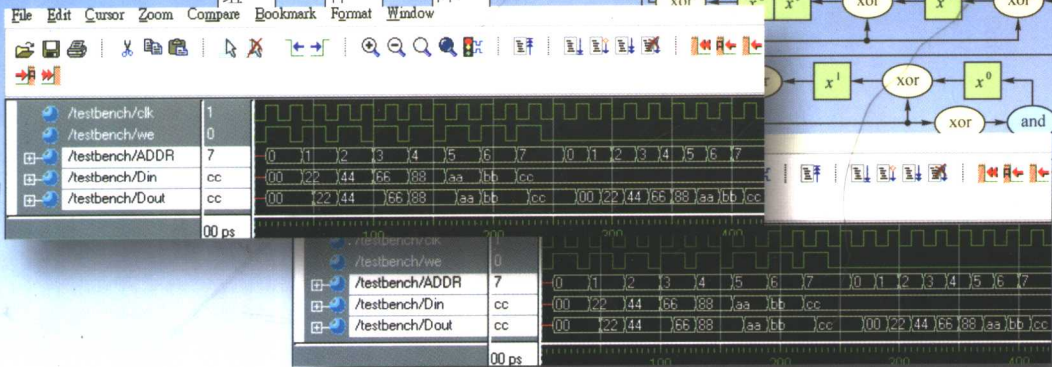
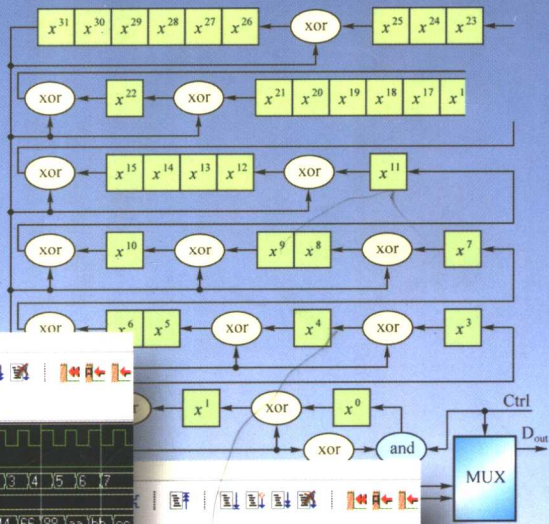
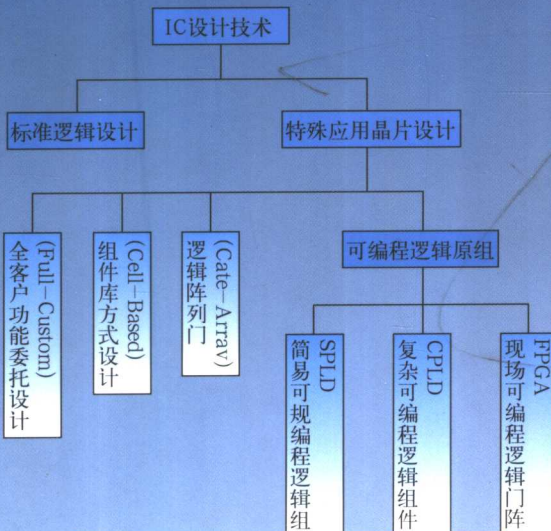


Verilog in FPGA Chip Design  
Verilog in FPGA Chip Design

# Verilog FPGA 芯片设计

林灶生 刘绍汉 编著



TP312  
2061D

# Verilog FPGA 芯片设计

林灶生 刘绍汉 编著

北京航空航天大学出版社

## 内 容 简 介

除讲述基本的设计技巧外,还深入介绍了多模块整合设计技术,适合各层次设计者参考使用。内容包括:数字逻辑设计与 Verilog 发展历史,Verilog 设计风格与概念,Verilog 设计结构,门级描述,数据流建模,行为描述,函数及任务,UDP 逻辑电路与状态机,Verilog 程序设计技巧,电路延时时序的设定及实用专题设计范例等。

本书适于大专院校电子类及计算机工程类本科及研究生,以及相关领域的工程设计人员使用。

### 图书在版编目(CIP)数据

Verilog FPGA 芯片设计/林灶生等编著. —北京:北京  
航空航天大学出版社,2006.7  
ISBN 7-81077-739-4

I. V… II. 林… III. ①硬件描述语言—程序设计  
②现场可编程门阵列—系统设计 IV. ①TP312②  
TP332.1

中国版本图书馆 CIP 数据核字(2006)第 078981 号

本书中文简体字版由台湾全华科技图书股份有限公司独家授权。仅限于中国大陆地区出版发行,不含台湾、香港、澳门。

© 2006,北京航空航天大学出版社,版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制或传播本书及其所附光盘内容。

侵权必究。

北京市版权局著作权全国登记号图字:01-2005-3680

### Verilog FPGA 芯片设计

杜灶生 刘绍汉 编著

责任编辑 李 键

责任校对 陈 坤

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:787×1092 1/16 印张:18.25 字数:467 千字

2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-739-4 定价:35.00 元(含光盘 1 张)

# 序 言

---

完成了《VHDL 芯片设计》一书后,我们深感硬件描述语言在电子工程及信息工程领域的重要性。一方面 Verilog 语言是一种非常流行的硬件描述语言,而市面上 Verilog 语言相关书籍屈指可数;另一方面我们在开设 Verilog 语言相关课程过程中,建立了相当多的模块程序,想将它们汇集成册。这是我们再次撰写一本有关 Verilog 语言书籍的动机。

本书除讲述基本的设计技巧外,还深入介绍了多模块整合设计技术,适合各层次设计者参考使用。希望这本书能带领读者进入以 Verilog 语言为主的各种相关设计领域中,让读者熟悉 Verilog 语言的全貌。更希望它能帮助读者完成各种芯片的设计。

本书根据 Verilog 语言的特性共分为 11 章——

- 第 1 章:详细介绍逻辑电路设计方法的进化过程、相关设计技术及其优缺点。此外,说明在“硅导计划”下,中国台湾学术界的策略联盟规划,使芯片设计课程在中国台湾学术界扎下坚实的根基。
- 第 2 章:深入讲解 Verilog 语言的设计风格。当设计一个大系统时,需由很多不同专长的团队分别来完成,每个团队设计完成的子系统之间信号的定义与沟通十分重要。尤其应注意当功能修改时,电路中增加了何种功能,电路作了何种修正,是由谁修改的,何时修改的。这些资料都会影响系统的设计成本。因此,如何设计出可读性、简单性、区域性、可移植性、重复使用性及重复规划性都优良的芯片,将是本章的重点。
- 第 3 章:解释 Verilog 语言设计结构。对由上而下(Top-down)及由下而上(Bottom-up)两种设计方法,说明其不同点。此外,还说明了行为模式、数据流模式及逻辑门级模式三者的差异;接着有针对性地对组成 Verilog 程序的不同模块结构详加说明;最后详细介绍了依照顺序式(In-order)及名称对应式(By-name)两种方式调用实例组件的不同点。
- 第 4 章:介绍逻辑门级的描述方法,并以译码器、比较器、多路输入选择器……范例说明如何用基本逻辑组件组成并设计相关的逻辑电路。
- 第 5 章:介绍数据流的描述方法,利用常用的组合逻辑电路说明其设计技巧。此外,针对表达式及布尔代数中所使用到的操作符,详细说明它们的使用方法。
- 第 6 章:介绍行为描述方法,说明如何以算法的方式去描述硬件的功能。诸如时钟信号的检查方式、语句区块的使用方式、if…else 结构、case…endcase 结构、循环的使

用方式……最后配合不同实例说明此描述方式的设计技巧。

第7章：介绍函数及任务的使用方法，并说明设计电路时须使用函数及任务的原因。一般函数及任务分为使用者定义及系统内建方式，文中以不同的范例说明使用者定义的函数及任务的不同点。最后以相当多的篇幅说明系统函数及任务的使用方法。

第8章：说明用户自定义原语(UDP)逻辑电路与状态机的设计方法。本章分为两部分加以介绍：首先谈如何以真值表的方式规划出使用者自定逻辑；接着介绍状态机中 Moore 状态机与 Mealy 状态机的不同点，并说明以 Binary code, Gray code 及 Hot code 方式产生状态的差异性。

第9章：就程序设计技巧列举设计法则加以说明。这些设计法则往往决定该程序设计流程花费的时间以及硬件执行效率。

第10章：主要说明电路延迟时序的设定方法。时序的设定与验证仿真在芯片设计流程中是不可或缺的步骤。本章将提供在仿真及验证过程中电路延迟时序设定的技巧，让设计者在时序仿真及验证时，能够快速完成其仿真验证程序的撰写。

第11章：在上述各章叙述完以 Verilog 硬件描述语言设计芯片的技巧后，本章将以较复杂的电路设计作实务专题范例应用。这些电路包括各种乘法器、除法器、CRC 电路……让设计者借助这些实例，应用于其它电路的设计。

本书所有范例均经过测试验证。希望本书各章所叙述的重点能为读者在使用 Verilog 语言设计芯片的过程中，提供重要的参考依据。

刘绍汉 谨识  
林灶生  
2006年2月

# 如何使用光盘

---

本书所有相关范例均经过综合测试,并以章节为主,储存于光盘内以章节为名的文件夹中。在各章节的文件夹中,每个子文件夹名称及其内部的 Project 名称均与该章节范例的(顶层)模块名称相同。读者可以以章节为主,以范例模块名称为对象,很轻易地就可在光盘中找到所要的范例文件。

在每一模块文件夹中可能仅有 Verilog 源代码程序(\*.v),亦可能包含仿真测试程序(\*.tf)。它们的使用方式为:

## ① XILINX 系统

- ▶ 先进入 iSE Project Navigator 系统中,建立新项目并选择一组件。
- ▶ 在选项中选择 Project→“Add Copy of Source...”,将光盘项目文件夹中所有文件加入目前的项目中。
- ▶ 接着进行综合并作仿真。

## ② 其它系统

由于系统不同,读者在复制文档时,仅能将项目文件夹中的源代码程序(\*.v)粘贴至所属系统中,稍加修正即可使用。



# 录

## 第 1 章 数字逻辑设计与 Verilog 的发展过程

1.1 计算机辅助设计与集成电路产业 .....	1
1.2 硬件描述语言 .....	3
1.3 集成电路芯片设计流程 .....	3

## 第 2 章 Verilog 设计风格与概念

2.1 设计风格 .....	5
2.1.1 设计者与修改原因 .....	5
2.1.2 设计代码的注释和说明 .....	5
2.1.3 设计过程注意事项 .....	6
2.2 基本概念 .....	7
2.2.1 操作符 .....	7
2.2.2 注释说明与空白 .....	7
2.2.3 数 字 .....	8
2.2.4 字符串 .....	9
2.2.5 标识符、关键词及系统函数 .....	10
2.2.6 信息显示于标准的输出 .....	12
2.2.7 仿真监视 .....	13
2.2.8 结束仿真 .....	14
参考文献 .....	15

## 第 3 章 Verilog 设计结构

3.1 设计方法 .....	17
3.2 模 块 .....	17
3.3 端 口 .....	19
3.4 模块实例的引用 .....	20
3.5 数据类型 .....	21
3.5.1 数值集合 .....	21
3.5.2 线 网 .....	22
3.5.3 寄存器 .....	23
3.5.4 向 量 .....	23
3.5.5 数 字 .....	24
3.5.6 参 数 .....	24
3.5.7 数组与内存 .....	25

3.5.8 三态	25
参考文献	26
<b>第4章 门级描述</b>	
4.1 and,or,nand,nor,xor 及 xnor 门	27
4.2 buf 及 not 门	29
4.3 实例说明	30
4.4 多路输入选择器实例	40
习 题	45
参考文献	45
<b>第5章 数据流建模</b>	
5.1 连续赋值	46
5.2 表达式	47
5.2.1 常数值表达式	47
5.2.2 操作数	47
5.2.3 操作符	48
5.3 应用实例	60
5.3.1 3-8 译码器	60
5.3.2 4-2 编码器	62
5.3.3 4 位加法器	64
5.3.4 1-4 多路输出选择器	65
5.3.5 3 位多数位表决器	67
5.3.6 1 位全减器	68
习 题	70
参考文献	70
<b>第6章 行为描述</b>	
6.1 过程性结构	71
6.1.1 initial 区块	71
6.1.2 always 区块	71
6.2 过程性赋值	76
6.2.1 阻塞过程性赋值	76
6.2.2 非阻塞过程性赋值	79
6.3 begin...end 块语句	82
6.4 if 语句	82
6.5 case 语句	93
6.6 casez 语句	99
6.7 casex 语句	102
6.8 循 环	105
6.8.1 for 循环	105
6.8.2 while 循环	115



6.8.3 forever 循环 .....	116
6.8.4 repeat 循环 .....	116
6.9 begin...end 区块语句 .....	119
6.10 fork...join 区块语句 .....	120
6.11 wait 语句 .....	121
6.12 实用范例 .....	122
6.12.1 两位数 BCD 计数器 .....	122
6.12.2 BCD 码加法器 .....	125
6.12.3 16 字节双端口 RAM .....	129
6.12.4 16 字节单端口 RAM .....	132
习 题 .....	135
参考文献 .....	135
<b>第 7 章 函数及任务</b>	
7.1 函 数 .....	136
7.2 任 务 .....	143
7.3 函数调用函数 .....	147
7.4 任务调用函数及任务 .....	149
7.5 系统函数与任务 .....	151
7.5.1 与实数有关的系统函数与任务 .....	151
7.5.2 显示与写入系统任务 .....	152
7.5.3 驱动器计数系统函数 .....	152
7.5.4 文件输出系统任务 .....	153
7.5.5 结束执行系统任务 .....	154
7.5.6 时序检查系统任务 .....	154
7.5.7 测试信号系统任务 .....	155
7.5.8 加载内存系统任务 .....	155
7.5.9 时间刻度系统函数与任务 .....	156
7.5.10 储存与重新激活系统任务 .....	158
7.5.11 复位系统任务 .....	158
7.5.12 其它系统任务与函数 .....	158
习 题 .....	160
参考文献 .....	160
<b>第 8 章 用户定义的原语逻辑电路与状态机</b>	
8.1 UDP 的设计方法 .....	162
8.2 用 UDP 描述组合逻辑电路 .....	163
8.3 用 UDP 描述时序逻辑电路 .....	166
8.4 状态机 .....	169
8.4.1 Moore 状态机 .....	169
8.4.2 Mealy 状态机 .....	181

习 题	199
参考文献	200
<b>第 9 章 Verilog 程序设计技巧</b>	
9.1 程序设计	201
9.1.1 如何设计综合电路	201
9.1.2 数学表达式的顺序及群集	202
9.1.3 if 语句与 case 语句的比较	203
9.2 编译器预处理命令	203
9.2.1 <code>\include</code>	204
9.2.2 <code>\define</code> 与 <code>\undef</code>	204
9.2.3 <code>\timescale</code>	205
9.2.4 <code>\resetall</code>	206
9.2.5 <code>\ifdef</code> , <code>\else</code> 与 <code>\endif</code>	206
参考文献	207
<b>第 10 章 电路延迟时序的设定</b>	
10.1 逻辑门延迟	208
10.2 连接线延迟	210
10.3 模块路径延迟	210
10.3.1 特定区块	210
10.3.2 特定参数	213
10.4 跳变沿敏感的路径延迟	214
10.5 状态相关路径延迟	216
10.6 延迟时间值设定	218
习 题	219
参考文献	220
<b>第 11 章 实用专题设计范例</b>	
11.1 0~9999 十进制计数器	221
11.2 16 位移位式乘法器	227
11.3 16 位固定点式乘法器	230
11.4 16 位布斯乘法器	234
11.5 16 位移位式除法器	237
11.6 16 位重存与非重存除法器	242
11.7 移动蛇控制电路	248
11.8 键盘控制电路	257
11.9 循环冗余校验电路	264
11.10 浮点数运算器电路	271
11.10.1 浮点数加法器	272
11.10.2 浮点数乘法器	277
参考文献	282

# 第 1 章

## 数字逻辑设计与 Verilog 的发展过程

### 1.1 计算机辅助设计与集成电路产业

数字逻辑设计在过去几年中飞速发展,由早期真空管、晶体管、集成电路到现在的系统芯片(System on a Chip, SoC),体积越作越小,功率消耗越来越低;但芯片由小型集成电路(Small Scale Integration, SSI)逐渐演化至中型集成电路(Medium Scale Integration, MSI)、大规模集成电路(Large Scale Integration, LSI)、超大规模集成电路(Very Large Scale Integration, VLSI)直至特超大规模集成电路(Ultra Large Scale Integration, ULSI),芯片电路的复杂度越来越高。与此同时,IC 制造技术也突飞猛进,由早期的 6 英寸晶圆片、8 英寸晶圆片到现在的 12 英寸晶圆片,每一晶圆片上可制造的芯片越来越多。工艺更由  $0.5\ \mu\text{m}$ ,  $0.25\ \mu\text{m}$ ,  $0.18\ \mu\text{m}$ ,  $0.13\ \mu\text{m}$  进入到纳米( $90\ \text{nm}=0.09\ \mu\text{m}$ )时代。技术进步使单位面积上可容纳更多的电路,从而使传统的设计方法不能满足设计者的要求,于是电子设计自动化(Electronic Design Automation, EDA)平台应运而生。往往须借助快速高效的设计平台才能应付超 10 万个晶体管的 VLSI/ULSI 日益复杂的电路芯片的设计。

近几年来,随着 IC 设计与制造技术日新月异的进步,产生了各种类型的集成电路芯片设计技术。图 1-1 列出了这些类型的设计技术。

就上述设计技术类型,分述如下:

#### ■ 全客户功能委托设计(Full-Custom Design)

此类设计技术,电路硅片面积的利用率最高,但须在晶体管级别上进行逻辑优化和人工布线,最终用晶体管完成电路的设计。采用此设计方法,费时费力,因此耗费成本高,但设计密度和速度最高,功率消耗最低。

#### ■ 组件库方式设计(Cell-Based Design)

此类设计技术属于高复杂度、大容量的设计方法,基本上每个组件均可由已设计完成的组件库中取得,且以硬件描述语言方式(Hardware Description Language, HDL)描述并完成设计,而且用自动布线方式实现具体的物理电路设计。设计流程较全客户功能委托设计方式简单,但完成的芯片,其效能较全客户功能委托设计差。

#### ■ 可编程逻辑组件(Programmable Logic Device)

设计方式同样以硬件语言描述式完成设计。其优点是设计流程简单,可重复编程,且无须制作掩膜,在实验室即可完成样机设计与验证,达到快速上市的目的;缺点则在于芯片昂贵,执

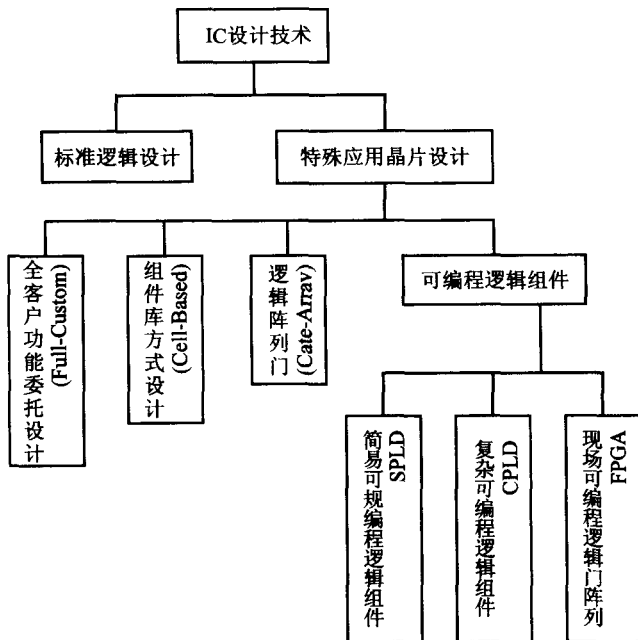


图 1-1 集成电路设计分类图

行效能较差,只有当产品成熟时才可通过设计平台转成组件库设计方式。目前此可编程逻辑芯片是集成电路设计公司前期设计不可或缺的设计工具。可编程逻辑组件有以下3种:①简易可编程逻辑组件(Simple Programable Logic Device, SPLD)属于小型的PLD,市面上PEEL18CV8即是其中之一;②复杂可编程逻辑组件(Complex Programmable Device, CPLD)则属于中型密度的PLD;③现场可编程门阵列(Field Programmable Gate Array, FPGA)则是高密度的PLD。一般而言,SPLD及CPLD均以EPROM,EEPROM及Flash结构呈现;而FPGA则常以SRAM形式呈现。

近二三年来,为了加入WTO及提升经济实力,中国台湾倡导了“硅导(Si-Soft)计划”,其中一项子计划即为系统芯片(SoC)计划。在此计划下,中国台湾产业界与学术界针对集成电路设计产业不同的领域,规划了6大领域策略联盟:

- ① 系统芯片策略联盟(System on a Chip, SoC);
- ② 先进技术联盟(Advanced Technology Consortium, ATV);
- ③ 自动化电子设计联盟(Electronic Design Automation, EDA);
- ④ 混合信号式集成电路设计联盟(Mixed-Signal IC Design Consortium, MSD);
- ⑤ 数字智产联盟(Digital Intellecture Properties Consortium, DIP);
- ⑥ 快速样机与布线联盟(Prot Typing & Layout, P&L)。

在这些联盟推动下,中国台湾各大学、技术学院在集成电路设计与制作相关课程规划上作了相当大的改革,IC设计领域的课程更是如雨后春笋般出现于各大学院中。希望这些发展能为中国IC设计产业注入新的活力。

## 1.2 硬件描述语言

长久以来,计算机程序设计语言(如 C 语言、FORTRAN 语言、BASIC 语言)均以顺序方式由上而下逐一执行指令。欲以此方式描述具有并行处理或并发(Concurrent)硬件动作的特性,传统计算机语言根本无法实现。因此,一种全新的、针对硬件并发特性而设计的硬件描述语言应运而生。一般而言,硬件描述语言须具有以下特性:

- ▶ 具有共享性;
- ▶ 好的设计方法及技术支持;
- ▶ 与技术及工艺无关;
- ▶ 广泛的描述能力;
- ▶ 具有设计的互换性;
- ▶ 利于大型 IC 设计及设计的重复使用。

目前业界使用较为普遍的硬件描述语言包括 VHDL 语言(请参考作者于全华书局出版的另一本著作——《VHDL 芯片设计》)与 Verilog 语言。本书主要以 Verilog 语言为设计工具。

Verilog 语言于 1983 年初创于 Gateway Design Automation 公司。其语法易懂易学,与流行的 PASCAL 语言及 C 语言相当类似。Verilog 语言可以描述内建组件、UDP(用户自定义原语)组件、用户设计的模块及时序测试模块等,以此来仿真实际电路;可以在综合之前完成仿真与测试。硬件描述语言初创时,只能用于数字系统的验证,而电路结构的设计必须用人工的方法绘制。直到 1980 年以后,由于设计方法的改进,数字逻辑电路才真正可使用缓存器传输级(Register Transfer Level, RTL)方式以硬件描述语言去描述它们。因综合器(Synthesize)尚未成熟,设计者仍须清楚地描述缓存器间数据的流动及数据的处理方式。只有综合工具才能把由 RTL 语法描述的模块自动地转换成逻辑部件及其模块之间详尽的连接关系,如此才能完成芯片的设计。因此综合器的开发,使硬件描述语言设计数字电路向前迈进一大步。而全新的集成电路芯片设计开发平台更是将语言描述、电路综合、电路仿真测试与验证、硬件实现与硬件电路下载及仿真测试全部整合为单一系统,如 XILINX 和 Altera 公司开发的 FPGA 芯片开发平台。开发工具平台的技术发展真是不可同日而语。本书所举的范例均以 XILINX 公司所开发的 iSE 6.1 完成设计,仿真则以 Model Technology 公司的 ModelSim 完成。

## 1.3 集成电路芯片设计流程

一般以硬件描述语言开发芯片的设计流程可以用图 1-2 表示。芯片设计前先制定系统规格,此时无须考虑其硬件实际的规划;接着以软件作仿真以确定其系统功能的可能性;然后再对硬件规格如功能、接口、信号标准电平及整体系统结构作功能性说明。

接着可用 HDL 方式去描述硬件结构的功能、效能及相关高层次行为以分析其功能。而 RTL 描述则进入正式的硬件设计阶段,设计者据此描述可实现硬件电路的数据流动模式,依同样的 HDL 语法功能作仿真及测试。

而综合工具则将 RTL 描述转换成门级网表(Gate Level Netlist),这个网表的功能是描述电路中逻辑门及连接线之间的关系。此时可再深入地进行时序仿真与测试,以提高电路的运

行速度。在开始执行逻辑组件的综合布局之前,必须先定义出整体设计中包括宏组件位置、电源/接地的布线和芯片引脚,才可布局区域的平面架构(Floor Plane),以规划出芯片区块的摆设位置,作为自动布线的依据。而门级网表可由自动布局与布线工具转换成实际芯片内部的布线,然后由人工布线修正后完成芯片内部最后的布线设计。

近几年来,设计已提升到系统设计的层次,即为综合工具可直接将行为描述转换成 RTL 描述,以缩短设计时间,从而降低相关设计成本。

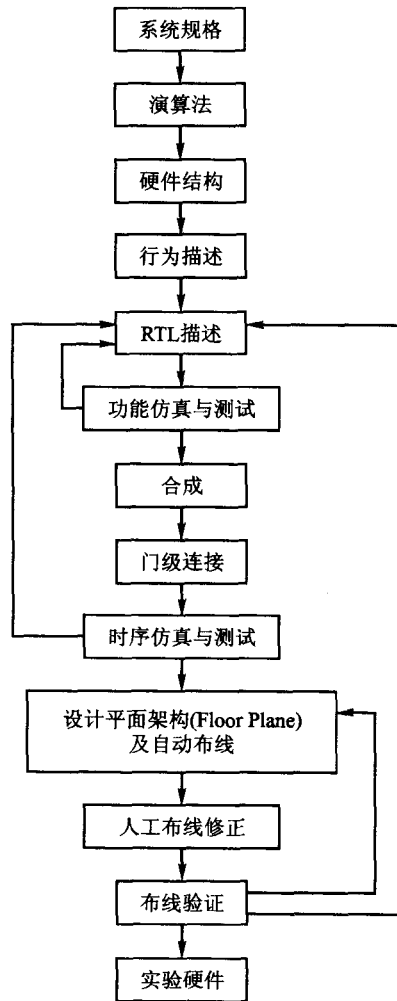


图 1-2 芯片设计流程图

# 第 2 章

## Verilog 设计风格与概念

### 2.1 设计风格

与一般计算机程序设计一样,硬件描述语言在设计时,其可读性如何,往往取决于设计技巧的优劣。尤其一个大的系统,须由很多不同专长的团队分别来完成。则每一个团队所完成的子系统间信号的定义与沟通益显重要,尤其应注意在功能修改时,电路中增加了何种功能,修正了何种电路,由谁修改的,是何时修改的。此外,设计者相关资料、版本演变、参数说明、复位与时钟规格、测试结构、电路时序与异步接口等等都会影响系统的设计成本。因此,Verilog 语言的可读性、简单性、区域性、可移植性、重复使用性及重复规划特性均是本书考虑的重点。本书就把设计风格置于前面,让读者了解其重要性,为了节省篇幅,在后面章节的程序中,将其去除。

#### 2.1.1 设计者与修改原因

当完成一系统的 HDL 设计时,必须将设计者相关资料置入程序中。其中包括规格修改及其原因,日期亦须标识。其表格如下:

作者	设计/修改	日期	联络方式
林灶生	MPU 的 Verilog 程序第 1 版	22-03-2003	jslin@chinyi.ncit.edu.tw
刘绍汉	测试/验证程序修正	28-03-2003	liou@chinyi.ncit.edu.tw

#### 2.1.2 设计代码的注释和说明

一个优秀的 Verilog 程序,应具有相当好的可读性。因此,在程序文件中须有详细的注释说明,让每个设计者(包括自己)很快就能了解其设计内容。依业界经验,一个系统的 HDL 程序须有 30% 以上的注释说明,才是一个好的设计习惯。在这些注释中,程序表头说明占了很重要的地位。一般而言,所有的程序都必须有以下表头:

```
//-----  
// Title: RS-232C IP Core 设计  
// Project: FPGA 教学仿真系统项目
```

```
// File:RS-232C.V
// Author: 林灶生 jslin@chinyi.ncit.edu.tw
// Organization: 国立勤益技术学院 资电所
// Last update: 08-04-2003
// Platform: XILINX iSE 6.1
// Simulators: Moldelsim
// Targets: XILINX Spartan II XC2S200-5PQ208
// Description: 本 IP Core 主要用于设计具有沟通能力且拥有-FIFO 的
//             RS-232C
//-----
// Copyright(C)notice 版权归属国立勤益技术学院资电所所有
// Revision: R5
// Revision Number: II
// Version: V2.0
// Date: 28-06-2003
// Modifier: 刘绍汉 <liou@chinyi.ncit.edu.tw>
// Description: 于 RS-232C 中加入-FIFO 以增强芯片与计算机间沟通的效能
//-----
```

### 2.1.3 设计过程注意事项

一个电路的执行效能,往往取决于设计者能否灵活使用硬件描述语言。在本小节中,列举了设计时应该注意的部分事项。

- ▶ 最好保持一行指令一行说明。
- ▶ 每一行指令不可太长,最好保持在 70~80 个字符。
- ▶ 输入/输出信号的安排次序应为:先输入(Input)再输出(Output);信号排列的次序则为时钟(Clock)、复位(Reset)、使能(Enable)、其它控制信号、地址总线(Address Bus);最后才是数据总线(Data Bus)。
- ▶ 实例组件(Instantiation)引用时,其对应信号的连接,最好选择按名称(By-name)对应方式,而不要使用按顺序(In-order)方式。
- ▶ 使用参数时,以大写表示常数,以小写表示连接信号变量。
- ▶ 总线位宽的声明,按习惯由大至小排列(例:[15:0]),而不是由小到大排列(例:[0:15])。
- ▶ 表达式中所使用的数据应为固定位宽。
- ▶ 使用语句描述电路时,必须了解其是否可综合。
- ▶ 使用内建组件时,务必使用通用型组件,避开与机器相关(Machine Dependent)的组件。
- ▶ 避免使用异步逻辑。
- ▶ 电路中的时序逻辑电路,避免混合使用正沿与负沿触发器。
- ▶ 同一变量,不要做重复的赋值。
- ▶ 将设计完成的各种 IP 电路存于组件库中,建立重复使用意识。





## 2.2 基本概念

Verilog 代码与一般计算机程序一样,其源文件由一连串记号(Tokens)和字符构成的语句组成。

这些记号和字符包括:

- ▶ 操作符(Operator);
- ▶ 注释说明(Comment);
- ▶ 空白(White Space);
- ▶ 数字(Number);
- ▶ 字符串(String);
- ▶ 标识符(Identifier);
- ▶ 关键词(Keyword)。

有关这些记号和字符的用法将在以下各节逐一说明。

### 2.2.1 操作符

操作符的类型包括单目操作符(Unary)、双目操作符(Binary)及三目操作符。单目操作符将操作符置于操作数左侧;双目操作符是将操作符放于两个操作数中间;而三目操作符则具有两个分开的操作符,以分开三个独立的操作数。Verilog 语法中仅有一个三目操作数,即条件操作符。以下就是这些操作符的例子。

---

```

Y = ~X;           //“~”为单目操作符,X 为操作数
W = X | Y;       //“|”为双目操作符,X 及 Y 为操作数
X = W ? X : Y;   //“?”为三目操作符,W, Y 及 Z 均为操作数

```

---

### 2.2.2 注释说明与空白

注释说明的目的在于让程序格式更具有可读性,且易于归档。在 2.1.2 小节中,已使用过注释说明。一般注释说明可分为单行注释与多行注释。单行注释是由两条斜线(//)起始;而多行注释则以“/\*”开始,终止于“\*/”。当 EDA 工作平台在执行综合时,会自动跳过这些注释说明。详见下面范例:

---

```

W = X & Y;           //X 与 Y 作 logic and 后存入 W,单行注释
/* 这是一个合法的
                    多行注释说明 */

```

---

在 Verilog 语法中,空白(White space)包括空格(Blank space)tab、换行(New lines)及跳行(Form feeds)。除了用来分隔字符及字符串中的空格不被忽略外,其它的空格都将被 Verilog 综合器所忽略。