
HZ BOOKS
电子工程丛书
(原书第2版)
VHDL设计
表示和综合
VHDL Design
Representation and Synthesis
(Second Edition)
(美) James R. Armstrong F. Gail Gray 著
李宗伯 王蓉晖 等译
机械工业出版社
China Machine Press
PH PTR
VHDL Design
Representation and Synthesis (Second Edition)

电子工程丛书

VHDL 设计

表示和综合

（原书第2版）

（美） James R. Armstrong
F. Gail Gray 著

李宗伯 王蓉晖 王 蕾 等译

机械工业出版社
China Machine Press

VHDL语言是一种主流的硬件描述语言。本书既不同于一般的VHDL语言教材，也区别于传统的关于逻辑设计的书籍，它把VHDL语言的介绍融合到了不同抽象层次的设计中，全面深入地讲述了从原始的高层模型到门级实现的各层次的设计技术，并辅以典型实例，既能使读者对有关数字系统设计的知识有一个全面了解，同时又能较好地掌握VHDL语言及其在不同设计层次中的应用方法，掌握基于VHDL的设计技术。

本书比较注重设计方法和基本概念的介绍，深入浅出，每章还配备了大量针对性很强的习题，非常适合作为电子工程、计算机等专业VLSI设计相关课程的本科生和研究生教材，也可供相关的工程技术人员参考。对于打算自学这方面内容的人来说，本书也是一本不可多得的好书。

本书版权登记号：图字：01-2000-1164

图书在版编目（CIP）数据

VHDL设计：表示和综合（原书第2版）/（美）阿姆斯特朗（Armstrong, J. R.），（美）格雷（Gray, F. G.）著；李宗伯等译. -北京：机械工业出版社，2002.4

（电子工程丛书）

书名原文：VHDL Design: Representation and Synthesis, Second Edition

ISBN 7-111-09539-1

Ⅰ.V… Ⅱ.①阿… ②格… ③李… Ⅲ.硬件描述语言，VHDL-程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2001）第081804号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：程代伟 姚 蕾

北京第二外国语学院印刷厂印刷 · 新华书店北京发行所发行

2002年5月第1版第1次印刷

787mm × 1092mm1/16 · 33印张

印数：0 001-4 000册

定价：65.00元（附光盘）

译 者 序

VHDL语言是一种主流的硬件描述语言，具有很强的描述和建模能力，能从多个层次对数字系统进行建模和描述，大大简化硬件设计任务，提高设计的可靠性。基于VHDL语言的设计方法得到了广泛的应用，VHDL语言已成为硬件描述语言的工业标准。

本书既不同于一般的VHDL语言教材，也区别于传统的关于逻辑设计的书籍，它把VHDL语言的介绍融合到了不同抽象层次的设计中，全面深入地讲述了从原始的高层模型到门级实现的各层次的设计技术，并辅以典型实例，既能使读者对有关数字系统设计知识有一个全面了解，同时又能较好掌握VHDL语言及其在不同设计层次中的应用方法，掌握基于VHDL的设计技术。

本书比较注重设计方法和基本概念的介绍，深入浅出，每章还配备了大量针对性很强的习题，非常适合作为计算机、电子、电气及控制等专业VLSI设计相关课程的本科生和研究生教材，也可供相关的工程技术人员参考。对于打算自学这方面内容的人来说，本书也是一本不可多得的好书。本书英文版已经作为本科高年级的教材使用，效果较好。

参加本书翻译工作的人员：李宗伯、王蓉晖、王蕾、刘芳、陆洪毅、宋辉、沈立、赵学秘、鲁健壮、陈虎等。限于译者水平，本书的翻译难免会有不少问题，恳请广大读者批评指正。

译 者

2002年4月

前　言

本书主要讲述如何将硬件描述语言应用到各个抽象级别的数字电路设计中。这一过程分为两个步骤：1) 用硬件描述语言建模；2) 将模型综合成ASIC逻辑电路或者FPGA。在讲述此过程时使用了VHDL，即VHSIC硬件描述语言（VHSIC Hardware Description Language）。该语言在1983年由美国国防部（DOD）发起创建，由IEEE进一步发展并在1987年作为“IEEE标准1076”发布。从那以后，该语言进一步得以改进并在1993年推出了一个更新的标准。从此，VHDL成为硬件描述语言的业界标准。从作者的观点来看，在所有的硬件描述语言中，它具有最容易为人所理解的模型结构，因此本书选择了VHDL语言作为基本语言。本教材以一种深入、统一的方式介绍了这种语言。

当前市场上大多数介绍硬件描述语言的书，特别是介绍VHDL的书有：1) 完整介绍VHDL语言的语言类丛书，它们没有介绍如何把语言结合到数字设计过程之中；2) 介绍中把VHDL模型作为一种仿真工具来验证传统方式设计结果的逻辑设计类丛书。本教材把VHDL完整地结合到数字设计过程中，这一过程从一种高层可执行模型开始，这种模型提供可执行的规格说明，直到门级实现结束。

本教材中，综合过程被分为多个步骤。先从自然语言描述转换到VHDL语言，再从VHDL语言转换到最终的电路图。本书从两个视角来讨论综合过程：1) 映射：强调VHDL语言结构和逻辑实现电路之间的对应关系，书中有完整的一章来介绍用于综合的模型的正确格式；2) 工具：介绍两个普遍使用的工具集，Synopsys Design Analyzer and Compiler（用于ASIC）和Xilinx Foundation Series（用于FPGA）。因为设计的目标是形成ASIC和FPGA，所以书中有一章用来介绍这两种技术。本书还有一章介绍从规格说明到逻辑综合这一“自顶而下”设计过程的整体方法。

本教材面向三个教育目标：1) 作为电子工程、计算机工程以及计算机科学本科生第二课程的逻辑设计教材；2) 作为硬件描述语言或其他设计方面的研究生课程教材；3) 作为实习工程师学习硬件描述语言的参考书。学习本书时要求读者已经掌握：1) 计算机组成和逻辑设计的基本课程；2) 高级语言，如C、C++或JAVA的某些知识。

作者在逻辑设计课程序列的第二门课程中使用了本教材。学生包括计算机工程专业的二、三年级学生（课程为必修课）和电子工程专业三、四年级的学生（课程为选修课）。在一个学期的课程里讲述了第1、2、3、4、5、9、10和11章的内容，主要强调如何产生可综合的保守算法风格的VHDL模型。实验环境包括Viewlogic公司的PC版Workview，来进行VHDL建模、仿真和原理图获取。FPGA的综合使用的是Xilinx软件和XS40/XTEND硬件板，还使用了Elanix公司的System View进行数字滤波器的高层设计。采用工作站版的Synopsys工具进行ASIC综合。因我系的所有学生都有自己的PC机，所以使用如Workview这样基于PC的系统，能够为我们在第二门数字设计课程中通常要教授的大量学生提供有效的支持。同样，我们通

过telnet和dc_shell脚本来进行Synopsys综合。指定的典型作业包括：

1) 引导学生熟悉Workview的VHDL建模、仿真和原理图获取环境。

2) 开发并仿真一个单个的VHDL行为模型。

3) 设计计数器或类似电路的模型。开发出计数器触发电路和门电路的VHDL行为模型，并用Workview的原理图获取工具构造计数器的结构模型。

4) 把系统描述翻译为VHDL行为模型并进行模拟。它通常是一个状态机模型，如接口协议、售货机或交通灯控制器等。

5) 引导学生使用Xilinx Foundation Series。

6) 用Synopsys ASIC逻辑电路和FPGA电路这两种形式实现一个小电路并比较运行速度。

7) 一个相当复杂的FPGA项目，如Booth乘法器、计算器、小型处理器、数字滤波器或图形显示。对于滤波器，XTEND主板的多媒体数字信号编解码器被用于A/D和D/A转换。Xilinx滤波器代码由System View产生。图形显示实现对VGA监视器上RAM中的存储模式的显示。

如果针对研究生课程，本教材可以在一个学期完成。课程可以包括语言结构及仿真和综合的语义细节检查。在弗吉尼亚工学院的研究生课程中，使用Synopsys进行综合并验证综合后的模型。课程研究如何控制综合以获得延时和面积优化的电路，也包括了如Express VHDL、SPW和System View等高层建模工具的使用，还进行了VHDL和Verilog的比较。

在此课程里，学生的实验任务包括：

1) 产生并仿真一个简单的VHDL行为模型。

2) 产生计数器或类似电路的模型。产生计数器触发电路和门电路的VHDL行为模型，然后产生整个系统的VHDL结构模型。

3) 使用复杂数据类型，如使用数组聚合和记录类型来实现有限状态机的列表表示。

4) 使用总线判决和总线协议的系统建模。该系统使用IEEE 9值逻辑系统。例子包括第6章中的URISC处理器系统或图像处理的直方图构造系统。

5) 用VHDL和Verilog编写、仿真、综合一个模型，并比较结果。

6) 学生选择一个系统建模作为学期项目。可以选择扩展语言的项目，例如一些非典型的应用，如为并行处理系统建模或者为非数字系统建模。

本书还包括几百个VHDL模型和代码段。所有的代码都用Synopsys VHDL系统做过分析、仿真与综合（除了VHDL 93的代码）。此外，还有超过300道不同难度的课后思考题。这些习题有简答题、简单设计题，包括设计、建模和仿真的复杂系统设计问题，以及需要研究设计或设计工具的问题。有些习题可以成为非常好的论文设计项目。

本书附有一张CD-ROM。CD中的内容有：1) 书中VHDL代码的源文件；2) 支持数据命令文件的项目集合；3) 支持公共设计范例的软件包。

写作这样一本教材是一项比较重的任务。写作过程中，我们得到了很多个人和组织的帮助与支持，在此要感谢：

1) Viewlogic公司、Synopsys公司和Xilinx公司提供了书中的VHDL代码以及在硬件描述语言课程中所使用的VHDL软件。

2) Intermetrics公司的Dave Barton负责审阅本书草稿。

3) Prentice Hall出版公司的产品编辑Vincent Janoski。

4) Prentice Hall出版公司的图书编辑Bernard Goodwin对这一项目的大力支持。

也非常感谢我们的妻子Marie和Caryl，感谢她们对我们在计算机前长时间工作的鼓励和支持，以及在过去两年的旅行和休假中忍受我们与笔记本电脑相伴。

作者简介

James R.Armstrong博士　美国弗吉尼亚工学院电子和计算机工程系教授，讲授本科生和研究生的计算机体系结构、硬件描述语言及逻辑设计课程。他是原IEEE标准化委员会的成员。著有《VHDL芯片级建模》（*Chip Level Modeling With VHDL*），以及与他人合著《VHDL结构化逻辑设计》（*Structured Logic Design With VHDL*），均由Prentice Hall公司出版。他在硬件描述语言的应用方面做了深入的研究。他的论文发表在多种IEEE杂志上，并在一些国际学术研讨会上宣读。

F. Gail Gray博士　美国弗吉尼亚工学院电子和计算机工程系教授，讲授本科生和研究生的计算机工程、逻辑设计、硬件描述语言、编码理论、容错计算、测试及微处理器系统设计课程。他的研究成果曾在“*IEEE Transactions on Computer*”、“*Journal of VLSI Signal Processing for Signal, Image, and Video Technology*”、“*Design Automation Conference*”、“*VHDL International Users Forum*”及其他核心期刊和国际研讨会上发表。

目　录

第1章　结构化设计概念

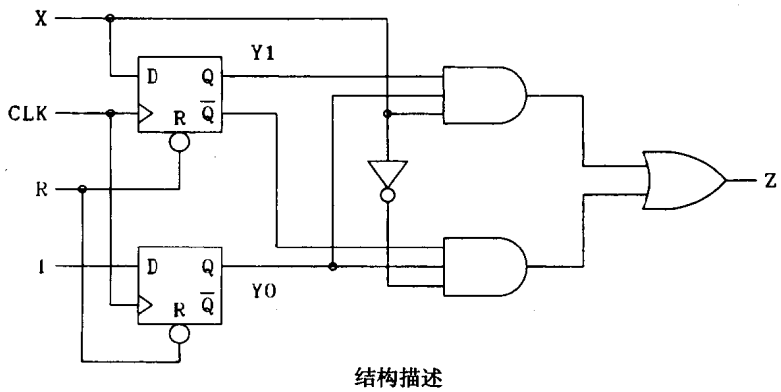
这一章给出与设计过程相关的基本定义。很有必要现在就来介绍它们，这对于解释其他概念是必需的。读者需要仔细研究这些定义才能理解后面将要介绍的内容。在后面的学习中再回过头来看看这一章也是很有益处的，因为这些定义的完整含义只有通过使用和实例才会变得更加明确。

1.1　抽象层次

本节介绍数字电路设计者所使用的抽象层次概念。抽象层次可以在以下两个域中表示：

- **结构域**：在结构域中，一个部件通过一些基本部件的互连来描述。
- **行为域**：在行为域中，一个部件通过定义它的输入/输出响应来描述。

图1-1给出了检测输入X的连续两个或更多1（或0）的逻辑电路的结构描述和行为描述。结构描述是指门和触发器等基本部件的互连，行为描述则是使用硬件描述语言（HDL）做文字性的表述。



结构描述

```
entity TWO_CONSECUTIVE is
  port(CLK,R,X: in BIT;Z: out BIT);
end TWO_CONSECUTIVE;

architecture DATAFLOW of TWO_CONSECUTIVE is
  signal Y1,Y0: BIT;
begin
  STATE: block((CLK = '1'and not CLK'STABLE) or R = '0')
  begin
    Y1 <= guarded '0' when R = '0' else X;
    Y0 <= guarded '0' when R = '0' else '1';
  end block STATE;
  Z <= Y0 and ((not Y1 and not X) or (Y1 and X));
end DATAFLOW;
```

行为描述

图1-1　一个采样电路的结构描述和行为描述

抽象层次定义如下：

抽象层次：一系列相关的表示层次，允许以不同的细节程度来描述一个系统。

一个典型的抽象层次如图1-2所示，层次中的第i层可以转换成第i+1层，细节程度一般随层次的下移而单调增加。

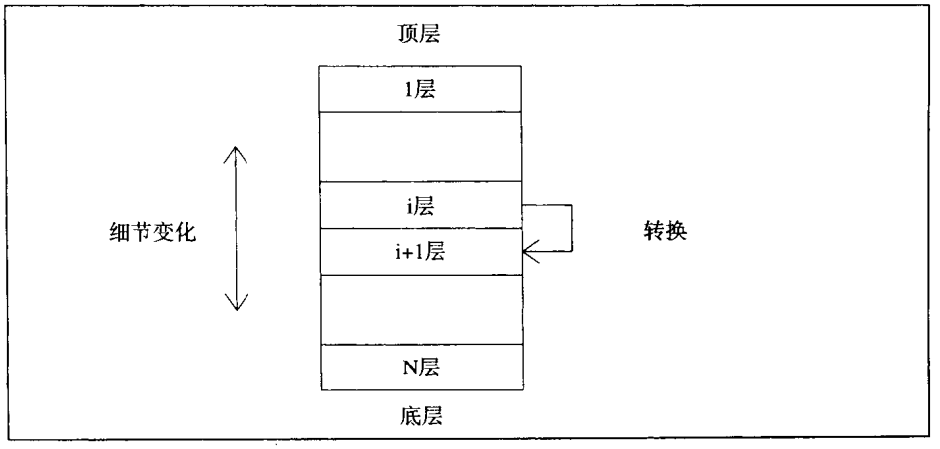


图1-2　抽象层次

本书使用的设计层次有6级：硅片级、电路级、门级、寄存器级、芯片级和系统级。表1-1举例说明了这种分层结构。硅片级是最低的层次，而系统级是最高的层次。一个设计可以用任何层次来表示。当设计从上而下进行时，该设计就逐步接近物理实现，在表示上就更少一些抽象。因此，表示一个设计所需的细节会随着它在层次中的下降而增加。在本书中，我们总是强调：在一个具体层次的设计中，保证它具有充分而不过多的细节是非常重要的。细节不充分会造成不精确的结果；相反，过多的细节则会使该层次的设计活动过多。

表1-1　设计的抽象层次

细节级别	行为域表示	结构域基本部件
系统	性能规格说明（自然语言，如英语）	计算机/磁盘/部件/雷达
芯片	算法	微处理器/RAM/ROM/串行端口/并行端口
寄存器	数据流	寄存器/ALU/计数器/多路复用器/ROM
门	布尔方程	与/或/异或/触发器
电路	微分方程	晶体管/电阻/电感/电容
版图/硅片	电子和空穴迁移方程	几何形状

表1-1显示了用结构化的基本部件及行为描述来表示每一层的这种层次结构的性质。结构化的基本部件通过互连可以在给定层次上形成结构化的模型。图1-3是各个层次上结构化的基本部件的例子，其行为表示则是在该层次上对设备的I/O响应的文本或图形描述。

在最低层，即硅片级，基本的单元是代表扩散区、多晶硅和硅片表面金属层的几何形状。这些图案的互连对设计者而言是对制造过程建立模型，这一级的行为描述则是描述在电子材料中电子和空穴迁移的物理公式。

在第2层，即电路级，其表示则是传统无源和有源电子电路元件的互连，这些电子电路元件包括电阻、电容、二极管和MOS晶体管等。元件的互连可以被用于以电压和电流的形式模

拟电路的行为，行为方面则通过微分方程的形式来表达。

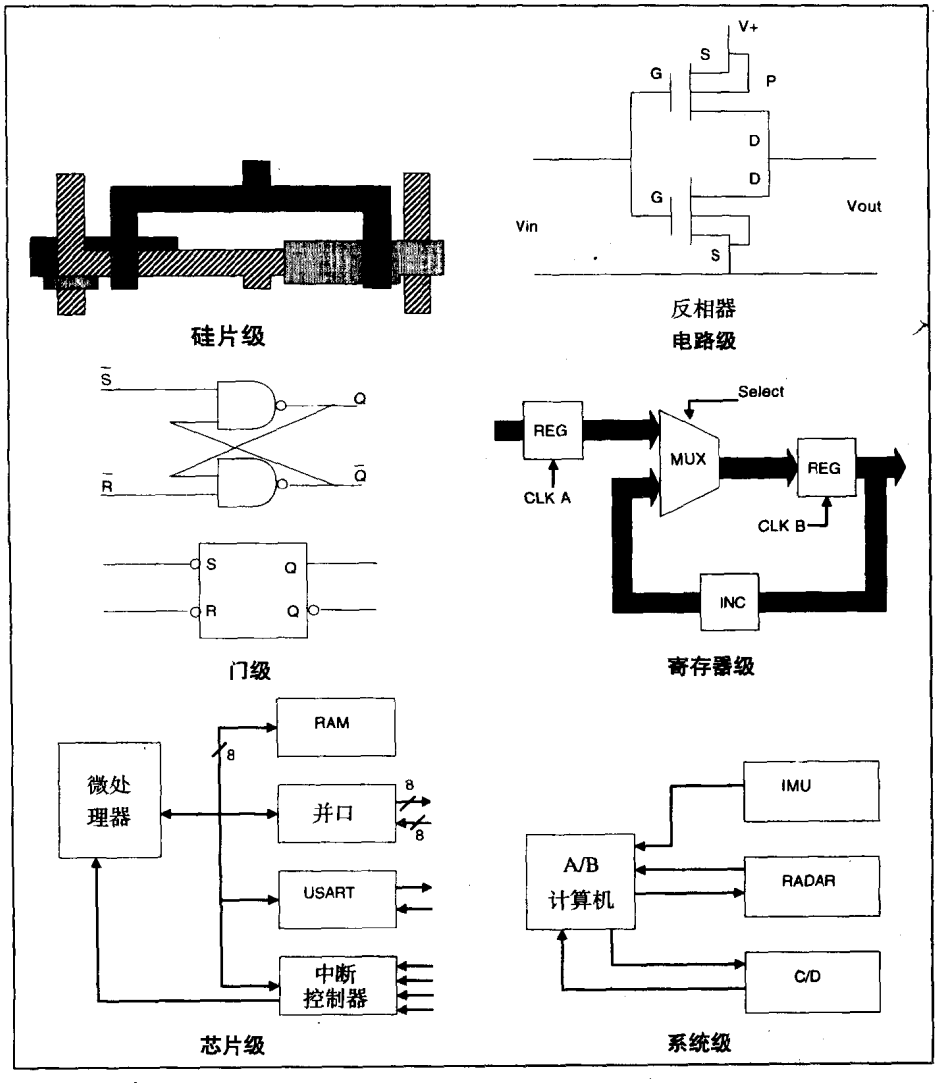


图1-3 结构域基本部件示例

第3层门级是传统数字器件的主要设计层次，其基本部件包括与/或/异或/反相操作门和不同类型的触发器。这些单元的互连构成组合和时序逻辑电路，布尔方程式定义了这一级的行为特征。

门级层次之上是寄存器级，这里的基本部件是寄存器、计数器、多路复用器和ALU。这些部件有时称为功能块，它们也对应着VLSI设计的宏，因此，寄存器级也称为功能级或宏级。虽然寄存器级的基本部件也可以用互连的门来表示，但是在进行这一层的设计时一般并不采用这种方法。寄存器级的基本部件采用真值表和状态表来表示，因此这两种形式可以用来进行这一层次的行为描述，并称之为*数据流*，也就是说它们反映在实际实现中数据的实际分布方式。在本书中，将会讲解这些数据流描述是如何在硬件描述语言（HDL）中实现的。

寄存器级之上是芯片级。在这一层次中，基本部件是微处理器、存储器、串口、并口和

中断控制器等部件。虽然芯片的边界一般就是模型的边界，但其他方案也是可能的。例如，组成一个功能部件的芯片集也可以作为单一的实体建模，可以在设计过程中把芯片的各部分建模为分离的实体。这一级的关键点在于建立一个大的逻辑模块，该模块对较长且集中的从输入到输出的数据路径进行建模。在行为域，层次结构每一级上的基本部件不是用其他更基本的部件构造的结构模型，而是用行为来描述。每个基本部件都是一个确定的模型实体。因此，如果一个串行I/O端口（UART）需要建模，该模型不是通过寄存器、计数器等更简单的功能模型连接而成——UART本身就是一个基本的模型实体。对于从其他制造商处购买芯片而不了解其内部门级结构的系统制造公司来说，行为域模型非常重要。复杂电路的芯片级模型被当作知识产权（Intellectual Property），常常由一个公司出售给另一个公司。芯片级模型的行为描述内容为该部件的I/O响应——该芯片实现的算法。在本书中，硬件描述语言（HDL）用来对这些算法的描述进行编码。

结构化层次的最高层是系统级。这一层的基本部件是计算机、总线接口部件、磁盘部件、雷达部件等。行为级的内容常常表达成性能规范，给出诸如处理器的MIPS速率、总线的带宽等，或使用统计模型来确定系统某一部分的利用率。如果在这一级使用一个确定性的模型，则它会使用较高层次的数据类型来表示系统间传递的信息。例如，如果要建立雷达系统的模型，“频率”类型的信息将在系统部件间传递。

1.2 文本表示与图形表示

设计表示可以是文字的，也可以是图形的。图1-1表示的是该电路的图形化逻辑原理图，其功能是检测两个或更多个连续出现的1或0（简称为TWO_CON）。图1-4则给出了它的框图、状态图、时序图、状态表、状态赋值和真值表。

这些都是图形化的例子。常用的文字性的表示方法是自然语言（如英语）、方程式（如布尔方程式或微分方程式）和计算机语言。在本书中，使用一种特定的计算机语言，称为*硬件描述语言*（HDL），其定义如下。

硬件描述语言：一种具有硬件建模所需特定结构的高级编程语言。

图1-1中的行为描述展示了对TWO_CON的硬件描述语言的文字性描述。

在开始一个设计过程时，一个重要的考虑是用图形表示还是用文本表示。在历史上，图形表示对于数字电路设计来说是一种更常用的表示方法，框图、时序和逻辑图（原理图）是基本的表示形式。然而，随着HDL的出现，文字性描述已经成为了主流。通过考查图1-1、图1-3、图1-4和表1-1，不难看出：结构描述基本上都是图形性的，而行为描述则是文字性的。这种分类方案的一些例外情况包括状态表、状态图和时序图，它们是图形性的，但是用来表示行为。至于应该使用图形还是文字这一基本问题，有人做了下述概括：*文字能更好地表达复杂行为，图形则更适合于阐明相互关系*。过度依赖于文字或图形都会造成观察上的损失，正如俗话说的：“只见树木，不见森林”。因此，在实际的设计系统中应权衡使用文字和图形——这也是本书所采用的方法。

1.3 行为描述的种类

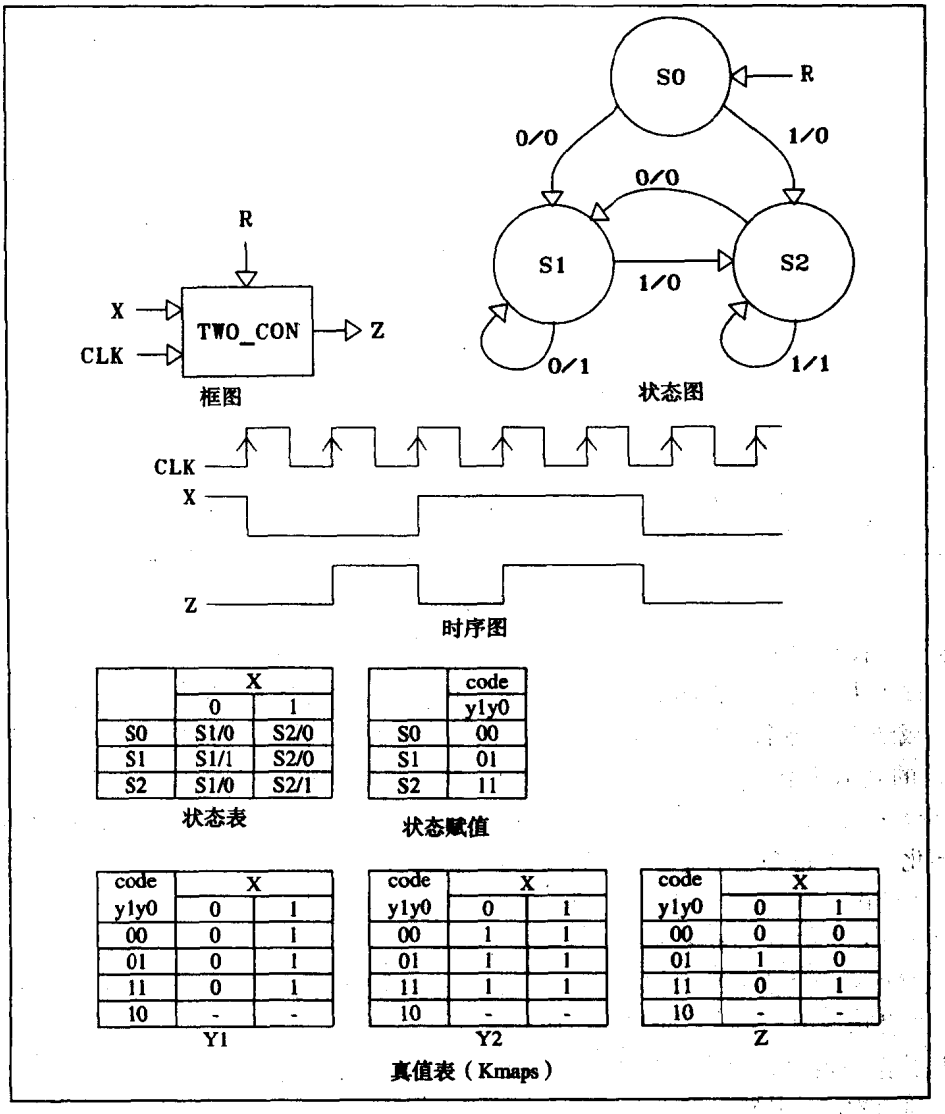
在HDL语言中的行为描述通常分为两类：*算法和数据流*。

算法：一种行为描述，其中定义I/O响应的过程没有隐含任何特定的物理实现。

因此，算法描述仅仅是一些写就的过程或程序，用于模拟器件的行为。以检验它是否执行正确的功能，而不考虑具体实现。

数据流：一种行为描述，它所描述的数据相关性与实际实现是相匹配的。

数据流描述说明数据如何在寄存器间移动。图1-1给出了TWO_CON的数据流描述，图1-5给出了该电路的算法描述。



	X	
	0	1
S0	S1/0	S2/0
S1	S1/1	S2/0
S2	S1/0	S2/1

状态表

	code
	y1y0
S0	00
S1	01
S2	11

状态赋值

code	X	
y1y0	0	1
00	0	1
01	0	1
11	0	1
10	-	-

Y1

code	X	
y1y0	0	1
00	1	1
01	1	1
11	1	1
10	-	-

Y2

code	X	
y1y0	0	1
00	0	0
01	1	0
11	0	1
10	-	-

Z

真值表（Kmaps）

图1-4 逻辑电路的图形表示

1.4 设计过程

我们将在本教材中描述一种结构化的设计过程，先从下面的一些定义开始。

设计：从系统的一种表示到另一种表示的一系列转化，直到最终的表示能被生产出来。

设计的途径涉及到综合（synthesis），它在字典上的定义为："把抽象的实体结合成单个或