

可编程器件 EDA应用开发技术

MODULE CTRL

"Inputs

SP,CLK pin;

"Output

EN pin istype 'com';

sta, stb node istype 'reg';

st=[sta, stb];

set0="B00; set1="B01; set2="B10; set3="B11;

EQUATIONS

st.clk=CLK;

STATE_DIAGRAM st

state set0:EN=0;

if(SP==1)then set1 else set0;

state set1:EN=1;

if(SP==1)then set1 else set2;

state set2:EN=1;

if(SP==0)then set2 else set3;

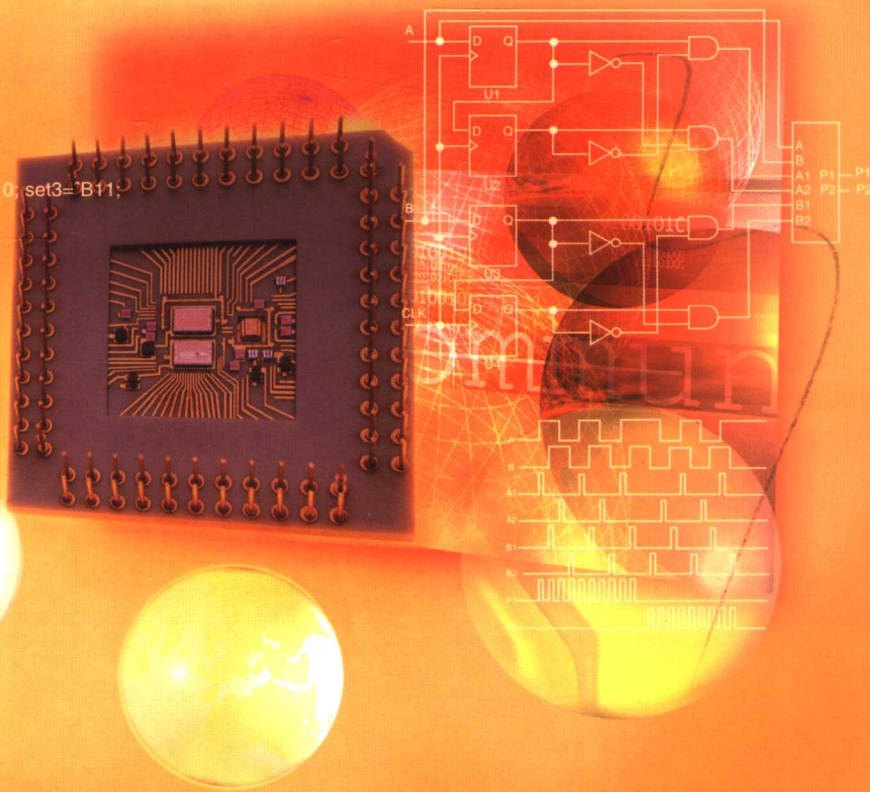
state set3:EN=0;

if(SP==0)then set0 else set3;

END

陈燕东 主编

李颖 刘胜平 李红 等编著



国防工业出版社

National Defense Industry Press

内 容 简 介

本书从电子电路系统设计的角度出发,结合可编程器件技术的最新发展现状,系统地介绍了可编程器件及其应用开发技术。内容主要包括:可编程器件与电子设计自动化、Lattice 公司的在系统可编程逻辑器件、VHDL 硬件描述语言、ABEL-HDL 硬件描述语言、ispLEVER 开发工具、数字电路系统设计、Lattice 公司的在系统可编程模拟器件、PAC-Designer 开发工具和模拟电路系统设计等。

本书可作为高等院校电气工程、自动控制等相关专业的教学用书和参考书,也可作为电子、通信和自动控制等领域中从事电子产品设计与开发工作的工程技术人员的技术参考书。

图书在版编目(CIP)数据

可编程器件 EDA 应用开发技术/陈燕东主编;李颖等
编著. —北京:国防工业出版社,2006. 5
ISBN 7-118-04467-9

I. 可... II. ①陈...②李... III. ①可编程逻辑器
件②电子电路—电路设计:计算机辅助设计
IV. ①TP332.1②TN702

中国版本图书馆 CIP 数据核字(2006)第 021608 号

※

国防工业出版社 出版发行
(北京市海淀区紫竹院南路 23 号 邮政编码 100044)
涿中印刷厂印刷
新华书店经售

*

开本 787×1092 1/16 印张 21¼ 字数 489 千字
2006 年 5 月第 1 版第 1 次印刷 印数 1—4000 册 定价 33.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422
发行传真:(010)68411535

发行邮购:(010)68414474
发行业务:(010)68472764

前 言

科学技术的发展给现代电路与系统的设计带来了新的变革。采用先进的设计方法与设计手段以及最新的元器件来进行电子自动化设计,已成为一个新的潮流。1992年,美国 Lattice 公司首推的在系统可编程逻辑器件,彻底改变了传统数字电子系统的设计和实现方法,开创了数字系统设计的里程碑。1999年,Lattice 公司又推出了在系统可编程模拟器件,翻开了模拟电路设计方法的新篇章,为电子设计自动化技术的应用开拓了更广阔的前景。

目前,国内各高校都引入了在系统可编程技术的教学,但由于新型器件层出不穷,器件本身的结构也有所变化,因而教学也有必要跟上新技术的发展,另外,教学中比较多地局限在数字可编程器件上,忽略了模拟可编程器件的发展与应用。从世界范围看,模拟可编程器件的发展、开发与应用并没有停顿,仍然在不断进步。本书以美国 Lattice 公司的可编程逻辑器件和模拟器件为例,详细介绍了可编程技术的发展现状、各器件原理、ispLEVER 和 PAC-Designer 开发工具以及应用实例等,给读者呈现出完整的在系统可编程技术。

全书共分 9 章,各章主要内容如下:

- 第 1 章:可编程器件与电子设计自动化
- 第 2 章:Lattice 公司的在系统可编程逻辑器件
- 第 3 章:VHDL 硬件描述语言
- 第 4 章:ABEL-HDL 硬件描述语言
- 第 5 章:ispLEVER 开发工具
- 第 6 章:数字电路系统设计
- 第 7 章:Lattice 公司的在系统可编程模拟器件
- 第 8 章:PAC-Designer 开发工具
- 第 9 章:模拟电路系统设计

本书可作为高等院校电气工程、自动控制等相关专业的教学用书和参考书,也可作为电子、通信和自动控制等领域中从事电子产品设计与开发工作的工程技术人员的技术参考书。

本书由陈燕东主编,参加本书编写工作的有陈燕东、李颖、刘胜平、李红、刘景琳、邓湘凤、张易明、叶南海、王强、桂华、张攀峰、李林、钱相辉等。

由于作者知识水平所限,书中难免有错漏和不足之处,恳请各位读者批评指正。

作者

(E-mail:xlcyd520@163.com)

2005 年 12 月

目 录

第 1 章 可编程器件与电子设计

自动化	1
1.1 可编程器件的发展过程	1
1.1.1 集成电路的发展历史	1
1.1.2 可编程器件的历史和现状	1
1.2 EDA 技术简介	2
1.2.1 EDA 设计的典型流程	2
1.2.2 EDA 技术的设计特点	3
1.3 可编程器件概述	4
1.3.1 可编程逻辑器件的分类	4
1.3.2 可编程逻辑器件的结构	5
1.3.3 CPLD 与 FPGA	7
1.4 先进的编程和测试技术	8
1.4.1 在系统可编程(ISP)技术	8
1.4.2 边界扫描测试技术(BST)	9
1.5 可编程器件主要生产厂商及其器件	9
1.5.1 Altera 公司的可编程器件产品	9
1.5.2 Xilinx 公司的可编程器件产品	9
1.5.3 Lattice 公司的可编程器件产品	9

第 2 章 Lattice 公司的在系统可编程逻辑器件

2.1 在系统可编程逻辑器件概述	11
2.1.1 在系统可编程 SPLD 系列	12

2.1.2 在系统可编程 CPLD 器件	13
2.1.3 在系统可编程 FPGA 器件	14
2.1.4 可编程数字开关及互连器件	17
2.1.5 ISP-PLD 的主流产品	18
2.2 ispLSI 器件及其结构原理	19
2.2.1 器件概述与技术特性	20
2.2.2 ispLSI 器件的结构原理	21
2.2.3 ispLSI1000 系列	27
2.2.4 ispLSI5000 系列	30
2.2.5 ispLSI8000 系列	30
2.3 ispLSI 器件在系统编程	32
2.3.1 ispLSI 器件的编程接口	32
2.3.2 ispLSI 器件的编程结构	34
2.3.3 编程状态机	34
2.3.4 编程连接方式	35
2.4 ispMACH4000 系列器件	37
2.4.1 器件概述	37
2.4.2 ispMACH4000 体系结构	39
2.4.3 ispMACH 4000Z 零功耗 CPLD 器件	43
2.5 LatticeECP/EC 系列器件	43
2.5.1 器件概述	44
2.5.2 器件结构	45
2.5.3 核心模块 PFU 和 PFF	46
2.5.4 时钟分布网络	48
2.5.5 系统存储器(EBR)及其配置	50

2.5.6	可编程 I/O 单元(PIC) ...	50	3.5.1	逻辑(LOGICAL)	
2.5.7	LatticeECP 的sysDSP 块 ...	56		运算符	101
2.6	LatticeXP 系列器件	62	3.5.2	算术(ARITHMETIC)	
2.6.1	器件概述	62		运算符	101
2.6.2	器件的主要特性	63	3.5.3	关系(RELATIONAL)	
2.6.3	器件结构及其系统配置 ...	63		运算符	102
2.7	MachXO 系列	65	3.5.4	并置运算符	103
2.7.1	MachXO 系列及其主要 特征	65	3.5.5	运算操作符的优先级 ...	104
2.7.2	器件结构及其 I/O 组	66	3.6	并行语句	104
2.7.3	MachXO 与 LatticsXP 器件比较	69	3.6.1	PROCESS 进程语句	104
2.8	ispGDS 和 ispGDX 系列 器件	72	3.6.2	Block 块语句	105
2.8.1	ispGDS 通用数字开关 器件	72	3.6.3	SUBPROGRAM 子 程序	105
2.8.2	ispGDX 通用数字交叉 阵列	74	3.6.4	ASSERT 断言语句	106
2.8.3	ispGDX 系列器件应用 ...	76	3.6.5	SIGNAL ASSIGNMENT 并行信号赋值语句	106
第 3 章	VHDL 硬件描述语言	78	3.6.6	GENERIC 参数数据传递 语句	107
3.1	硬件描述语言概述	78	3.6.7	COMPONENT 通用模块 与元件调用语句	107
3.2	VHDL 设计单元	78	3.6.8	GENERATE 生成 语句	107
3.2.1	VHDL 的基本语素	78	3.7	顺序语句	108
3.2.2	基本结构	81	3.7.1	变量赋值语句	108
3.2.3	实体	82	3.7.2	IF 条件语句	108
3.2.4	结构体	84	3.7.3	CASE 条件语句	110
3.2.5	库、包与配置	86	3.7.4	NULL 空操作语句	110
3.3	VHDL 数据对象	92	3.7.5	LOOP 循环语句	111
3.3.1	常量	92	3.7.6	WAIT 等待语句	112
3.3.2	变量	92	3.7.7	RETURN 返回语句	113
3.3.3	信号	92	第 4 章	ABEL-HDL 硬件描述语言	114
3.4	VHDL 基本数据类型	93	4.1	ABEL-HDL 的基本元素与 语法规则	114
3.4.1	数据类型的分类	93	4.1.1	ABEL-HDL 语言元素	114
3.4.2	标准数据类型	94	4.1.2	基本语法规则	120
3.4.3	自定义数据类型	96			
3.4.4	数据类型的转换方法	99			
3.5	VHDL 运算符	101			

4.2	ABEL-HDL 源文件结构	120	输入	154
4.2.1	ABEL 源文件基本格式	120	5.4.1	建立顶层原理图
4.2.2	ABEL 源文件结构层次	120	5.4.2	建立底层 ABEL-HDL 源文件
4.3	ABEL-HDL 语言的语句	122	5.4.3	编译 ABEL 源文件
4.3.1	文件头部	122	5.4.4	仿真测试
4.3.2	定义段(DECLARATIONS)	123	5.4.5	器件适配
4.3.3	逻辑描述段	125	5.4.6	层次化设计方法
4.3.4	测试矢量段(TEST_VECTORS)	129	5.5	约束条件编辑器
4.3.5	结束段	129	5.6	ISP 器件的编程实现
4.4	指示字(DIRECTIVES)	130	5.7	VHDL 和 Verilog 语言的设计方法
4.5	ABEL-HDL 输出文件结构	133	5.7.1	VHDL 语言的输入
4.6	ABEL-HDL 语言应用举例	133	5.7.2	Verilog 语言的输入
第 5 章	ispLEVER 开发工具	137	5.8	仿真工具 ModelSim 的使用
5.1	ISP 器件开发概述	137	5.9	ispLEVER 的 FPGA 设计
5.1.1	ISP 器件的开发过程	137	5.9.1	项目创建
5.1.2	开发工具软件简介	138	5.9.2	源文件设计输入
5.1.3	ispLEVER 软件及其主要特征	140	5.9.3	编译与仿真
5.2	原理图的输入	141	5.9.4	设计实现
5.2.1	项目创建与器件选择	141	5.10	ispDesign EXPERT 开发工具
5.2.2	原理图源文件的添加	144	5.10.1	概述
5.2.3	编辑原理图	144	5.10.2	器件编程实现
5.2.4	连线命名与标注	145	5.10.3	MACH 器件下载
5.2.5	引脚属性定义 (Attributes)	146	第 6 章	数字电路系统设计
5.2.6	建立元件符号	147	6.1	数字电路系统设计概述
5.3	编译与仿真	147	6.1.1	设计流程
5.3.1	建立仿真和设计编译	147	6.1.2	设计方法
5.3.2	功能仿真	149	6.1.3	设计准则
5.3.3	时序仿真	154	6.2	组合逻辑电路设计
5.4	ABEL 语言与原理图的混合		6.2.1	格雷码/二进制码变换器
			6.2.2	4 位全加器设计
			6.2.3	求补器
			6.2.4	乘法器的设计

6.2.5	编码器设计	213	7.4.3	ispPAC30 的工作原理	267
6.2.6	译码器设计	215	7.4.4	SPI 接口编程模式	269
6.2.7	多路选择器设计	217	7.5	ispPAC80 与 ispPAC81	272
6.2.8	总线缓冲器	218	7.5.1	器件功能结构	273
6.3	时序逻辑电路设计	220	7.5.2	ispPAC80 性能指标	274
6.3.1	循环移位寄存器	220	7.5.3	ispPAC80 的工作原理	275
6.3.2	计数器	221	7.6	可编程电源管理芯片	279
6.4	有限状态机	228	7.6.1	器件概述	279
6.4.1	有限状态机简介	228	7.6.2	ispPAC-Power1208 的 结构	280
6.4.2	编码方式	234	7.6.3	ispPAC-Power604 的 结构	280
6.4.3	剩余状态码	236	7.7	在系统可编程时钟发生器	281
6.5	倍频鉴相器设计	237	第 8 章	PAC-Designer 开发工具	283
6.5.1	倍频鉴相的原理	237	8.1	PAC-Designer 软件概述	283
6.5.2	倍频鉴相器设计	238	8.1.1	软硬件配置要求	283
6.6	交通信号灯控制器设计	241	8.1.2	软件主要功能	283
6.6.1	系统功能设计	241	8.1.3	软件的安装与注册	284
6.6.2	系统功能模块	242	8.1.4	软件设计过程	285
6.6.3	电路原理图/ABEL-HDL 混合输入设计	243	8.2	PAC-Designer 软件的使用	286
6.6.4	仿真与测试	247	8.2.1	进入图形设计环境	286
第 7 章	Lattice 公司的在系统可编程 模拟器件	250	8.2.2	软件用户图形界面	287
7.1	可编程模拟器件概述	250	8.2.3	原理图设计输入	289
7.2	ispPAC10	251	8.2.4	器件参数修改设置	290
7.2.1	ispPAC10 功能结构	251	8.2.5	软件菜单详解	292
7.2.2	ispPAC10 性能特点	253	8.2.6	典型电路宏库	301
7.2.3	ispPAC10 的工作原理	254	8.3	设计仿真	301
7.3	ispPAC20	259	8.3.1	设置仿真参数	302
7.3.1	ispPAC20 功能结构	259	8.3.2	执行仿真操作	303
7.3.2	ispPAC20 性能特点	261	8.4	器件编程	304
7.3.3	ispPAC20 的工作原理	262	8.5	ispPAC30 的软件设计	304
7.4	ispPAC30	265	8.6	ispPAC80 的软件设计	305
7.4.1	ispPAC30 功能结构	265	8.7	ispPAC-POWER1208 的 软件设计	308
7.4.2	ispPAC30 的性能特点	266			

8.7.1 设计过程	308	9.3.1 双二阶型函数电路	318
8.7.2 时序控制器的设置	310	9.3.2 双二阶滤波器的实现 ...	319
第9章 模拟电路系统设计	312	9.4 激光二极管温度控制电路 ...	321
9.1 输入输出接口电路	312	9.5 电桥测量电路	322
9.1.1 ispPAC 输入接口电路	312	9.5.1 电桥测量原理	323
9.1.2 ispPAC 模块缓冲电路	313	9.5.2 温度补偿式测量电桥 ...	323
9.1.3 单端应用	314	9.5.3 桥式测量电路的实现 ...	324
9.1.4 输入共模电压的范围 ...	314	9.6 压控振荡器	325
9.2 ispPAC 的增益调整	315	9.6.1 压控振荡器工作原理 ...	325
9.2.1 整数增益设置	315	9.6.2 压控振荡器的实现	326
9.2.2 分数增益设置	317	9.7 ispPAC Power Manager 器件的应用	328
9.3 有源滤波器	318	9.7.1 电压监控	328
		9.7.2 电源控制系统	328

第 1 章 可编程器件与电子设计自动化

现代电子产品在性能提高、复杂度增大的同时,价格却一直呈下降趋势,而且产品更新换代的速度也越来越快,其主要原因是可编程器件技术及电子系统设计的发展。随着 VLSI 向更高层次的发展,电子设计技术已迈入一个全新的阶段,EDA 技术正向着深亚微米领域进军,它们与 MCU、MPU、DSP、A/D、D/A、ROM 和 RAM 等独立器件之间的功能界限将日益模糊。特别是软/硬件 IP 芯核产业的迅猛发展,兼顾嵌入式通用与标准可编程器件的片上系统(SOC)已研制开发成功。技术的发展打破了软硬件之间最后的屏障,使软硬件工程师有了共同的语言。可以预测,未来的电子设计将是 EDA 的时代,而掌握 EDA 这门技术无疑已成为现代每一位电子设计工程技术人员必不可少的基本技能。

1.1 可编程器件的发展过程

1.1.1 集成电路的发展历史

1947 年,美国贝尔实验室发明了半导体点接触式晶体管,从而开创了人类的硅文明时代。1965 年,当时还是仙童公司工程师的摩尔在《电子学》杂志上发表文章预言,半导体芯片上集成的晶体管和电阻数量将每年翻一番,这就是著名的“摩尔定律”。在过去 40 年的时间里,“摩尔定律”一直在起着作用。尽管目前这一技术进步的周期已经从最初预测的 12 个月延长到近 18 个月,但“摩尔定律”依然有效。目前最先进的集成电路已有超过 17 亿个晶体管。

随着硅平面工艺的发展,20 世纪 60 年代先后出现了双极型和 MOS 型两种重要的集成电路(Integrated Circuit, IC)。集成电路采用专门的设计技术和特殊的集成工艺技术,把构成半导体电路的晶体管、二极管、电阻和电容等基本单元器件制作在一块半导体单晶片或者陶瓷等绝缘基片上,按照电路要求完成元器件间的互连并封装在一个外壳内,完成特定的电路功能或者系统功能。按照适用范围的不同,可以将集成电路分为标准通用集成电路和专用集成电路。通用集成电路是指社会需求量大,通用性强,集成度不高的标准系列产品。专用集成电路则是专门满足某种电子产品或系列产品的特定需求而专门设计的集成电路。

1.1.2 可编程器件的历史和现状

早期的可编程器件(Programmable Device)只有可编程只读存储器(PROM)、紫外线可擦除只读存储器(EPROM)和电可擦除只读存储器(EEPROM)3 种。由于结构的限

制,只能完成简单的数字逻辑功能。

后来出现了一类结构上稍复杂的可编程芯片 PLD(Programmable Logic Device),它能够完成各种数字逻辑功能。典型的 PLD 由一个“与”门和一个“或”门阵列组成,而任意一个组合逻辑都可以用“与—或”表达式来描述,所以 PLD 能以乘积和的形式完成大量的组合逻辑功能。这一阶段的产品主要有 PAL(可编程逻辑阵列)和 GAL(通用阵列逻辑)。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 和 EEPROM 技术等。

随着技术的进步,在 PAL 的基础上出现了一种通用阵列逻辑(GAL),其代表产品是 Lattice(莱迪思)公司的 GAL 系列产品,采用了 EEPROM 和 CMOS 工艺,实现了电可擦除和电可改写等功能。GAL 具有很强的灵活性,直到现在某些系统还在使用 GAL 完成简单的逻辑功能。

这些早期的 PLD 器件的主要缺点是结构简单,只能实现规模较小的电路。20 世纪 80 年代中期,Altera 公司和 Xilinx 公司分别推出了复杂可编程逻辑器件 CPLD(Complex Programmable Logic Device)和现场可编程门阵列 FPGA(Field Programmable Gate Array),如图 1.1 所示。

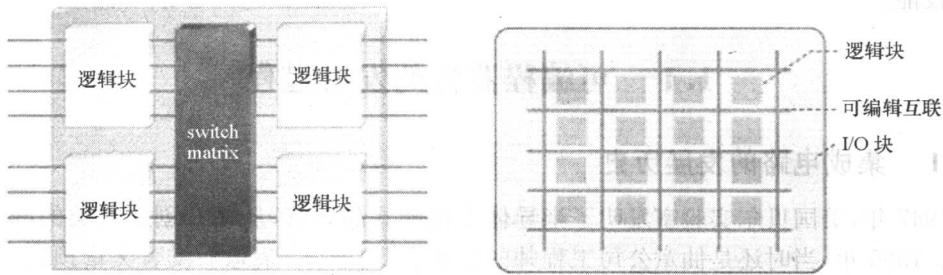


图 1.1 CPLD/FPGA 结构

1992 年,Lattice 公司率先推出了在系统可编程器件 ISP(In - System Programmable PLD,ISP - PLD)。它是一种采用了在系统可编程技术的 PLD,与传统编程技术的最大区别是它不使用编程器,而是通过下载电缆与计算机直接相连,用户在自己设计的目标系统中直接对器件编程。

1999 年底,Lattice 公司又推出了在系统可编程模拟器件(In - System Programmable Analog Circuits,ispPAC),从而翻开了模拟电路设计方法的新篇章,为电子设计自动化(EDA)技术的应用开拓了更广阔的前景。可编程模拟器件是近年来崭露头角的一类新型集成电路。它既属于模拟集成电路,又同可编程逻辑器件一样,可由用户通过现场编程和配置来改变其内部连接和元件参数从而获得所需要的电路功能。

1.2 EDA 技术简介

1.2.1 EDA 设计的典型流程

进入 20 世纪 90 年代以来,电子信息类产品的开发明显出现两个特点,一是产品的复

杂程度加深;二是产品的上市时限紧迫。这两点对 EDA 技术提出了新的要求,为此业界开始使用一种高层次的电子设计方法,也称为系统级的设计方法。

高层次设计是一种“概念驱动”形式的设计。使用高层次设计方法,设计人员不需要再通过门级原理图来描述电路,而只要针对设计目标进行功能描述。这样设计人员就可以摆脱电路细节的束缚,把精力集中在创造性的方案与概念构思上,一旦这些概念构思以高层次描述的形式输入到 EDA 系统中之后,EDA 系统就能以规则驱动的方式自动完成整个设计。

采用高层次设计方法,可以把新概念迅速转化为产品,产品的研制周期大大缩短。并且高层次设计只定义系统的行为特性,不涉及实现工艺。高层次设计方法的典型流程如图 1.2 所示。

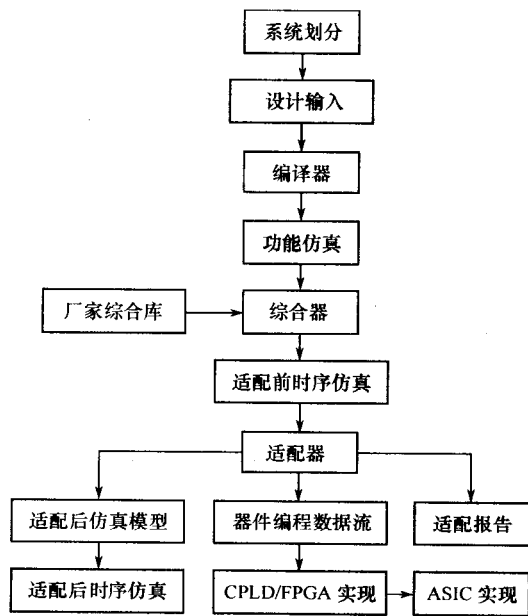


图 1.2 高层次电子设计方法流程

1.2.2 EDA 技术的设计特点

与传统的电子设计技术相比,采用 EDA 技术的可编程器件设计具有以下的特点。

1. 行为级(系统级)设计

高层综合(High Level Synthesis, HLS)的理论与方法取得较大进展,将 EDA 设计层次提高到了行为级,并划分为逻辑综合和测试综合。

逻辑综合是对不同层次和不同形式的设计描述进行转换,通过综合算法,以具体的工艺背景实现高层目标所规定的优化设计,通过设计综合工具,可将电子系统的高层行为描述转换到低层硬件描述和确定的物理实现,使设计人员无须直接面对低层电路,不必了解具体的逻辑器件,从而把精力集中到系统行为建模和算法设计上。

测试综合是以设计结果的性能为目标的综合方法,以电路的时序、功耗、电磁辐射和负载能力等性能指标为综合对象。测试综合是保证电子系统设计结果稳定可靠工作的必

要条件,也是对设计进行验证的有效方法。

2. 硬件描述语言(HDL)

EDA 技术使用硬件描述语言对电子芯片进行设计,主要包括 VHDL 和 Verilog HDL 两种标准的硬件描述语言。它们均支持不同层次的描述,使得复杂 IC 的描述规范化,便于传递、交流、保存与修改,也便于重复使用。采用硬件描述语言可对数字电子系统进行抽象的行为描述或者具体的内部线路结构描述,从而可以在电子设计的各个阶段、各个层次进行计算机模拟验证,无须构建实际的电路,既能保证设计过程的正确性,又可以大大降低设计成本,缩短设计周期。

3. 良好的可重用性

EDA 的设计语言是标准化的,不会由于设计对象的不同而改变,其设计成果是通用性的,具有良好的可移植性与可重用性,为高效高质的系统开发提供了可靠的保证。

由于使用硬件描述语言进行 EDA 设计,用户可以利用库(Library)实现设计的复用,通过库的不断扩充,EDA 工具将能够完成更多的自动设计过程。EDA 设计具有自主知识产权,目前大多数芯片厂商已经把资金重点投到芯片 IP 核的开发设计上,芯片的生产则交由专业的生产商组织完成。

1.3 可编程器件概述

可编程器件包括可编程逻辑器件(PLD)和可编程模拟器件(Programmable Analog Device)等。可编程逻辑器件用于实现数字电路和数字系统设计;可编程模拟器件用于实现模拟电路和模拟系统设计。本节将主要介绍可编程逻辑器件,关于可编程模拟器件将在第 7 章进行介绍。

1.3.1 可编程逻辑器件的分类

由于历史的原因,可编程逻辑器件的分类没有一个统一的标准。按其结构的复杂程度及性能的不同,可编程逻辑器件一般可分为 4 种:SPLD、CPLD、FPGA 及 ISP 器件。

1. 简单可编程逻辑器件(SPLD)

简单可编程逻辑器件 SPLD(Simple Programmable Logic Device)是可编程逻辑器件的早期产品。最早出现在 20 世纪 70 年代,主要是可编程只读存储器(PROM)、可编程逻辑阵列(PLA)、可编程阵列逻辑(PAL)及通用阵列逻辑(GAL)器件等。简单 PLD 的典型结构是由与阵列及或阵列组成的,能有效实现以“乘积和”为形式的布尔逻辑函数。

2. 复杂可编程逻辑器件(CPLD)

复杂可编程逻辑器件 CPLD(Complex Programmable Logic Device)出现在 20 世纪 80 年代末期。其结构上不同于早期 SPLD 的逻辑门编程,采用了基于乘积项技术和 E²PROM(或 Flash)工艺的逻辑块编程,不但能实现各种时序逻辑控制,更适合做复杂的组合逻辑电路。如 Altera 公司的 MAX 系列,Lattice 公司的大部分产品,Xilinx 公司的 XC9500 系列等。

3. 现场可编程门阵列(FPGA)

现场可编程门阵列 FPGA(Field Programmable Gate Array)是由美国 Xilinx(赛灵

思)公司率先开发的一种通用型用户可编程器件。FPGA 与 SPLD 和 CPLD 的结构完全不同,它不包括与门和或门,目前应用最多的 FPGA 是采用对基于查找表技术和 SRAM 工艺的逻辑块编程来实现所需的逻辑功能的。同 CPLD 相比,它的逻辑块的密度更高、触发器更多、设计更灵活,多用于大规模电路的设计,尤其更适合做复杂的时序逻辑。但由于 FPGA 采用的是 SRAM 工艺,掉电后数据会丢失,因此实际应用时还须外挂一个 E² PROM 或 Flash Memory 来存储编程数据。典型的器件如 Altera 公司的所有 FLEX、ACEX、APEX、Cyclone(飓风)、Stratix 系列,Xilinx 的 Spartan、Virtex 系列等。

4. 在系统可编程逻辑器件(ISP - PLD)

在系统可编程逻辑器件 ISP(In - System Programmable PLD, ISP - PLD),是 1992 年由美国的 Lattice(莱迪思)公司率先推出的。它是一种采用了在系统可编程技术的 PLD,与传统编程技术的最大区别是它不使用编程器,而是通过下载电缆与计算机直接相连,用户在自己设计的目标系统中直接对器件编程。这种全新的设计方法可以使可编程逻辑器件先装配后编程,成为产品后还可以反复编程,使生产维护和系统更新都发生了革命性的变化。在系统编程是使用一根下载电缆一端连在计算机的并行打印口上,另一端接在装配了可编程逻辑器件的 PCB 板上的插头中(JTAG 口),早期的 PLD 是不支持 ISP 技术的,目前的 CPLD、FPGA 都支持 ISP 技术可实现在线编程。

1.3.2 可编程逻辑器件的结构

1. 可编程逻辑器件的基本结构

可编程器件主要由与阵列、或阵列以及输入输出电路等构成。与阵列产生有关“与”项,或阵列将所有与项构成“与或”的形式。由于任何组合逻辑函数均可化成与或的形式,而任何时序电路均可由组合逻辑电路加上存储元件(触发器)构成,所以 PLD 的与或结构对实现数字电路具有普遍意义。PLD 的基本结构如图 1.3 所示。

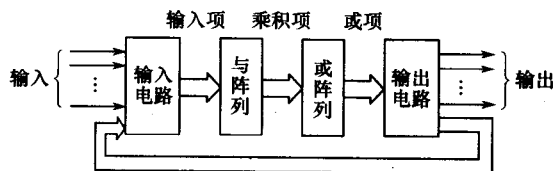


图 1.3 PLD 基本结构框图

2. 基于乘积项(Product - term)的 PLD 结构

采用这种结构的 PLD 通常称之为 CPLD,这种 PLD 芯片主要有:Altera 的 MAX7000,MAX3000 系列(EEPROM 工艺),Xilinx 的 XC9500 系列(Flash 工艺),以及 Lattice 和 Cypress 的大部分产品(EEPROM 工艺)。对于 EEPROM 或 flash 工艺制造的芯片,一上电就可以工作,无需其他芯片配合。图 1.4 给出了基于乘积项的 PLD 内部结构。

这种 PLD 由三个主要功能模块组成:宏单元(Macrocell)、可编程互连阵列(Programmable Interconnect Array,PIA)以及 I/O 控制块。其中宏单元是 PLD 的基本逻辑单元,它以乘积之和(sum - of - product)的形式执行布尔运算。16 个宏单元集中在一起构成一个逻辑阵列块(Logic Array Block,LAB)。可编程互连阵列负责连接各部分资源

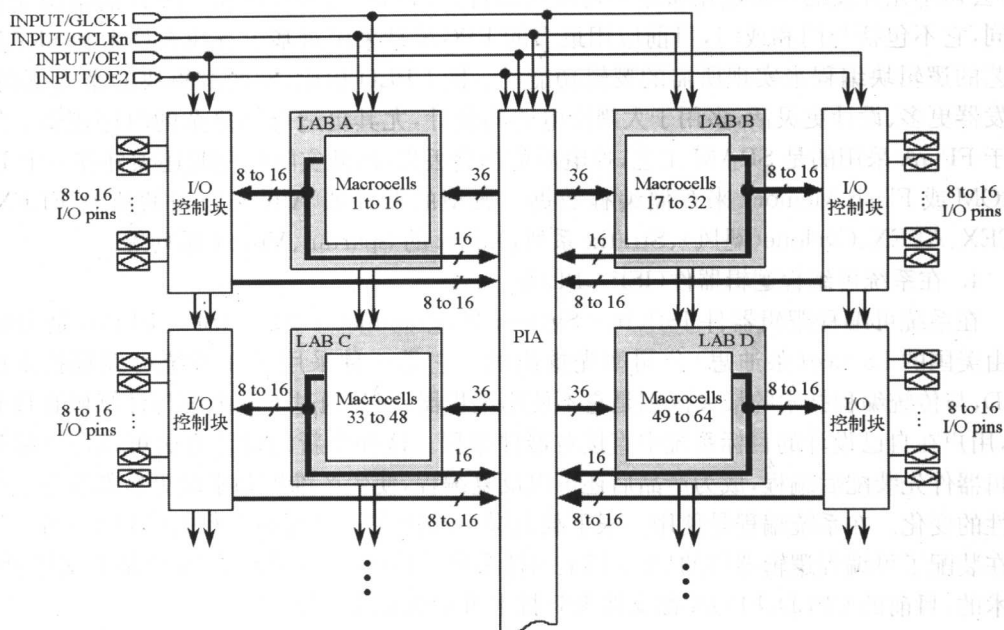


图 1.4 基于乘积项的 PLD 内部结构

来实现信号传递。图中的“INPUT/GCLK1”，“INPUT/GCLRn”，“INPUT/OE1”，“INPUT/OE2”是全局时钟、清零和输出使能信号，这四个信号通过专用连线与 PLD 中每个宏单元相连，信号到每个宏单元的延时相同并且延时最短。

3. 基于查找表(Look - Up - Table)结构的 PLD

采用这种结构的 PLD 通常称之为 FPGA，如 Altera 的 FLEX, APEX 系列, Xilinx 的 Spartan, Virtex 系列。图 1.5 所示为基于查找表的 PLD 内部结构。

查找表(Look - Up - Table)，简称 LUT，本质上就是一个 RAM。目前广泛使用的 FPGA 大多是基于查找表 LUT 结构的，可以把 n 个输入信号看成是一个 n 位的地址，一个 n 位的查找表就是一个 $2^n \times 1$ 的 RAM。每个 LUT 就是一个真值表，它把用户使用电路图或硬件描述语言建立的逻辑电路的输入逻辑与输出逻辑的对应关系，以真值表的形式保存起来。

目前 FPGA 中多使用 4 输入的 LUT，所以每个 LUT 可看成是一个有 4 位地址线 16×1 的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后，FPGA 的开发软件会自动计算逻辑电路的所有可能的结果，并把结果事先写入 RAM，这样每输入一个信号进行逻辑计算就等于输入一个地址进行查表，找出地址对应的内容，然后输出即可。

由于 LUT 主要适合 SRAM 工艺生产，所以目前大部分 FPGA 都是基于 SRAM 工艺的，而 SRAM 工艺的芯片在掉电后信息就会丢失，一定需要外加一片专用配置芯片，在上电的时候。由这个专用配置芯片把数据加载到 FPGA 中，然后 FPGA 就可以正常工作，由于配置时间很短，不会影响系统正常工作。也有少数 FPGA 采用反熔丝或 Flash 工艺，对这种 FPGA，就不需要外加专用的配置芯片。

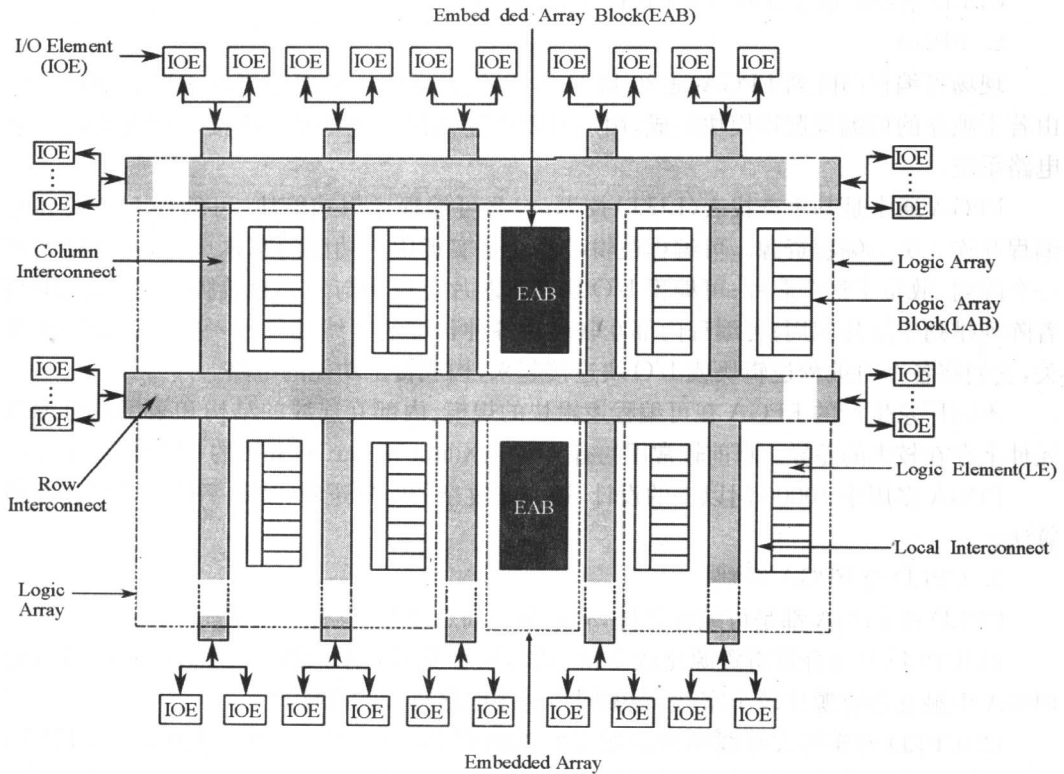


图 1.5 基于查找表的 PLD 内部结构

1.3.3 CPLD 与 FPGA

CPLD(复杂可编程器件)与 FPGA(现场可编程门阵列)是目前应用较广泛的两种可编程器件。它们是在 PAL 和 GAL 等逻辑器件的基础之上发展起来的。它们将 ASIC 集成度高的优点和可编程器件设计生产方便的特点结合在一起,特别适合于样品研制或小批量产品开发,使产品能以最快的速度上市。而当市场扩大时,它可以很容易的通过 ASIC 实现,开发风险大大降低。此外,CPLD/FPGA 的静态可重复编程或在线动态重构特性,使硬件的功能可像软件一样通过编程来修改,不仅使设计修改和产品升级变得十分方便,而且极大地提高了电子系统的灵活性和通用能力。

1. CPLD

CPLD 通常是基于乘积项(product-term)技术,采用 E²PROM(或 Flash)工艺,可反复编程,一上电就可以工作,无须其他芯片配合。采用这种结构的商用 CPLD 的芯片较多,如 Altera 公司的 MAX 系列、Lattice 公司的大部分产品及 Xilinx 公司的 XC9500 系列,这种 CPLD 都支持 ISP 技术在线编程,也可用编程器编程,并且可以加密。

CPLD 主要是由可编程逻辑宏单元(Logic Macro Cell, LMC)围绕中心的可编程互连矩阵单元组成,其中 LMC 逻辑结构较复杂,并具有复杂的 I/O 单元互连结构,可由用户根据需要生成特定的电路结构,完成一定的功能。CPLD 内部采用固定长度的金属线进行各逻辑块的互连,这种连续式的互连结构使得设计的逻辑电路具有时间可预测性。

CPLD 主要适用于 5000 门以下的中小规模设计,适合作复杂的组合逻辑。

2. FPGA

现场可编程门阵列 FPGA 是 20 世纪 80 年代出现的一种新型可编程逻辑器件。它由若干独立的可编程逻辑模块组成,用户可以通过编程将这些模块连接成所需要的数字电路系统。

FPGA 通常是基于查找表(LUT)技术,包含可编程逻辑功能块、可编程 I/O 块和可编程互连 3 类可编程资源。可编程逻辑功能块是实现用户功能的基本单元,通常排列成一个阵列,散布于整个芯片;可编程 I/O 块完成芯片上逻辑与外部封装脚的接口,常围绕着阵列排列于芯片四周;可编程内部互联包括各种长度的连线线段和一些可编程连接开关,它们将各个可编程逻辑块或 I/O 块连接起来,构成特定功能的电路。

不同厂家生产的 FPGA 在可编程逻辑块的规模、内部互连线的结构和采用的可编程元件上存在较大的差异。目前较常用的有 Altera、Xilinx 和 Lattice 公司的 CPLD/FPGA。

FPGA 多用于 10000 门以上的设计,适合作复杂的时序逻辑,如数字信号处理和各种算法。

3. CPLD 与 FPGA 区别

CPLD 和 FPGA 都是可编程器件,有很多共同点,同时也有所区别。

(1)CPLD 中组合逻辑资源比较丰富,适合做触发器有限而组合电路较多的设计,而 FPGA 中触发器资源比较丰富,适合做同步时序电路较多的设计;

(2)CPLD 的连续式布线结构决定了它的时序延迟是均匀的和可预测的,而 FPGA 的分段式布线结构决定了其延迟的不可预测性;

(3)CPLD 主要是基于 EEPROM 或 flash 存储器进行编程,系统断电时编程信息也不丢失;FPGA 大部分是基于 SRAM 编程,每次上电时需从器件外部将编程数据重新写入 SRAM 中,可在工作中快速编程,从而实现板级和系统级的动态配置。

尽管 CPLD 与 FPGA 在某些方面有一些差异,但是对于用户而言,两者的设计方法是相似的,使用 EDA 软件的设计过程也没有太大的区别。设计时只需根据所选器件型号充分发挥各自的特性即可。CPLD/FPGA 器件已成为现代高级电子系统设计方法的实现载体。

1.4 先进的编程和测试技术

1.4.1 在系统可编程(ISP)技术

20 世纪 80 年代末,Lattice 公司率先提出了一种先进的编程技术,即在系统可编程(In-System Programmable,ISP)技术。所谓“在系统可编程”是指对器件、电路板或整个电子系统的逻辑功能可随时进行修改或重构。这种重构或修改可以在产品设计、制造过程中的每个环节,甚至在交付用户之后进行。支持 ISP 技术的可编程逻辑器件称为在系统可编程逻辑器件(ispPLD)。

ISP 技术体现了一种全新的“软硬件”的概念,可以说代表了当前数字技术应用和发展的新趋势。采用 ISP 技术之后,硬件设计就可以变得像软件那样灵活而易于修改。这

不仅扩展了器件的用途,缩短了系统调试周期,也给目标设备的现场升级和维护工作带来了极大的方便。

1.4.2 边界扫描测试技术(BST)

边界扫描测试(Boundary Scan Test, BST)技术是一种将可测性设计与硅片设计融为一体的技术,它是在 20 世纪 80 年代后期提出的,主要是为了解决芯片的测试问题。

1990 年,IEEE 正式采用了由组织联合测试行动组(Joint Test Action Group, JTAG)提出的边界扫描测试的 JTAG 标准,并将其定为 IEEE1149.1-90 标准,用来解决高密度引线器件和高密度电路板上的元件的测试问题。

JTAG 标准提供了板级和芯片级的测试规范。标准的边界扫描测试只需要四根信号线,通过测试传输口能够对电路板上所有支持边界扫描的芯片内部逻辑和边界管脚进行测试。应用边界扫描技术能增强芯片、电路板甚至系统的可测试性。

1.5 可编程器件主要生产商及其器件

目前,在全球可编程器件领域,主要有三大可编程器件生产商,分别为 Altera、Xilinx 和 Lattice 公司。

1.5.1 Altera 公司的可编程器件产品

Altera 公司是目前全球最大的 FPGA/CPLD 器件生产商之一,其产品系列包括:Cyclone™、Stratix™GX、Stratix™、Excalibur™、APEX™、APEX II™、FLEX™、MAX™和 ACEX™系列,FLEX(Flexible Logic Element MatriX,灵活逻辑单元矩阵)系列器件采用查找表(LUT)结构来实现逻辑功能,而 MAX(Multiple Array MatriX 多阵列矩阵)采用“与”可编程“或”固定的乘积项(product-term)结构。APEX(Advanced Programmable Element MatriX,先进可编程逻辑矩阵)系列器件采用查找表(LUT)和嵌入式寄存器结构,Excalibur 系列的结构特征是基于 ARM 和基于 MIPS 的硬核微处理器。

1.5.2 Xilinx 公司的可编程器件产品

Xilinx 公司成立于 1984 年,是 FPGA 的发明者。1999 年 Xilinx 收购了 Philips 的 PLD 部门,成为最大可编程器件供应商之一,总部位于美国加州圣约瑟。Xilinx 公司产品种类较全,其主流高密度 PLD 产品有:属于 CPLD 的 XC9500、Coolrunner(XPLA3)系列,属于 FPGA 的 Vertex、Spartan、XC4000、XC3000 和 XC5200 等系列更低密度的 FPGA 器件系列。

1.5.3 Lattice 公司的可编程器件产品

作为 ISP 技术的发明者,Lattice 公司是世界上最大的可编程器件供应商之一。目前 Lattice 公司的 CPLD/FPGA 的开发软件为 ispLEVER。在 ispLEVER 中可以进行 VHDL、Verilog 及 ABEL 语言的设计输入、综合、适配、仿真和在线下载。Lattice 公司的主要器件系列有 CPLD 系列、FPFA 系列以及可编程模拟器件 ispPAC 系列等。