

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

高等学校电工电子实践系列

现代电子技术

—VHDL与数字系统设计

杨刚 龙海燕 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

高等学校电工电子实践系列

现代电子技术——VHDL 与数字系统设计

杨 刚 龙海燕 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书作为现代电子技术课程教学的重要组成部分,将 VHDL 语言与数字系统设计有机地结合起来。书中第 1 章为 VHDL 语言基础,介绍 VHDL 的语句和语法;第 2,3 章为组合电路、时序电路的 VHDL 基础实验,使读者轻松入门并巩固数字电路基础知识;第 4 章为状态机设计;第 5 章为存储器设计;第 6 章为 DSP 设计实例;第 7 章为数字系统设计实例。全书共有 108 个设计范例,读者可结合 EDA 工具和 FPGA/CPLD 开发系统进行实践,参照设计范例先做改动设计,逐渐熟悉 VHDL 开发后,就可独立设计出自己的源程序。

本书可作为高等学校电类和绝大部分非电类专业本科生、专科生教材,亦可作为电视大学、职业学校以及远程教育、网络教育的教材,本书还可用做从事数字系统设计的工程技术人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

现代电子技术:VHDL 与数字系统设计/杨刚,龙海燕编著. —北京:电子工业出版社,2004. 4

(高等学校电工电子实践系列)

ISBN 7-5053-9812-1

I. 现... II. ①杨... ②龙... III. ①数字电路—电路设计—高等学校—教材②硬件描述语言,VHDL—程序设计—高等学校—教材 IV. TN79

中国版本图书馆 CIP 数据核字(2004)第 028593 号

策划编辑:章海涛

责任编辑:王颖

印刷者:北京牛山世兴印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×980 1/16 印张:21 字数:437 千字

印 次:2004 年 4 月第 1 次印刷

印 数:5000 册 定价:25.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

VHDL 硬件描述语言是目前国内外数字系统的主流设计语言,也是现代电子技术发展的重要依托之一。随着片上系统(SOC)的不断推陈出新,VHDL 的语言内涵也在向外延伸,现在已经有不少扩充子集支持 SOC 的开发。与此同时,各种 EDA 工具广泛支持 VHDL 的 IEEE 标准,并且近年来在 VHDL 建模、综合、仿真、测试和验证上都取得了较大突破。随着 FPGA/CPLD 器件规模在千万门级基础上继续扩大,随着嵌入式微处理器(如 ARM, Nios, MIPS)迅猛发展,VHDL 的应用也从传统的数字系统设计,扩展到超大规模集成电路设计、嵌入式系统设计当中,并在以通信、自动控制、计算机为代表的信息领域中占据重要地位。

基于 VHDL 的广泛应用和发展前景,近年来国内高校纷纷开设与 VHDL 相关的各种 EDA 教学。我校的 VHDL 教学开展也较早,通过这些年的教学和实践,我们深刻体会到借助 EDA 工具开展 VHDL 教学,对学生掌握数字电路基础知识,增强实践能力有事半功倍的效果。另外,VHDL 教学作为信息类专业基础平台课开设,对后续专业课程的学习有着重要的铺垫作用。例如,通过对状态机 VHDL 设计、存储器 VHDL 设计,对后续“计算机组成原理”、“嵌入式系统设计”和“计算机控制系统”等课程的学习而言,无疑是起着承上启下的作用;又如,通过数字信号处理(DSP)的 VHDL 设计,对于后续“信号与系统”、“数字信号处理原理”等课程而言,VHDL 提供了一种理论联系实际的桥梁,从而使老师教起来轻松,学生学起来有兴趣。

感谢电子工业出版社将本书列入其规划出版的《高等学校电工电子实践系列》教材丛书中,同时也鞭策我们尽快地总结这些年来的教学经验,同时吸纳当前 EDA 技术进展,从而形成教材。经过我们艰苦而仔细的耕耘,本书得以按时完稿。在写作风格上,我们力图从实践角度入手,没有拘泥于传统教学方式——将 VHDL 当做一门语言课(如 C 语言)来教。读者翻开本书可以看到,本书实践部分共有 108 个设计范例,前面的范例是以数字电路各组成要素为主线,密切结合数字电路基础教学内容来安排的。对学生而言,有了一种比照的基础,就能轻松入门并动手操作;对于 VHDL 的初学者而言,这样的安排,也能使其巩固数字电路基础知识并理解 VHDL 基本组成模块的设计。在掌握 VHDL 基本设计以后,读者可以回过头来仔细阅读第 1 章 VHDL 语法,并结合实例操作。在这之后,本书安排有状态机设计、存储器设计、DSP 设计以及数字系统设计实例,读者可结合 EDA 工具和 FPGA/CPLD 开发系统进行实践,参照设计实例先做改动设计,逐渐熟悉 VHDL 设计和仿真方法后,就可独立开发出自己的源程序了。

本书由杨刚、龙海燕共同编著。四川省自动化仪表研究所所长、四川大学电气信息学院教授汪道辉先生担任本书主审,同时参加本书审稿工作还有四川大学电气信息学院的田远富教授、黄华教授、夏建刚副教授,感谢他们审阅全书,并提出宝贵意见。作者在编写过程中还得到四川大学电气信息学院各位老师和领导的鼓励和支持,在此一并表示感谢!本书不足之处,恳请读者批评指正。

作 者
2004年1月于四川大学

目 录

第 1 章 VHDL 语言基础	1
1.1 概述	1
1.2 VHDL 程序结构	1
1.2.1 VHDL 程序的基本结构	1
1.2.2 实体	3
1.2.3 结构体	5
1.2.4 配置	7
1.3 VHDL 的设计资源	10
1.3.1 库	10
1.3.2 程序包	12
1.4 数据对象、类型及操作符	15
1.4.1 VHDL 文字	16
1.4.2 VHDL 数据对象	17
1.4.3 VHDL 数据类型	19
1.4.4 VHDL 操作符	27
1.4.5 属性描述与定义语句	31
1.4.6 VHDL 描述方式	35
1.5 VHDL 顺序语句	38
1.5.1 赋值语句	38
1.5.2 转向控制语句	40
1.5.3 等待语句	45
1.5.4 子程序调用语句	45
1.5.5 返回语句	46
1.5.6 空操作语句	46
1.5.7 其他语句	47
1.6 VHDL 并行语句	48
1.6.1 进程语句	49
1.6.2 块语句	52
1.6.3 并行信号赋值语句	55
1.6.4 并行过程调用语句	57
1.6.5 元件例化语句	58

1.6.6	生成语句	59
1.7	子程序	61
1.7.1	函数	62
1.7.2	重载函数	65
1.7.3	过程	65
1.7.4	重载过程	66
第 2 章	组合逻辑电路基础实验	67
2.1	基本门电路	67
2.1.1	基本门电路描述实验	67
2.1.2	VHDL 实验快速入门	68
2.2	三态门	82
2.3	总线缓冲器	84
2.3.1	单向总线缓冲器	84
2.3.2	双向总线缓冲器	86
2.3.3	74HC245 双向总线缓冲器	87
2.4	数据选择器	88
2.4.1	四选一数据选择器 1	88
2.4.2	四选一数据选择器 2	90
2.4.3	八选一数据选择器	91
2.4.4	八选一数据选择器的宏调用	92
2.5	数据分配器	93
2.6	编码器	95
2.6.1	八-三优先编码器 1	95
2.6.2	八-三优先编码器 2	97
2.6.3	二-十进制优先编码器	98
2.7	译码器	99
2.7.1	二-四译码器	99
2.7.2	三-八译码器	101
2.7.3	两位 BCD-七段静态 LED 译码器	102
2.7.4	十进制-二进制译码器	104
2.8	比较器	105
2.9	加法器	107
2.9.1	半加器	107
2.9.2	四位全加器	108
2.9.3	八位全加器	109
2.10	减法器	110

2.10.1	半减器	110
2.10.2	四位全减器	111
2.11	奇偶校验电路和数据检测器	112
2.11.1	奇偶校验位发生器	112
2.11.2	奇偶校验位检测器	114
2.12	移位器	115
2.13	补码生成电路	117
2.14	锁存器	117
2.15	算术逻辑单元 ALU 设计	119
2.15.1	两位算术逻辑单元 ALU 设计	119
2.15.2	带进位四位算术逻辑单元 ALU 设计	121
第 3 章	时序逻辑电路基础实验	125
3.1	RS 触发器	125
3.2	T 触发器	127
3.3	JK 触发器	128
3.4	D 触发器	130
3.4.1	D 触发器基本设计	130
3.4.2	D 触发器宏调用	133
3.4.3	三级 D 触发器延迟电路	134
3.5	寄存器	135
3.5.1	n 位寄存器	135
3.5.2	24 位寄存器	137
3.5.3	24 位寄存器的 LPM 设计	138
3.5.4	多种功能的寄存器设计比较	140
3.6	移位寄存器	143
3.6.1	双向移位寄存器	143
3.6.2	串入串出(SISO)移位寄存器	144
3.6.3	串入并出(SIPO)移位寄存器	146
3.6.4	并入串出(PISO)移位寄存器	147
3.7	异步计数器	148
3.8	同步计数器	150
3.8.1	同步二十四进制计数器	150
3.8.2	同步四十进制计数器	152
3.9	可逆计数器	154
3.10	移存型计数器	155
3.10.1	自启动格雷码钮环计数器	155

3.10.2	移存型七进制计数器	157
3.11	模可变计数器	159
3.11.1	离散模可变计数器	159
3.11.2	模 256 以内任意可变计数器	161
3.12	分频器	163
3.12.1	加法分频器	163
3.12.2	减法分频器	164
3.12.3	半整数分频器	166
3.12.4	积分分频器	167
3.13	脉冲发生和整形	168
3.13.1	顺序脉冲发生器 1	168
3.13.2	顺序脉冲发生器 2	170
3.13.3	由 D 触发器构成的脉冲整形电路	171
3.13.4	脉冲整形计数器	172
第 4 章	状态机设计	175
4.1	Moore 状态机	175
4.1.1	简单的 Moore 状态机	175
4.1.2	Moore 状态机设计比较	177
4.1.3	采用 Moore 状态机设计计数器	180
4.2	Mealy 状态机	183
4.2.1	采用 Mealy 状态机设计计数器	183
4.2.2	Mealy 状态机 1	186
4.2.3	Mealy 状态机 2	188
4.3	特定序列脉冲发生器	191
4.4	循环彩灯控制器	192
4.5	交通灯控制器	194
4.5.1	交通灯控制器 1	194
4.5.2	交通灯控制器 2	197
第 5 章	存储器设计	202
5.1	RAM	202
5.1.1	RAM 的 LPM 设计	202
5.1.2	RAM 的快速定制	204
5.2	双口 RAM	208
5.2.1	简单的双口 RAM 设计	208
5.2.2	单地址双口 RAM 的 LPM 设计	210
5.2.3	双地址双口 RAM 的 LPM 设计	212

5.2.4	双口 RAM 的宏调用设计	214
5.3	FIFO	216
5.3.1	FIFO 的 LPM 设计 1	216
5.3.2	FIFO 的 LPM 设计 2	218
5.4	ROM	221
5.4.1	简单的 ROM 设计	221
5.4.2	用 ROM 实现九九乘法表	222
5.5	堆栈	226
第 6 章	DSP 设计实例	229
6.1	串行数据检测器	229
6.2	伪随机码发生器	231
6.3	汉明编码器和译码器	232
6.3.1	汉明编码器	232
6.3.2	汉明译码器	233
6.4	循环冗余校验(CRC)	235
6.5	15 位流水线加法器	238
6.6	乘法器	241
6.6.1	普通乘法器	241
6.6.2	移位相加 8 位乘法器	243
6.6.3	乘法器的 LPM 设计	246
6.7	除法器	247
6.8	FIR 滤波器	249
6.8.1	可编程 FIR 滤波器	249
6.8.2	四抽头直接形式 FIR 滤波器	251
6.9	IIR 滤波器	253
6.9.1	IIR 滤波器 1	253
6.9.2	IIR 滤波器 2	254
6.9.3	IIR 滤波器 3	255
第 7 章	数字系统设计实例	258
7.1	BCD-七段动态 LED 译码显示电路	258
7.2	汽车尾灯控制电路	261
7.3	频率计	265
7.4	数字钟	272
7.5	密码锁	283
7.6	8 位脉宽调制器	289
7.7	硬件电子琴	292

7.8	函数发生器	294
7.9	AD0809 模拟信号采集	301
7.10	VGA 彩条信号发生器	305
7.11	通用异步收发器(UART).....	309
7.12	数字系统设计课题.....	313
7.12.1	乒乓球游戏机.....	313
7.12.2	四人抢答器.....	314
7.12.3	数字秒表.....	314
7.12.4	出租车计费器.....	315
7.12.5	汽车停车场停车车位显示系统.....	315
7.12.6	五层电梯控制器.....	315
7.12.7	自动售货机的控制电路.....	316
附录	掌宇 CIC310 CPLD/FPGA 开发系统使用简介	317
参考文献	324

第 1 章 VHDL 语言基础

1.1 概述

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仿真、验证和设计、综合等方面。目前常用的硬件描述语言有 VHDL、VerilogHDL、ABEL 等。其中 VerilogHDL 起源于集成电路的设计,ABEL 起源于可编程逻辑器件的设计,而 VHDL 则起源于 VHSIC 计划,这是美国国防部于 1980 年制定的,目标是为下一代集成电路的设计和制造而制定一种新的描述方法。VHSIC 计划和同期的 ARPANET(先进的计算机体系结构)计划,成为后来信息高速公路计划的基础。VHDL 的英文全称为 Very-High-Speed Integrated Circuit Hardware Description Language,是 IEEE 标准化的硬件描述语言,并且已经成为系统描述的国际公认标准,得到众多 EDA 公司的支持。

VHDL 有两个标准版本,最早是在 1987 年底,IEEE 公布了 VHDL 的标准版本 IEEE-1076。1993 年,IEEE 对 VHDL 进行了修订,从更高的抽象层次和系统描述能力方面扩展了 VHDL 的内容,公布了新版本的 VHDL,即 IEEE 标准的 1076-1993 版本。这两种版本在书写格式上有细微差别,不过绝大部分 EDA 软件都支持这两种版本的书写格式。1997 年,能够同时描述数字和模拟集成电路的 VHDL 语言标准 IEEE1076.1 发布,即 VHDL-AMS。

VHDL 语言覆盖面广,描述能力强,能支持硬件的设计、验证、综合和测试,是一种多层次的硬件描述语言。其设计描述可以是描述电路具体组成的结构描述,也可以是描述电路功能的行为描述。这些描述可以从最抽象的系统级直到最精确的逻辑级,甚至门级。

运用 VHDL 语言设计系统一般采用自顶向下分层设计的方法,首先从系统级功能设计开始,对系统高层模块进行行为描述和功能仿真。系统的功能验证完成后,将抽象的高层设计自顶向下逐级细化,直到与所用可编程逻辑器件相对应的逻辑描述。

1.2 VHDL 程序结构

1.2.1 VHDL 程序的基本结构

VHDL 主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外,VHDL 的语言形式和句法与一般的计算机高级语言十分相似。VHDL 程序的特点是将一项工程设计,或者称为设计实体(可以是一个元件、一个电路模块或一个系统)分成外部和内部两部分。外部为端口,是可视部分,内部为设计实体的功能和算法部分,是不可视部分。在对一个设计实体定义了外部界面后,一旦其内部开发完成后,其他的设计就可以直接调用这个实体。这种将设计实体分成内外部分的概念是 VHDL 系

统设计的一个重要特点。

下面是二-四译码器的 VHDL 描述。

【例 1. 2. 1】 VHDL 程序的结构示例(见 2. 7. 1 节中“二-四译码器”)

```
library ieee;
use ieee. std_ logic_ 1164. all;
entity d24_ conc is
    port(i:in std_ logic_ vector(1 downto 0);
         o:out std_ logic_ vector(3 downto 0));
end d24_ conc;
architecture a of d24_ conc is
begin
    process (i)
    begin
        case i is
            when "00" => o <= "0001 ";
            when "01" => o <= "0010 ";
            when "10" => o <= "0100 ";
            when "11" => o <= "1000 ";
            when others => o <= "XXXX ";
        end case;
    end process;
end a;
```

从上例可见,二-四译码器程序分为 library、entity、architecture 几个部分描述。一般而言,一个相对完整的 VHDL 程序(或称为设计实体)至少应包括三个基本组成部分:①库(library)、程序包(package)使用说明;②实体(entity)说明;③实体对应的结构体(architecture)说明。其中,库、程序包使用说明用于打开(调用)本设计实体将要用到的库、程序包;实体说明用于描述该设计实体与外界的接口信号说明,是可视部分;结构体说明用于描述该设计实体内部工作的逻辑关系,是不可视部分。在一个实体中,允许含有一个或多个结构体,而在每一个结构体中又可以含有一个或多个进程以及其他的语句。根据需要,实体还可以有配置说明语句。配置说明语句主要用于以层次化的方式对特定的设计实体进行元件例化,或是为实体选定某个特定的结构体。图 1. 2. 1 显示出了 VHDL 程序设计中比较全面的组成结构。

为了便于对 VHDL 语法描述,以及便于程序的阅读和调试,本书对有关符号作如下约定:

①语句结构描述中方括号“[]”内的内容为可选内容,“{ }”内的内容为可重复选项。

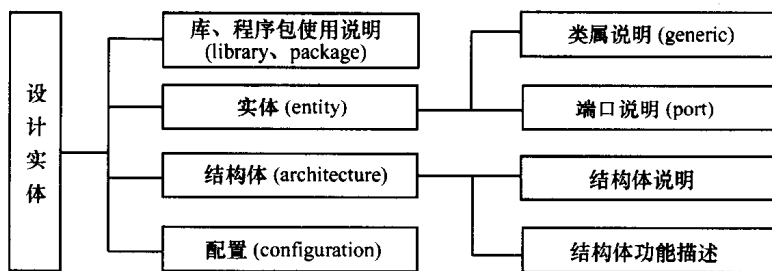


图 1.2.1 VHDL 程序设计基本结构

②对于 VHDL 的编译器和综合器来说,程序文字的大小写是不加区分的。本书一般使用小写。

③为了使程序的结构清晰,易于阅读与调试,书写和输入程序时,使用层次缩进格式,同一层次的语句对齐,低层次的语句较高层次的语句缩进两个字符。

④程序中的注释使用双横线“--”。在 VHDL 程序的任何一行中,双横线“--”后的文字都不参加编译和综合。

1.2.2 实体

实体(entity)是一个设计实体的表层设计单元,其功能是对这个设计实体与外部电路进行接口描述。它规定了设计单元的输入输出接口信号和引脚,是设计实体与外界的一个通信界面。

1. 实体语句结构

实体说明单元的常用语句结构如下:

```
entity 实体名 is
    [generic(类属表);]
    [port(端口表);]
end entity 实体名;
```

实体说明单元必须以语句“entity 实体名 is”开始,以语句“end entity 实体名;”结束,其中的实体名是设计者自己给设计实体的命名,可供其他设计实体对其进行调用时使用(考虑到 MAX+PLUS II 要求源程序文件的名字与实体名必须一致,因此建议各个源程序文件的命名均与其实体名一致)。中间在方括号内的语句描述,在特定的情况下并非是必需的。例如构建在 VHDL 仿真测试平台时就不需要方括号中的语句。

2. 类属说明语句

类属(generic)参量是一种端口界面常数,通常以一种说明的形式放在实体或块结构体前的说明部分。类属为所说明的环境提供了一种静态信息通道,类属的值可以由设计实体外部提供。因此,设计者可以从外面通过类属参量的重新设定而方便地改变一个设

计实体或一个元件的内部电路结构和规模,而不需要修改实体内部程序。

类属说明的一般书写格式如下:

```
generic([常数名;数据类型[:设定值]]{;常数名:数据类型[:设定值]});
```

【例 1.2.2】 类属语句的用法(见 6.5 节中 15 位流水线加法器)

```
entity add15 is
    generic( width:integer:=15;
             width1:integer:=7;
             width2:integer:=8;
             one:integer:=1);
    port(x,y:in std_logic_vector(width-1 downto 0);
          sum:out std_logic_vector(width-1 downto 0);
          clk:in std_logic);
end add15;
```

类属参量以关键词 generic 引导一个类属参量表,在表中提供时间参数或总线宽度等静态信息。类属表说明用于确定设计实体和其外部环境通信的参数,传递静态的信息。类属说明在所定义的环境中的地位十分接近常数,但却能从环境(如设计实体)外部动态地接受赋值,其行为又有点类似于端口 port。因此,常如以上的实体定义语句那样,将类属说明放在其中,且放在端口说明语句的前面。

3. 端口说明

由 port 引导的端口说明语句是对于一个设计实体界面的说明。实体端口说明的一般书写格式如下:

```
port(端口名:端口模式 数据类型;{端口名:端口模式 数据类型});
```

其中,端口名是设计者为实体的每一个对外通道所取的名字;端口模式是指这些通道上的数据流动方式,如输入或输出等;数据类型是指端口上流动的数据的表达格式。由于 VHDL 是一种强类型语言,它对语句中的所有操作数的数据类型都有严格的规定。一个实体通常有一个或多个端口,端口类似于原理图部件符号上的管脚。实体与外界交流的信息必须通过端口通道流入或流出。

IEEE1076 标准包中定义了 4 种常用的端口模式,各端口模式的功能及符号分别如表 1.2.1 和图 1.2.2 所示。在实际的数字集成电路中,in 相当于只允许输入的引脚,out

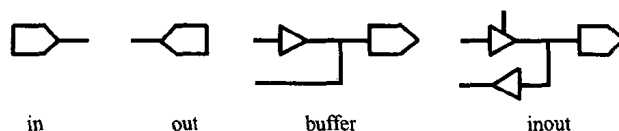


图 1.2.2 端口模式符号图

相当于只允许输出的引脚,buffer 相当于带输出缓冲器并可以回读的引脚(与三态引脚不同),而 inout 相当于双向引脚(即 bidir 引脚),此模式的端口是在普通输出端口(out)基础上增加三态输出缓冲器和输入缓冲器构成的。

表 1.2.1 4 种常用的端口模式说明

端口模式	端口模式说明
in	输入,只读模式,将变量或信号信息通过该端口读入
out	输出,单向赋值模式,将信号通过该端口输出
buffer	具有回读功能的输出模式,可以读或写,只能有一个驱动源
inout	双向,可以通过该端口读入或写出信息

1.2.3 结构体

结构体(architecture)是用于描述设计实体的内部结构以及实体端口间的逻辑关系。结构体内部构造的描述层次和描述内容可以用图 1.2.3 来说明。一般而言,一个完整的结构体由以下两个基本层次组成:

- 对数据类型、常数、信号、子程序和元件等元素的说明部分。
- 以行为、数据流、结构描述方式表达实体逻辑行为或功能的描述语句。

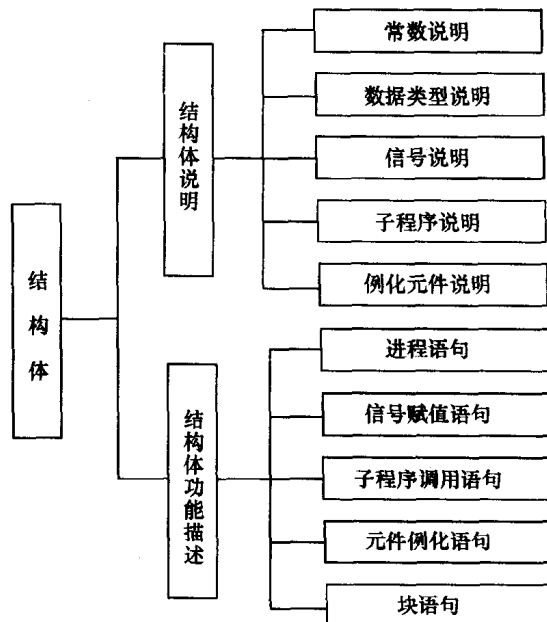


图 1.2.3 结构体构造图

结构体将具体实现一个实体。每个实体可以有多个结构体,每个结构体对应着实体的不同结构和算法的实现方案,其间的各个结构体的地位是同等的,但同一结构体不能为

不同的实体所拥有。结构体不能单独存在,它必须有一个界面说明,即对应着一个实体。对于具有多个结构体的实体,必须用配置(configuration)语句指明实际用于综合的结构体和用于仿真的结构体,即在综合后的可映射于硬件电路的设计实体中,一个实体只对应一个结构体。在电路中,如果实体代表一个器件符号,则结构体描述了这个符号的内部行为。当把这个符号例化成一个实际的器件安装到电路上时,则需配置语句为这个例化的器件指定一个结构体(即指定一种实现方案),或由编译器自动选定一个默认的结构体。

1. 结构体的一般语句格式

结构体的语句格式如下:

```
architecture 结构体名 of 实体名 is
    [说明语句]
begin
    [功能描述语句]
end architecture 结构体名;
```

其中,实体名必须与设计实体名字一致,而结构体名可以由设计者自己选择,但当一个实体具有多个结构体时,结构体的取名不可重复。

2. 结构体说明语句

结构体中的说明语句是对结构体的功能描述语句中将要用到的信号(signal)、数据类型(type)、常数(constant)、元件(component)、函数(function)和过程(procedure)等加以说明的语句。但在一个结构体中说明和定义的数据类型、常数、元件、函数和过程只能用于这个结构体中,若希望其能用于其他的实体或结构体中,则需要专门的程序包来处理。

3. 功能描述语句结构

如图 1.2.3 所示的结构体功能描述可以含有五种不同类型的,且是以并行方式工作的语句结构。而在每一语句结构的内部可能含有并行运行的逻辑描述语句或顺序运行的逻辑描述语句。各语句结构的基本组成和功能分别是:

①块语句是由一系列并行执行语句构成的组合体,它的功能是将结构体中的并行语句组成一个或多个模块,从而使程序结构清晰易读。

②进程语句定义顺序语句模块,用以将从外部获得的信号值,或内部的运算数据向其他的信号进行赋值。

③信号赋值语句将设计实体内的处理结果向定义的信号或界面端口进行赋值。

④子程序调用语句用于调用一个已设计好的子程序。

⑤元件例化语句对其他的设计实体进行元件调用说明,并将此元件的端口与其他的元件、信号或高层次实体的界面端口进行连接。