的系统软件结构内的各种功能。高级语言在减少软件复杂性研制计划以及在提高程序员编程效率和软件的可靠性方面,起着重要的作用。这些优点是以增加存贮器空间及程序执行时间为代价来获得的。大多数的通用操作系统都具有若干套程序,这些程序能使用户有效地分享计算机硬件设备:它们也提供对各种资源的自动控制。其服务包括存贮器管理、进程管理、中断处理、文件系统管

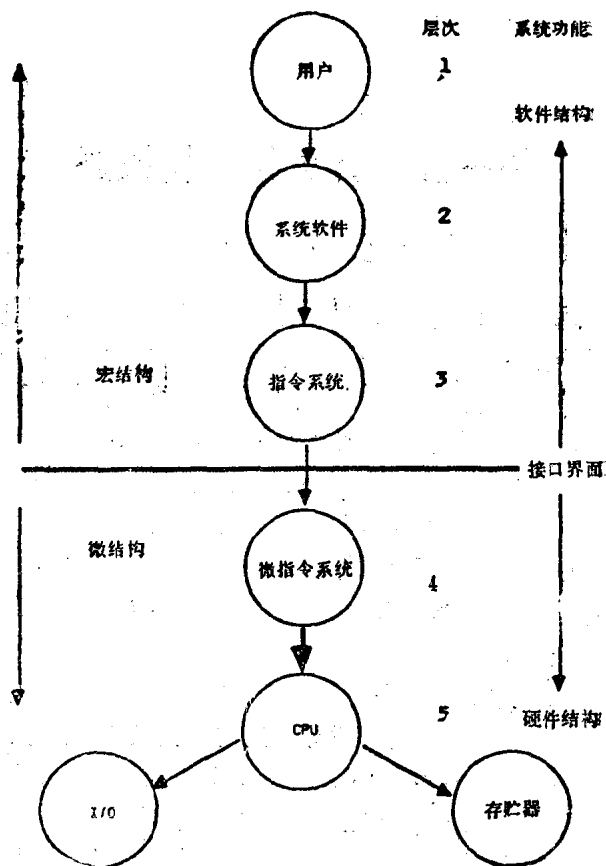


图 1.1 计算机系统结构层次

理以及系统维护。这样，操作系统就使程序员从重复地指明机器的细节，如物理特性以及用于输入/输出操作的外设协议中解脱出来。

下一个层次即第三层次是指令系统。它是程序员能访问的

最低层。指令系统是能被软件设计者所能看到的一组裸机的硬件性能和特点。这些性能包括数据运算符、寻址方式、陷阱和中断序列,以及寄存器组织。

一,二,三层次的结构形成了计算机系统结构的软件部分,或称宏结构。在指令系统级下面的功能层次形成了系统结构的硬件部分,或称微结构。在硬件/软件两部分的分配中,对微代码程序的安排已给予相当的注意。因为微程序是放置在只读存储器(ROM)中,它们称为固件(也就是说,它们是一种不受暂时性变化影响的、“牢固保护”的软件模块)。在微程序控制的机器中,固件是独立的,并考虑为硬件结构的第四层次。

虽然由于技术进步,实现硬件结构的器件已得到令人注目地进展,但今日系统结构的第五层次构成,仍有着四十年前相同的成份:中央处理单元,存储单元(存储器)、输入输出单元,以及它们之间的控制和相互连接。数字逻辑,现在尽管由先进的芯片技术,例如金属氧化物半导体和发射极耦合逻辑器件来实现,但仍然是硬件结构的基本构造模块。

本章提供一个基本的计算机结构概述。所介绍的数字电路部分是为了提供在本书中所使用的逻辑设计中的背景知识。为了补充这些背景信息,第一章复习计算机的主要物理元器件,并讨论 VLSI 技术及其对计算机系统结构的影响。

1.2 数字逻辑

1.2.1 基本逻辑门

实现数字控制逻辑需要三种基本运算符:或(OR)、与(AND)和非(NOT)。组合起来使用,这三种功能就能实现任何逻辑操作。它们用简单的门电路就能实现。在大多数的逻辑电路中,例如象二极管那样的输入元件就能实现这种逻辑操作,再

跟随晶体管输出级，就能提供适当的信号电平及足够的驱动能力(扇出)。

大多数的数字电路其输入、输出端都是处在 0 或 1 两个状态中的一个状态。有些逻辑门具有第三个输出状态，即高阻状态。下面仅对两个状态的逻辑电路进行讨论。通常惯用两种约定来表达这些逻辑值。第一种约定采用两个电平中较高的电平来表示 1 状态，称为正逻辑。第二种约定采用较低的电平来表示 1 状态，称为负逻辑。正逻辑约定更为普遍，本书就采用正逻辑表示。对于 T、T_L、L 逻辑门系列来说，其 0 状态接近为零电平，而 1 状态大致为 3 伏左右。

“或”(OR)门

“或”电路其逻辑关系是，当它的输入端中任一个处在 1 状态时，则其输出端就为 1 状态。“或”门的真值表和逻辑符号给出在图 1.2(a)。

“与”(AND)门

“与”电路要求它的所有输入端都是 1 状态时，其输出才为 1，其真值表见图 1.2(b)。“与”门的逻辑符号也给出在图中。

“非”(NOT)门

这种电路是一个反相器，它的输入输出信号是反相的。“非”电路的真值表及逻辑符号见图 1.2(c)。“非”函数是要求实现更为复杂的逻辑操作如“或非”(NOR)、“与非”(NAND)和“异或”(XOR)这些操作中的一种操作。

“或非”(NOR)门

组合“或”(OR)和“非”(NOT)两种操作的门便称为“或非”(NOR)门。其真值表及逻辑符号见图 1.3(a)。NOR 门的符号除了在其输出端加上一个小圆圈表示信号相反之外，其余都与 OR 门的符号相同。