

Verilog  
VHDL

FPGA / CPLD

# 边练边学

——快速入门Verilog/VHDL

吴厚航 编著



源程序下载地址：  
<http://www.buaapress.com.cn>的“下载中心”



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS

TP332.1/116

2013

# FPGA/CPLD 边练边学 ——快速入门 Verilog/VHDL

吴厚航 编著

北方工业大学图书馆



C00338817

RFID

北京航空航天大学出版社

## 内 容 简 介

本书面向广大的 FPGA/CPLD 初学者,从零开始讲述可编程逻辑器件(FPGA/CPLD)以及相关的基础知识,并以一个入门级的学习套件为实验平台,12 个应用实例贯穿其中,不仅有基本的 Verilog/VHDL 语法讲解,而且有设计思路和背景知识的详细描述;手把手地将开发工具(Quartus II+ModelSim)的使用图文并茂地展示给读者。书中例程源程序可在北京航空航天大学出版社网站的“下载中心”免费下载。

本书内容重基础,文字诙谐幽默,适合广大 FPGA/CPLD 的学习者作为入门之选,也可作为具有一定专业知识背景的电子工程师、电子信息类在校本科生、研究生等的参考资料。

### 图书在版编目(CIP)数据

FPGA / CPLD 边练边学:快速入门 Verilog / VHDL / 吴厚航编著. — 北京:北京航空航天大学出版社,2013.9  
ISBN 978-7-5124-1169-2

I. ①F… II. ①吴… III. ①可编程序逻辑器件—系统设计②VHDL 语言—程序设计 IV. ①TP332.1②TP312

中国版本图书馆 CIP 数据核字(2013)第 139418 号

版权所有,侵权必究。

### FPGA/CPLD 边练边学

#### ——快速入门 Verilog/VHDL

吴厚航 编著

责任编辑 王慕冰 龚荣桂 王平豪

\*

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:710×1 000 1/16 印张:17.25 字数:368 千字

2013 年 9 月第 1 版 2013 年 9 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-1169-2 定价:39.50 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

# 序 言

---

很高兴能为吴厚航(网络大名:特权同学)的《FPGA/CPLD 边练边学——快速入门 Verilog/VHDL》一书写序言。其实我一直是特权同学的粉丝,很欣赏他的风格、他的执着,诚恳和幽默诙谐的语言很能打动我。特权同学能在这么短的时间就把FPGA 给玩儿“转”了,可见他是着实下了一番苦功的。

这本书从讲故事开始,历数可编程器件的发展。即便你对FPGA/CPLD 一窍不通,在你看完这本书之后,也会对当今非常火爆的FPGA/CPLD 有一个全面的了解。哪怕把它当小说看,也会把一个“菜鸟”引领到神奇的数字世界里。

本书非常适合初学者,尤其是在大学里刚刚开始学数字逻辑课程的学生们。首先,在对可编程器件一无所知的情况下,这本书会给你一个清晰的概览。其次,只有走过一遍流程,才会对硬件描述语言 Verilog/VHDL 和 Quartus II 软件有一个概念上的了解,也因之特权同学才会给它起书名叫:边练边学。

我极力推荐这本书,也同时希望特权同学能再接再厉,不断地写出更优秀的作品。

陈卫中

Altera 公司大学计划中国区经理

2013 年 8 月于成都

# 前言

2010年6月和2011年10月,笔者(网名:特权同学)先后出版了图书《深入浅出玩转FPGA》和《爱上FPGA开发——特权和你一起学NIOS II》。这两本书都受到了广大读者的好评,前者在一年内两次重印且于2013年7月推出了第2版;后者也成为了当年Altera公司所举办的大学年会上的指定赠书。此外,笔者也在著名电子网站EDNChina和ChinaAET(《电子技术应用》杂志和电子发烧友网站)建立了专门的FPGA/CPLD助学小组,分享大量的学习资料,也提供了一些帮助初学者快速入门和进阶的开发套件。

## 1. 为什么写这本书

FPGA技术在国内的发展相对较晚,但FPGA强大的功能,其固有的灵活性和并行性使得很多应用场合非它不可。因此,越来越多的电子工程师和电子专业在校学生希望能够掌握这门技术。笔者经常收到读者或者开发套件使用者的邮件,所提问题五花八门,笔者虽然尽力答复大家,但着实有些难以招架。其中很大一部分初学者的问题都非常简单和基础,鉴于此,笔者在对这些不断重复的问题的回答过程中,也萌生了写一本基础的FPGA入门教程的想法。

## 2. 本书的侧重点和读者对象

由于笔者已经出版的两本书中,《深入浅出玩转FPGA(第2版)》重点在FPGA设计的经验和技巧的分享,《爱上FPGA开发——特权和你一起学NIOS II》重点在片上系统的入门开发,因此,你所见到的现在这本书,它的重点则是HDL语法的掌握、软件工具(主要是Quartus II+ModelSim)的使用以及整个开发流程的体验。可以说,它会是前面两本书的基础。

需要提醒读者注意的是,本书是基于一个CPLD学习套件来设计各种基本入门例程的。很多人会嗤之以鼻,这不是偷换概念吗?非也,FPGA和CPLD的区别笔者不想过多讨论,仁者见仁智者见智;笔者认为FPGA和CPLD在代码设计、工具使用和流程上几乎是一致的,虽然略有差别。但是对于初学者,你尽可以将CPLD当

作入门级的 FPGA 来用,这样有两个好处:一是降低学习成本(毕竟 FPGA 的价格较高);二是降低学习门槛(CPLD 无论从硬件设计还是内部逻辑上都要比 FPGA 简单得多)。可以肯定地说,如果你通过本书好好地掌握了 HDL 代码设计、工具使用和基本流程,那么再学 FPGA 将变得异常轻松。甚至使用其他的开发工具也会很快上手,因为它们之间都有很大的相通性,很多知识点都是换汤不换药的。

总之,如果你是一个 FPGA 或 CPLD 的初学者,没问题,选择此书,有条件再配套书中的 CPLD 套件进行实践,那么入门 FPGA/CPLD 会变得非常容易。

### 3. 本书的基本内容

本书共 7 章。第 1 章是基础中的基础,讲述可编程器件的发展历史并阐述相关的基本概念;第 2 章则对后续例程中将要使用到的 CPLD 开发平台进行介绍,深入详细地剖析了整个实验平台的硬件板级设计;第 3 章从最基础的 0 和 1 开始回顾数字电路的基础,也会深入探讨读者所关心的可编程器件的内部架构和原理;第 4 章是 HDL 语言基础,介绍使用最广泛的 Verilog 和 VHDL 语言的基本语法;第 5 章中手把手教大家建立 PC 机上的开发环境,并且图文并茂地让大家体验一个完整的工程创建流程;第 6 章通过 10 个基本实例来掌握 HDL 语言,同时也熟悉工具的使用(Quartus II 和 ModelSim);第 7 章则是 CPLD 内部特有资源的应用案例。本书有对基础理论知识专门的讲解,也有对实践的演练和讲解,更多的是在实践中传递实用的设计技巧和方法,非常适合初学者。

在此要感谢广大的 FPGA/CPLD 爱好者们,你们的学习热情是我编写此书的源动力。非常感谢为书前的学习套件前期设计和推广做出过努力的陈卫东、余国峰和黄娜。感谢就职于骏龙科技的“师弟”张亚峰一直以来给予的技术支持。另外,也要特别感谢著名电子网站 EDNChina 的王志华和庞玉婷,ChinaAET 的杨晖、曾超宇、乐卫平、陈颖莹和贾志梅,电子发烧友的高美笛和钱姗姗,你们为广大网友们提供了一个非常友好的交流互动平台。

非常荣幸能够邀请到 Altera 公司的大学计划中国区经理陈卫中先生作序,在此表示由衷的感谢。最后,要向我的家人致敬,你们是我前进路上的最坚强后盾。尤其要感谢我的妻子一直以来所给予我的鼓励和支持。

吴厚航

2013 年 7 月于上海

02	.....	基于新的组合逻辑	5.1
13	.....	Verilog 语言	5.2.1
30	.....	VHDL 语言	5.2.2
60	.....	代码风格与代码规范	5.3
61	.....	代码风格	5.3.1
62	.....	代码风格	5.3.2
18	.....	本章小结	5.4
31	.....	实验平台开发	2.1
18	.....	软件平台开发	2.1.1
38	.....	硬件平台开发	2.1.2
88	.....	本章小结	2.2
38	.....	可编程器件发展简史与基本概念	1
38	.....	1.1 可编程器件的由来与发展	1
38	.....	1.2 设计方式与工具链	3
38	.....	1.3 应用领域和发展趋势	5
38	.....	第 2 章 实验平台板级设计	8
38	.....	2.1 FPGA/CPLD 板级电路设计五要素	8
38	.....	2.1.1 能量供应——电源电路	9
38	.....	2.1.2 心脏跳动——时钟电路	11
38	.....	2.1.3 状态初始——复位电路	12
38	.....	2.1.4 灵活定制——配置电路	13
38	.....	2.1.5 自由扩展——外设电路(I/O 应用)	14
38	.....	2.2 CPLD 实验板 DIY	14
38	.....	2.2.1 读懂器件手册	14
38	.....	2.2.2 CPLD 核心电路设计	19
38	.....	2.2.3 外设扩展电路设计	25
38	.....	2.2.4 I/O 引脚分配	30
38	.....	第 3 章 数字电路基础	36
38	.....	3.1 0 和 1——精彩世界由此开始	36
38	.....	3.2 表面现象揭秘——逻辑关系	39
38	.....	3.3 内里本质探索——器件结构	43
38	.....	第 4 章 Verilog 与 VHDL 语法基础	48
38	.....	4.1 语法学习的经验之谈	48

4.2	可综合的语法子集	50
4.2.1	可综合的 Verilog 语法	51
4.2.2	可综合的 VHDL 语法	56
4.3	代码风格与书写规范	60
4.3.1	代码书写规范	61
4.3.2	代码风格	63
<b>第 5 章</b>	<b>第一个完整的工程实践案例</b>	<b>81</b>
5.1	软件开发平台搭建	81
5.1.1	软件下载和 License 申请	81
5.1.2	Quartus II 的安装	85
5.1.3	ModelSim 的安装	88
5.2	基本开发流程概述	90
5.3	第一个工程实例	91
5.3.1	工程创建与设计输入	92
5.3.2	行为仿真	97
5.3.3	引脚分配与编译	104
5.3.4	门级仿真	106
5.3.5	板级调试	109
<b>第 6 章</b>	<b>基础实验与拓展练习</b>	<b>112</b>
6.1	基于时钟分频的 PWM 发生器	112
6.1.1	实验原理分析	112
6.1.2	Verilog 参考实例	113
6.1.3	VHDL 参考实例	114
6.1.4	仿真验证与板级调试	115
6.1.5	实验流程与注意事项	117
6.1.6	拓展练习	120
6.2	经典的按键消抖实例	121
6.2.1	实验原理分析	121
6.2.2	Verilog 参考实例	123
6.2.3	VHDL 参考实例	124
6.2.4	仿真验证与板级调试	126
6.2.5	实验流程与注意事项	130
6.2.6	拓展练习	131
6.3	基于 Johnson 计数器的流水灯实验	131



6.3.1	实验原理分析 .....	131
6.3.2	Verilog 参考实例 .....	132
6.3.3	VHDL 参考实例 .....	135
6.3.4	仿真验证 .....	138
6.3.5	实验流程与注意事项 .....	143
6.3.6	拓展练习 .....	143
6.4	数码管驱动显示实验 .....	144
6.4.1	实验原理分析 .....	144
6.4.2	Verilog 参考实例 .....	146
6.4.3	VHDL 参考实例 .....	148
6.4.4	仿真验证 .....	149
6.4.5	实验流程与注意事项 .....	152
6.4.6	拓展练习 .....	154
6.5	乘法器设计实验 .....	155
6.5.1	实验原理分析 .....	155
6.5.2	Verilog 参考实例 .....	156
6.5.3	VHDL 参考实例 .....	157
6.5.4	仿真验证 .....	158
6.5.5	实验流程与注意事项 .....	161
6.5.6	拓展练习 .....	161
6.6	VGA 显示驱动实验 .....	161
6.6.1	实验原理分析 .....	161
6.6.2	Verilog 参考实例 .....	163
6.6.3	VHDL 参考实例 .....	165
6.6.4	仿真验证 .....	168
6.6.5	实验流程与注意事项 .....	169
6.6.6	拓展练习 .....	170
6.7	UART 串口收发实验 .....	170
6.7.1	实验原理分析 .....	170
6.7.2	Verilog 参考实例 .....	171
6.7.3	VHDL 参考实例 .....	180
6.7.4	仿真验证 .....	188
6.7.5	实验流程与注意事项 .....	190
6.7.6	拓展练习 .....	191
6.8	PS/2 键盘解码实验 .....	192
6.8.1	实验原理分析 .....	192

# 目 录

181	6.8.2 Verilog 参考实例	194
181	6.8.3 VHDL 参考实例	198
181	6.8.4 仿真实验	203
181	6.8.5 实验流程与注意事项	207
181	6.8.6 拓展练习	207
181	6.9 基于 I <sup>2</sup> C 通信的 EEPROM 读/写实验	208
181	6.9.1 实验原理分析	208
181	6.9.2 Verilog 参考实例	210
181	6.9.3 VHDL 参考实例	222
181	6.9.4 仿真实验	235
181	6.9.5 实验流程与注意事项	241
181	6.9.6 拓展练习	241
181	6.10 SRAM 读/写测试实验	242
181	6.10.1 实验原理分析	242
181	6.10.2 Verilog 参考实例	244
181	6.10.3 VHDL 参考实例	247
181	6.10.4 仿真实验	251
181	6.10.5 实验流程与注意事项	253
181	6.10.6 拓展练习	254
	<b>第 7 章 器件资源应用实例</b>	<b>255</b>
181	7.1 MAX II 内部振荡时钟使用实例	255
181	7.2 MAX II 的 UFM 模块使用实例	258
181	<b>参考文献</b>	<b>263</b>

# 第1章

## 可编程器件发展简史与基本概念

### 1.1 可编程器件的由来与发展

20世纪60年代中期, TI公司设计制造了各式各样的实现基本逻辑门电路功能的芯片, 相信今天很多的工程师仍然很熟悉这些主要面对军工应用的54xx和商业应用的74xx芯片。据说早期的工程师甚至能够单凭着这些芯片就可架构出一颗简单CPU的功能。还真别小瞧这些基本逻辑门电路, 俗话说“万丈高楼平地起”, 如果说今天在嵌入式领域呼风唤雨的各种功能强大的ARM7、ARM9、DSP是万丈高楼, 那么称这些基本的逻辑门电路为一砖一瓦倒是一点也不为过。

从1971年Intel公司的第一颗4位微处理器Intel 4004到20世纪80年代初被奉为经典的8051单片机, 再到今天各大嵌入式处理器厂商竞相使用的由ARM公司推出的各种Cortex内核, 嵌入式处理器的发展不可不说是翻天覆地。不过, 如果深入处理器的底层结构, 会发现它们最本质的东西并没有太大的改变。而处理器再强大, 一颗芯片尽可以将各种外设嵌入其中, 但对于任何一颗已经批量生产的芯片而言, 它的功能是固定的, 若想在既有外设功能的基础上有任何的扩展, 或许不是遇到电气特性不支持就是遇到I/O太少的尴尬, 而这些问题也就催生了可编程逻辑器件的诞生。今天的CPU周围已很难看见54或74字样的ASIC了, 取而代之的可能是引脚密集的可编程逻辑器件, 因为在系统的可扩展性和灵活性方面, FPGA/CPLD有着得天独厚的优势。当然, 今天动辄上百万门的FPGA器件可不是为干这点活而制造的, 它更多地被应用到了高速通信、数据采集、网络传输等对数据速率和吞吐量有更高要求的场合。

今天大家熟知的FPGA/CPLD也不是一开始就有的, 第一款可编程逻辑器件(PLD)最初是在1970年以PROM的形式进入人们的视野的, 这种PROM结构的可编程逻辑器件可以实现简单的逻辑功能, 很容易替代当时流行的54或74系列逻辑门电路。

受限于PROM的结构, 第一款可编程逻辑器件输入相对较少。因此, 可编程逻辑阵列(PAL)便孕育而生, PAL由一个可编程的“与”平面和一个固定的“或”平面构成, “或”门的输出可以通过触发器有选择地被置为寄存状态。PAL器件是现场可编

程的,它的实现工艺有反熔丝技术、EPROM技术和EEPROM技术。PAL的问题在于其实现方式使得信号通过可编程连线的相对时间较长。在PAL的基础上,又发展了一种通用阵列逻辑GAL,要比PAL速度快许多。它采用了EEPROM工艺,实现了电可擦除、电可改写,其输出结构是可编程的逻辑宏单元,因而其设计具有很强的灵活性,至今仍被许多人使用。

这些早期的PLD器件的一个共同特点是可以实现速度特性较好的逻辑功能,但其过于简单的结构也使它们只能实现规模较小的电路。电子领域的发展趋势总是朝着速度更快、功能更强、体积更小、成本更廉价的方向迈进。复杂可编程逻辑器件(CPLD)的诞生也就顺理成章了。Altera公司于1984年发明了基于CMOS和EPROM技术相结合的CPLD。CPLD可实现的逻辑功能相比PAL和GAL有了大幅度的提升,已经可以胜任设计中复杂性较高、速度也较快的逻辑功能,尤其在接口转换、总线控制和扩展方面有着较多的应用。经过几十年的发展,今天的CPLD功能和性能也得到了进一步的提升,其基本结构由可编程I/O单元、基本逻辑单元、布线池以及其他相关辅助功能块组成。Altera、Xilinx和Lattice公司是主要的CPLD供应商。

其实无论是前面提到的PAL、GAL,还是CPLD,要实现大规模的复杂逻辑电路都显得无能为力。而ASIC的设计既耗时又费钱,而且功能固定,在流片后很难随意更改。鉴于此,Xilinx公司的创始人之一Ross Freeman发明了现场可编程门阵列(FPGA),FPGA是一块全部由“开放式门”组成的计算机芯片。采用该芯片,工程师可以根据需要进行灵活编程,添加各种新功能,以满足不断发展的协议标准或规范,工程师们甚至可以在设计的最后阶段对它进行修改和升级。Freeman先生当时就推测低成本、高灵活性的FPGA将成为各种应用中定制芯片的替代品。也正是由于此项伟大的发明,让Freeman先生于2009年荣登美国发明家名人堂。

从发明FPGA至今,伴随着制造工艺的不断进步,FPGA在深亚微米甚至深亚纳米时代一直走在了创新第一线。如今的FPGA器件,其组成不仅限于基本的可编程I/O单元、可编程逻辑单元、丰富的布线资源,而且还拥有灵活的时钟管理单元、嵌入式块RAM以及各种通用的内嵌功能单元,很多器件还顺应市场需求内嵌专用的硬件模块。近些年来,可编程器件的龙头老大Xilinx和Altera公司更是相继推出了硬核CPU+FPGA的产品,此举大有单芯片横扫千军的架势。

电子行业在继续挑战摩尔定律的征程中,无论是可编程器件继续大放光彩,还是ASIC能够重获新生,可编程器件,尤其是FPGA/CPLD的发明和大量应用已经足够让我们竖然起敬。相信对于很多即将或者已经走上电子硬件设计的同仁们,对可编程器件的了解、熟悉甚至精通将会成为自身技术能力提升的一项重要指标。

## 1.2 设计方式与工具链

FPGA/CPLD 的设计方式和大多数的嵌入式微处理器类似,首先要有一台电脑,其次要搭建一块有 FPGA/CPLD 的电路板并且预留出供代码烧录的调试接口。通俗理解 FPGA/CPLD 的开发,也就是在 PC 机的各种工具上做好设计,最终产生一个 bit 流(就是一个充满 0 和 1 的只有我们的 FPGA/CPLD 器件才能够读懂的“机器语言”),通过下载线把这个 bit 流再送给 FPGA/CPLD 运行。说着简单,其实也着实不易,首先是 FPGA/CPLD 的设计输入有自己专门的语言(HDL,即硬件描述语言),目前应用最广泛的是 VHDL 和 Verilog。无论是 VHDL 还是 Verilog,尤其 Verilog 语法和 C 很相似,但是整个思想上完全是对过往软件语言的一种颠覆。软件编程是顺序思想,即便是操作系统根本上也还是顺序执行的,而 HDL 完全不同,在 FPGA/CPLD 器件中实现的功能是并行的(当然,也可以让它顺序地干活,毕竟软件都是在硬件基础上实现的,单纯软件能干的活,硬件没有干不了的,因为本来就是硬件在干;而硬件能干的活,软件就不见得能胜任了)。

再来看看 FPGA/CPLD 一般的开发流程。如图 1.1 所示,这也许算不上一个完全意义上完整的开发流程,但是这些步骤绝对都是 FPGA 开发流程中不可缺少的部分(行为仿真和时序仿真有时只是作为可选步骤),其他的设计方法、技巧都是贯穿于其中的。

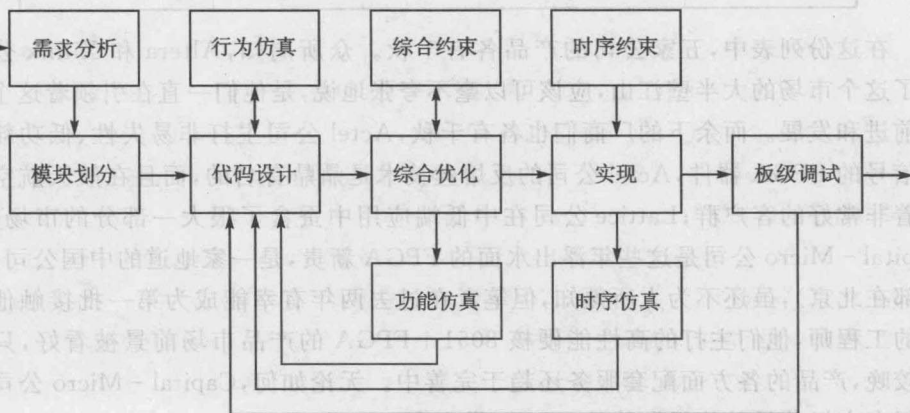


图 1.1 FPGA/CPLD 开发流程

此外, FPGA 设计的最大特点就是迭代性很强,并不是一个简单的顺序流程。在开发设计过程中,设计者在测试验证中一旦发现问题,往往需要回到前面的步骤重新审查、修改,然后重新综合、实现、仿真验证,直到最终的设计符合需求。

看到这些错综复杂的框图和连线,希望不会吓到曾经热情高涨的初学者们。路是一步一步走出来的,只要用心,通过本书后面各个章节的学习,相信入门绝不是难事。

接下来,我们会讨论一些关于 FPGA 工具链的话题。这里所谓的工具,指的是针对 FPGA 开发流程所需要涉及的各种 EDA 工具。这部分知识只是作为一个参考,希望有志于从事可编程器件相关工作的同仁们对本行业内的生态链也有所了解。下文所列出的很多供应商以及他们提供的 EDA 工具很有可能随着时代的变迁逐渐淡出历史舞台——消失在人们的视野中;也很可能在读者阅读本书时,更多新的 EDA 工具和厂商在不断地涌现。这些都很正常,因为从已往几十年 EDA 工具的更迭来看,能够长期稳定地在这份榜单中占有一席之地真的不是一件容易的事。

首先,我们需要罗列从事和制造可编程逻辑器件(这里主要针对 FPGA/CPLD 而言)的供应商,这些供应商通常都会有针对自己器件的比较完备的集成 EDA 开发工具。如表 1.1 所列,这些应该是目前大家比较熟悉并且网络上搜索得到的 FPGA/CPLD 供应商,至于那些网络上都很难寻觅到的供应商(这样的 FPGA/CPLD 供应商还是有不少的),笔者也实在无法一一罗列。

表 1.1 FPGA/CPLD 主要供应商

公司	网址	产品	工具
Actel	www.actel.com	FPGA	Libero IDE
Altera	www.altera.com	FPGA/CPLD	Quartus II
Capital - Micro	www.capital-micro.com	FPGA	Primace
Lattice	www.latticesemi.com	FPGA/CPLD	Diamond
Xilinx	www.xilinx.com	FPGA/CPLD	ISE

在这份列表中,五家公司的产品各有千秋。众所周知,Altera 和 Xilinx 公司占据了这个市场的大半壁江山,应该可以毫不夸张地说,是他们一直在引领着这个行业的前进和发展。而余下的厂商们也各有千秋,Actel 公司主打非易失性、低功耗及混合信号的 FPGA 器件,Actel 公司的反熔丝技术是鼎鼎有名的,而且在航天航空领域有着非常好的客户群;Lattice 公司在中低端应用中蚕食了很大一部分的市场份额;Capital - Micro 公司是这些年浮出水面的 FPGA 新贵,是一家地道的中国公司(研发总部在北京),虽还不为人所熟知,但笔者在过去两年有幸能成为第一批接触他们产品的工程师,他们主打的高性能硬核 8051 + FPGA 的产品市场前景被看好,只是起步较晚,产品的各方面配套服务还趋于完善中。无论如何,Capital - Micro 公司是一家很有市场前景并值得期待的 FPGA 供应商。

如图 1.2 所示,我们可以看到 FPGA/CPLD 开发过程中涉及多个步骤并会有不同的网表生成,这些不同的网表使得使用官方既有开发工具的过程中也能够支持第三方 EDA 工具的使用。综合、布局布线、仿真、时序设计,虽然这几个步骤在一般的开发过程中都能够使用器件原厂提供的集成开发工具完成,但是也可以交给第三方 EDA 工具来完成。一般而言,尤其是综合、仿真和时序设计,第三方的一些工具有更强的竞争力;而布局布线这方面则是器件原厂来得更优化一些。无论如何,用户可以

根据项目的实际需要或者自身的开发习惯,选择不同的工具帮助完成设计。

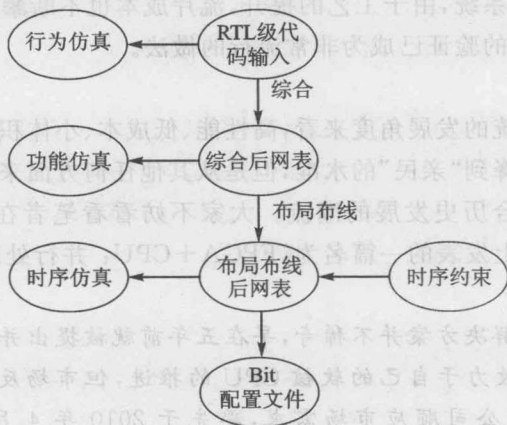


图 1.2 FPGA/CPLD 网表生成

表 1.2 罗列出了目前比较常见的 FPGA/CPLD 开发使用的第三方综合、仿真工具。

表 1.2 FPGA/CPLD 主要第三方 EDA 工具

公司	网址	产品	功能
Aldec	www.aldec.com	Active-HDL 等	仿真工具
Model Tech	www.model.com	ModelSim	仿真工具
Synopsys	www.synopsys.com	VCS	仿真工具
Synopsys	www.synopsys.com	Synplify Pro 等	综合工具
Mentor	www.mentor.com	LeonardoSpectrum	综合工具

### 1.3 应用领域和发展趋势

近些年来,在嵌入式领域被大多数工程师所认可的技术中,ARM、DSP、FPGA 可谓三足鼎立。而 FPGA/CPLD 目前虽然还受制于较高的学习门槛以及器件本身高昂的价格,从应用的普及率上来看和 ARM、DSP 还是有一定差距,但是在非常多的应用场合,工程师们还是要别无选择地使用 FPGA/CPLD。FPGA/CPLD 因其某些固有的特性是其他芯片所不具备的,所以它的应用领域涵盖得很广。从技术角度来看,主要有以下需求的应用场合:

- 逻辑粘合,如简单的地址或外设扩展,CPLD 器件尤其适合。
- 实时控制,如液晶屏或电机等设备的驱动控制。
- 高速信号采集,如高速 A/D 前端或图像的预处理。

- 协议实现,如更新较快的各种通信标准,使用 FPGA 比 ASIC 更有竞争力。
- 各种仿真验证系统,由于工艺的提升,流片成本也不断攀升,而在流片前使用 FPGA 做前期的验证已成为非常流行的做法。
- 片上系统。

从整个嵌入式系统的发展角度来看,高性能、低成本、小体积是大势所趋,而片上系统虽然成本还无法降到“亲民”的水准,但是从其他任何方面来看,FPGA 上所实现的片上系统越来越迎合历史发展的潮流。大家不妨看看笔者在 2011 年 11 月刊的《电子系统设计》杂志上发表的一篇名为《FPGA+CPU:并行处理大行其道》文章中的一些论述。

FPGA+CPU 的解决方案并不稀奇,早在五年前就被提出并付诸实践,Xilinx 和 Altera 公司也一直在致力于自己的软核 CPU 的推进,但市场反应显然没有达到预期。穷则思变,Xilinx 公司顺应市场需求,率先于 2010 年 4 月发布了集成 ARM Cortex-A9 CPU 和 28 nm FPGA 的可扩展式处理平台(Extensible Processing Platform)架构。时隔不到一年,可扩展处理平台 Zynq-7000 系列又被搬上了前台,Xilinx 公司的良苦用心可见一斑。Altera 公司也不示弱,英特尔在 2010 年秋季发布的凌动 E600C 可配置处理器中就集成了 Altera 公司的 FPGA,并且 Altera 公司即将推出的同样集成 Cortex-A9 CPU 的 SOC FPGA 明显是要与 Xilinx 公司唱对台戏。

厂商的明争暗斗不是咱们这些芯片级的小喽啰们真正关心和在意的,我们更多的是需要去探讨和思索这种新的开发平台是否真的满足我们的客户日益增长的“物质文化”需求。我们也不禁会问:FPGA+CPU 的集成架构到底是顺应了历史发展的趋势,还是仅仅昙花一现、转瞬即逝?

如图 1.3 所示,一个比较简化的传统嵌入式系统如左图,单片集成了 CPU 的 FPGA 架构则如右图。单从硬件架构层面来看,好像没有太大的花头,仅仅只是二合一而已。但是真正做过系统开发的工程师都知道,这个二合一所带来的不仅仅是 Bom 和 Layout 的简化,更多的利好是我们肉眼看不到的软硬件底层衔接的优化和无形之中的灵活性以及潜在的性能提升。

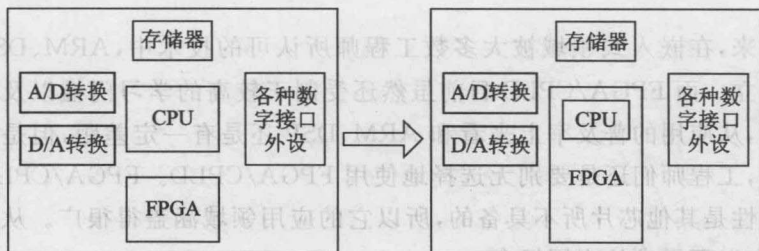


图 1.3 CPU 和 FPGA 器件从分立到集成的转换

这里可以罗列出来基于 FPGA 的 CPU 集成带来的一些潜在的优势:



- 更易于满足大多数系统的功能性需求。
- 潜在地改善了系统的性能。
- 在某些应用中的灵活性和可升级性大大提高。
- 处理器到外设的接口能够得到优化。
- 软硬件互联的接口性能获得极大的提升。
- 有利于设计的重用和新设计的快速成型。
- 简化单芯片甚至整板的 PCB 布局布线。

FPGA+CPU 的单片集成相较于传统应用的优势可见一斑,但从另一个角度看,正如 CPU 从单核到多核演进在延续着摩尔定律的“魔咒”一样,FPGA+CPU 的强势出击更像是并行处理在嵌入式应用的大行其道。

### 思考

1. FPGA 和 CPLD 到底有何异同呢?
2. 使用 FPGA/CPLD 器件需要具备怎样的一个软件开发环境?

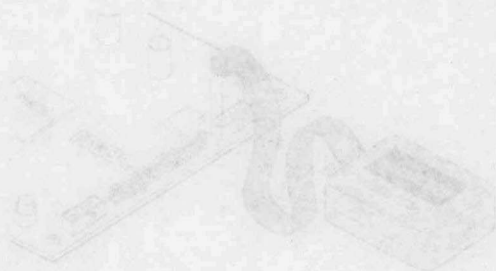


图 1.1 不可编程器件

图 1.1 所示,对于任何一个 FPGA/CPLD 开发而言,一条下装线(或称 FPGA/CPLD 器件的电路板)是承载 FPGA/CPLD 器件的作用。最基本的硬件准备,电路板的作用不言而喻。我们设计的目的便是希望通过对板载的 FPGA/CPLD 器件编程,以控制电路板上的其他外设按照我们规定的功能运转;而下装线起到连接 PC 机和电路板的作用,它协助

开发人员已经在 PC 机的软件工具上预先设计好的功能配置数据流固化到电路板上的 FPGA/CPLD 器件中(可用于前期的开发、调试以及产品化开发)。板载的 FPGA/CPLD 器件的电路板,那么这板电路板如何设计,当然需要一块板载 FPGA/CPLD 器件的电路板,那么这板电路板如何设计,如图 1.2 和图 1.3 所示,分别对 Xilinx 公司的 Nexys3 FPGA 开发板和 Altera