

应用电子教育系列

# VHDL 语言设计技术

陈耀和 著

电子工业出版社

**Publishing House of Electronics Industry**

北京·BEIJING

## 内 容 简 介

本书按照一种全新的方式编排内容,即按照 VHDL 语言程序总体结构中实体、构造体、库、程序包和元件配置等五大相对独立设计单元内容进行阐述。首先介绍 VHDL 语言的基本知识、程序总体结构、实体和构造体语句使用规则,然后按构造体的各种描述方法、库、程序包、元件配置、各种设计共享语句、属性描述、状态机设计和设计验证的顺序全面详细介绍其语法规则及程序设计方法。第 10 章从 6 个方面阐述 VHDL 语言的编程技巧,第 11,12 章列举了 74LS 系列通用集成电路、工控机组合逻辑、I/O 接口芯片、微处理器等典型设计实例,可供从事集成电路设计的工程技术人员参考,对提高 VHDL 语言设计水平很有帮助。

本书内容简明扼要,范例众多,通俗易懂。可供高等院校本科生、研究生作为教学参考书,对于集成电路设计工程技术人员来说,它是一本有价值的参考资料,也可作为有一定专业基础的读者的自学资料。

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

VHDL 语言设计技术 / 陈耀和著. —北京:电子工业出版社,2004.3

(应用电子教育系列)

ISBN 7-5053-9752-4

.V... 陈... .硬件描述语言, VHDL - 程序设计 .TP312

中国版本图书馆 CIP 数据核字(2004)第 018732 号

责任编辑:邓小瑜(dxym@phei.com.cn)

印 刷:

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:787×1092 1/16 印张:21 字数:538 千字

印 次:2004 年 3 月 第 1 次印刷

印 数:5000 册 定价:29.00 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 zllts@phei.com.cn,盗版侵权请发邮件至 dbqq@phei.com.cn。

# 前 言

现在集成电路成为电子、计算机、通信等高新产业的核心，已渗透到国民经济的各个领域，有力推动了社会生产力的发展和社会信息化程度的提高，也有助于保护自主知识产权。因此，集成电路已成为世界各国国民经济新的增长点。我国的集成电路生产，特别是设计产业相对落后。为加快设计产业的发展，在全国建设了多个产业基地，其任务是催生和扶持我国集成电路产业的发展。可以预见，在今后 20 年内，我国的集成电路设计产业将会快速发展，对高素质集成电路设计技术人才的需求将急剧增加，应大大加速高校的人才培养。作者多年来从事 VHDL 语言、数字系统高层次综合、工业控制计算机、集散控制系统、计算机容错技术等方面的教学、科研和设计工作。本书是作者多年来实践经验的总结，献给读者以抛砖引玉，希望能为促进我国集成电路设计产业的发展贡献一份微薄的力量。

VHDL 语言称硬件描述语言，是一门独具特色的高级语言，牵涉知识面之广，是其他高级语言无法比拟的。以作者的经验，学习 VHDL 语言必须有较好的硬件电路、C 语言、算法等基础，最好学点数字系统的高层次综合理论，高层次综合是行为描述逻辑综合的理论基础，可加深对 VHDL 语言更深层的理解，对提高设计技术很有益处。VHDL 语言受国外语言习惯的影响较大。本书尽量遵循中国人的语言习惯写作，且内容按照 VHDL 语言程序总体结构的规律进行编排，基本概念用易于理解的方式叙述，力求准确贴切。构造体主要用于对 IC 芯片硬件电路功能的描述，是 VHDL 语言最复杂的部分。作者把构造体常用的三种描述：逻辑、行为和结构描述语句，分别以三个章节的篇幅加以介绍，条理清楚，让读者容易掌握三种描述方法常用语句的使用规则。把部分语句归入设计共享，是因为不论哪种描述都要使用这些语句。属性描述比较抽象，主要用于较复杂的设计，简单设计除少数属性较常用外，一般很少用，将另外作为一章介绍。状态机设计是 VHDL 语言独具特点的描述风格，单独设一章。时序仿真是设计验证的手段，第 9 章以设计验证为题，强调验证的重要性。编程技巧一章则是作者经验的总结，读者可以各取所需。第 11 章介绍 74 系列通用集成电路设计，罗列了 74 系列各种类型电路的实用设计方法。第 12 章是综合设计举例，列举三个较复杂的设计范例，以供实用设计时参考。第 13 章列出 VHDL 语言 93 版对规范的修订，附录 A 中包含 IEEE 库标准逻辑包，为读者查阅提供方便。本书完整的例子都经过上机编译和综合，力求避免错误，读者可到中山大学理工学院或中山大学专用集成电路设计中心的网页 <http://spe.zsu.edu.cn> 下载引用。

由于作者水平有限，错漏难免，恳望广大读者指正。读者是我最好的老师，读者的批评指正或技术切磋、学术研讨，是我不断进步的源泉。

本书在写作过程中，得到刘动、庞志勇、谢清荣、陈天文、洪晋炜、宋菲、黄春艳、谢德英等同志的无私帮助，特此致谢！

作 者

中山大学专用集成电路设计中心

E-mail: yaohe@hotmail.com

# 目 录

第 1 章 集成电路设计概述 .....	1
1.1 集成电路技术发展的现状与预测 .....	1
1.2 集成电路设计方法学的诞生与发展 .....	2
1.3 集成电路设计方法的分类 .....	3
1.4 集成电路设计流程 .....	5
1.5 数字集成系统的设计与实现 .....	6
第 2 章 VHDL 语言设计基础 .....	7
2.1 VHDL 语言的由来与特点 .....	7
2.2 VHDL 语言的基础知识 .....	8
2.2.1 对象 (OBJECT) .....	8
2.2.2 数据类型 .....	10
2.2.3 类型转换 .....	14
2.2.4 运算操作符 .....	16
2.2.5 VHDL 语言的文字规则 .....	18
2.3 VHDL 语言的程序结构 .....	21
2.4 VHDL 语言的实体语句 .....	22
2.5 VHDL 语言构造体语句 .....	23
2.6 VHDL 语言构造体常用的几种描述方法 .....	24
2.7 VHDL 语言的逻辑综合与优化 .....	29
2.8 全加器的仿真 .....	30
2.9 VHDL 语言设计环境 .....	32
第 3 章 构造体逻辑 (数据流) 描述 .....	33
3.1 并行信号赋值语句 .....	33
3.2 信号驱动源 .....	36
3.3 VHDL 语言的延迟分类 .....	38
3.4 并行信号赋值语句的特殊形式 .....	42
第 4 章 构造体的行为描述 .....	44
4.1 进程语句 (PROCESS) .....	44
4.1.1 等待语句 (WAIT) .....	44
4.1.2 顺序信号赋值语句 .....	46
4.1.3 构造体多进程程序结构 .....	47
4.1.4 构造体的多进程描述 .....	48
4.2 顺序控制语句 .....	51
4.2.1 IF 条件语句 .....	51

4.2.2	CASE 分支语句 .....	60
4.2.3	LOOP 循环语句与 NEXT 退出语句 .....	62
4.2.4	FOR 循环语句 .....	64
4.2.5	WHILE 循环语句 .....	65
4.2.6	NEXT 跳出循环语句 .....	66
4.3	过程语句 (PROCEDURE) .....	67
4.3.1	过程说明 .....	67
4.3.2	过程体 .....	67
4.3.3	过程调用 .....	68
4.3.4	过程的复用 (重载) .....	69
4.4	REPORT 与 NULL 语句 .....	71
<b>第 5 章</b>	<b>构造体的结构描述 .....</b>	<b>73</b>
5.1	元件 (COMPONENT) 与例元 (INSTANCE) .....	73
5.1.1	元件说明语句 .....	73
5.1.2	例元语句 (INSTANCE) .....	74
5.1.3	结构描述中的信号 .....	75
5.2	规则结构 .....	75
5.2.1	生成语句 (GENERATE) .....	75
5.2.2	FOR 生成方式 .....	76
5.2.3	IF 生成方式 .....	77
5.3	参数传递语句 (GENERIC) .....	79
5.3.1	GENERIC 实体 .....	80
5.3.2	GENERIC 元件与例元 .....	82
5.4	构造体结构描述的程序结构 .....	83
5.5	结构描述的设计举例 .....	86
<b>第 6 章</b>	<b>VHDL 语言设计共享 .....</b>	<b>90</b>
6.1	块语句 (BLOCK) .....	90
6.2	断言语句 .....	95
6.2.1	顺序断言语句 .....	96
6.2.2	并行断言语句 .....	97
6.3	子程序 .....	98
6.4	函数 (FUNCTION) .....	99
6.4.1	函数说明 .....	99
6.4.2	函数体 .....	99
6.4.3	RETURN 返回语句 .....	100
6.4.4	类型转换函数 .....	100
6.4.5	数学运算函数 .....	103
6.4.6	决断函数 .....	105
6.4.7	函数的调用 .....	107
6.5	程序的复用 (重载) 设计 .....	107

6.5.1	复用 (重载) 函数.....	107
6.5.2	运算符的复用 (重载) .....	110
6.6	库 .....	111
6.6.1	库的分类 .....	111
6.6.2	库语句 (LIBRARY) .....	112
6.6.3	USE 语句.....	113
6.7	程序包 (PACKAGE) .....	113
6.7.1	程序包说明 .....	114
6.7.2	程序包体 (BODY) .....	114
6.7.3	标准库的程序包 .....	116
6.7.4	程序包集合举例 .....	119
6.8	元件配置 (CONFIGURATION) .....	124
6.8.1	体内配置 .....	125
6.8.2	体外配置 .....	130
6.8.3	默认配置 .....	134
6.9	块配置 .....	135
6.10	VHDL 语言的混合描述.....	137
<b>第 7 章</b>	<b>VHDL 语言属性描述 .....</b>	<b>140</b>
7.1	数值类属性.....	140
7.1.1	常用数据的数值属性.....	140
7.1.2	数组的数值属性 .....	142
7.2	函数类属性.....	143
7.2.1	数据类型属性函数.....	143
7.2.2	数组属性函数 .....	145
7.2.3	信号属性函数 .....	147
7.3	信号类属性.....	151
7.4	数据类型类属性.....	157
7.5	数组区间类属性.....	158
7.6	用户自定义属性 (ATTRIBUTE) .....	159
<b>第 8 章</b>	<b>有限状态机设计.....</b>	<b>161</b>
8.1	有限状态机的分类.....	161
8.2	有限状态机的描述方法.....	161
8.3	有限状态机的设计举例.....	162
<b>第 9 章</b>	<b>VHDL 语言设计验证 .....</b>	<b>167</b>
9.1	时序仿真输入激励信号的描述 .....	167
9.1.1	用时间表达式描述.....	167
9.1.2	用读文件 (TEXTIO) 描述 .....	172
9.2	VHDL 语言的设计验证.....	174
9.2.1	简单验证 .....	174
9.2.2	自动验证 .....	175

第 10 章 VHDL 语言编程技巧.....	179
10.1 各种描述方法的选择.....	179
10.2 库与资源利用.....	186
10.3 元件配置.....	188
10.4 算法优化.....	190
10.5 时序仿真和硬件验证.....	193
10.6 特殊信号的描述方法.....	195
10.6.1 脉冲信号发生器描述.....	195
10.6.2 时钟边沿的描述.....	195
10.6.3 异步复位置位触发器描述.....	197
10.6.4 同步复位置位触发器描述.....	197
10.6.5 异步复位同步置位 D 触发器描述.....	198
10.6.6 主从 D 触发器的描述.....	199
10.7 特殊实体的描述技巧.....	200
10.7.1 编码电路的描述.....	200
10.7.2 时序与组合电路混合系统的描述方法.....	202
10.7.3 跳变触发的时序配合.....	204
10.7.4 电平触发提高抗干扰能力.....	205
10.7.5 用空操作节约硬件资源.....	207
第 11 章 通用集成电路设计.....	209
11.1 逻辑门电路实用设计.....	209
11.1.1 74LS04 六反相器.....	209
11.1.2 74LS08 四 2 输入与门.....	210
11.1.3 74LS10 三 3 输入与非门.....	212
11.1.4 74LS32 四 2 输入或门.....	213
11.1.5 74LS86 四 2 输入异或门.....	213
11.2 触发器、锁存器实用电路设计.....	214
11.2.1 74LS74 双 D 型正跳变触发器.....	214
11.2.2 74LS103 双 J-K 型负跳变触发器.....	216
11.2.3 74LS373 8 位数据锁存器.....	218
11.3 编码器、译码器实用电路设计.....	219
11.3.1 74LS49 BCD 七段编码器.....	220
11.3.2 74LS138 3-8 线译码器.....	222
11.3.3 74LS145 BCD-十进制译码器.....	223
11.3.4 74LS157 四 2-1 线数据多路开关.....	225
11.4 数据缓冲器.....	226
11.4.1 74LS244 8 位单向数据缓冲器.....	226
11.4.2 74LS245 8 位总线双向缓冲器.....	228
11.5 移位寄存器.....	229
11.5.1 74LS166 8 位移位寄存器.....	229

11.5.2	74LS194 4 位双向移位寄存器 .....	231
11.6	计数器实用电路设计 .....	234
11.6.1	74LS393 双 4 位二进制计数器 .....	234
11.6.2	74LS92 十二进制计数器 .....	236
<b>第 12 章</b>	<b>VHDL 语言综合设计举例 .....</b>	<b>238</b>
12.1	8255 可编程 I/O 接口芯片设计 .....	238
12.1.1	8255 的技术指标与引脚功能 .....	238
12.1.2	8255 接口芯片的内部结构 .....	239
12.1.3	8255 的工作方式编程 .....	240
12.1.4	8255 I/O 芯片的程序设计 .....	242
12.1.5	8255 接口芯片设计的 VHDL 语言描述 .....	243
12.2	4 位微处理器的设计 .....	248
12.2.1	AM2901 微处理器的内部结构 .....	248
12.2.2	AM2901 微处理器的微指令系统 .....	250
12.2.3	AM2901 4 位微处理器顶层实体设计 .....	251
12.2.4	AM2901 4 位微处理器顶层元件包设计 .....	253
12.2.5	AM2901 微处理器的元件设计 .....	255
12.2.6	创建基本元件库 .....	263
12.3	STD 总线容错工业控制计算机组合逻辑设计 .....	273
12.3.1	STD 总线多模容错工业控制计算机简介 .....	273
12.3.2	逻辑电路的 VHDL 语言设计 .....	274
12.3.3	CPLD 芯片的 VHDL 开发设计 .....	275
<b>第 13 章</b>	<b>VHDL 语言 93 版对规范的修订 .....</b>	<b>289</b>
13.1	VHDL 语言 93 版对规范的修订 .....	289
13.1.1	文件是 VHDL 语言新的对象 .....	289
13.1.2	在端口映射中使用常量表达式 .....	290
13.1.3	定义共享变量 .....	291
13.1.4	定义 GROUP .....	291
13.1.5	定义新的属性 FOREIGN .....	292
13.1.6	语句描述上的区别 .....	292
13.1.7	扩展标注的标号 .....	293
13.1.8	纯函数和非纯函数 .....	294
13.1.9	标识 (Signature) .....	294
13.1.10	文件操作的定义 .....	294
13.1.11	扩大属性的使用范围 .....	295
13.1.12	增加逻辑操作符 .....	295
13.1.13	REPORT 语句 (报告语句) .....	295
13.1.14	信号延时可指定脉冲宽度限制 .....	296
13.1.15	可对信号赋无效值 .....	296
13.1.16	延迟过程 .....	296

13.1.17	增加参数传递语句 ( GENERIC ) .....	296
13.1.18	生成语句可包含端口说明 .....	296
13.1.19	扩展字符集 .....	297
13.1.20	定义扩展标识符 .....	297
13.1.21	位串 .....	297
13.1.22	增加预定义属性 .....	297
13.1.23	扩充标准包集合 ( STANDARD ) .....	299
13.2	87 版 93 版的移植问题 .....	299
附录 A	IEEE 库标准逻辑程序包 .....	300

# 第 1 章 集成电路设计概述

## 1.1 集成电路技术发展的现状与预测

在现代高新科学技术时代，集成电路已成为电子、计算机、通信等产业的核心。从 1958 年美国得州公司（TI，Texas Instrument）设计出第一片集成电路，四十多年来，集成电路的发展经历了小规模—中规模—大规模—超大规模（VLSI，Very Large-Scale Integrated）阶段，现在已进入特大规模（ULSI—Ultra Large-Scale Integrated）阶段。特别是近二十年来，计算机技术的发展促进集成电路技术的发展，近年来集成电路技术发展速度十分惊人，以特征尺寸为标志的生产工艺从微米（ $1\mu\text{m}$  以上）亚微米（ $0.1 \sim 0.9\mu\text{m}$ ）向深亚微米（小于  $0.1\mu\text{m}$ ）的深度发展，当前集成电路生产工艺已进入  $0.12\mu\text{m}$  阶段，实验室水平达  $0.1\mu\text{m}$ 。集成电路生产工艺从微米发展到深亚微米，对集成电路的设计及其设计方法学均提出很多新问题。预计到 21 世纪初，集成电路的生产进入深亚微米阶段，将对集成电路设计提出新的挑战，首先要解决的是建立起深亚微米精确的器件模型、时序模型及互连模型。在深亚微米级电路设计中，一个突出矛盾是布版布图后的时序模拟和速度符合问题，因互连线分布参数引起的延迟将超过门的延迟，且由于集成电路工作频率的进一步提高，时序引起的容差变小，传输延迟的影响越来越大，这给集成电路设计方法学的研究带来相当大的困难。再过 20 年，集成电路生产工艺将进入  $0.07\mu\text{m}$  以下，也就是进入纳米级，可以预言目前传统的集成电路生产工艺将不适用。现在已有公司在研究量子效应晶体管和纳米器件，届时沿用现有的集成电路生产工艺是无法实现的，集成电路从设计到生产将面临一场技术革命。

1994 年世界半导体工业协会公布了 1995 ~ 2010 年 15 年中集成电路技术发展的预测，见表 1-1。

表 1-1 1995 ~ 2010 年集成电路技术发展预测

年 代	1995	1998	2001	2004	2007	2010	
特征尺寸 ( $\mu\text{m}$ )	0.35	0.25	0.18	0.13	0.10	0.07	
DRAM (Mb/片)	64	256	1024	$4 \times 1024$	$16 \times 1024$	$64 \times 1024$	
SRAM (Mb/ $\text{cm}^2$ )	2	6	20	50	210	300	
ASIC (个晶体管/ $\text{cm}^2$ )	$2 \times 10^6$	$4 \times 10^6$	$7 \times 10^6$	$12 \times 10^6$	$25 \times 10^6$	$40 \times 10^6$	
MPU (个晶体管/ $\text{cm}^2$ )	$4 \times 10^6$	$7 \times 10^6$	$13 \times 10^6$	$25 \times 10^6$	$50 \times 10^6$	$90 \times 10^6$	
频率 (MHz)	一般	150	200	300	400	500	625
	高性能	300	450	600	800	1000	1100
尺寸 ( $\text{mm}^2$ )	DRAM	190	280	420	640	960	1400
	MPU	250	300	360	430	520	620
	ASIC	450	660	750	900	1100	1400
连线层数 (层)	4 ~ 5	5	5 ~ 6	6	6 ~ 7	7	
工作电压 (V) 桌上	3.3	2.5	1.8	1.5	1.2	0.9	
电池 (V)	2.5	1.8 ~ 2.5	0.9 ~ 1.8	0.9	0.9	0.9	

从近几年来集成电路技术的发展速度来看，以上时间表已经落后。

## 1.2 集成电路设计方法学的诞生与发展

随着集成电路技术的不断发展和集成度的迅速提高，集成电路设计越来越复杂，因而急需对设计方法和设计工具不断变革，进行“设计革命”。各种计算机辅助设计工具的涌现及设计方法学的诞生，正是为了这样的需求。

回顾三十多年来集成电路设计自动化的发展大致可分为以下三个阶段。

20 世纪 70 年代，第一代集成电路设计工具称计算机辅助设计（CAD，Computer Aided Design）。以交互式图形编辑和设计规则检查为特点，硬件平台采用 16 位小型计算机。从逻辑图输入开始，可以完成逻辑模拟、电路模拟与版图设计等功能。但版图设计与验证是分别进行的，设计人员需对两者的结果进行多次比较及反复修改，才能得到正确的掩膜版图。第一代 CAD 工具使设计人员摆脱了繁复且易出错的手工画图、刻红膜的传统方法，大大提高了效率，但尚不能适应较大规模芯片的设计。

20 世纪 80 年代出现第二代集成电路设计工具，常称计算机辅助工程（CAE，Computer Aided Engineering）。以 32 位工作站作为硬件平台，设计人员仍然以逻辑图输入的方式开始设计集成电路，不仅有设计全定制集成电路的版图编辑工具，还包括门阵列、标准单元的自动设计。有经过集成电路制造商验证，针对不同工艺的单元库，对门阵列、标准单元电路可完成自动布局布线。更重要的是引入了版图与逻辑图间的一致性检查工具（Layout Versus Schematic），对版图参数进行提取得到相应的逻辑图，与设计所依据的原理图进行比较，可发现版图设计是否有错，还将版图分布参数引入电路图作为电路后模拟，进一步检查时序和速度是否仍符合原设计要求，一般均可保证集成电路投片的一次成功。但一致性检查和“后模拟”仍在最后阶段才实施，一旦发现错误，还需修改版图或逻辑图，仍需付出相当的代价和时间，逻辑图输入的方法仍不适应复杂大规模集成电路的设计。

进入 90 年代，集成电路越来越复杂，数十万门、甚至数百万数千万门的电路设计需求越来越多，依靠原理图输入的方式已无法实现，第三代称为电子设计自动化（EDA，Electronic Design Automation）设计工具的主要标志就是引入 VHDL（VHSIC Hardware Description Language）硬件描述语言，其特点是高层次设计自动化，现在世界上常用版本有两种：VHDL 语言和 Verilog HDL 语言。首先对设计 IC 进行行为描述、逻辑（数据流）描述、结构描述或混合描述，由逻辑综合工具综合后得到逻辑原理图及网表文件。在 VHDL 语言输入、编译后，可进行前仿真，即时序仿真，大大提高处理复杂设计的能力，设计周期大大缩短，综合优化工具使芯片的品质，如：面积、速度、功耗等获得优化。由于 VHDL 硬件描述语言可读性强，易于修改和发现错误，这种高层次设计与工艺无关，一个芯片设计的 VHDL 语言可通过逻辑工具综合为复杂可编程逻辑器件（CPLD，Complex Programmable Logic Device）或现场可编程门阵列（FPGA，Field Programmable Gate Array）的代码，由相关编程器写入 CPLD 或 EEPROM（Electrically-Erasable Programmable Read-Only Memory），立即可进入 PCB，大大缩短产品的设计周期，使产品一方面迅速占领了市场，另一方面相当于该设计进行硬件验证正确无误。VHDL 语言的原码又可以综合成某个工艺支持的专用集成电路（ASIC，Application Specific Integrated Circuit）大批量生产。另外，工艺技术的进步，需采用更先进的工艺，仍可利用 VHDL 语言原码逻辑综合后得到的网表文件，重新布局布线，得到新工艺支持的掩膜版图，使设计变得非常灵活。

采用 VHDL 语言高层次设计自动化工具，使设计人员在逻辑综合前就可进行时序仿真，根据时序仿真结果修改 VHDL 语言，从而获得最优化的设计方案，这样避免做到掩膜版图才回过头来修改 VHDL 语言的麻烦，赢得宝贵的设计时间，大大缩短了设计周期。

## 1.3 集成电路设计方法的分类

目前集成电路已渗透到国民经济的各个应用领域，从高性能复杂的微处理器到家电、玩具使用的简单专用集成电路，其产量、性能、价格均有很大差别，因而设计方法和手段也就有所不同。我们希望能尽可能短的时间内以最低的成本来获得最佳的设计方案，所设计的芯片面积最小，功耗最低。但实际上要全面达到这些指标是很困难的，只能进行某种折中，这就派生了各种集成电路的设计方法，现将集成电路各种不同的设计方法简介如下。

### 1. 全定制法 ( Full-custom Design Approach )

对 ASIC 设计模块的功能用 VHDL 语言进行高层次描述，经过编译、时序仿真、逻辑综合优化得到网表文件，根据网表文件进行版图设计、版图参数提取验证，完成集成电路设计的全过程。

其特点是适用于要求得到最高速度、最低功耗和最省面积的 ASIC 芯片，但设计周期长，设计成本高，适于年产 10 万片以上的产品。

### 2. 定制法 ( Custom Design Approach )

定制法也叫库单元法。把精心设计的各种物理单元 ( 版图 ) 存入单元库中，根据电路的要求从库中调出所需的单元及压焊块进行布版布图，直接得到所要求的 ASIC 掩膜版图。单元通常分两大类。

- 标准单元：物理单元的高度相同，宽度不同。
- 通用单元：有积木法和混合法两种，各物理单元的高度和宽度不再相等。

其特点是设计的自由度较大，芯片面积较小，设计周期较全定制法短，成本也低，但建立一个较完善的物理单元库，需要很大的初始投资。适用于芯片性能指标要求较高，生产批量较大的芯片设计。

### 3. 半定制法 ( Half-custom Design Approach )

半定制法就是对一批已完成连线以外所有芯片加工工序的芯片，进行单独设计和制作过线孔和压焊块，以完成特定电路要求。这种已加工好的芯片有数字电路阵列和模拟电路阵列两大类。

其特点是大大缩短芯片设计制作的周期，成本大大下降，但门阵列中门的利用率较低。适用于芯片要求设计成本低、周期短，而生产批量较小的芯片设计。用半定制法迅速设计出产品占领市场后，再用全定制法等其他方法进行“再设计”扩大生产。

### 4. 模块编译法 ( Module Compile )

模块编译法首先对设计芯片的性能进行描述，通过编译直接得到芯片的掩膜版图，编译时需要专用的编译器和版图单元库。

这种方法主要适用于 ROM、RAM、ALU、移位寄存器等规则结构和模块化结构的芯片设计。

### 5. 可编程逻辑器件法 (Programmable Logic Device Approach)

可编程逻辑器件是一种完成了全部工艺制造,可直接从市场上购得的产品。新芯片没有任何逻辑功能,但一旦经编程器写入代码,就可在该器件上实现设计人员所要求的逻辑功能。前面四种设计方法首先由设计人员完成芯片设计,最后还要交集成电路制造商完成。但 CPLD 的编程,由设计人员通过开发工具便可完成用户功能的设计,直接上 PCB (Printed Circuit Board) 变成产品,大大缩短设计周期,让产品以最快的速度占领市场。可以说,可编程逻辑器件的出现给硬件电路设计带来根本性的变革。

PLD 包括 :PLA( Programmable Logic Array )可编程阵列逻辑、GAL( Generic Array Logic )通用可编程阵列逻辑、CPLD 复杂可编程逻辑器件等。这种器件以“与”“或”矩阵结构为基础,其特点是把编程程序综合成代码,写入片内 EEPROM,保密性好,但其容量较小,编程次数也很有限,一般为 1 万次以下。

### 6. 可配置逻辑门阵列法 (Configurable Logic Gate Array)

可配置逻辑门阵列器件也称 FPGA ( Field Programmable Gate Array ),即现场可编程门阵列。但这个称谓并不确切,FPGA 不是一种门阵列,其内部是由可配置逻辑功能块( Configurable Logic Block )排成阵列,芯片四周为可编程 I/O 功能块( Programmable I/O Block ),单个芯片规模从几千门到数百万门以上。FPGA 也是通过开发工具对其编程实现设计人员所要求的功能,其特点是可实现比 CPLD 更复杂的芯片功能,其编程代码存于片内 RAM 中,编程次数不受限制,但在 PCB 上要增加一片串行 EEPROM,以便在加电时把代码从 EEPROM 调入 FPGA 的内部 RAM 中运行,才能实现用户所需功能。因其价格较贵,一般用做小批量生产,让产品占领市场后再设计成全定制的 ASIC 芯片,扩大生产。

要对上述各种设计方法作出比较是困难的,但从图 1-1 所示的成本与产量关系可以看出,不同的设计方法要求有不同的最小产量,如 :对全定制设计芯片只有产量超过 10 万以上,其成本才合算。

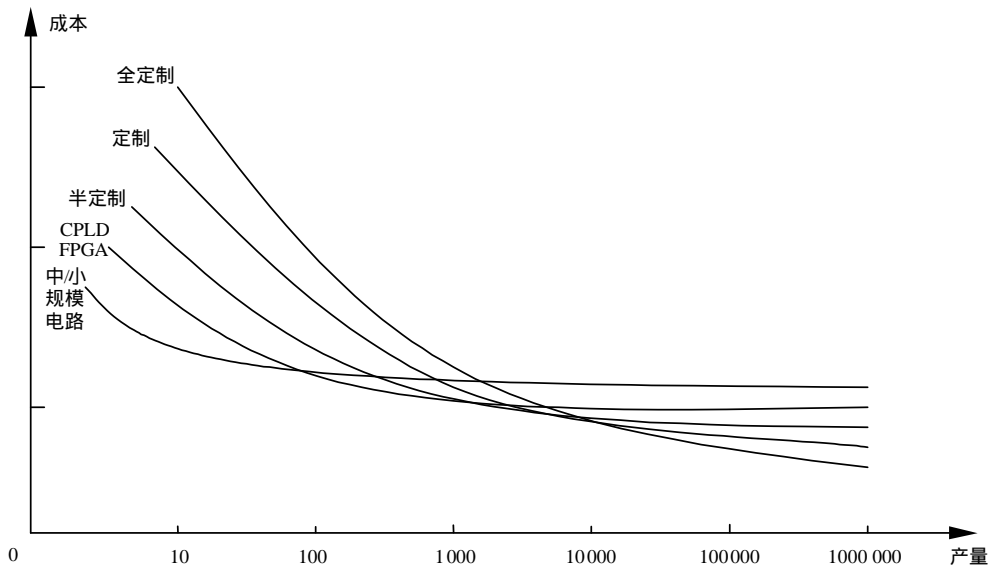


图 1-1 不同设计方法的成本与产量关系图

## 1.4 集成电路设计流程

集成电路设计一般分为三个子过程。

### 1. 高层次设计

集成电路功能用 VHDL 语言进行高层次的行为描述、逻辑(数据流)描述、结构描述或混合描述,同时通过高层次的前仿真,即时序仿真进行验证。

### 2. 高层次逻辑综合

通过逻辑综合工具将高层次的描述转换成低层次的逻辑单元,最后生成网表文件。

### 3. 物理综合

将网表文件转换成掩膜版图,即完成整个芯片的设计。专用集成电路 ASIC 的典型设计流程如图 1-2 所示。

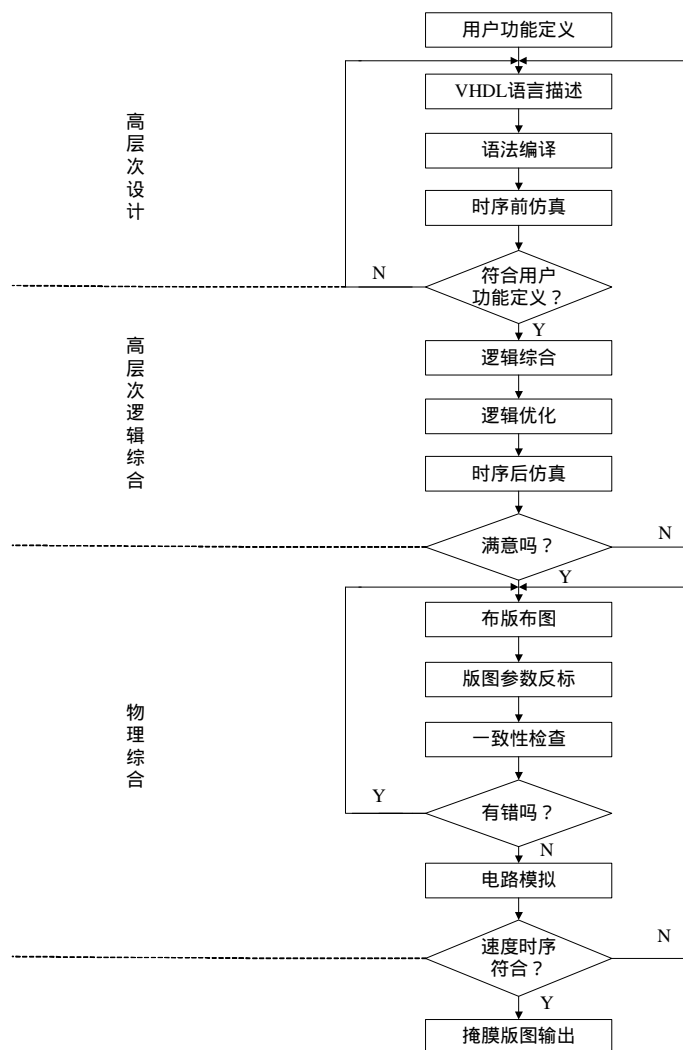


图 1-2 ASIC 的典型设计流程

### 1.5 数字集成系统的设计与实现

数字集成系统设计最常用的方法：首先用 VHDL 语言进行高层次描述，用 CPLD、FPGA 等可编程逻辑器件支持的逻辑综合工具综合得到代码，写入相关的 CPLD、FPGA 芯片，马上可进入 PCB 形成产品。与全定制的 ASIC 比较，CPLD、FPGA 的显著优势是开发周期短，投资风险小，产品上市速度快，市场适应能力强和软硬件升级余地大，且当产品定型扩产时，可将生产中经充分验证的 VHDL 设计迅速实现 ASIC 投产。实用集成电路设计实现流程如图 1-3 所示。

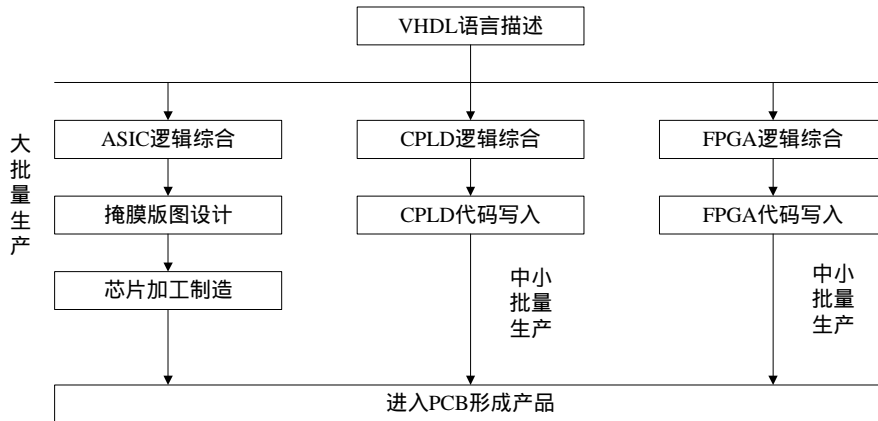


图 1-3 实用集成电路设计实现流程图

数字集成系统的产品设计同样可采用 VHDL 语言进行描述和逻辑综合，前两个流程与集成电路设计十分相似，只是电子产品需要布版成 PCB 版图，最后经安装焊接加工成产品。这就是第三代设计自动化工具称为电子设计自动化（EDA），而不是集成电路设计自动化的道理。电子产品与集成电路的高层次设计与综合已经统一，只是版图设计不一样，这与两者的加工工艺完全不同有关。电子产品的设计流程如图 1-4 所示。

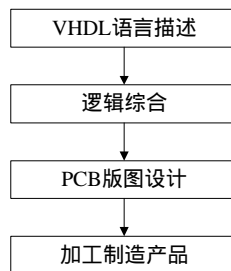


图 1-4 电子产品设计流程

## 第 2 章 VHDL 语言设计基础

在进行 VHDL 语言程序设计前，首先要熟悉对象、数据类型、运算操作符、文字使用规则等基础知识。本章还介绍了 VHDL 语言的总体程序结构、实体语句、构造体语句的使用规则以及 VHDL 语言的几种常用描述方法与逻辑综合，使读者了解 VHDL 语言的总体概貌。

### 2.1 VHDL 语言的由来与特点

硬件描述语言 HDL (Hardwave Description Language) 是由美国 Iverson 于 1962 年提出的，自此以后，各个公司、高等院校、科研单位开发了众多的专用 HDL 语言，仅高校及科研单位便有上百种之多。20 世纪 80 年代初，美国国防部为发展其超高速集成电路 VHSIC (Very High Speed Integrated Circuit) 计划，委托很多公司及个人进行设计。因各自使用的设计语言不同，使承包人的成果不能再利用，信息交换相当困难。为解决这一问题，1983 年提出硬件描述语言 VHDL (VHSIC HDL) 的标准化问题。1987 年 12 月，国际电气与电子工程师协会 (IEEE, Institute of Electrical and Electronics Engineers) 组织批准了 VHDL 语言为硬件描述语言的工业标准，即 IEEE STD 1076-1987 (LRM87)。以后根据集成电路技术发展的需要，于 1993 年在 87 版本的基础上进行了若干修订，增加一些功能，成为 IEEE STD 1076-1993 (LRM93) 版本。VHDL 93 版与 VHDL 87 版不完全兼容，93 版本增加了一些运算符及保留字，并删去某些属性。VHDL 87 版本的源程序需作相应的修改，才能在 VHDL 93 版本上通过编译。1999 年国际 IEEE 标准协会批准了数字模拟混合 VHDL 版本 VHDL-AMS (Analog and Mixed-Signal) 作为工业标准，即 IEEE STD 1076-1999 版，VHDL 语言有较强的高层次抽象描述能力，还有一种工业标准叫 Verilog HDL 硬件描述语言，在电路级、开关级和寄存器传输有精确的描述能力，这种语言是国外集成电路设计中使用最广泛的硬件描述语言，但国内设计人员使用 VHDL 语言为多。

VHDL 语言是一种大型硬件描述语言，软件的售价昂贵。在短时间全面掌握其编程技术是困难的。首先学会一些常用主要语句的编程方法，用以设计较简单的集成电路，全面掌握 VHDL 语言的编程技术要有一个融会贯通、熟能生巧的过程。学习 VHDL 语言编程最好有 C 语言的基础，需要有较好的数字电路和其他硬件电路基础，还要有数值计算方法等基础，学习过程需要掌握 VHDL 语言的结构特点。VHDL 语言有以下特点：

- VHDL 语言以行为描述见长，能从较抽象的角度描述电子实体的功能，叫行为描述。
- VHDL 语言是结构化语言，因而能从比较具体的角度描述电子实体的结构，叫结构描述。
- VHDL 语言具备从比较抽象到比较具体的多层面电子实体的混合描述能力，叫混合描述。
- VHDL 语言用于描述电子实体的程序既能综合，又能模拟仿真。综合：把抽象的描述自动转化为逻辑图、网表文件；模拟仿真：用软件便可验证设计的正确性。

- 通用性好，支持面广。VHDL 语言是工业标准，凡大型 CAD 软件都支持 VHDL 语言的设计环境，因此用 VHDL 编程的设计文件可通行于多种不同的设计工具。
- 重复使用性好。VHDL 语言的描述与具体生产工艺无关，变换不同的工作库便可适应不同的生产工艺。只要设计正确，生产工艺进步了，采用新的工作库，重新布版布图，便可继续使用。
- 可读性好。VHDL 语言用一种高级语言描述电子实体，容易理解。VHDL 语言被认为既是设计实现，又是技术说明，设计与说明集于一体。

VHDL 语言是一种硬件描述语言，其描述与其他高级语言最大的不同点是与硬件电路密切相关，学习 VHDL 语言要注意掌握其共性，更要突出其个性。

VHDL 语言的并行性要求设计人员摆脱一维的思维方式。VHDL 语言是一种并行语言，其语言要素及设计概念最早是从美国军用计算机语言 ADA 发展而来的。虽然 VHDL 语言的语句格式与 C 语言有很多相同之处，但其编程思想与传统的顺序执行语言有很大区别，必须适应实际硬件电路的工作方式，用并行或顺序多种语句方式描述在同一时刻中所有可能发生的事件。VHDL 语言具有描述由相关和不相关的多维时空组合的复合系统功能，要求设计人员以多维并发的思路完成 VHDL 语言的程序设计。

VHDL 语言与传统电路设计语言，如 ABEL 语言的设计思想也有很大的区别。ABEL 语言描述的是电路本身，包括电路的连接关系，设计工作量大而繁杂。而 VHDL 语言描述是电路的行为或结构，设计简单明了，工作量小，两种语言的语句规则也不相同。

使用 VHDL 语言进行集成电路设计时，要十分注意节约硬件资源。设计时芯片硬件资源的节省，不但使芯片的面积更小，功耗更低，且大大降低成本，这是集成电路设计追求的主要目标之一。一条不恰当的语句，一个不恰当的算法，一个不恰当的数值范围，一项本可省去的操作，都有可能使逻辑综合后生成的硬件电路更加复杂，占用硬件资源大幅上升。这与设计者的人为因素有关，即与设计人员编程的熟练和优化程度有关。VHDL 语言进行逻辑综合后，具有非常先进的逻辑优化功能，使 VHDL 语言设计尽可能少占硬件资源，每一设计的硬件资源占用情况，即标准门数等可直接从逻辑综合的适配报告中获得，方便设计人员修改设计方案。

## 2.2 VHDL 语言的基础知识

VHDL 语言的基础知识与其他高级语言有许多相同之处，但因 VHDL 语言是一种描述硬件电路的语言，故其基础知识也有自己的特点，如 VHDL 语言中信号是一种特有的对象。VHDL 语言还可以由设计人员自己定义数据类型，这一点也是其他高级语言做不到的。

### 2.2.1 对象 (OBJECT)

在 VHDL 语言中，凡是可以赋予一个值的客体叫对象。对象也可认为是数值的载体，VHDL87 版只定义三个对象：常量、变量、信号，VHDL93 版新增加文件这个对象，因为文件也是大量数据的载体。

#### 1. 常量 (CONSTANT)

常量是被赋予一个固定值的量。在程序中常量只能进行一次赋值，赋值通常在程序开始