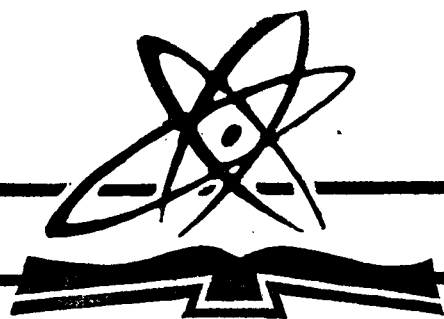


数字系统计算机辅助设计

北京工业学院

刘明业 编

国防工业出版社



数字系统计算机辅助设计

北京工业学院

刘明业 编



国防工业出版社

数字系统计算机辅助设计

北京工业学院

国防工业出版社出版

北京市书刊出版业营业许可证出字第 074 号

解放军第七二二六工厂印刷 内部发行

*

787×1092¹/₁₆ 印张 17¹/₈ 399 千字

1980 年 12 月第一版 1980 年 12 月第一次印刷 印数 1-6,000 册

统一书号: N15034(四教 53) 定价 1.80 元

内 容 简 介

本书内容是讨论计算机设计领域中的计算机辅助设计。重点是逻辑设计阶段的内容，即硬件描述语言、逻辑综合及逻辑模拟。

全书共分五章。头一章介绍计算机设计各个阶段中运用计算机辅助设计的内容和辅助设计系统的组成。第二章讨论自动逻辑综合的理论和算法。第三章为寄存器传输级硬件描述语言DDL 及其应用。第四章讨论门级和功能级模拟模型的建立，模拟语言和模拟算法。最后一章则着重讨论印制板自动布线，而划分及布局只作简单介绍。

本书为计算机专业统编的试用教材。但也可供从事计算机设计自动化工作的同志参考。

前 言

本书系高等学校工科电子类计算机专业统编的试用教材之一。其内容是计算机设计领域中的计算机辅助设计,即借助计算机的帮助来设计计算机。

“数字系统计算机辅助设计”或“计算机设计自动化”是一门新兴的学科。它所涉及的范围可概括如下:

- (1) 计算机系统模拟和性能评估、硬件描述语言、自动逻辑综合、微程序自动设计以及逻辑模拟,等等。
- (2) 工程化设计——逻辑划分、布局和布线等。
- (3) 数字系统的故障诊断。
- (4) 电子电路分析和集成电路的辅助制版。

由于这是一门仅有五十学时的选修课教材,编写中本着以计算机设计的基本理论为重点,试图反映新理论、新技术,针对教学上的需要与可能,选取自动逻辑综合(第二章)、硬件描述语言(第三章)及逻辑模拟(第四章)三个部分作为本书的主要内容。而工程化设计(第五章)则着重讨论印制板自动布线,逻辑划分及布局只作简单介绍。此外,如系统级模拟和其它计算机设计的广阔领域中的许多内容,在开头的一章里作概括介绍。至于其它方面,如故障诊断等,已另有课程,本书不再重复。

关于本课程的教学安排,目前还缺乏实践,也无可借鉴的经验,很难提供有益的意见。从课程次序上来看,安排在授课的最后一个学期较为合适。另外,本教材在内容上考虑了留有适当选择余地。除头一章的概论为各章所共有而外,其它各章的内容基本上是互相独立的。因而,讲授时内容的取舍是方便的。中间的三章都附有习题。逻辑综合在附录中给出一个算法的程序。书末提供所用术语英汉对照表,供读者参考。

本书的审阅是在复旦大学徐研人同志(主审)和张根度同志主持下进行的。初审分工如下:第一章 鲍振东;第二章 张根度、吴时霖、高传善;第三章 徐研人、袁雨飞;第四章 彭澄康;第五章 陈增荣、陈行瑶。最后由徐研人和彭澄康两位同志全面复审。审阅中同志们提出许多宝贵的修改意见,推荐补充资料,使本书的内容得以充实和较大改进。

本书编写过程中承蒙中国科学院计算技术研究所林宗楷同志多方面帮助,提供许多珍贵的资料,并对初稿提出改进意见。

对上述各位同志,编者谨此一并深谢。

尽管同志们多方关怀和热心帮助,但由于本人水平和见识所限,特别是某些新内容还是一边学习一边编写的。因此,对书中某些问题理解欠妥,甚至造成谬误是在所难免的。恳请读者同志批评指正。

编 者

一九八〇年六月

目 录

前 言

第一章 计算机设计自动化概论

§1.1 计算机设计自动化的含义和发展	(1)
1. 计算机设计自动化的产生	(1)
2. 设计自动化各主要领域的形成	(2)
§1.2 总体设计阶段的系统级模拟	(3)
1. 系统模拟	(3)
2. 系统的目标函数	(4)
3. 系统模拟的步骤和措施	(4)
4. 模拟结果分析	(8)
§1.3 逻辑设计阶段的设计自动化	(9)
1. 逻辑设计语言(硬件描述语言)	(9)
2. 逻辑综合和逻辑分析	(11)
3. 逻辑模拟	(12)
4. 逻辑图的自动绘制	(12)
§1.4 工程实现阶段的设计自动化	(13)
§1.5 计算机辅助设计系统	(13)
1. CAD系统中的硬件	(13)
2. CAD系统中的程序设计语言	(14)
3. CAD系统中的软件	(14)
4. 交互式设计系统	(16)
§1.6 计算机设计自动化实现的效果	(17)

第二章 自动逻辑综合

§2.1 自动逻辑综合的内容及意义	(19)
§2.2 用多维体表示逻辑函数	(20)
1. 问题的提出	(20)
2. 多维体的几何表示	(22)
3. 函数的初始覆盖	(23)
§2.3 多维体间的蕴涵($\subseteq$)、相并($\cup$)和相交($\cap$)运算	(23)
1. 蕴涵运算($\subseteq$)	(23)
2. 并集($\cup$)运算	(24)
3. 相交($\cap$)运算	(25)
§2.4 多维体的相容($\star$)运算	(26)
§2.5 多维体的取补($\#$)运算	(27)

§ 2.6 函数的质蕴涵项及最小代价覆盖	(31)
1. 函数的质蕴涵项	(31)
2. 函数的最小代价覆盖	(31)
§ 2.7 质蕴涵项的计算	(31)
1. 取补运算求质蕴涵项	(32)
2. 相容运算求质蕴涵项	(33)
3. 划分表法计算质蕴涵项	(34)
§ 2.8 求解覆盖问题	(37)
1. 选择极值法	(37)
2. 覆盖最多点法	(40)
§ 2.9 寻求无冗余覆盖的消去法	(42)
1. 消去多余变量的运算	(42)
2. 消去多余的多维体的运算	(43)
3. 寻求无冗余覆盖的消去法	(43)
§ 2.10 寻求无冗余覆盖的弛张法	(45)
§ 2.11 函数最小化的改进 Quine-McCluskey 方法	(47)
1. 基本概念和定义	(47)
2. 基本定理	(48)
3. 函数最小化算法	(51)
§ 2.12 多输出函数的综合问题	(53)
§ 2.13 综合多输出函数的弛张法	(55)
§ 2.14 多级网络及因子分解	(59)
§ 2.15 提取多维体集合的因子	(60)
1. 多维体的因子	(60)
2. 多维体集合的因子及其质量因数	(61)
3. 多维体集合因子的计算	(61)
§ 2.16 多维体列阵提取因子的增长因子法	(62)
§ 2.17 扇入有限的单输出多级与非网络的综合	(66)
1. 用两个与非门解决扇入问题	(66)
2. 实现多级网络的三种方案	(67)
3. 综合算法	(71)
4. 综合算法的改进	(75)
§ 2.18 扇入有限的多输出与非网络的综合	(75)
§ 2.19 扇出有限的网络的综合	(78)
1. 直接增加门扇出	(78)
2. 提取因子解决扇出	(79)
§ 2.20 扇入、扇出有限的多输出与非网络的综合	(81)
第二章习题	(84)

第三章 寄存器传输语言及其翻译

§3.1 引言	(88)
§3.2 寄存器传输语言的组成	(89)
1. 构成数字系统的基本单元	(89)
2. 说明语句	(89)
3. 操作语句	(89)
4. 条件语句	(90)
5. 状态控制及状态语句	(91)
§3.3 寄存器传输语言 DDL 的规定	(91)
1. 标识符	(91)
2. 语句	(92)
3. 部件引用的规定	(93)
§3.4 简单部件说明	(94)
1. 引线说明	(94)
2. 寄存器、存储器和暂存器说明	(95)
3. 定时和延迟说明	(95)
4. 布尔说明	(96)
5. 单元说明	(98)
§3.5 操作语句	(99)
1. 取头、截尾和扩充操作	(99)
2. 结合	(100)
3. 变反和选择操作	(100)
4. 简化	(100)
5. 关系和算术操作	(101)
6. 传统的逻辑算符	(102)
7. 布尔表达式	(102)
§3.6 条件操作语句	(103)
1. 相容操作集合的条件操作	(103)
2. 按值子句	(105)
§3.7 连接和传输操作语句	(106)
§3.8 移位和计数操作	(109)
§3.9 标识符说明和相容操作集合	(112)
§3.10 复杂部件的算符说明	(113)
§3.11 复杂部件描述的状态语句和状态说明	(116)
§3.12 同步时序线路的设计	(117)
§3.13 自动机说明和系统说明	(121)
§3.14 寄存器传输语言的翻译及其主要阶段	(126)
1. 输入阶段的处理	(127)

2. 数据流分析阶段	(128)
3. 控制流分析阶段	(129)
4. 方程产生阶段	(130)
§ 3.15 DDL 用于微程序设计举例	(130)
1. 机器的工作及组成	(130)
2. 机器的微程序设计	(132)
3. 操作的执行和机器结构	(135)
第三章习题	(137)
第四章 逻辑模拟	
§ 4.1 逻辑模拟的目的及意义	(141)
§ 4.2 逻辑模拟模型的建立及分级	(142)
1. 寄存器级模拟	(142)
2. 门级逻辑模拟	(143)
3. 功能块级模拟	(143)
§ 4.3 模拟元件的延迟时间	(143)
1. 零延迟时间	(144)
2. 指定常数的延迟时间	(144)
3. 指定最小延迟时间和模糊区域	(144)
4. 分别指定正跳变和负跳变的延迟时间	(144)
§ 4.4 模拟信号的状态值	(145)
§ 4.5 模拟时钟及定时算法	(146)
1. 模拟时钟	(146)
2. 定时算法	(146)
§ 4.6 门级逻辑线路描述语言	(148)
1. 门级线路的描述	(148)
2. 功能块级线路的描述	(150)
3. 内部故障的描述	(153)
4. 含有功能元件的线路描述方法	(153)
§ 4.7 门级逻辑模拟命令语言	(154)
§ 4.8 门级逻辑模拟算法的分类	(154)
§ 4.9 逻辑模拟的编译法	(156)
§ 4.10 逻辑模拟的表驱动法	(158)
1. 下一事件的同步模拟及其实现	(158)
2. 时间映射异步模拟原理	(159)
3. 表驱动法的基本表格结构	(161)
4. 表驱动法的时间流程	(164)
5. 表驱动法的主要优点	(166)
§ 4.11 多值模拟理论	(167)

1. 静态冒险分析数据的产生	(167)
2. 无延迟元件时序冒险分析	(168)
3. 有延迟的三值模拟	(170)
4. 三值模拟信号值的计算	(171)
5. 初值建立及四值模拟	(173)
6. 五值逻辑模拟的概念及其应用	(174)
7. 八值逻辑模拟的概念	(176)
§ 4.12 故障模拟	(177)
1. 故障模拟模型	(177)
2. 平行模拟方法及故障的传播	(177)
3. 故障传播的演绎法	(178)
§ 4.13 举例: 门级逻辑模拟程序 MLM111	(180)
1. 功能	(180)
2. 模拟语言	(180)
3. 源程序	(186)
4. 汇编	(187)
5. 模拟算法	(187)
6. 故障模拟	(189)
7. 功能块的处理	(189)
8. 模拟程序的使用及举例	(190)
第四章习题	(197)

第五章 组装设计

§ 5.1 引言	(199)
§ 5.2 逻辑图的图论表示法	(199)
§ 5.3 逻辑划分	(200)
§ 5.4 逻辑选择	(204)
§ 5.5 布局	(205)
1. 布局的目标函数表达式	(206)
2. 改善布局的交换方法	(206)
3. 分线	(207)
4. 主元素交换迭代法	(207)
5. 构造性算法	(208)
§ 5.6 布线	(210)
1. 布图	(210)
2. 插件逻辑图的初始连接信息	(211)
3. 分线产生的连线表	(212)
4. 分层定顺序	(212)
5. 布障碍墙	(214)

6. 走 线	(214)
7. 借孔处理	(217)
8. 走线控制程序	(217)
9. 布线结果输出	(218)
附录	(219)
附录一 改进的 Quine-McCluskey 方法 (算法 2.11-1) 的 FORTRAN 程序	(219)
附录二 逻辑元件使用附号说明	(257)
附录三 本书使用的 CAD 术语英汉对照表	(228)
主要参考书籍及文章	(262)

第一章 计算机设计自动化概论

§ 1.1 计算机设计自动化的含义和发展

由于电子计算机具有计算精度高、操作速度快、记忆容量大、逻辑分析能力强及工作高度自动化等突出的优点,所以它不但适于数值计算,而且也具有较强的数据处理及对象模拟的能力。因而目前诸如飞机、船舶、光学仪器及半导体器件等设计越来越多地使用计算机,并统称为“计算机辅助设计”。

而所谓“数字系统计算机辅助设计”或称“计算机设计自动化”,就是使用已有的计算机帮助设计和研制新的计算机。使计算机的设计和研制工作在计算机的帮助之下,达到自动化或半自动化,从而减轻人的劳动并提高设计质量。因为设计工作是按设计者预先的安排,编出程序由计算机去自动完成的,因此称为“设计自动化”。但由于有些工作还是要发挥人的创造力,在人的密切参与下进行,计算机只限于帮助设计者完成一部分工作,仅起到“助手”的作用,故称为“计算机辅助设计”可能更适合些。上述两种说法之间的界限比较含混,常常交错使用。仔细区分它们,则设计自动化主要还是个目标,并非真的都是直接可以实现的。较现实的是把计算机应用于设计领域,将人解决问题的聪明才智和计算机本身巨大的计算能力及数据处理能力结合起来,更有效地完成设计工作。

1. 计算机设计自动化的产生

当计算机还处于第一代的时候,就显示出其非凡的能力。因此随着计算机的设计工作日趋复杂,人们早在五十年代就已经注意到使用计算机来帮助设计新计算机的问题,并且通过设计工作的实践,逐渐形成设计自动化这个新学科。

设计自动化是从第二代计算机的设计开始的,即使用第一代的计算机来设计第二代计算机。美国在五十年代曾有一项 STRETCH 工程,其目的是研究大型第二代的晶体管计算机系统。此项工程对计算机的设计自动化起了很大的推动作用。当时计算机的设计过程已逐渐划分为逻辑设计阶段及工程实现阶段。

在逻辑设计阶段,一台计算机当时已包含几万个逻辑电路,予感到计算机的结构是越来越复杂。设计资料、图纸越来越多。对设计文件的检查、修改、管理遇到很大的困难。于是计算机的设计者们便考虑使用计算机建立逻辑设计文件。并在实践中发现,如果将逻辑图保存在磁带文件里,每次进行工程更改时,只需要通过若干张书写更改数据的卡片,就可以在短时间内获得修改后的新结果。

在工程实现阶段最早使用计算机的是底板绕线。当时,计算机工业里已出现了一些自动辅助制造方面的机器。如美国在五十年代末,由 Gardner Denver 工具公司生产的绕线机曾被广泛使用。然而,用手工准备绕线机所需要的控制数据很麻烦,且容易出错。于是人们开始用计算机来产生控制数据。同时,随着晶体管器件的出现,新元件的体积大大缩小。元件

越小, 电路速度越增加。可是, 与此同时连线也更接近, 线间感应的干扰也随之增加。所以, 走线时必须考虑到平行走线不能过长的问题。这便迫使人们使用计算机选择两条平行信号线最短的路径, 并要保证线路的最好分布。从而, 底板的绕线速度也大大提高。

设计自动化的进一步发展, 人们自然想到应将逻辑设计文件和底板的布线以及其它方面连结成一个统一的系统。即考虑建立设计自动化系统的问题。早期的设计自动化系统, 都包含一些检查程序。将从前用手工进行的逻辑检查工作交给计算机去做。例如通过计算机程序可以检查电路的逻辑连接是否正确。当发现错误时就打印出相应的信息。并且由于硬件工艺的进展, 搞逻辑设计的人员已不再从事电路设计, 而是只选取现成的电路器件。搞电路设计的在提供元件时, 还提出使用这些电路的一套规则。于是计算机还用于完成这些规则的检查。例如规定某电路只有两个输入, 如果使用这个电路时有三个输入, 规则检查程序就会给出错误信息。

2. 设计自动化各主要领域的形成

现代电子计算机的设计过程常因机器规模和所采用的工艺的不同而有所差异, 但总的可分为如下三个阶段:

(1) 总体设计阶段

在开始设计一台新机器时, 必须针对其用途决定机器的规模、速度、功能以及指令系统等。然后再决定控制器、运算器、存储器等功能部件的结构框图、各部件的组成和结构, 以及它们之间信息交换的路径等等。这就是总体设计阶段的任务。

(2) 逻辑设计阶段

逻辑设计阶段的任务是要进一步设计出各个部件(如控制器、运算器、存储器等)的详细逻辑线路图。逻辑线路是用一些基本单元(如与非门、或非门及由它们构成的触发器等)来实现的。由基本单元构成的逻辑线路图便是下一个阶段(工程实现阶段)的初始信息。

(3) 工程实现阶段

工程实现也称为工程化, 或硬件的组装。现代电子计算机通常是由分成好几级的很多结构单元组装而成的。由于所使用的器件及工艺的不同, 分成多少级也是有差异的。例如第三代计算机, 通常是将组件当作最小结构单元, 而它又是一些基本逻辑单元(即门电路)构成。再由若干组件构成插件, 插件布满底板, 底板组成机柜。组装设计的具体任务有划分、布局 and 布线。此外还需要编制和产生供组装及制造用的各种文件、资料、表格以及供调试、诊断用的有关资料。

藉助计算机完成总体设计阶段的任务通常采用模拟的方法, 称为系统级模拟。而以计算机辅助来完成逻辑设计阶段的任务的过程, 则称为逻辑设计自动化, 其中所涉及的内容有硬件描述语言, 自动逻辑综合, 逻辑模拟以及自动绘制逻辑图等等。这是本教材要做较深入讨论的课题。目前在计算机设计自动化的各领域中较为成熟并已广泛地获得应用的是计算机辅助组装设计, 即第三个阶段的设计任务。其中包括自动逻辑划分、布局、多层印制板的自动布线, 以及各种设计资料、文件、表格的自动产生和修改, 等等。

随着计算机的迅速发展, 计算机的各基本单元(例如插件板)生产中故障测试问题也显得更为重要。人工测试逐渐感到无能为力。如果利用计算机来自动测试故障, 将会大大缩短

机器的研制周期。由于这方面的需要,“故障测试码的产生”这个重要课题的研究近几年来也取得许多成果。

集成电路从六十年代开始出现以来,集成度在不断地增加,由只包括简单的几个门的小规模集成电路,发展为中规模集成电路和大规模集成电路。随着这些硬件工艺的发展,计算机已由第三代进入到第四代。对于研制第四代计算机而言,设计自动化似乎是不可缺少的手段。

集成电路的设计和生,很快就用上了计算机辅助设计。例如,集成度很高的多层电路的生产和印制,已使用了计算机控制的掩模产生设备,掩模已由设计自动化系统提供的数据产生。这些数据的电性能和逻辑上的关系(或称相容性),必须彻底进行检查。因为电路一经制成,修改实际上就不再可能了。在投入制造之前,必须通过计算和验证,保证设计无误。因此,对于集成电路的制造,掩模设计程序、模拟程序、测试和诊断程序已经成为必不可少的。

综括上述,随着机器规模越来越大,结构越来越复杂,性能越来越高。于是,对设计、制造都提出了更加严格的要求。并且设计的工作量显著增加。因而对使用已有计算机来设计新计算机的要求也更加迫切。一台机器的设计,从方案论证、逻辑设计、组装设计到制造及调试,是一个复杂而繁琐的过程。设计过程中有一点疏忽都会引起一系列互相关联的修改,甚至返工。而有些问题全靠人的精力也难于发现。以致使机器设计、调试周期拖长,甚至中途发现隐患造成较大的返工。因此,目前从总体方案的论证、逻辑设计、电路分析、组装设计,直到器件筛选、插件测试、底板测试、机箱检查,乃至全机的诊断等,计算机研制、生产和维修的各个阶段的大量工作,人们都在尝试发挥计算机的专长。并且许多方面取得的喜人成果已广泛地应用到计算机的生产实践当中。

§ 1.2 总体设计阶段的系统级模拟

在总体设计阶段,必须根据所设计机器的用途确定体系结构。其内容包括:拟定指令系统,选择机器结构参数和性能,如存储容量、运算字长、操作速度等;确定功能部件的设置,合理地配备外部设备,如磁鼓、磁带、磁盘及输入机、打印机等,以及如何安排中断设备、分时系统及多道程序等。这些工作主要还是要积极发挥人的创造能力,由为数不多的专门技术人员来完成。但是为了对这些方案性的抉择有一个较全面的分析作为依据,可利用现有的计算机对所要设计的机器功能进行模拟并估价其性能。此即系统级模拟。

系统模拟首先用于系统设计,预测系统的性能,进而改进设计方案,使之趋于更合理、更有效。所以,系统模拟为更科学地进行系统设计提供依据。

1. 系统模拟

系统模拟模型是建立一个同实际系统相类似的工作状态模型。一个系统设计方案的选择通常要经过大量的模拟,由于模拟模型能够较灵活地描述出实际的系统,人们就能从为数众多的模拟结果中得出某些有用的结论。一般来说,通过模拟只能选择较好的方案,而不一定能选择到最佳方案。由于现代计算机具有较强的对象模拟能力,所以它就成为实现计算机系

统模拟的一个有力工具。目前，系统模拟方法已广泛地用在计算机系统设计。

计算机系统模拟可分为粗的模拟（或称宏观模拟）和细的模拟（或称微观模拟）。这种划分是相对的，它们之间没有明确的界限。通常可以从两个方面的标准来区分。首先是从模型简化程度来区分：例如，在粗的模拟模型中 CPU 为单元块，磁盘系统为单元块，存储器系统为单元块等等；而在细的模拟模型中，以寄存器、各种栈、运算器、超高速缓冲存储器等为单元块。其次，也可以从模拟模型中所使用的数据来区分：粗的模拟模型所使用的数据，基本上都是用统计方法产生；而细的模拟模型使用的数据，一般采用有关真实计算任务的程序，或者加上些统计数据。在尚未掌握适当的统计数据的情况下，可选择各种类型的作业程序作为数据，对所设计的机器进行系统模拟。

2. 系统的目标函数

计算机系统的目标函数不是单一的，而是由许多目标函数构成的集合。例如，可将系统的速度 V 、成本 S 、可靠性 U 、技术上的可行性 K 及可维护性 W 等与系统各部件的参数 $a_1, a_2, a_3, \dots$ 之间的关系表示成：

$$V = f_1(a_1, a_2, a_3, \dots),$$

$$S = f_2(a_1, a_2, a_3, \dots),$$

$$U = f_3(a_1, a_2, a_3, \dots),$$

$$K = f_4(a_1, a_2, a_3, \dots),$$

$$W = f_5(a_1, a_2, a_3, \dots).$$

... ..

例如，对某一被模拟的机器而言，累加器的个数、先行取数栈的个数、后行送数栈的个数、指令堆栈的工作方式、指令堆栈的容量、存储器的体数、存储器的存取周期、指令缓冲器的存储体数目、工作方式、每条指令的运算时间、硬设备间连接方式、指令系统的功能等等都可表征为一系列参数。

设计一个计算机系统若能使上述目标函数达到最佳，则说该计算机系统在这些条件下是最佳系统。实际上很难达到系统的全部目标函数都是最佳。同时有些目标函数目前尚无法作定量的描述。如技术上的可行性、可维护性等目标函数还无法将它们描述为数量关系。而有的目标函数，例如成本、可靠性等，目前在模拟中使用仍存在着一定的困难。为简化问题的研究，现在假定仅采用一个目标函数来描述系统的“好”或“坏”。采用模拟方法模拟出该系统的真实情况。获得速度的函数关系为

$$V = f_1(a_1, a_2, a_3, \dots).$$

3. 系统模拟的步骤和措施

系统模拟有粗细之分。因此在具体作法上差异可能较大。

粗一些的示例为由一个中央处理机（CPU）、一个中央存储器（CM）和一个浮动磁头的磁盘所组成的系统（图 1.2-1）。现假定模拟分析的目的是确定采取下述措施后系统性能的改善：

(1) 增加一个附加的中央存储器，或者

(2) 用一个更快的模型来代替磁盘。

在这一课题中 CPU 可作为一个独立单元存在，而不需要再作简化。

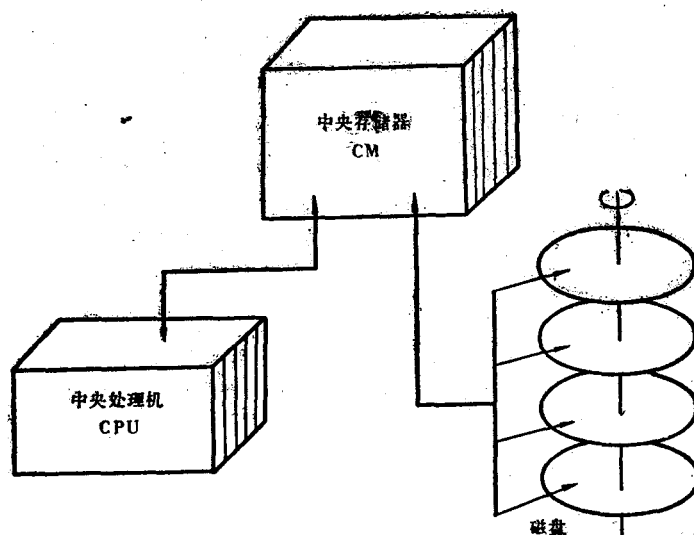


图 1.2-1 一个多道程序系统的组成

如果需要对 CPU 系统模拟，则要按下列步序对其采取措施：

(1) 简化 CPU 系统构成模型

简化系统是建立模拟模型的第一步，也是重要的一步。只有将系统简化成模型才有可能进行模拟，但简化后的系统也必须反映所模拟系统的基本情况。同时，也只有将系统作合理简化了，才有可能在已有的机器上模拟更大型机的 CPU 系统。

假定所要模拟的大型机具有一个功能较强、种类繁多的指令系统。例如有一台机器其操作码为 7 位，共有 128 条指令。地址码占 23 位。因为我们是选取主机速度当作目标函数，因此凡对主机速度影响不大的因素均可进行简化。

1) 同类操作指令进行合并

为使模拟模型简单，编制模拟程序方便，首先必须对 128 条指令进行归并。当然只能对同类型操作的指令进行合并。例如，在指令系统中有这样一些指令，小于比较指令、大于比较指令、按绝对值小于比较指令、按绝对值大于比较指令、全等比较指令等。这些指令在运算器内运算时间相似，它们的操作逻辑也类似。因此，对于这类指令可以把它们合并成一条指令，即比较指令。同样可把定点加、定点减、逻辑加、逻辑乘、按位加、阶加阶、阶减阶、阶加整数、阶减整数等指令归并成一条加法指令。经过这样合并后，便将 128 条指令压缩到 32 条指令。

2) 指令中操作数地址进行简化

指令地址码占 23 位，其中 3 位表示字节地址，4 位表示体地址，其余 16 位表示同体内单元地址。而每条指令根据特征位控制其操作是全字长、半字长、一字节、二字节等运算。但无论一字节、二字节、半字长的运算，都类似于全字长运算，即运算时间一样。因此，可以把字节地址简化掉。相应地，在系统结构图上就可以把有关字节运算的部件，如字节控制

器、字节寄存器等去掉。

模拟时涉及到存储器的问题，仅关心取、送数时数据在存储器控制部分的冲突时间，也就是关心同一存储体内取、送数冲突的时间。因为这会影响到整个计算机系统的速度。而与此相反，同一体内取那一个单元却是无关紧要的，因为这不影响系统的目标函数速度。因此，可取消 16 位体内地址。而 23 位的地址码只要保留体地址就行了。例如被模拟系统的存储器有 16 个体，则体地址只要 4 位就行了。于是又简化了一大步。这样将有关部分都作相应简化后，便获得最后简化的指令系统，与其相应的 CPU 系统简化后的框图如图 1.2-2 所示。

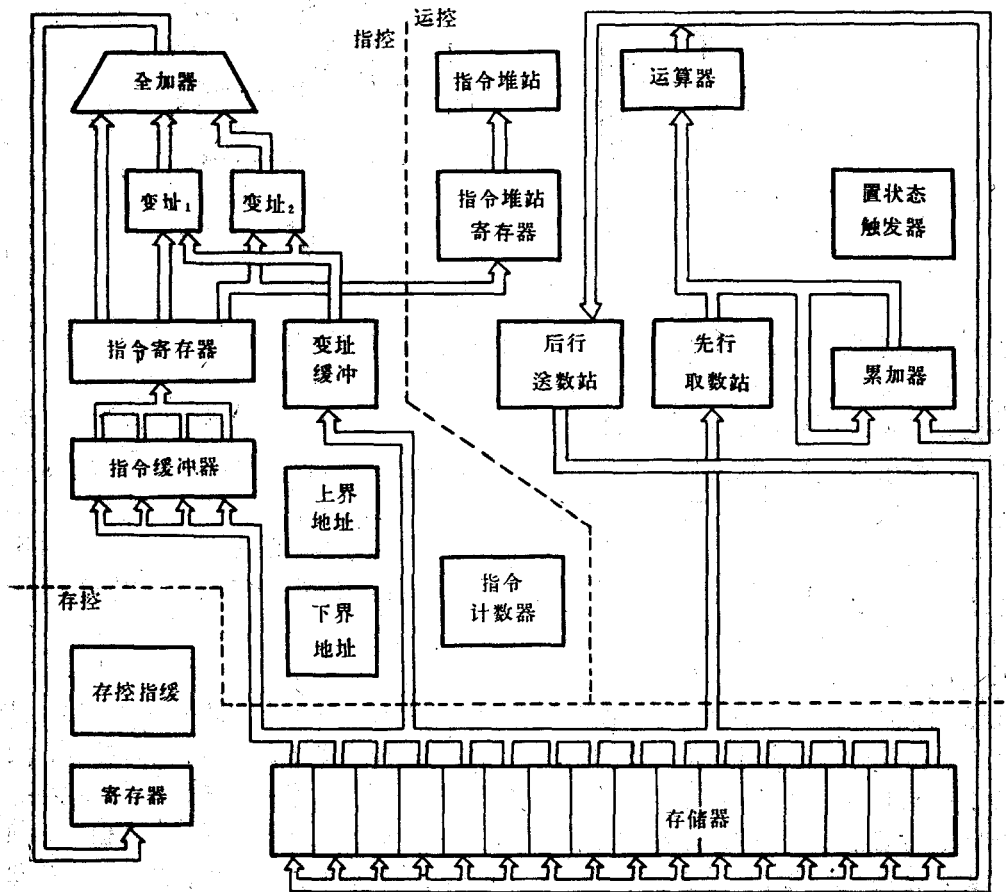


图 1.2-2 大型计算机系统简化框图

(2) 选择典型作业，并产生指令流、地址流。

假定我们所设计的计算机主要用于科学计算，则根据典型作业选择一系列典型计算方法，如选取全主元素消去法和平方根法，再按简化后的指令系统编制程序。并将这两个程序当作整个模拟系统的输入数据，称它们为被解析程序。

模拟 CPU 的工作过程首先是把被解析程序展开成指令流、地址流，并记录在磁带上，