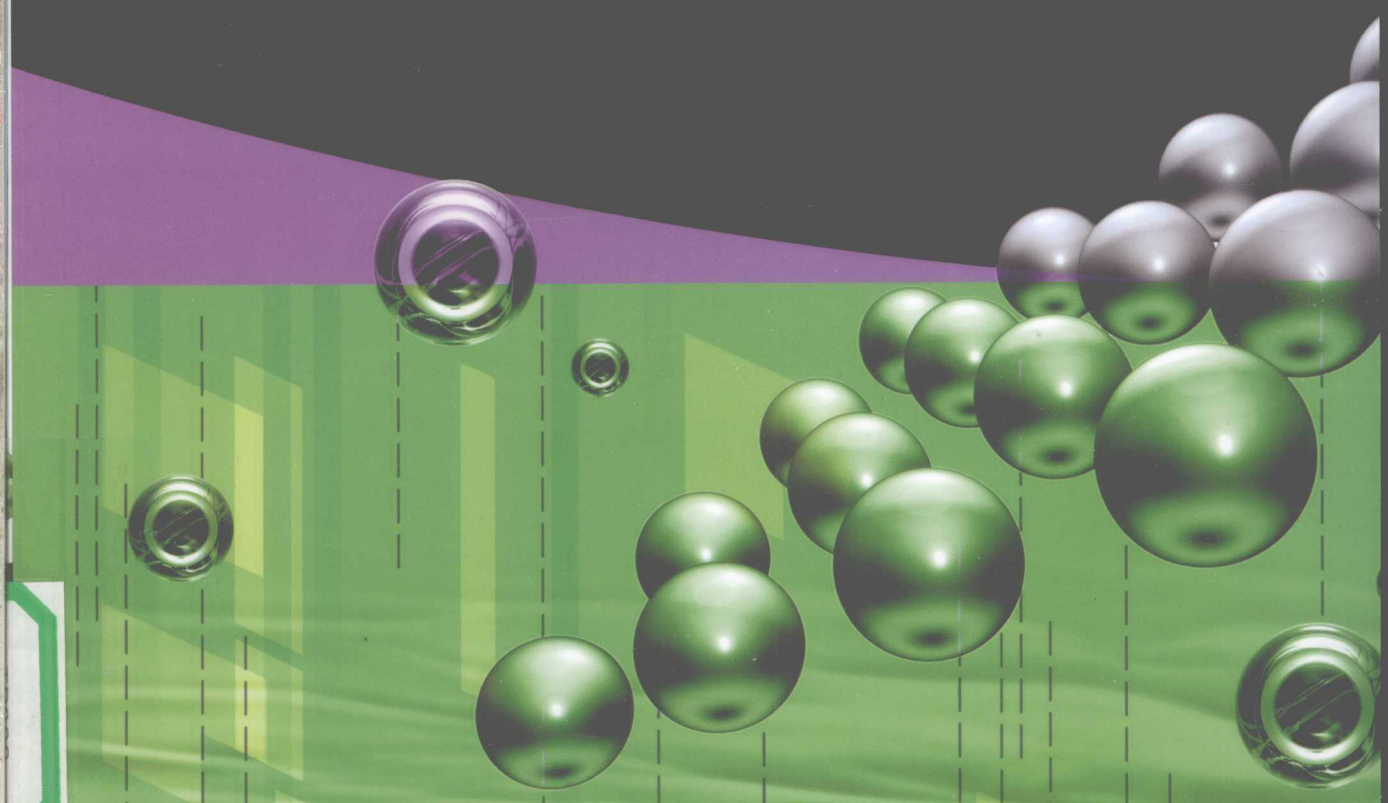
EDA应用技术

http://www.phei.com.cn

基于Nios II 内核的FPGA电路系统设计

赫建国　倪德克　郑　燕　编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

EDA 应用技术

基于 Nios II 内核的 FPGA 电路系统设计

赫建国　倪德克　郑　燕　编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书介绍了一种能够在 FPGA 芯片中同时获得数据传送速度快和数据处理能力强两个优点的设计方案——在片可编程系统（System On Programmable Chip，SOPC）解决方案。该方案通过在 FPGA 芯片中置入一个软核处理器系统（Nios Ⅱ软核处理器系统）来增强对信号的处理能力。

本书系统地描述了 Nios Ⅱ软核处理器系统的开发知识。内容包括 Altera 公司 FPGA 芯片的介绍、可编程逻辑器件开发软件 Quartus Ⅱ的使用、硬件描述语言 VHDL 的简介、Nios Ⅱ软核处理器系统创建工具 SOPC Builder 和 Nios Ⅱ集成开发环境（Nios Ⅱ IDE）的使用。书中还包括了大量的基础实验和应用系统的设计实例，能够帮助读者更快、更容易地掌握及应用这门技术。

本书适合从事 Altera 公司 FPGA 芯片开发设计的研究生和本科高年级学生使用，也适合从事该方面工作的工程师使用。

图书在版编目（CIP）数据

基于 Nios Ⅱ内核的 FPGA 电路系统设计/赫建国，倪德克，郑燕编著. —北京：电子工业出版社，2010.4
（EDA 应用技术）
ISBN 978-7-121-10647-7

Ⅰ. ①基… Ⅱ. ①赫… ②倪… ③郑… Ⅲ. ①微处理器－应用－数字电路－电路设计：计算机辅助设计 Ⅳ. ①TN790.2

中国版本图书馆 CIP 数据核字（2010）第 058034 号

策划编辑：柴 燕
责任编辑：徐 萍 文字编辑：徐 磊
印 刷：涿州市京南印刷厂
装 订：涿州市桃园装订有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：18 字数：460.8 千字
印 次：2010 年 4 月第 1 次印刷
印 数：4 000 册 定价：38.00 元

凡所购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

2004 年春天，作为指导教师，作者首次参加了由 Altera 公司组织的 EDA/SOPC 电子设计竞赛。Altera 公司提出的数字系统 SOPC（System On Programmable Chip）解决方案深深地吸引了我。

在此之前，作者虽然从事数字系统的教学和开发设计工作，并对用来实现数字系统的微控制器和可编程逻辑器件的使用也已多年，但是一直把它们看做是两种各具特点的目标器件。如果需要对信号进行较复杂的处理，首选的目标器件为微控制器；如果需要进行高速信号传送，首选的目标器件为可编程逻辑器件；如果希望同时获得复杂的信号处理和高速的信号传送，那么将联合使用微控制器和可编程逻辑器件。Altera 公司提出的数字系统 SOPC 解决方案能够在可编程逻辑器件中形成一个用户可配置的软核处理器——Nios Ⅱ 软核处理器，这样就在一个芯片上同时实现了信号的复杂处理和信号的高速传送。

利用数字系统 SOPC 解决方案实现应用系统设计需要掌握 Quartus Ⅱ 可编程逻辑器件开发软件和 Nios Ⅱ 集成开发环境。利用开发软件 Quartus Ⅱ，使用者能够创建以 FPGA 芯片为目标器件的 Nios Ⅱ 软核处理器系统及其他外围模块；利用 Nios Ⅱ 集成开发环境，使用者能够完成 Nios Ⅱ 软核处理器系统的 C 语言设计文件的编辑、编译，并下载设计到 Nios Ⅱ 软核处理器系统中进行调试和运行。

掌握一种技术需要大量的理论和实践学习，同时也需要一个好的学习方法，如果有一个好的学习指导是可以获得事半功倍的效果的。通过这些年来对参加 Altera 公司的 EDA/SOPC 电子设计竞赛和全国大学生电子设计竞赛的学生的赛前训练的总结，作者编写了本书。本书在撰写内容上力求实用，并尽量做到理论和实践相结合；在内容编排上，尽量做到由易到难。对书中列举的基础实验和应用设计，作者努力做到不仅描述如何进行，而且分析这样做的原因，使读者能够比较容易地举一反三。

本书的内容包括 3 个部分。第一部分包括第 1 ~ 4 章，这部分描述的是可编程逻辑器件开发的基础知识。如果读者具有使用可编程逻辑器件的经历，这部分可以跳过去，直接从第二部分开始。

第 1 章介绍数字集成电路的基础知识及 Nios Ⅱ 软核处理器的特点。

第 2 章描述 FPGA 器件的基础知识。包括 FPGA 器件和标准逻辑器件在功能上的对应关系，FPGA 器件的内部结构，以及 Altera 公司生产的 FPGA 器件的性能介绍。

第 3 章介绍可编程逻辑器件开发软件 Quartus Ⅱ 的使用。内容包括设计项目的创建，设计的输入、编译、模拟，以及向目标器件的下载。

第 4 章介绍硬件描述语言 VHDL。这里只介绍了设计工作中最常用的语法元素。

第二部分包括第 5 ~ 13 章，这部分讲述了 SOPC 解决方案的核心内容——Nios Ⅱ 软核处

理器系统的基础知识。

第5章结合流水灯控制电路的设计，完整地描述了Nios Ⅱ软核处理器系统的开发过程。先了解了工作过程，这样在后面学习新内容时就可以利用开发软件来验证，使内容变得直观易懂。

第6章讲述Nios Ⅱ软核处理器的性能。Nios Ⅱ软核处理器的性能指标是可以配置的，但是不同的性能需要占用的逻辑资源具有明显的差别。这里还介绍了一种程序运行时间的测量方法。

第7章结合数码管显示电路的设计，描述了Nios Ⅱ软核处理器最常用的外围模块——并行输入/输出模块（PIO）的输出使用方法。

第8章结合键盘输入电路的设计，描述了并行输入/输出模块（PIO）的输入使用方法和Nios Ⅱ软核处理器外部中断的使用方法。

第9章描述了使用EPCS串行配置芯片对FPGA进行内部逻辑配置的过程，并介绍了Nios Ⅱ程序的两种启动方式，即从EPCS引导程序和从CFI_ FLASH引导程序。

第10章描述了定时器（Interval Timer）内核的使用，并利用该内核实现产生时钟信号、周期测量和看门狗电路的设计。

第11章通过SDRAM控制器内核将片外SDRAM作为程序存储器和数据存储器，并驱动LCD12864液晶模块实现文字显示、打点、画线和画图等操作。

第12章介绍JTAG UART内核的功能、配置，并通过实例介绍JTAG UART内核获取、发送数据的方法。

第13章描述了串行外围接口（SPI）内核的使用，并通过该内核发送SD卡的CMD命令，实现了SD卡的复位、初始化和扇区读写等操作。

第三部分包括第14～15章，这部分向读者呈现了作者这几年来采用SOPC解决方案，使用Nios Ⅱ软核处理器系统完成的一些应用设计。这部分内容可使读者获得如何把Nios Ⅱ软核处理器用于应用系统的方法，同时也可以使读者了解应用系统设计的过程和撰写设计报告的基本格式。

第14章给出了一个具有量程自动切换能力的频率测量电路的设计过程。系统的频率测量范围为$(1.000 \sim 9.999) \times 10^6$Hz，并具有频率和周期两种显示方式。

第15章给出了信号频谱分析电路的设计过程。利用快速傅里叶变换（FFT），分析电路能够以满意的分析速度和精度完成周期信号的频谱分析。

本书是作者在多年教学和科研的基础上编写的。书中的内容不仅包括了作者的工作经验，还包括了许多学生的学习经验。在学习过程中，我们教授给学生知识，同时也从学生那里学到了很多东西。书中的许多硬件电路和软件程序都是由学生参与设计和调试完成的，在这里向张伟、张侠、李兆朋、冯鑫，以及许多没能列出姓名的同学表示衷心的感谢。

在本书的编写过程中，作者得到了西安邮电学院许多老师的支持和帮助，在这里向他们表示衷心的感谢。另外，作者也参考了许多专家和学者的著作和研究成果，在这里也向他们表示衷心的感谢。

本书能够顺利出版与电子工业出版社和柴燕编辑的大力支持是分不开的。柴燕编辑在本书的编写过程中给予了热心的帮助和督促，在这里向她表示衷心的感谢。

本书适合从事 Altera 公司 FPGA 芯片开发设计的研究生和本科高年级学生使用，也适合从事该方面工作的工程师使用。市场上已经有了许多关于这方面内容的书籍，作者也不认为本书是最优秀的一本，但是我们意识到掌握一种技术必须做到理论和实践紧密结合，这就是本书的编写原则。由于本书的内容是通过一系列具有明确目的的设计任务来组织编写的，因此在内容的完整性方面可能有所欠缺，加之作者水平有限，书中的错误与不妥之处在所难免，敬请读者批评指正。

编著者

目　　录

第1章　引言 …… 1
1.1　数字集成电路的分类 …… 2
1.1.1　标准逻辑器件 …… 2
1.1.2　微处理器 …… 2
1.1.3　可编程逻辑器件 …… 3
1.2　Nios Ⅱ软核处理器 …… 3
1.2.1　Nios Ⅱ软核处理器系统简介 …… 4
1.2.2　可配置软核处理器的优点 …… 5
第2章　现场可编程门阵列器件 …… 6
2.1　可编程逻辑器件概述 …… 7
2.2　可编程逻辑器件的发展历程 …… 7
2.2.1　简单 PLD 的基本结构 …… 8
2.2.2　FPGA 的基本结构 …… 12
2.3　Altera 公司 Cyclone Ⅱ器件的工作原理 …… 14
2.4　Cyclone Ⅱ系列器件的主要技术指标 …… 19
2.5　小结 …… 22
第3章　Quartus Ⅱ开发软件的使用 …… 23
3.1　简介 …… 24
3.2　创建工程 …… 25
3.3　设计输入 …… 29
3.3.1　建立文本设计文件 …… 30
3.3.2　建立图形设计文件 …… 31
3.3.3　层次化设计 …… 33
3.4　设计的编译 …… 35
3.5　设计的仿真验证 …… 37
3.5.1　创建仿真波形文件 …… 37
3.5.2　设计仿真 …… 39
3.6　引脚分配 …… 41
3.7　器件配置 …… 42
3.8　小结 …… 44
第4章　VHDL 语言基础 …… 45
4.1　VHDL 的历史 …… 46
4.2　VHDL 的程序结构 …… 46

4.2.1 VHDL 程序的基本结构 …… 47
4.2.2 实体 …… 47
4.2.3 结构体 …… 49
4.2.4 包集 …… 50
4.2.5 库 …… 50
4.3 VHDL 的语言元素 …… 52
4.3.1 标识符 …… 52
4.3.2 对象类别与定义 …… 52
4.3.3 数据类型 …… 54
4.3.4 运算符 …… 56
4.4 并行语句 …… 58
4.4.1 并行信号赋值语句 …… 59
4.4.2 进程语句 …… 61
4.5 顺序语句 …… 62
4.5.1 顺序信号赋值语句 …… 62
4.5.2 条件（IF）语句 …… 62
4.5.3 选择（CASE）语句 …… 66
4.5.4 循环（LOOP）语句 …… 68
4.5.5 空操作（NULL）语句 …… 69
4.6 小结 …… 69
第 5 章 Nios Ⅱ 软核处理器系统的开发过程 …… 71
5.1 概述 …… 72
5.2 配置 Nios Ⅱ 软核处理器系统 …… 73
5.2.1 创建 Nios Ⅱ 软核处理器系统 …… 74
5.2.2 配置 Nios Ⅱ 软核处理器系统 …… 75
5.3 产生 Nios Ⅱ 软核处理器系统 …… 79
5.3.1 产生 Nios Ⅱ 软核处理器系统模块 …… 79
5.3.2 Nios Ⅱ 软核处理器系统的产生 …… 80
5.4 创建 Nios Ⅱ IDE 环境下的应用工程 …… 81
5.4.1 Nios Ⅱ IDE 工程创建 …… 81
5.4.2 C 语言源文件的编辑 …… 83
5.4.3 C 语言源文件的编译 …… 85
5.5 C 语言源程序的调试 …… 86
5.5.1 在目标电路板上运行程序 …… 86
5.5.2 在目标电路板上调试程序 …… 88
5.5.3 配置目标 FPGA 器件 …… 89
5.6 小结 …… 90
第 6 章 Nios Ⅱ 软核处理器——程序运行时间的测量 …… 91
6.1 Nios Ⅱ 软核处理器的结构 …… 92
6.2 Nios Ⅱ 软核处理器 …… 94

6.2.1 “Core Nios Ⅱ”选项卡 …… 94
6.2.2 “Caches and Memory Interfaces”选项卡 …… 96
6.2.3 “Advanced Features”选项卡 …… 98
6.2.4 “JTAG Debug Module”选项卡 …… 98
6.2.5 “Custom Instructions”选项卡 …… 100
6.3 程序运行时间的测量 …… 101
6.3.1 程序运行时间的测量方法 …… 102
6.3.2 Nios Ⅱ软核处理器对程序运行时间的影响 …… 103
6.4 浮点专用指令的使用 …… 105
6.4.1 浮点专用指令的添加 …… 105
6.4.2 浮点专用指令测试程序 …… 106
6.4.3 浮点专用指令测试结果 …… 108
6.5 小结 …… 109
第7章 8段数码管显示电路——并行输入/输出（PIO）内核的使用 …… 110
7.1 8段数码管 …… 111
7.2 并行输入/输出（PIO）内核 …… 112
7.2.1 并行输入/输出（PIO）内核的寄存器 …… 113
7.2.2 并行输入/输出（PIO）内核的配置 …… 113
7.2.3 C语言编程 …… 114
7.3 1位数码管的显示实验 …… 115
7.3.1 产生数码管的显示控制电路 …… 115
7.3.2 数码管显示控制程序 …… 116
7.3.3 目标芯片的配置 …… 119
7.4 多位数码管显示实验 …… 119
7.4.1 多位数码管显示控制电路 …… 119
7.4.2 4位数码管显示驱动函数 …… 121
7.4.3 4位数据的分离 …… 121
7.5 小结 …… 123
第8章 按键电路——中断的应用 …… 124
8.1 按键电路 …… 125
8.2 并行输入/输出（PIO）内核的中断 …… 126
8.2.1 并行输入/输出（PIO）内核涉及中断的相关寄存器 …… 126
8.2.2 并行输入/输出（PIO）内核中断的配置 …… 127
8.2.3 C语言编程 …… 130
8.3 Nios Ⅱ处理器的中断 …… 132
8.3.1 异常 …… 132
8.3.2 C语言编程 …… 133
8.4 1位按键电路的实验 …… 134
8.4.1 产生按键的控制电路 …… 134
8.4.2 1位按键控制程序 …… 135

8.4.3 数据类型 …… 137
8.4.4 alt_main( )和 main( )的区别 …… 138
8.5 4 位按键电路的实验 …… 138
8.5.1 4 位按键控制电路 …… 138
8.5.2 4 位按键控制程序 …… 139
8.6 小结 …… 142
第 9 章 Flash 的编程——EPCS 控制器、CFI 控制器的使用 …… 144
9.1 EPCS 控制器 …… 145
9.1.1 EPCS 控制器概述 …… 145
9.1.2 EPCS 控制器配置 …… 146
9.2 CFI 控制器 …… 147
9.2.1 CFI 控制器概述 …… 147
9.2.2 CFI 控制器配置选项 …… 148
9.2.3 CFI 控制器 C 语言编程 …… 149
9.3 Flash 的编程实例 …… 150
9.3.1 硬件系统的 SOPC 设计——从 EPCS 引导程序方式 …… 150
9.3.2 系统软件设计——从 EPCS 引导程序方式 …… 152
9.3.3 从 CFI_FLASH 引导程序方式 …… 154
9.4 小结 …… 155
第 10 章 时钟信号的产生与测量——定时器（Interval Timer）内核的使用 …… 157
10.1 定时器内核 …… 158
10.1.1 定时器内核的组成 …… 158
10.1.2 定时器内核的寄存器 …… 158
10.1.3 定时器（Interval Timer）内核的配置 …… 160
10.1.4 C 语言编程 …… 161
10.2 时钟信号产生实验 …… 162
10.2.1 时钟信号产生电路 …… 162
10.2.2 时钟信号产生电路控制程序 …… 164
10.3 定时器（Interval Timer）内核的中断实验 …… 165
10.4 信号的周期测量 …… 167
10.4.1 信号周期测量电路 …… 167
10.4.2 周期测量控制程序 …… 169
10.5 “看门狗”电路实验 …… 170
10.5.1 “看门狗”电路 …… 170
10.5.2 “看门狗”电路控制程序 …… 172
10.6 小结 …… 173
第 11 章 LCD12864 液晶模块的驱动设计——SDRAM 控制器内核的使用 …… 175
11.1 LCD12864 液晶模块简介 …… 176
11.2 SDRAM 控制器内核 …… 179

11.2.1　概述 …… 179
11.2.2　SDRAM 控制器内核的配置选项 …… 181
11.2.3　时钟、PLL 和时序 …… 184
11.2.4　SDRAM 内核的 C 语言编程 …… 186
11.3　LCD12864 模块的驱动实例 …… 187
11.3.1　LCD12864 模块接口电路 …… 187
11.3.2　硬件系统的 SOPC 设计 …… 189
11.3.3　系统软件设计 …… 192
11.4　小结 …… 204
第12章　JTAG UART 通信——JTAG UART 内核的使用 …… 205
12.1　JTAG UART 内核 …… 206
12.1.1　JTAG UART 内核概述 …… 206
12.1.2　JTAG UART 内核配置选项 …… 208
12.1.3　JTAG UART 内核的 C 语言编程 …… 210
12.2　JTAG UART 通信实例 …… 210
12.2.1　硬件系统的 SOPC 设计 …… 210
12.2.2　系统软件设计 …… 212
12.3　小结 …… 215
第13章　SD 卡读写控制设计——SPI 内核的使用 …… 216
13.1　SD 卡简介 …… 217
13.2　SPI 内核 …… 219
13.2.1　SPI 内核综述 …… 219
13.2.2　SPI 内核配置选项 …… 221
13.2.3　SPI 的 C 语言编程 …… 223
13.3　SD 卡读写实例 …… 224
13.3.1　SD 卡与 FPGA 接口电路 …… 224
13.3.2　硬件系统的 SOPC 设计 …… 224
13.3.3　系统软件设计 …… 227
13.4　小结 …… 236
第14章　数字信号频率测量电路的设计 …… 237
14.1　频率测量原理和电路设计要求 …… 238
14.2　频率测量电路的系统设计 …… 238
14.3　频率测量电路的单元电路设计 …… 240
14.3.1　计数器电路 …… 240
14.3.2　计数器控制电路 …… 242
14.3.3　Nios Ⅱ 软核处理器系统 …… 245
14.4　频率测量电路的系统调试 …… 248
14.4.1　系统集成 …… 248
14.4.2　测试方案和使用仪器 …… 250

14.4.3 测量数据及数据分析 …… 250
14.5 设计总结 …… 251
第15章 信号频谱分析电路的设计 …… 252
15.1 频谱的概念 …… 253
15.2 离散傅里叶变换 …… 253
15.3 信号频谱分析电路的系统设计 …… 255
15.4 信号频谱分析电路的单元电路设计 …… 256
15.4.1 频谱分析模块 …… 256
15.4.2 采样速率控制电路 …… 264
15.5 信号频谱分析电路的系统调试 …… 268
15.5.1 系统集成 …… 268
15.5.2 信号频谱分析程序框图 …… 270
15.5.3 系统测量和数据分析 …… 271
15.6 设计总结 …… 272
参考文献 …… 274

第1章 引 言

- 数字集成电路的分类
- Nios Ⅱ软核处理器

1.1 数字集成电路的分类

尽管本书的目的是讨论如何利用在片可编程系统（System On Programmable Chip，SOPC）解决方案，使用 Nios Ⅱ 软核处理器系统实现要求的设计功能，但是考察一下可供选择的各种器件对数字系统的设计者来说还是有益的，因为它有助于我们更好地理解所有可供选择的方案。同时我们也可以意识到虽然描述数字系统和数字电路的方法和用来实现这些方法的技术在不断变化，但是作为理论基础的基本原理并没有改变。

1.1.1 标准逻辑器件

标准逻辑器件在集成度方面属于中小规模集成电路。它包括各种逻辑门、触发器、译码器、多路选择器、寄存器和计数器等器件。标准逻辑器件有 3 种主要类型：TTL、CMOS 和 ECL。TTL 是一种成熟的技术，新的系统设计已经很少采用 TTL 逻辑器件，但是正在运行的系统中仍然包含这种器件。CMOS 器件是当前最流行的标准逻辑器件，它的优点是功耗低。ECL 器件主要用于高速系统中。

作为传统数字系统设计中使用的主要器件，标准逻辑器件已经使用了 40 多年。标准逻辑器件的产量很大，生产成本低廉，价格便宜。当我们的设计不是很复杂时，这些器件仍然是有用的。标准逻辑器件对于研究数字系统基本构成模块的工作原理具有重要的意义，它在许多基础的理论和实验教学课程中仍然占据重要的位置。《数字电路逻辑设计》课程目前仍然以标准逻辑器件为主进行讲授。

由于标准逻辑器件集成度较低，因此采用它们设计数字系统需要较多的器件，这就使得电路连线复杂，系统的可靠性降低。由于用户无法修改这类器件的功能，要修改系统设计就必须通过对电路的重新设计和组装来实现，因此设计的灵活性很低。

1.1.2 微处理器

数字技术已经进入众多的技术领域，其中数字计算机是最著名的，也是应用最广泛的。尽管计算机影响了人类生活的许多方面，但是许多人并不能准确地说出它的主要特点。简单地说，计算机是一个能完成信号的算术运算、逻辑运算和比较判断的数字系统。

微型计算机（PC）是最常见的计算机，它由一些数字集成电路芯片组成，这些芯片包括微处理器芯片、存储器，以及输入/输出接口等。在大多数情况下，凡是人能干的，计算机都能干，而且计算机还能干得更快更精确。尽管事实上计算机每次只能完成所有计算中的一步，但是计算机完成每一步的速度非常快，它的高速度弥补了它的低效率。

微处理器依靠所运行的软件（程序）来完成工作。这个软件是人们给微处理器的一组完整的指令，指令告诉微处理器其操作的每一步应该干什么。这些指令以二进制代码的形式存储在存储器中，微处理器从存储器中一次读取一条指令代码，并完成由指令代码指定的操作。

通过编写软件可以控制微处理器完成不同的工作，这个特点使得设计的灵活性得到了提

高。当修改系统设计时，设计者只需要改变软件，不需要或者较少需要修改电路连线。由于微处理器一次只能执行一条指令，因此它的主要局限性是工作速度。采用硬件方案设计的数字系统总是比采用软件方案设计的数字系统的工作速度快。

1.1.3 可编程逻辑器件

微处理器的每一条汇编指令对应一个电路模块，可编程逻辑器件可以认为是对标准逻辑器件的直接升级。它在一块芯片内部集成了非常多的逻辑门和触发器，使得一块可编程逻辑器件芯片具有了实现一个应用系统的逻辑资源，从而减少了系统中使用器件的数量，提高了系统的可靠性。芯片内部的逻辑资源的连接不需要手工进行，与微处理器一样，用户只需要编写设计文件即可，从而使得设计灵活性得到了提高。

可编程逻辑器件在下载设计文件后，在它的内部将形成对应的硬件电路，这些电路是可以同时工作的。例如，向两个数码管传送显示代码，这时可以同时进行，而在微处理器中，这个工作是逐个传送的。可编程逻辑器件内部电路模块中信号传输的时间延迟只来源于硬件电路产生的时间延迟，不存在指令读取和执行产生的时间延迟。上述工作特点使得可编程逻辑器件的工作速度比微处理器芯片快。

可编程逻辑器件的开发设计语言有许多种，其中 VHDL 和 Verilog HDL 这两种硬件描述语言已经获得较为广泛的应用，并且成为 IEEE 的标准。以 VHDL 为例，创建 VHDL 的最初目标是用于标准文档的建立和系统功能的模拟，基本想法是在高层次上描述系统或元件的行为。到了 20 世纪 90 年代初，人们发现 VHDL 不仅可以作为系统模拟的建模工具，而且可以作为电路和系统的设计工具，即可以利用软件工具把 VHDL 源代码转换为基本逻辑元件的连接关系。

VHDL 没有得到可编程逻辑器件开发软件的全面支持。例如，Altera 公司的 Quartus Ⅱ 可编程逻辑器件开发软件不支持浮点类型数据，不支持乘法和除法算术运算等。这些不足限制了可编程逻辑器件的应用。不支持浮点类型数据减小了所处理信号的动态范围，不支持乘、除法算术运算使得一些信号处理功能不容易实现。在这方面，微处理器开发中大量使用的 C 语言具有明显的优势。

1.2 Nios Ⅱ 软核处理器

基于 VHDL 语言设计的可编程逻辑器件应用电路具有数据传送速度快的优点，基于 C 语言设计的微处理器应用电路具有数据处理能力强的优点。如果希望应用电路同时具有数据传送速度快和数据处理能力强这两个优点，则可联合使用可编程逻辑器件和微处理器这两种器件。

要联合使用可编程逻辑器件和微处理器，必须先实现它们之间的通信。在需要完成高速数据传送的情况下，这对电路板的设计和制作都提出了比较高的要求。Altera 公司提出的数字系统 SOPC 解决方案使得处理器能够配置到可编程逻辑器件之中，这样的处理器被称为软核处理器。

1.2.1 Nios Ⅱ 软核处理器系统简介

Altera 公司于 2000 年推出了能够配置到可编程逻辑器件之中的软核处理器系统，它被称为 Nios 处理器。2004 年，Altera 公司又推出了升级产品——Nios Ⅱ 软核处理器系统。Nios Ⅱ 软核处理器是一种 32 位 RISC 嵌入式处理器，具有超过 200DMIP 的性能，在低成本的 FPGA 芯片中的实现成本只有几十美分。

Nios Ⅱ 软核处理器系统的最大特点为它是一种软核、可配置的系统。可配置意味着处理器系统的组成和性能可以根据需要进行调整。如图 1.1 所示为开发软件中 Nios Ⅱ 软核处理器配置窗口的部分内容。

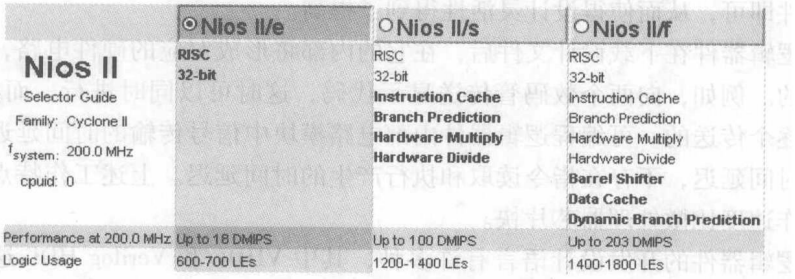


图 1.1 Nios Ⅱ 软核处理器的配置窗口

Nios Ⅱ 软核处理器具有 3 种内核供用户选择，它们分别为经济型内核“Nios Ⅱ/e”、标准型内核“Nios Ⅱ/s”和快速型内核“Nios Ⅱ/f”。快速型内核“Nios Ⅱ/f”具有最高的程序执行速度，在 200MHz 系统时钟的情况下可以获得 203DMIP 的性能，但是实现这样的内核将占用可编程逻辑器件 1 400 ~ 1 800 个逻辑单元。同样在 200MHz 系统时钟的情况下，经济型内核“Nios Ⅱ/e”只能获得 18DMIP 的性能，不过这时内核只占用可编程逻辑器件 600 ~ 700 个逻辑单元。

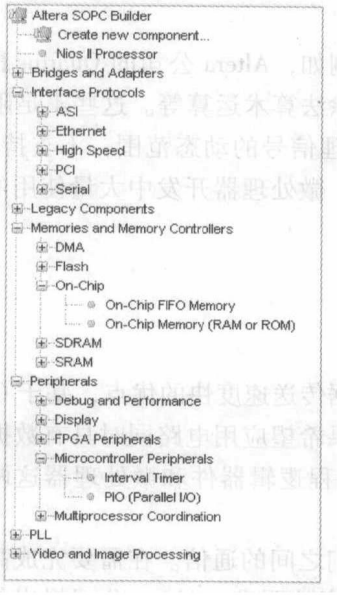


图 1.2 Nios Ⅱ 软核处理器外围模块的配置窗口

Nios Ⅱ 软核处理器还可以通过添加硬件进一步提高程序的执行速度，如可以添加硬件乘法器来实现乘法运算。在没有向 Nios Ⅱ 软核处理器中添加硬件乘法器时，乘法运算将通过调用函数来完成；在向 Nios Ⅱ 软核处理器中添加硬件乘法器后，乘法运算将通过硬件电路来实现。对于完成一项指定的操作，使用硬件完成总是比使用软件完成快，当然这需要使用额外的逻辑资源。

Nios Ⅱ 软核处理器系统还提供了 60 多种处理器外围模块，如图 1.2 所示。这些外围模块包括存储器、输入/输出接口和定时器等，可供系统设计时选择。

创建 Nios Ⅱ 软核处理器系统的工具 SOPC Builder 被包含在可编程逻辑器件开发软件 Quartus Ⅱ 之中，创建过程与在

Quartus Ⅱ开发环境下利用原理图输入方法开发可编程逻辑器件的过程类似。

生成的Nios Ⅱ软核处理器系统可以利用Quartus Ⅱ开发环境提供的原理图输入方式与Quartus Ⅱ开发软件提供的其他基本电路单元、宏功能模块，以及用户利用硬件描述语言产生的电路模块连接在一起，实现系统集成，并一起下载到一片可编程逻辑器件之中。

Nios Ⅱ软核处理器系统的最大特点为它是一种软核、可配置的系统。软核表示处理器的目标器件、可编程逻辑器件，只有在下载设计文件后才具有处理器的功能。Nios Ⅱ软核处理器系统可以工作在Altera公司的Stratix、Stratix Ⅱ、Cyclone、Cyclone Ⅱ等系列的可编程逻辑器件上。

1.2.2 可配置软核处理器的优点

1. 提供合理的性能

基于处理器进行应用系统设计时，设计者很难在众多的处理器芯片中选择到合适的芯片。因为处理器芯片的功能和技术指标是确定的，为了满足某一些性能和技术指标，常常不得不选择一种还具有多余功能的处理器芯片，这就导致了产品成本的提高。若为了满足产品成本的要求，常常又会使设计达不到理想的性能和技术指标。

采用Nios Ⅱ软核处理器系统，设计者能够根据自己的想法来配置处理器系统，包括选择合适的处理器内核、希望的外部设备，以及处理器与外部设备之间的接口。设计者还能够在Nios Ⅱ软核处理器系统中集成自己专有的逻辑功能，如浮点算术运算电路、数字信号处理（DSP）电路等。

2. 提升系统的性能

设计者不仅可以采用快速型内核“Nios Ⅱ/f”，并添加硬件电路来提升Nios Ⅱ软核处理器系统的性能，还可以通过在一个系统中添加多个处理器内核来提高应用系统的性能。

Altera公司现在提供多个系列的可编程逻辑器件，既有包含大量逻辑资源的芯片，又有经济、价廉的芯片。例如，高性能的Stratix Ⅱ系列的EP2S180芯片，一个Nios Ⅱ软核处理器只占用这种芯片1%的逻辑资源，这使得在一个可编程逻辑器件中能够实现多个Nios Ⅱ软核处理器系统，这样便提高了应用系统的性能。

4. 延长产品的生命周期

使用Nios Ⅱ软核处理器进行应用系统设计可以从以下几个方面延长产品的生命周期。

首先，在硬件方面，Nios Ⅱ软核处理器的目标器件为可编程逻辑器件，它的可编程特性避免了专用集成电路设计制造周期长的缺点；在软件方面，Altera公司提供了完整的开发软件、大量的设计参考和调试电路。

其次，Nios Ⅱ软核处理器系统的软核、可配置特点使得用户能够容易地对应用系统的硬件电路部分进行升级。即使产品已经交给了用户，仍然可以方便地进行升级。

最后，Nios Ⅱ软核处理器系统可以方便地移植到新的可编程逻辑器件中。例如，当一个设计被确定，并且准备大批量生产时，可以选择将它移植到Altera公司的HardCopy（一种结构化的专用集成电路）上，使成本降低。