



国外经典教材·电子信息

PEARSON
Prentice
Hall

Principles of Digital Design

数字设计原理

(美) Daniel D. Gajski 著

李敏波 译



清华大学出版社

国外经典教材·电子信息

数字设计原理

(美) Daniel D.Gajski 著

李敏波 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Principles of Digital Design by Daniel D.Gajski, Copyright © 2003

EISBN: 0-13-301144-5

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-1773

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

数字设计原理/(美)盖吉斯基(Gajski,D.D.)著; 李敏波译.—北京: 清华大学出版社, 2005.5

书名原文: Principles of Digital Design

(国外经典教材·电子信息)

ISBN 7-302-09745-3

I. 数… II. ①盖… ②李… III. 数字电路—电路设计 IV. TN79

中国版本图书馆 CIP 数据核字(2004)第 105436 号

出版者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-62770175

组稿编辑: 曹 康

封面设计: 久久度文化

印刷者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 22 字数: 563 千字

版 次: 2005 年 5 月第 1 版 2005 年 5 月第 1 次印刷

书 号: ISBN 7-302-09745-3/TN·217

印 数: 1~4000

定 价: 39.80 元

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

文稿编辑: 侯 彧

版式设计: 康 博

装 订 者: 三河市金元装订厂

出版说明

近年来,我国的电子信息产业发展迅速,力争要成为“21世纪的世界制造中心”。这就迫切需要我国的高校能够培养出大量符合企业需求的高素质电子信息类人才。但是,目前传统的电子信息类教材已经落后于时代发展对电子信息教育的要求,急需一批门类齐全、具有国际水平的经典教材。引进国外的优秀电子信息教材,一方面可以了解和吸收国际现今的教学思想和教学方法,推动国内高校的课程改革和教学国际化进程,乃至对国内重点大学建设国际一流大学都能够提供宝贵的借鉴作用;另一方面能够培育出更多具有国际水准的电子信息类人才,提高我国电子信息产业的核心竞争力。为此,清华大学出版社从全球最大的高等教育出版机构——培生教育出版集团(Pearson Education Group)引进了这套“国外经典教材·电子信息”教材。

在该套教材的引进和出版过程中,我们邀请了一大批国内高校电子信息类专业的知名教授和相应领域的专家,与我们共同成立了专门的教材编审委员会,根据国内高校电子信息各专业的课程体系和培养方向,分期分批地从 Pearson 出版的电子信息系列教材中挑选出质量高、针对性强、适合国内教学的优秀教材。确定选题后,编委会委员直接参与或组织教材的翻译和审校,以确保教材内容的高质量和适应性。另外,为了更好地服务于教学和增强立体化教学的力度,我们在引进教材的同时也引进了与教材相配套的教学资料。

该套教材的范围是电子信息和电子工程学科所属各专业的专业基础课和专业课,读者对象是对应专业的本科生,同时兼顾相关工程学科各专业的本科生或研究生。该套教材既可作为相应课程的教材或教学参考书,也适于相应技术领域的工程师和技术人员参考或自学。

由于我们自身能力有限,该套教材在使用过程中很可能还会发现一些缺憾或问题,欢迎使用过的广大师生向我们提出意见或建议。同时,也非常期望广大读者向我们推荐优秀的国外电子信息类教材,同我们一起建设适合我国高等院校电子信息教学的优秀教材。

清华大学出版社

前 言

基本原理

随着 VLSI 技术的发展，现在已经可以制造拥有几百万个晶体管的芯片。手工设计这些芯片几乎是不可能的，而 CAD 工业的涌现，为这些大型芯片的设计提供了软件工具。因此，以前介绍的大多数数字设计技术，现在都集成在 CAD 工具中，设计者可以将产品的设计到制造作为一个整体进行考虑。为此，设计者现在必须掌握多种技术：需求分析、规范制定、设计建模、软硬件协同设计、芯片综合、验证、仿真和测试。从这个意义上说，设计知识的概念需重新定义，由涵盖所有设计技术的知识转变为对原理的理解，以及利用 CAD 工具开发各种设计方案和实现技术的能力。

本书的目标

1. 设计原理

本书深入介绍了设计的基本原理，但并不要求读者记忆许多烦琐的技术细节。本书引入了通用元件库，从而使设计工作得以简化。这些通用元件库反映了实际的设计约束，比如成本和延时。书中利用这个通用的库来解释提出的所有概念，并实现书中练习的所有示例。

2. 设计流程

整体来看，本书的结构主要是对完整的设计流程的介绍，而不是介绍某个设计阶段的设计技术。因此，本书首先介绍数字设计的流程，然后贯穿全书，说明各阶段在整个设计流程中的作用。

3. 理论和实际相结合

遗憾的是，许多关于数字设计的书籍都倾向于将基本的原理一带而过，或者正好相反，讨论那些很少用到的细节性的理论。为了避免这些问题，在对本书的组织上，从一开始就体现出原理与实际设计之间的必然联系。只有当理论有助于简化设计的说明时，本书才会介绍这些理论。

4. 详细的例题

每一章中都会给出若干个综合性例题，以说明设计流程中的每一个步骤。详细的例题是最好的学习工具，因为读者可以跟踪设计步骤，评价设计方案，并探究设计决策背后的原因。与文字相比，读者更偏爱详细的例题，因为通过例题可以很容易地领会新概念的涵义。通过这种

方式，利用详细的例题，代替简单的定义概念，让读者将概念与设计流程相结合，并得出从一个设计步骤到下一个步骤的原因和方法。

5. 寄存器传输级设计

除此之外，本书的另一个重要目的，是让读者在设计方法学方面，由逻辑和时序级设计，扩展到包括通用和专用处理器，以及芯片设计的寄存器传输级设计。在介绍寄存器传输级设计时，本书的目标在于说明逻辑和时序设计与计算机的组织结构之间的联系。这个部分在许多书籍中都被省略了，将如何在处理器和其他芯片的设计中使用加法器、计数器和存储器的问题留给了读者。

本书的结构

1. 现代设计流程

在第1章中，首先介绍系统设计学科，定义了设计流程及其主要任务。这一章是后续章节的基础，同时也为整本书提供了引导图。本章还讨论了现代设计技术，以及促进其使用的CAD工具。另外，还简要介绍了制造的流程，并讨论了设计与制造之间的关系。第1章的目的，是通过介绍现实世界中设计科学的广泛应用，激发读者学习系统设计的兴趣。

2. 二进制数以及数据的表示法

在第2章中，介绍了数字系统中使用的数据模型。对二进制表示法的理解，对于理解数字系统中的运算是很关键的，因此本章将详细介绍二进制数（定点数和浮点数），并讲解二进制中的各种算术运算。此外，还说明了几种其他类型的数据模型，以及如何有效地使用这些数据模型。

3. VLSI 技术的影响

设计技术必须能够在给定系统的要求与可行的实现技术之间提供一种折衷方案。因此，本书很早就介绍了VLSI技术的原理和约束，并几乎在每一章中都讨论它们对各种设计技术的影响。但是，本书不要求读者具备电子学知识，也不需要读者处理电路。相反，我们使用表示标准布尔、算术算子以及简单和复杂数据结构的元件。本书不使用特殊的元件库，如TTL或商用CMOS库，因为本书编写所遵循的原则是：理解各种元件的原理以及了解如何使用这些元件，比知道所有商用电路的确切名称和封装更重要。因此，本书使用不依赖于任何特有技术的通用元件。

4. 现代逻辑和时序设计方法

现代设计库一般都很小，以便简化其维护以及使用这些库的设计算法和CAD工具。本书全文使用一个简单的门库，并用该基本库中的元件搭建所需要的高层次元件。例如，在第5章和第7章中，介绍了处理器和专用集成电路设计所需要的所有组合和时序元件。但同时，丰富的CAD工具也不容忽视，因为这些工具可以减少设计者进行过多的复杂的手工优化。因此，本书着重于优化的基本目标和原理，并举例说明现代设计流程。

5. 寄存器传输级设计的方法

在过去,超越了时序综合这一级的设计技术都被认为是一门艺术,在设计课本中几乎没有涉及过。为了填补这个空缺,第8章中将介绍带有数据通路的有限状态机的概念。定义了这个概念之后,我们将介绍如何把用标准流程图或寄存器传输图表示的任意算法,映射到由数据通路和控制单元组成的硬件上,其中控制单元包含在第5章和第7章中定义的组合元件和时序元件中。

6. 处理器设计

在第9章中,将介绍如何利用第8章中介绍的寄存器级设计体系和方法来设计CPU。另外,本章还将介绍指令集、计算机体系结构和处理器设计方法学。为了说明这些概念,第9章还利用数据预取和分支预测,介绍了16位CISC和32位RISC处理器的设计。

编写特点

1. 引导图

为了方便读者学习,我们利用总结了本书所述论题的论题框图,来定义它们的相互关系。每一章的开始都总结了这一章的目标,以及所要论述的主题。当引入新的概念时,这个概述可以帮助读者把前面的章节中提出的概念和这些新概念联系起来。

2. 设计过程

每种设计技术都是逐步提出的,最后总结成流程图的形式。流程图让读者以有序的方式解决问题,并且建议如何将某种技术应用到新的领域,或者将来涌现的新技术。

3. 综合例题

本书利用带有解答的例题来说明与应用相关的原理和设计过程。在每一章中,我们利用少量的例题来说明各种可选的设计方案,引导读者从定义到最后的原理图,遍历必要的设计步骤。

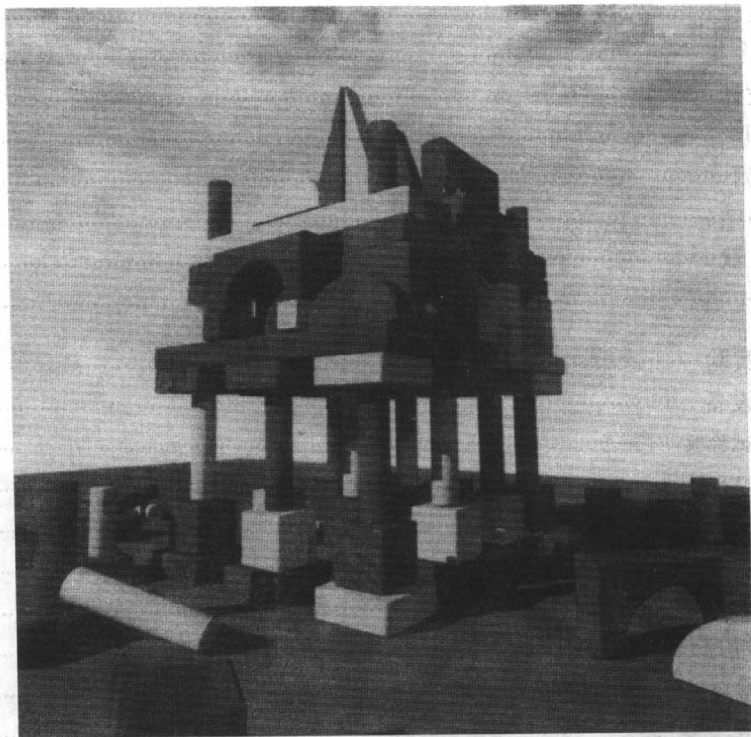
课程选择

基于本书的组织结构,在两种不同类型的课程中,本书应该有相同的效果。例如,在关于数字设计的入门课程中,本书可以突出传统的组合和时序设计,这在国外的大多数计算机科学和计算机工程课程中都有论及。为这样一个课程准备的材料在第1~7章给出。另外,我们给出了关于通用和专用处理器的寄存器传输级设计的材料,这通常在数字设计的第二课程和计算机体系结构的入门课程中论及。

在这个课程中,首先学习基本的寄存器传输元件:第5章中组合元件,第7章的学习时序元件。课程主要由寄存器级设计技术和第8章介绍的专用处理器的综合设计流程组成。最后在第9章中,读者将学习现代指令集和通用CISC与RISC处理器的设计。

在任何一种情况下，我们都建议用实验课程对讲座进行补充。在实验课程中，学生可以利用用于设计捕获、建模、仿真、验证、综合、时序分析、平面图设计、物理设计以及流程设计其他方面的现代 CAD 工具来完成实际的设计问题。以我们的经验，实验课程是非常受学生欢迎的，因为这样可以使他们在真实的工作环境中，将学到的概念与实际设计联系起来。作为教师的辅助工具，附录中介绍了几种不同的实验设备并给出了实验室实验的几个例子。

Daniel D.Gajski



设计不只是一套技术或者是如何实现的方法，而是一个过程，在这个过程中，为了最终产品，不同的人共同为之努力。这些人各自从不同的角度看待产品，有不同的技术专长，使用不同的工具。综合起来之后，他们就将一个概念变成一个真实的产品。

目 录

第 1 章 引言	1
1.1 设计描述	1
1.2 抽象层次	5
1.3 设计流程	7
1.3.1 设计规范	7
1.3.2 库的开发	7
1.3.3 设计综合	7
1.3.4 设计分析	8
1.3.5 文档	8
1.3.6 制造	9
1.4 CAD 工具	9
1.4.1 设计捕获和建模	9
1.4.2 综合工具	10
1.4.3 验证和仿真	10
1.4.4 物理设计	10
1.4.5 测试	11
1.5 典型的设计流程	11
1.6 引导图	11
1.7 本章小结	13
1.8 参考文献	13
1.9 习题	14
第 2 章 数据的类型与表示	17
2.1 定位数制	17
2.2 八进制和十六进制数	18
2.3 数制转换	20
2.4 进制数的加法与减法	23
2.5 负数的表示	25
2.5.1 有符号表示	25
2.5.2 补码数制	27
2.6 二进制补码的加法和减法	29
2.6.1 加法规则	29
2.6.2 减法规则	30
2.7 二进制乘法	31

2.8	二进制除法	34
2.9	浮点数	35
2.10	十进制数的二进制码	37
2.11	字符码	38
2.12	检错和纠错码	39
2.12.1	检错码	41
2.12.2	纠错码	42
2.13	汉明码	43
2.14	本章小结	45
2.15	参考文献	45
2.16	习题	46
第3章	布尔代数与逻辑设计	51
3.1	代数的性质	51
3.2	布尔代数的公理化定义	52
3.3	布尔代数的基本定理	54
3.4	布尔函数	56
3.4.1	函数的反	57
3.4.2	代数变换	58
3.5	正则形式	59
3.6	标准形式	64
3.7	其他逻辑运算	66
3.8	数字逻辑门	68
3.9	多输入和多算子的扩展	71
3.10	门的实现	75
3.10.1	逻辑电平	75
3.10.2	噪声容限	76
3.10.3	扇出	78
3.10.4	功耗	79
3.10.5	传输延时系列	79
3.10.6	双极型逻辑系列	80
3.10.7	MOS 逻辑系列	82
3.11	VLSI 技术	85
3.12	本章小结	91
3.13	参考文献	91
3.14	习题	92
第4章	布尔函数的化简	97
4.1	图表示法	97
4.2	化简的卡诺图法	106
4.3	不确定状态	109

4.4	制表法	111
4.4.1	质蕴涵项的生成	111
4.4.2	最小覆盖的生成	113
4.5	门阵列的技术映射	116
4.6	定制库的技术映射	122
4.7	无险象设计	125
4.8	本章小结	128
4.9	参考文献	129
4.10	习题	129
第5章	组合元件	133
5.1	行波进位加法器	133
5.2	超前进位加法器	134
5.3	加法器/减法器	138
5.4	逻辑单元	139
5.5	算术逻辑单元	140
5.6	译码器	143
5.7	选择器	145
5.8	总线	148
5.9	优先编码器	149
5.10	比较器	151
5.11	移位器和循环移位器	153
5.12	只读存储器	156
5.13	可编程逻辑阵列	158
5.14	本章小结	160
5.15	参考文献	160
5.16	习题	160
第6章	时序逻辑	165
6.1	SR 锁存器	166
6.2	门控 SR 锁存器	168
6.3	门控 D 锁存器	169
6.4	触发器	170
6.5	触发器类型	174
6.6	时序逻辑的分析	178
6.7	有限状态机模型	183
6.8	时序逻辑的综合	185
6.9	FSM 模型的获得	187
6.10	状态化简	188
6.11	状态编码	191
6.12	记忆元件的选择	196

6.13	优化和时序	198
6.14	本章小结	200
6.15	参考文献	200
6.16	习题	200
第 7 章	存储器件	207
7.1	寄存器	207
7.2	移位寄存器	209
7.3	计数器	210
7.4	十进制(BCD)计数器	213
7.5	异步计数器	214
7.6	寄存器文件	216
7.7	随机存取器	219
7.8	下推栈	225
7.9	先进先出队列	228
7.10	简单数据通路	232
7.11	通用数据通路	238
7.12	控制单元的设计	240
7.13	本章小结	242
7.14	参考文献	243
7.15	习题	243
第 8 章	寄存器传输级设计	247
8.1	设计模型	247
8.2	FSMD 定义	249
8.3	算法状态机图	253
8.4	利用 ASM 图进行综合	259
8.5	寄存器共享(变量合并)	264
8.6	功能单元共享(算子合并)	268
8.7	总线共享(连线合并)	273
8.8	寄存器合并	276
8.9	链接和多循环	277
8.10	功能单元流水线	279
8.11	数据通路流水线	282
8.12	控制流水线	285
8.13	进程安排	287
8.14	本章小结	294
8.15	参考文献	295
8.16	习题	295

第 9 章 处理器设计	299
9.1 指令集	300
9.2 寻址方式	303
9.3 处理器设计	305
9.4 指令集设计	307
9.5 CISC 设计	309
9.6 精简指令集	316
9.7 RISC 设计	319
9.8 数据预取	322
9.9 分支预测	324
9.10 本章小结	327
9.11 参考文献	327
9.12 习题	328
附录 A 实验室实验	330
A.1 实验室设备	330
A.1.1 试验板	330
A.1.2 FPGA 板	331
A.1.3 输入和仿真实验室	331
A.1.4 描述和综合实验室	332
A.2 实验室种类	332
A.3 试验板和 FPGA 实验	334
A.4 仿真实验	335
A.5 建模实验	335
A.6 逻辑和行为综合实验	337
A.7 参考文献	337

第 1 章 引 言

对于任何一个特定产品，其设计流程包括将其功能概念化，到最终形成制造设计图的所有任务。在这个过程中，会涉及到很多人。每个人执行一个特定的任务。

例如，市场部研究市场需求，并决定新产品的需求。技术人员选择技术、元件和供应商，而辅助小组则采购或开发软件工具，这些工具可以支持产品及其各部分的设计。产品设计者把产品的需求转换成制造的设计图。测试工程师研究出测试策略，以验证设计的正确性，并对制造出来的产品进行故障测试，而生产工程师则研究生产计划和生产进度。

1.1 设计描述

在产品的定义、设计和制造过程中，所涉及到的每个人都会有不同的观点，从而衍生出来不同的产品，这里，每个人都需要特定的信息来进行支持。因此，每一件产品，每一个设计，都会有几种不同的描述或观点，这些描述和观点所侧重的信息是不同的。另外，相同的描述在设计 and 制造周期的不同阶段，通常要求不同的细节层次。最常用的三个描述类型为行为描述、结构描述和物理描述。

行为或功能描述定义为：把设计看成一个黑盒子，将其行为表示成其输入值和过去时间的函数。换句话说，行为描述描写其功能，但并不给出具体的设计实现，定义了黑盒子对输入组合的响应，但没有描述如何利用具体元件设计和搭建该黑盒子。

相比较而言，结构描述是将黑盒子定义成一组元件，以及元件之间连接的一种描述。和行为描述不同，它规定了产品的实现，但不明确涉及功能。当然，在某些情况下，具体功能可以从它的互连元件推导出来。然而，以这种方法推导一个设计的功能是很困难的，且容易产生误差，因为有些元件并没有用到全部的功能，或者其功能也有可能通过对元件的输入和输出进行编码而修改。此外，在元件数目非常大的情况下，比如 10 000，要发现它们的功能是不可能的，尤其是当今的技术，已经可以制造出具有几百万门晶体管的芯片。

最后，物理描述是指规定黑盒子的物理特性，提供在结构描述中包含的各元件和连接的尺寸和位置。注意，当结构描述提供设计的连通性时，只有物理描述说明了各元件之间精确的空间关系。换句话说，物理描述用来说明用于制造的产品设计，规定其重量、大小、散热、功耗和各输入、输出管脚的位置。

例 1.1 简单的闹钟

问题：列出一个简单闹钟的行为、结构和物理设计描述。

解答：一个简单的闹钟由液晶显示器(LCD)以及五个开关组成，液晶显示器用来显示秒、分和小时，五个开关分别用作时间设定(S_1)、闹铃设定(S_2)、分针拨快(S_3)、时针拨快(S_4)和闹铃使能(S_5)。当 S_1 闭合的时候，可以通过按动 S_3 和 S_4 来设定时间。 S_3 和 S_4 每按动一次，分针或时针就调快 1 个单位，并且将调整的分针和时针的变化在闹钟的显示器上显示出来。当 S_2 闭

合的时候，可以用相同的方式设定闹醒时间，通过按动 S_3 和 S_4 来拨快闹醒时间的分针和时针。当设定闹醒时间时，闹醒时间的时和分都在 LCD 上显示出来。最后，当开关 S_5 闭合的时候，闹铃使能，从而，当闹钟到达闹醒时间时，产生 5 秒钟的嗡嗡声。

该闹钟的行为可以用三个同时发生的任务或进程来描述：计时、设置和闹铃。这些进程都可以依次用一个简单的流程图描述，流程图由描述判断的菱形框和用于描述运算的矩形框组成。表示这三个进程的流程图如图 1.1 到 1.3 所示。

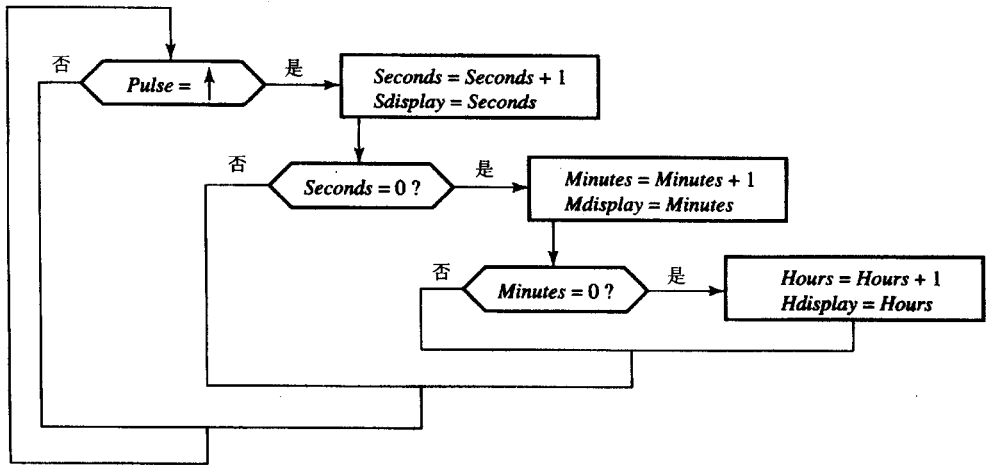


图 1.1 闹钟的行为描述：计时进程

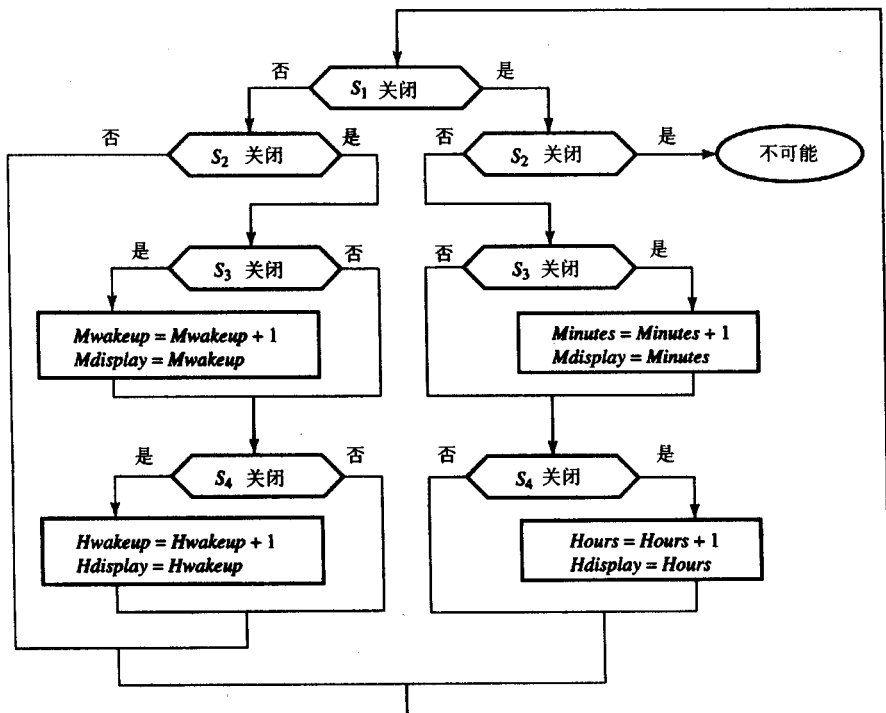


图 1.2 闹钟的行为描述：设置进程

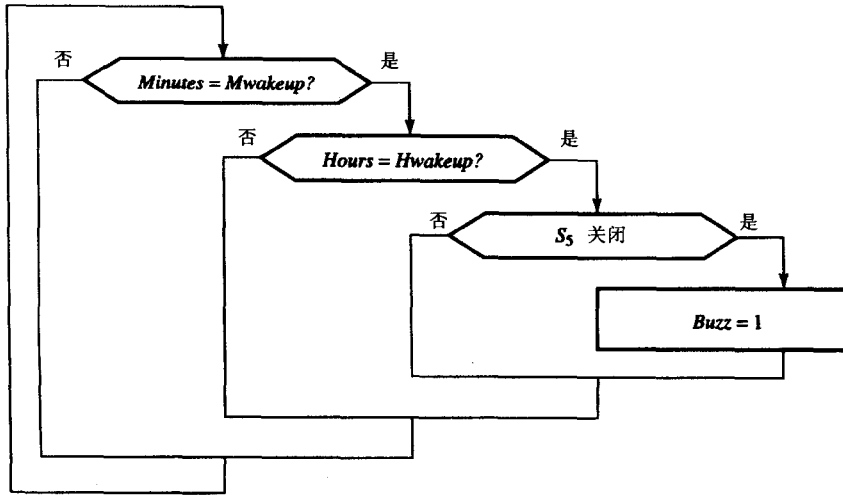


图 1.3 闹钟的行为描述：闹铃进程

如图 1.1 所示，计时进程有一个输入信号 *Pulse* 和六个内部信号：*Seconds*、*Minutes*、*Hours*、*Sdisplay*、*Mdisplay* 和 *Hdisplay*。*Pulse* 信号用于在闹钟内部对秒钟计数，持续时间为 1 秒，在第一个半秒内值为 1，下一个半秒内值为 0。三个内部变量——*Seconds*、*Minutes* 和 *Hours*——用于对秒、分和时计数。*Seconds* 和 *Minutes* 以 60 为模计数——也就是说，只要计数达到 59 就清零。*Hours* 以 12 为模计数，从 1 开始，计数到 12 时返回到 1。运行时，每当 *Pulse* 信号从 0 变为 1，在流程图中用 ↑ 指示，*Seconds* 就加 1。类似的，每当 *Seconds* 达到 0，*Minutes* 就加 1；每当 *Minutes* 达到 0，*Hours* 就加 1。

另外三个内部变量——*Sdisplay*、*Mdisplay* 和 *Hdisplay*——分别用来显示秒、分和时。注意，每当相应的计时变量——也就是 *Seconds*、*Minutes* 和 *Hours*——发生变化时，这些显示变量各自单独更新。

与计时进程不同，设置进程(图 1.2)描述如何将计时时间或闹醒时间设置成需要的任意时间。这里使用两个附加的内部变量，*Mwakeup* 和 *Hwakeup*，来存储闹醒时间的分和时的值。通过闭合开关 *S₃* 和 *S₄*，我们可以分别增大 *Mwakeup* 和 *Hwakeup* 所存储的闹醒时间的分和时。这样，通过先闭合开关 *S₁* 或 *S₂*，当然这取决于是要设定计时时间还是闹醒时间，然后反复按开关 *S₃* 和 *S₄*，我们就可以设置合适的时间。注意，不可以同时闭合 *S₁* 和 *S₂*。还要注意，通过闭合 *S₃* 或 *S₄* 设定的时间，将立即显示在闹钟的显示器上。

最后一个进程，闹铃进程(图 1.3)的设计为：通过在计时时间等于闹醒时间的时刻，将输出变量 *Buzz* 设置为 1，将闹铃设置为“开”，这里假设闹铃开关 *S₅* 是闭合的。换句话说，每当 *Minutes*=*Mwakeup*，*Hours*=*Hwakeup*，且 *S₅* 闭合时，*Buzz* 的值为 1。

因为行为描述并不规定明确的结构，所以，对于上面介绍的闹钟的行为描述，可能有许多不同的结构描述。相反，它规定一些内部变量用来存储数据，和一些赋值语句用来进行数据转换。利用给定的这些信息，我们将变量转换成记忆元件，并推测算术元件，执行赋值语句所指定的数据转换。该转换将产生最显而易见的结构描述，但并非是最优的结构描述，如图 1.4 所示。