

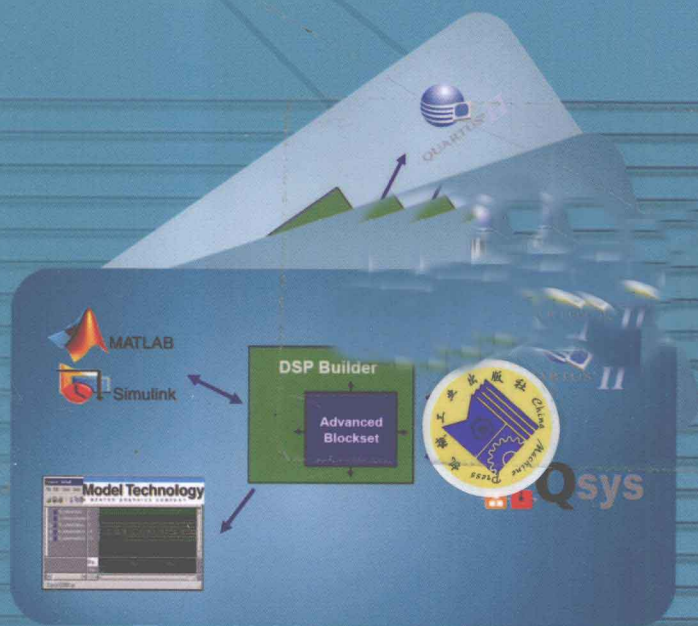
Altera 公司与 MathWorks 公司联合推荐

电气信息工程丛书

基于模型的设计

——Qsys篇

刘杰 等编著



- Qsys
- Simulink HDL Coder
- DSP Builder 标准模块库设计
- DSP Builder 高级模块库设计
- 基于模型设计与传统设计融合
- 软硬件设计融合
- 对传统设计理念的挑战

电气信息工程丛书

基于模型的设计

——Qsys 篇

刘 杰 等编著



机械工业出版社

本书以 Quartus II 11.0/DSP Build11.0 & Matlab R2011a 为软件平台,围绕基于模型设计这一核心思想,介绍了完整的传统 FPGA 开发流程、MathWorks 公司发布的基于 Simulink HDL Coder & Filter Builder 的开发 FPGA 的最新解决方案、基于标准模块库的 DSP Builder 设计方法,基于高级模块库的 DSP Builder 设计方法、基于高级模块库的设计实例。

本书可作为航天军工、通信与图像处理、FPGA/ASIC、数字信号处理等领域广大开发人员的技术手册,也可以作为高校电类专业学习 FPGA/DSP 或基于模型设计的教材。

图书在版编目(CIP)数据

基于模型的设计. Qsys 篇/刘杰等编著. —北京:机械工业出版社, 2012. 1

(电气信息工程丛书)

ISBN 978 - 7 - 111 - 37086 - 4

I. ①基… II. ①刘… III. ①微控制器—程序设计 IV. ①TP332.3

中国版本图书馆 CIP 数据核字 (2012) 第 009218 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:时 静 张丽萍

责任印制:杨 曦

北京圣夫亚美印刷有限公司印刷

2012 年 2 月第 1 版·第 1 次印刷

184mm×260mm·22.75 印张·565 千字

0001—3500 册

标准书号:ISBN 978 - 7 - 111 - 37086 - 4

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

电话服务

网络服务

社服务中心 : (010)88361066

门户网: <http://www.cmpbook.com>

销售一部 : (010)68326294

教材网: <http://www.cmpedu.com>

销售二部 : (010)88379649

封面无防伪标均为盗版

读者购书热线: (010)88379203

序 言 1

随着电子科学与集成电路制造技术的飞速发展，ASIC/FPGA 的系统开发人员难免在现代通信与信息系统的的设计中面临一系列的开发瓶颈。谁掌握了先进的设计思想与开发手段，谁就将在激烈的市场竞争中占得先机。传统手工作坊式的开发模式已很难适应目前产品的大代码量、快速上市与可靠性等的要求。为了解决这些问题，把可视化的基于模型设计思想引入到 FPGA 的应用研究之中，在国外已深入人心，并得到了广泛的应用。Altera 公司为了适应这种新技术浪潮，也实时推出并持续升级了自己的基于模型设计产品——DSP Builder 工具。特别是从 8.1 版推出 DSP Builder 高级模块库后，让 Altera 工具、知识产权 (IP)、第三方工具（比如 Simulink、ModelSim 等）；能更好地无缝连接，代码的 RTL 级验证与 Quartus II 工程的生成只是点一下鼠标的事情；参考设计支持更高效的开发过程，实现更好的结果，从而降低了系统风险；甚至在几天或几个星期内就可以实现原本需要几个月时间才能完成的 FPGA 开发任务，这大大压缩了开发周期与大幅度降低了开发成本，让推出的新产品极具竞争优势。其实施步骤可简要概括如下：

① 创建精准的算法模型：DSP Builder 模块可用于实现在 Simulink 中建模的系统构建硬件。通过采用具有位真和周期准确性的 Altera 模块集来构建模型，可以在 Simulink 环境下快速准确地评估 Altera FPGA 的硬件实现。Altera 模块集包含算术函数、逻辑函数、存储接口，以及 Avalon 内存映射和流接口。也可将经过实际验证的或手写 HDL 代码与 Altera MegaCore IP 内核的模型集成到设计中。

② 采用高级模块库优化设计：如果采用传统涉及 HDL 代码技术的 DSP 设计，在 FPGA 中要达到最佳性能，整个过程会非常耗时。DSP Builder 高级模块库能在数分钟之内在 Simulink 中创建出针对 Altera FPGA 优化的设计，且无需直接使用 HDL 代码。只需在 Simulink 模型中指定诸如所需的时钟频率和通道数等顶级设计约束，DSP Builder 将自动生成针对所选的 Altera FPGA 器件优化的流水线化的 RTL。这是由于高级库使用时分复用技术来优化逻辑利用率，同时自动插入流水线和寄存器来满足设计约束，并且提高了电路的最大频率 (Fmax)。由此得出的模块代码可获得与手工优化的 HDL 代码相近的性能，我们推荐在所有基于 DSP Builder 的设计中采用高级模块库进行。

③ 设计的验证与下载：综合用户的设计，在 Quartus II 工程中编译设计，利用自动生成的 testbench 对 HDL 代码进行 RTL 级仿真，观察其行为是否和在 Simulink 中仿真的结果一致，然后将其下载到支持的实际硬件平台上运行以验证设计的正确性。若要验证设计的实时性和加快仿真速度，可以采用硬件在环测试 (HIL)。

为了推动 DSP Builder 技术在国内的应用普及，Altera 中国区大学计划部和刘杰博士及其团队合作来完成这项工作，并荣幸代表 Altera 公司为本书作序。在书中刘杰博士介绍了许多基于模型设计的技术细节，感谢刘杰博士对 DSP Builder 在国内的应用所做的卓有成效的工作和杰出贡献。

书中紧扣 Altera FPGA 的最新技术，以刘杰博士一贯的基于模型设计思想（化整为零与

搭积木的方式)为主线,分别介绍了传统开发流程与 Qsys 的使用方法、标准模块库设计方法、高级模块库设计方法。突出了 Qsys (下一代 SOPC) + IP (Qsys 自带或第三方 IP) + Nios II 软核处理器 (这里仅起管理这些 IP 的作用) + 用户自定义 IP 新的设计理念,即可以使用 DSP Builder 将用户算法模型非常方便地打造成自定义 IP,然后将这些 IP 通过 Qsys 集成到大系统中,这将成倍提高开发效率与降低开发难度,也将引领今后 FPGA/ASICs 设计发展的潮流。

为了消除大多数用户对 DSP Builder 生成代码效率低的疑虑,刘杰博士在书中专门选用了—个极端的例子来回答这个问题。从他所得出的实验结果看:无论在资源占用上还是 Fmax 上,自动代码都可以和手写代码媲美,甚至超过手写代码的水平。

基于上述我郑重向用户推荐此书。

Altera 中国区大学计划部经理 陈卫中

2011 年 11 月 18 日于成都

序 言 2

很荣幸再次受刘杰博士之邀，并代表 MathWorks 公司为新书《基于模型的设计 - Qsys 篇》作序，同时也感谢刘老师近年来对国内基于模型设计方面所做的突出贡献和持续的推动工作。

在 DSP 篇作序时已经介绍了基于模型设计 (MBD - Model Based Design) 的系统开发理念，其主要意义在于：用 Simulink 模型描述系统和子系统物理原型，并在统一仿真环境中对整个系统进行仿真，以便及时做出设计改进。核心思想是可执行的规范、快速地控制原型设计、早期验证和代码自动生成，这已经日益成为嵌入式系统开发的主要手段。

基于模型设计的 FPGA 开发基本流程可包括 Simulink 建模、定点化处理、HDL 代码自动生成、联合仿真和硬件测试等环节，其中一主要工具就是 Simulink HDL Coder。书中充分介绍了 Simulink HDL Coder 的产品特点和最新特性。它可以把 Simulink 模型、Matlab 代码和 Stateflow 框图生成位真 (bit - true)、周期精确 (cycle - accurate)、可综合的 Verilog 和 VHDL 代码。通过使用工业标准化设计工具，可以对自动生成的 HDL 代码进行仿真和综合，并进一步映射到 FPGA 和 ASIC 芯片上。该产品经过 5 年的不断升级发展，已逐渐获得了工业界和教育界的广泛认可。

书中用很大篇幅介绍了 Altera 的 Simulink 至 FPGA 的开发流程，这在业界独具特色。Altera 与 MathWorks 是全球合作伙伴，DSP Builder 是第二代基于模型的综合技术，Simulink 是 MathWorks 基于模型设计的业界最佳工具。将 Simulink 作为高性能 DSP 设计的建模、仿真和实现环境，借助 DSP Builder 大大提高了在 Altera FPGA 上实现 DSP 时的效能。同时书中还阐述了 Qsys (SOPC Builder) + IP + Nios II 软核处理器 + 用户解决方案的新开发 FPGA 的方法，以及在现有工程中添加新功能的方法，这对于加快现有或传统设备的升级换代很有借鉴意义。

在同一本书中，在同一个基于模型设计的环境中，尝试使用不同开发流程是本书的最大特色。当前，无论是市场上，还是高校教学中，都缺少系统的中文教材或书籍以帮助广大技术人员、在校的研究生和高年级本科生尽可能全面学习整个 MBD 开发流程。我相信随着本公司和 Altera 公司对基于模型设计技术的进一步深入研究和刘杰博士《基于模型的设计 - Qsys 篇》这本书的正式出版，将大大帮助老师和学生学习和实践高效的嵌入式系统开发，丰富目前国内的 FPGA 开发模式。因此我郑重向各位推荐此书。

MathWorks 中国教育业务发展总监

陈炜博士

2011 年 11 月

前 言

前面的序言部分 Altera 和 MathWorks 公司的两位嘉宾对基于模型设计开发 FPGA 的优势与特点已经谈得比较透彻了，在此不再重复这些表述。我主要谈谈本书的特点、建模方法与阅读本书需要具备的知识。

一、本书的核心思想

近年来 Altera 公司一直推崇软硬件结合的方法开发基于 FPGA 的产品，即 SOPC Builder 或 Qsys + Nios II 软核处理器的开发模式。这里，用 Qsys 集成软件自带的 IP、第三方提供的算法 IP 和用户自定义（解决方案）IP。其中用户自定义 IP 是用户在开发新产品时需要解决的关键问题，它也是本书讨论的核心问题，本书就是围绕如何使用 DSP Builder 工具创建用户 IP 而展开的。因此用户必须了解 Nios II 的相关知识，熟悉 Qsys 提供的 API 函数的使用方法，但这些已经超过了本书的范围，在阅读本书时，请用户首先学习基于 Nios II 软核处理器的软件开发。

Altera 公司为基于 DSP Builder 高级库的设计提供了面向不同应用领域的 80 多个实例，为基于标准库的设计提供了 20 个左右的范例。一般地，用户只要根据自己的实际应用，参照这些范例对模型中的参数或个别算法模块稍做修改就可以轻松实现自己的算法模型。然后利用书中介绍的方法打造用户解决方案 IP，最后用 SOPC Builder 或 Qsys 集成这些解决方案 IP 完成整个系统的设计。

二、创建用户算法模型的方法

基于模型设计的核心就是算法模型，简单来说就是用户能用 Simulink 或 DSP Builder Blockset 中的模块实现其头脑中的一些想法。那么这就要求用户必须具备坚实的数学基础和丰富的工程背景，以及能熟练使用 Matlab/Simulink/Stateflow/DSP Builder Blockset 工具搭建算法模型。这就要求用户必须掌握对一个系统的建模方法，例如公式法（正向）和数据驱动法（逆向）等。只要用户能用微分方程或数学公式解析法表述一个真实的物理系统，那么这个系统一般是可以 Simulink 自带的基本模块实现的。如果用户觉得这样的建模方式所搭建的模型不太优化（实现同一个功能所用的模块越多，生成 HDL 代码的效率当然就越低），这时可以对给出的微分方程或数学公式进行数学变换，将其化简成为一个合理的数学表达形式，例如，用状态空间表述微分方程等。如果在设计中找不到相应的模块实现算法，这时可以手写 HDL 代码来实现这部分功能，然后用 Black Box 模块将其打包添加到所搭建的系统模型中即可。在设计中如果遇到一些常用算法，这时模块库中又没有直接实现这些算法的模块，如果再花大量的时间对这部分单元建模就没有必要了，这时可以用 Black Box 模块将这些经过实践检验的 HDL 代码打包添加到模型中来。

三、传统开发模式与基于模型设计的关系

自 DSP Builder Blockset 出现高级模块库之后，虽然在顶层模块中也可以对整个设计的自动流水线、寄存器和 F_{max} 等进行设置，但它已经和过去的标准模块库设计有很大的不同。在高级库中，除了自动生成算法设计的 Quartus II 工程与利用 ModelSim 对模型代码进行 RTL

级验证之外，其他大部分的工作都留给传统方法完成。因此用户必须学会用传统方法开发 FPGA 的流程，特别是要掌握 ModelSim 和 SignalTap II 的正确使用方法。为了使读者能在阅读本书之前，对基于模型设计的开发流程有一个清晰的了解，这里简单介绍如下：①对所创建的算法模型在 Simulink 中进行功能仿真，以验证是否实现了预期目标；②在代码层面上验证能否实现用户的设计，即利用 ModelSim 对于模型代码进行 RTL 级仿真，以验证生成的 HDL 代码能否实现在 Simulink 中实现的功能；③对算法进行硬件在环测试（添加 HIL 模块）；④采用在线逻辑分析仪对算法的硬件实现进行分析（添加 SignalTap II 模块）；⑤对这个设计进行硬件测试。如果得出的结果无法满足需求分析规定的技术指标，将对所创建的算法模型重新进行设计或在顶层重新设计约束参数，然后重复上述过程直到满意为止。如果用一句话来表达的话，基于模型设计就是在可视化的环境中从概念到实现一步完成，规避传统方法在开发 FPGA 时一般需重复多次的弊端。

四、用户关心的采用基于模型设计的效率问题

这是大家谈论比较多的问题，也是为什么 DSP Builder 很少被大家接受的关键问题。目前在国内外的网络上普遍流传着采用基于模型设计所生成的 HDL 代码占用资源较多的传闻，为了回答读者这个问题，我们特别选了一个极端的范例来探讨这个问题。实验条件和结论如下：

① 手写 HDL 代码 21 行，综合后的结果为：占用资源 41 个 LE； $F_{\max} = 87 \text{ MHz}$ 。

② 用 DSP Builder 生成的 HDL 代码大于 220 行，综合后的结果为：占用资源 36 个 LE； $F_{\max} = 80 \text{ MHz}$ 。

③ 用 Simulink HDL Coder 生成的 HDL 代码大于 260 行，综合后的结果为：占用资源 54 个 LE； $F_{\max} = 77 \text{ MHz}$ 。（参看第 3 章）

从表面看，DSP Builder 生成的 HDL 代码长度是手写 HDL 代码长度的 10 倍；但从综合后的结果看，这个结论就站不住脚了，这个颠覆性的结论很好地回答了大家有关效率的问题。

除了本人之外，郭丹、周宇博全程参与了本书的设计与撰写工作，翁公羽、林寒毅、史进、刘高阳、林建材、周贵鸿、杨云飞、张德勇、李晗、陈添丁、李昌、吴仪炳、杨元挺、王衍龙、张一丁、黄凤英、吴文龙、刘宗铭、徐林、傅金雨、陈炳飞、余洪波、林淑玲、黄涛、万君、欧阳峰、易凤忍、陈玉岭、杨村明、陈财淦、张灵峰、刘翔、吴元树、魏源、林敏辉、温永杰、朱伟龙、许家胜也参与了个别章节的编写与资料收集工作。

在本书的撰写过程中，得到了 Altera 中国区大学计划部经理陈卫中的大力支持；衷心感谢 MathWorks 公司中国教育业务发展总监陈炜博士对本人研究基于模型设计的帮助和在百忙之中为本书所作的序言；感谢机械工业出版社时静编辑对本书提出的宝贵意见，正是大家的共同努力才促成了本书的出版。

本书的大部分内容以 Altera 公司提供的资料为蓝本进行介绍，不过，最核心的部分为我们实验室原创。由于资料多、时间紧、作者水平有限，对外文的理解肯定存在偏差。本书只起到抛砖引玉的作用，不妥之处恳请读者指正。

刘杰
2011 年 11 月

目 录

序言 1

序言 2

前言

第 1 章 Altera 系列 FPGA 的传统开发	1
1.1 FPGA 的传统开发流程	1
1.2 建立一个工程	2
1.2.1 启动 Quartus II 软件	2
1.2.2 创建工程目录	2
1.2.3 打开 Quartus II 新工程向导	2
1.2.4 设置工程目录、名称以及顶层模块	2
1.2.5 添加设计文件	3
1.2.6 指定芯片	3
1.2.7 配置第三方工具	4
1.3 设计输入	5
1.3.1 添加设计文件	5
1.3.2 配置芯片属性	6
1.3.3 语法分析	6
1.4 仿真和验证	7
1.4.1 添加 Altera 仿真库	8
1.4.2 行为仿真 (RTL 级仿真)	11
1.4.3 功能仿真	16
1.4.4 时序仿真	18
1.5 下载运行	20
1.6 用 SignalTap II 观测信号	21
1.6.1 新建 SignalTap II 文件 (stp)	21
1.6.2 设置 JTAG 链	22
1.6.3 添加观察信号	22
1.6.4 SignalTap 信号配置	22
1.6.5 观察信号波形	23
1.7 时序约束 TimeQuest	24
1.7.1 使用 TimeQuest	24
1.7.2 执行时序分析	28
1.7.3 查看时序分析结果	28
1.8 优化	28

1.8.1	全局优化	28
1.8.2	综合优化	29
1.8.3	适配优化	30
1.8.4	使用优化向导	31
1.9	功耗分析	32
1.10	用图形方式开发	33
1.10.1	生成自己的模块	33
1.10.2	新建图形设计文件	34
1.10.3	添加设计模块	34
1.11	Qsys 系统集成工具入门	38
1.11.1	Qsys 的设计流程	39
1.11.2	Tutorial 简介	40
1.11.3	创建 Qsys 功能模块系统	43
1.11.4	整合层次化系统	59
1.11.5	在系统控制台进行硬件验证	76
1.11.6	对自定义组件进行仿真	83
第2章	Simulink HDL Coder & Filterbuilder	97
2.1	基于 Simulink HDL Coder 的最新开发流程	97
2.1.1	搭建流水灯的功能模型	97
2.1.2	流水灯的代码模型	105
2.1.3	联合仿真	106
2.1.4	自动生成代码并建立工程	110
2.1.5	SignalTap II 测试	116
2.1.6	硬件测试	119
2.2	基于 filterbuilder 的滤波器设计实验	122
2.2.1	选择要设计的滤波器类型	122
2.2.2	设计高通滤波器参数	122
2.2.3	查看所设计的高通滤波器响应曲线	123
2.2.4	查看滤波器的纹波系数	124
2.2.5	查看滤波器的冲激响应与极零点	124
2.2.6	查看生成的滤波器系数	125
2.2.7	创建定点模型	125
2.2.8	生成 VHDL 代码	127
2.2.9	速度优先与面积优先	128
2.2.10	创建高通滤波器模块	128
2.2.11	模型的功能验证	129
2.2.12	用 Fixed-Point Advisor/Fixed-Point Tool 作定点化处理	132
2.2.13	重新对定点模型进行功能验证	142
2.2.14	生成符合要求的高通滤波器代码	143

2.2.15 对生成的 VHDL 代码进行 SignalTap II 测试	143
第 3 章 DSP Builder 标准模块库设计	144
3.1 DSP Builder 标准模块库常用模块简介	145
3.1.1 Signal Compiler 模块	145
3.1.2 SignalTap II Logic Analyzer 模块	146
3.1.3 TestBench 模块	146
3.1.4 LUT (Look - Up Table) 查找表模块	147
3.1.5 Delay 模块	148
3.1.6 Input 模块	149
3.1.7 Output 模块	150
3.1.8 Clock 模块	150
3.1.9 HDL Import 模块	151
3.1.10 Avalon - MM Master 模块	153
3.1.11 Avalon - MM Slave 模块	156
3.1.12 Avalon - MM Read FIFO 模块	158
3.1.13 Avalon - MM Write FIFO 模块	160
3.2 DSP Builder 的开发流程	161
3.2.1 Qsys & DSP Builder 的设计流程	161
3.2.2 DSP Builder 标准库设计的特点	162
3.2.3 使用 DSP Builder 标准库的情况	162
3.3 添加 DSP Builder 设计到现有工程	162
3.3.1 用 DSP Builder 标准库实现流水灯	163
3.3.2 模型的功能验证	171
3.3.3 硬件在环测试	171
3.3.4 SignalTap II 测试	172
3.3.5 硬件测试	175
3.3.6 在 Quartus 中设计数码管流动	183
3.3.7 在现有工程中添加 DSP Builder 模型功能	189
3.4 手写代码与模型自动代码所占资源的对比	196
3.5 硬件在环测试 (HIL)	198
3.5.1 边缘检测的简介	198
3.5.2 HIL 的测试方法	199
3.5.3 Burst 模式	200
3.5.4 HIL 图像边缘检测设计实例	201
3.6 集成手写或遗留 HDL 代码	207
3.6.1 隐式黑盒接口	207
3.6.2 显式黑盒接口	207
3.6.3 黑盒集成示例	207
3.6.4 SOPC Builder 集成 DSP Builder Design	223

3.6.5	Avalon - MM FIFO 设计实例	252
3.7	SOPC Builder + IP + Nios II + DSP Builder Design	263
3.7.1	创建流水灯模型	263
3.7.2	在 SOPC 系统中集成模型 IP	266
3.7.3	Nios II 软件设计	267
第 4 章	DSP Builder 高级模块库设计	271
4.1	使用 DSP Builder 高级模块库的情况	271
4.2	DSP Builder 高级模块库设计流程	272
4.2.1	创建 TestBench	274
4.2.2	硬件实现	274
4.2.3	设置系统参数	274
4.2.4	硬件生成	276
4.2.5	硬件验证	277
4.2.6	高级模块库开发流程图示	277
4.3	DSP Builder 高级模块库常用模块介绍	278
4.3.1	基本模块库 (Base Blocks)	279
4.3.2	ModelIP 库	287
4.3.3	ModelBus 库	288
4.3.4	基原库 (ModelPrim)	289
4.3.5	FFT 库	292
4.4	DSP Builder 高级模块库设计规则	293
4.4.1	周期精确与延时	293
4.4.2	连接模块协议	294
4.4.3	时分复用 (TDM) 方法	294
4.4.4	参数定义	295
4.4.5	矢量化数据 I/O	295
4.4.6	连接 ModelIP 模块	295
4.4.7	ModelIP 模块的延时显示	296
4.4.8	基原子系统的延时显示	296
4.4.9	基原子系统的延时约束	297
4.4.10	ModelIP 模块的延时约束	297
4.4.11	延时与 Fmax 约束冲突	298
4.4.12	连接 ModelIP 模块与 ModelPrim 子系统	298
4.5	Fibonacci 模型设计	298
4.5.1	Fibonacci 数列的由来	299
4.5.2	Fibonacci 模型的创建	299
4.5.3	在 Simulink 中进行模型的功能仿真	306
4.5.4	修改 fibonacci 模型	307
4.5.5	输出生成的文件	309

4.5.6	在 ModelSim 中进行代码的 RTL 级验证	309
4.5.7	硬件在环测试	310
4.5.8	SignalTap II 测试	310
4.5.9	创建 Quartus II 工程	310
4.5.10	将 fibonacci 设计集成到 Qsys 中	314
4.6	数字下变频 (DDC) 系统设计	316
4.6.1	DDC 原理介绍	316
4.6.2	DDC 系统建模	319
4.6.3	在 Simulink 中进行功能验证	330
4.6.4	输出生成的文件	331
4.6.5	在 ModelSim 中进行 RTL 级代码验证	333
4.6.6	硬件在环测试	334
4.6.7	SignalTap II 测试	334
4.6.8	创建 Quartus II 工程	334
4.6.9	将 DDC 设计集成到 Qsys 中	336
4.7	高级模块库与标准模块库混合使用	339
4.7.1	同时使用这两种模块的情况	339
4.7.2	构建数字上变频混合模型	339
4.7.3	在高级模块库中进行硬件在环测试	345
参考文献		351

第 1 章 Altera 系列 FPGA 的传统开发

无论是采用 Simulink HDL Coder 还是 DSP Builder 高级模块库进行 FPGA 的应用研究，从 RTL 仿真之后的工作一般都采用传统模式去完成。那么，能否熟练掌握传统的 FPGA 开发流程，熟练使用 Quartus II 11.0 软件、ModelSim 仿真、SignalTap II 在线逻辑分析仪、功率分析等工具是基于模型设计的关键所在。本书的核心是讲述 Qsys + DSP Builder Design 的思想，而 Qsys 是 Altera 公司在 Quartus II 11.0 刚发布的新功能（SOPC Builder 的下一代产品）。在 Quartus II 11.0 以后的版本中，SOPC Builder 工具将逐步淡出甚至被取消，因此本章以 Altera 公司提供的 tutorial 为用例大量的篇幅介绍了 Qsys 的使用方法。

本章的主要内容包括：

- 1) 介绍 FPGA 的传统开发流程。
- 2) 基于图形的开发模式。
- 3) Test Bench 的编写。
- 4) ModelSim 仿真。
- 5) SignalTap II 测试。
- 6) TimeQuest 时序分析。
- 7) 优化。
- 8) 功耗分析。
- 9) Qsys 系统集成工具入门。

1.1 FPGA 的传统开发流程

本小节将以一个简单的流水灯为例子，介绍整个开发过程，如图 1-1 所示。

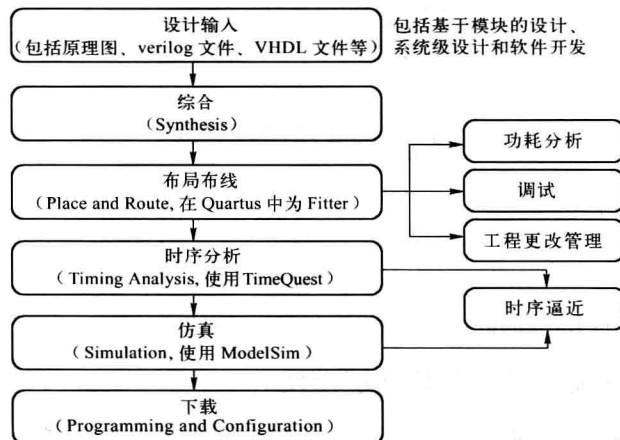


图 1-1 FPGA 开发流程

1.2 建立一个工程

1.2.1 启动 Quartus II 软件

安装 Quartus II 软件后，在桌面上单击 Quartus II 的快捷方式或单击“开始”→“所有程序”→Altera→Quartus II 11.0，启动 Quartus II 软件。启动后的界面如图 1-2 所示。

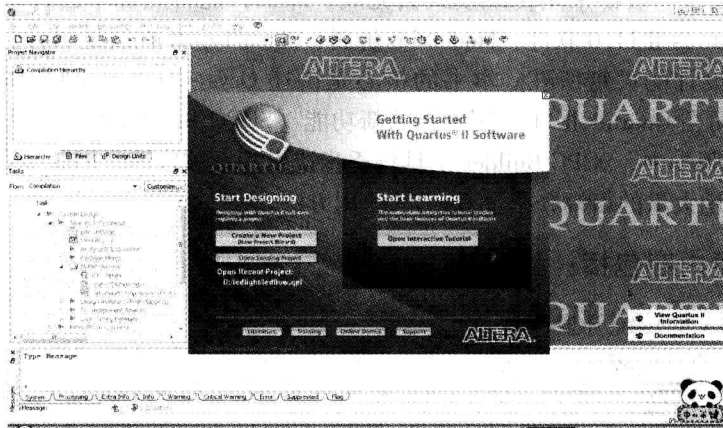


图 1-2 Quartus II 启动界面

1.2.2 创建工程目录

开发 Altera 的 FPGA，需要先新建一个工程，可以先新建一个空文件夹，用于存放所有关于该工程的相关文件。因为这里做的是流水灯试验，所以将该文件夹命名为 ledlight。

1.2.3 打开 Quartus II 新工程向导

启动 Quartus II 软件后，单击 File 菜单中的 New Project Wizard 选项，打开新建工程向导，如图 1-3 所示。

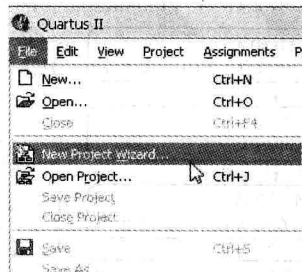


图 1-3 新建工程

在新建工程向导的 Introduction 页面中单击 Next 按钮进入下一步。

1.2.4 设置工程目录、名称以及顶层模块

在 Introduction 的下一个页面中，需要对这个新建的工程进行配置。将工程目录指向刚才新建的空文件夹 ledlight，并命名该工程，如图 1-4 所示。

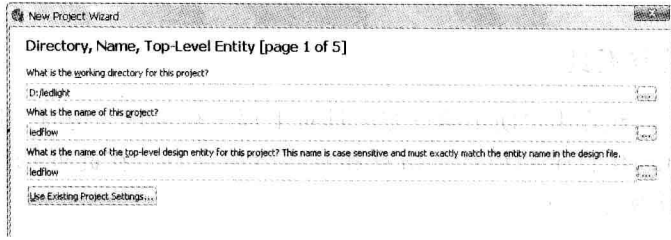


图 1-4 工程属性设置

完成后单击 Next 按钮进入下一步。

1.2.5 添加设计文件

设置完工程的目录、名称和顶级模块后，单击 Next，进入 Add Files 页面，如图 1-5 所示。在这里可以将已有的设计文件加入工程中，若没有也可以不添加，直接单击 Next 按钮，进入下一步。

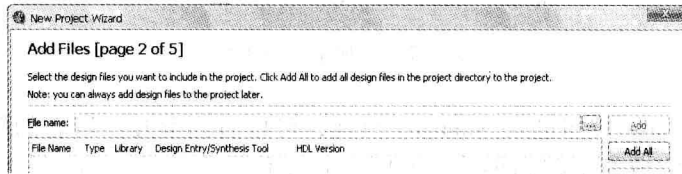


图 1-5 添加设计文件

1.2.6 指定芯片

在完成了设计文件添加的步骤之后，单击 Next，进入 Family & Device Settings 页面。在这个页面中，需要为新建的工程指定相应的芯片型号和参数，所选取的芯片型号将在完全编译时将工程设计映射到相应的逻辑资源上。由于本次 tutorial 使用 Altera DE2 开发板进行，此开发板的核心芯片为 Cyclone II 系列的 EP2C35F672C6 芯片，因此在新建工程时就可直接指定 EP2C35F672C6 为本次 tutorial 的 FPGA 芯片，如图 1-6 所示。

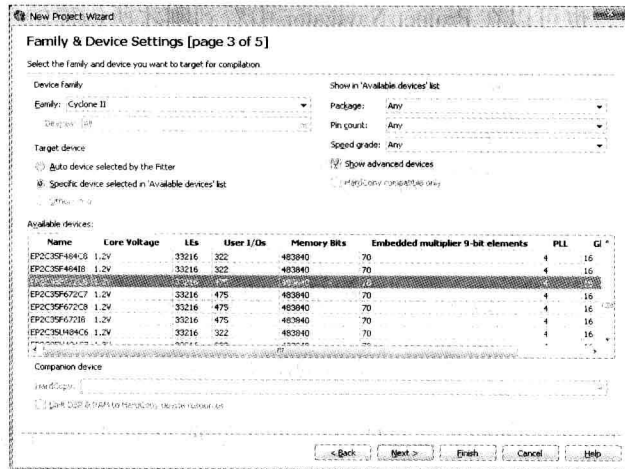


图 1-6 指定 FPGA 芯片

1.2.7 配置第三方工具

在 Quartus II 中，允许用户使用第三方工具对工程进行设计、仿真，以及时序分析。当然也可以使用 Quartus II 自带的设计、仿真和时序分析工具。在此处选用第三方仿真工具 ModelSim，如图 1-7 所示。

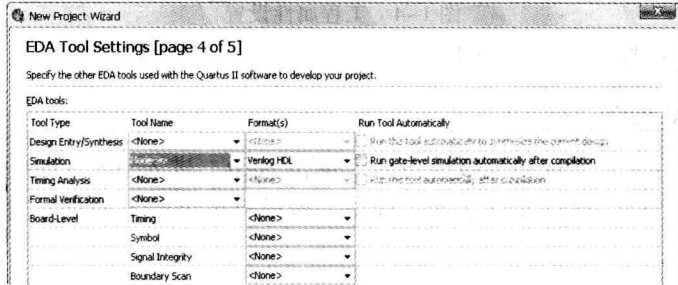


图 1-7 添加第三方工具

完成之后单击 Next 按钮进入 Summary 页面，可以看到前几步所有的配置，确认正确后单击 Finish 按钮，完成新工程的创建。此时工程文件夹中将出现 db 子文件夹、ledflow.qpf 文件和 ledflow.qsf 文件，如图 1-8 所示。在 db 子文件夹中存放的是 ledflow 工程的一些基本信息。

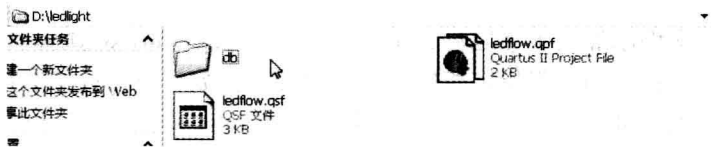


图 1-8 工程目录中的有关文件

Quartus II 软件中，也可以在 Project Navigator 中看到刚才所创建的工程，如图 1-9 所示。

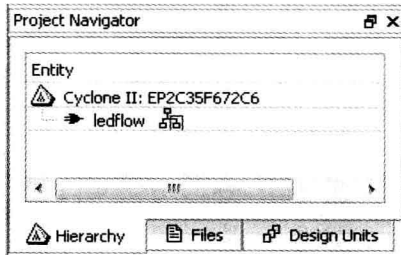


图 1-9 新建的流水灯工程

通过下方的 Hierarchy Files Design Units 可以分别查看工程结构、工程文件和设计单元。