电子设计与嵌入式开发实践丛书

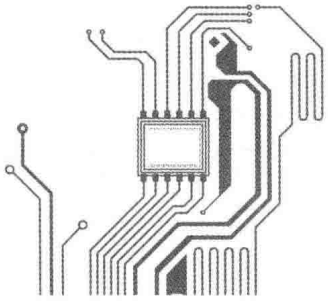
勇敢的芯 伴你玩转Nios II

◎ 吴厚航 编著

- 基于Altera Cyclone IV FPGA，由浅入深地引领读者从嵌入式系统设计的大处着手，玩转软核处理器Nios II
- 从入门到精通，全面展示相关基础知识、电路设计要点、Qsys硬件系统架构、外设连接、软件编程、量产固化等设计细节
- 提供PPT课件和源码，配套开发板，可通过网站论坛和书友会与作者互动

清华大学出版社

电子设计与嵌入式开发实践丛书



勇敢的芯
伴你玩转Nios II

◎ 吴厚航 编著

清華大學出版社
北京

内 容 简 介

本书使用 Altera 公司的 Cyclone Ⅳ FPGA 器件，由浅入深地引领读者从嵌入式系统设计的大处着手，玩转软核处理器 Nios Ⅱ。基于特定的 FPGA 实验平台，既有足够的理论知识深度作支撑，也有丰富的例程进行实践学习，并且穿插着笔者多年 FPGA 学习和开发过程中的各种经验和技巧。

对于希望快速入手嵌入式系统软硬件开发的初学者，以及希望从系统层面提升嵌入式开发能力的学习者，本书都是很好的选择。

图书在版编目(CIP)数据

勇敢的芯伴你玩转 Nios Ⅱ/吴厚航编著. --北京：清华大学出版社，2016
(电子设计与嵌入式开发实践丛书)
ISBN 978-7-302-43784-0

Ⅰ. ①勇… Ⅱ. ①吴… Ⅲ. ①微处理器－系统设计 Ⅳ. ①TP332

中国版本图书馆 CIP 数据核字(2016)第 100113 号

责任编辑：刘　星
封面设计：刘　键
责任校对：徐俊伟
责任印制：沈　露

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：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：12.75　　**字　　数**：319 千字
版　　次：2016 年 8 月第 1 版　　**印　　次**：2016 年 8 月第1次印刷
印　　数：1～2500
定　　价：45.00 元

产品编号：069445-01

前言

2015年底的智能硬件展上，和FPGA原厂的两位大学计划经理闲聊的当儿，被问及对于诸如Xilinx Microblaze与Altera Nios Ⅱ这样的FPGA内嵌软核处理器的看法时，笔者的第一反应便是“它的实用价值可能并不大，但是非常具有教学价值”。此话一出，大家也一定很好奇，且听笔者娓娓道来。

笔者以为，软核处理器哪怕是在今天的Zynq或SoC FPGA（内嵌多个硬核ARM Cortex-A9处理器的FPGA器件）还未面世之时，它的实用性其实就一直颇具争议。硬核处理器经过优化设计，已经流片成型，用户拿到手以后，一般也无法再更改处理器本身的性能参数，其使用量通常也非常大，可靠性、稳定性都会做得很好；而反观FPGA中内嵌的软核处理器，其本身就不是针对任何一个特定器件型号的FPGA定制的，而是FPGA器件的一个“通用”软核，因此，它在FPGA器件上跑起来势必在性能上也会大打折扣，与此对应的，其可靠性、稳定性恐怕也欠佳。当然，并不是说在FPGA器件上跑起来的软核处理器就一定差强人意，笔者并没有一棍子打死的意思，并不排除某些发烧级FPGA设计者能够从时序设计和底层布局布线上将软核处理器的性能发挥到极致的情况，只是一般比较来看，硬核处理器确实在性能、成本、可靠性等方面相对于软核处理器都有更明显的优势。因此，放眼望去，很难找到有多少电子产品中用上了软核处理器，但与此相反的是硬核处理器则无处不在。

话说回来，FPGA器件中内嵌的软核处理器也并非一无是处，否则它就没有存在的意义了，话说“存在即是合理”。一点不假，FPGA器件中的软核处理器是可定制的，它的性能水平通常有多个可选项供设计者“编程”设定，并且其周边的外设也可以完全“定制化”，从这点来看它比硬核处理器要灵活很多。某些特别需要这种“灵活定制”的场合通过软核处理器还真是“门当户对”了，只是要玩转这样一个灵活的嵌入式处理器系统也并非易事，它涉及纯粹的软、硬件设计以及FPGA等多方面的知识，一个能真正玩转软核处理器“灵活性”的FPGA设计者，一定对处理器及其外设架构了然于心。换句话说，如果大家都还记得当年大学里面那门纯理论的“微机原理”课程，那么玩转软核处理器的过程就是“活生生”的实践版“微机原理”课程的再现。

说到这，或许读者有些明白了，没有错，之所以说FPGA器件的软核处理器具有很高的教学价值，就是基于它的实践过程中能够帮助读者对整个处理器的架构有更清楚的了解和

Foreword

认知。比如笔者在学习 Nios Ⅱ处理器的过程中，需要将处理器的数据总线、指令总线和外设进行连接；需要分配地址；需要连接中断；需要自己编写外设连接到总线上；需要揣摩外设的寻址方式、读和写时序，甚至一些常见通信接口的时序……的确，掌握了这些东西，很大程度上就能帮助学习者对嵌入式系统的整个架构有了一个更全面的认知，这些体验是传统理论书本给不了的。除此以外，它也是很多正式产品开发调试过程中的好帮手，例如在很多产品的原型开发或测试验证阶段，恰巧需要一个简单的 CPU 干点活，这时软核处理器也就派上用场了。

以笔者自身的经历来说，也正是通过软核处理器的“磨练”，才对软硬件的认知有了很大的提升，虽然这些年多从事偏于硬件设计方面的工作（包括一些 FPGA 设计工作），但是在很多的调试过程中，尤其是需要软硬件协同调试的过程中，往往能够快速地区分和定位问题是出于软件还是硬件，甚至还能够协助软件工程师解决一些具体的问题。笔者以为，软核处理器的教学意义在于，它能够帮助学习者深入了解处理器系统设计的架构。而在如今的电子产品设计中，软硬件分工越来越细，很难在实际的开发过程中跨越“鸿沟”，但是具备这样系统性设计思维的工程师，必定是“人见人爱”的。

基于以上这些考量，笔者在第一本 Nios Ⅱ图书《爱上 FPGA 开发——特权和你一起学 Nios Ⅱ》出版五年后（由于书中的平台较旧，考虑到市场因素，第一次印刷售罄后就没有复印），决定重新梳理这方面的知识，在 Quartus Ⅱ的 Qsys 平台上大干一场，同时借助 Altera Cyclone Ⅳ FPGA 入门平台“勇敢的芯”（可访问淘宝网店了解该 FPGA 平台详情：https://myfpga.taobao.com/），和大家一起重拾玩转 Nios Ⅱ嵌入式处理器的激情。

作　者

2016 年 1 月于上海

目录

Contents

第1章

基于 Nios Ⅱ 处理器的嵌入式系统

1.1 片上系统概述

数字电路高度集成化是现代电子发展的大势所趋，片上系统(SOPC)的概念也就应运而生。它是指在单个芯片上集成一个完整的系统，一般包括系统级芯片控制逻辑模块、微处理器/微控制器内核模块、数字信号处理器模块、存储器或存储器控制模块、与外部通信的各种接口协议模块、含有 ADC/DAC 的模拟前端模块、电源及功耗管理模块，它是一个具备特定功能、应用于特定产品的高度集成电路。

片上系统其实就是系统小型化的代名词。如图 1.1 所示，一个相对复杂的系统采用传统的设计方案可能需要一个 CPU 做整体控制，一个 FPGA 做接口的逻辑粘合和一些信号的预处理，还需要一个 DSPs 做复杂的算法实现，Flash 和 SDRAM 分别作为程序存储器和数据缓存器，此外还会有一些专用的外设模块，这些器件都放置在一块或者数块电路板上。这样一个系统显得相当繁杂，不仅调试难度大，而且系统维护也不方便。

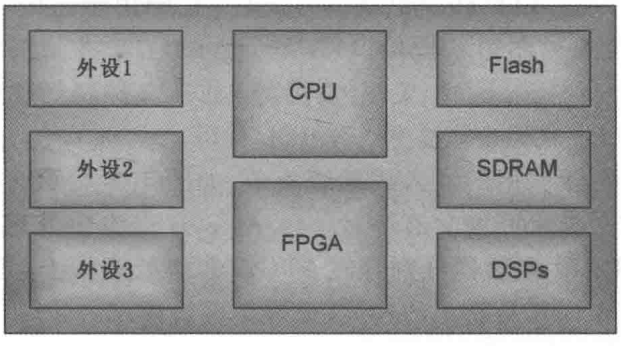


图 1.1 传统的复杂系统

基于 FPGA 的片上系统提出了这样一种解决方案：如图 1.2 所示，FPGA 内部集成了 CPU、DSPs 以及各种接口控制模块，对有些存储量要求不大的系统甚至集成了外部的 Flash 和 SDRAM。

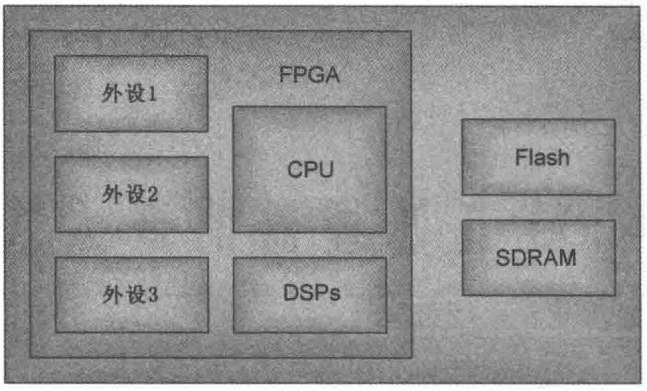


图 1.2 基于 FPGA 的片上系统

可以看出,SOPC 就是一颗比 MCU 更强大的 MCU。它的伟大之处在于系统的完全自主定制性,有了 SOPC,设计者就不需要再拿着选型手册海选既必须具有这个外设又必须满足那个条件的处理器了;甚至有时都不需要考虑处理器都能够挂上什么样的存储器来读写数据、运行程序。只要有 SOPC,一切就能轻松搞定,想加什么外设就加什么,一切由你做主。这就是 SOPC 相对于以往的嵌入式系统设计最大的特点和优势。

SOPC 需要一个强大的系统开发工具。Altera 的 FPGA 开发工具 Quartus Ⅱ中集成的 Qsys 可以帮助用户定义并生成一个完整的片上可编程系统(System-on-Programmable-Chip),它比传统的手动集成方式要方便得多。Qsys 中可以添加各种 Altera FPGA 器件可以使用的硬核或软核处理器、常用外设以及用户自定义的定制外设,非常灵活方便。Qsys 使用起来就如同小朋友们的乐高积木一样简单,并没有传说中的那么"高深",只要大家跟着教程一步一个脚印往下走,相信大家很快就可以玩转基于 Qsys 的 Nios Ⅱ嵌入式处理器系统。

用户可以使用 Qsys 生成一个基于 Nios Ⅱ处理器的嵌入式系统。然而,Qsys 远不止一个 Nios Ⅱ处理器而已,它还可以生成一个不包含处理器或者包含 Nios Ⅱ以外的软核处理器的系统。

使用传统的设计方法,用户必须手动编写 HDL 代码用于连接各个子系统。而使用 Qsys,用户只要通过傻瓜的图形界面接口(GUI)就可以自动生成各个组件的互连逻辑。Qsys 生成了系统所有组件的 HDL 文件,顶层的 HDL 文件则例化好系统的所有组件。Qsys 既能够生成 Verilog 代码也能够生成 VHDL 代码。

再看一个更接近实际应用的嵌入式系统板卡,如图 1.3 所示。在这块 PCB(Printed Circuit Board)上,单论芯片可能只有一片 FPGA、一片作为协处理器(Co-Processor)的 CPU、两片 DDR2 SDRAM 存储器分别挂在 FPGA 和协处理器上,还有一个叫作总线桥(Bus Bridge)的接口(也许只是简单的连线,也许是一块协议芯片)。这个系统中,看似简单,其实不然,FPGA 里大有文章可做。

从总线桥开始说,通俗地理解,桥就是用来连接河两岸的,比如主板上的南桥和北桥(CPU 和内存居然还隔着条河?)。CPU 很好很强大,可以处理海量数据,但是再强大也没法发出声音、显示图像,术业有专攻,CPU 就是负责数据运算和控制,别的基本不管。因此,CPU 需要通过桥和外围设备进行信息交互,把需要进行处理的数据接收进来,把处理完的

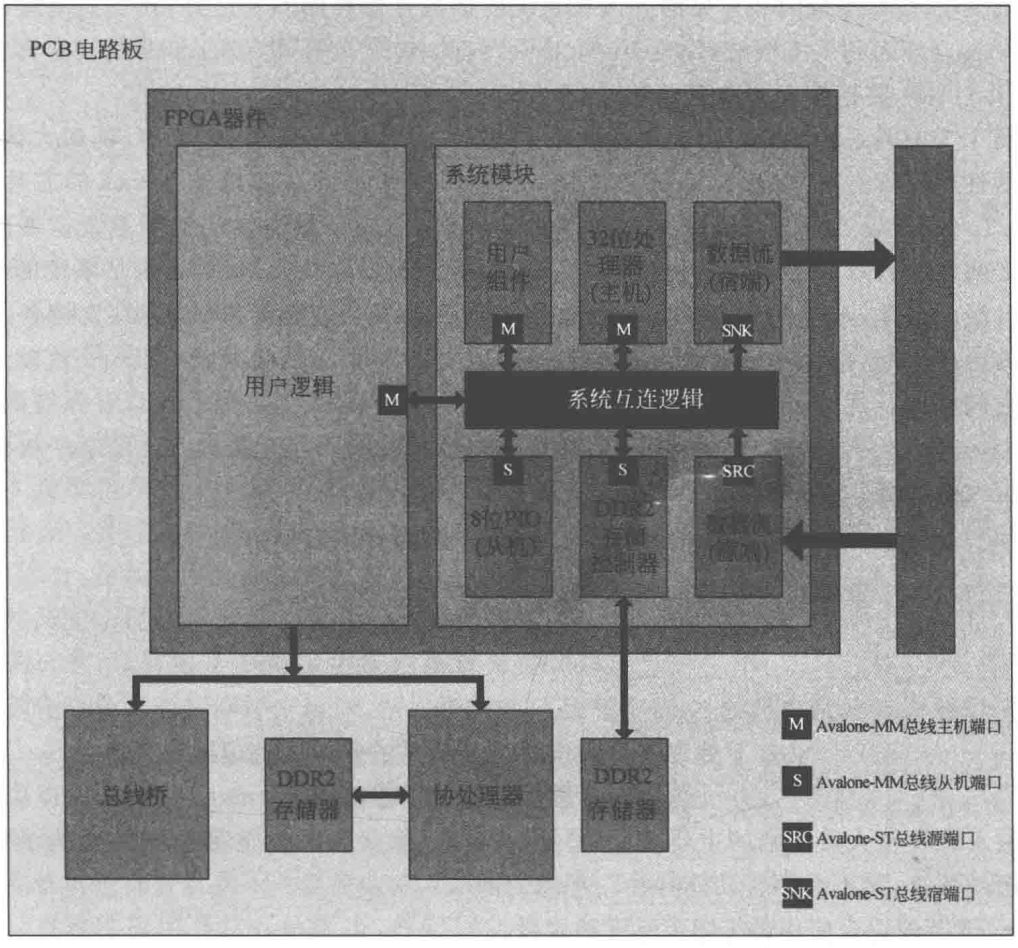


图 1.3　FPGA 上的 SOPC 系统实例

数据发送出去，可能说得不是很专业，但是基本就是这样。那么总线又是什么？CPU 的引脚终归是有限的，如果一个 CPU 要和所有外设都搭个“独木桥”，恐怕 CPU 要像巴掌那么大才够在肚子底下容下那么多“脚”了。这么看，这个桥真不能是独木桥，至少该是一座纵横南北的“立交桥”，再形象一点说，这座立交桥的交错中心点是贯通的，处于这个中心点的车可以通过处于任何高度的道路驶向四面八方。那么，CPU 就处于这样的核心位置。这里不再深入，总线其实就是 CPU 的一组满足一定协议的引脚的集合，这组引脚可以和多个同样满足这个特定协议的不同外设进行连接。当 CPU 要用这个总线和某个外设交互信息时，就会在它们之间搭起一座独木桥，其他外设就只能望桥兴叹。总线从某种意义上看就是为了节约引脚而出现的，当然从另一种意义上看，也是为了统一信息交互方式。这里 FPGA 外面挂了个“总线桥”，用于这个系统和外部设备交互，其实 FPGA 内部的 SOPC 也有个总线桥，它的名字叫作 Avalon，Avalon 总线，以后大家越使用它越会发现它的强大。上面提到 Nios Ⅱ只是一个处理器，而 Avalon 总线就是要把 Nios Ⅱ和所有其他在 FPGA 内（如图 1.3 中的 PIO）甚至 FPGA 外定制的外设（如图 1.3 中的 DDR2 存储器）连接起来。当然也可以理解那个系统互连逻辑（System Interconnect Fabric）就是 Avalon，但是 Avalon 只是系统互连逻辑的一种形式而已，想了解其他的形式就得读者自己参考数据手册。

其实 FPGA 系统内和常见的嵌入式系统的架构有着异曲同工之妙，比如协处理器(Co-Proccessor)外面和系统模块(System Module)内的 32 位处理器(Proccessor，可认为它就是 Nios Ⅱ)一样，要挂接 DDR2 存储器(Memory)。

图 1.3 中的系统里最核心的东西就是 FPGA。用 FPGA 搭一个 SOPC 的最大优势，就是灵活性。传统的 51 单片机系统常常是一个 MCU 旁边挂很多诸如 74xxx 的芯片，而当 16 位、32 位 CPU 在嵌入式系统中大行其道的时候，用户依然常常困扰于系统扩展的各种接口之间无法有效的得到控制和管理，甚至会感觉引脚数量受限，大大影响了系统的性能和扩展升级。因此，基于 FPGA 的 SOPC 被推上前台。因为它有足够的引脚，支持各式各样不同的电压标准，可以并行处理各类复杂任务，可以像一张白纸任大家涂画……这就是大家学习它的理由。其实图 1.1 和图 1.2 已经完全阐释了 FPGA 上 SOPC 与以往系统的不同，基本上一片 FPGA 就可以集成大多数常用的外设，无论从 BOM 成本上还是电路板面积上都有很大的优势。

1.2 Nios Ⅱ的优势在哪里

先看看 Altera 为自己的 Nios Ⅱ产品打的广告：

Nios Ⅱ处理器——世界上最通用的嵌入式处理器

迅速构建最合适的处理器系统

嵌入式开发人员面临的主要挑战，是如何选择一款最合适的处理器，既不会为了提高性能而超过预算，又不会牺牲功能特性。理想的嵌入式解决方案：

- 选择最适合应用的 CPU、外设和接口；
- 现场远程更新，保持竞争，满足需求的变化；
- 不必改动电路板设计，提升性能——针对需要的功能进行加速；
- 避免处理器和 ASSP 过时的风险；
- 将多种功能在一个芯片中实现，降低了总成本、复杂度和功耗。

通过最合适的 CPU、外设和存储器接口，以及定制硬件加速器，达到每一新设计周期的独特目标，Nios Ⅱ处理器以极大的灵活性满足了设计者的需求。

坦白地说，作者也不知道世界上用得最多的嵌入式处理器到底是哪款，但是却赞同“Nios Ⅱ是世界上最通用的嵌入式处理器”这句话。所谓通用，就是有很强的兼容性，在不同的项目、不同的应用中都具有一定的适用性。SOPC 本来就是为“通用”而生的，Nios Ⅱ更是加快了它的通用性步伐。

前面也提到了，在面对一个新项目时，设计者在评估处理器、选型时往往需要考虑很多问题，例如处理器的速度、性能是否满足运算需求？支持的存储器是否满足代码量、数据量的存储需求？是否满足对各种不同外设的需求？是否有足够的可扩展接口？支持何种电平标准……实际情况往往不是这款处理器速度太慢，就是那款处理器外设太少，最终的解决办法通常就是使用多个内核进行互补式的级联。要知道，在电路板上多一块芯片，就多一点面积、多一点成本、多一个不稳定因素。百万门甚至千万门 FPGA 的出现，足够让用户架构一个很强大的 CPU 系统，因此，这个系统的灵活性、通用性也会异乎寻常地让人为之振奋。

在 Altera 的这个系统中，Nios Ⅱ是当之无愧的主角。不可否认，Nios Ⅱ的优势在于它所依托的 FPGA 上架构起来的 SOPC 系统。

1.3 基于 Nios Ⅱ处理器的 FPGA 开发流程

基于 Nios Ⅱ处理器的开发流程如图 1.4 所示。Qsys 是 Quartus Ⅱ中集成的一个工具，在 Qsys 中搭建 Nios Ⅱ处理器系统，并添加和配置各种外设。随后在 Quartus Ⅱ和 EDS 开发工具中分别进行 FPGA 和嵌入式软件设计。FPGA 设计包括设计输入（Verilog/VHDL 代码编写等）、设计约束（引脚约束、时序约束等）、编译（综合、布局布线）和生成配置文件。嵌入式软件开发则包括 C 源码创建、编写、编译和最终的调试运行（前提是 FPGA 生成的配置文件已经预先烧录到目标器件中）。

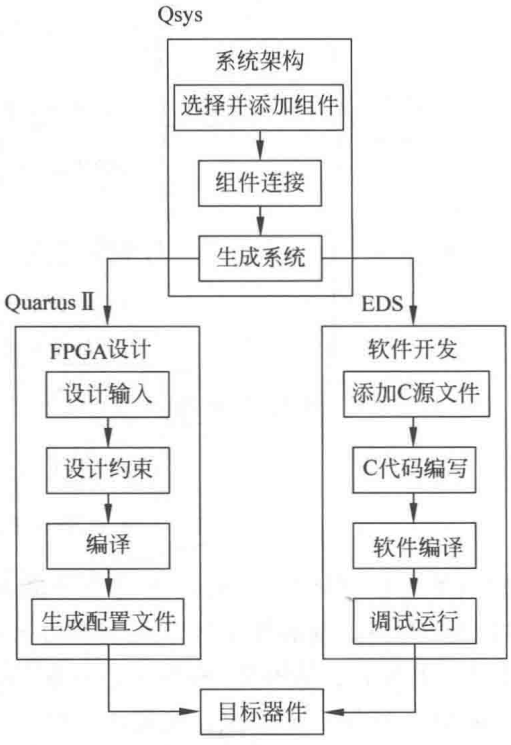


图 1.4 SOPC 系统开发流程

Qsys 生成系统后，会自动产生所有外设相关的驱动，包括在 system.h 中定义系统各个外设的基址，自动编译各种可供调用的函数。衔接软硬件的这部分就是 HAL，中文名叫硬件抽象层。在软件应用开发人员看来，他们只要弄明白 HAL 提供的所有可用函数的用法就可以玩转整个系统了。

按照笔者的理解，其实完全可以抛开图 1.4。对于通常相对简单的 Nios Ⅱ处理器开发项目而言，如图 1.5 所示，用脑图总结出来的一些基本步骤就足以代表在整个项目中所涉及的主要方面。

关于图 1.5 本身就不过多进行分析了，这里只是罗列一些步骤供读者参考。当然了，它要和前面的图 1.4 相比就显得有些“不规范”和“不官方”了，姑且可以称之为“草根流程”。在很多场合下，其实也是没有条件和办法去完完全全“规范化”开发流程的，但这并不妨碍对流程的正确理解和有条件地执行。总之，设计者要记住一件事：所有规范和流程的制定，都是为了更好的服务于产品开发。

另外，FPGA 开发设计的迭代性特点决定了基于 Nios Ⅱ处理器的软硬件开发同样存在着这个特点，也许这一点在图 1.4 和图 1.5 中都没法很好地表现出来。所谓迭代性，就是重复性，当开发到某一个环节时如果出现问题了，很多时候问题不会仅仅停留在当前环节，设计者会考虑往流程的上游找问题，大多数时候问题是在前面的某一个环节中解决了，然后从那里重新开始继续往下走。

迭代性如图 1.6 所示，假设一个流程中有 4 个主要的步骤，正常情况下从步骤 1 执行到步骤 4，如果不出问题就结束了。但这么顺利的过程在电子产品（特别是 FPGA 的开发流

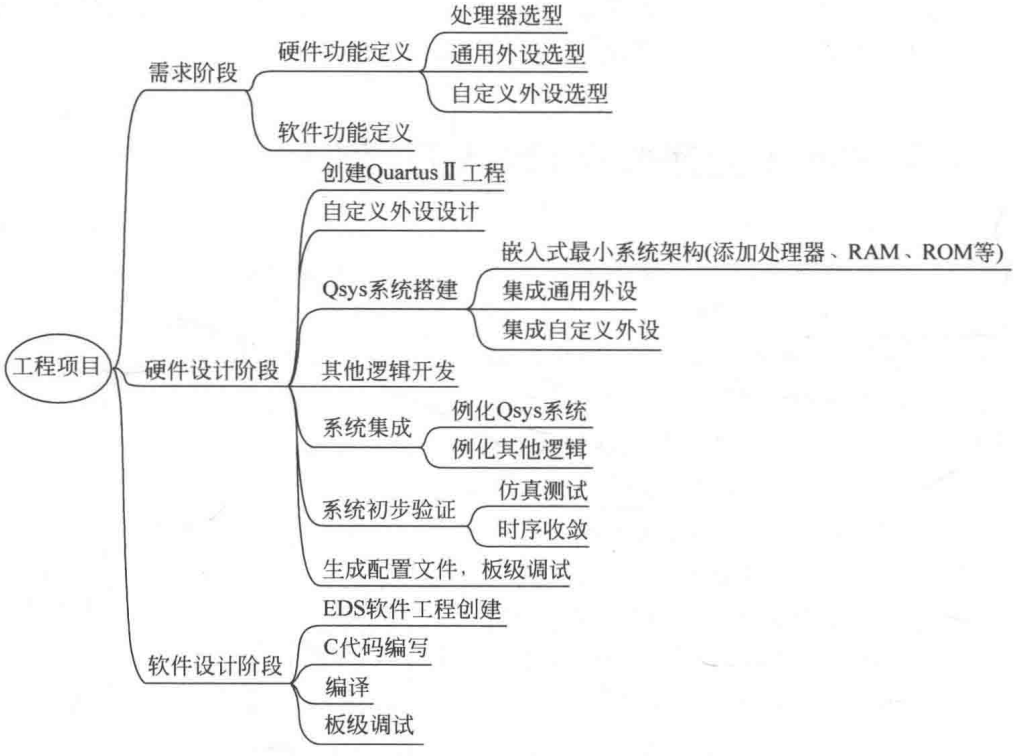


图 1.5 基于 Nios Ⅱ处理器的 FPGA 项目开发步骤

程)中是非常罕见的,有时甚至每个步骤都有出问题的可能。在日常生活中,人们做事的各个环节经常是互不相干的,哪个步骤要是出现问题就在哪个步骤解决,不会牵涉到别的步骤中,而 FPGA 的开发则大不相同,例如步骤 1 执行完进入步骤 2,如果在步骤 2 出了问题,也许在步骤 2 本身解决不了,那么就得回到步骤 1 找问题,如果步骤 1 解决了问题,那么此时整个流程就从步骤 1 开始重新执行。同样的,如图 1.6 所示,当进入步骤 4 的时候,如果有了问题,解决问题可能需要回到前面 3 个步骤的任意一个步骤重新开始执行。试想想,一个开发过程通常不会只有 4 个步骤,有时要经历几十个甚至上百个大大小小的步骤,那么出了问题就麻烦了。这是 FPGA 设计独有的特点,也是难点和重点。希望读者能够在实践中好好感受这个折磨人的特点,也多总结出一些经验和办法来应对。当然,笔者所能够想到和做到的就是在设计中认真认真再认真;然后多分模块,分解这些大迭代为小迭代,尽可能地降低设计之间的相互依赖性,降低问题定位难度,从而减少工作量。

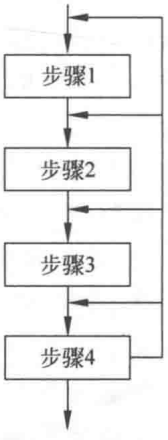


图 1.6 设计迭代性示意图

第2章

实验平台“勇敢的芯”板级电路详解

2.1 板级电路整体架构

如图 2.1 所示，“勇敢的芯”FPGA 实验平台是笔者与国内知名 FPGA 培训机构至芯科技携手打造的一款基于 Altera Cyclone Ⅳ FPGA 器件的入门级 FPGA 学习平台。

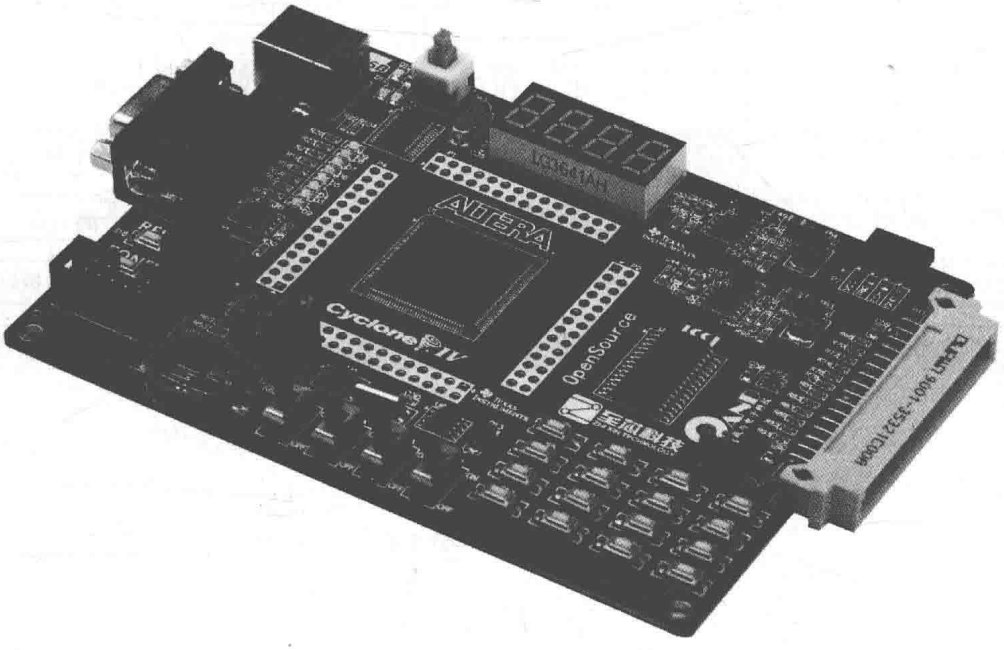


图 2.1 FPGA 实验板实物图

该实验平台板载丰富的常用外设，提供丰富的 FPGA 例程，包括逻辑例程和 Nios Ⅱ例程。如图 2.2 所示，这是整板外设器件的示意图。

如图 2.3 所示，围绕着 FPGA 器件，各个外设的连接一览无遗。