

计算机硬件技术系列丛书

VHDL

电子设计硬件描述语言

VHDL

SECOND EDITION

Douglas L. Perry



希望

McGRAW-HILL
学苑出版社

计算机硬件技术系列丛书

电子设计硬件描述语言

VHDL

Douglas L. Perry 著

周祖成 译

周祖成 审校

学苑出版社

(京)新登字 151 号

内 容 提 要

VHDL——硬件描述语言是高层次电子设计的工具语言。本书按三大部分组织。第一部分共 8 章,介绍 VHDL 的特点;第二部分(九、十章)讨论 VHDL 的综合使用;第三部分以一个自动售货机控制器为实例介绍了一个用 VHDL 的 Top-Down 设计的例子。最后四个附录列举了用 IEEE-1164-STD-LOGIC 程序包等有关 VHDL 语言的标准。

需要本书的读者,请直接与北京海淀 8721 信箱书刊部联系,电话:2562329,邮政编码:100080。

版 权 声 明

本书英文版名为《VHDL》,由 McGraw-Hill 公司出版,版权归 McGraw-Hill 公司所有。本书中文版由 McGraw-Hill 公司授权出版。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

计算机硬件技术系列丛书

电子设计硬件描述语言 VHDL

著 者: Douglas L. Perry
译 者: 周祖成
审 校: 周祖成
责任编辑: 陆卫民
出版发行: 学苑出版社 邮政编码: 100036
社 址: 北京市海淀区万寿路西街 11 号
印 刷: 兰空印刷厂
开 本: 787×1092 1/16
印 张: 20.75 字 数: 472 千字
印 数: 1~5000 册
版 次: 1994 年 10 月北京第 1 版第 1 次
ISBN7-5077-0822-5/TP·20
本册定价: 26.90 元

学苑版图书印、装错误可随时退换

PREFACE

I was really honored to hear that this book was to be translated into Chinese. When I wrote the first version of this book 5 years ago my goal was to show the typical hardware design engineer how VHDL could be used to model hardware. I wanted to show how the typical designer would use the language, not just what the language contained. The examples and the style of the book are meant to be very practical and hardware designer oriented.

I have been very pleased by the responses of people all over the world who have read the book, and given me their feedback. I hope that the Chinese version is as well received as the English version has been in other parts of the world.

I had a chance to visit China a few months ago and was very impressed with the people and the culture of the country. I think that China has a very bright future ahead in the electronics industry will be very honored to think that this book can play even a small part in helping engineers in China to design with VHDL.

Many thanks to Lin Yang for helping me get the chance to visit China, and to Zucheng Zhou for the Chinese translation.

Douglas L. Perry

译者序

“VHDL”second edition 出版后被清华大学电子工程系选作研究生“电子系统仿真与 VHDL”课的教材,在“1994—1995”教学年度给学生上课的同时将它译成中文。

五年前,我得到 Douglas L·Perry 的“VHDL”(第一版)一书时,就被此书的风格所吸引。也许是在我所读过的 VHDL 书中,此书以描述实际硬件见长,所以在我对通信与电子系统专业的学生讲课时曾谈到:“学 VHDL 最好从描述硬件的角度去学,而 Douglas L·Perry 的‘VHDL’一书为我们学 VHDL 提供了一个示范。”

第二版作者增加了 VHDL 的 synthesis 的内容,作者及时地反映了电子设计从分析向综合的发展趋势,对正从 CAE 转向 ESDA 的国内读者讲,这无疑是一本入门的好书。本书还以“自动售货机”的自顶向下(Top-down)设计的例子展示了这种设计方法。

为了能让读者早日有一本中文的 VHDL 的入门书,除我抓紧译了 1—10 章之外,请毛世文同志译了第 11、12 章,我做了全书的校对。本书没有经过讲义阶段直接出书,很匆忙,也难免有不少疏漏与错误,敬请读者指正。

杨林博士(ALTa Group of Cadence System)为此书的英译中做了推荐工作,作者 Douglas 给了我这个机会,McGraw—Hill Professional Publishing Group 的 Germa Velten 女士,McGraw—Hill Int’l Enterprises Inc. (Taiwan)的洪钦镇先生都对此书中文版出版给予了支持,本人在此致谢。

周祖成
于清华大学

前 言

本书的第二版相对第一版有一些明显的不同,书中所有例子都使用了 IEEE 标准的 1164 值系统。这增强了与其它模型的兼容性。同时,本书现在还包括了对 VHDL 综合的讨论。第九、十两章讨论了基本的综合过程并且给出了一些综合的例子。对第一版中的自动售货机控制器设计的例子作了相应的修改,以便使它能被综合。上一版由于使用文本处理不当还存在着很多文法错误,在第二版中这些错误得到了更正,并且插图也得到了很大的改善。

本书旨在帮助硬件设计工程师们学习如何更好地为他们的设计建模。从最初的设计规范到门级的设计实现,对硬件设计每个层次 VHDL 描述的编写,它都将加以指导。本书还试图帮助那些对 VHDL 知之甚少或一无所知的设计者达到编写复杂的 VHDL 描述的水平。它并不希望展示每种可能用的不同的 VHDL 结构,而是力图告诉设计者如何简洁、高效并且正确地写硬件设计的 VHDL 描述。

本书按三个合理部分组织:第一部分介绍 VHDL 的特点;第二部分讨论了 VHDL 综合过程;第三部分以自动售货机控制器为例介绍从最初的设计规范到最终可综合设计实现的过程。书后附有大量的附录,包括了关于语言本身和本书中使用的所有例子的有用资料。第一部分中,每次介绍一个或几个 VHDL 的特点。介绍每个特点的时候,将给出一个或几个实例来说明该特点是如何应用的。第一部分包括第一至第八章,第一章介绍 VHDL 基本的描述能力,它讨论了 VHDL 设计与其它典型的计算机辅助工程(CAE)设计方法学的对照关系,并且介绍了该语言的基本术语。第二章描述了某些 VHDL 的基本概念,它们包括可供选用的不同的延时机理、如何使用具体元件的特定数据并谈及关于 VHDL 驱动器。第二章讨论了并行语句,而 VHDL 顺序语句将在第三章中介绍。第四章介绍了 VHDL 可采用的宽范围的类型。针对每一个类型给出了一些例子,并示出它们在实际应用中是如何使用的。第五章引入子程序和程序包的概念。这里不但给出了函数的使用,而且还介绍了 VHDL 程序包的特点。

第六章介绍了 VHDL 可使用的五种类别的预定义属性。针对每个类别还给出了例子,例举了如何使用特定的属性来最好地满足设计者的需要。还有一些例子说明了每一种属性的意图。第七章讨论对某种设计的结构如何配置,讨论了每一种类型的配置,同时还有一些例子加以示范。第八章向读者介绍 VHDL 的一些更高级的特性。这一章讨论了重载、用户定义的属性、生成语句以及 TextIO。所有这些概念是该语言更高级的特性,当设计者对 VHDL 了解更多的时候将使用到它们。

本书的第二部分包括第九和第十章。这两章介绍了基本的综合过程。类似如何编写可综合的 VHDL、什么是库、什么是综合的过程、什么是约束条件和属性以及什么是优化过程这样的问题都将被讨论到。

本书的第三部分通过一个自动售货机控制器的描述,讨论了从行为级到门级综合输出的全过程。第十一章在行为级上描述了一个自动售货机。第十二章将自动售货机分解为三个可综合的元件并且讨论了综合的结果。

最后,书后有四个附录以供参考。附录 A 列出了本书所使用的 IEEE—1164—STD—LOGIC 程序包。附录 B 给出了第十二章中自动售货机综合过程得到的门级网表。附录 C 是书中余下内容精选而成的一套很有用的速查表。最后,附录 D 讨论了如何阅读 BNF。希望读者在阅读这本书和使用 VHDL 的时候能得到和我编写它一样多的乐趣。

致 谢

如果没有很多人的帮助,这本书的编写是不可能的,在此我向他们表示感谢。Rod Farrow, Cary Ussery, Alcc Stanculescu 和 Ken Scott 解答了某些关于 VHDL 难以预测的行为难题。Ken Scott 和 Kjell Nielsen 审阅了本书的第一稿, Dierdre Hanford 审阅了第二稿中的综合部分。他们的评论很有见地并且很有帮助。Rick Herick 提议使用一个自动售货机作为教学的例子。这是一个很容易理解的例子,并且对展示该语言的一些有趣的特点也足够复杂。Paul Krol 推导了第七章中描述类属的图表。Keith Irwin 帮我确定了一些章节的风格。Brent Gregory 和 Russ Segal 在我理解 VHDL 综合上给予了很大的帮助。我还要感谢 Vantage Analysis Systems。Synopsys 和 Redwood Design Automation 的所有小组成员,他们提供了工具,使验证本书中的概念成为可能。我还要感谢 Redwood Design Automation 的管理部门,尽管他们的公司非常年青,但他们仍对这一努力给予了支持。

Douglas L. Perry

目 录

第一章 VHDL 引论	1
1.1 VHSIC 计划	1
1.2 VHDL 作为一个标准	1
1.3 学 VHDL	1
1.4 VHDL 术语	2
1.5 传统设计方法	2
1.6 传统的电原理图	3
1.7 ENTITLES 与符号对应	4
1.8 Architectures 与电原理图对应	4
1.9 元件具体安装语句	5
1.10 行为级描述	5
1.11 并行信号赋值语句	6
1.12 事件处理	6
1.13 语句的并行性	7
1.14 顺序行为	7
1.15 进程语句	8
1.16 顺序语句	8
1.17 结构体选择	8
1.18 配置语句	9
第二章 行为建模	11
2.1 行为建模引论	11
2.2 传输延时和固有延时的对比	14
2.3 仿真 Δ	16
2.4 驱动	18
2.5 类属(generics)	20
2.6 块语句	22
第三章 顺序进程	27
3.1 进程语句	27
3.2 信号赋值和变量赋值的对比	29
3.3 顺序语句	32
3.4 被动进程	49
第四章 类型和属性	52
4.1 对象的类型	52
4.2 数据类型	55

4.3	标量类型	56
4.4	复合数据类型	62
4.5	寻址类型	70
4.6	文件类型	75
4.7	子类型	77
第五章	子程序包和程序包	80
5.1	子程序	80
5.2	过程	99
5.3	程序包	101
第六章	预定义属性	106
6.1	值类属性	106
6.2	函数类属性	111
6.3	信号类属性	119
6.4	类型类的属性	126
6.5	范围类属性	126
第七章	配置	128
7.1	默认配置(配置指定)	128
7.2	元件配置	130
7.3	映射库实体	136
7.4	配置中的类属	140
7.5	在结构体中类属值的详细说明	142
7.6	在配置中的类属详细说明	144
7.7	板—插座—芯片的仿真	149
7.8	块配置	152
7.9	结构体配置	154
第八章	先进的内容	158
8.1	重载	158
8.2	别名	166
8.3	限定表达式	166
8.4	用户定义属性	168
8.5	生成语句	170
8.6	文本输入/输出(TextIO)程序包	174
第九章	综合	179
9.1	RTL级描述	179
9.2	约束	183
9.3	属性	184
9.4	工艺库	187
9.5	综合	188
第十章	VHDL综合	194

第十一章 顶层系统设计	213
11.1 自动售货机控制器.....	213
11.2 顶层程序包.....	213
11.3 顶层实体.....	215
11.4 顶层结构体.....	217
11.5 自动售货机的配置.....	223
第十二章 自动售货机: 第一步分解	224
12.1 硬币处理器.....	224
12.2 商品处理器.....	225
12.3 找钱器.....	226
12.4 结构型结构体.....	227
12.5 硬币处理器.....	230
12.6 商品处理器.....	234
12.7 找钱器.....	241
12.8 下层配置.....	245
12.9 群配置.....	245
12.10 小组的配置	246
附录 A 标准逻辑程序包	249
附录 B 自动售货机网表	268
附录 C VHDL 参考表	306
附录 D 阅读 VHDL BNF	313
英汉名词对照表(按字母顺序)	315

第一章 VHDL 引论

VHDL (VHSIC Hardware Description Language) 是用来描述从抽象到具体级别硬件的工业标准语言, VHDL 正迅速地接纳为一种通用的设计交换媒介。计算机辅助工程工作站制造厂家的整个业界正在把 VHDL 作为它们工具输入与输出的标准。这些工具包括仿真工具、综合工具和布图工具等。

本章我们将探讨 VHDL 的基础知识, 介绍 VHDL 的历史, 然后将定义一些 VHDL 的基本术语, 最后将把 VHDL 和传统设计进行对照比较。

1.1 VHSIC 计划

VHDL 是由美国国防部在 70 年代末和 80 年代初提出的 VHSIC (Very High Speed Integrated Circuit) 计划的产物, VHSIC 计划的目标是为下一代集成电路的生产, 计划开发这类非常复杂的 IC, 参加者推出在 IC 设计与制造的每个阶段应达到的工艺极限, 并且工艺方面的目标已非常好地实现了, 但设计者还发现当时的工具很难完成大型设计的任务。设计者现成能用的工具差不多是门级基础的, 建立 10 万门级设计, 采用门级工具将是一项极富挑战性的任务, 因此需制订一项新的描述方法。

1.2 VHDL 作为一个标准

在 1981 年提出了一种新的 HDL, 称之为 VHSIC Hardware Description Language 或者像现在我们熟知的 VHDL, 新语言的目标有两方面: 首先是设计者想用这种语言描述他们试图描述的复杂电路, 其次他们希望这种语言成为一种标准, 使之在 VHSIC 计划中各种成员能按标准的格式向其他的成员提供设计, 并规定任何子合同商向他们的主要合同洽谈时, 都要用这种单一标准格式。

在 1986 年 VHDL 被建议作为 IEEE 标准, 经过了多数个再版和更改之后, 直到 1987 年 12 月它才被接纳作为 IEEE 1076 标准, IEEE 1076-1987 标准的 VHDL 是本书中采用的 VHDL, 所有的例子均按 IEEE1076 VHDL 描述, 并采用 Vantage Analysis System 的 VHDL 仿真环境描述、编辑和仿真, 本书综合的例子用 synopsys 综合工具作综合。

1.3 学 VHDL

如果整本地读 VHDL 语言参考手册 (Language Reference Manual_LRM) 来学 VHDL, 可能会认为它是非常难学的语言, 语言参考手册为 VHDL 的实现者描述 VHDL, 它决不能作为 VHDL 的用户指南。VHDL 是一种大型语言, 因而学会有关它的全部可能是个非常大的任务, 学会初步地写一个有用的模块这并不需要掌握整个语言, 可以初步掌握语言的一些主要内容,

而当需要写更复杂的模块时,再学与用一些更为复杂的特点。

VHDL 包含各种层次的表达方式,从双向开关级到系统级,及其中间的各种层次的表达方式,入门 VHDL 最好的方法是学会试做一些小型设计够用的语言,当你对这些 VHDL 的内容变得足够熟悉,您感到写一个 VHDL 已很轻松时,再转入学习一些新的语言特征并且试着做更新的设计。

1.4 VHDL 术语

在进一步做事之前,让我们先定义将在整本书中要用的一些术语,有些基本的 VHDL 构件几乎要用于每一种描述,这意味着所作的工作是不同于一般水平的设计,因此在 VHDL 中要重新定义一些有关的术语。

实体(entity):表达所有设计均与实体有关,实体是设计中最基本的模块,设计的最顶层是顶级实体,如果设计分层次那么将有一个较低级别的描述含在顶级描述之中,这类较低级别描述是含在顶层实体描述中的较低级实体。

结构体(architecture):所有能被仿真的实体都有一个结构体描述,结构体描述实体的行为功能,一个实体可以有多个结构体,一种结构体可能为行为描述,而另一种结构体可能为设计的结构描述。

配置(configuration):用配置语句安装连接具体元件到一个实体——结构体对,配置被看作是设计的另件清单,它描述对每个实体用哪一种行为,所以它非常像一个描述设计每部分用哪一种零件的清单。

程序包(package):它是设计中用的子程序和公用数据类型的集合,可以认为程序包是构造设计工具的工具箱。

母线(bus):母线一词通常被看作引出的一组信号,或是在硬件设计中用的一种特殊通信方式,在 VHDL 中母线是被它的驱动源关闭的特殊信号。

驱动(Driver):它是有关信号的源,如果信号由两个三态反相器所驱动,当两个反相器起作用时信号将有两个驱动源。

属性(attribute):属性是附到 VHDL 对象上的数据或者是有关 VHDL 对象的预定义数据,例如缓冲器的电流驱动能力或器件最大工作温度之类的数据。

类属(generic):VHDL 中它是传递信息参数到实体的术语,例如,如果实体是带上升和下降延时的门级模块,上升与下降延时值将由类属传给实体。

进程(process):在 VHDL 中进程是基本的执行单元,在 VHDL 描述仿真时,将把所有的运算都划分为单个或多个进程。

1.5 传统设计方法

今天,当设计师开发一个新的硬件时,它或许在 CAE 工作站上做设计,为了在典型的 CAE 工作站做设计,设计者将为设计画一张线路图(HDL 为基础的设计方法将在后面介绍)。

通常由线(信号)和表示基本设计单元的符号连在一起组成线路图,符号取自设计者用于构造线路图的另件库。可利用的符号类型与设计者正在建立的设计类型有关,如果设计者正在

建立用标准逻辑零件的板级设计线路图,那么在线路图中将示出设计者现成可用的标准零件符号,如果设计者正为 ASIC(专用集成电路)建一个线路图,那么可用的符号是有关这种类型的 ASIC 库可用的专用宏单元。

符号由信号(或网线)连在一起,信号使符号互连并建立设计所需的特定连接,这样从互连得出一种网表,设计实现之前网表建立验证设计的仿真模型,一旦设计已经被验证,为了由实际的设计网表向布线软件包提供所需的信息,布线软件将建立实际的连接数据,或者是为了建立 PCB 板所需的连线信息,或者是建立为 ASIC 所需的层信息。

图 1.1 示出在线路图中所用符号的例子,它是 R-S 触发器的符号,符号为设计者描述了下面几部分信息:

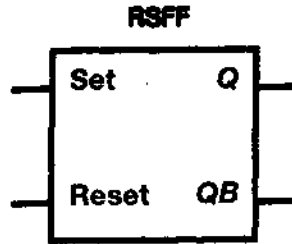


图 1.1 RSFF 符号

器件输入脚的数目:此例中是 2 个(置位(Set)与复位(Reset))。

器件输出脚的数目:此例中输出脚数是 2 个 Q 和 QB)。

器件的功能:此例由符号名描述器件的功能。在简单门级情况下,由符号的形状就可以描述符号的功能。

符号规定了对设计者的接口和功能,当放符号到线路图图纸上并由信号连在一起时就形成设计的线路图,对 RS 触发器的简单线路图的例子示于图 1.2。

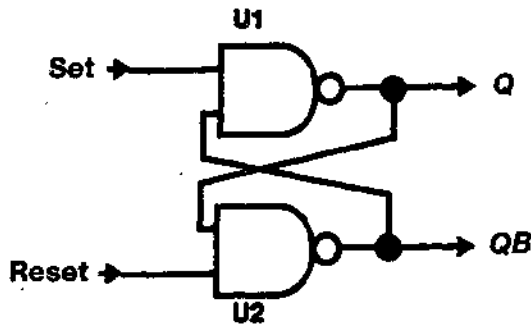


图 1.2 RSFF 线路图

1.6 传统的电原理图

这个线路图含两个 NAND2 符号的具体元件和四个端口的具体元件,四根网线把符号和端口连接在一起形成 RS 触发器,每个端口有唯一的名字,该名字还规定了连接它的信号名字,每个符号的具体元件有唯一的标识号(U1,U2),具体元件标识符将作为一个具体元件的唯一参考标识。

当编辑的这个线路加到一典型的门级仿真器时,它起一个 RS 触发器的作用,在复位端上加'0'电平将产生一个为'0'值的复位信号,这将引起在 U2(nand2) 的 QB 信号上输出一个'1'值,且与 nand2 的另一输入端的值无关,QB 上的'1'值馈送到 U1(nand2)的一个输入端,如果置位输入不起作用(空脚为 1 值),那么 U1(nand2)将有两个'1'值作输入,就引起在 Q 输出上输出'0'值,RS 触发器就完成了复位。让我们看同样的设计在 VHDL 将如何做,首先在 VHDL 中将如何表示符号呢? 在 VHDL 中将看到什么样的原理图呢?

1.7 ENTITIES 与符号对应

由实体建立所有的设计,在 VHDL 中实体直接与传统的 CAE 工作站设计方法中的符号相对应,让我们看前面描述的在顶层 rsff 元件符号的实体(entity),对 rsff 实体写一段 VHDL 描述,即:

```
ENTITY rsff IS
  PORT ( set, reset : IN BIT;
        q, qb : BUFFER BIT);
END rsff;
```

关键词 ENTITY 是实体语句的开头,本书通篇示出这种描述形式,即由 STANDARD 程序包提供该语言的关键词与数据类型,它们总是全部用大写表示,例如前例中的关键词 ENTITY、IS、PORT、IN、BUFFER 和 BIT 等,提供的标准数据类型是 BIT。而用户建立的一些对象,如前例的 rsff 等用斜体。实体名 rsff 是早就描述的符号名,在 ENTITY 语句的 PORT 付语中有四个端口,两个端口是 IN 模式,两个端口是缓冲 BUFFER 模式,用 BUFFER 模式端口代替 OUT 模式端口的原因将在稍后些讲述,两个输入端口与 CAE 工作站有关符号的输入脚直接相对应,两个缓冲端口与符号的输出端口直接相对应,所有的端口都有一个 BIT 类型。

实体描述对外界的接口,它规定端口数目、端口的方向和端口的类型,还有许多比这儿示出的更进一步的信息也能放到实体中,但这儿给出了有关我们在后面建立实体的一些基础。

1.8 Architectures 与电原理图对应

在 VHDL 中也有 rsff 元件线路图的相对应部分,它叫结构体,结构体总是对应实体而言并描述那个实体的行为。对原来描述的 rsff 器件的结构体如下所示:

```
ARCHITECTURE netlist OF rsff IS
  COMPONENT nand2
    PORT ( a, b : IN BIT;
          c : OUT BIT);
  END COMPONENT;
BEGIN
  U1: nand2
    PORT MAP (set, qb, q);
  U2: nand2
```

```

PORT MAP (reset, q, qb);
END netlist;

```

在关键字 ARCHITECTURE 标明的语句中描述实体的结构,结构体名字是 netlist,正在描述的结构体叫 rsff 实体的结构体。实体与结构体之间的关系是实体可以有多种描述实体行为的功能的结构体,例如一种结构体可能是行为功能的描述,而另一个可能是像上例的结构描述。在关键词 ARCHITECTURE 和 BEGIN 之间应插入关于 COMPONENT 和局部信号的描述,下面将会介绍它们。本例在结构体中放了两个 nand2 的具体元件,编辑器要了解在结构中放的元件接口,元件说明语句将描述的信息通知给编辑器。结构体语句区由关键字 BEGIN 开头,在 BEGIN 和 END 之间所有的 netlist 语句称之为并行语句,因为它们可以并行地执行,这种概念将在后面更详细地讨论。

1.9 元件具体安装语句

上例在该语句区中有两个元件具体安装语句,每个语句在模块中建立一个具体安装的元件,本例中每个语句建立一个二输入与非门(nand2)的具体安装元件,第一个 U1 直接和图 2.1 原理图中的二输入与非门(nand2)U1 相对应,该具体元件安装方法如下,用第一个端口连到 set 信号,第二个端口连到 qb 信号,最后一个端口连到 q 信号。

如果我们再看一下二输入与非门元件的描述,我们看到第一个端口是 IN 端口(称作 a),第二个端口是 IN 端口(称作 b),最后一个是 OUT 端口(称为 c),因此,元件具体安装将连 nand2 元件的 a 端口到 set 信号,b 端口到 qb 信号和 c 端口到 q 信号,这样我们能使实际安装参数(set, qb 和 q)与相对应描述的形式参数(a, b 和 c)一致。

只要端口不指定排序次序,就还有另一种映射端口的方法,这儿示出一种用元件具体安装语句的方法:

```
U1 : nand2 port map ( a => set, b => qb, c => q );
```

这种形式称为名字关联,并直接映射端口,不必顾及有关的次序。实际上面的语句也可以写成:

```
U1 : nand2 port map ( b => qb, c => q, a => set );
```

在 netlist 结构体的第二个元件具体安装语句中,用标识 U2 建立另一个 nand2 元件的具体元件,这些具体安装元件和图 1.2 中线路图的具体装配元件直接相对应,这个模块和图 1.2 线路图一致,我们称这种类型的 VHDL 表示方式为结构模块,或结构表示方式,做这种描述的结构一定是在结构体内部要做元件具体装配,下几章我们将讨论结构型、行为型和混合结构——行为型的描述。

结构型 netlist 结构体非常类似于典型的 CAE 工作站仿真器中的网表,为描述这种功能;本书中的例子用了一些结构化零件,但行为级建模和可综合的 RTL 描述仍然是本书的主要焦点。

1.10 行为级描述

同一电路的另一种描述方法是用行为结构体,行为结构体的一个例子将在下面结构体的

举例中示出,这个结构体用并行信号赋值语句,当名字隐含时,含在模块中的语句给信号赋值,这类语句不同于在典型编程语言中的赋值语句,因为语句的执行是并行而不是串行。

```

ARCHITECTURE behave OF rsif IS
BEGIN
    q <= NOT ( qb AND set ) AFTER 2 ns
    qb <= NOT ( q NAD reset ) AFTER 2 ns
END behave

```

1.11 并行信号赋值语句

在典型的编程语言如 C 或 pascal 中,每个赋值语句按规定的次序一个接在另一个之后执行,执行的次序由源文件中语句的次序决定,在 VHDL 结构体内部没有规定赋值语句的次序,(稍后,我们会看到在进程语句中信号赋值语句是有次序的),执行的次序仅由对赋值语句敏感的信号发生的事件规定,对 behave 结构体的第一个赋值语句探讨如下:

```
q <= NOT ( qb AND set ) AFTER 2ns;
```

由<=符号作为信号赋值的标识,它完成 qb 和 set 的逻辑与求反并对信号赋值,每当或者 qb 或者 set 上有一事件发生时就执行这个语句,信号上的事件是指那个信号上值的变化,所谓信号赋值语句对某些有关信号变化敏感,即是对<=符号右边的信号是敏感的。例如这个信号赋值语句对 qb 和 set 敏感,在 behave 结构体中另一个信号赋值语句对信号 q 和 reset 敏感。

在信号赋值中 AFTER 付句用来模仿电路中的传输延时,在 qb 和 set 上的某种事件(值发生改变)可能迟 2ns 以后在信号 q 上引起一个变化。

让我们看看这些语句实际上是如何工作的,假设我们有一个稳态条件,条件为 set 和 reset 两者都是位于'1'值情况,而且 q 信号当前是处在'0'值,qb 信号将处在'1'值,因为它正好与 q 相反(除了当 set 和 reset 两者都处在'0'值情况之外)。现在当我们假设在 set 信号上有一事件发生,使它的值变为'0',当这件事发生时,将激活第一个信号赋值语句并执行之,因为发生这件事时 set 在<=右边,并且是在对第一个信号赋值语句隐含的敏感表中。

当第一个语句执时,它将从<=符号右边的信号表达式中算出一个新值赋给 q,表达式的计算将用到式中包含的所有信号的当前值。

信号表达式将计算什么?set 信号现在等于'0',既然它的值刚变化,那么 qb 信号等于'1',因为它还没改变,对 q 信号的新值是这两个值与在一起后的求补,因而将得到的'1'值赋值给信号 q。

1.12 事件处理

但对 q 信号的赋值并未立即发生,我们前而讨论的 AFTER 付句将把新值延时 2ns 赋给 q,延时新值的机制称之为事件处理,赋新值给 q 是未来的 2ns 中使信号 q 含有新值的事件处理,当事件成熟(2ns 已满足)q 信号将接受此新值。