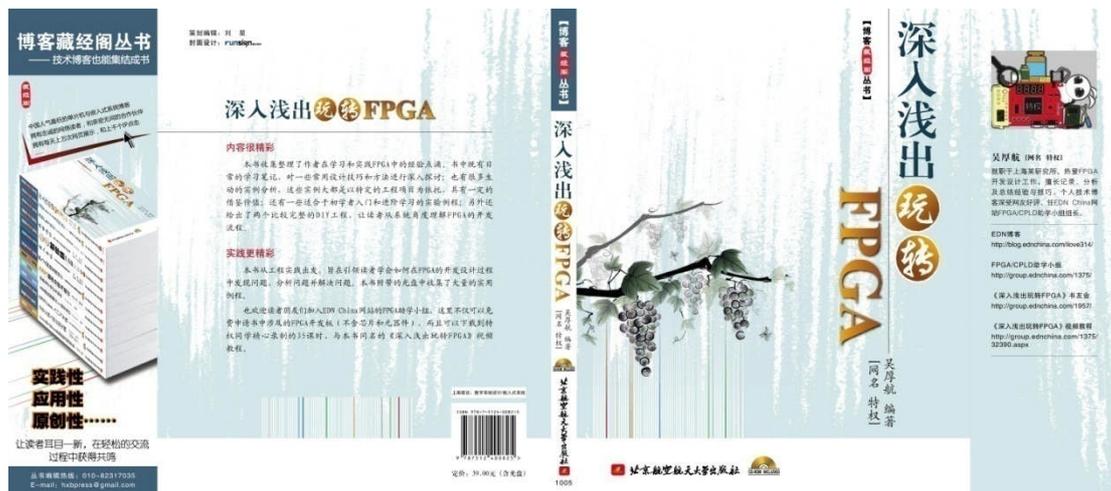


# 特权和你一起学 NIOS2

欢迎加入 EDN FPGA/CPLD 助学小组共同交流学习：<http://group.ednchina.com/1375/>

本教程仅适用于特权制造的 SF-NIOS2 FPGA 开发板，转载请注明出处！

版本信息		
时间	说明	备注
2010-9-21	创建	V1.0



## 第一章 海阔天空聊概念

虽然这个教程的大标题美名其曰“特权和你一起学 NIOS2”，但是更确切的说应该称之为“特权和你一起学 SOPC”。之所以用 NIOS2 替代 SOPC，其实只是因为 SOPC 的概念着实有些大了，而 NIOS2 是全球两大 FPGA 厂商之一的 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“NIOS2”了。

### 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。说概念咱就要说得清清楚楚，特权同学在本章的目标是扫盲，所以大家别嫌我啰嗦，如果你觉得都是废话，证明你的等级够高，那么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，说正紧的。处理器，不懂电子的也知道电脑的 CPU 中文名也叫处理器，而咱们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，我还是喜欢拿大家熟悉的 PC 机来做比方。一台 PC 机，有 CPU（大家都知道它是大脑）、有硬盘（存储海量的数据）、有内存（暂时存放数据的地方，高速运转的处理和运算都要靠它，操作系统跑起来还是有赖于它的）、有电源、有显卡、有键盘鼠标等等，当然还有一些乱七八糟的外设了，如光驱啊、音箱啊、USB 口啊、串口啊、并口啊，总之是五花八门应有尽有。而这个小小的 MCU 哪怕是 DIP40 封装的“大家伙”也只是食指大小，可别看它小，它可是“麻雀虽小、五脏俱全”，它其实在芯片家族中不算小，但比起那个动辄上千个“脚脚”的电脑 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 比还真不能比性能，咱们要比概念。但凡用过 MCU 的同志们都知道它的“强大”，它不仅能够跑程序做运算和控制（处理器），而且能够存程序和数据（ROM 和 RAM，堪比硬盘和内存），当然也有一些简单的外设了，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特权叫都叫不上来的接口。那么，你现在应该明白 MCU 是什么了吧？啊，说完这些发现还没有谈论 CPU 是什么。CPU 就是处理器，处理器就是 Central Process Unit，Central Process Unit 就是中央处理单元，中央处理器就是能够跑程序的一颗芯片而已，哈哈，没把你绕晕吧。CPU 上面可以跑程序，程序可以做很多事，你可以做一个运算，实现各种各样的复杂算法；你也可以写一段驱动，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，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实确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 家的什么 Virtex XX 系列号称内嵌了最多数个 PowerPC（想知道它是什么吗？自己动手丰衣足食，搜去），那么这里的“内嵌”就意味着这个 FPGA 器件的 CPU 是“硬核 CPU”。而 Altera 家主推它的“NIOS2 软核 CPU”，软就是只要资源足够那么就可以用逻辑堆砌起来的一个“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）活生生的搭建了一个 CPU 来。这么说好像还不过瘾，那就打个比方加深一下理解。记不记得小时候咱们刚上学时练字的“美好时光”，我是记得还有什么专门的一节“写字课”，开始练字的时候还要用字帖（字帖上模模糊糊的有字的轮廓，照着写就行），练到一定水平就要抛开字帖，拿着白纸写就可以了。而硬核就好比这个字帖，每个要写的字都是固定好的，你必须照着去写；而软核就好比这个白纸，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，软核上什么都没有。硬核临摹一下就好，并且只能写这个字；软核靠自由发挥，你也可以写别的字。“临摹”出来的硬核写的肯定漂亮，并且稳定性好、性能佳，但是可“临摹”的功能是固定的，这块电路是“死”的，不能写别的“字”了；而自由发挥的软核虽然可能没那么端端正正、稳定性可靠性都差一些，但是可写的“字”不受局限，灵活性也高得多。

## SOPC 是什么

好了，时间不早，要继续下一个议题。上面说到了 MCU 是“麻雀虽小、五脏俱全”，下面就不跑题直接聊 SOPC 的“十脏俱全”。唉，5:30 下班，6:00 出门，7:00 到家，8 点开始写，都快 10 点了，这第一章才刚扯到这里。为了加快一下教程进度，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 1.1，一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。

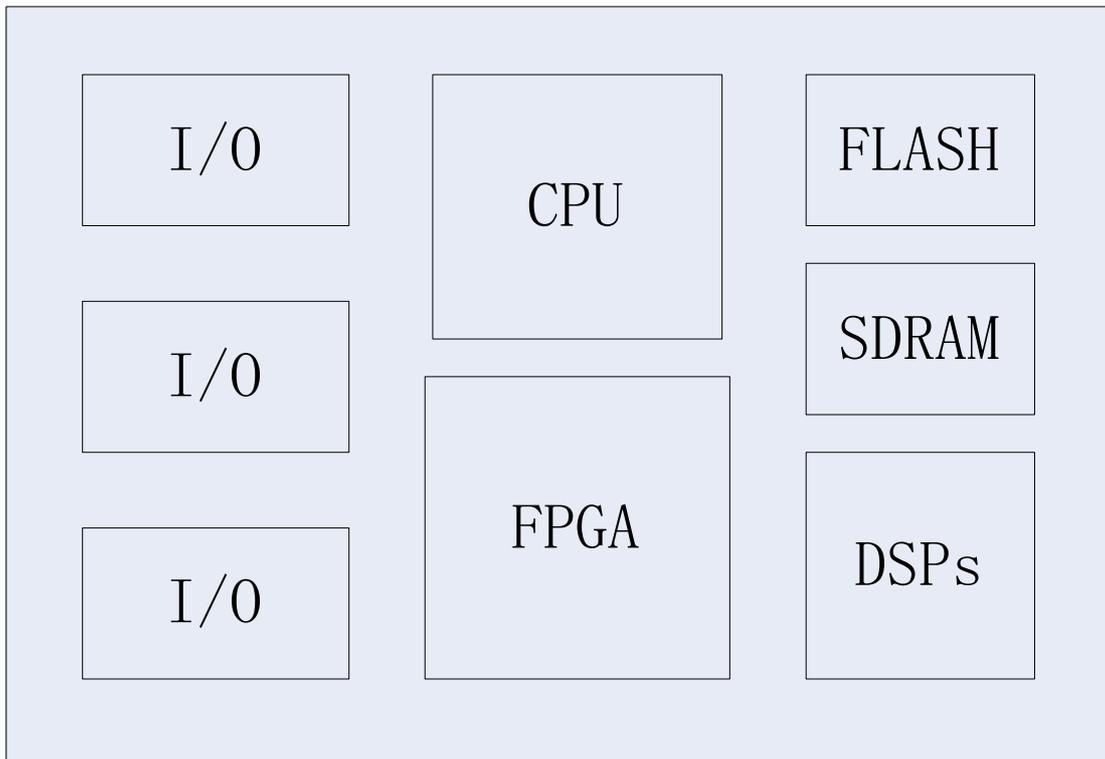


图 1.1 传统的复杂系统

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、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。

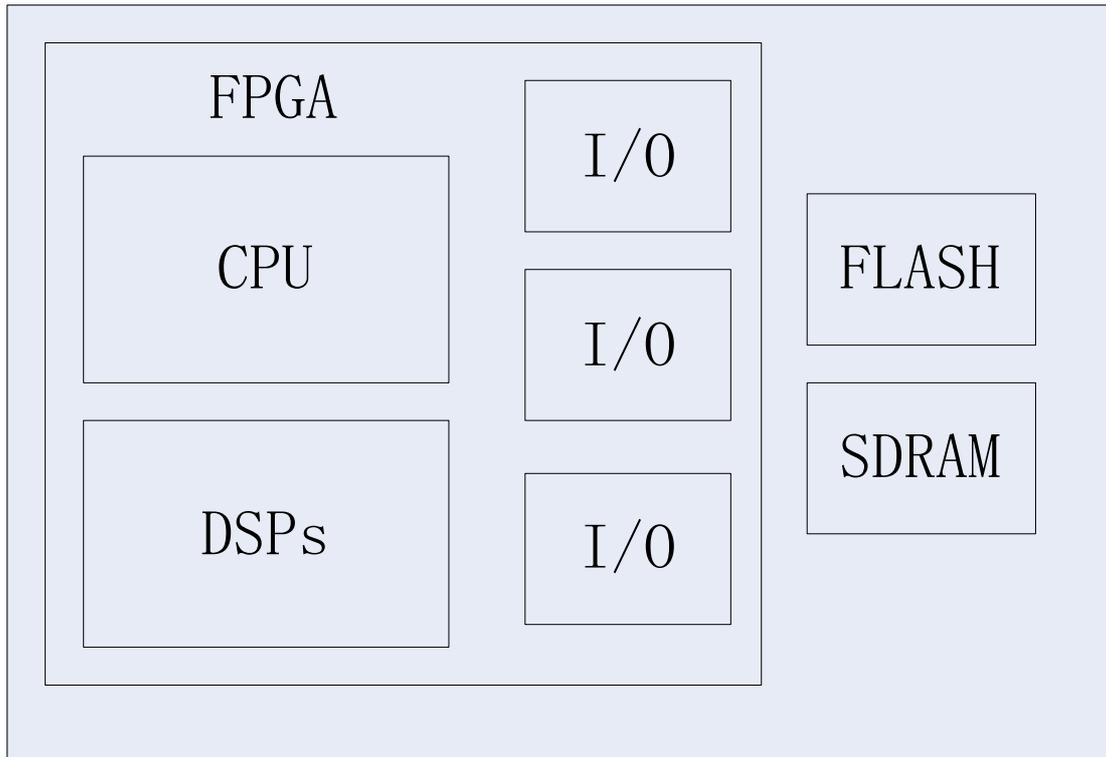


图 1.2 基于 FPGA 的片上系统

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 这个东东，设计者就不需要再拿着选型手册海选既必须有这个外设又必须有那个外设的处理器了；甚至有时也不需要考虑处理器都能够挂上什么样的存储器来存数据跑程序。只要有 SOPC，一切轻松搞定，外设？想加什么就加什么，那句广告词怎么说来着，“我的地盘听我的”。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，那么就套用一点官话来彰显一下本教程的权威性，哈哈！

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 NIOS2 处理器的系统。然而，SOPC Builder 远不止一个 NIOS2 builder 而已，它也可以生成一个不包含处理器或者包含 NIOS2 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 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 NIOS2，甚至有点搞混了。不要紧，再上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、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)。

这个系统中，看似简单，好像没啥花头。其实不然，FPGA 里有大文章可做。想继续往下说，但是发现会有一些痛苦，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。

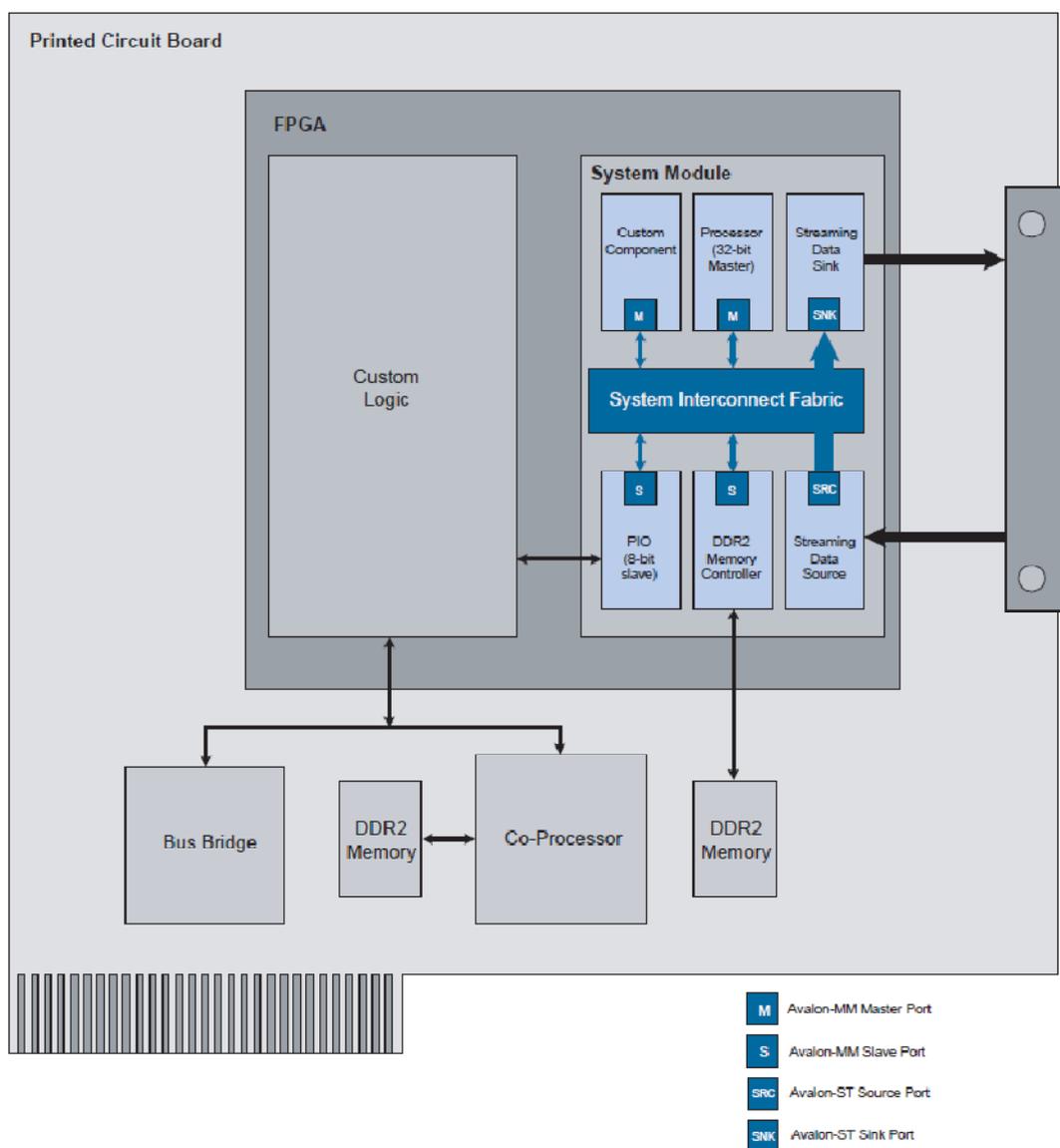


图 1.3 FPGA 上的 SOPC 系统实例

从总线桥开始吧，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，但是再强大也没法发出声音显示图像啊，术业有专攻嘛，CPU 就是干数据运算和传输的活，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 的管脚终归是有限的，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 要比得上巴掌大才够在肚子底下容下那么多“脚”了，这么看，这个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，我们所说的 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管脚的集合（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，特权同学的数学没白好过），这组管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”，其它外设就只能望桥兴叹。总线从某种意义上看就是为了节约管脚而出现的，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“Avalone”，这个名字是不是很炫啊，对，Avalone 总线，以后当你越是使用它也就会越多的发现它的强大。上面提到 NIOS2 只是一个处理器而已，而 Avalone 总线就是要把 NIOS2 和所有其它在 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。当然你可以理解那个 System Interconnect Fabric 就是 Avalone，但是请记住，Avalone 只是 System Interconnect Fabric 的一种形式而已，想了解其它的形式就自己翻 datasheet 去（\(^o^)/~，偷个懒，就不说）。

说个桥就说到 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，个中奥妙只能用心的人才会体味得到了，不知道读者你感悟到了没有。你看那个 Co-Processor 外面是不是和 System Module 内的 Processor（我们就认为它就是 NIOS2 吧）一样要挂个叫 DDR 的 Memory 呢。所以要我说嘛，事物都是有共性的，重在归纳和总结！跟特权一起学不关要学知识本身，更要学怎么学知识，学比知识更重要哦。

继续说，图 1.3 的系统里最核心的东西恐怕就是那个叫做 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，都作信徒的榜样。”做人如此，做技术可也一样，年轻同样可以有作为。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 旁边挂了一大堆什么 74HCxxx 的芯片，而当 16 位、32 位 CPU 在嵌入式系统行业大行其道的时候，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，甚至会感觉管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，有足够的管脚；它，支持各式各样的电压标准；它，可以并行处理各类复杂任务；它，可以像一张白纸任你涂画……所以，这就是你学习它的理由。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 可以集成大多数常用的外设，无论从 BOM 成本还是电路板面积上都有很大的优势。

### NIOS2 的优势在哪里

不妨先看看 Altera 为自己的 NIOS2 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帮助您：

- 选择最适合您应用的 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您的需求。

坦白的说，特权同学也不知道世界上用得最多的嵌入式处理器到底是哪颗。但是，倒是蛮赞同 NIOS2 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，NIOS2 更是加快了它的通用性步伐。

前面也提到过了，在面对一个新项目时，设计者在评估处理器的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存储需求？是否满足对各种不同外设的需求？可扩展接口数量，支持电平标准？……而实际情况是，往往不是这款处理器速度太慢，就是那款处理器外设太少，最终的解决办法就是使用多个 Core 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。因此，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，这个系统的灵活性通用性也会异乎寻常的让人为之振奋。在 Altera 的这个系统中，NIOS2 是当之无愧的主角。不可否认，NIOS2 的优势在于它所依托的 FPGA 上架构起来的 SOPC 系统。