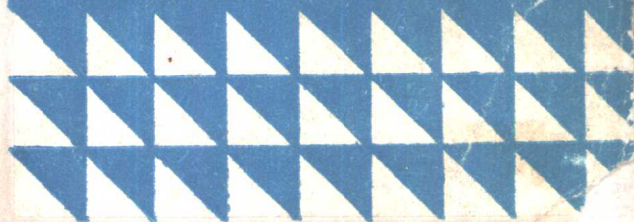


●〔美〕理查德G·陶得 著
●王懋江 吕汇川 译 叶庆凯 校

怎样绘制流程图 和编写程序



北京科学技术出版社

73.87221

9009933

怎样绘制流程图和编写程序

[美]里查德 G. 陶得 著

王懋江 吕汇川 译

叶庆凯 校

北京科学技术出版社

内 容 提 要

这是一本程序员的入门书。共分三部分十二章。第一部分介绍计算机的功能和工作原理,以及各种流程图符号。第二部分重点介绍设计程序语言PDL,以及如何用PDL绘流程图。第三部分介绍数据处理用的一些专用符号及系统流程图。本书可作为大中专院校计算机程序设计课程的一本教学参考书,也可供计算机程序设计人员参考。

From flowchart
to program

Richard G. Todd
TAB Books Inc. 1985.

怎样绘制流程图和编写程序

〔美〕里查德 G. 陶得 著

王懋江 吕汇川 译

叶庆凯 校

*

北京科学技术出版社出版
(北京西直门外南路19号)

新华书店北京发行所发行 各地新华书店经售
昌平建华印刷厂印刷

*

850×1168毫米 32开本 7印张 173千字
1989年12月第一版 1989年12月第一次印刷
印数1—2850册

ISBN 7-5304-0386-9/T.74 定价: 3.10 元

译者前言

本书是绘制流程图的一本入门书，作者凭着他在数学及计算机科学方面的丰富知识，写成此书。那些希望使自己有优美设计风格的初学程序员，应该学习此书。

在本书中作者首先表明，成功地程序设计和开发程序的秘诀在于画出恰当的流程图。作者在本书第一部分中，简要地介绍了计算机的功能及工作原理，然后逐步深入说明确定待解决问题的一些习惯作法——设计算法、绘流程图、将流程图翻译成语言代码。其次介绍了如何使用判定和循环，以及如何利用各种类型的流程图来表达程序的思路和逻辑。在本书的第二部分中，介绍了结构化程序设计技巧和在绘制流程图中如何使用“设计程序语言(PDL)”，以及如何使用子程序和子程序结构，并说明如何使用子程序层次图来防止偶然地数据破坏。在第三部分中，作者说明如何使用专用的数据处理符号及海存符号。作者通过强调程序设计基础而不是具体语言指令来表明，怎样开发一个高效无错、易调试、易修改扩充且易为他人理解的程序。

译者从观察一些人们工作习惯中看到，不少人编写程序无一定之规，想到那里就写到那里，程序调完之后，整理不出一套框图。因此，这些程序使用维护起来就特别困难，有感于此，译者译出此书，希望它能对国内广大计算机程序设计人员有所帮助。本书第1至第4章由王懋江译，第5至第12章由吕汇川译出，全书由北京大学力学系副教授叶庆凯校订。为了便于初学者阅读，原书程序中的英文，有的在程序清单语句后面的括号中译出，有些比较简单的词语，虽然在程序中未译出，但在正文中有注明。

由于译者水平有限，不妥之处在所难免，请广大读者随时指正。

王懋江

1988年5月于北京

引 言

这本书原打算作为一本绘制流程图的简单入门书。由于演化的缘故，显然把流程图和程序设计原则隔裂开来是困难的，而且是不明智的。不了解这些设计原则的人也能绘制流程图，但是这种流程图意义不大。因此，这本书就成了一本流程图和程序设计原则的综合读物。

因为本书是绘流程图的入门，因此，程序设计常处于次要地位。很多实例都不表示是解决某个具体问题的“最佳”方法，这些例子旨在说明如何使用流程图表示一个程序或过程。读者不妨利用新掌握的程序设计方法来改进随便哪一个例子。

目 录

引言

第一部分 绘制流程图和问题求解1

第 1 章 入门1

1-1 二进制：使用 0 和 12

1-2 位、字节和字8

1-3 简单的计算机8

1-4 计算机的功能11

1-5 小结12

1-6 练习12

第 2 章 问题求解过程的开始15

2-1 程序的功能15

2-2 产生程序的三个阶段16

2-2-1 确定问题16

2-2-2 确定算法17

2-2-3 绘流程图18

2-3 计算算术表达式22

2-4 一个例子23

2-5 小结25

2-6 练习25

第 3 章 问题求解过程的精加工27

3-1 流程图的作用27

3-2 结束问题求解过程29

3-2-1 转成微流程图29

3-2-2 转成BASIC 程序31

3-2-3 运行和测试程序31

3-3 问题求解过程的回顾34

3-4 计算表达式，第二部分34

3-5 使用公式36

3-6	使用注解符号	37
3-7	小结	39
3-8	练习	41
第 4 章	做判定	42
4-1	判定和汇合符号	42
4-2	使用判定符号	45
4-3	用BASIC实现判定符号	47
4-3-1	第一遍通过流程图	48
4-3-2	第二遍通过流程图	52
4-4	布尔代数	52
4-4-1	真和假	52
4-4-2	布尔表达式	53
4-5	再谈漏气轮胎问题	56
4-6	小结	60
4-7	练习	61
第 5 章	循环	63
5-1	公式和函数	63
5-2	条件循环	64
5-3	数组	72
5-4	迭代循环	74
5-5	FOR-NEXT 语句	77
5-6	小结	80
5-7	练习	81
第二部分	绘制流程图和程序设计技巧	82
第 6 章	设计程序语言 (PDL)	82
6-1	PDL概述	82
6-2	PDL程序是什么?	83
6-3	控制构造	84
6-3-1	DO语句	84
6-3-2	IF语句	86
6-3-3	WHILE构造	88
6-3-4	REPEAT语句	89
6-3-5	FOR 循环	90

6-4	PDL的概念	91
6-5	一个PDL程序设计的例子	92
6-5-1	逐步细化	94
6-5-2	字符串	95
6-5-3	对表的抽象看法	97
6-5-4	表管理	99
6-5-5	全局数据	100
6-5-6	第一级细化	104
6-5-7	第二级细化	107
6-5-8	实现一个表	107
6-5-9	第三级细化	112
6-6	小结	117
6-7	练习	118
第 7 章	流程图和设计程序语言 (PDL)	119
7-1	PDL程序的符号表示	121
7-1-1	进一步考察符号表示	121
7-1-2	表示PDL的控制构造	124
7-2	把PDL转换成流程图	128
7-3	连接符号	133
7-4	转换电话号码簿程序	136
7-5	小结	146
7-6	练习	147
第 8 章	子程序	148
8-1	参数和副作用	148
8-2	一个程序设计例子	153
8-3	无参数的子程序	154
8-4	在BASIC中传递参数	156
8-5	执行子程序	159
8-6	多个子程序嵌套	163
8-7	小结	165
8-8	练习	165
第三部分	绘制流程图的其它用处	167
第 9 章	数据处理和数据流程图	167

9-1	工作环境	187
9-2	排序	187
9-3	合并	172
9-4	使用排序和合并	175
9-5	抽取	175
9-6	小结	177
9-7	练习	177
第 10 章	专用 I/O 设备	179
10-1	输出设备和要求	179
10-2	输入设备和要求	183
10-3	I/O 接口设备	184
10-4	I/O 需要的流程图符号	184
10-5	使用 I/O 符号	185
10-6	小结	190
10-7	练习	190
第 11 章	大容量存储设备	191
11-1	存储概念	191
11-2	文件操作	191
11-3	文件存取方法	193
11-4	BASIC 和文件	194
11-5	大容量存储设备的一些例子	195
11-6	对大容量存储的要求	196
11-7	大容量存储的流程图符号	196
11-8	一个例子	197
11-9	小结	200
11-10	练习	200
第 12 章	系统流程图	201
12-1	准备系统流程图	201
12-1-1	标明系统要求	201
12-1-2	数据文件	201
12-1-3	程序要求	203
12-1-4	输入数据	203
12-1-5	输出数据	204

12-2	构造系统流程图	208
12-3	使用系统流程图	210
12-4	小结	210
12-5	练习	211

第一部分 绘制流程图和问题求解

第 1 章 入门

每个人都希望他所从事的事情获得成功。程序员也不例外。但是每一年，成千上万个学习编制计算机程序的人发现，他们不得不拚命干那些排除语法错误和程序错误的工作。

成功的程序设计不是不可能的。对于一个要获得成功的程序员来说，有三个要点是必需的：

①**设备**。程序设计需要实践。几乎任何计算机都行，但是袖珍计算器和某些便携式计算器可能不便于使用，因为它们显示的东西太小了。你还应该有一个盒式磁带或软盘系统来存你写的程序。

②**动力**。象其它技术一样，学习程序设计需要有充分的动力。你必须乐意花费一些时间来认真地学习本书所介绍的一些概念。

③**指导**。本书阐述了问题求解的一些原则和一些程序设计的技巧。不要忽略其他的知识来源：程序设计课程、其它程序员的经验，书籍和期刊论文可能是十分有用的，特别对于一些专门问题会有所帮助。

你的进步取决于你当前的状态，也就是取决于你是计算机用户呢，还是用户团体的成员。评估你的状态：你将使用的是哪类计算机？关于怎样使用它，你知道多少？你知道谁能帮助你解决问题？你有多少时间以及你乐意花费多少时间学习这个材料？对你的计算机你知道的愈多，你所认得的有经验的程序员愈多，以及你能专心用于程序设计技术的实践时间愈多，则你产生满意的程序的速度愈快。目前我们将考虑计算机的工作原理。不一定需

要了解计算机所有的内部工作情况——很多程序员不了解，但是，一般性地知道计算机如何工作，将会有助于你了解为什么一个程序是这样工作的。当你知道计算机用BASIC指令做些什么的时候，你将看出，许多BASIC指令都是合理的。

1-1 二进制：使用0和1

计算机的一个重要功能是它能存储数据。最早的计算机利用机械开关来做这件事情。开关，也称做**继电器**，它可能处于接通或断开的状态。可以通过电子装置把它们转换即**触发**。适当的电信号可以把一个断开的继电器接通，或把接通的继电器断开。

继电器有两个状态：接通或断开。正好适合于存储二进制系统的数据。二进制数使用数0和数1。若继电器处于断开状态，则它代表0。若处于接通状态，则它代表1。利用一串继电器可以代表完整的数。图1-1表明如何存储二进制数0110。连到继电器上

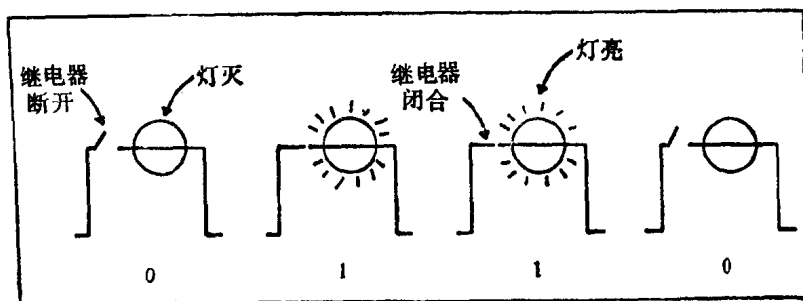


图 1-1 用继电器表示0110

的灯泡指示出继电器是否是接通状态。

在近代计算机中，现代化技术已经取消了继电器。就象继电器是过去计算机的基础一样，固态器件，有时称作**触发器**，正是今天计算机的基础。虽然它们没有活动部件，但它们也以开关方式工作，而且仍能看做是一个小开关，因而，现代化的计算机技术仍然使用二进制。

在我们考虑二进制数以前，让我们仔细考虑我们已非常熟悉的十进制数，并看看它们是如何起作用的。

十进制使用十个数字。其中九个代表数量1, 2, 3, 4, 5, 6, 7, 8, 9。第十个数是0，用于占位，它不代表一个数量，但在数的表示中它占据位置。

让我们来考虑一个普通的数，并看看它实际是什么。数139能够看作是 $100+30+9$ 。其中每一个都使用了可能跟有（或可能没跟有）零的数（1~9）。最右边的数9处在数139的个位上；9代表了9个。中间的数3处在十位上；它代表了三个十。最后一个数1处于百位上；代表了一个一百。这一事实形象地表示在图1-2上。想象分组的另一方法是：若你能挑选139件东西，则你可以首先一次挑100个，接着三次每次挑十个，最后九次每次挑1个。

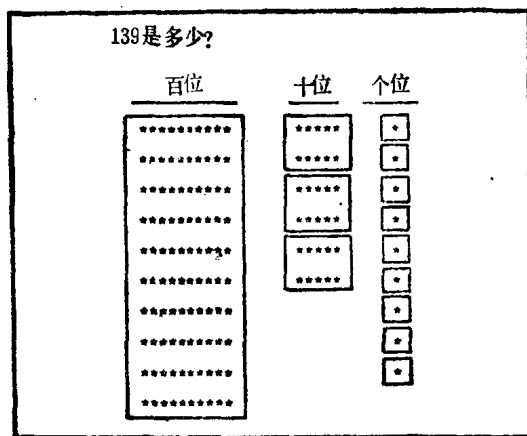


图 1-2 139分成个位、十位和百位三组

通过把十进制数中每一位数表示成十的方幂的方法可以把十进制数的表示一般化。即第二位值是10，第三位值是 10×10 ，即100，第四位值是 $10 \times 10 \times 10$ ，即1000，如此等等。第一位的值是1，这对任何进制的数都一样。值可以用指数表示。指数表明

一个数自身相乘多少次。具有指数为5的数意味着该数要乘自身5次。若指数是负的，则应取该数的倒数。具有指数为-3的一个数，意味着该数乘自身3次，然后取倒数。表1-1是十进指数表。

表 1-1 十 的 方 幂 表

位	值
1	$10^0=1$
2	$10^1=10$
3	$10^2=100$ $=10 \times 10$
4	$10^3=1,000$ $=10 \times 10 \times 10$
5	$10^4=10,000$ $=10 \times 10 \times 10 \times 10$
6	$10^5=100,000$ $=10 \times 10 \times 10 \times 10 \times 10$
7	$10^6=1,000,000$ $=10 \times 10 \times 10 \times 10 \times 10 \times 10$
8	$10^7=10,000,000$ $=10 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10$

二进制数只使用两个数：0和1。不象十进制，二进制数每一位的值是二的方幂，如1，2，4，8，16等，而不是十的方幂。表1-2列出了二的方幂及其值。因为在任何进制内，0都用于占位，故在二进制内，只允许我们用数字1表示一个值。

在十进制数内，数字2～9代表了每一位值的倍数。记着，在十位上的3代表了三个十。然而，在二进制数内，只有一个数字1表示数量，所以我们决不会与倍数打交道。若二进制数的某特定位上有1，则我们仅仅把那个位的值加到该数的总值上。这样以来，二进制数1101的值是1（最右边位上的值）加上4（第三位的值），加上8（第4位的值）： $1+4+8=13$ ，所以1101的值是13。

一旦你知道如何使用十进制数表示二进制数的各位的值，就

表 1-2 二 的 方 幂 表

位	值
1	$2^0 = 1$
2	$2^1 = 2$
3	$2^2 = 4$ $= 2 \times 2$
4	$2^3 = 8$ $= 2 \times 2 \times 2$
5	$2^4 = 16$ $= 2 \times 2 \times 2 \times 2$
6	$2^5 = 32$ $= 2 \times 2 \times 2 \times 2 \times 2$
7	$2^6 = 64$ $= 2 \times 2 \times 2 \times 2 \times 2 \times 2$
8	$2^7 = 128$ $= 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$

能够很容易地把一个二进制数转换成十进制数。为了计算二进制数的十进制值，只要把该二进制数有1的各位上的值加在一起即可；图1-3中表示了此过程。图1-4是从二进制数转到十进制数的习题。

二进制数101101₂是十进制数多少？

位数:	6	5	4	3	2	1
每位的内容:	1	0	1	1	0	1

因6,4,3和1位均有1,我们把它们的值相加。

第一位的值是	1
第三位的值是	4
第四位的值是	8
第六位的值是	<u>+32</u>
总数是	45

101101₂的十进制数值是45₁₀。

图 1-3 二进制到十进制的转换

从十进制数到二进制数转换略微麻烦点。从找出小于等于十

二进制数11101₂是十进制数多少?

位数:	5	4	3	2	1
每位的内容	[]	[]	[]	[]	[]

因[], [], [], 和[]位均有1, 我们把它们的值相加,

第[]位的值是	[]
第[]位的值是	[]
第[]位的值是	[]
第[]位的值是	+ []
总数是	[]

因此, 11101₂的十进制数是[]₁₀.

图 1-4 从二进制到十进制转换的练习

进制数的2的最大方幂(1, 2, 4, 8, 16...)开始, 例如, 我们的十进制数是10, 则我们选8。在代表我们选出值的位上该二进制数将有1, 在此例中它是第4位。把原来的数减去2的方幂后, 对其差重复这一过程, 在目前情况下它是2(10-8), 一直到所得的差为零时结束。图1-5是如何做这件事情的一个例子。作为练

十进制数27₁₀是二进制数多少?

27	
-16	→ 第五位的值
11	
-8	→ 第四位的值
3	
-2	→ 第二位的值
1	
-1	→ 第一位的值
0	→ 转换完成

我们知道所求的二进制数在第一、二、四和五位中有1。

位数: 5 4 3 2 1

每位的内容, 1 1 0 1 1

因此, 27₁₀的二进制数是 11011₂.

图 1-5 十进制到二进制的转换

习把图1-6的问题从十进制转成二进制。

十进制数 55_{10} 是二进制数多少？

55			
$\begin{array}{r} -[] \\ \hline [] \end{array}$	$\longrightarrow$	第[]位的值	
$\begin{array}{r} -[] \\ \hline [] \end{array}$	$\longrightarrow$	第[]位的值	
$\begin{array}{r} -[] \\ \hline [] \end{array}$	$\longrightarrow$	第[]位的值	
$\begin{array}{r} -[] \\ \hline [] \end{array}$	$\longrightarrow$	第[]位的值	
$\begin{array}{r} -[] \\ \hline [] \end{array}$	$\longrightarrow$	第[]位的值	
0	$\longrightarrow$	转换完成	

我们知道所求的二进制数在第[], [], [], []和[]位中有1。
 位数： 6 5 4 3 2 1
 每位的内容， [] [] [] [] [] []
 因此, 55_{10} 的二进制数是 []₂.

图 1-6 从十进制到二进制转换的练习

你可能已经发现，关于二进制数和十进制数的这种讨论有些混乱，因为难于把一个十进制数同一个二进制数区分开。例如，考虑图1-7，计算交点的个数。有多少？是1110呢 还是14呢？你可能说，两者都对，因为二进制数1110与十进制数14是相等的。知道数是以什么数制写出的，对解释该数的正确值是必需的。为了表明一个数使用什么样的数制。在该数后以下标形式写出它的

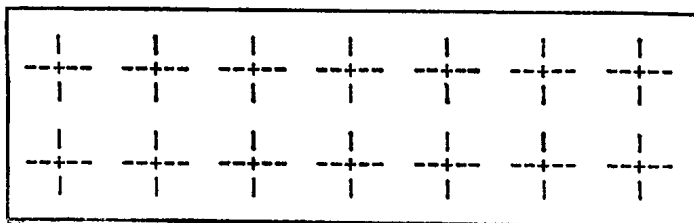


图 1-7 交点的个数