

73.87
111

计 算 机 逻 辑 导 论

【美】小内格尔，卡罗尔，欧文 编著

李 作 新 译

李宗杰 王飞龙 校

人 民 邮 电 出 版 社

An Introduction to Computer Logic

H. TROY NAGLE, Jr.

B. D. CARROLL

J. DAVID IRWIN

Prentice-Hall, Inc.

1 9 7 5

内 容 提 要

本书系统地阐述了计算机逻辑和数字系统的基本理论及其分析与综合的方法。主要内容包括以下几部分：数制和代码；组合逻辑电路；时序逻辑电路；故障和诊断；计算机辅助设计；以及多值逻辑、阈值逻辑、线性时序机、累接网络等专题。

本书十三章各章都有大量例题、习题和参考文献，并专辟一章讨论应用实例，为读者深入研究提供方便。

本书系美国奥本大学开关理论与逻辑设计的课本。可作为我国高等院校电子工程系、计算机科学和工程系有关专业的教学参考书，也适于从事计算机和数字系统工作的科技人员自学。

计 算 机 逻 辑 导 论

[美]小内格尔 卡罗尔 欧文 编著

李 作 新 译

李宗杰 王飞龙 校

人民邮电出版社出版

北京东长安街27号

河北省邮电印刷厂印刷

新华书店北京发行所发行

各地新华书店经售

开本：850×1163 1/32 1984年5月 第 一 版

印张：19 4/32 页数：306 1984年5月河北第一次印刷

字数：504 千字 印数：1—14,000册

统一书号：15045·总2710—有5290

定价：2.95 元

目 录

第一章 引 论

1.1 计算机的历史	(1)
1.2 数字计算机	(3)
1.3 计算机代码	(4)
1.3.1 运算码	(4)
1.3.2 输入输出码	(5)
1.3.3 指令码	(6)
1.4 计算机的指令	(6)
1.5 应用程序	(6)
1.6 系统程序	(7)
1.7 计算机硬件	(8)
概要	(12)
参考文献	(12)

第二章 数制和计算机代码

2.1 数制	(13)
2.1.1 并置记数法	(14)
2.1.2 多项式表示法	(14)
2.1.3 数制表	(15)
2.1.4 带符号的数	(16)
2.1.5 浮点记数法	(17)
2.2 算术运算	(19)
2.2.1 二进制加法	(20)
2.2.2 二进制减法	(22)
2.2.3 二进制乘法	(23)

2.2.4	二进制除法	(24)
2.3	数基变换	(25)
2.3.1	变换公式	(25)
2.3.2	通用变换算法	(33)
2.3.3	基数变换的实例	(34)
2.3.4	用基数 2^k 表示的变换	(36)
2.4	补数运算	(38)
2.4.1	补码	(39)
2.4.2	补码数制	(39)
2.4.3	二进制补码	(40)
2.4.4	二进制补码计算器	(42)
2.4.5	二进制补码运算	(44)
2.4.6	降基补码	(47)
2.4.7	二进制反码	(47)
2.5	计算机代码	(49)
2.5.1	二进制编码的十进制码	(50)
2.5.2	4221码	(50)
2.5.3	葛莱码	(52)
2.5.4	余3码	(53)
2.5.5	简单奇偶码	(54)
2.5.6	五中取二码	(55)
2.5.7	二五混合进制码	(56)
2.5.8	字母数字码	(56)
	概要	(59)
	参考文献	(59)
	习题	(59)

第三章 开关函数的代数

3.1	布尔代数	(66)
3.1.1	公设的文氏图	(66)

3.1.2	对偶性	(70)
3.1.3	布尔代数的定理	(71)
3.1.4	再用文氏图	(80)
3.1.5	布尔函数	(82)
3.1.6	布尔函数的化简	(87)
3.1.7	布尔代数的函数定理	(89)
3.2	开关电路	(90)
3.2.1	符号和作用	(91)
3.2.2	开关网络的分析	(97)
3.3	开关函数的代数形式	(100)
3.3.1	积之和以及和之积形式	(100)
3.3.2	标准形式	(101)
3.3.3	展开成为标准式	(109)
3.4	开关网络的综合	(111)
3.4.1	“与—或”及“与非”网络	(111)
3.4.2	“或—与”及“或非”网络	(113)
3.4.3	二级“与非”门网络的标准形式	(114)
3.4.4	二级“或非”网络的标准形式	(114)
3.4.5	时间图	(116)
3.4.6	串—并行多路转接器	(117)
3.4.7	扇入和扇出	(119)
3.4.8	俱乐部自动入门卡片	(119)
	概要	(121)
	参考文献	(122)
	习题	(122)

第四章 开关网络的最简化

4.1	卡诺图	(130)
4.1.1	在卡诺图上表示布尔函数	(135)
4.1.2	布尔函数的化简	(141)

4.1.3	用卡诺图求和之积形式	(148)
4.1.4	任选情况	(151)
4.1.5	卡诺图的其他应用	(155)
4.1.6	布尔代数和卡诺图	(161)
4.2	开关网络的冒险	(163)
4.3	奎因—麦克卢斯基 (<i>Quine—McCluskey</i>) 方法	(166)
4.3.1	覆盖方法	(171)
4.3.2	有任选项的函数	(176)
4.3.3	多输出系统	(178)
	概要	(181)
	参考文献	(181)
	习题	(182)

第五章 开关网络的计算机辅助设计

5.1	引言	(186)
5.2	网格结构	(187)
5.2.1	n 立方	(187)
5.2.2	十进数变换	(189)
5.2.3	包含性	(191)
5.3	最简化算法	(197)
5.4	举例	(202)
	概要	(207)
	参考文献	(207)
	习题	(208)

第六章 组合逻辑专题

6.1	函数的分解	(211)
6.1.1	引言	(211)
6.1.2	简单断离分解	(211)
6.1.3	简单非断离分解	(215)
6.1.4	复杂分解	(216)

6.1.5 通过分解减少硬件	(216)
6.2 “与非”和“或非”综合	(218)
6.2.1 引言	(218)
6.2.2 “与非”综合	(218)
6.2.3 “或非”综合	(225)
6.3 多值逻辑	(226)
6.3.1 多值代数	(226)
6.3.2 函数表示法	(231)
6.4 只读存储器逻辑	(235)
6.4.1 引言	(235)
6.4.2 基本的ROM	(236)
6.4.3 应用	(237)
6.5 阈值逻辑	(240)
6.5.1 基本定义	(240)
6.5.2 基本性质	(242)
6.5.3 布尔函数的实现	(243)
概要	(247)
参考文献	(247)
习题	(248)

第七章 时序电路引论

7.1 时序电路模型	(252)
7.1.1 框图表示法	(252)
7.1.2 状态表和状态图	(253)
7.1.3 米利模型	(256)
7.1.4 穆尔模型	(257)
7.2 时序电路的分析	(258)
7.2.1 概述	(258)
7.2.2 举例	(258)
7.3 存储器件	(263)

7.3.1 延迟线	(264)
7.3.2 触发器	(265)
7.3.3 其他存贮器件	(274)
7.4 时序电路的类型	(274)
7.4.1 脉冲同步电路 (类型 1)	(276)
7.4.2 电平同步电路 (类型 2)	(281)
7.4.3 脉冲异步电路 (类型 3)	(284)
7.4.4 电平异步电路 (类型 4)	(290)
概要	(293)
参考文献	(293)
习题	(293)

第八章 同步时序电路的综合

8.1 综合方法	(304)
8.2 完全确定电路	(307)
8.2.1 等价状态	(307)
8.2.2 触发器输入表	(318)
8.2.3 设计举例	(320)
8.2.4 状态指定	(334)
8.3 不完全确定电路	(352)
8.3.1 状态简化	(354)
8.3.2 状态指定和电路实现	(368)
概要	(370)
参考文献	(371)
习题	(372)

第九章 异步时序电路

9.1 异步电路的类型	(388)
9.2 脉冲型电路的综合	(390)
9.2.1 脉冲型电路的设计方法	(390)
9.3 基本型电路的分析	(396)

9.3.1	引言	(397)
9.3.2	表格表示法	(398)
9.3.3	分析方法	(400)
9.4	基本型电路的综合	(402)
9.4.1	综合方法	(403)
9.5	竞争,循环和冒险的介绍	(413)
9.5.1	竞争与循环	(415)
9.5.2	竞争条件的避免	(421)
9.5.3	无竞争的状态指定	(423)
9.5.4	冒险	(433)
9.5.5	分析	(436)
	概要	(436)
	参考文献	(436)
	习题	(436)

第十章 时序电路的计算机辅助设计

10.1	设计算法	(451)
10.2	状态最简化	(451)
10.2.1	完全确定电路	(451)
10.2.2	不完全确定电路	(453)
10.3	状态指定	(460)
10.4	开关函数的产生	(464)
10.5	开关函数的最简化	(466)
10.6	设计实例	(466)
	概要	(468)
	参考文献	(469)
	习题	(469)

第十一章 时序电路专题

11.1	时序机的分解	(471)
11.2	线性时序机	(479)

11.2.1	定义	(479)
11.2.2	模型	(479)
11.2.3	多项式乘法和除法	(483)
11.2.4	应用	(484)
11.3	累接网络	(487)
11.3.1	引言	(487)
11.3.2	模型	(488)
11.3.3	累接网络的设计	(489)
11.3.4	实际考虑	(492)
	概要	(492)
	参考文献	(492)
	习题	(493)

第十二章 数字逻辑的应用

12.1	数字模件和子系统	(495)
12.1.1	$n-2^n$ 译码器	(495)
12.1.2	移位寄存器	(496)
12.1.3	加法器	(500)
12.1.4	累加器	(504)
12.1.5	单稳	(506)
12.1.6	计数器	(507)
12.1.7	二进制计数器	(507)
12.1.8	升/降计数器	(509)
12.1.9	模 N 计数器	(511)
12.1.10	特殊序列计数器	(516)
12.2	数字设计的问题分析	(517)
12.2.1	数字分数乘法器	(517)
12.2.2	可视电话同步脉冲检测器	(519)
12.2.3	停车场控制器	(523)
12.2.4	光束计数器	(526)

12.2.5 数字组合锁.....	(528)
12.2.6 异步数据转换器.....	(530)
12.2.7 自适应交通灯控制器.....	(533)
概要.....	(537)
参考文献.....	(537)
习题.....	(537)

第十三章 逻辑电路的故障诊断

13.1 引言.....	(541)
13.2 故障模型.....	(542)
13.3 组合逻辑网络.....	(543)
13.3.1 测试信号的产生.....	(544)
13.3.2 故障的检测.....	(559)
13.3.3 故障定位和诊断.....	(562)
13.4 时序逻辑电路.....	(564)
13.4.1 测试信号的产生.....	(565)
13.4.2 测试步骤.....	(574)
概要.....	(578)
参考文献.....	(578)
习题.....	(579)
专业词汇对照表.....	(585)

第一章 引 论

近年来数字式电子计算机对世界各地的科学家、工程师、教育家和实业家来说，已经成为重要的越来越普及的数据处理工具。可以在实验室里看到数字计算机监视与控制实验，可以看到它在工业生产线、商业会计和编目系统、科学数据处理中心里工作，或许很快在普通家庭中也可以看到数字电子计算机。在阅读这本教科书的内容即计算机逻辑之前，为使读者学起来方便，我们先介绍有关数字计算机的一些基础知识和一般术语。

1.1 计算机的历史

世界上第一个计算机大概是算盘，它在中国已经用了三千多年以上。这个今天还在使用的装置，一直到1600年约翰·奈培 (John Napier) 将对数用为乘法装置的基础的时候，才遇到了竞争。奈培的研究导致计算尺的发明。然后，在1642年布莱斯·帕斯卡尔 (Blaise Pascal) 制造了一架能做加法的机器，它有一些齿轮，很像现代的里程表。

1820年查尔斯·巴贝奇 (Charles Babbage) 制造了第一个应用现代计算机原理的装置。他的机器是用差分法求解多项式。他还设想出一部类似近代有存贮部件和运算部件的机械式机器。但是制造这种机械传动齿轮所需要的精度，超出了他那个时代工匠的能力。

1930年末，当哈佛大学的霍华德·艾肯 (Howard Aiken) 和贝尔电话实验室的乔治·斯利必兹 (George Slibitz) 研制了一台使用继电器网络的自动计算器的时候，才第一次真正地向电子数字计算机前进了一步。第二次世界大战时，为计算大炮弹道研制了其

他继电器式机器。虽然这些机器运算相当缓慢，体积也比较大，但显示了电子计算机的强大功能。本世纪四十年代初，宾夕法尼亚大学的约翰·莫切利（John Mauchly）和小普雷斯普·埃克特（J.P-resper Eckert Jr.）设计、制造了一台真空管电子计算机，并把它称为电子数字积分计算机（ENIAC）。这台机器于1945年完成，并装在马里兰州的阿伯丁（Aberdeen）试验场。ENIAC使用了18,000个电子管，它耗电巨大，故障率高，而且编制程序也困难，因为需要用插板。

后来有了三个非常重要的改革之后，才开始向当代数字计算机飞速发展。首先是约翰冯·纽曼（John Von Neumann）提出了在计算机的存贮器中存放程序，并可在存贮器中随意改变的想法，这就解决了ENIAC编制程序的困难。其次是1947年发明了晶体管，由于用它取代了电子管，所以计算机的体积和所需要功率急剧减小。第三是福雷斯特（J.W. Forrester）和他在麻省理工学院的同事研制成存贮容量很大的磁心存贮器。

ENIAC和四十年代末到五十年代初的其他真空管计算机被称作第一代数字计算机。晶体管的出现产生了第二代计算机，它体积较小，运算较快，并较第一代具有更好的性能。五十年代末到六十年代初出现了第三代计算机。这些计算机的特征是使用了把几个晶体管电路封装成为超小型器件的集成电路，这就使计算机的体积大为减小。封装和存贮器工艺的改进也改善了第三代计算机。

六十年代末还出现了另一类型的计算机即小型计算机。许多制造厂除了生产大而复杂的计算机以外，还在市场上出售一些小的性能有限的通用计算机。这些由于体积和价格而得名的小型计算机已经在许多方面应用，它们的普及率正以引人注目的速率增加。在大约30年间，我们看到了三代数字计算机。预测第四代计算机会有什么样的改革是令人感兴趣的。

知道了这些简要的历史背景之后，现在让我们介绍一些现代计算机的术语。

1.2 数字计算机

什么是数字计算机呢？数字计算机是一部可分成输入输出设备、存贮单元、运算单元和控制单元的电子装置。图1.1说明了这些组成部分的相互作用。存贮单元含有数据和已存入的一组指令，控制单元如果不是遇到下列意义类似的特殊指令，如分支、转移、

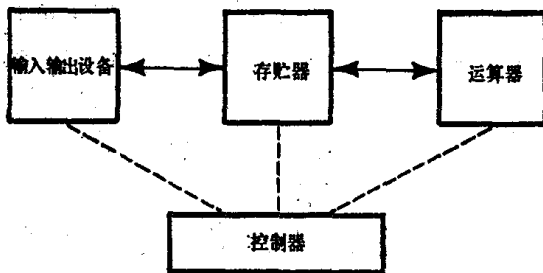


图 1.1 数字计算机

跳越或传递就按次序从存贮器中取出指令。分支指令能够产生循环和判定程序。存贮程序计算机的一个重要特点是它能够根据程序表中的一条指令，取出表中任何另一条指定的指令，并把它放到运算单元，按照需要改变它的内容，再把它送回存贮器的程序表。存贮程序计算机的大多数能力都是由于程序具有改变它自身的能力而来的。

数字计算机的运算器和控制器通常是把半导体逻辑元件用各种不同的组装方式构成的。较老的第二代计算机是在印刷电路板上安装晶体管、电阻、二极管等元件，而较新的第三代计算机用的是集成电路。数字计算机的存贮器件有些是用高速半导体元件构成的，但是占主要地位的还是磁性器件。磁性存贮器件包括：磁芯，它是标准的计算机存贮元件；磁盘、磁带、磁鼓，这些都是为存贮大量程序和数据用的辅助存贮器；还有磁性涂复线，它代替磁芯用于某些高速存贮器。存贮部件分成若干称为字的单元，每一单元根据它

的具体位置即存贮地址命名。存贮部件的特性参数是它的存贮周期。存贮器的存贮周期可以定义为从存贮器取出（读出）一个字再送回到原来那个单元所需要的时间间隔。

一个存贮单元的存贮地址这个概念相当于一个邮箱的邮寄地址。例如，邮局里有几排邮箱，每个邮箱用一个单独的号码确认它的位置。与此相类似，每个存贮单元都有一个用独一的号码标明的位置，这个号数就是存贮地址。

现在让我们进一步说明图1.1所示计算机每个部件的作用。按照前面的讨论，程序是存在计算机的存贮器中。但是，程序是通过通常称为外部设备的输入输出设备（I/O）和控制器送入存贮器的。程序通常用穿孔纸带或卡片形式送给计算机。稍后将说明输入输出设备怎么处理这些纸带或卡片。

计算机能用几种类型的外部设备输出数据。典型的输出设备是电传打字机或行式打印机。为了显示程序的计算结果可以用阴极射线管（CRT）。使用得最普遍的输入输出设备有模数转换器、数模转换器、绘图机、纸带读入机和穿孔机、读卡机和卡片穿孔机、磁带读入和记录装置以及电传打字机。

概括起来说，存贮器中存有特定程序的数据和指令。控制部件根据已存入的指令表，指挥运算单元和输入输出设备进行工作，直到程序运算完了为止。每个部件都在控制部件的同步指挥下，执行各自的任務。

1.3 计算机代码

1.3.1 运算码

我们已经简要地讨论了数字计算机存贮器中存放的指令和数据，但是没有提到这些指令和数据的形式。在计算机的存贮器里，数是以二进制（基数是2）的数制存贮起来的。二进制数（简称比

特)，除了仅有两个可能的数字 1 和 0 之外，非常像十进制数。

例如，十进制数 129 意味着 $1 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$ ，即每位数代表一个 10 的加权幂。请注意，这 10 个数字是从 0 到 $10-1=9$ 。二进制数例如 1011 里的每位数，可用 2 的加权幂来表示，即 $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ 。为了把二进制数变换成十进制数，上述加权和可按下式来确定：

$$(1011)_2 = 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = (11)_{10}$$

就是说二进制数的一零一一等于十进制数的十一。

数据是以二进制数的形式存贮在计算机的寄存器中，形式如下：

1	0	1	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---

这是一个在运算器或存贮器里都有的十位寄存器。在存贮器的寄存器里，数据叫做一个字（本例中字长是十位）。一些 1 或者 0 的组合是计算机的寄存器或存贮器所能存贮的唯一信息。对二进制数的各种排列样式赋予一定的含义就叫做编码。大多数计算机对数据的编码就是把刚刚提到过的二进制加权表示方式做一些简单的变化。

1.3.2 输入输出码

虽然计算机采用二进制数据，但是在诸如销售记录、品名表格和考试等级之类的记录中，用户还是喜欢用字母及数字的数据表示法。许多计算机都把一组称为字符集合的允许使用的字母数字符号，根据专用的二进制代码（称为美国信息交换标准代码 *ASCII*）进行编码。这种代码中，每个字母数字的及其他的专用字符（标点符号、代数运算符号等等）都用 8 位数编码。第二章里给出这种代码的一部分。假定我们想给计算机一个“ADD 1”（加 1）的信息。这个信息有 5 个字符，第四个字符是一个空位。用 *ASCII* 代码时，这信息就变成：

符 号	ASCII代码
A	0 1 0 0 0 0 0 1
D	0 1 0 0 1 1 0 0
D	0 1 0 0 0 1 0 0
	1 0 1 0 0 0 0 0
I	1 0 1 1 0 0 0 1

这信息送到计算机之后，计算机存贮器中的程序将接受它并且根据它来动作。

1.3.3 指令码

计算机的指令总是贮存在通常用磁心构成的主存贮器中。根据定义，指令也是用一些1或0的代码来表示。通常把指令分成单独编码的小字段。这些小字段是操作码（OP码）和存贮器地址。这些代码将在下一节说明。

1.4 计算机的指令

数字计算机的控制器从存贮器取出指令加以执行，这可能有几种类型的操作：（1）**算术指令**，它按照计算机程序员所编制的程序可以执行二进制数的加、减、乘、除。（2）**测试和比较操作指令**，可以用来确定两个二进制数据之间的关系（大于、小于、等于等等）（3）**分支及跳越指令**，它可以根据测试或比较的结果改变程序执行顺序。这类功能显著增加了程序的灵活性。（4）**输入输出指令**，它能把信息读入计算机，或者从计算机写出来并能控制外部设备。（5）**逻辑及移位操作指令**，它使计算机有能力解释和翻译所用的各种代码。这些指令在程序控制之下可实行按位操作。任何数字计算机的所有指令都可归结为这五类指令中的一种。

1.5 应用程序

数字计算机的程序设计是一门为使计算机有效地执行特定任务