

集成电路/计算机硬件描述语言 VHDL 高等教程

刘明业 等 编著

Advanced Course of Integrated Circuit /
Computer Hardware Description Language VHDL

```
package ellipf_type is
    subtype short is integer range 0 to 65535;
end ellipf_type;
use work.ellipf_type.all;

entity ellipf is
    port ( clk : in BIT;
          in_rdy : in BIT;
          out_req : in BIT;
          inp : in short;
          outp : out short;
          sv2 : in short;
          ...
          out_rdy : out bit);
end ellipf;

architecture ellipf of ellipf is
begin
    process
```



100

TP312VH-43

L73

集成电路/计算机硬件描述语言 VHDL 高等教程

刘明业 等编著

清华大学出版社
北 京

内 容 简 介

本教程是参照 IEEE Standards VHDL Language Reference Manual (IEEE Std 1076, 2000 年版)编写。全书共分十三章。

第 1 章设计实体和组装。实体是 VHDL 语言基本硬件的抽象,而组装则描述出各实体的配置,以构成系统的完整设计。第 2 章子程序和包,前者用于定义计算数值或展示行为算法,后者则提供一种定义资源的方法,使不同设计单元可共享这些资源。第 3 章阐明 VHDL 支持的各种类型与预定义类型。所有预定义类型都包含在附录 D 的 STANDARD 当中。第 4 章阐述类型和子类型说明、各种对象说明、别名说明、属性说明、群组和群组模板说明等。第 5 章讨论将附加信息与已说明的实体相关联的方法,包括属性规定、组装规定和拆接规定。第 6 章给出应用于各种形式名称的命名规则。第 7 章讨论适用于各种不同形式的表达式及其计算的规则。第 8 章讨论按出现顺序定义执行一个子程序或进程的算法。第 9 章讨论各自异步执行的并发语句,它们用于定义互相连接的块和进程,而这些块和进程用于共同描述出设计的行为和结构。第 10 章阐明用于定义说明范围的规则与描述文本中各个位置上的标识符可见的规则。第 11 章阐明 VHDL 语言描述的总体组织以及设计库中对描述的分析及定义。第 12 章明确说明生效的过程,称为该说明的确立。只有定义该模型的各设计层次和结构的说明项都确立了,才能进行模型的模拟。第 13 章以神经元计算机的一个完整的全机性的描述和模拟向量编制说明 VHDL 语言的设计实践。所用程序量涵盖了大量的 VHDL 语言的语法现象。

在附录中给出 VHDL 语言词法元素、语法产生式及词汇表等。为了使已经熟悉 C++ 语言的读者能尽快掌握和理解 VHDL 语言,特地在绪论中扼要讨论了 VHDL 与 C++ 的类比。

本书的目标是全面、系统和准确地讲述超高速集成电路(VHSIC)的硬件描述语言(VHDL)。为了维护 VHDL 语言标准的权威性,本书力求既保持其完整、严谨和原汁原味,同时又补充了部分例题,帮助读者理解有关的语法现象。通过阅读这些例题将有助于提高读者编写 VHDL 语言描述的能力。本书主要面向从事 VHDL 语言高级综合和模拟验证工具研究、开发的研究生、教师、科研工作人员和从事 EDA 专业的高级技术人员,也可供高等学校计算机、信息处理、自动控制、电子工程和通信技术等专业

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

集成电路/计算机硬件描述语言 VHDL 高等教程/刘明业等编著. —北京:清华大学出版社,2003
ISBN 7-302-06289-7

I. 集… II. 刘… III. 硬件描述语言, VHDL—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 007563 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 薛 慧

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 27.25 字数: 560 千字

版 次: 2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号: ISBN 7-302-06289-7/TP·4754

印 数: 0001~5000

定 价: 43.00 元

目 录

集成电路/计算机硬件描述语言 VHDL 高等教程

第 0 章 绪论(代前言)	1
0.1 VHDL 语言的产生和发展	1
0.2 VHDL 语言的特点	2
0.3 VHDL 语言与模拟	3
0.4 编译型模拟方法及 VHDL 与 C++ 的类比	4
0.5 本教程的编写说明及读者对象	5
0.5.1 语法描述.....	6
0.5.2 语义说明.....	7
第 1 章 设计实体和组装	9
1.1 实体说明	9
1.1.1 实体头	10
1.1.2 实体说明部分	13
1.1.3 实体语句部分	14
1.2 结构体.....	15
1.2.1 结构体说明部分	16
1.2.2 结构体语句部分	17
1.3 组装说明.....	17
1.3.1 块组装	18
1.3.2 元件组装	21
第 2 章 子程序和包	29
2.1 子程序说明.....	29
2.1.1 形式参数	30

2.2	子程序体	33
2.3	子程序重载	36
2.3.1	操作符重载	37
2.3.2	标记	37
2.4	分辨函数	38
2.5	包说明	44
2.6	包体	45
第3章	类型	59
3.1	标量类型	60
3.1.1	枚举类型	61
3.1.2	整数类型	62
3.1.3	物理类型	63
3.1.4	浮点类型	65
3.2	复合类型	66
3.2.1	数组类型	66
3.2.2	记录类型	71
3.3	存取类型	72
3.3.1	非完全类型说明	72
3.3.2	对象的分配和释放	74
3.4	文件类型与操作	74
3.4.1	文件操作	74
3.5	被保护类型	77
3.5.1	被保护类型说明	77
3.5.2	被保护类型体	78
第4章	说明	85
4.1	类型说明	85
4.2	子类型说明	86
4.3	对象	87
4.3.1	对象说明	88
4.3.2	接口说明	94
4.3.3	别名说明	100
4.4	属性说明	103

4.5	元件说明	104
4.6	群组模板说明	104
4.7	群组说明	105
第 5 章	规定	116
5.1	属性规定	116
5.2	组装规定	119
5.2.1	装配指示	120
5.2.2	缺省装配指示	125
5.3	拆接规定	126
第 6 章	名称	134
6.1	命名规则	134
6.2	简单名称	136
6.3	选择名称	136
6.4	下标名称	139
6.5	片名称	139
6.6	属性名称	140
第 7 章	表达式	142
7.1	表达式规则	142
7.2	操作符	143
7.2.1	逻辑操作符	144
7.2.2	关系操作符	145
7.2.3	移位操作符	146
7.2.4	加类操作符	147
7.2.5	符号操作符	150
7.2.6	乘类操作符	150
7.2.7	杂类操作符	152
7.3	操作数	152
7.3.1	字面量	152
7.3.2	聚集	153
7.3.3	函数调用	156
7.3.4	限定表达式	156

7.3.5	类型转换.....	157
7.3.6	分配符.....	158
7.4	静态表达式	159
7.4.1	局部静态初等项.....	159
7.4.2	全局静态初等项.....	160
7.5	通用表达式	162
第 8 章	顺序语句	163
8.1	等待语句	163
8.2	断言语句	166
8.3	报告语句	166
8.4	信号赋值语句	167
8.4.1	更新预计输出波形.....	169
8.5	变量赋值语句	172
8.6	过程调用语句	174
8.7	if 语句	174
8.8	case 语句	175
8.9	loop 语句	177
8.10	next 语句	178
8.11	exit 语句.....	179
8.12	return 语句.....	179
8.13	空语句(null)	180
第 9 章	并发语句	185
9.1	块语句	185
9.2	进程语句	186
9.3	并发过程调用语句	188
9.4	并发断言语句	189
9.5	并发信号赋值语句	190
9.5.1	条件信号赋值.....	192
9.5.2	选择信号赋值.....	194
9.6	元件例示语句	199
9.6.1	元件的例示.....	200
9.6.2	设计实体的例示.....	202

9.7 生成语句	205
第 10 章 范围和可见性	210
10.1 说明区	210
10.2 说明的范围	211
10.3 可见性	212
10.4 use 子句	215
10.5 重载分辨的上下文	216
第 11 章 设计单元分析	221
11.1 设计单元	221
11.2 设计库	222
11.3 上下文子句	223
11.4 分析的顺序	223
第 12 章 确立及执行	229
12.1 设计层次的确立	229
12.2 块头的确立	231
12.2.1 类属子句	231
12.2.2 类属映射部分	231
12.2.3 端口子句	231
12.2.4 端口映射部分	232
12.3 说明部分的确立	233
12.3.1 说明的确立	233
12.3.2 规定的确立	236
12.4 语句部分的确立	237
12.4.1 块语句	237
12.4.2 生成语句	237
12.4.3 元件例示语句	239
12.4.4 其他并发语句	239
12.5 动态确立	239
12.6 模型的执行	240
12.6.1 驱动源	241
12.6.2 信号值的传播	241

VI	集成电路/计算机硬件描述语言 VHDL 高等教程	
12.6.3	更新隐式信号	244
12.6.4	模拟周期中信号值的计算	246
第 13 章	设计举例：神经元计算机的描述	251
13.1	神经元计算机简介	251
13.2	NPS 的行为级描述	251
13.3	NPS 的测试	279
参考文献		286
附录 A	VHDL 语言词法元素	287
A.1	字符集	287
A.2	词法元素、分隔符以及限界符	289
A.3	标识符	290
A.3.1	基本标识符	290
A.3.2	扩展标识符	290
A.4	抽象字面量	291
A.4.1	十进制字面量	291
A.4.2	基数字面量	292
A.5	字符字面量	292
A.6	字符串字面量	293
A.7	位串字面量	293
A.8	注释	295
A.9	保留字	295
A.10	允许的字符替代者	296
附录 B	VHDL 语言语法的产生式描述	298
附录 C	VHDL 语言词汇表	318
附录 D	VHDL 预定义语言环境	340
D.1	预定义属性	340
D.2	STANDARD 包	357
D.3	TEXTIO 包	365
索引		371

第 0 章

绪论(代前言)

电子设计自动化(EDA)的关键技术之一一是要求用形式化方法来描述数字系统的硬件电路。硬件描述语言 HDL(hardware description language)在电子设计自动化中扮演着重要的角色,它是 EDA 技术研究的重点之一。作为 IEEE 标准的硬件描述语言 VHDL(VHSIC hardware description language)已在设计中得到了广泛的应用,且影响日益深远。

0.1 VHDL 语言的产生和发展

硬件描述语言 HDL 是一个划时代的进步,它的出现和发展使得自顶向下的数字系统(集成电路/计算机)设计方法成为可能,也促使新一代 EDA 工具的出现,并使得电子设计自动化方法主流从传统的自底向上的设计方法发展到自顶向下的设计方法。

硬件描述语言的发展至今已有几十年的历史,并已成功地应用到系统的模拟、验证和设计综合等各方面。到 20 世纪 80 年代后期,已出现上百种硬件描述语言,它们对设计自动化起到了促进和推动作用,但是由于它们各自针对特定的设计领域,没有统一的标准,从而使一般用户难以使用。广大用户所企盼的是一种面向设计的多层次、多领域且得到一致认同的标准的硬件描述语言。20 世纪 80 年代后期由美国国防部开发的 VHDL 语言(VHSIC hardware description language,其中的 VHSIC 为 very high speed integrated circuit 的缩写)恰好满足了上述要求,并在 1987 年 12 月由 IEEE 标准化(定为 IEEE std 1076—1987 标准,后又在 1993 年进一步修订,定为 ANSI/IEEE std 1076—1993 标准)。它的出现为电子设计自动化的发展和推广奠定了坚实的基础。

在此期间,许多地方形成了 VHDL 的用户组织,VHDL 语言的标准化使得它在国际用户中获得了广泛的理解和支持,从而又进一步促进 VHDL 的发展。随后,VHDL 进入广泛的应用时代。特别是进入 20 世纪 90 年代以后,有关 VHDL 语言的发展成为一种主流技术。VHDL 语言逐渐成为数字系统设计的主要描述语言,它在世界各国得到了广泛

应用。有文献表明,90%以上的用户已在使用或即将使用 VHDL。1995 年,中国国家技术监督局组织编撰并出版、发布的《CAD 通用技术规范》中也明确推荐采用 VHDL 作为我国电子设计自动化硬件描述语言的国家标准。

0.2 VHDL 语言的特点

VHDL 语言能够形式化地表示电路的结构与行为,支持逻辑设计中层次与领域的描述,并借用高级语言的精巧结构简化电路的描述,具有电路模拟与验证机制,保证设计的正确性,支持电路描述由高层向低层的综合变换,易于理解和重用。此外,它还是一种与实现技术相独立的语言,既不束缚于某一特定的模拟程序或数字装置上,也不把设计方法强加于设计者。它允许设计者在其使用范围内选择工艺和方法。为了适应未来的数字硬件技术,VHDL 还提供了便于将新技术引入现有设计的潜力。VHDL 语言的最大特点是描述能力极强,可覆盖逻辑设计的诸多领域和层次,并支持众多的硬件模型。其特点包括:

(1) 设计技术齐全,方法灵活,支持广泛

VHDL 语言支持自顶向下(top down)和基于库(library-based)的设计方法,而且还支持同步电路、异步电路以及其他随机逻辑电路的设计。其范围之广是其他 HDL 语言所不能比拟的。另外,由于 VHDL 语言早在 1987 年 12 月就已作为 IEEE std 1076 标准公开发布,因此,目前大多数 EDA 工具几乎在不同程度上都支持 VHDL 语言。这样就给 VHDL 语言的进一步推广和应用创造了良好的环境。

(2) 系统硬件描述能力强

VHDL 语言具有多层次描述系统硬件功能的能力,可以从系统的数学模型直到门级电路;并且,高层次的行为描述可以与低层次的 RTL(寄存器传送语言)描述和门级描述混合使用。VHDL 语言能够进行系统级的硬件描述,这是它的一个最突出的优点。

(3) VHDL 语言可以与工艺无关地进行编程

在用 VHDL 语言设计系统硬件时,没有嵌入与工艺相关的信息(当然这些信息也是可以用 VHDL 描述的)。与大多数 HDL 语言的不同之处是,当门级或门级以上层次的描述通过模拟验证之后,再用相应的工具将设计映射成不同的工艺(如 MOS, CMOS 等)。这样,在工艺更新时,就无须修改原设计,只要改变相应的工艺映射工具即可。

(4) VHDL 语言标准、规范,易于共享和重用

由于 VHDL 语言已作为一种 IEEE 的工业标准,这样设计的成果便于重用和交流,反过来又能进一步推动 VHDL 语言的推广和完善。另外,VHDL 语言的语法比较严格,风格类似于 Ada 语言,给阅读和使用都带来极大的方便。

0.3 VHDL 语言与模拟

VHDL 是为数字电路的建模和模拟(simulation)而制定的,是一种面向模拟的语言,它的语法中有许多方面均考虑到模拟的因素。包括 VHDL 的硬件相关结构、并发性特征和混合级描述及混合级模拟。

(1) VHDL 中的硬件相关结构

VHDL 具有许多与数字电路结构直接相关的概念,其中最主要的是元件,它是数字硬件结构——“黑盒”的抽象。VHDL 中的元件由实体、结构体两个概念共同描述完成。其中实体描述元件与外部环境的接口,其内部行为及结构是完全隐蔽的。实体的功能定义在称为结构体的分离单元中,结构体规定设计实体输入及输出之间的关系。一个实体可存在多个对应的结构体,分别以行为、结构、数据流及各种方式混合的描述方法实现。元件的存在使 VHDL 脱离普通程序语言的范畴,成为描述数字电路的专用硬件设计语言。

VHDL 中的信号(signal)概念是数字电路中连线的抽象,它是各元件、各进程之间进行通信的数据通路。VHDL 中信号的状态会影响与信号相关的进程的运行,体现数字系统的各单元的输入及输出的状态。

VHDL 中的进程(process)完成电路行为的描述,由一系列顺序语句组成,是 VHDL 设计中进行功能描述的基本单元。由于进程的执行是并发的,因此在 VHDL 中引入 delta 延迟概念,用于表示时间上无穷小的模拟步,是 VHDL 中模拟进程同步机制的关键。一个模拟时刻包括若干 delta 延迟,所有进程均可能在特定条件下,在同一时刻的任一 delta 延迟点上激活。设计者的设计意图有时希望忽略在 delta 延迟点上的变化,着重于计算一个模拟时刻结束时的稳定阶段的状态。因此 VHDL'93 引进了延迟进程的概念。该类进程只在某一时刻的最后一个 delta 延迟时激活,这样可以降低处理频率,尤其是当用于时序检查的时候。

(2) VHDL 并发性特征

VHDL 的并发性体现在两个方面,首先在使用 VHDL 进行数字电路设计时存在并发性,即 VHDL 支持设计分解,可使被分解的各子部分的设计并行完成。一个模型的设计主要由 3 部分组成:定义实体部分——确立模型与环境的接口;定义结构体部分——完成模型的功能描述;定义测试部分——为模型生成测试向量,并捕获模型输出信号状态以供分析。在设计流程方面,在系统分析阶段,系统分析者可将设计对象分为若干独立的子元件,交给若干设计小组实现。此阶段,系统分析者严格定义元件接口,并将元件之间的相互作用以文档形式提供给各设计小组。然后,各设计小组可独立并行地对子元件进行详细设计,并模拟验证子元件,确保正确无误。最终,系统设计者集成各子元件形成完整

的设计,对整个设计进行模拟验证。设计的并发性可极大地加快整体设计进程并提高设计质量。

其次,VHDL之所以称为硬件描述语言,很重要的一点是因为它在模拟执行上具有并发性,这一点很适于描述电路活动的并行性特点,是其他程序设计语言所不具备的。VHDL中的进程类似于UNIX操作系统的进程概念,它们的挂起、活动均是独立的。并发性使得VHDL的设计模拟可在并行机上进行,这样大大提高了模拟效率,是解决模拟时间瓶颈的方法之一。在并行模拟中,VHDL源描述经编译后,结构确立除完成通常的VHDL确立工作,还将每个进程静态地映射到特定的处理器上,计算的加载、通信频带的估算均来自于暂时性的分析及波形传播分析,以帮助获得合理的平衡的静态映射。

(3) 混合级描述及混合级模拟分析

VHDL的描述范围覆盖系统级、算法级、寄存器传输级和门级,具有连续性、完整性的特点。VHDL结构描述方式与行为描述方式有机结合,各描述层次之间彼此衔接,协调一致。目前,较常用的大规模集成电路的设计方法包括基于标准单元库的自底向上(bottom-up)的设计方法和自顶向下(top-down)便于早期优化的设计方法,以及自底向上和自顶向下有机结合的设计方法。由于设计规模日益增大,设计复杂度急剧增加,传统的设计起点偏重低层的方法,会因设计规模的庞大带来极大的工作量。因此提高设计层次,注重早期优化,是现行较好的设计方式。目前,设计对象整体的设计过程经历多个层次。首先,在较高的抽象层次,进行前期的概念设计,优化设计模型;然后经由高级综合工具综合,产生寄存器传输级网表;最后经低级综合工具,形成最终的设计结果。

因此,由于存在多层次设计,就需要多个层次上的模拟。VHDL模拟器可以完成混合级模拟,为各个层次的硬件设计提供有效的模拟,反映设计意图,供设计者调试其设计,是适应当前电路设计的最佳选择之一。

0.4 编译型模拟方法及 VHDL 与 C++ 的类比

模拟算法可分为解释型和编译型两种。解释型模拟器中存在一个模拟核心,它不断读取VHDL源描述或其编译后的中间格式数据,并对每一条语句解释性地执行。它的优点是概念清晰、关系明确,并且编程实现要简便一些;其明显的缺点就是对每一条语句的每次执行都要重新解释并执行,会带来很多不必要的预处理,效率很低。编译型模拟器将VHDL源描述直接转化为功能等价的可执行二进制代码,这样,在每一条语句的模拟执行过程中消除了冗余的预处理,将极大地改善系统性能。

编译型模拟器将VHDL源描述转换为功能等价的C++源代码,这种策略主要是基

于对 VHDL 和 C++ 语言语法特点的比较后得出的。从语言层次上讲,这两种语言都属于高级语言,在许多方面均具有相似性,可以从这种相似性出发,完成它们之间的转化。这些相似性主要体现在如下几个方面:

(1) VHDL 中的实体(entity)与 C++ 中的类(class)概念类似,对于 VHDL 中的任一实体,均可以将它翻译成 C++ 中的一个类。而与实体对应的结构体(architecture)则可以从前面的实体类中派生,这样它就可以共享在实体类中定义的所有数据。

(2) VHDL 中的进程(process)继续从相对应的结构体类中进行派生,这样它就可共享所有在结构体类中定义的数据。当一个结构体包含多个进程时,每一个进程都是结构体的派生类,为了避免同一结构体对象的重复定义,可以通过虚(virtual)基类的方式进行派生。

(3) VHDL 中的端口概念可以映射到 C++ 中的函数(包括构造函数)参数,类属则可以映射为函数的默认参数。

(4) 可以为一些 VHDL 特有的数据类型使用 C++ 定义类的方式定义一个新的数据类型。例如 VHDL 中的信号(signal)可为其定义一个类 QSignal,在类中保存信号的属性值,例如上次信号变化的时间及值等。这样,对于信号类型对象赋值的特殊性就可以通过等号运算符的重载来解决。此外,信号的一些预定义属性的运算也可以转化为对类对象的某个成员函数的调用求解过程。

(5) VHDL 的层次关系的实现方式主要由元件说明、例示和组装完成。可以对应于 C++ 语言中的成员类对象的概念。对于不同结构体的组装,则可以在定义成员类对象时,通过给出不同构造函数的参数选取不同的结构体。

(6) VHDL 行为描述中的几种顺序语句:变量赋值语句、if 语句、case 语句、loop 语句、next 语句、exit 语句、return 语句、null 语句、assert 语句、过程调用语句在 C++ 语言中均有类似的语法或库函数,做少量的修改即可进行转换。

0.5 本教程的编写说明及读者对象

本教程参照 IEEE Standards VHDL Language Reference Manual (IEEE Std 1076, 2000 版)撰写。其目标是全面、系统,并力求准确地阐明超高速集成电路的硬件描述语言(VHDL)。为了维护 VHDL 语言标准的权威性,作者力求既保持其完整和严谨性,原汁原味地奉献给读者,同时又补充部分例题,帮助读者理解有关的语法现象。希望通过阅读这些例题有助于提高读者编写 VHDL 描述的能力。本书主要面向从事 VHDL 语言高级综合和模拟验证工具研究、开发的研究生、教师、科研工作人员以及从事 EDA 专业的高级技术人员,也可作为高等学校计算机、信息处理、自动控制、电子工程和通信技术等专业

本书在组织结构上,每一章都是针对该语言的某一特定领域(方面)。每一小节讨论其独立的结构和概念,开始是一段介绍,然后按其固定文法的产生式说明该结构的语法。继之,以叙述形式说明结构的意义和一系列限制。每一章的最后大多以示例、注意和对其其他相关章节的引用为结束。

0.5.1 语法描述

VHDL 描述的形式以上下文无关文法形式来说明,而上下文无关文法是使用巴科斯范式的一种简单变体表示,特别是:

(1) Roman 字体的小写单词,有些包含内嵌的下划线,用于代表句法类别,例如:

```
formal_port_list
```

无论何时,如果使用句法类别的名称,那么除了语法规则本身以外,都采用空格代替下划线(这样,当提及上述句法类别时,在叙述性描述中出现的是“formal port list”)。

(2) 粗体单词用于代表保留字,例如 **array** 保留字只能在语法指示的地方使用。

(3) 一个产生式包含一个左边(left-hand side)、符号“::=”(读做“可以代替为”)以及一个右边(right-hand side)三部分。产生式的左边总是一个句法类别,右边是替代规则。产生式的意义是一种文本替代规则:左边出现部分都可以由右边的一个实例替代。

(4) 竖杠(|)将产生式右边的各个替代项分隔开来,除非它直接出现在一个左花括号的后面,在这种情况下,它代表自己本身,如下所示:

```
letter_or_digit ::= letter|digit  
choices ::= choice{|choice}
```

在第 1 个实例中,“letter_or_digit”的出现可以由“letter”或“digit”替代。对于第 2 个实例,“choices”可以由一系列“choice”替代,每一个由竖杠分隔开(参见说明花括号含义的第(6)项)。

(5) 方括号[]括起产生式右边的可选项,因此,以下两个产生式是等价的:

```
return_statement ::= return[expression];  
return_statement ::= return;|return expression;
```

但是,需要注意的是,产生式右边用于标记(参见 2.3.2 节)的开始方括号和结束方括号是标记语法的一部分,并不指示整个右边都是可选的。

(6) 花括号{}括起产生式右边的一个或多个重复项。这些项可能出现 0 次或多次;正如一个等价的左递归规则一样,重复性地从左到右出现。因此,以下两个产生式是等价的:

```
term ::= factor{multiplying_operator factor}
```

$\text{term} ::= \text{factor} | \text{term multiplying_operator factor}$

如果任何句法类别的名称以一个斜体部分开始,则它与没有斜体部分的类别名称等价。斜体部分的目的是传递某一语义信息。例如,*type_name* 和 *subtype_name* 与单独的 name 在句法上是等价的。

术语 *simple_name* 用于已经代表了某一已说明实体的标识符的出现。

0.5.2 语义说明

在句法产生式之后,立即有一系列的叙述性规则以说明一个特定结构的意义和限制。在这些规则中,斜体的项指示该项的定义,完全以大写字母出现的标识符所指的是 STANDARD 包中的定义(参见附录 D.2)。

以下术语在这些语义描述说明中被使用,其意义如下:

易错的: 表示一种形式不正确的说明;但是,实现时没有必要探测并报告这一情况。只有在语言的处理过程中,通常没有可能探测到这种情况时,才确认该情况是易错的。

错误: 表示一种形式不正确的说明;但是,实现必须探测该情况并对工具的使用者报告错误。

非法: “错误”的同义词。

合法: 表示一种形式完好的说明。

本标准与许多其他 IEEE 标准不同,它们是采用动词应当指示标准的强制要求,用动词可能指示可选择的特征。而本标准则从头至尾一律采用动词是。于是在所有情况下,动词是都被解释为具有强制重要性。此外,还使用必须一词指示强制重要性。该词比常见的应当一词更合适,因为必须在本标准中对不同的读者具有不同的含义。

(1) 对于 VHDL 语言设计工具的研发者来说,必须表示标准强加的要求。要求最终的实现必须执行这一要求。

(2) 对于 VHDL 语言的模型研发者来说,必须表示 VHDL 语言的特征是该语言定义的自然结果。要求模型的研发者坚持这些特征所暗示的约束。

(3) 对于 VHDL 语言模型的用户来说,必须代表该模型的特征是语言定义的自然结果。模型的用户可依靠其 VHDL 语言源文本所暗示的模型的特征。

本书的读者最好已初步熟悉 VHDL 语言。为了能够更顺利而深入地掌握 VHDL 标准文本,建议读者先阅读并理解本书附录中有关 VHDL 语言的术语和语义的形式说明等内容。此外,为了使已熟悉 C++ 语言的读者能尽快地掌握和理解 VHDL 语言,特地在绪论中扼要讨论了 VHDL 与 C++ 的类比。

刘明业教授主持并完成本书稿的撰写。参加撰写工作的还有吴清平博士、谢巍博士、

袁媛博士以及叶梅龙教授。陈辉煌教授在百忙中审阅了原稿,提出了许多宝贵意见。此外,宋磊、张明杰、梁玉民及刘涛等同志为本书稿的校对、修改和部分录入工作付出了大量艰辛的劳动。本书的撰写工作承蒙厦门大学各级领导、师长和朋友的关怀、指导和帮助,在此一并致以诚挚的谢意。

限于我们的工作实践和认识水平有限,书中难免存在缺点、疏忽甚至错误之处,殷切希望广大读者批评指正。

刘明业

2002 年秋

于厦门大学