高性能FPGA系统—时序设计与分析

崔嵬　王巍　编著

高等教育出版社

高性能FPGA系统——时序设计与分析

崔 嵬 王 巍 编著

Gaoxingneng FPGA Xitong

阜阳师范学院图书馆 藏书

高等教育出版社·北京

内容简介

本书全面系统地讨论了高性能 FPGA 时序设计、分析的基本原理与实现方法。全书共分 7 章：第 1 章以 Xilinx FPGA 为例，对 FPGA 的设计流程进行了概述；第 2 章对 FPGA 时序参数定义、流水线与并行处理设计技术、时序路径分类、时钟的非理想性（时钟偏斜与时钟抖动）等进行了探讨；第 3 章主要讨论了 FPGA 时序约束设计要点，包括 Xilinx FPGA 时序约束语法规则、时序约束分组方法以及不同路径的时序约束方法等；第 4 章介绍了 FPGA 时序约束分析的原理和方法，主要包括周期约束分析、偏移约束分析、时钟偏斜和时钟不确定性分析，此外还介绍了时序分析器 Timing Analyzer 的基本使用方法；第 5 章介绍了 FPGA 时序收敛的流程，分析了代码风格以及逻辑综合优化对时序收敛的影响，指出了有助于提高 FPGA 性能的设计方法；第 6 章和第 7 章分别以 Xilinx Spartan－3 与 Virtex－5/6 系列 FPGA 为例，对面向时序性能的 FPGA 逻辑综合技术进行了深入的探讨。全书条理清晰，内容先进，讲解透彻，便于自学。

本书可作为信息与通信工程、电子科学与技术、计算机科学与技术、控制科学与工程或相关专业的高年级本科生和研究生的教材，同时也是从事 FPGA 技术研究与微电子技术研究、生产及应用的工程技术人员的重要参考书。另外，对于其他专业想了解高性能 FPGA 时序设计与分析的工程技术人员，也是一本很有价值的参考书。

图书在版编目（CIP）数据

高性能 FPGA 系统——时序设计与分析/崔嵬，王巍编著. --北京：高等教育出版社，2014.7

ISBN 978－7－04－039849－6

Ⅰ.①高… Ⅱ.①崔… ②王… Ⅲ.①可编程序逻辑器件－系统设计 Ⅳ.①TP332.1

中国版本图书馆 CIP 数据核字（2014）第 095895 号

策划编辑	平庆庆	责任编辑	平庆庆	封面设计	赵 阳	版式设计	童 丹
插图绘制	杜晓丹	责任校对	刘 莉	责任印制	田 甜		

出版发行	高等教育出版社	咨询电话	400－810－0598
社 址	北京市西城区德外大街 4 号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	北京四季青印刷厂	网上订购	http://www.landraco.com
开 本	787mm×1092mm 1/16		http://www.landraco.com.cn
印 张	14.25	版 次	2014 年 7 月第 1 版
字 数	320 千字	印 次	2014 年 7 月第 1 次印刷
购书热线	010－58581118	定 价	22.70 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

版权所有 侵权必究

物 料 号 39849－00

前言

作为一种可编程逻辑器件，FPGA 在短短二十多年中从电子设计的外围器件逐渐演变为数字系统的核心。伴随半导体工艺技术的进步，FPGA 器件的设计技术取得了飞跃发展及突破。随着 FPGA 向更高密度、更大容量、更低功耗和集成更多 IP 的方向发展，系统设计人员在从这些优异性能获益的同时，不得不面对由于 FPGA 前所未有的性能和能力水平而带来的新的设计挑战。如何实现快速的时序收敛、降低功耗和成本、优化时钟管理并降低 FPGA 设计的复杂性等问题，一直是 FPGA 设计者需要面对的关键问题。

考虑到 FPGA 技术的发展情况和未来趋势，编写一本对 FPGA 时序设计与分析的基本知识进行介绍以及对 FPGA 性能优化方法进行探讨的教科书是非常必要和迫切的。目前关于 FPGA 的技术类书籍和教材品种较多、内容广泛，对普及 FPGA 技术，推动相关领域的技术进步起到了积极的作用。但目前的 FPGA 技术类书籍和教材多面向基础应用，内容主要为基于 FPGA 的数字系统基本设计方法，设计用例多偏重于单元、模块设计，缺少顶层的 FPGA 设计与优化策略介绍，特别是对高性能 FPGA 时序设计的探讨不深入，且部分教材的技术实践内容多直接取材于国外 FPGA 厂商手册和技术解决方案。

我和王巍老师近些年一直工作在教学与科研一线，在高性能 FPGA 的时序设计与分析方面积累了一点经验。在教学实践中不断探索教学理论与工程实践相结合的教学模式，使教学过程成为融理论及创新实践于一体，让学生在探索的过程中学习，在学习的过程中实践，在实践的基础上创新的过程，该教学模式获得了学生的广泛好评。此外，本书的两位作者均受聘为中国电子学会 FPGA 技术高级研修班授课教师，在长期的与 FPGA 高级研发人员的交流与探索中，深刻体会到掌握高水平的 FPGA 设计技术不仅需要设计者掌握各自研究领域的专业知识，还需要掌握 FPGA 特有的资源构成、VLSI 原理及设计方法、数字信号处理、计算机体系结构等多学科、多领域知识，更需要通过时序设计与分析深入理解 FPGA 设计的本质，并在工程实践中不断提升认知水平。

在教学与相关科研工作的基础上，我们编写了这本《高性能 FPGA 系统——时序设计与分析》，由该书主体内容构成的相关讲义与内部教材已在作者讲授的研究生课、本科生课中先后试用多次，内容的新颖性与实用性获得了学生的充分肯定；在课程教学过程中，及时根据反馈情况以及相关技术的发展对教材内容进行了多次修订，不断将国际上该领域的最新研究成果充实到

教材内容中，成为研究生、本科生入门和了解该方向前沿进展的重要资源。故本书既可以作为信息与通信工程、电子科学与技术、计算机科学与技术、控制科学与工程或相关专业高年级本科生的教材，也可以作为研究生的教学参考书。本书对从事 FPGA 技术研究、生产及应用的工程技术人员以及其他专业想了解高性能 FPGA 时序设计与分析相关知识的师生和工程技术人员而言，也是一本有价值的参考书。

本书第 1 章、第 2 章、第 3 章、第 4 章的第 1 至第 4 节由北京理工大学的崔嵬老师编写，第 4 章的第 5 至第 8 节、第 5 章、第 6 章与第 7 章由天津工业大学的王巍老师编写。崔嵬对全部初稿进行了修改与定稿，统编全书。北京航空航天大学的张弘教授审阅了本书的全稿，并提出了许多宝贵的意见。我的学生王凤云、李霖、金倩玉、詹天祥，王巍老师的学生殷俊、代静、李莹、张美杰等帮助绘制了部分插图。我们还参阅并引用了 FPGA 设计技术有关著作以及美国 Xilinx 公司设计手册的部分内容，在此一并表示感谢。另外，感谢高等教育出版社为本书的出版做了大量工作。

限于编者水平，加之时间仓促，书中谬误之处在所难免，恳请各兄弟院校的师生及广大读者批评指正。作者 Email：cuiwei@ bit. edu. cn；wangweibit@ 163. com

崔　嵬

2014 年 1 月

试读结束：需要全本请在线购买：　www.ertongbook.com

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话　(010) 58581897　58582371　58581879
反盗版举报传真　(010) 82086060
反盗版举报邮箱　dd@hep.com.cn
通信地址　北京市西城区德外大街 4 号　高等教育出版社法务部
邮政编码　100120

目录

第1章　FPGA设计流程概述

FPGA(Field Programmable Gate Array)设计一般采用自顶向下方法,即从系统级设计开始,划分为若干个二级单元,然后再把各个二级单元划分为下一层次的基本单元,一直下去,直到能够使用基本模块或者IP核直接实现为止。流行的FPGA开发工具都提供了层次化管理,可以有效地梳理错综复杂的层次,能够方便地查看某一层次模块的源代码以修改错误。

1.1　FPGA设计流程[1,2]

FPGA的设计流程就是利用开发软件和编程工具对FPGA芯片进行开发的过程。高层次的FPGA设计流程包含了需求分析、结构设计、实现以及验证。FPGA的设计流程如图1-1-1所示。在需求分析阶段,定义并完善高层次的需求,这一阶段的任务是完成系统功能的说明。下一阶段是结构设计阶段,这一阶段进行器件厂商、器件型号和开发工具的选择。该阶段需要对设计中固定的功能模块与可编程模块进行划分,并对软硬件的执行进行划分。结构设计阶段完成后

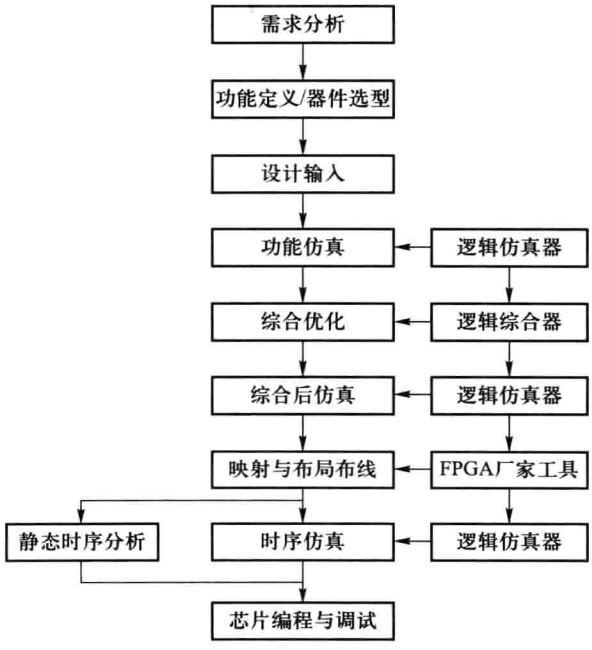

图1-1-1　FPGA的设计流程

是设计实现阶段。设计实现阶段包括设计输入、功能仿真、综合优化、综合后仿真、实现、布线后仿真等主要步骤。在紧接着的验证阶段中，为了确保系统需求的正确执行，需要对设计进行测试。表 1-1-1 列出了不同设计阶段的主要行为。

表 1-1-1　不同设计阶段的主要行为

设计阶段	主要行为
需求定义阶段	定义并完善高层次工程项目的详细功能和性能需求
结构设计阶段	选择功能实现技术；选择器件厂商、器件型号和开发工具；定义系统架构，考虑设计实现的可升级性；分割固定的功能模块与可编程模块；定义设计模块功能和接口
实现阶段	实现整个设计，设计输入、复查、约束、整合；初始设计仿真、时序验证、报告分析
验证阶段	设计测试、时序验证、必要的设计更改；产生下载到目标板的配置文件；在目标板上调试和验证功能；使用基于 FPGA 的嵌入式逻辑分析仪测试功能

清晰、完整并合理的需求定义将使结构设计的效率得到提高。细致的结构设计有利于设计模块的划分与实现。如果很好地对设计模块进行划分并对其功能、性能和接口都给出定义，那么这个设计模块的开发、实现、复查、评估和测试就更加容易。越清楚有效地开发、执行并测试设计，进行整合、改进与维护也就越容易。如果这个有效的开发顺序能够被持续不断地执行下去，就能够使设计周期最小化。下面将对设计的主要阶段所必须完成的任务进行描述。

1.1.1　需求定义阶段

设计需求说明书：定义并详细描述功能需求、接口、性能以及设计余量。修订并保持设计需求说明书，使其成为“活生生的文档”。

1.1.2　结构设计阶段

系统工程化和模块划分：为设计划分功能块、分配功能性与分析性能需求。在 FPGA 设计开始之前，必须有系统功能的定义和模块的划分，即编制任务书。另外，就是要根据任务要求，如系统的功能和复杂度，对工作速度和器件本身的资源、成本以及连线的可布性等方面进行权衡，选择合适的设计方案和合适的器件类型。一般采用自顶向下的设计与划分方法。

1.1.3　实现阶段

1. 设计输入

设计输入是将所设计的系统或电路以开发软件要求的某种形式表示出来，并输入给 EDA 工具的过程。常用的方法有原理图输入方法和硬件描述语言（HDL）。原理图输入方法是一种最

试读结束：需要全本请在线购买：　www.ertongbook.com

直接的描述方式，它将所需的器件从元件库中调出来，画出原理图。这种方法直观并易于仿真，但设计效率低，且不易维护，不利于模块构造和重用。另一方面，可移植性差，当芯片升级后，所有的原理图都需要进行一定的改动。目前，在实际开发中应用最广的是 HDL 输入法，利用文本描述设计，可以分为普通 HDL 输入法和行为 HDL 输入法。普通 HDL 有 ABEL、CUR 等，支持逻辑方程、真值表和状态机等表达方式，主要用于简单的小型设计。在中大型工程中，主要使用行为 HDL，其主流语言是 Verilog HDL 和 VHDL，其共同的突出特点有：语言与芯片工艺无关，利于自顶向下设计，便于模块的划分与移植，可移植性好，具有很强的逻辑描述和仿真功能，且输入效率很高。除了 IEEE 标准语言外，还有厂商自己的语言。此外，也可以用 HDL 为主、原理图为辅的混合设计方式，以发挥两者的各自特色。

2. 功能仿真

功能仿真也称为前仿真，是 FPGA 设计过程中最基本的仿真验证，主要目的是验证设计文件的逻辑功能是否正确，是否满足设计要求。此时的仿真没有延迟信息，处于理想状态，仅对功能进行检测。仿真前，要先利用波形编辑器和 HDL 等建立波形文件和测试向量（即将所关心的输入信号组合成序列），仿真结果将会生成报告文件和输出信号波形，从中便可以观察出各个节点信号的变化。如果发现错误，则返回设计修改逻辑设计。常用的功能仿真工具有 Model Tech 公司的 ModelSim、Sysnopsys 公司的 VCS 和 Cadence 公司的 NC - Verilog、NC - VHDL 等软件。

3. 逻辑综合

逻辑综合是将较高级抽象层次的描述转化成较低层次的描述，即将设计输入编译成由**与**门、**或**门、**非**门、RAM、触发器等基本逻辑单元组成的逻辑连接网表，而并非真实的门级电路。真实具体的门级电路需要利用 FPGA 制造商的布局布线功能，根据逻辑综合后生成的标准门级结构网表来产生。为了能转换成标准的门级结构网表，HDL 程序的编写必须符合特定综合器所要求的风格。常用的综合工具有 Synplicity 公司的 Synplify 软件、Synopsys 公司的 FPGA Compiler 软件以及各个 FPGA 厂家自己推出的综合开发工具。在 FPGA 设计过程中，设计的综合效果主要取决于代码设计风格和综合工具的综合能力。

4. 综合后仿真

综合后仿真检查综合结果是否和原设计一致。仿真时，通过把综合生成的标准延时文件反标注到综合仿真模型中，来估计门延时带来的影响。这一步骤不能估计线延时，和布线后的实际情况存在一定的差距，对于一般的设计可以省略这一步，但如果在布局布线后发现电路结构和设计意图不符，则需要回溯到综合后仿真来确认问题之所在。在功能仿真中介绍的软件工具一般都支持综合后仿真。

5. 映射与布局布线

布局布线是利用实现工具把逻辑映射到目标器件可用的资源中，决定逻辑的最佳布局，选择逻辑与输入、输出功能链接的布线通道进行连线，并产生相应文件（如配置文件与相关报告）。映射是将综合生成的逻辑网表对应到具体的 FPGA 芯片功能单元上。布局将逻辑网表中的硬件原语和底层单元合理地配置到芯片内部的固有硬件结构上，并且往往需要在速度最优和面积最

优之间做出选择。布线根据布局的拓扑结构，利用芯片内部的各种连线资源，合理正确地连接各个元件。目前，FPGA 的结构非常复杂，特别是在有时序约束条件时，需要利用时序驱动的引擎进行布局布线。布线结束后，软件工具会自动生成报告，提供有关设计中各部分资源的使用情况。由于只有 FPGA 芯片生产商对芯片结构最为了解，所以布局布线一般选择芯片开发商提供的专用工具。

6. 时序仿真

时序仿真也称为后仿真，是指将布局布线的延时信息反标注到设计网表中来检测有无时序违规（即不满足时序约束条件或器件固有的时序规则，如建立时间、保持时间等）现象。一旦 FPGA 设计在目标器件上完成了布局布线，相应的逻辑延时和走线延时就会被反馈到固定的数据库文件中。使用更新后的数据库文件进行仿真，从而验证布局布线后的动态时序。时序仿真包含的延迟信息最全，也最精确，能较好地反映芯片的实际工作情况。由于不同芯片的内部延时不一样，因此不同的布局布线方案会给延时带来不同的影响。在布局布线后，通过对系统和各个模块进行时序仿真，分析其时序关系，估计系统性能以及检查和消除竞争冒险是非常有必要的。如果仿真结果显示由于延时影响而造成逻辑错误，就需要在设计输入时对关键电路进行设计约束，可直接在设计输入中修改受影响的路径或利用设计约束文件加以限制，最终消除延时对电路的影响。在功能仿真中介绍的软件工具一般都支持布局布线后仿真。

7. 静态时序分析

传统上通过对布局布线后的门级电路进行动态仿真（时序仿真）来验证一个设计的功能和时序。随着设计规模的增大，时序仿真所需要的测试向量数量以指数规律增长，且这种方法不能保证足够的测试覆盖率。在大型 FPGA 设计中，如果仅采用传统的时序仿真方法，则仿真时间与工作量都难以承受。

静态时序分析（Static Timing Analysis，STA）可以简单地定义为：设计者提出特定的时序要求，并利用足够精确的时序模型对 FPGA 布局布线后的门级电路进行时序分析，以使得 FPGA 时序满足设计需求。

静态时序分析提供了一种针对大规模 FPGA 时序验证的有效解决方法，它可以检查设计中路径的时序，测试覆盖率达到 100%，从而有效地降低了时序验证的复杂度。静态时序分析不需要任何测试向量，分析所需要的时间也远小于时序仿真所需要的时间。但静态时序分析不能完成设计功能验证，设计功能验证还必须通过功能仿真（注意，不是时序仿真）来实现。另外，如果设计中包含部分异步电路，则通过时序仿真对异步电路进行验证更为直观方便。一般来说，一个设计的时序验证应包含寄存器传输级（RTL）的功能仿真、静态时序分析以及时序仿真。静态时序分析和时序仿真各有优点，互相补充，一起使用可以有效地保证 FPGA 设计的正确性和可靠性。

1.1.4 验证阶段

在验证阶段中，确认测试包括芯片编程与调试。芯片编程是指产生使用的位数据流文件，然后将位数据流文件下载到 FPGA 芯片中。芯片编程需要满足一定的条件，如编程电压、编程时序和编程算法等方面。逻辑分析仪是 FPGA 设计的主要调试工具，但需要引出大量的测试引脚，且逻辑分析仪价格昂贵。目前，主流的 FPGA 芯片生产商都提供了内嵌的在线逻辑分析仪（如 Xilinx ISE 中的 ChipScope、Altera Quartus II 中的 SignalTap II 以及 SignalProb）来解决上述矛盾，它们只需要占用芯片少量的逻辑资源，具有很高的实用价值。

1.2 基于 FPGA 的 SoPC 设计方法[3,4]

目前，由于 FPGA 性能提升价格下降，同时嵌入越来越多的内核，因此很自然地，很多 IC 设计公司将 FPGA 用于 ASIC 原型验证，把 FPGA 可编程的优点带到了 SoC 领域，其系统由嵌入式处理器内核、DSP 单元、大容量存储器、高速收发器、混合逻辑、IP 以及原有的设计部分组成。SoPC 平台的核心部分是内嵌的处理内核，其硬件是固定的，软件则是可编程的；外围电路则由 FPGA 的逻辑资源组成，大都以 IP 的形式提供，例如存储器接口、USB 接口以及以太网 MAC 层接口等，用户根据自己的需要在内核总线上添加，并能自己订制相应的接口 IP 和外围设备。

以往的 SoC 设计面向 ASIC 设计流程，通常采用全定制和半定制电路设计方法，开发周期长、费用大，一旦设计投片完成，就不能再改动。而 SoPC 则面向可编程逻辑器件设计流程，设计灵活便捷，不仅性能、速度、连接具有优势，而且可以随时修改，在缩短产品上市时间以及节省成本等方面的优势非常明显。在现代电子系统设计中，可编程逻辑器件尤其是 FPGA 呈现出一枝独秀的增长态势，越来越多地成为系统级芯片设计的首选。世界顶级 PLD 制造商如 Xilinx、Altera、Atmel 和 Quick Logic 等公司都已推出了 SoPC 芯片，其设计性能及性价比已完全能够与 ASIC 抗衡，如 Xilinx 公司的高性价比 FPGA 产品 Virtex－6 和 Spartan－6、Altera 公司的 Stratix III 和 Cyclone III。

1.2.1 基于FPGA 的典型SoPC 开发流程

SoPC 芯片拥有丰富灵活的 IP 资源，包括功能强大的 DSP 核，可以在许多领域替代通用 DSP。加快 FPGA 实现的一个方法是采用 IP 模块。

SoPC 还拥有动态可重构技术，可以随时改变硬件和软件，升级维护方便。例如，Xilinx 公司的 Virtex 系列平台级 FPGA 提供了在系统部分可重构功能，可实现某个信号处理 IP 功能在线重构，而不影响其他信号处理 IP 的工作。Xilinx 公司还提出了另一种全新重构理念，可以通过 In-

ternet 实现对远程 FPGA 系统的软硬件重构，从而提供更加优越的性能。单芯片动态可重构信号处理器实质上是一个可以完成复杂数字逻辑功能的可编程软硬件平台，其最大的优点就是设计灵活、功能强大，可以在逻辑资源允许的条件下构建任意复杂的数字系统。

基于 FPGA 的典型 SoPC 开发流程如下。

1. 芯片内设计

从设计生成开始，设计人员需要从软硬件协同设计的思路入手，找出只能在系统集成阶段才会被发现的软、硬件缺陷，然后选择合适的芯片以及开发工具，在综合过程得到优化，随后进行精确的实现，以满足实际需求。由于设计规模越来越大，工作频率也到了数百兆赫兹，布局布线的延迟将变得非常重要。为了确保满足时序，需要在布局布线后进行静态时序分析，对设计进行验证。

2. 板级验证

在芯片内设计完毕后，需要再进行板级验证，以便在印刷电路板（PCB）上保证与最初设计功能一致。因此，PCB 布局以及信号完整性测试应被纳入设计流程。由于芯片内设计所做的任何改变都将反映在下游的设计流程中，因此各个过程之间的数据接口和管理也必须是无误的。预计 SoPC 系统以及所必需的额外过程将使数据的大小成指数增长，因此，管理各种数据集本身是极具挑战性的任务。

1.2.2 SoPC 的开发环境

根据 SoC 设计流程、SoC 自身的特点以及软硬件协同设计的一般流程，基于 FPGA 的 SoPC 集成设计环境可以归纳为三个层次，如图 1-2-1 所示。这个集成环境构成典型的软硬件协同设计集成平台，由三个不同层次、不同功能的 EDA 集成设计环境组成。下面结合 Xilinx 和 Altera 两大 SoPC 厂商的开发工具对这三个层次进行说明。

1. 系统级开发环境

系统级开发环境主要用于完成 SoPC 的系统级设计。首先，需要根据客户的要求，进行系统功能定义和性能评估，以便确定系统规约。其次，根据已经确定的系统规约，应用系统级描述语言（C++或 SystemC 等）进行系统设计描述与设计验证，以便确定所定义的系统规约在功能上是否可以实现。再次，在证明了系统规约可以实现后，进行系统软硬件功能划分，以便确定系统的哪些功能是由软件系统完成的，哪些功能是由硬件系统完成的，哪些功能需要软硬件协同完成。对于既可以通过软件系统完成也可以通过硬件系统完成的功能，需要进行性能与成本的评估。

进行系统设计与验证可以使用各种权威的工具软件，如 Cadence 公司的 Virtual Component Co-Design（VCC）、Synopsys 公司的 CoCentric、Synopsys 公司的 COSSAP、Cadence 公司的 Processing Worksystem 等。

另外，System-Generator 和 DSP-Builder 分别是 Xilinx 和 Altera 公司的 DSP 系统生成工具，

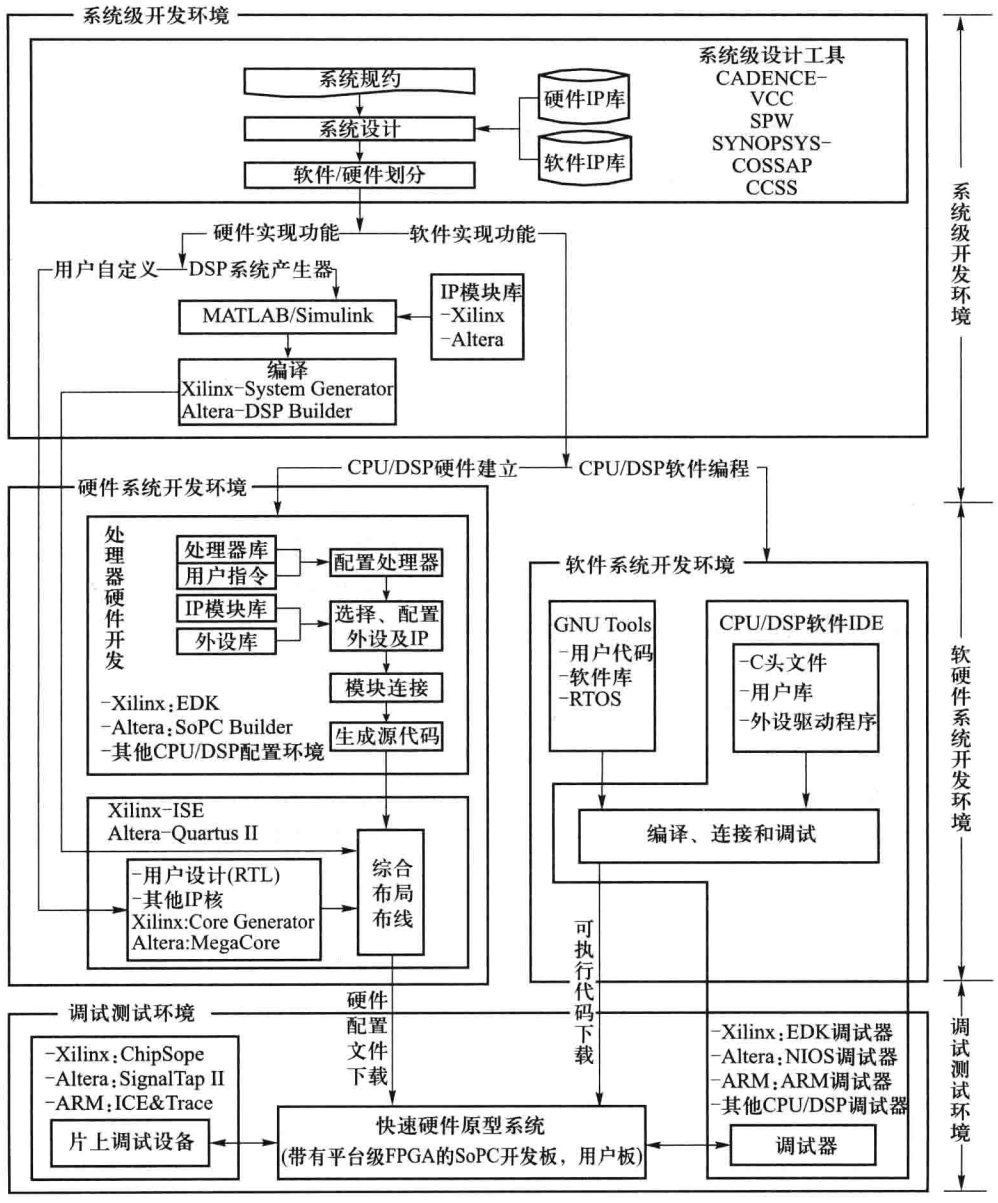

图 1-2-1　SoPC 开发环境示意图

可看成是 Simulink 的一个程序包，能自动把在 MATLAB 和 Simulink 上生成的系统设计翻译成可靠的、可综合的有效硬件网表。

2. 软件系统和硬件系统开发环境

软件系统和硬件系统开发环境主要用于完成 SoPC 的软件系统设计和硬件系统设计。根据

系统级设计中的功能划分，分别进行 SoPC 的硬件系统设计和软件系统设计，硬件系统设计和软件系统设计是并行进行的。

硬件系统通常包括：

① DSP 系统生成器生成的硬件模块，一般是算法映射出来的纯粹硬件模块。

② 处理器(CPU/DSP)硬件模块，指软件运行的硬件平台，用于运行系统软件或应用软件。通常由厂家提供的嵌入式系统开发工具生成，如 Xilinx 公司的 EDK 和 Altera 公司的 SoPC Builder。

③ 用户自定义硬件模块和其他 IP 模块。前两种途径不能提供的功能，由用户自行使用 HDL 语言开发相应的硬件模块，或使用厂商提供的 IP 实现。

在软件系统设计中，通常使用 SoPC 厂商提供的软件集成开发工具，如 Xilinx 的 EDK 环境、Altera 的 Nios II 环境和 ARM 的 Realview 环境。

在软硬件系统设计过程中，为了确保系统的性价比达到最优，需要在软硬件协同验证环境中不断进行软硬件协同设计，从而使得所实现的芯片级电子系统的性价比达到最优。

3. 系统调试与测试环境

SoPC 的主要逻辑设计是在以 BGA 封装为主的可编程逻辑器件内部，传统的调试设备已很难进行直接测试分析，从而对以仿真技术为基础的软硬件协同设计技术提出更高的要求，使新的调试技术不断涌现出来，如 Xilinx 公司的片内逻辑分析仪 ChipScope Pro 和 Altera 公司的 Signal-Tap II。

ChipScope Pro 实际上是将逻辑分析仪 ILA 和总线分析仪 IBA 核心嵌入到设计中。这些嵌入核心允许用户观察所使用的 FPGA 器件中的所有内部信号和节点。另外，只需要使用 FPGA 上的 JTAG 接口或极少量引脚，ChipScope Pro 还可以实现与安捷伦公司 FPGA 跟踪端口分析仪的接口，从而为用户提供更大容量的跟踪存储器空间、更快的时钟速度和更多的触发选择。

试读结束：需要全本请在线购买： www.ertongbook.com