

Constraining Designs for Synthesis
and Timing Analysis

综合与时序分析的设计约束

Synopsys设计约束 (SDC) 实用指南

[美] 斯里达尔·甘加达兰 (Sridhar Gangadharan) 著
[印度] 桑杰·丘里瓦拉 (Sanjay Churiwala)

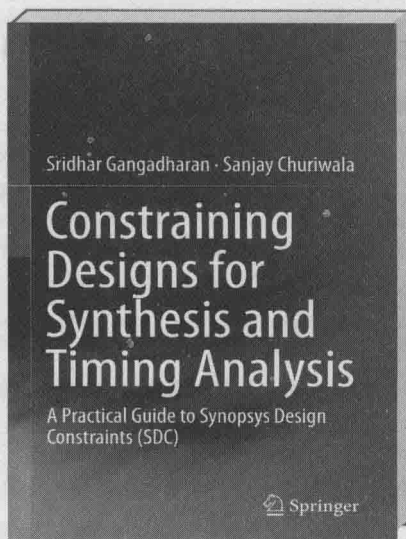
韩德强 张丽艳 王宗侠 等译



机械工业出版社
China Machine Press



电子与嵌入式系统
设计译丛



Constraining Designs for Synthesis
and Timing Analysis

综合与时序分析的设计约束

Synopsys设计约束 (SDC) 实用指南

[美] 斯里达尔·甘加达兰 (Sridhar Gangadharan) 著
[印度] 桑杰·丘里瓦拉 (Sanjay Churiwala)

韩德强 张丽艳 王宗侠 等译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

综合与时序分析的设计约束: Synopsys 设计约束 (SDC) 实用指南 / (美) 斯里达尔·甘加达兰 (Sridhar Gangadharan) 等著; 韩德强等译. —北京: 机械工业出版社, 2018.1
(电子与嵌入式系统设计译丛)

书名原文: Constraining Designs for Synthesis and Timing Analysis

ISBN 978-7-111-58894-8

I. 综… II. ①斯… ②韩… III. 集成电路 - 电路设计 - 指南 IV. TN402-62

中国版本图书馆 CIP 数据核字 (2018) 第 003101 号

本书版权登记号: 图字 01-2015-5951

Translation from the English language edition: Constraining Designs for Synthesis and Timing Analysis by Sridhar Gangadharan and Sanjay Churiwala.

Copyright © Springer Science + Business media New York 2013.

Springer-Verlag New York, Inc. is part of Springer Science+Business Media.

All rights reserved.

本书中文简体字版由 Springer Science+Business Media 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

综合与时序分析的设计约束 Synopsys 设计约束 (SDC) 实用指南

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 蒋 越

责任校对: 李秋荣

印 刷: 三河市宏图印务有限公司

版 次: 2018 年 2 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 12

书 号: ISBN 978-7-111-58894-8

定 价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

译者序

时序在数字系统中占有至关重要的地位，时序约束对数字系统的设计起着显著的作用，定义时序约束是一个相当复杂的过程。

本书作者 Sridhar Gangadharan 和 Sanjay Churiwala 在 EDA 方面有着丰富的实践经验，长期从事时序约束的研究。他们通过大量的实例，详细介绍了时序约束应该“怎么做”。本书以 Synopsys 公司的 SDC 为基础，详细介绍了指定时序要求所需要的概念，其中，包括综合、静态时序分析以及布局和布线等受时序约束影响的关键方面。本书是 SoC 和 ASIC 设计方面不可多得的一本实践指南。

本书由北京工业大学信息学部计算机学院的部分教师翻译，其中第 11~13 章和前言等由韩德强翻译，第 1、5、14 章由张丽艳翻译，第 2~4 章由王宗侠翻译，第 6~8 章由杨淇善翻译，第 9、10 章由邵温翻译，第 15~17 章由鲁鹏程翻译，全书的审校由韩德强完成。

在本书的翻译过程中得到了机械工业出版社华章公司的缪杰编辑的大力支持，在此表示由衷的感谢！

限于译者水平有限，翻译中难免有错误或不妥之处，真诚希望各位读者指正。

韩德强

2017 年 10 月于北京工业大学

推 荐 序

有人说“时序就是一切”。如果你正在展示业务，确实如此，如果你正在设计片上系统（system-on-chip, SoC），同样如此。SoC 是推动手持式消费电子产品革命的中坚力量，它们使智能手机和平板电脑成为可能。正确定义和管理 SoC 的时序约束是至关重要的任务，这些工作完成的如何将影响芯片设计的成功。

SoC 通常是来自多个供应商的许多复杂基本电路单元的集合。设计人员的工作就是将所有这些电路单元组合在一起，实现芯片在功率、性能和成本上相互竞争的目标。所有这一切都发生在整个团队受到巨大进度压力的条件下。事实上，如此多的 SoC 器件第一次能够运行，就是一个奇迹。与 SoC 设计有关的许多挑战和许多重要的技术有助于使之成为可能。

在本书中，两位作者对时序约束进行了深入的研究，并且讨论了时序约束对 SoC 设计项目成功的显著影响。同时本书为 ASIC 和 FPGA 设计范例提供了很多实例。从表面上看，定义时序约束似乎是一个简单的过程。事实上，这是一个复杂的过程，在这里面有许多重要的细微差别和相互关系。作者做了一件很有意义的事，解释了许多相关实例的过程，并详细解释了“怎么做”。

随着设计的复杂性日益增加，为了提高设计效率和管理风险而进行了大量的工作。时序约束对二者的影响尚未完全展现。管理不善或不正确的约束可能对设计工作产生明显的负面影响，并可能导致芯片设计失败。这种情况发生的概率随着新技术节点而增加。我认为时序约束正在成为设计挑战的主要方面，我祝贺他们为这一重要主题开发了一个如此完整的指南。我也希望读者觉得这本书很有用。

Ajoy Bose 博士

Atrenta 公司董事长、总裁兼首席执行官

美国加利福尼亚州圣何塞

前 言

在当今的深亚微米世界中，对于设计人员而言，时序已经成为开发专用集成电路（ASIC）或片上系统（SoC）的重要挑战。设计工程师在设计流程的不同阶段之间进行多次循环迭代，以满足时序要求。时序不仅是芯片的响应时间，还是芯片功能的一个组成部分，可以确保其能够与系统上其他元器件无缝地通信。这就提出了问题，什么是时序？如何指定它？

本书是一本实践指南，有助于读者编写和理解集成电路设计中的时序约束。通过本书读者将学习到如何有效和正确地编写约束，以便实现 IC 或 FPGA 设计的期望性能，包括关于约束重用的注意事项。其覆盖范围包括受时序约束影响的设计流程的关键方面，有综合、静态时序分析以及布局布线。本书详细说明了指定时序要求所需要的概念，然后将其应用于设计流程中的特定阶段，所有这些都包含在 Synopsys 设计约束（SDC）的上下文中，SDC 是业界领先的用于指定约束的格式。

我们经常从许多设计工程师那里听到有一些书中介绍了综合和静态时序分析等概念，这些概念涵盖了时序约束，但从来没有详细描述过。根据我们多年在时序特性描述、延迟计算、时序分析以及创建和验证约束等领域的工作，本书介绍了指定时序要求所需的概念。

本书结构

下面是本书的结构安排。

第 1~3 章介绍时序分析的主题，包括其在设计周期背景下的需求。这些章节中的描述是与供应商、语言和格式无关的。

第 4 章提供 Tcl 语言的概述，因为 SDC（Synopsys 设计约束）是 Tcl 的扩展。本章还介绍 SDC 的概念。

可以将前 4 章看作入门部分。

第 5~8 章一同构成讨论时钟的部分，介绍如何应用与时钟相关的约束。这些章介绍各种时钟及其关系，以及如何在 SDC 中指定它们。

第 9 章和第 10 章介绍如何对其余的（非时钟）端口应用约束。有了这部分内容，就覆盖了所有的主要端口。

第 11 ~ 13 章介绍对时序异常的需求。这些章介绍如何在 SDC 中正确地指定异常。

第 14 章和第 15 章涉及更专业的话题，这些概念较少论及个别的约束。相反，它们深入了解设计团队是如何管理大量的约束的，因为它们在跨流程时从前端到后端，将完整的设计划分成电路单元，然后再整合各个电路单元。

第 16 章介绍一些其他的 SDC 命令，这些命令可能会对时序分析产生影响。

本书还有一些命令没有涵盖。但是，通过本书可以对时序分析和 SDC 有基本了解，用户可以轻松理解其余的任何命令，包括 SDC 未来版本中可能出现的任何扩展。

支持 SDC 的大多数工具通常还允许对 SDC 进行一些扩展，以便使用特定工具实现更高的准确性或更好的易用性。第 17 章概述针对 Xilinx Vivado 扩展的 SDC 时序约束。

其他资源

SDC 是由 Synopsys 公司发布的开源版本。SDC 文档和解析器可从 Synopsys 网站免费下载。

意见反馈

我们已尽最大努力来提供准确的概念描述。也得到了业内一些专家的帮助，以检查素材的准确性。但是，如果你发现一些描述混乱或有错误的地方，请告知我们。

祝阅读愉快！

Sanjay Churiwala
Sridhar Gangadharan

致 谢

我要感谢 Mark Aaldering, 他首先告诉我需要一本关于 SDC 的书。他介绍说, 越来越多的工具正在转向 SDC, 却没有关于这个主题的书籍。随着 Xilinx 接纳了 SDC, 将有更多 Xilinx FPGA 的用户需要学习 SDC。

还要感谢我的朋友 Sridhar Gangadharan, 感谢他为编写本书所做的努力。

感谢 Springer 的 Charles Glaser, 因为他很容易地接受了我们对这本书的提议。

我想把这本书献给那些帮我理解 SDC 和时序约束的人, 具体是我受益于他们在我职业生涯的不同阶段对于时序分析主题的大量讨论。这个列表并不详尽, 一些著名的人包括 Fred Revenu (Xilinx)、Greg Daughtry (Xilinx)、Nupur Gupta (ST Microelectronics)、Pankaj Jain (ST Microelectronics)、Olivia Riewer (ST Microelectronics)、K A Rajagopalan (Texas Instruments)、Subrangshu (Shubro) Das (Texas Instruments)、Satish Soman (Atrenta)、Manish Goel (Atrenta)、Shaker Sharwary (Atrenta) Pratyush Prasoon (Cadence)、Sneh Saurabh (Cadence)、Girjesh Soni (Synopsys)、Rohan Gaddh (目前, 回到学校), 等等。

我要感谢我所有的老师, 但特别要提到印度理工学院的 Swapna Banerjee 教授, 他在我即将毕业之时带领我进入了 VLSI 领域。在 20 世纪 90 年代初, Dhiraj Sogani (Red Pine)、Mithilesh Jha (Masamb)、Ashutosh Varma (Cadence) 和 Late Nalyder Jain 博士带领我进入了时序领域。然后, 2000 年初, Sushil Gupta (Atrenta) 和 Vivek Gupta 首先带领我进入了 SDC 领域。

感谢 Girjesh Soni (Synopsys)、Subrangshu (Shubro) Das (Texas Instruments)、Satish Soman (Atrenta)、Shrinivasraj Muddey (Xilinx)、Olivier Florent、Ravi Balachandran 和 Russell Roan (Atrenta), 因为他们帮助审阅了本书的部分内容, 确保了素材的准确。也要感谢编写本书第 17 章的 Fred Revenu (Xilinx)。还要感谢 Charu Puri, 他提供了本书中的所有插图。

我要感谢管理部门和 Xilinx 的法律团队鼓励我参与这项活动。这些人包括 Mark Aaldering、Salil Raje、Vidya Rajagopalan、Scott Hover Smoot 和 Sue Lynn Neoh。Ramine Roane (Xilinx) 也经常鼓励我。他唯一的抱怨是: 为什么是在 2013 年中期? 为什么不早一点?

特别要感谢 Mike Gianfagna (Atrenta), 他参与和联系的此类任何活动几乎都是确保所有障碍得到解决的保证。

编写本书的信心来源于我以前出版的书得到的积极反响。我得到的最好的反响来自 Ramesh Dewangan (Atrenta), 他送给了他女儿一本, 也来自 Ahmed Hemani 教授 (瑞典皇家理工学院), 他把这本书推荐给了学生。在印度, 如果人们非常认可一本书, 并且认为他的学生或孩子也应该学习这本书, 这便是对这本书最佳的褒奖之一。感谢他们对我的这种信任。

最后很重要的一点是感谢我的家人和朋友。他们的通力配合和良好祝愿一直是我前进的动力。而且, 我的小女儿 (Lubha Churiwala - Std. VI) 还帮助我随机检查语法和一致性。

Sanjay Churiwala

印度海德拉巴

致 谢 二

要不是 Sanjay Churiwala 提出的想法，我从来没有想过会出版一本关于时序约束的书。我想感谢一些帮助我的人，因为没有他们的支持，我不可能实现这一里程碑式的目标。

首先，感谢父母为教育我所做的一切牺牲并且一直鼓励我，给予我灵感，我把这本书献给他们。还要感谢我的妻子 Bulbul 和儿子 Pratyush 对我的支持，尽管这本书让我疏忽了他们。

我要感谢我的导师 Ajoy Bose 博士 (Atrenta 有限公司 CEO)。Bose 博士 21 年前把我带进了 EDA 世界，并且他一直是我的职业生平的指路明灯。还要感谢我的主管 Mo Movahed，感谢他在我编写本书时给出的支持和建议。我要衷心感谢 Mike Gianfagna 帮助我签订合同、校对和提供所有必要的资源。

20 年前，Ameesh Desai (LSI Logic) 带领我走进时序领域。感谢 Ameesh !

非常幸运有一批来自 EDA 和半导体行业的审校者，他们坦率的反馈和帮助让本书的内容更加丰富。感谢 Girjesh Soni、Olivier Florent、Ravi Balachandran、Russell Roan、Satish Soman、Shrinivasraj Muddey 和 Subrangshu Das。谢谢 Fred Revenu (Xilinx) 撰写关于 Xilinx 扩展的章节。还要感谢 Charu Puri，他提供了本书的所有插图。

我要感谢最亲爱的朋友 S. Raman (LSI 印度分公司总经理)，感谢他在我职业生涯的每一阶段对我的鼓励。

我要感谢在印度理工学院德里分校的老师，他们带领我走进了 VLSI 领域。也是由于我的客户和同事 (Amit Handa、Chandan Kumar、Hemant Ladhe、Irene Serre、Jean Philippe Binois、Manish Goel、Mark Silvestri、Neelu Bajaj、Nupur Gupta、Olivier Florent、Osamu Yaegashi、Pankaj Jain、Pratyush Prason、Shaker Sarwary、Sanjay Churiwala、Satish Soman、Soenke Grimpen、Ramesh Dewangan、Russell Roan、Tanveer Singh、Tom Carlstedt-Duke 及 Tanigawa Yutaka) 的特别信任，帮助我加深了对这一领域的理解。如果遗漏了任何人，我表示抱歉。

感谢 Sanjay 同意与我合作并编写本书。

最后非常重要的一点，感谢 Synopsys 公司提出了像 SDC 这样的标准，这个标准极大地帮助了设计社区，使其成为敏捷项目的一部分。

我希望你觉得这本书很有用。

Sridhar Gangadhar

美国加利福尼亚州圣何塞

目 录

译者序	
推荐序	
前言	
致谢一	
致谢二	
第 1 章 绪论	1
1.1 ASIC 设计流程	1
1.2 FPGA 设计流程	4
1.3 ASIC 和 FPGA 设计流程中的时序 约束	6
1.4 纳米级设计中的时序约束问题	6
1.5 小结	7
第 2 章 综合的基础知识	8
2.1 综合的解释	8
2.2 时序约束在综合中的作用	8
2.2.1 优化	9
2.2.2 输入重排序	9
2.2.3 输入缓冲	10
2.2.4 输出缓冲	11
2.3 综合中面临的普遍问题	11
2.3.1 设计划分	11
2.3.2 更新约束	12
2.3.3 多时钟设计	12
2.4 小结	12
第 3 章 时序分析与约束	14
3.1 静态时序分析	14
3.2 时序约束在 STA 中的作用	15
3.2.1 约束作为声明	16
3.2.2 约束作为断言	16
3.2.3 约束作为指令	16
3.2.4 约束作为异常	17
3.2.5 约束的角色变化	17
3.3 STA 中的常见问题	18
3.3.1 无功能检查	18
3.3.2 无声明检查	18
3.3.3 要求正确	18
3.3.4 约束中的常见错误	19
3.3.5 好约束的特征	20
3.4 延迟计算与 STA	21
3.5 时序路径	21
3.5.1 起点和终点	22
3.5.2 打断路径	23
3.5.3 功能路径与时序路径	23
3.5.4 时钟路径与数据路径	23
3.6 建立与保持	24
3.6.1 建立分析	24
3.6.2 保持分析	24
3.6.3 其他分析	25
3.7 裕度	25

3.8	片上变化	26	5.4	虚拟时钟	42
3.9	小结	27	5.5	其他时钟特征	43
第4章	通过 Tcl 扩展 SDC	28	5.6	时钟规格的重要性	43
4.1	时序约束的历史	28	5.7	小结	44
4.2	Tcl 基础知识	29	第6章	生成时钟	45
4.2.1	Tcl 变量	29	6.1	时钟分频器	45
4.2.2	Tcl 列表	30	6.2	时钟乘法器	46
4.2.3	Tcl 表达式与运算符	31	6.3	时钟门控	46
4.2.4	Tcl 的控制流语句	31	6.4	create_generated_clock	47
4.2.5	其他 Tcl 命令	33	6.4.1	定义生成时钟的对象	47
4.3	SDC 综述	33	6.4.2	定义生成时钟的源	48
4.3.1	时序约束	34	6.4.3	时钟命名	48
4.3.2	面积与功率约束	34	6.4.4	设定生成时钟的特性	48
4.3.3	设计规则约束	34	6.4.5	时钟沿位移	51
4.3.4	接口约束	34	6.4.6	多个同源时钟	52
4.3.5	特定模式和配置约束	34	6.4.7	使能组合电路路径	53
4.3.6	设计约束异常	35	6.5	生成时钟相关的注意事项	54
4.3.7	其他命令	35	6.6	小结	54
4.4	SDC 中的设计查询	35	第7章	时钟组	55
4.5	SDC 作为标准	36	7.1	建立和保持时序检查	55
4.6	小结	36	7.1.1	高速至低速时钟	56
第5章	时钟	37	7.1.2	低速至高速时钟	57
5.1	时钟周期和频率	37	7.1.3	多个时钟于不同周期内 同步	57
5.2	时钟沿和占空比	38	7.1.4	异步时钟	58
5.3	creat_clock	39	7.2	逻辑和物理独立时钟	58
5.3.1	定义时钟周期	39	7.3	串扰	59
5.3.2	标识时钟源	39	7.4	set_clock_group	60
5.3.3	命名时钟	40	7.5	时钟组相关的注意事项	62
5.3.4	指定占空比	40	7.6	小结	62
5.3.5	同源多时钟	41			
5.3.6	注释时钟	42			

第 8 章 其他时钟特性	63	9.4.1 时钟规格	83
8.1 过渡时间	63	9.4.2 -level_sensitive	83
8.2 set_clock_transition	64	9.4.3 rise/fall 限定符	84
8.3 偏斜和抖动	65	9.4.4 min/max 限定符	84
8.4 set_clock_uncertainty	65	9.4.5 -add_delay	84
8.4.1 内部时钟不确定度	66	9.4.6 时钟延迟	84
8.4.2 交互时钟不确定度	66	9.4.7 完成输出延迟约束	84
8.5 时钟延迟	67	9.5 输入延迟和输出延迟之间的关系	84
8.6 set_clock_latency	68	9.6 时序分析实例	85
8.7 时钟路径的单边性	70	9.6.1 输入延迟: 最大输入延迟	86
8.8 set_clock_sense	71	9.6.2 输入延迟: 最小输入延迟	87
8.9 理想网络	72	9.6.3 输出延迟: 最大输出延迟	87
8.10 小结	73	9.6.4 输出延迟: 最小输出延迟	88
第 9 章 端口延迟	74	9.7 负延迟	89
9.1 输入有效	74	9.8 小结	90
9.1.1 最小和最大有效时间	75	第 10 章 完整的端口约束	91
9.1.2 多时钟	75	10.1 驱动能力	91
9.1.3 理解输入信号的到达时间	76	10.2 驱动单元	93
9.2 输出要求	77	10.3 输入过渡	97
9.2.1 最小和最大要求时间	78	10.4 扇出数	98
9.2.2 多个参考事件	78	10.5 扇出负载	98
9.2.3 理解输出要求时间	79	10.6 负载	99
9.3 set_input_delay	79	10.6.1 净电容	99
9.3.1 时钟规格	79	10.6.2 调整引脚负载	99
9.3.2 -level_sensitive	80	10.6.3 负载类型	100
9.3.3 rise/fall 限定符	80	10.6.4 负载和扇出负载	100
9.3.4 min/max 限定符	81	10.6.5 输入负载	101
9.3.5 -add_delay	81	10.7 小结	101
9.3.6 时钟延迟	82	第 11 章 虚假路径	102
9.3.7 完成输入延迟约束	83	11.1 简介	102
9.4 set_output_delay	83		

11.2	set_false_path	102	13.2	set_min_delay	124
11.3	路径规格	103	13.3	输入 / 输出延时	124
11.4	过渡过程规格	105	13.3.1	用无关时钟约束	124
11.5	建立 / 保持规格	107	13.3.2	用虚拟时钟约束	125
11.6	虚假路径的类型	107	13.3.3	用相关时钟约束	125
11.6.1	组合电路的虚假路径	108	13.4	最小 / 最大延迟与输入 / 输出延迟的对比	126
11.6.2	时序电路的虚假路径	108	13.5	直通	127
11.6.3	动态激活的虚假路径	108	13.6	点到点异常	129
11.6.4	时序的虚假路径	109	13.7	路径阻断	130
11.6.5	基于总线协议的虚假路径	110	13.8	小结	131
11.6.6	虚拟时钟和真实时钟之间的虚假路径	110	第 14 章 模式分析		132
11.7	set_disable_timing	112	14.1	使用模式	132
11.8	虚假路径的问题	112	14.2	多模式	132
11.9	小结	113	14.3	单模式与合并模式的对比	133
第 12 章 多周期路径		114	14.4	设置模式	134
12.1	多周期路径的 SDC 命令	114	14.5	其他约束	136
12.2	路径和过渡过程规格	115	14.6	模式分析挑战	136
12.3	建立 / 保持规格	115	14.6.1	时序终止迭代	136
12.4	位移量	116	14.6.2	时序路径丢失	137
12.5	多周期规格实例	119	14.7	冲突模式	137
12.5.1	基于 FSM 的数据传输	119	14.8	模式名称	138
12.5.2	源同步接口	119	14.9	小结	139
12.5.3	复位	121	第 15 章 约束管理		140
12.5.4	异步时钟	121	15.1	自顶向下的方法	140
12.5.5	大数据路径宏	122	15.2	自底向上的方法	140
12.5.6	多模式	122	15.3	自顶向下和自底向上相结合的设计方法	143
12.6	小结	122	15.4	多模式合并	145
第 13 章 组合电路路径		123	15.4.1	选择悲观时钟	147
13.1	set_max_delay	123			

15.4.2	时钟互斥	147	16.6	面积约束	163
15.4.3	部分专用时钟	148	16.7	功率约束	164
15.4.4	合并功能和测试模式	149	16.7.1	电压岛	164
15.4.5	合并同一时钟的 I/O 延迟	150	16.7.2	电平移位器	165
15.4.6	使用不同的时钟合并 I/O 延迟	151	16.7.3	功率目标	165
15.5	管理约束的挑战	151	16.8	小结	166
15.6	小结	153	第 17 章 XDC: Xilinx 对 SDC 的扩展		
第 16 章 其他 SDC 命令			17.1	时钟	167
16.1	工作条件	154	17.1.1	主时钟和虚拟时钟	167
16.1.1	多分析条件	155	17.1.2	生成时钟	168
16.1.2	set_operating_conditions	156	17.1.3	查询时钟	169
16.1.3	降额	157	17.1.4	时钟组	170
16.2	单位	158	17.1.5	传播时钟和延迟	171
16.3	层次分离器	158	17.1.6	时钟不确定度	172
16.3.1	set_hierarchy_separator	159	17.2	时序异常	173
16.3.2	-hsc	159	17.3	布局约束	173
16.4	设计范围	160	17.4	在 Xilinx Tcl Shell 中集成 SDC	174
16.5	导线载荷模型	161	17.5	小结	174
16.5.1	导线载荷最小尺寸	162	参考文献		
16.5.2	导线载荷模式	162	175		
16.5.3	导线载荷选择组	163			

1.1 ASIC 设计流程

典型的 ASIC 设计流程可以分为逻辑设计、物理设计两部分。逻辑设计开始于系统级设计，包括芯片架构、芯片内部结构、引脚配置、时序（设计目标的电压）和时序（设计目标的速

第 1 章

绪 论

专用集成电路 (Application-Specific Integrated Circuit, ASIC) 是针对特定应用的集成电路, 例如, 用于游戏机上运行图形界面的芯片, 消费电子产品中诸如 USB、PCI 那样的标准接口, 用于控制汽车电子产品的特殊功能芯片, 智能手机芯片等。在芯片设计早期, ASIC 只有几千个门电路。随着深亚微米技术的发展, 现在的 ASIC 可以包含数以百万计的门电路。现如今, 某些更复杂的 ASIC 还结合了处理器、存储单元、其他 ASIC 或 ASIC 派生出来的知识产权 (Intellectual Property, IP), 这称作 SoC 或片上系统。实际上现在的 SoC 将作为将来 SoC 设计的基本单元或 IP。ASIC 开发的复杂特性要求有结构良好的设计流程, 据此设计的 ASIC 可扩展性更好, 并且可为设计者和 SoC 集成商提供足够的灵活性, 同样为无缝设计定义了方法。

另一个典型的 IC 设计范例就是近年来越来越流行的现场可编程门阵列 (Field-Programmable Gate Array, FPGA)。可以用 FPGA 实现由 ASIC 开发的任何功能, 唯一不同的是 FPGA 设计的 IC 在完成生产之后还能由用户进行编程。FPGA 具有现场编程 IC 的能力, 这样就不用预定任何的硬件功能来约束用户, 并且 IC 可以根据提供的修改标准做适当调整, 降低 ASIC 不能重复设计的成本, 缩短上市周期, 但是它对性能和功耗有一定的影响。

本章将给出 ASIC 和 FPGA 设计中的典型设计流程的概况, 同时也包括时序约束对设计流程的影响。

1.1 ASIC 设计流程

典型的 ASIC 设计流程可以分为逻辑设计和物理设计两部分。逻辑设计开始于高层设计规范 and 芯片架构。芯片架构捕获高层功能、功率 (设计消耗的电量) 和时序 (设计运行的速