



可编程逻辑 器件设计 技术详解

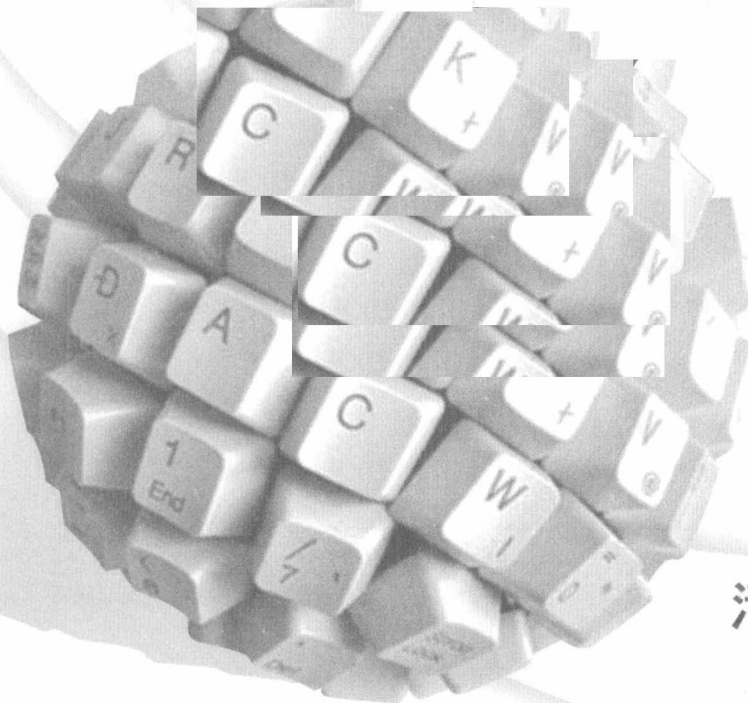
何 宾 编著



清华大学出版社



何 宾 编著



清华大学出版社
北 京

内 容 简 介

本书系统、全面地介绍了基于 Xilinx 可编程逻辑器件设计的方法、理论和应用。全书共分 11 章,内容包括:可编程逻辑器件设计指南、可编程逻辑器件设计方法、VHDL 高级设计技术、IP 核设计技术、基于 HDL 的设计输入、基于原理图的设计输入、设计综合和行为仿真、设计实现和时序仿真、设计下载、ChipScope Pro 调试工具、PlanAhead 工具及应用。本书参考了大量的最新的设计资料,内容新颖,理论与应用并重,书中介绍了 Xilinx 可编程逻辑器件的许多关键设计方法和设计技术,将这些设计方法和设计技术的介绍有机地贯穿于完整的设计流程中。

本书可作为从事 Xilinx 可编程逻辑器件设计工程技术人员的参考用书,以及电子信息类专业本科高年级学生和研究生教学和科研用书,同时也可以作为 Xilinx 公司的培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Xilinx 可编程逻辑器件设计技术详解 / 何宾编著. —北京:清华大学出版社, 2010. 3
ISBN 978-7-302-21071-9

I. X… II. 何… III. 可编程序逻辑器件—系统设计 IV. TP332.1

中国版本图书馆 CIP 数据核字(2009)第 168935 号

责任编辑:夏兆彦 林都嘉

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:190×260 印 张:21.25 字 数:526 千字

版 次:2010 年 3 月第 1 版 印 次:2010 年 3 月第 1 次印刷

印 数:1~5000

定 价:39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:033833-01

FOREWORD

前言

随着半导体技术的飞速发展,新电子产品上市周期的缩短,以及数字化处理技术的不断提高,大规模可编程逻辑器件 PLD 设计成为电子系统设计中一个重要的研究方向和应用领域。可编程逻辑器件越来越广泛地应用在通信、消费类、工业监控和医疗、广播、汽车、航空和军用等领域。摩尔定律的持续有效使芯片密度和性能持续提升,同时也在开发和生产复杂性方面带来巨大挑战。业内人士认为在发展传统集成电路 ASIC 和专用标准器件 ASSP 的同时,下一代设计中会越来越多地使用以 FPGA 为代表的 PLD 器件。

作为全球知名的可编程逻辑器件的生产厂商, Xilinx 公司以其卓越的软件和硬件产品性能,为中端和高端用户提供了很好的解决方案,拥有世界上 51% 的 PLD 市场份额。作为基于 Xilinx 产品进行设计的工程技术人员,只有系统深入地掌握其产品的使用方法、设计技巧,才能很好地发挥出 Xilinx 产品的性能优势。作者结合多年从事 Xilinx 可编程逻辑器件设计的经验和方法,并在参考大量 Xilin 技术资料的基础上编写了本书,目的是为基于 Xilinx 平台完成电子系统设计的工程技术人员提供一本全面系统介绍其设计方法和关键技术的参考用书,使他们能系统深入地掌握这些技术,从而更好地从事设计工作。

该书的编写是以典型的设计流程为主线,将设计流程所涉及的各种方法和技巧有机地贯穿于这条主线。同时该书也是对作者编写的《EDA 原理及应用》教学用书的重要补充。本书共分为 11 章,包括可编程逻辑器件设计流程导论、可编程逻辑器件设计方法、VHDL 高级设计技术、IP 核设计技术、基于 HDL 的设计输入、基于原理图的设计输入、设计综合和行为仿真、设计实现和时序仿真、设计下载、ChipScope Pro 调试工具及应用、PlanAhead 工具及应用。其各部分主要包括:

- (1) 第 1 章主要介绍了 Xilinx 的 CPLD 和 FPGA 的设计流程。
- (2) 第 2 章主要介绍了可编程逻辑器件的工艺, CPLD 和 FPGA 的结构特点, Xilinx CPLD、FPGA 和 PROM 产品的性能及特点。
- (3) 第 3 章主要介绍了 VHDL 语言的高级编程技巧,该部分内容包含了设计人员在使用 VHDL 语言进行设计时经常使用的设计技巧、设计方法和注意事项。
- (4) 第 4 章主要介绍了 IP 核设计技术,包括 IP 分类、IP 优化、IP 生成和 Xilinx IP 核应用技术。
- (5) 第 5 章主要介绍了基于 VHDL 语言和 Xilinx ISE 软件设计平台的设计输入实现过程。
- (6) 第 6 章主要介绍了基于原理图和 Xilinx ISE 软件设计平台的设计输入实现过程。

(7) 第7章主要介绍了设计综合和行为仿真,在该部分介绍了综合和仿真的概念, Xilinx ISE 综合和仿真参数的设置,综合和仿真的实现过程。

(8) 第8章主要介绍了设计实现和时序仿真,该部分介绍了实现的过程,时序仿真的概念, Xilinx ISE 软件实现过程的参数设置及一些高级设计工具的使用。

(9) 第9章主要介绍了设计下载,该部分介绍了 Xilinx 下载配置电路、下载文件的生成和下载的实现过程。

(10) 第10章主要介绍了 Xilinx 的在线逻辑分析仪调试工具 ChipScope Pro 的原理和使用方法。

(11) 第11章主要介绍了 Xilinx 的布局布线高级设计工具 PlanAhead 的原理和使用方法。

对于基于 Xilinx 芯片的片上可编程系统及数字信号处理的部分,作者将在清华大学出版社随后出版的两本书《片上可编程系统原理及应用》和《FPGA 数字信号处理原理及应用》中进行详细的介绍和说明。

参加本书编写工作的有李保敏、朱红林、左芳、王瑜。第1~3章由左芳整理资料,并负责编写部分内容。第4~6章由朱红林整理资料,并负责编写部分内容。第7~9章由李保敏整理资料,并负责编写部分内容。第10~11章由王瑜整理资料,并负责编写部分内容。全书由何宾统稿和定稿。

该书的编写得到 Xilinx 大学计划经理谢凯年博士的支持和帮助,正是由于他的支持和帮助才能使作者完成本书的编写工作。同时,也要感谢清华大学出版社的各位编辑和领导,也正是由于你们的辛勤工作,才能使该书高质量地出版。

因编写时间仓促,难免会有疏漏之处,欢迎读者批评指正,我们表示衷心感谢。

作 者

2009年5月于北京

CONTENTS

目 录

第1章 可编程逻辑器件设计流程导论	1
1.1 设计流程概述	1
1.2 设计输入和综合	3
1.2.1 层次化设计	4
1.2.2 原理图输入	4
1.2.3 HDL 输入和综合	5
1.3 设计实现	6
1.4 设计验证	8
1.4.1 仿真	9
1.4.2 静态时序分析	12
1.4.3 电路验证	13
1.5 FPGA 设计技巧概论	13
第2章 可编程逻辑器件设计方法	16
2.1 可编程逻辑器件基础	16
2.1.1 可编程逻辑器件概述	16
2.1.2 可编程逻辑器件的发展历史	17
2.2 PLD 芯片制造工艺	18
2.3 PLD 芯片结构	19
2.3.1 CPLD 原理及结构	19
2.3.2 FPGA 原理及结构	20
2.3.3 CPLD 和 FPGA 比较	24
2.3.4 PLD 选择原则	25
2.4 Xilinx 公司芯片简介	26
2.4.1 Xilinx CPLD 芯片介绍	26
2.4.2 Xilinx FPGA 芯片介绍	29
2.4.3 Xilinx PROM 芯片介绍	38
第3章 VHDL 高级设计技术	40
3.1 层次化设计	40
3.1.1 层次化设计的优缺点	40
3.1.2 在分层设计中使用综合工具	41
3.2 数据类型选择	42
3.2.1 使用 Std_logic (IEEE 1164)	42
3.2.2 声明端口	42
3.2.3 端口声明中的数组	43
3.3 混合语言设计	44

3.4	if 和 case 语句比较	45	3.19.2	例化块 SelectRAM 的编码 实例	81
3.4.1	if 语句设计描述	45	3.19.3	推断块 SelectRAM	83
3.4.2	case 语句设计描述	46	3.19.4	Virtex-4 和 Virtex-5 中的块 SelectRAM	83
3.4.3	避免出现锁存器	46	3.19.5	实现分布式 SelectRAM	85
3.5	逻辑结构设计	49	3.19.6	实现 ROMs	87
3.6	逻辑复制和复用技术	49	3.19.7	实现 FIFOs	89
3.6.1	逻辑复制技术	49	3.20	实现移位寄存器	89
3.6.2	逻辑复用(共享)技术	50	3.20.1	通用移位寄存器	89
3.7	控制信号	51	3.20.2	实现线性反馈移位寄存器 (LFSRs)	90
3.7.1	全局置位/复位(GSR)	52	3.21	实现多路复用器	91
3.7.2	使用时钟使能引脚代替门控时钟	53	3.22	并行和流水线技术	92
3.8	寄存器、锁存器、移位寄存器和 RAMs 的初始状态	54	3.22.1	并行设计技术	92
3.9	有限自动状态机设计	55	3.22.2	流水线设计技术	93
3.9.1	有限状态机原理	55	3.23	同步和异步单元处理技术	94
3.9.2	有限状态机分类	55	3.23.1	同步单元处理技术	94
3.9.3	有限状态机设计	58	3.23.2	异步单元处理技术	97
3.10	例化元件和 FPGA 原语	61	第 4 章	IP 核设计技术	99
3.10.1	例化 FPGA 原语	61	4.1	IP 核分类	99
3.10.2	例化核生成器模块	61	4.2	IP 核优化	100
3.11	属性和约束	62	4.3	IP 核生成	101
3.12	全局时钟缓冲	64	4.4	IP 核应用	101
3.13	高级时钟管理	66	4.4.1	数字时钟模块使用	101
3.14	专用的全局置位/复位资源	70	4.4.2	块 RAM 存储器使用	103
3.15	隐含编码	70	第 5 章	基于 HDL 的设计输入	107
3.16	输入和输出的实现	71	5.1	软件环境	107
3.17	IOB 寄存器和锁存器	72	5.2	综合工具介绍	109
3.18	实现操作符及产生模块	77	5.3	工程建立	109
3.18.1	DSP48 中运算符实现和模型 生成(Virtex-4 和 Virtex-5 器件)	77	5.4	设计描述	111
3.18.2	在乘法器中操作符实现和模型 生成	78	5.5	添加设计和检查	112
3.18.3	计数器中操作符实现和模型 生成	79	5.6	创建基于 HDL 的模块	113
3.18.4	比较器中操作符实现和模型 生成	80	5.7	IP 核产生和例化	114
3.18.5	编码器和解码器中操作符实现和 模型生成	80	5.7.1	IP 核的生成	115
3.19	存储器实现	80	5.7.2	IP 核的例化	116
3.19.1	块 RAM 实现	81	第 6 章	基于原理图的设计输入	118
			6.1	工程建立	118

6.2	设计描述	119	8.5	设计翻译	167
6.3	创建原理图模块	120	8.6	设计约束	168
6.3.1	原理图编辑器操作	120	8.6.1	设计约束概述	168
6.3.2	定义模块符号	121	8.6.2	时序约束	169
6.3.3	创建模块符号	124	8.6.3	引脚和面积约束	174
6.4	创建状态图模块	124	8.7	设计映射及时序分析	178
6.4.1	添加状态	125	8.7.1	设计映射	178
6.4.2	添加迁移	126	8.7.2	使用时序分析评估块延迟	180
6.4.3	添加行为	127	8.8	布局布线及验证	181
6.4.4	添加复位条件	127	8.8.1	布局布线流程	181
6.4.5	设计输出和添加	127	8.8.2	布局布线的实现	182
6.5	设计完成	128	8.8.3	布局布线验证	183
			8.8.4	布局后时序评估	184
			8.8.5	改变分区 HDL	185
第 7 章	设计综合和行为仿真	129	8.9	功耗分析	186
7.1	设计综合	129	8.9.1	启动功耗分析器	186
7.1.1	行为综合描述	129	8.9.2	XPower 的操作流程	187
7.1.2	基于 XST 的综合概述	130	8.9.3	简易的功耗分析方法	189
7.1.3	综合属性的设置	130	8.10	时序仿真实现	192
7.1.4	约束及设计综合的实现	139	8.10.1	时序仿真概述	192
7.1.5	RTL 符号的查看	139	8.10.2	使用 ModelSim 进行时序仿真	193
7.2	行为仿真的实现	140	8.10.3	使用 ISE 仿真器进行时序仿真	197
7.2.1	生成测试向量	140			
7.2.2	基于 Modelsim 行为仿真实现	143	第 9 章	设计下载	200
7.2.3	基于 ISE 行为仿真实现	147	9.1	可编程逻辑器件配置接口	200
第 8 章	设计实现和时序仿真	150	9.1.1	主串行模式	201
8.1	实现过程概述及约束	150	9.1.2	主 SPI 模式	202
8.1.1	实现过程概述	150	9.1.3	主 BPI 模式	204
8.1.2	建立约束文件	150	9.1.4	主并行模式	205
8.2	实现属性参数设置	151	9.1.5	从并行模式	205
8.2.1	实现属性设置步骤	151	9.1.6	从串行模式	205
8.2.2	翻译属性选项	152	9.1.7	JTAG 配置模式	206
8.2.3	映射属性选项	153	9.2	配置属性	208
8.2.4	布局布线属性	158	9.2.1	通用选项	208
8.2.5	映射后静态时序报告属性	161	9.2.2	配置选项	209
8.2.6	布局布线后静态时序报告属性	161	9.2.3	启动选项	211
8.2.7	仿真模型属性	163	9.2.4	回读选项	212
8.2.8	Xplorer 属性	165	9.3	创建配置数据	213
8.3	创建分区	166	9.3.1	配置属性设置	213
8.4	创建时序约束	167	9.3.2	创建 PROM 文件	214

9.4 下载实现	216	11.3.3 查看逻辑网表的层次	291
9.4.1 下载环境	216	11.3.4 使用新网表和约束更新工程	292
9.4.2 下载实现	216	11.4 设计分析和研究	297
9.4.3 JTAG 诊断	221	11.4.1 显示设计统计信息	297
9.4.4 建立 SVF 文件	222	11.4.2 运行 DRC	298
第 10 章 ChipScope Pro 调试工具	225	11.4.3 运行 WASSO 分析	299
10.1 ChipScope Pro 调试工具概述	225	11.4.4 查看逻辑层次	300
10.2 ChipScope Pro 核描述	227	11.4.5 研究原理图内的逻辑	301
10.2.1 ICON 核	227	11.4.6 修改设计时序约束	303
10.2.2 ILA 核	227	11.4.7 导入和分析 ISE 实现结果	305
10.2.3 VIO 核	230	11.4.8 导入和分析 Tree 时序结果	306
10.2.4 ATC2 核	230	11.5 分割设计	308
10.2.5 IBERT 核	231	11.5.1 分割和布局顶层设计	308
10.3 ChipScope Pro 核插入器使用	232	11.5.2 调整视图选项	310
10.3.1 ISE 中插入器使用	232	11.5.3 拆分和布局较低级物理块	311
10.3.2 ChipScope Pro 核插入器特性	233	11.5.4 察看设计时钟域	313
10.3.3 片内逻辑分析仪的使用	248	11.5.5 显示绑定网络的内容	313
第 11 章 PlanAhead 工具及应用	274	11.5.6 调整物理块的布局 and 大小	313
11.1 PlanAhead 工具概述	274	11.6 实现设计	314
11.2 I/O 引脚分配	275	11.6.1 运行 ExploreAhead 来配置和 启动运行	314
11.2.1 创建 PinAhead 工程	275	11.6.2 检查 ExploreAhead 结果	316
11.2.2 输入和分析 I/O 端口列表	277	11.6.3 同时启动多个布局规划	317
11.2.3 创建和配置 I/O 端口	278	11.6.4 输出布局规划用于实现	317
11.2.4 创建 I/O 端口接口	279	11.7 平面布局调整	318
11.2.5 布局 I/O 端口	279	11.7.1 分析 ISE 结果来调整平面布局	318
11.2.6 运行 DRC 和 WASSO 分析	283	11.7.2 使用连接显示来识别到布局 规划的逻辑	320
11.2.7 输出 I/O 配置	285	11.7.3 锁住关键逻辑	322
11.3 导入网表的设计	286	11.7.4 为减少阻塞创建额外的物理块	324
11.3.1 创建新工程	286	11.7.5 查看改善布局规划的结果	330
11.3.2 查看芯片资源和时钟域	289		

可编程逻辑器件设计流程导论

本章的目的是使读者初步了解 Xilinx 的可编程逻辑器件的设计流程，以便读者更好地掌握本书后面的内容。

1.1 设计流程概述

```
graph TD
    DI[设计输入] --> DS[设计综合]
    DI --> DV[设计验证]
    DS --> DT[设计工具]
    DT --> DXT[下载到Xilinx器件]
    DV --> FF[功能仿真]
    FF --> STA[静态时序分析]
    STA --> TS[时序仿真]
    TS --> CV[电路内部验证]
    CV --> DI
    DS --> FF
    DT --> STA
    DXT --> CV
```

图 1.1 Xilinx 标准的设计流程

1. 输入设计和综合

在设计流程的这一步中，通过原理图编辑器、硬件描述语言（HDL）或者两种混合方法来创建自己的设计。如果使用 HDL 创建设计输入，就必须将 HDL 文件综合到一个 EDIF 文件中；如果使用 Xilinx 的综合工具（Xilinx Synthesis Technology, XST），就必须把 HDL 文件综

合到一个 NGC 文件中。

2. 设计实现

通过执行特定的 Xilinx 架构, 转换逻辑设计文件格式, 例如 EDIF, 这样就可以将设计输入和综合步骤创建到物理文件格式中。这些物理信息存储于 FPGA 的本地电路说明文件 NGC 和 CPLD 的 VM6 文件中。再通过这些文件创建一个比特流文件, 并为随后器件编程创建一个 PROM 或 EPROM 文件。

3. 设计验证

使用门级仿真器或者下载电缆, 确定设计满足时间要求, 并能正常运行。查看 Xilinx 的下载电缆和演示版信息的在线帮助。

图 1.2 给出了详细的 Xilinx FPGA 设计的软件流程图。

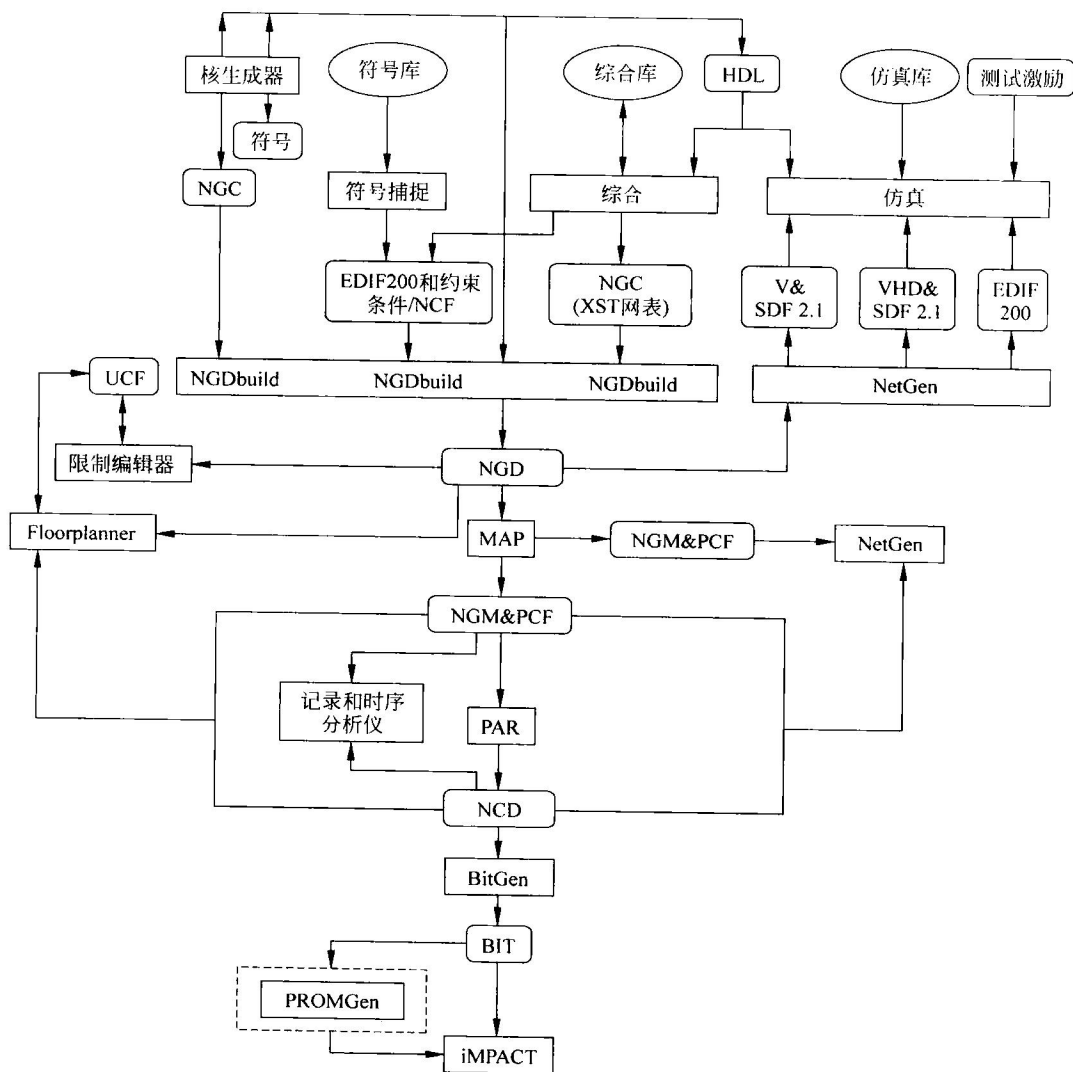


图 1.2 Xilinx FPGA 设计软件流程图

整个设计流程是一个不断地输入、执行、验证，直到设计是正确和完整的过程。Xilinx 的开发系统通过设计流程周期，允许快速反复设计。因为 Xilinx 器件允许无限制的编程，所以调试设计的电路时，不必丢弃已经编程的器件。

图 1.3 给出了详细的 Xilinx CPLD 设计的软件流程图。

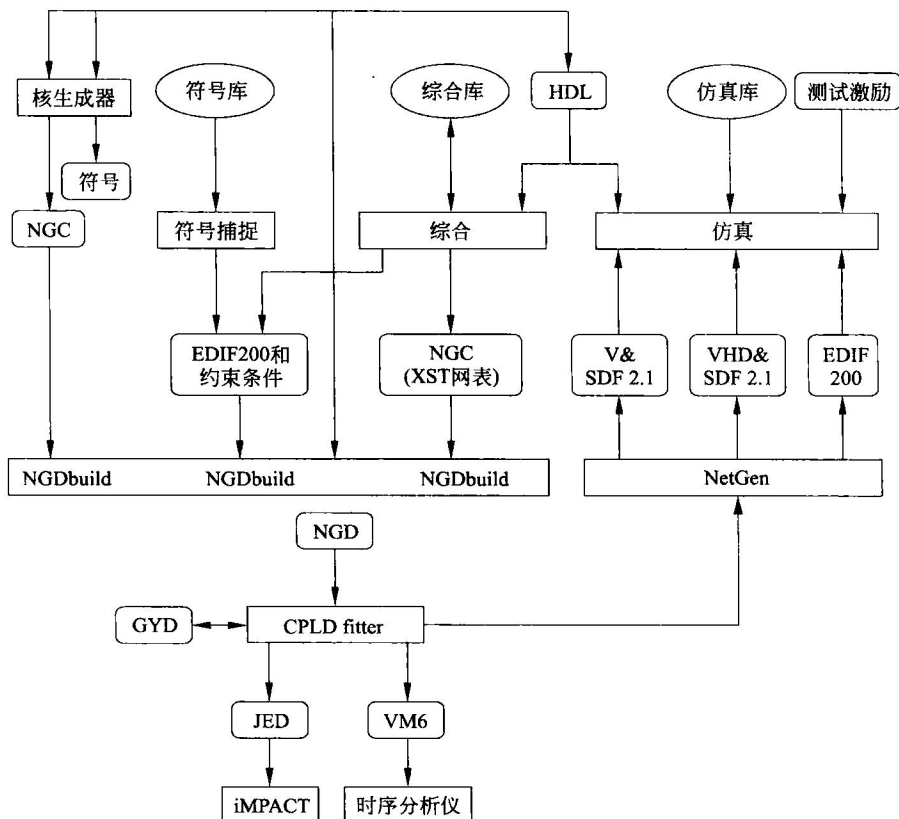


图 1.3 Xilinx CPLD 设计软件流程图

1.2 设计输入和综合

通过原理图编辑器或者文本编辑器工具输入一个设计。设计输入开始于一个设计概念，即使用画图或功能描述来表示设计。从最初的设计，创建网表，再综合并转化本地通用对象（Native Generic Object, NGO）文件。这个文件输入 Xilinx 的软件程序 NGDBuild，该程序产生一个逻辑本地通用数据库（Native Generic Database, NGD）文件。图 1.4 给出了输入设计和综合步骤。

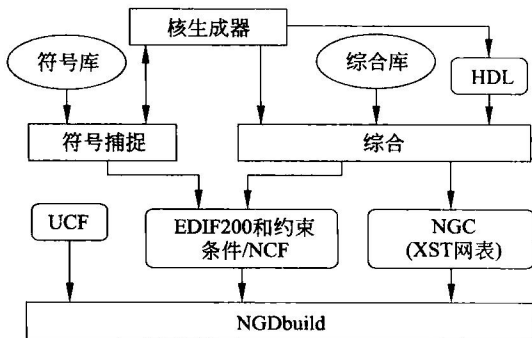


图 1.4 设计输入流程图

1.2.1 层次化设计

层次化设计对于原理图和 HDL 输入都很重要，这是因为：

- ☐ 可以将设计概念化。
- ☐ 将设计结构化。
- ☐ 使调试设计更容易。
- ☐ 使设计的不同部分的不同输入设计方法（原理图，HDL，本地编辑）能更容易结合。
- ☐ 使更新设计更容易，其中包括设计、实现，以及在设计过程中验证个别元件。
- ☐ 减少优化时间。
- ☐ 便于并行设计。

在层次化设计中，一个特定的等级名字区分了每个库元件、特定的模块，以及实例创建。

对于设计中的每一个元件和网络都应该命名，这些名称通过 FPGA 编辑工具储存和调用，还用来备份注释并出现在调试和分析工具中。如果设计者没有命名，那么原理图编辑器会自动生成名字。对于自动生成名称的电路很难进行分析，因为这些名称仅对 Xilinx 的软件有意义。

1.2.2 原理图输入

原理图工具为原理图设计输入提供了一个图形界面。通过这些工具将设计中的符号连接在一起，这些符号表示设计中的逻辑元件。设计人员可以使用独立的门来建立设计，或者连接多个门建立功能块。

1. 库元件

元件和宏都是元件库中的结构单元。Xilinx 库提供了原语，以及高级的宏功能。原语是基本的电路元件，例如与门和或门。每个原语都有唯一的库名，标号和描述。宏包括多个库元件，这些库元件包含原语和其他宏。Xilinx 的 FPGA 可以使用以下类型的宏：

- ☐ 软宏（软核）具有预先定义的功能，但有灵活的映射、布局和布线。软宏可用于所有的 FPGA。
- ☐ 相关布局宏（RPMs）有固定映射和相对布局。RPMs 可用于所有芯片，除了 XC9500 系列的 CPLD。

宏不能用于综合，因为综合工具有它们自己的模块生成器并且部要求 RPMs。如果想要覆盖模块生成，设计者可以例化核生成器模块。对于大多数的高级综合工具，这并不能提供优势，除非是对于不能推断的模块。

2. 核生成器工具（仅限于 FPGA）

Xilinx 的核生成器传递参数化的核，这个核针对 Xilinx FPGA 进行了优化。这些库包含从简单延迟元件到复杂数字信号处理的滤波器以及多路复用器。

1.2.3 HDL 输入和综合

一个典型的硬件描述语言 (HDL) 支持混合描述, 在这个描述中门和网表结构被用于功能描述。这种混合描述能力可以以高度抽象的形式描述系统结构, 并逐步完善设计中的门级实现细节。HDL 语言描述具有下列优点:

- 在设计阶段就可以及早验证设计功能。一个用 HDL 写的设计可以立即进行仿真。在这个高层次和门级的设计仿真, 在门级实现之前, 就可对结构和设计决策进行评估。
- HDL 描述比网表或者原理图描述更易阅读和理解。对于一个设计, HDL 描述提供独立技术文档以及功能。
- 对于大的设计, HDL 工具比原理图工具更容易处理。

在创建 HDL 设计之后, 要对它进行综合。在综合时, HDL 文件中的行为信息被转换为结构网表, 并且为 Xilinx 芯片优化设计。Xilinx 支持一些第三方厂商的综合工具。此外, Xilinx 也提供自己的综合技术 (Xilinx Synthesis Technology, XST) 工具。

1. 功能仿真

在创建设计之后, 可对其进行仿真。功能仿真是对设计中的逻辑进行测试, 以判断其是否正常工作。如果在设计流程中尽早地进行功能仿真, 可以为随后的设计步骤节省时间。

2. 约束

如果想要对设计中的时间参数或者布局参数进行约束, 设计者可以指定映射、块布局以及时间规范。

可以手工输入约束, 或者使用约束编辑器 Floorplanner 或者 FPGA 编辑器 (Xilinx 提供的图形用户界面工具); 还可以使用时序分析图形界面或者 TRACE 命令行程序来评估电路所违反的这些约束条件 (通过生成设计的静态时序分析)。

1) 映射约束

在 Spartan 系列和 Virtex 系列的 FPGA 中, 可以指定使用 FMAP 来确定逻辑块如何映射到可配置的逻辑块 (Configurable Logic Block, CLB)。这些映射符号可在原理图中使用。但是, 如果过多地使用这些映射限制, 则会使得设计中的布线非常困难。

2) 模块布局

块布局可限制在指定位置, 可以是多个位置中的其中一个, 或者是一个位置区域。具体位置可以用综合工具在原理图中指定出来, 或者在用户约束文件 (User Constraint File, UCF) 中指定出来。不恰当的块布局会影响设计的布局和布线。只有 I/O 块要求布局满足外部引脚的要求。

3. 时序规范

可以指定设计中路径的时间要求。在对设计进行布局和布线时, PAR 使用这些时间规范来达到最佳性能。

1.3 设计实现

设计实现过程从逻辑设计文件映射或适配到指定的器件开始,到物理设计布线成功并生成比特流文件时结束。在实现阶段也可像在设计初期一样改变约束条件。图 1.5 给出了 FPGA 的设计实现过程。

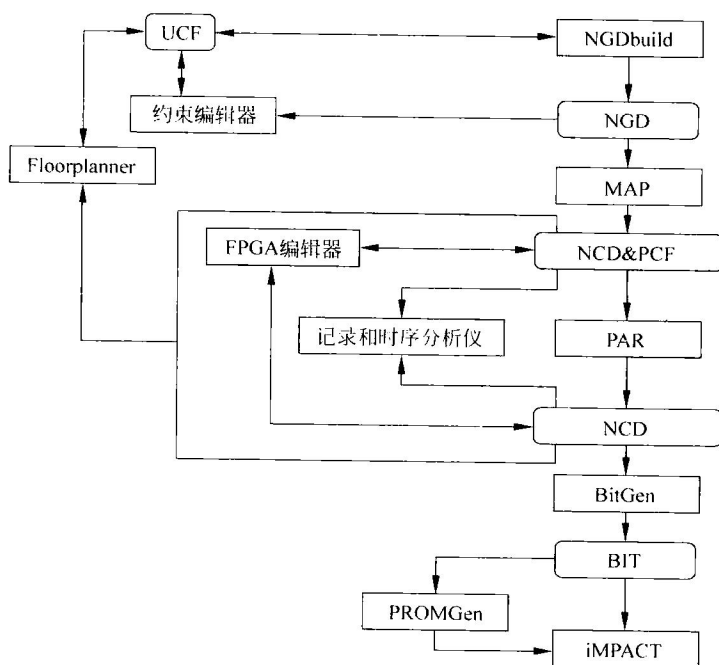


图 1.5 FPGA 的设计实现流程

下面对关键步骤进行说明：

1. 映射 (MAP, 仅限于 FPGA)

对于 FPGA, MAP 命令程序将逻辑设计映射到 Xilinx 的 FPGA 芯片。MAP 的输入是 NGD 文件, NGD 文件包括设计的逻辑描述, 这些逻辑描述的方式包括层次化的元件, 低层次的 Xilinx 原语和任意数量的硬布局布线宏 (NMC) 文件 (该文件包含物理宏的定义)。然后 MAP 将逻辑映射到 Xilinx FPGA 内的元件 (逻辑单元, I/O 单元和其他元件)。

MAP 的输出设计是一个 NCD 文件, 是设计被映射到 Xilinx FPGA 内元件的物理描述。然后使用 PAR 命令程序, 可以对 NCD 文件进行布局 and 布线。

2. 布局和布线 (PAR, 仅限于 FPGA)

对于 FPGA, PAR 命令程序用映射的 NCD 文件作为输入, 对设计进行布局 and 布线, 输出一个布局布线的 NCD 文件, 该文件被比特流生成器 BitGen 使用。当设计变化时, 设计者对设计反复进行布局 and 布线时, NCD 文件可以作为一个向导文件。在以下情况也可使用 FPGA Edit 的 GUI 工具:

- 对于一个完整的设计，在使用自动布局和布线工具前先布局布线关键的元件。
- 手动修改布局和布线，编辑器允许手动和自动的元件布局和布线。

3. 比特流的生成 (BitGen, 仅限于 FPGA)

对于 FPGA, BitGen 命令程序为 Xilinx 设备配置生成比特流。BitGen 以完整的布线的 NCD 文件作为输入, 并生成一个配置比特流 (.bit 扩展名的二进制文件)。BIT 文件包含所有的来自 NCD 文件定义内部逻辑和 FPGA 互连的配置信息, 还有来自和目标设备相关的文件。

在生成 BIT 文件之后, 使用 IMPACT 图形工具可将其下载到芯片。也可使用 PromGen 命令程序将 BIT 文件格式化为 PROM 文件, 并用 IMPACT 工具下载到 PROM。

图 1.6 给出了 CPLD 的设计实现流程。

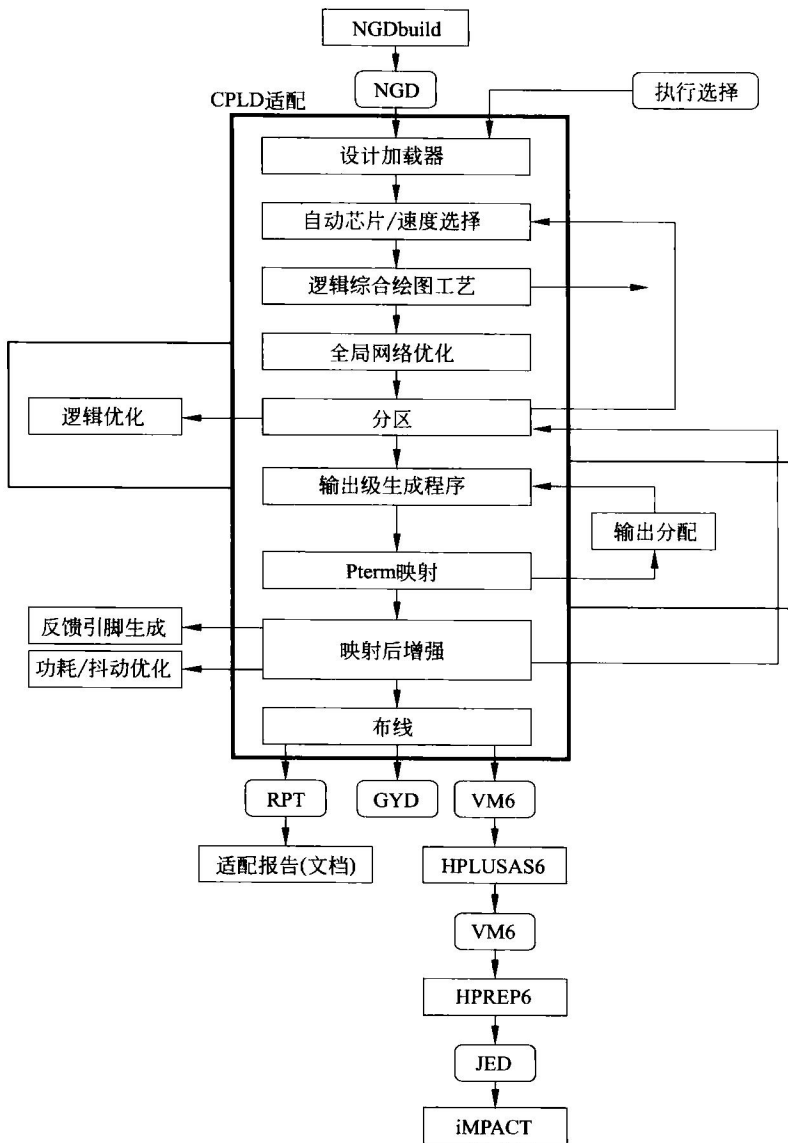


图 1.6 CPLD 的设计实现流程

1.4 设计验证

设计验证是对设计的功能和性能的测试，可以按照以下几个方法对 Xilinx 的设计进行验证：（1）仿真（功能和时间）；（2）静态时序分析；（3）电路验证。表 1.1 列出了不同的设计工具适用的验证类型。

表 1.1 验证工具

验证类型	工具
仿真	第三方模拟器（集成和非集成的）
静态时序分析	TRACE（命令行程序）；时序分析器（GUI） Mentor Graphics TAU 和 Innoveda BLAST 软件使用 STAMP 文件格式（仅用于验证输入/输出的时序）
内部电路验证	设计规则检查器（命令行程序） 下载电缆

设计验证步骤应贯穿设计的全部过程中，图 1.7 给出了 FPGA 设计流程中的三种验证方法。图 1.8 给出了 CPLD 设计流程中的三种验证方法。

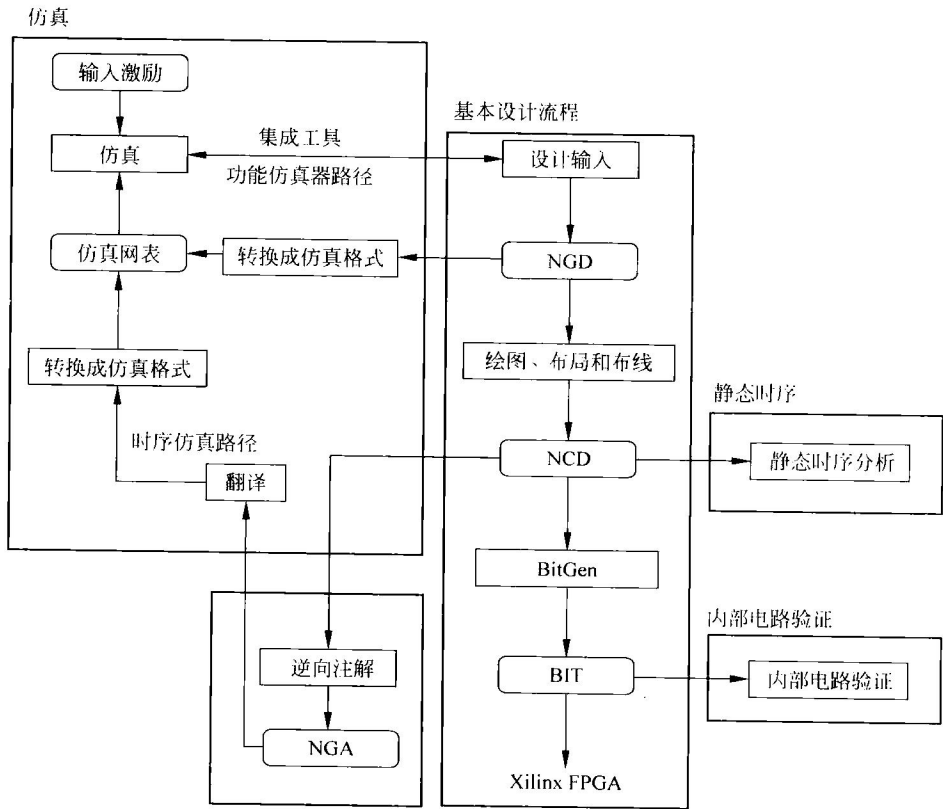


图 1.7 FPGA 设计流程中的三种验证方法