

电子技术论坛鼎力推荐
27个应用实例、3个综合案例

Altera FPGA 工程师成长手册

(8小时多媒体教学视频)

陈欣波 等编著

- ◎ 基于多年教学经验，合理安排理论知识和实践内容
- ◎ 按照学习的认知次序组织内容，力求深入浅出，简单易懂
- ◎ 涵盖从基本逻辑电路设计到DSP模块设计，再到基于软核处理器的设计
- ◎ 列举大量实例讲解难于理解的内容，并给出详细说明和实现步骤
- ◎ 着力贯彻自顶向下的设计思路，培养层次化和模块化的设计思想



清华大学出版社

Altera FPGA 工程师成长手册

陈欣波 等编著

清华大学出版社

北 京

内 容 简 介

本书以 Altera 公司的 FPGA 为例, 由浅入深, 全面、系统地详细讲述了基于可编程逻辑技术的设计方法。本书讲解时穿插了大量典型实例, 便于读者理解和演练。另外, 为了帮助读者更好地学习, 本书提供了配套语音教学视频, 请在清华大学出版社网站上搜索到本书页面后查看下载方式。

本书涉及面广, 从基本的软件使用到一般电路设计, 再到 Nios II 软核处理器的设计, 几乎涉及 FPGA 开发设计的所有知识。具体内容包括: EDA 开发概述、Altera Quartus II 开发流程、Altera Quartus II 开发向导、VHDL 语言、基本逻辑电路设计、宏模块、LPM 函数应用、基于 FPGA 的 DSP 开发设计、SOPC 系统构架、SOPC 系统硬件开发、SOPC 系统软件开发、Nios II 常用外设、LogicLock 优化技术等。

本书适合学习 FPGA 开发设计的各个院校的本科学生阅读, 也适合各类使用 FPGA 进行开发的初级工程技术人员使用。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目 (CIP) 数据

Altera FPGA 工程师成长手册 / 陈欣波等编著. —北京: 清华大学出版社, 2012. 6
ISBN 978-7-302-28099-6

I. ①A… II. ①陈… 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 (2012) 第 030228 号

责任编辑: 夏兆彦

封面设计: 欧振旭

责任校对: 徐俊伟

责任印制: 王静怡

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

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 25.5 字 数: 637 千字

版 次: 2012 年 6 月第 1 版 印 次: 2012 年 6 月第 1 次印刷

印 数: 1~5000

定 价: 49.00 元

前 言

目前,随着高性能 FPGA 的出现,在数字系统的设计中 FPGA 几乎无所不能,广泛应用于数字产品的各个领域。FPGA 技术具备开发成本低和上市速度快的特点,只要安装相应的开发软件并具备一套简陋的开发板就可以进行创新设计,这为具有创新能力的个人和小型公司提供了生存的机会。

笔者从事多年硬件描述语言、FPGA 理论和实践教学工作。发现学生在学习的过程中缺乏相关的背景知识;对使用硬件描述语言编写的较长程序理解不深;没有树立现代电子设计中处理器+存储器+外设=系统的理念。基于教学经验,笔者根据学习的认知习惯编写了这本书,希望各位读者能在本书的引领下跨入 FPGA 开发设计的大门,学习完本书后,读者应该可以具备独立进行项目开发的能力。

本书特色

1. 提供大量源代码,学习效果好

本书提供了大量的源程序和实例讲解,使读者能直观地学习本书内容,提高学习效率。这些源代码收录于配书光盘中。

2. 内容全面、系统、深入

本书介绍了 Quartus II 软件的基础知识、设计流程、宏模块和 LPM 函数在设计中的应用、DSP Builder 软件的使用、基于 Nios II 软核处理器的硬件和软件设计方法,最后还详细介绍了案例的开发。

3. 讲解由浅入深,循序渐进,适合各个层次的读者阅读

本书从 FPGA 技术的基础开始讲解,逐步深入到嵌入 Nios II 软核处理器的高级开发技术及应用,内容梯度从易到难,讲解由浅入深,循序渐进,适合各个层次的读者阅读。

4. 贯穿大量的开发实例和技巧,迅速提升开发水平

本书在讲解知识点时贯穿了大量短小精悍的典型实例,并给出了大量的开发技巧,帮助读者更好地理解各种概念和开发技术,体验实际编程,迅速提高开发水平。

5. 提供技术支持,答疑解惑

读者阅读本书时有任何疑问,可发 E-mail 到 pzhueric@163.com 或者 lymlrl@163.com 获得相关帮助。

本书内容及体系结构

第1篇 FPGA开发基础（第1～5章）

本篇主要内容包括：Quartus II 软件的基本知识和使用方法、VHDL 语言的基本知识、使用 VHDL 语言描述基本逻辑电路的方法。通过本篇的学习，读者可以掌握 FPGA 开发的软件的使用和设计流程。

第2篇 FPGA实例开发（第6～7章）

本篇主要内容包括：宏模块和 LPM 函数在设计中的应用、使用 DSP Builder 设计 DSP 器件。通过本篇的学习，读者可以掌握存储器、锁相环等宏模块的使用，并掌握使用 MATLAB 进行算法仿真和在 Quartus II 中进行硬件仿真的方法。

第3篇 FPGA高级应用（第8～14章）

本篇主要内容包括：Nios II 软核处理器的基础知识、基于 Nios II 软核处理器的硬件和软件的设计方法、基于 LogicLock 的优化技术、数字系统设计实例。通过本篇的学习，读者可以掌握基于 Nios II 软核处理器的嵌入式设计方法。

本书读者对象

- ☐ FPGA 开发初学者；
- ☐ 想全面学习 FPGA 开发技术的人员；
- ☐ 利用 FPGA 做开发的工程技术人员；
- ☐ FPGA 开发爱好者；
- ☐ 大中专院校的学生；
- ☐ 社会培训班学员。

本书作者

本书由陈欣波编写。其他参与编写的人员有陈世琼、陈欣、陈智敏、董加强、范礼、郭秋滢、郝红英、蒋春蕾、黎华、刘建准、刘霄、刘亚军、刘仲义、柳刚、罗永峰、马奎林、马味、欧阳昉、蒲军、齐凤莲、王海涛、魏来科、伍生全、谢平。在此表示感谢！

编著者

目 录

第 1 篇 FPGA 开发基础

第 1 章 EDA 技术概述	2
1.1 EDA 技术及发展	2
1.1.1 何谓 EDA 技术	2
1.1.2 基于大规模可编程逻辑器件的数字系统设计	2
1.2 可编程逻辑器件的发展简介	3
1.2.1 逻辑设计基本流程	3
1.2.2 可编程逻辑器件 PAL	5
1.2.3 从 PAL 到 PLD 到复杂可编程逻辑器件 CPLD	7
1.2.4 从 CPLD 到 FPGA 的产生	9
1.2.5 在系统编程问题的解决	9
1.3 FPGA 系统结构和资源	10
1.3.1 可编程逻辑单元 (LE)	10
1.3.2 可编程布线	12
1.3.3 可编程 I/O	13
1.3.4 嵌入式存储器 RAM	13
1.3.5 嵌入式乘法器	13
1.3.6 时钟	13
1.3.7 锁相环	14
1.3.8 FPGA 与 CPLD 的对比	14
1.4 FPGA 的设计流程	15
1.5 Altera 公司 FPGA 低成本器件——Cyclone II	16
1.5.1 主要特性	16
1.5.2 基于数字信号处理 (DSP) 应用	17
1.5.3 专用外部存储器接口	17
1.5.4 嵌入式锁相环	18
1.5.5 单端 I/O 特性	19
1.5.6 差分 I/O 特性	20
1.5.7 自动 CRC 检测	20
1.5.8 支持 Nios II 嵌入式处理器	21

1.6	Altera 公司 FPGA 低成本器件——Stratix III 器件	21
1.6.1	主要特性	21
1.6.2	体系架构	22
1.6.3	TriMatrix 嵌入式存储器	23
1.6.4	DSP 块	24
1.6.5	时钟网络和锁相环	25
1.6.6	高速 I/O 信号和接口	25
1.6.7	设计安全性	26
1.7	小结	27
第 2 章	Altera Quartus II 软件开发流程	28
2.1	Quartus II 综述	28
2.1.1	Quartus II 软件的特点	28
2.1.2	Quartus II 设计软件的流程和集成的工具	29
2.1.3	Quartus II 软件的用户界面	30
2.2	设计输入	33
2.2.1	建立工程	33
2.2.2	输入方式	34
2.3	约束输入	36
2.3.1	使用分配编辑器	36
2.3.2	使用引脚规划器	38
2.3.3	使用 Settings 对话框	39
2.4	综合	40
2.4.1	使用 Quartus II 软件集成的综合工具	40
2.4.2	使用其他 EDA 综合工具	42
2.4.3	使用 RTL 查看器和状态机查看器分析综合结果	43
2.5	布局布线	45
2.5.1	设置 Fitter 选项	46
2.5.2	设置物理综合优化选项	48
2.5.3	通过反标保留分配	50
2.6	仿真	50
2.6.1	Quartus II 仿真器设置	52
2.6.2	建立用于仿真的波形文件	53
2.7	编程与配置	56
2.7.1	建立编程文件	56
2.7.2	器件编程和配置	59
2.8	小结	60
第 3 章	Altera Quartus II 软件开发向导	61
3.1	模块编辑及设计流程	61
3.1.1	原理图输入文件的建立	61
3.1.2	图表模块输入	65

3.1.3	原理图设计流程	71
3.1.4	波形仿真	77
3.1.5	引脚分配	79
3.1.6	下载验证	79
3.1.7	Quartus II 的几个常用功能	83
3.2	文本编辑及设计流程	84
3.2.1	建立文本文件	84
3.2.2	文本设计流程——建立新工程	88
3.2.3	文本设计流程——编译工程	88
3.2.4	文本设计流程——建立矢量波形文件	89
3.2.5	文本设计流程——仿真波形	92
3.2.6	文本设计流程——引脚分配及下载验证	94
3.3	混合设计	94
3.3.1	建立计数器文件	94
3.3.2	建立七段译码显示电路文件	95
3.3.3	设计流程	96
3.4	使用 Signal Tap II 的实时测试	100
3.4.1	打开 Signal Tap II 的编辑窗口	100
3.4.2	调入待测信号	100
3.4.3	设置 Signal Tap II 参数	102
3.4.4	文件存盘	103
3.4.5	编译选择	104
3.4.6	启动 Signal Tap II 进行采样分析	104
3.4.7	Signal Tap II 的其他设置和控制方法	104
3.5	小结	105
第 4 章	VHDL 语言基础	106
4.1	VHDL 语言基本结构	106
4.1.1	实体	108
4.1.2	结构体	109
4.1.3	配置	110
4.1.4	库	111
4.2	VHDL 语言要素	114
4.2.1	VHDL 语法规则	114
4.2.2	VHDL 数据对象	116
4.2.3	数据类型	118
4.2.4	操作符	120
4.3	顺序语句	123
4.3.1	赋值语句	123
4.3.2	IF 语句	124
4.3.3	CASE 语句	126

4.3.4	LOOP 语句	127
4.3.5	跳出循环的语句	129
4.3.6	RETURN 语句	130
4.3.7	NULL 语句	131
4.4	并行语句	131
4.4.1	并行信号赋值语句	131
4.4.2	进程 (PROCESS) 语句	133
4.5	子程序	136
4.5.1	过程	137
4.5.2	函数	140
4.6	VHDL 语言描述风格	143
4.6.1	行为描述	143
4.6.2	数据流描述	144
4.6.3	结构化描述	145
4.7	小结	148
第 5 章	基本逻辑电路设计	149
5.1	组合逻辑电路设计	149
5.1.1	门电路设计	149
5.1.2	三态门及总线缓冲器设计	151
5.1.3	编码器、译码器设计	153
5.1.4	多路数据选择器和多路数据分配器设计	155
5.2	时序逻辑电路设计	157
5.2.1	触发器设计	158
5.2.2	寄存器设计	159
5.2.3	计数器设计	161
5.3	有限状态机电路设计	165
5.3.1	有限状态机概述	165
5.3.2	有限状态机的算法描述	166
5.3.3	有限状态机的 VHDL 描述模式	167
5.4	设计实例: 交通信号灯控制器设计	171
5.4.1	交通信号灯控制器的设计要求	171
5.4.2	交通信号灯控制器的设计分析	172
5.5	小结	179

第 2 篇 FPGA 实例开发

第 6 章	宏模块和 LPM 函数的应用	182
6.1	存储器模块的用法	182
6.1.1	RAM 的使用	182

6.1.2 ROM 的建立过程	187
6.1.3 FIFO 的建立使用	191
6.2 乘法器和锁相环的使用	193
6.2.1 乘法器的使用	193
6.2.2 锁相环的使用	196
6.3 NCO IP 核的使用	199
6.4 基于宏模块的设计实例	204
6.4.1 正弦波信号发生器的设计	204
6.4.2 流水线乘累加器的设计	205
6.5 小结	208
第 7 章 基于 FPGA 的 DSP 开发设计	209
7.1 概述	209
7.2 DSP Builder 功能简介与设计流程	210
7.2.1 DSP Builder 功能简介	210
7.2.2 DSP Builder 设计流程	210
7.3 基于 DSP Builder 技术的设计示例——调幅电路	212
7.3.1 在 MATLAB/Simulink 中建立算法模型	212
7.3.2 准备工作	212
7.3.3 在新模型窗口中添加单元模块	215
7.3.4 在 Simulink 环境中仿真	219
7.3.5 在 Modelsim 环境中进行功能仿真	222
7.3.6 在 Quartus II 环境中进行时序仿真	224
7.4 基于 DSP Builder 的层次化设计——FIR 滤波器	229
7.4.1 FIR 滤波器的原理	229
7.4.2 建立系统设计模型	230
7.4.3 建立子系统的模型	232
7.4.4 在 Simulink 和 Modelsim 中仿真	234
7.5 Megacore 函数的使用	236
7.5.1 安装 Megacore 函数	236
7.5.2 使用 Megacore 函数的设计流程	236
7.5.3 使用 Megacore 函数设计 FIR 滤波器	236
7.6 小结	241

第 3 篇 FPGA 高级应用

第 8 章 SOPC 技术开发概述	244
8.1 SOPC 的概念	244
8.2 SOPC 系统的核心——Nios II 处理器	245
8.3 SOPC 系统开发流程	247

8.3.1 SOPC Builder 的设计流程	247
8.3.2 SOPC 的设计阶段	248
8.4 SOPC 系统开发环境	249
8.5 小结	251
第 9 章 SOPC 系统构架	252
9.1 Nios II 处理器体系结构	252
9.1.1 Nios II 的内部寄存器	254
9.1.2 Nios II 存储器与 I/O 组织	256
9.2 Nios II 的异常处理	259
9.2.1 硬件中断	259
9.2.2 软件异常	259
9.2.3 Nios II 的异常处理流程	260
9.3 算术逻辑单元和复位信号	261
9.3.1 算术逻辑单元	261
9.3.2 复位信号	262
9.4 JTAG 调试模块	262
9.5 Avalon 总线	263
9.5.1 Avalon 互连规范	264
9.5.2 Avalon 总线的概念	265
9.5.3 Avalon 总线信号	267
9.5.4 Avalon 的中断与复位信号	271
9.5.5 Avalon 总线传输	272
9.6 小结	272
第 10 章 SOPC 系统硬件开发	273
10.1 数字钟的设计要求	273
10.2 硬件开发流程	273
10.3 创建 Quartus II 工程	274
10.3.1 创建 Quartus II 工程	274
10.3.2 创建顶层实体文件	276
10.4 创建 Nios II 系统模块	277
10.4.1 创建新系统	277
10.4.2 加入 Nios II 处理器	278
10.4.3 加入外围模块	280
10.4.4 分配系统各 IP 模块的地址和中断号分配、Nios II 系统配置	284
10.4.5 生成 Nios II 并添加到工程中	286
10.4.6 建立锁相环 PLL 模块	288
10.5 编译和下载	294
10.5.1 引脚分配	294
10.5.2 配置工程	295
10.5.3 编译设计	296

10.5.4 程序配置下载	296
10.6 小结	297
第 11 章 SOPC 系统软件开发	298
11.1 Nios II IDE 简介	298
11.1.1 Nios II IDE 的功能模块	298
11.1.2 Nios II IDE 开发流程	300
11.2 基于 Nios II IDE 软件示例——数字钟软件	301
11.2.1 建立软件工程	301
11.2.2 编译工程	305
11.2.3 运行	308
11.3 数字钟的程序设计	310
11.4 HAL 系统库	313
11.4.1 HAL 简述	313
11.4.2 目前提供的主要 HAL 资源	315
11.5 使用 HAL 开发应用程序	317
11.6 小结	318
第 12 章 Nios II 常用外设使用	319
12.1 并行输入/输出内核 (PIO)	319
12.1.1 PIO 内核简介	319
12.1.2 PIO 内核的配置	320
12.1.3 PIO 内核的 C 语言编程	323
12.2 定时器的使用	328
12.2.1 内核定时器简介	329
12.2.2 定时器内核的配置	330
12.2.3 定时器内核的 C 语言编程	331
12.3 Flash 接口控制器的使用	333
12.3.1 Flash 接口控制器简介	333
12.3.2 CFI 控制器的配置	334
12.3.3 CFI 控制器的 C 语言编程	335
12.4 SDRAM 控制器的使用	336
12.4.1 SDRAM 控制器内核概述	336
12.4.2 SDRAM 内核配置	338
12.4.3 软件编程	340
12.5 UART 的使用	341
12.5.1 UART 内核简介	341
12.5.2 UART 内核的寄存器	343
12.5.3 UART 内核配置	346
12.5.4 软件编程	348
12.6 小结	350
第 13 章 LogicLock 优化技术	351
13.1 LogicLock 优化技术简介	351

13.1.1	LogicLock 设计方法目标	351
13.1.2	LogicLock 的区域	352
13.1.3	锁定区域的基本方式	353
13.1.4	LogicLock 技术的应用流程	354
13.2	为应用 LogicLock 技术准备的具体实例	355
13.2.1	数字滤波器结构及其 VHDL 描述	355
13.2.2	滤波器设计和结果	358
13.3	LogicLock 优化设计——底层模块设计	359
13.3.1	建立底层模块工程	360
13.3.2	建立父区域	360
13.3.3	定义逻辑锁定子区域	363
13.3.4	将设计实体移至锁定区域	365
13.3.5	编译优化锁定后的 filter 模块	365
13.3.6	输出逻辑锁定后的 VQM 文件	367
13.4	LogicLock 优化设计——顶层设计	368
13.4.1	建立顶层工程	368
13.4.2	将 VQM 文件加到顶层工程中	368
13.4.3	导入 LogicLock 约束	370
13.5	小结	373
第 14 章	数字系统设计实例——电子乐器	374
14.1	电子乐器的设计	374
14.1.1	设计要求	374
14.1.2	设计原理	374
14.1.3	乐曲硬件演奏电路的层次化设计方案	376
14.2	FFT 设计	381
14.2.1	FFT 的原理	381
14.2.2	基于 DSP Builder 设计 FFT 的方法	382
14.3	SD 卡驱动的设计	384
14.3.1	SD 卡和 SPI 内核简介	385
14.3.2	SD 卡与 FPGA 接口电路	386
14.3.3	硬件系统的 SOPC 设计	387
14.3.4	系统软件设计	388
14.4	小结	394

第 1 篇 *FPGA* 开发基础

- ▶▶ 第 1 章 EDA 技术概述
- ▶▶ 第 2 章 Altera Quartus II 软件开发流程
- ▶▶ 第 3 章 Altera Quartus II 软件开发向导
- ▶▶ 第 4 章 VHDL 语言基础
- ▶▶ 第 5 章 基本逻辑电路设计

第 1 章 EDA 技术概述

当今数字电子技术得到了飞速发展，有力地推动和促进了社会生产力和社会信息化的提高。数字电子技术逐步渗透到人类生活的各个领域。从消费电子产品、工业自动化设备到航天技术都能看到数字电子技术的身影。在技术发展的同时，电子产品的设计方式也发生了巨大的改变，采用 EDA（电子设计自动化）技术成为数字电子设计的主要方式。

1.1 EDA 技术与发展

数字电子技术工程师现在正面临着前所未有的挑战。一方面，电子公司要求工程师在更短的时间里，使用更少的资源来设计新产品，而且性能要比竞争产品好。另一方面，技术变化非常快，不同的客户有完全不同的需求，要求有更具个性化的产品。因此，EDA 技术应运而生，成为解决以上“所有”问题的很好的技术选择。

1.1.1 何谓 EDA 技术

EDA 技术是一门迅速发展的新技术。它以大规模可编程逻辑器件为设计载体，以硬件描述语言为系统逻辑描述的主要表达方式，以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具。它能用软件的方式设计电子系统，自动完成硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真，最后在特定的目标芯片中完成适配编译、逻辑映射、编程下载等工作，形成集成电子系统或专用集成芯片。利用 EDA 技术进行电子系统的设计具有以下几个特点。

- ❑ 用软件的方式设计硬件。
- ❑ 用软件的方式设计的系统到硬件系统的转换是由有关的开发软件自动完成的。
- ❑ 设计过程中可用有关软件进行各种仿真。
- ❑ 系统可现场编程，在线升级。
- ❑ 整个系统可集成在一个芯片上，体积小、功耗低、可靠性高。因此，EDA 技术是现代电子设计的发展趋势。

EDA 技术是数字系统设计的核心技术，是电子类专业技术人员必须掌握的基本技能之一。目前的大规模可编程逻辑器件是 CPLD（复杂可编程逻辑器件）和 FPGA（可编程逻辑阵列）。

1.1.2 基于大规模可编程逻辑器件的数字系统设计

现代数字系统设计相当大一部分是基于大规模可编程逻辑器件的，这是因为基于大规

模可编程逻辑器件的设计拥有面市时间快、灵活性大、可定制解决方案、开发成本低和具有现场更新能力等优点。工程师首先对系统或者设计进行构思，然后在计算机上采用高级语言（Verilog HDL 语言或者 VHDL 语言）来描述这一构思，设计出软件代码。

最重要的是，可以使用设计工具软件检查设计中有没有错误。确定设计适合目标的可编程逻辑器件后，检查设计是否达到了性能要求，可以把设计下载到目标器件中，直接在硬件中调试功能。

本书讨论的大规模可编程逻辑器件是 FPGA。现在的 FPGA 设计与几年前的 FPGA 设计有很大不同，它具有全功能，可以实现电路板级的集成，同时降低了成本。例如，一个典型的系统设计如图 1.1 所示。

这块电路板上有很多芯片，诸如 CPU、I/O 单元、小规模 FPGA、闪存和 SDRAM 存储器及一个 DSP 模块。该系统需要的电路板比较大，这样才能容纳这些芯片。这提高了设计成本和复杂度。但是，现在能够把 CPU、I/O 和 DSP 都放在一个可编程逻辑芯片中，如图 1.2 所示。采用一片 FPGA，可以从系统中去掉很多硬件，从而降低了成本和功耗。

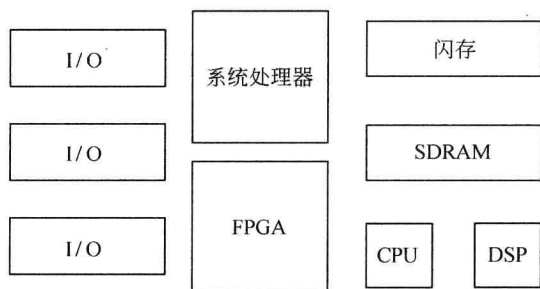


图 1.1 典型的系统设计框图

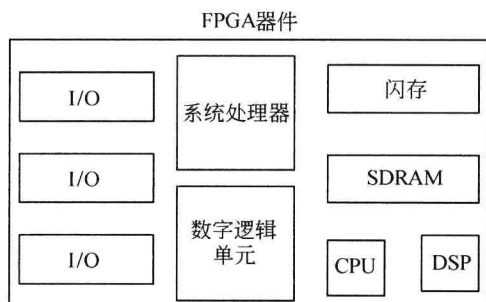


图 1.2 集成后的系统

现在可编程逻辑几乎无处不在。从家里的 HDTV 到附近的蜂窝电话发射塔，直至银行的 ATM，这些都含有可编程逻辑器件形式的数字逻辑，它实现了从控制系统怎样工作的 CPU 到网络和通信应用中高速交换流量管理等方方面面的功能。

1.2 可编程逻辑器件的发展简介

在早期的数字逻辑设计中，设计人员在电路板或者面包板上把多个芯片连在一起构成系统。每个芯片包括一个或者多个逻辑门（如 NAND、AND、OR 或者非门），或者简单逻辑结构（如触发器和复用器等）。20 世纪 60 和 70 年代的很多设计都采用美国德州仪器公司的 7400 系列 TTL，即晶体管-晶体管逻辑器件。设计 TTL 时，其目的般是以尽量少的芯片来实现设计，以降低成本，减小电路板面积。而且，还需要尽量采用已有的器件来进行设计。

1.2.1 逻辑设计基本流程

在实现逻辑功能时，首先要建立真值表，如表 1.1 所示。真值表列出了逻辑所有可能

的输入及输入组合可能产生的相关输出。对于 n 输入，有 2^n 种可能的输入组合，这些都需要进行考虑。根据真值表，我们可以建立卡诺图，如图 1.3 所示。用卡诺图很容易建立简单的逻辑表达式。

表 1.1 真值表

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

		CD			
		00	01	11	10
AB	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1

图 1.3 根据真值表建立的卡诺图

根据卡诺图中含有 1 的每个框，结合每个框的公共输入，我们可以建立函数的一个“乘积和”式的逻辑表达式，如公式 1 所示。

$$Y = AB + CD + BD + BC + AD + AC \quad (\text{公式 1})$$

表达式由 6 个乘积项组成，每个乘积项对应一个与门。要在硬件中直接实现这一功能，需要 6 个 2 输入与门，1 个 6 输入或门，如果希望同步输出，还需要一个输出寄存器或者触发器。6 输入或门不支持 TTL，因此，需要级联更小的或门，这增加了延时和元件数量。为解决这些问题，TTL 设计人员使用与非逻辑重写公式，如公式 2 所示。

$$Y = \overline{AB \cdot CD \cdot BD \cdot BC \cdot AD \cdot AC} \quad (\text{公式 2})$$