

EDA 工 | 程 | 技 | 术 | 丛 | 书 |

XILINX Xilinx公司推荐用书!

E-ELEMENTS 依元素公司推荐用书!



THE APPLICATION GUIDE ON XILINX VIVADO DESIGN SUITE

Xilinx新一代FPGA设计套件 Vivado应用指南

孟宪元 陈彰林 陆佳华 编著
Meng Xianyuan Chen Zhanglin Lu Jiahua

清华大学出版社



HDA 工 | 程 | 技 | 术 | 丛 | 书 |



THE APPLICATION GUIDE ON XILINX VIVADO DESIGN SUITE

Xilinx新一代FPGA设计套件 Vivado应用指南

孟宪元 陈彰林 陆佳华 编著
Meng Xianyuan Chen Zhanglin Lu Jiahua

清华大学出版社

内 容 简 介

本书系统论述了新一代 FPGA 设计套件 Vivado 的性能、使用方法以及 FPGA 的开发方法。全书内容包括 Vivado 设计套件的特性,全面可编程 FPGA 器件的架构,使用 Vivado 套件创建复杂数字系统设计项目,仿真系统功能,RTL 分析产生网表文件,性能要求的时序约束及综合,布局布线及静态时序分析和生成位流文件等全部设计过程,基于项目和非项目批作业两种用 Tcl 指令的设计模式,同步设计技术、HDL 编码技术、时序收敛和 HLS 优化 DSP 算法等关键技术,并以实例介绍了嵌入式系统的设计方法等。本书适合作为高校电子信息类专业的实践教学用书和工程技术人员的参考用书。

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 新一代 FPGA 设计套件 Vivado 应用指南/孟宪元,陈彰林,陆佳华编著. —北京:清华大学出版社,2014

EDA 工程技术丛书

ISBN 978-7-302-36683-6

I. ①X… II. ①孟… ②陈… ③陆… III. ①可程序逻辑器件—系统设计—指南 IV. ①TP332.1-62

中国版本图书馆 CIP 数据核字(2014)第 117265 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:梁毅

责任印制:宋林



出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:21.75

字 数:539 千字

版 次:2014 年 8 月第 1 版

印 次:2014 年 8 月第 1 次印刷

印 数:1~3000

定 价:69.00 元

产品编号:059547-01

我们生活在一个快速连接的世界中——全球有超过 60 亿台移动计算设备相互连接,并且每天都持续增加约 100 万台移动计算设备。预计到 2020 年,全球移动计算设备总数将达到 300 亿台。随着物联网(IoT)以及万联网(IoE)的发展,海量大数据的存储、传输、处理、挖掘技术出现了极大的挑战。从处理响应速度来看,计算的处理响应速度从文字时代的秒级,到多媒体时代的百毫秒级、视频时代的十毫秒级,会迅速推进到 5G 时代的 1 毫秒级。对海量数据在 1 毫秒内完成处理,将是未来数十年摆在电子信息系统设计工程师面前的巨大难题。

随着摩尔定律走向深纳米时代,在 20nm 以下的工艺节点,每个节点的性价比提高幅度会比上一代逐渐减少,而前期的一次性工程费用(NRE)投入巨大,服务客户数量稀少,使得专用集成电路(ASIC)及专用标准集成电路(ASSP)在商业模式上步入绝境,赢利的公司数量锐减直至最后消亡,尚能存活的将是可编程器件。

面对海量的计算任务,多核并行曾是解决方案之一,但受制于算法可并行部分的局限,更多的核并不能带来更高的效率,加速效能也逐渐走到了尽头。此外,受单颗芯片发热量密度限制,即使芯片上集成的晶体管越来越多,但可同时运行的晶体管数目却趋于恒定,多余的晶体管将沦为暗硅(Dark Silicon)。因此,设计者不得不将目光转向冯·诺依曼架构之外的计算构架,例如领域定制化计算(Domain Specific Computing),它可在保持灵活性的同时,发挥每一个晶体管的计算能力,当然这也离不开可编程器件技术的长足发展。

在系统级别,大数据与软件定义一切,虚拟化一切的趋势,使得系统构架工程师不得不寻求更灵活、更智慧、更快速、更绿色的解决方案。而这些解决方案的核心往往与软件、硬件及 I/O 均可编程的芯片——赛灵思公司的 All Programmable 芯片相关。

在教育领域,除了需要培养能够应对未来数十年技术挑战的电子信息系统工程师之外,教学本身也充满了变革和机遇。随着大型开放式网络课程(MOOC)的兴起,在统一平台下通过互联网,以翻转课堂的方式,打破业界与教育界的壁垒,完成软件与硬件、理论与实验、年级与院系的全面贯通,将是很多电子信息类学科教育工作者的更高追求。

赛灵思大学计划将不遗余力地帮助教育工作者应对这些变革,与清华大学出版社合作将 All Programmable 全面可编程技术系统地引入到新型知识传播体系中去,培养能够应对下一代电子系统设计挑战的卓越工程师,为实现将“中国制造”变成“中国智造”的梦想,提供充足的智力和人才保障。

谢凯年

Xilinx 大学计划大中华区经理

前言

本书是由依元素科技有限公司陈俊彦经理提议和发起,由 Xilinx ATP 讲师和技术支持工程师编写的介绍 Xilinx 新一代 FPGA 设计套件 Vivado 及其应用的教材。本书编写目标是让广大读者对 Vivado 有一个初步的全面认识,以便迅速从 ISE 升级到 Vivado。书中除了系统介绍 7 系列器件的结构和特点,也梳理了陈彰林年初赴美国 Xilinx 公司总部参加 UltraScale 器件培训的收获,对 UltraScale 技术做了明确的介绍。这为今后进一步编写该系列教材进行了一次有益的尝试和实践。

本书主要以 ATP 讲师多年来的实践经验为基础,针对读者的切实需求将有关培训内容重新组织和编写。本书以设计项目创建、功能仿真、RTL 分析、设计综合、设计实现和生成位流文件的设计过程为主线,既涵盖了以 Tcl 脚本文件的设计和与设计项目有关的编程技巧、同步设计、设计收敛和设计诊断等内容,也介绍了与信号处理和嵌入式系统设计有关的 Vivado HLS 和 EDK 两部分的设计方法。

本书在内容的叙述方式上,全面地将 Vivado 的特点与原来采用的 ISE 进行了比较,以便读者能更好地利用 Vivado 进行设计。为了让读者能针对 Xilinx FPGA 的架构进行有效的设计,也介绍了 7 系列的硬件结构特点和设计实例。此外,书中也介绍了最新的 UltraScale 器件的结构在布线资源、时钟性能以及 CLB、DSP 和 BRAM 等的改进。UltraScale 达到了更高的性能,继续保持领先一代的高水平。

本书的各个章节都安排了实验内容。全书以 wavegen 的设计实例为线索,并为简洁起见,以其中的 uart_led 为主进行实验步骤的说明。相关的实验可以在出版社的网站下载,设计所用到的 Vivado 的版本以 2013.2 为主,有些实验需要更高的 2013.3 版本。软件的基本操作步骤对于更高的版本也是适用的。

实验用的板卡采用 Xilinx 官方授权培训课程指定的 KC705 板卡和 Xilinx 大学计划的 Nexys 4 板卡,嵌入式系统的设计部分则要采用 ZedBoard 板卡,可以满足不同用户群体的多种不同的需求。

本书的第 1 章由陈彰林和孟宪元共同编写;第 2 章由张剑森和陈彰林共同编写;第 3 章由孟宪元编写;第 4 章和第 8 章由王刚编写;第 5 章和第 9 章由黄磊编写;第 6 章和第 10 章由仝信编写;第 7 章、第 11 章和第 12 章由孟宪元编写;第 13 章由 Xilinx 大学计划部的陆佳华编写。全书由孟宪元和陈彰林统稿。

本书的出版得到了 Xilinx 公司亚太区销售与市场副总裁杨飞的关心和大力支持,并得到 Xilinx 亚太区官方授权培训部的 Hedi Choy 女士、Xilinx 大学计划部的谢凯年博士和 Digilent 公司的赵峰博士等的大力支持和帮助,他们均提出了宝贵的意见和建议。在此表示衷心的感谢!

FPGA 技术发展迅速,软件版本每年会有几次更新。作者水平有限,编写时间较紧,书中的错误和疏漏之处,请读者指正。

编著者

2014 年 7 月

第 1 章	Vivado 设计套件	1
1.1	单一的、共享的、可扩展的数据模型	2
1.2	标准化 XDC 约束文件——SDC	3
1.3	多维度解析布局器	4
1.4	IP 封装器、集成器和目录	6
1.5	Vivado HLS 把 ESL 带入主流	8
1.6	其他特性	10
1.6.1	快速的时序收敛	10
1.6.2	提高器件利用率	11
1.6.3	增量设计技术	11
1.6.4	Tcl 特性	12
1.7	Vivado 按键流程执行设计项目	13
1.7.1	KC705 开发板实现计数器	14
1.7.2	在 Nexys4 开发板实现计数器	18
第 2 章	7 系列 FPGA 架构和特性	21
2.1	7 系列结构特点	21
2.1.1	采用统一的 7 系列架构	21
2.1.2	高性能和低功耗结合的工艺	23
2.2	扩展 7 系列的 UltraScale 架构	25
2.3	可配置逻辑模块 CLB	27
2.3.1	Slice 的结构和功能	28
2.3.2	SliceM 配置为 SRL	29
2.3.3	SliceM 配置为分布式 RAM	31
2.4	7 系列专用模块：Block RAM/FIFO 和 DSP 模块	34
2.4.1	Block RAM/FIFO	34
2.4.2	DSP 模块	38
2.4.3	I/O 模块	42
2.4.4	时钟资源	44
2.5	由 RTL 代码推论实验	49
2.5.1	计数器程序	49
2.5.2	实验结果	50

目录

第 3 章 创建设计项目	53
3.1 wave_gen 设计概述	53
3.2 启动 Vivado	58
3.3 仿真设计	66
3.3.1 添加仿真需要的信号	67
3.3.2 运行仿真和分析仿真结果	68
3.4 利用时钟向导配置时钟子系统	69
3.5 产生 IP 集成器子系统设计	71
3.5.1 产生 IP 集成器模块设计	71
3.5.2 定制 IP	72
3.5.3 完成子系统设计	74
3.5.4 产生 IP 输出产品	75
3.5.5 例示 IP 到设计中	76
第 4 章 RTL 级分析和设计网表文件	78
4.1 网表文件	78
4.1.1 设计项目数据库	78
4.1.2 网表文件	79
4.1.3 推演的设计网表文件	79
4.1.4 综合的设计网表文件	81
4.1.5 实现的设计网表文件	82
4.2 RTL 设计分析	83
4.2.1 RTL 网表文件	84
4.2.2 RTL 设计规则校验	84
4.2.3 浏览设计的层次	85
4.2.4 平面规划布图	85
4.2.5 时钟规划布图	86
4.3 网表文件的设计对象	86
4.3.1 通过 get_* 命令来寻找网表中的对象	87
4.3.2 设计层次	87
4.3.3 pin 的层次与名称	87
4.3.4 层次展平化	88

4.3.5	Nets 的层次分段	88
4.4	设计对象特性	88
4.4.1	查看对象的特性	89
4.4.2	Cell 的特性	90
4.4.3	Port 的特性	91
4.4.4	Pin 的特性	91
4.4.5	用户自定义特性	92
4.4.6	使用特性过滤对象	92
4.5	对象连通性	93
4.5.1	层次结构下 get_pins 命令的使用	94
4.5.2	GUI 的使用	94
4.5.3	图形化帮助界面	96
4.6	RTL 分析实例	97
	本章小结	102
第 5 章	设计综合和基本时序约束	103
5.1	设计综合	103
5.1.1	Vivado IDE 的综合环境	104
5.1.2	Vivado IDE 综合常用设置	104
5.1.3	Vivado IDE 综合流程	106
5.1.4	Vivado 支持 SystemVerilog	107
5.1.5	Vivado 工具的特点	107
5.2	基本的时序约束	108
5.2.1	静态时序通道	108
5.2.2	建立时间和保持时间校验	109
5.2.3	输入和输出的时序约束	110
5.2.4	编辑静态时序约束	113
5.2.5	约束查看器	115
5.3	综合报告	116
5.3.1	利用率分析报告	116
5.3.2	时序报告摘要	117
5.3.3	时钟网络分析(report_clock_network)	119
5.3.4	时钟关连分析(report_clock_interaction)	119

目录

5.3.5	设计规则检查(report_drc)	120
5.3.6	噪声分析(report_ssn)	120
5.3.7	使用资源分析(report_utilization)	121
5.3.8	设计功耗分析(report_power)	122
5.3.9	功耗利用的细节	122
5.4	综合实例	123
5.4.1	运行综合设计	123
5.4.2	执行基本的网表分析	125
5.4.3	在时序约束管理器窗口校验存在的时序约束	127
	本章小结	131
第 6 章	设计实现与静态时序分析	132
6.1	设计实现流程	132
6.1.1	网表优化(opt_design)	132
6.1.2	功率优化(power_opt_design)	134
6.1.3	布局设计(place_design)	134
6.1.4	物理优化(phys_opt_design)	135
6.1.5	布线设计(route_design)	135
6.2	设计实现报告	136
6.3	基础的静态时序分析	139
6.4	实现后的设计分析	141
6.4.1	在 Device 窗口观察时序通道	142
6.4.2	在原理图窗口中观察时序通道	143
6.4.3	FPGA 编辑器	143
6.4.4	产生位流文件	145
6.5	设计实现实例	146
6.5.1	完成设计实现	146
6.5.2	资源利用率报告	147
6.5.3	生成基本时序报告	148
6.5.4	分析设计中关键时序通道	148
6.5.5	利用 iMPACT 编程 KC705 演示卡	152
	本章小结	153

第 7 章	Tcl 命令设计项目	154
7.1	Tcl 的基本知识	154
7.1.1	变量	155
7.1.2	命令替换	155
7.1.3	数学表达式	156
7.1.4	反斜杠替换	156
7.2	基于项目的设计	156
7.2.1	设计项目目录和文件	156
7.2.2	项目运行管理器	158
7.2.3	管理运行	160
7.2.4	约束管理	161
7.2.5	基于项目设计流程实例	162
7.3	非项目的批作业流程	167
7.3.1	产生 RTL 设计的推敲过的网表文件	168
7.3.2	产生综合设计	168
7.3.3	随后的进程	169
7.3.4	约束管理	169
7.3.5	非项目批作业流程实例	170
7.4	脚本文件编写	175
第 8 章	同步设计技术	179
8.1	概述	179
8.1.1	同步设计	179
8.1.2	异步输入信号	180
8.1.3	亚稳态信号存在的危害	181
8.2	单比特同步电路	182
8.2.1	求解亚稳态	182
8.2.2	亚稳态的消释电路	182
8.2.3	单比特同步电路的约束条件	183
8.2.4	单比特同步电路的跨时钟域约束	184
8.2.5	复位桥	185
8.3	总线同步电路	186
8.3.1	总线一致性	186

目录

8.3.2	低速总线同步电路	187
8.3.3	总线同步电路的约束	187
8.3.4	同步计数序列	188
8.3.5	时钟交互 FIFO 的结构	191
8.3.6	时钟交互 FIFO 的约束	191
8.4	时钟和同步器	193
8.4.1	时钟之间的关系	193
8.4.2	分析时钟交互	194
8.4.3	时钟交互分析报告	195
8.4.4	使用最大延迟分析时钟交互	196
8.5	复位实验	196
8.5.1	异步复位实验	196
8.5.2	无复位实验	200
8.5.3	同步复位实验	202
8.5.4	高扇出同步复位实验	204
	本章小结	207
第 9 章	HDL 编码技巧	208
9.1	概述	208
9.2	控制集 Control Sets	210
9.2.1	FPGA 中的寄存器资源	210
9.2.2	控制集 Control Sets	212
9.3	控制信号设计要点	216
9.3.1	控制端口使用规则	216
9.3.2	控制信号的问题	216
9.3.3	低电平有效控制信号的问题	217
9.4	置位/复位信号的使用	218
9.4.1	复位的类型	218
9.4.2	全局复位与 GSR	219
9.4.3	同步置位/复位信号优势	219
9.4.4	HDL 编码中复位信号同步化	220
9.5	其他设计要点	221
9.5.1	I/O 寄存器使用要点	221

9.5.2	Block RAM 使用要点	222
9.5.3	时钟使能要点	222
9.5.4	DSP 应用中的加法器树	222
9.5.5	综合选项要点	223
9.5.6	编码方式改善可靠性、性能、功耗	223
9.6	设计实例	224
	本章小结	230
第 10 章	时序收敛	231
10.1	基线——Baselining	231
10.1.1	求解时序收敛	231
10.1.2	基线设计	232
10.1.3	设置时序报告找根源	233
10.1.4	综合/优化后的时序结果	234
10.1.5	布局布线后的时序结果	235
10.1.6	提取时序信息	236
10.2	解决常见的时序瓶颈	236
10.2.1	高扇出网线	236
10.2.2	长逻辑通道	237
10.3	最后里程的策略	239
10.3.1	时序达标的策略	239
10.3.2	综合的策略	240
10.3.3	物理优化	241
10.3.4	实现的策略	243
10.3.5	增量布局	244
10.3.6	关键电路预先布线	245
10.3.7	避免过渡约束	245
10.4	时序收敛实验	246
10.4.1	基线方法时序收敛流程	246
10.4.2	优化内部路径——基线	247
10.4.3	执行设计的基线实现	250
10.4.4	生成时序报告和分析路径	250
10.4.5	优化整个芯片	253

目录

10.4.6	添加时序例外和精细调整设计	254
本章小结	254
第 11 章	硬件诊断	255
11.1	设计诊断概述	255
11.1.1	诊断方法	255
11.1.2	Vivado 逻辑分析仪	256
11.1.3	Vivado 逻辑诊断好处	256
11.2	Vivado 逻辑诊断 IP 核	257
11.2.1	ILA 核	258
11.2.2	VIO 核	258
11.2.3	标志诊断	259
11.2.4	诊断核集线器	259
11.3	逻辑诊断探测流程	259
11.3.1	网表插入流程	260
11.3.2	HDL 例示流程	262
11.4	硬件诊断实验	263
11.4.1	HDL 例示法添加 ILA 核	263
11.4.2	系统内诊断 uart_led 设计	267
11.4.3	网表插入法添加诊断核	274
11.4.4	添加 VIO 诊断核	277
第 12 章	Vivado HLS	281
12.1	高级综合	281
12.1.1	高级综合的调度和装配	281
12.1.2	数据通道 + 控制器架构	282
12.1.3	理解 Vivado HLS	285
12.2	高级综合的优化方法	291
12.3	Vivado HLS 设计例子	292
12.3.1	HLS 工具的流程	292
12.3.2	HLS 生成 IP 核	298

第 13 章 嵌入式系统 Zynq 设计	305
13.1 Zynq 概述	305
13.2 Zynq 设计入门	306
13.2.1 Vivado 工程创建	306
13.2.2 由 Vivado 创建 Zynq 嵌入式系统	308
13.2.3 SDK 应用程序编写	313
13.3 ZYNQ 嵌入式系统调试方法	320
13.3.1 Vivado 硬件调试	320
13.3.2 使用 SDK 进行 ZYNQ debug	323
13.4 Booting Linux on ZedBoard	327
13.4.1 创建 FSBL.elf	327
13.4.2 从 SD 卡启动 Linux	328
13.4.3 从 QSPI 启动 Linux	330

第 1 章 Vivado 设计套件

采用 28nm 的半导体工艺, Xilinx 公司推出了领先一代的硬件、软件和 I/O 全面可编程的 SoC——Zynq 7000 系列, 在单芯片上将双核 ARM Cortex-A9 处理器系统(PS)与 7 系列 Artix 或 Kintex FPGA 器件相同的可编程逻辑(PL)完美地结合在一起, 在器件的架构上具有以下的特点:

(1) 处理器系统作为主设备负责为存储器和通信外设等提供支持硬件, 并且能够在不对可编程逻辑部分加电或配置的情况下自主运行, 按照正常的软件引导过程, 从片内的非易失存储器 ROM 启动, 随后执行更复杂的引导载入程序。

(2) 可编程逻辑部分与处理器系统之间可以实现全面的互连传输, 除了可以通过 JTAG 接口进行配置之外, 也可以通过处理器配置访问端口载入部分或完整的配置。可编程逻辑部分的器件架构与 7 系列 FPGA 是完全相同的, 所以在性能、规模和功耗上都有提高。

利用 7 系列的完全可编程 FPGA 和 SoC 实现数字系统、DSP 系统或嵌入式系统都需要更好的开发工具和手段, 以满足设计规模和要求的不断增长。赛灵思(Xilinx)公司的 Vivado 设计套件在经历了四年的开发和一年的测试, 并通过早期试用计划之后, 才向客户郑重推出和公开发布。

Vivado 设计套件提供了高度统一的设计环境, 并配有全新的系统级和芯片级工具, 构建在共享、可扩展的数据模型架构和通用调试环境的基础上。Vivado 套件还是一款基于业界标准的开放式环境, 诸如 AMBA[®] AXI4 互联、IP-XACT IP 封装元数据、工具命令语言(Tcl)、Synopsys 设计约束(SDC)以及其他有助于设计流程满足用户需求的业界标准。Xilinx 设计的 Vivado 套件支持各类可编程技术组合使用, 并可扩展到 1 亿个 ASIC 等效门的设计。

由于任何 FPGA 器件的集成设计套件的核心都是物理设计流程, 包括综合、布局规划、布局布线、功耗和时序分析、优化和 ECO。所以后文结合物理设计流程分析 Vivado 设计套件的特性及超越前期的 ISE 软件的优越之处。

1.1 单一的、共享的、可扩展的数据模型

Xilinx 公司利用 Vivado 设计打造了一个最先进的设计实现流程,可以让客户更快地实现设计收敛。为了减少设计的迭代次数和总体设计时间,并提高整体生产力,Xilinx 采用一个单一的、共享的、可扩展的数据模型架构,建立其设计实现流程,这种框架也常见于当今最先进的 ASIC 设计环境。这种共享的、可扩展的数据模型架构可以让实现流程中的综合、仿真、布局规划、布局布线等所有步骤在内存数据模型上运行,故在流程中的每一步都可以进行调试和分析,这样用户就可在设计流程中尽早掌握关键设计指标的情况,包括时序、功耗、资源利用和布线拥塞等。而且这些指标的估测将在实现过程中随着设计流程的推进而趋向于更加精确。

具体来说,这种统一的数据模型使 Xilinx 能够将其新型多维分析布局布线引擎与套件的 RTL 综合引擎、新型多语言仿真引擎,以及 IP 集成器 (IP Integrator)、引脚编辑器 (Pin Editor)、布局规划器 (Floor Planner)、器件编辑器 (Device Editor) 等各工具紧密集成在一起。客户可以通过使用该套件的全面交叉观测功能来跟踪并交叉观测原理图、时序报告、逻辑单元或其他视图,直至 HDL 代码中的给定问题。

用户现在可以对设计流程中的每一步进行分析,而且环环相扣。综合后,还可对设计流程的每一步进行时序、功耗、噪声和资源利用分析。这样设计者就能够很早发现时序或功耗问题并通过几次迭代快速、前瞻性地解决问题,而不必等到布局布线完成后通过长时间执行多次迭代来解决。

这种可扩展的数据模型架构提供的紧密集成功能还增强了按键式流程的效果,从而可满足用户对工具实现最大自动化、完成大部分工作的期望。同时,这种模型还能够满足用户对更高级的控制、更深入的分析以及掌控每个设计步骤进程的需要。表 1-1 将 FPGA Vivado 设计套件与原有的 ISE 设计软件进行了比较。

表 1-1 Vivado 与 ISE 对比

Vivado	ISE
流程是一系列 Tcl 指令,运行在单个存储器中的数据库上,灵活性和交互性更大	流程由一系列程序组成,利用多个文件运行和通信
在存储器中的单个共用数据模型可以贯穿整个流程运行,允许做交互诊断、修正时序等许多事情: (1) 模型改善速度; (2) 减少存储量; (3) 交互的 IP 即插即用环境 AXI4, IP_XACT	流程的每个步骤要求不同的数据模型 (NGC, NGD, NCD, NGM): (1) 固定的约束和数据交换; (2) 运行时间和存储量恶化; (3) 影响使用的方便性
共用的约束语言(XDC)贯穿整个流程: (1) 约束适用于流程任何级别; (2) 实时诊断	实现后的时序不能改变,对于交互诊断没有反向兼容性
在流程各个级别产生报告——Robust Tcl API	RTL 通过位文件控制: (1) 利用编制脚本,灵活的非项目潜能; (2) 专门的指令行选项

续表

Vivado	ISE
在流程的任何级别保存 checkpoint 设计： (1) 网表文件； (2) 约束文件； (3) 布局和布线结果	在流程的各个级别只利用独立的工具： (1) 系统设计：Platform Studio, System Generator (2) RTL: CORE Generator, ISim, PlanAhead (3) NGC/EDIF: PlanAhead tool (4) NCD: FPGA Editor, Power Analyzer, ISim, PlanAhead (5) Bit file: ChipScope, iMPACT

1.2 标准化 XDC 约束文件——SDC

FPGA 器件的设计技术,随着其规模的不断增长而日趋复杂,设计工具的设计流程也随之不断发展,而且越来越像 ASIC 芯片的设计流程。

20 世纪 90 年代,FPGA 的设计流程跟当时的简易 ASIC 的设计流程一样,如图 1-1(a)所示。最初的设计流程以 RTL 级的设计描述为基础,在对设计功能进行仿真的基础上,采用综合及布局布线工具,在 FPGA 中以硬件的方式实现要求的设计。

随着 FPGA 设计进一步趋向于复杂化,FPGA 设计团队在设计流程中增加了时序分析功能,以此帮助客户确保设计能按指定的频率运行。今天的 FPGA 已经发展为庞大的系统级设计平台,设计团队通常要通过 RTL 分析来最小化设计迭代,并确保设计能够实现相应的性能目标。为了更好地控制设计流程中集成的设计工具,加速设计上市进程,设计人员需要更好地了解设计的规模和复杂性。

当今的 FPGA 设计团队正在采用一种新型的设计方法,在整个设计流程中贯穿约束机制。即借鉴 ASIC 的设计方法,添加比较完善的约束条件,然后通过 RTL 仿真,时序分析,后仿真来解决问题,尽量避免在 FPGA 电路板上来调试。Xilinx 最新的 Vivado™ 设计流程就支持当下最流行的一种约束方法——Synopsys 设计约束 (SDC) 格式,可以通过 SDC 设计约束让设计项目受益。

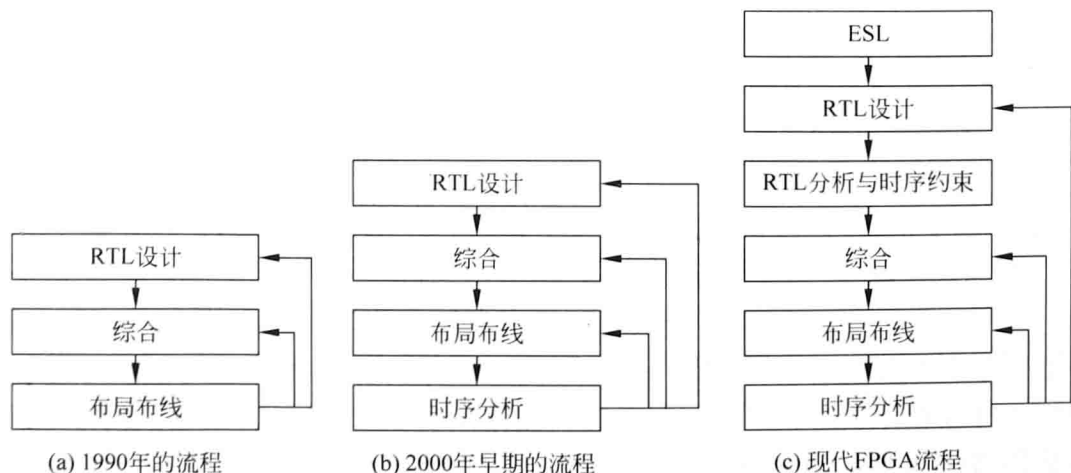


图 1-1 FPGA 工具设计流程的变迁