EDA设计
实战系列

爱上FPGA开发

——特权和你一起学NIOS II

吴厚航　编著

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

DVD-ROM INCLUDED　程序源码+20课时视频教程

爱上 FPGA 开发
——特权和你一起学 NIOS II

吴厚航　编著

北京航空航天大学出版社

内 容 简 介

结合一款基于 Altera 公司 Cyclone II 系列 FPGA 的开发板，从一些嵌入式开发的基本术语和概念入手，到手把手第一个工程的构建；再从一个稳定的 SOPC 平台设计，到 NIOS II 软件编程的入门；最后软硬件结合，像模像样地搭建了一些有实用价值的工程。本书内容可谓由浅入深，为在其他嵌入式平台上已有一定开发基础的 NIOS II 初学者量身打造。字里行间，不仅透露出一个年轻工程师对技术的执着和认真，而且很多诙谐幽默的文字和真实的感悟伴随着知识也传递给读者，相信一定能够带给读者更多耳目一新的感觉。本书配套 DVD 光盘，内含程序源码和 20 课时视频教程，方便读者学习。

本书的主要读者对象为电子、计算机、控制及信息等相关专业的在校学生，从事 FPGA 开发设计的电子工程师以及所有电子设计制作的爱好者们。

图书在版编目(CIP)数据

爱上 FPGA 开发 ：特权和你一起学 NIOS II /吴厚航编著. -- 北京 ：北京航空航天大学出版社，2011.10

ISBN 978-7-5124-0542-4

Ⅰ. ①爱… Ⅱ. ①吴… Ⅲ. ①可编程序逻辑器件 Ⅳ. ①TP332.1

中国版本图书馆 CIP 数据核字(2011)第 149936 号

爱上 FPGA 开发

——特权和你一起学 NIOS II

吴厚航 编著

责任编辑 杨 昕 刘爱萍

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：emsbook@gmail.com 邮购电话：(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本：787×960 1/16 印张：20.5 字数：459 千字

2011 年 10 月第 1 版 2011 年 10 月第 1 次印刷 印数：4 000 册

ISBN 978-7-5124-0542-4 定价：45.00 元(含光盘 1 张)

前　言

特权同学在《深入浅出玩转 FPGA》出版之后，并不像某一些博文丛书的作者就此销声匿迹了(前辈们若不小心看到此文别拍砖，那就说明你们和特权同学一样依然活跃在这个领域☺)。特权同学每天依然忙碌地工作着，没有离开热爱着的 FPGA 开发设计工作。因为年轻，所以依然在追寻着年少时的梦想。特权同学的博客(http://blog.ednchina.com/ilove314/、http://blog.chinaaet.com/ilove314/和 http://www.eefocus.com/ilove314/blog/)也时常更新，工作中新的感悟和体会都会在第一时间和工程师网友们分享。当然，也许因为这本《爱上 FPGA 开发——特权和你一起学NIOS II》的整理和编写，BLOG 会更新的慢一些。

之所以给这本书起名《爱上 FPGA 开发——特权和你一起学 NIOS II》，是因为特权同学个人感觉书名表达很形象，也很贴切。特权同学在 FPGA 逻辑设计方面的学习有一段时间了，《深入浅出玩转 FPGA》一书的出版是对特权同学付出的一个很好的回报，也听到了很多支持的声音(不仅在“生我养我”的 EDN 如此，甚至在电子工程专辑、ourdev、当当、电驴等各种网站)。一方面特权同学觉得很欣慰，能够将自己的所学所悟分享给大家，并且得到大家的认可，也确确实实能够对大家的学习起到一个不错的促进作用；但是另一方面，也深深觉得自己才疏学浅，工作年日不长，经验还不够丰富，项目做得不够多、不够深、不够广，所以在很多方面仍然有所欠缺，对有些问题的看法和见解还不够全面和深入。尤其在写完《深入浅出玩转 FPGA》一书之前，对 SOPC 这个很时髦的 FPGA 应用领域还没有太多主观的认识和亲身实践。庆幸的是，经过最近半年多以来新项目在 NIOS II 平台上的应用，特权同学对 SOPC 的整个架构和应用有了更深层次的理解和认识。因此，也算是“现学现卖”吧。

那天，tengjingshu 网友通过 EDN 的娜娜姐询问特权是否可以推荐一些写得好的 verilog 语法或是相关书籍，特权同学正思考着罗列一些书名，忽然记起能想到的好书应该都列在自己新书的参考文献中，还有就是建议多看 Altera 和 Xilinx 官方的 handbook 和 application note。于是娜娜妹就回了一句“我哥的回答越来越官方了”。作为读者的你是不是也会担忧这本书可能很“官方”？虽然特权同学可以很坦白地告诉各位看将：本书中出现的所有官方的说法基本都应该是来自 http://www.altera.com.cn/literature/lit-nio2.jsp 上的 9.1 版本 的 NIOS II 处理器的主要文档。但是，也请大家放心，特权没有“特权”，也是平民一个，绝对没有官腔官调，因此所有官方的东西只可能在必要的时候引用，但凡不需要太“官方”的地方绝对会以最通

俗易懂的方式并以自己的理解来透析也许对初学者仍有些生涩难懂的知识点。前面说过，特权同学要和您一起学，所以，我们的距离只是高级初学者和低级初学者。很多时候，请记住，特权同学会尽量不把大家落得太远，因为我们要一起学，所以我们才可以名副其实亲切地互称“同学”。

另外，光搞理论不是办法，我们都学了至少 9＋3＋n 年的理论了，很多时候会感觉理论就像是一个忽悠人的幌子。如果大家看看市面上和 NIOS II 相关的书就会发现，基本上（当然不是绝对的），很多高校某某大牌教授写的 XXX 教程都是由一大帮学生在那里翻译的 Altera 文档，而且特权同学还看到过有些被翻译的甚至风马牛不相及。哎，只觉得这类书不明确标明“XXX. pdf 翻译”字样也就罢了，如果漏洞百出让大家学得晕头转向就真的不应该了。所以，特权同学会保持以往的风格，继续让大家有机会在实践的基础上学习理论。其实早在 2010 年初的时候，特权同学就曾借 EDN 的助学小组推出了 SF－NIOS2 学习板的第一版，写了一些和以往风格迥异的“傻瓜”教程，承认当时做事有些浮躁了，只希望早点推出这个套件以弥补小组里 SOPC 方面学习板的空白，自己当时也是一个“傻瓜”初学者，那个“傻瓜”板子虽然和市面上的同类板子比起来足够便宜，但是缺乏实质性的一些可操作的 DIY 工程。俗话说，玩也要玩得有技术含量一些嘛，深深的内疚更是催促特权同学下决心要好好花一些时间和精力在 NIOS II上，编写教程的过程是艰辛的（就像当初我写书）；但是，在这个过程中同样也是不断地督促自己再学习，帮助他人进步的同时自己也在进步。如果最后能获得大家的拍手叫好那肯定是最开心最有成就感的事情了。还想提一点，为这本书量身打造的第二版 NIOS II 学习板还叫SF－NIOS2，但是整个风格乃至板子的外设方面都会有较大的变动。读者也不用担心自己买到的会是第一版的学习板，如果您看到了这里的文字，那么市面上就只有第二版的 SF－NIOS2了。

特权是做硬件出身的，画原理图、画 PCB，甚至焊接和编程都不在话下。这年头，全能战士可能不太值钱，因为老板会说“你什么都会，说明你什么都不精”。不过依个人之见，也算是给大家一点建议，如果你是个还处于上升阶段的电子专业学生或是电子行业的初级工程师（偶当年刚毕业的时候，在国企那套体制里，半年后才称得上“助理工程师”），我想说什么？哦，“不要什么都不会，最好什么都会点；还有就是不要什么都不精，至少要在某一方面有点过人之处。如果，那个‘点’字换成‘些’字就更好了。”呵呵，仔细想想这句话，也许对任何人在任何行业的职业发展都适用吧！因为每个正常人的智商都是差不多的，如果又发现了一个爱因斯坦式的天才，我想肯定不会是你，也不会是我。所以，对于每一个人而言，其时间和精力都是相对有限的，而如何利用这些有限的时间来最大化地展现自己的能力，就是成功和失败的关键了。话扯远了，忽然发现这个“前言”老是扯得有些文不对题，其实特权同学也只是希望在这里说一些技术本身之外的、大家也许比较关心的东西，再借着这篇文章或许能够拉近我们的距离，因为，我们都要成为“同学”了嘛。

不过谦虚点说，也许特权同学在软件编程方面不是强项，甚至最多也还是只能算“菜鸟”一个。所以，纯粹想来学习 NIOS II 软件编程方面知识的朋友可能会有些失望了（可以这么说，如果你本来就是一个 C 高手，那么要成为一个 NIOS II 软件高手也是很容易的。所以，希望不在特权身上，也不在这本书上，完全在于你自己）。但是，特权同学想说的是，如果你希望在 SOPC 的整个硬件架构上，甚至系统的架构上有多一些的了解和实践，那么我们可以轻松上路。这本书的重点是硬件架构和设计思想，软件也是实现一个完整系统必不可缺的部分，所以，特权同学还是会竭尽所能地让它们“和谐”地揉和在一块。

总之，套用《圣经》里的一句话，就是“一切智慧和知识的宝藏，都藏在祂里面”。所以，读《圣经》吧，相信祂能带给你许多意外的收获。

致谢…

细细算来，在 EDN 开博已 3 年有余了（2008 年 3 月 17 日写下第一篇博文），按照 yulzhu 兄的说法，特权同学也算是为数不多几个能够能迈过 2 年这道坎，并且已经走过 3 年的博主。几年下来，虽然博文的数量算不上高产，但这里不是夸口，质量还可称得上是有所保证的，毕竟写出一篇让人拍手叫好的加“酷”博文都是要费些脑细胞的。

看着一不小心点击量都已经迈过百万大关了，确实有些感慨万千。记得很久很久以前，写了一些懵懵懂懂的入门性质的技术文章，扔在了那时非常流行的 qq 空间里“炫耀”，那时我才刚大四。之后忽然有一天，在找寻参考代码的过程中 Google 把我带到了 EDNChina，忘记了是哪位博主的文章，加上一些自我鼓励的文字，就一下子把我吸引住了，于是就和这里有种“一见钟情，相见恨晚”的感觉，便决心扎根此地，而后，正可谓“一发而不可收拾”。

但是，3 年说长也长，道短也短，真可谓稍不留神，时间转眼即逝，往事便如飞而去。其实这里不是要感慨自己走过的所谓“风风雨雨”，不喜欢做个“矫揉造作，无病呻吟”的人，所以一直只是在博客里坚持尽可能原创的技术性文字，而非太多个人的见解、看法甚至怨恨和牢骚。只想说，EDN 这个平台在不知不觉中赋予了我太多太多，我其实非常非常爱这里的一切。我的几位挚友是在 EDN 结识的，我现在的这份工作从某种意义上也是 EDN 论坛里某个“广告贴”成就的，《深入浅出玩转 FPGA》完全就是 EDN 博客的“翻版”。当然，我也不会忘记你们曾经慷慨的 banner。可以说，网友们的信任和支持，很大一部分是因为你们。所以，在这里我势必要发自心底地再次对 EDNChina 网站的工作人员以及所有活跃在这个平台上的战友们道声“谢谢”！

在这里，我尤其要感谢 EDN 的社区专员黄娜美眉，这两年来有你的支持和帮助，才使得 FPGA/CPLD 助学小组始终处于长盛不衰、人气高涨的态势。虽然在本书出版之时，也许你已经离开了职业生涯开始的地方——EDNChina，但无论你走到哪里，我们这帮兄弟们依然会为你加油、为你喝彩。我们也坚信，你的下一站会比这里更精彩。

除了EDN，特权同学也应邀在ChinaAET、EET－China和EEFOCUS开博，拓宽了与广大工程师网友们交流互动的平台。在这里也对两位资深媒体人敬荣强、张迎辉和贺潇荃表示感谢。

在浩如烟海的网络中，意外结识的两位阳光向上的“奥特拉师弟”张亚峰和韩彬，谢谢你们给了特权同学很多创作的灵感和动力，也很感激你们提出的许多宝贵建议与想法。

还有身边朝夕相处一年多的同事刘铭，谢谢你为本书提供的几个简单实用的小软件。当然了，本书的顺利出版还离不开公司两位领导杨志成和王成鑫的支持和关注，特权同学打心底佩服你们的开明和远见。我始终相信，作为一个公司，也只有不停地创新、不断地超越、适时地总结和分享，才能够换来长远的可持续发展。

同样，特权同学能有时间和精力完成第二本书，家人的功劳是最大的，尤其是我的妻子。这本书同样是献给我所有的亲朋好友，祝福你们都有一个光辉灿烂的前程。

最后，特权同学也很想对所有参与本书编辑和整理校对的工作人员道一声“你们辛苦了，谢谢你们”！

篇幅所限，还有很多需要感激的人和事，这里却无法一一罗列。只想送上同样作为特权同学邮件签名的一句引自《圣经》的祝福语：

“愿主耶稣基督的恩，神的爱，圣灵的交通，与你们众人同在。”

特权同学

2011年5月于上海

目　录

第1章

海阔天空聊概念

虽然这本教程的大标题美其名曰“特权和你一起学 NIOS II”，但是更确切地说应该称之为“特权和你一起学 SOPC”。之所以用 NIOS II 替代 SOPC，其实只是因为 SOPC 的概念着实有些大了，而 NIOS II 是全球两大 FPGA 制造商之一的 Altera 公司为他们的 SOPC 系统架构量身打造的软核处理器。咱比较厚道，就小不就大，只能给个小名“NIOS II”了。

1.1 CPU 之软核与硬核

既然说到了处理器，而且还给它戴了个“软核”的帽子，那么就不能让这个“帽子”戴得不明不白。说概念就要说得清清楚楚、明明白白，特权同学在本章的目标是扫盲，所以大家可别嫌啰嗦，如果觉得都是废话，而你的等级够高，那么就请无视这些啰嗦的叙述，直接跳到下一章，是个不错的选择～-～。好了，下面说正题。处理器，不懂电子的也知道计算机的 CPU 中文名也叫处理器，而做嵌入式这一行的就不能够让处理器一词仅仅局限于“CPU”了。从最简单的 51 单片机来看，它也算是一个包含了“CPU”的“MCU”(Micro Controller Unit)，其实非要寻根问底搞清楚 MCU 为何物的话，还是拿大家熟悉的 PC 机来打比方。一台 PC 机，有 CPU(就像是人的大脑)、硬盘(存储海量的数据)、内存(暂时存放数据的地方，高速运转的数据处理和运算都要靠它，操作系统“跑”起来也还是有赖于它)、电源、显卡、键盘鼠标等，当然还有一些必要的外设，如光驱、音箱、USB 口、串口、并口等等，总之是应有尽有。而这个小小的 MCU 哪怕是 DIP - 40 封装的“大家伙”也只有食指般大小。可别看它小，它可是“麻雀虽小、五脏俱全”，其实在芯片家族中它不算小，但比起那个动辄上千个“脚脚”的计算机 CPU 和那个庞大的机箱来，可真算得上是“迷你精致”了。拿它和 PC 机比还真不能比性能，咱们要比概念。但凡用过 MCU 的都知道它的“强大”，它不仅能够运行程序并对各种数据进行运算和控制(处理器)，而且能够存储程序和数据(ROM 和 RAM，堪比硬盘和内存)，当然也有一些简单的外设，常见的如 GPIO、串口、定时器、看门狗，强大一点的集成 USB 口、网口，甚至一些奇形怪状的，连特权也都叫不上来的接口。那么，现在应该明白 MCU 是什么了吧？啊，说完这些还没有谈论 CPU 是什么。CPU 就是处理器，处理器就是 Central Process Unit，也就是中央处理单元，而中央处理单元就是能够“跑”程序的一颗芯片而已，哈哈，没绕晕吧？CPU 里面可以跑程序，

程序可以做很多事，比如可以做一次运算，实现各种各样的复杂算法；也可以写一段驱动，控制各种各样的外围设备。总之，人脑可以想到的事，绝对都是CPU可以办到的事。这么说，就知道CPU的powerful了吧。

说完CPU，再说为什么是“软核CPU”，是不是也有“硬核CPU”呢？答案是肯定的，通常大家看到的CPU都是指一些芯片制造商完全定制好强大运算功能且硬件结构固定的“芯”。用户没有办法对它做的硬件构造进行任何的更改，只能够看着制造商提供的芯片datasheet进行编程控制。而“软核”、“硬核”的说法主要还是针对可编程逻辑器件如FPGA上嵌入的CPU而言的。那么有些人可能就纳闷了，既然都嵌在FPGA上，又为何有软硬之分呢？FPGA不都是硬件吗？没有错，FPGA是硬件，它所实现的逻辑功能也确确实实是各种各样门电路的集合。但这里的软硬叫法和通常所言的软件、硬件叫法是有区别的。别急，马上就要说区别了。如果一个FPGA像Xilinx公司的什么Virtex XX系列号称内嵌了最多数个PowerPC硬核CPU(想知道它是什么吗？自己动手丰衣足食，搜去)，请注意它这里内嵌在FPGA器件里的CPU是“硬核CPU”。而Altera公司主推它的“NIOS II软核CPU”，无论是低成本的Cyclone系列还是面向高端的Stratix系列，号称都可以内嵌数个NIOS II软核。说白了，软就是只要资源足够那么就可以用逻辑堆砌起来一个“CPU”。再具体点说，硬核就是FPGA本身就嵌入了一个这样的CPU硬件结构，只是用与不用、如何使用的问题交给设计者决定而已；而软核则是FPGA本来并不存在这样的硬件结构，设计者使用了某种硬件描述语言(或者说是最基本的逻辑门电路)活生生地搭建了一个CPU来。这么说好像还不过瘾，那就打一个比方加深理解。记不记得小时候刚上学时练字的“美好时光”，其中有一节专门的“写字课”，开始练字的时候是描红(上面有字的轮廓，照着描就行)，练到一定水平就要抛开字帖，拿着白纸写就可以了。而硬核就好比是字帖，每个要写的字都是固定好的，必须照着去写；而软核就好比是白纸，上面什么都没有，要写什么字都可以。同样写一个“爱”字，在写之前，硬核是有轮廓在那里的，这个字写得不能太大也不能太小，位置也相对固定，好的临摹就要求要不偏不倚地把原有的字重现出来。而软核上什么都没有，软核靠自由发挥，可以写得方方正正，也可以歪歪扭扭；可以写得大一些，也可以写得小一点，甚至原本一个字的位置也可用同样的多个“爱”字来填充。“临摹”出来的硬核写得肯定漂亮，并且稳定性好、性能佳，但是“临摹”的功能是固定的，这块电路是“死”的，不能写别的“字”了；自由发挥的软核虽然没那么端端正正，稳定性、可靠性都差一些，但是可写的“字”不受局限，灵活性会高很多。

1.2 SOPC是什么

好了，继续下一个议题。上面说到了MCU是“麻雀虽小、五脏俱全”，下面就直接聊SOPC的“十脏俱全”。为了加快教程进度，请允许特权同学引用一段特权老师在《深入浅出玩转FPGA》笔记2中的一段话来阐述SOPC的概念。

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(SOC)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 分别作为程序存储和数据缓存，这些器件都放置在一块或者数块 PCB 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。

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、DSP 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外部的 Flash 和 SDRAM 都集成了。

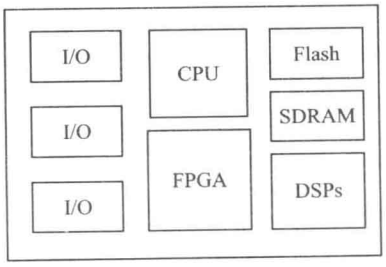

图 1.1　传统的复杂系统

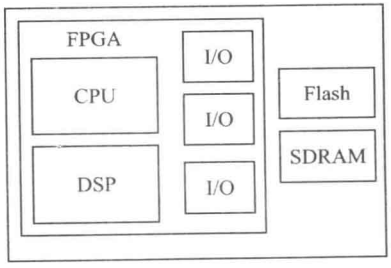

图 1.2　基于 FPGA 的片上系统

明白了吧，SOPC 就是一颗比“MCU”更 powerful 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，必须有这个外设又必须满足那个条件的处理器了；甚至有时也不需要考虑处理器都能够挂什么样的存储器来读写数据运行程序。只要有 SOPC，一切轻松搞定，外设？想加什么就加什么，套用一句广告词“我的地盘听我的”，对，这里的一切由你做主。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 好像说得还不够尽兴，那么下面就套用一些官方用语来彰显一下本教程的权威性。

SOPC 是一个强大的系统开发工具。SOPC Builder 可以帮助用户定义并产生一个完整的 system-on-programmable-chip(SOPC，这个还是用英文原文吧)，它比传统的手动集成方式要方便得多。SOPC Builder 工具集成在 Quartus II 软件中。

用户可以使用 SOPC Builder 生成一个基于 NIOS II 处理器的系统。然而，SOPC Builder 远不止一个 NIOS II Builder，它也可以生成一个不包含 NIOS II 的其他处理器系统，甚至是不

包含任何处理器的纯外设互联系统。

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 SOPC Builder，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联逻辑。SOPC Builder 产生了系统所有组件的 HDL 文件，顶层的 HDL 文件则例化好系统的所有组件。SOPC Builder 既能够生成 Verilog 也能够生成 VHDL 代码。

说了这么多，可能很多不注意听讲的同学还是不太能够区分 SOPC 和 NIOS II，甚至有点搞混了。不要紧，再出示一张图让大家饱饱眼福。如图 1.3 所示，是一个简单的嵌入式系统。在一块 PCB（Printed Circuit Board）板上，单论芯片可能只是一片 FPGA、一片作为协处理器（Co - Processor）的 CPU、两片 DDR2 的 SDRAM 存储器分别挂在 FPGA 和协处理器上，另外还有一个叫做总线桥（Bus Bridge）的接口（它也许只是简单的连线而已，也许是一块协议芯片）。

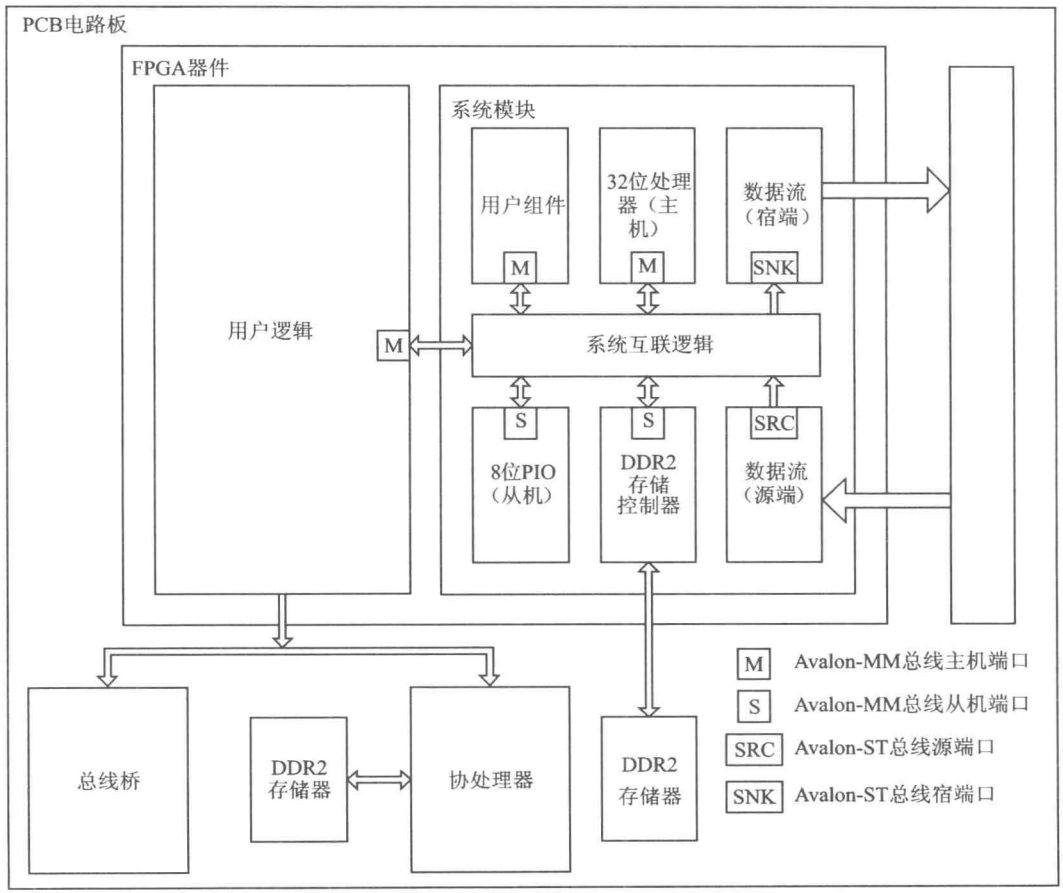

图 1.3　FPGA 上的 SOPC 系统实例

这个系统看似简单，其实不然，FPGA有大文章可做。想继续往下，但是会发现有一点痛苦，因为这里要普及概念，所以不得不把该说的通通说一遍，哎，可能要费些笔墨，还望各位看官耐心听讲。

从总线桥开始，先说桥，通俗的理解桥就是用来连接河两岸的，一定听说过主板上的南桥和北桥吧(CPU和内存什么的居然还隔着条河)。CPU很好很强大，可以处理海量数据，但是再强大也不能发出声音、显示图像啊，术业有专攻嘛，CPU就是干数据运算和处理的活，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，把处理完的数据发送出去，可能说得不是很专业，但是基本就是这样了。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，那么恐怕咱们的CPU要比手掌大才能够在肚子底下容下那么多“脚”了。这么看，这里的“桥”还真不能是独木桥。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，所说的CPU就处于这样的核心位置。好，这里不再深入了，总线其实就是CPU的一组满足一定协议的引脚的集合(对不起，不是很“官方”的总线解释，怎么感觉有点在做数学概念的定义，嘿嘿，特权同学的数学没学好过)，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而出现的。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，它的名字叫做“Avalon”，这个名字是不是很炫啊，对，Avalon总线，以后当你越是使用它也就会越多地发现它的强大。上面提到NIOS II只是一个处理器而已，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中与DDR2 Memory Controller连接的SDRAM)连接起来。当然也可以理解“系统互联逻辑”(System Interconnect Fabric)就是Avalon，但是请记住，Avalon只是System Interconnect Fabric的一种形式而已，想了解其他的形式就自己翻datasheet去(\(^o^)/～，偷个懒，就不说)。

说到“桥”就说到FPGA里面去了，说完才发现其实FPGA系统内和常见的嵌入式系统的架构有着异曲同工之妙，个中奥妙只有用心的人才能体会到，不知道读者感悟到了没有。观察那个协处理器(Co-Proccessor)外面是不是和系统模块(System Module)内的32位处理器(Proccessor，就认为它就是NIOS II吧)一样要挂DDR2的存储器(Memory)呢？所以说，事物都是有共性的，重在归纳和总结！跟特权一起不光要学知识本身，更要学怎么学知识，还有比知识更重要的哦。

继续说，图1.3的系统里最核心的应该是那个叫做FPGA的家伙。别看它年轻，《圣经》提摩太前书四章12节说：“不可叫人小看你年轻，总要言语、为人、爱、信、纯洁上，都做信徒的榜样。”做人如此，做技术也一样，年轻同样可以有大作为。用FPGA搭一个SOPC的最大优势，用三个字最贴切——“灵活性”。没有错，传统的51系统常常是一个MCU旁边挂了一大