

目 录

绪 论	1
一、可编程器件对电子系统设计的影响	1
二、系统集成芯片的设计要求	2
三、设计方法的发展趋势	3
第一章 可编程 ASIC 器件	5
1.1 CPLD	5
1.1.1 CPLD 结构	5
1.1.2 典型 CPLD 器件	7
1.2 FPGA	10
1.2.1 FPGA 的结构	10
1.2.2 SRAM-查找表类型	15
1.2.3 反熔丝多路开关类型	18
1.3 计算逻辑阵列	20
2.3.1 计算逻辑阵列的结构	20
2.3.2 第一代可配置计算阵列	21
2.3.3 第二代的可配置计算阵列—XC6200 系列	21
1.4 可编程 ASIC 的基本资源	23
1.4.1 功能单元	23
1.4.2 输入—输出焊盘	25
1.4.3 布线资源	27
1.4.4 片内 RAM	30
1.5 边界扫描技术	33
1.6 可编程 ASIC 的编程元件	36
1.6.1 熔丝型开关	36
1.6.2 反熔丝开关	37
1.6.3 浮栅编程技术	39
1.6.4 静态存储器 (SRAM)	43
第二章 可编程 ASIC 的设计	47
2.1 数字系统的综合	47
2.1.1 抽象的级别	47
2.1.2 综合的定义	49
2.1.3 系统级综合	50
2.1.4 寄存器转移级综合	51
2.1.5 逻辑级综合	54
2.2 可编程 ASIC 的逻辑综合	57
2.2.1 逻辑综合概述	57
2.2.2 两级逻辑最小化	59

2.2.3	基于查找表结构的多级逻辑优化.....	63
2.2.4	立方体归并(cube-paching).....	72
2.2.5	工艺映射	73
2.2.6	基于 MUX 结构的多级逻辑优化.....	74
2.3	状态机设计	78
2.3.1	二进制编码	78
2.3.2	一个有效编码	79
2.4	FPGA 的布局和布线.....	81
2.4.1	布局.....	81
2.4.2	布线.....	85
2.4.3	布通率和布线资源	90
2.4.4	网线延时	91
2.5	Xilinx 的设计流程.....	94
2.5.1	设计输入	94
2.5.2	设计实现.....	95
2.5.3	设计验证	96
2.5.4	Xilinx FPGA 详细的设计流程	97
2.5.5	Xilinx CPLD 详细设计流程	98
第三章	CPLD — XC9500 系列	99
3.1	结构描述	99
3.1.1	功能块 (FB)	99
3.1.2	宏单元.....	99
3.1.3	乘积项分配器.....	102
3.1.4	FastCONNECT 开关矩阵	102
3.1.5	I/O 块 (IOB)	102
3.1.6	持续性.....	106
3.1.7	设计保密性.....	106
3.1.8	低功率模式.....	106
3.1.9	加电特性.....	106
3.2	XC9500 时序模型	108
3.2.1	时序模型.....	108
3.2.2	基本时序模型的参数	109
3.3	系统内编程.....	112
3.3.1	JTAG 边界扫描接口	112
3.3.2	边界扫描系统中利用系统内编程.....	115
3.4	引脚锁定能力.....	121
3.4.1	出腿预分配.....	121
3.4.2	XC9500 布线资源	121
3.4.3	数据通道的估算.....	121
3.4.4	控制通道估算.....	122

3.4.5	出腿预分配	122
3.5	设计优化	123
3.5.1	优化密度	123
3.5.2	优化时序	124
3.5.3	优化原理图设计	124
3.5.4	优化 ABEL 设计	124
3.5.5	优化 VHDL 设计	126
第四章	FPGA—XC4000 系列	127
4.1	概述	127
4.2	结构	128
4.2.1	基本积木块	128
4.2.2	可配置逻辑功能块 (CLB)	128
4.2.3	输入/输出功能块 (IOB)	143
4.2.4	三态缓冲器	149
4.2.5	沿边宽译码器	151
4.2.6	片内振荡器	151
4.3	可编程互连	152
4.3.1	互连概述	152
4.3.2	CLB 布线连接	152
4.3.3	I/O 布线	159
4.3.4	全局网线和缓冲器	160
4.4	功率分布	165
4.5	边界扫描电路	166
4.5.1	XC4000/XC5000 边界扫描特性概述	166
4.5.2	与 IEEE 标准的偏差	166
4.5.3	边界扫描硬件描述	167
4.5.4	利用边界扫描电路	171
4.6	配置	175
4.6.1	专用引腿	175
4.6.2	配置模式	175
4.6.3	设置 CCLK 频率	178
4.6.4	数据流格式	178
4.6.5	配置和读回的 CRC 校验	179
4.6.6	配置顺序	180
4.6.7	配置时序	186
第五章	可编程计算阵列—XC6200 系列	194
5.1	概述	194
5.2	功能描述	194
5.2.1	逻辑和物理组织	194
5.2.2	单元、块和砖	194

5.2.3	布线资源.....	196
5.2.4	功能单元.....	197
5.2.5	布线开关.....	200
5.2.6	时钟分布.....	200
5.2.7	清除分布.....	202
5.2.8	输入/输出功能块 (IOB)	202
5.2.9	I/O 布线.....	203
5.3	XC6200 的设计	205
5.3.1	XC6200 的板级设计	205
5.3.2	XC6200 的逻辑设计	205
5.3.3	XC6200 的软件设计	207
5.4	寄存器存取.....	207
5.4.1	映射寄存器 (Map Register)	208
5.4.2	屏蔽寄存器 (Mask Register)	210
5.5	编程.....	210
5.5.1	并行编程接口.....	210
5.5.2	通配符寄存器.....	211
5.5.3	串行编程接口.....	213
5.5.4	复位和初始化.....	215
第六章	硬件设计描述语言	216
6.1	硬件模型.....	217
6.1.1	行为模型: (Behavioral Model)	217
6.1.2	时间模型(Time Model)	219
6.1.3	结构模型.....	220
6.2	VHDL 基础.....	222
6.2.1	结构和行为.....	222
6.2.2	目标和目标类型.....	229
6.2.3	结构间挂接.....	233
6.2.4	主要的 VHDL 结构.....	235
6.2.5	并行性和时序域.....	245
6.2.6	库(library).....	248
6.3	VHDL 设计数字系统.....	251
6.3.1	编程组合逻辑.....	252
6.3.2	编程时序逻辑.....	261
6.3.3	编程有限状态机.....	267
6.3.4	VHDL 的类型综合.....	272
6.3.5	属性.....	276
第七章	ABEL 语言	285
7.1	概述.....	285
7.2	基本句法.....	290

7.2.1	支持 ASCII 字符	290
7.2.2	标识符	290
7.2.3	常数	291
7.2.4	块	292
7.2.5	注释	294
7.2.6	数	294
7.2.7	字符串	295
7.2.8	运算符、表达式和方程	295
7.2.9	集合	299
7.2.10	幅值和幅值置换	303
7.3	基本结构	305
7.3.1	头部	305
7.3.2	说明 (Declarations)	305
7.3.3	逻辑描述	308
7.3.4	测试矢量部分	310
7.3.5	结束语句	311
7.3.6	其它语句	311
7.4	状态机设计方法	312
7.4.1	状态机的例子	312
7.4.2	状态图	312
7.4.3	状态表	312
7.4.4	状态机实现	314
7.5	使用 XABEL 和 CPLD	315
7.5.1	利用真值表	315
7.5.2	利用状态图	316
7.5.3	利用特性语句	319
7.5.4	设计例子	221
第八章	Xilinx 可编程 ASIC 设计和实现	325
8.1	简介	325
8.2	设计准备工作	325
8.2.1	安装设计项目	325
8.2.2	设计的目录及文件	325
8.2.3	启动工程项管理器 (Project Manager)	326
8.2.4	拷贝设计文件	328
8.2.5	启动原理图编辑器 (Schematic Editor)	328
8.2.6	输入命令	328
8.2.7	屏幕操作	328
8.2.8	使用 XC9500 系列器件	330
8.3	完成 Calc 设计	330

8.3.1	设计说明	330
8.3.2	建立 ANDBLK2 符号	331
8.3.3	创建 ANDBLK2 原理图	332
8.3.4	完成 ALU 原理图	336
8.3.5	对 Xilinx 库元素的说明	340
8.3.6	返回 Calc 原理图	342
8.3.7	使用 XC4000E 晶振	342
8.4	从原理图控制设计实现	344
8.4.1	管脚定位 (仅限于 XC4000 系列)	344
8.4.2	控制转换速率	345
8.4.3	使用 I/O 触发器	346
8.4.4	储存 Calc 原理图	347
8.5	基于非 XC4000 系列器件的修改	347
8.5.1	RAM 堆栈实现	347
8.5.2	删除 XC4000E 晶振	348
8.6	使用 LogiBLOX	349
8.7	使用状态编辑器	351
8.7.1	创建状态机宏	351
8.7.2	定义状态	351
8.7.3	定义转移、条件和操作	352
8.7.4	生成并编译 VHDL 码	355
8.7.5	放置创建的宏	355
8.8	使用 HDL 编辑器和 X-VHDL	356
8.8.1	创建 VHDL 宏	356
8.8.2	创建并编辑 VHDL 码	357
8.8.3	编译 VHDL 码	359
8.8.4	放置 VHDL 宏	359
8.9	完成设计输入	359
8.9.1	对特殊元件的说明	359
8.9.2	使用控制文件	361
8.10	功能仿真	362
8.10.1	启动逻辑仿真器 (Logic Simulator)	362
8.10.2	选择欲测试的连线	363
8.10.3	有关总线的操作	365
8.10.4	指定激励源	366
8.10.5	储存输入波形	368
8.10.6	仿真过程	368
8.11	使用设计实现工具	370
8.12	其他操作	373
8.12.1	时序仿真	373

8.12.2	用 EPIC 检查布线后设计	375
8.12.3	使用 Xilinx 演示板检验设计	375
8.12.4	实现增量设计	376
8.13	Express 设计 calc	378
附录一	CALC 原理图	392
附录二	CALC —VHDL 文件	401
附 2.1	Calc.VHD	401
附 2.2	OSC4.VHD	402
附 2.3	ALU.VHD	403
附 2.4	SW7.VHD	404
附 2.5	DEBOUNCE.VHD	405
附 2.6	SEG7DEC.VHD	405
附 2.7	CONTROL.VHD	406
附 2.8	STACK_4K.VHD	410
参 考 书		411

绪 论

集成电路技术的发展已经为在一个芯片上集成一个系统的越来越大部分创造了条件,而光刻工艺技术的进展有可能将几种更加复杂的功能集成在同一个芯片上,因此 VLSI 技术已经发展到系统芯片化的新时代或芯片集成系统的时代,即 System on Chip 的时代。

当前,集成电路的线宽已经达到 $0.25\mu\text{m}$, DRAM 的集成度达到 256Mb 以上,逻辑电路的芯片布线达到 4-5 层金属, I/O 端口多达 900 个,相应的微处理器时钟频率已达到 300MHz,典型的 $0.35\mu\text{m}$ 工艺制作的电路密度大约为 $20000 \text{ 门}/\text{mm}^2$ 。半导体工艺正在从亚微米级向 $0.35\mu\text{m}$ 以下的深亚微米级发展,在本世纪末期望达到 $0.18\mu\text{m}$ 的集成电路制造工艺,而 $0.18\mu\text{m}$ 的工艺将会把密度提高到 $50000 \text{ 门}/\text{mm}^2$ 。综观最近 30 年集成电路和电子技术的发展,大致是 CPU 技术的集成度每 6 年提高 8 倍, DRAM 的集成度每 6 年提高 12 倍。因此预计到 2010 年集成电路的线宽达到 $0.07\mu\text{m}$, 芯片集成度达到 1G 以上的元件, DRAM 集成度达到 64Gb, 微处理器时钟频率达到 1GHz 以上。集成电路的高速发展,推动了电子技术的发展,带来了电子系统设计的不断变革。

一、可编程器件对电子系统设计的影响

电子系统设计的变革是从八十年代的中期开始的, 1984 年 Xilinx 公司发明了现场可编程门阵列 (FPGA), 随后出现了复杂可编程逻辑器件 (CPLD)。这些器件由于具有用户可编程的特性,使得电子系统的设计工程师利用与器件相应的 CAD 软件,在办公室或实验室里就可以设计自己的 ASIC 器件,实现用户规定的各种专门用途,因此构成了可编程专用集成电路 (ASIC) 的一类器件。采用可编程 ASIC,对于半导体制造厂家可按照一定的规格以通用器件大量地生产,对于用户可按通用器件从市场上选购,再由用户自己通过编程实现 ASIC 的要求,由于这种方式对厂家和用户都带来好处而受到欢迎,发展也就特别迅速,已经成为一个很重要的实现 ASIC 的手段。

可编程 ASIC 器件不仅使设计的电子产品达到小型化,集成化和高可靠性,而且器件具有用户可编程特性,大大缩短了设计周期,减少了设计费用,降低了设计风险。不仅如此,如果可编程 ASIC 具有为设计者提供系统内可再编程(或可再配置)的能力,即可编程 ASIC 器件除了具有用户可编程的能力,还具有将器件插在系统内或电路板上就能对其进行编程或再编程的能力,这就为设计者进行电子系统设计和开发提供了可实现的最新手段,在此之前是不可能做到的。采用系统内可再编程的技术,使得系统内硬件的功能可以象软件一样地被编程来配置,从而可以实时地进行灵活和方便的更改和开发。这种称为“软”硬件的全新的系统设计概念,使新一代的电子系统具有极强的灵活性和适应性,它不仅使电子系统的设计和开发以及产品性能的改进和扩充变得十分简易和方便,而且使电子系统具有多功能性的适应能力,为实现许多复杂的信号处理和信息加工提供新的思路和方法。

可编程 ASIC 的性能主要通过两方面来改进和提高,一方面是集成电路工艺水平的不断提高,另一方面是器件本身的内部结构不断改进。工艺上采用亚微米技术和三层以上金属布线以及器件结构本身的改进都使可编程 ASIC 器件的密度和运行速度有极大提高,如出现了集成度达十几万门的可编程 ASIC 器件,有代表性的是 Altera 的 FLEX10KA 和 Xilinx 的 XC4000XV 系列。98 年内 Xilinx 将推出达一百万门采用 Veritec 结构的 FPGA 器件。

从结构上看,下几代 FPGA 在芯片内将包含内置的逻辑分析仪,支持 D/A 和 A/D,并包含达到 500MHz 的差分接口;采用片内锁相环,在支持高速时钟的同时减少信号的畸变和实现时钟的复用;在提供更丰富的布线资源的同时,对逻辑和 I/O 功能块作进一步的简化;为满足用户对不同规模 RAM 的要求,在芯片内提供分布 RAM、块 RAM 和对芯片外 DRAM 高速存取的接口;简化的逻辑功能块将包含独立的快速进位链,为实现有效的乘法功能增加的专门电路,可满足 DSP 等应用的要求;逻辑功能块有快速的局部布线和相互之间的通用布线,提供虚拟诊断互连(或称为基于矢量的互连),可以精确地预测网线的延时;I/O 功能块有快速的 I/O 驱动,寄存的输入和输出,三态使能控制等,以及可编程的扭曲率、上拉和输入延时等控制的特性。此外,随芯片线宽的缩小芯片工作电压相应降低,I/O 块必须具有兼容几种电压标准的能力,保证新的工作电压下器件能与前几代器件在不同电压下连接,例如,2.5V 的芯片,内芯和 I/O 使用不同的电压,在保证降低内部功耗的同时能与 3.3V 和 5V 的器件连接。在编程技术方面,由于芯片上集成系统增加了对多次编程器件的需求,许多厂家推出采用 SRAM 作编程元件的 FPGA, Xilinx 发明的 FPGA 一开始就采用基于 SRAM 的查找表结构,即由 SRAM 存储的数值控制器件中可编程节点的通断来实现要求的功能。Actel 开始要推出的系统可编程门阵列(SPGA)没有使用其通常采用的反熔丝技术,而是采用 SRAM 技术;Cypress 也不采用它的 ViaLink 可编程反熔丝元件,而要推出基于 SRAM 的 FPGA, ViaLink 由 QuickLogic 公司生产的 pASCII 系列继续采用;Xilinx 在 96 年 10 月左右放弃它的 ViaMicro 反熔丝技术,这些都说明需求引起的变化。

近几年国外的大学和研究机构积极开展可再配置计算技术的研究,采用可配置逻辑器件作为专门的可再配置协处理器,执行器件具有全部或部分可再配置的能力,配置速度也大大提高,此外还有用户可存取的 SRAM 和通用微处理器接口,使硬件具有高速进行复杂算法计算的能力,还具有软件的灵活性,并以重复利用硬件来降低成本,模糊了硬件与软件之间的界限。大多数早期的可配置计算技术集中在利用大的 FPGA 阵列实现相对复杂的算法,目前的应用趋向于较小规模的阵列,某些情况甚至是单片 FPGA,这些应用包括为语音识别和纠错的隐含马尔柯夫模型、模糊逻辑控制、DSP 应用中的线性摺积、高带宽的图象获取、实时机器视觉处理和神经网络加速等。

总之,可编程 ASIC 器件的性能和设计软件都使其具有将数字系统集成化的可能。

二、系统集成芯片的设计要求

当可编程 ASIC 器件的集成度达到近万门时,原来采用原理图输入的设计方法就显得过于繁琐,因此器件集成度的提高正迫使设计工程师从原理图的输入方法向硬件描述语言(HDL)的设计和综合方法转变,目前至少是原理图输入和 HDL 输入两种方法的混合使用。VHDL 等硬件描述语言常常强调其具有工艺无关的设计特性,实际上不与具体器件相结合进行综合时,所谓的优化仅仅是准优化。

为了缩短 VLSI 的开发周期,逻辑合成方式在九十年代初成为设计标准,在以后的自顶向下的设计系统中,以逻辑合成技术为主,增补了各种的工具,使具有多功能的复杂 VLSI 设计能够用 EDA 软件来实现。利用语言而无须画电路图的设计方式对缩短 VLSI 的开发周期有重要的作用。面向计算机的语言式设计是用高度抽象的语言来描述系统的功能,需要相应的计算能力才能将语言描述的功能转换为芯片上具体实现的实际功能。

为了在一个芯片上实现系统集成的设计,能够在短时间内将包含数千万只晶体管的单

片集成方案开发出来, 需要采用综合利用知识产权 (IP) 功能块进行 VLSI 设计的方法。所谓 IP 功能块是以 VHDL 或 HDL 等语言描述的构成 VLSI 中各种功能单元的软件群。IP 功能块是要提供中央处理器 (CPU)、数字信号处理器 (DSP)、外设互连接口 (PCI) 和通用串行总线 (USB) 等足够可靠的各种功能的功能块。供应商在提供 IP 功能块时, 已经排除了语言描述的冗余性, 并且经过验证, 所以, 系统设计者采用 IP 功能块进行设计时, 可以集中精力去解决系统中的重点课题, 并可用优化的 IP 功能块合并到其定制的核心电路中进行逻辑合成。

如果在板级集成系统时是选择各个厂家的器件按装在印制板构成系统的话, 在芯片上集成系统是选择各个厂家的 IP 功能块综合到芯片上构成系统。VLSI 的这种设计方法也扩展和渗透到可编程 ASIC 器件, 许多 ASCII 设计者常利用 FPGA 作为产品的样本, 把大的 IP 功能块划分到几个 FPGA 来进行硬件仿真。

为了解决各个厂商的 IP 功能块之间的兼容性, 96 年 9 月多于 35 家全世界最大的厂商宣布建立国际性企业联合组织—虚拟接插接口联盟 (VSIA), 以适应系统级集成芯片工业的日益繁荣, 制定 IP 功能块的相应标准。可编程 ASCII 的厂商在提供 IP 功能块时, 允许用户修改一定的参数实施控制来增加有自身特色的功能。有的厂商如 Actel 和 Crosspoint 等与提供 IP 的公司签订合同, 建立针对其器件优化的 CPU、DSP、通信和多媒体等的核心库提供用户, 而 Altera 和 Xilinx 分别制定了 Megafuncation Program 和 LogiCore 等计划, 如 Xilinx 公司在通信系统、基于 FPGA 的 DSP 应用、PCI 及 USB 等标准接口系统内动态方式的配置结构 (RADD) 和其他基本功能等几个领域开发 LogiCore 的系统级集成的模块, 为用户系统级设计高性能和高密度的 FPGA 提供预先设计和校验的可寻访 (drop-in) 的模块。Actel 推出的 SPGA 分为面向应用和面向用户的两种模式。

设计软件的工作平台从 UNIX 的工作站转向 Windows NT 的 PC 机似乎已是必然的趋势, Xilinx 与 Synopsys 共同投资开发的针对 Xilinx FPGA 结构的 Express 综合软件, 首先推出 PC 机的版本, 然后再移植到工作站, Xilinx 的布局和布线软件 M1 版本要求 PC 机为 200MHz 时钟的 CPU, 64MB 的内存, 可以大大改善布局和布线的性能和速度。

三、设计方法的发展趋势

从九十年代初期开始, 伴随 World Wide Web (WWW) 的发展, 全世界正在进入 "Internet 的时代。今后网上工程师将主要通过 WWW 购买可再利用的设计或知识产权-IP 的产品。Internet 将越来越多地用于销售和分发各个厂家有关核心库的各种信息, 以提供设计工程师进行设计选择用。Xilinx、Altera、Actel 和 Lucent 等公司已在销售可再利用的核心库。但是, 网上工程师更长远的依靠是公司的 Intranet, 这一方面是由于 Internet 通信拥挤, 另一方面是由于 Intranet 具有更高的安全性, 可以使第三方提供的外部信息和公司内部现有的数据库集成在一起。公司的 Intranet 还具有更多的性能优势, 可以容纳公司的任何重要信息, 其中包括更换器件方面的信息等。

有许多因素正促使以 Internet 为中心的电子设计环境加速地形成。随着完成一项设计所需要的信息量和知识量的不断增加, Internet 和 Intranet 成为通信和传递信息的预选工具, 基于 Internet 的设计环境所具有的灵活性和先进性将极大地提高设计效率, 这个设计环境中, 设计工具、数据、信息和知识都是即插即用的, 并且完全和办公室自动化工具和通信工具集成在一起的。选用哪些 EDA 工具主要由承担更大责任的设计工程师决定, 除了极

先进的设计工具之外，所有工具在设计环境中都是即插即用的，各个工具价格之低使设计工程师可以相当自由地选择所喜爱的工具，因此数百万美元购买 EDA 工具的时代已经一去不复返。EDA 工具和知识产权是作为基于标准的模块化产品提供的，它们专门解决特定的设计任务，设计工程师可以在 Internet 上选购这些产品，在全球范围获取无限的信息资源，自由地混合和搭配它们，建立更富创造性、更高生产率的设计环境。

但是系统级芯片集成的电路设计必须由多学科的设计组来完成，其中包括来自许多不同学科的专家，因此设计组的每个成员，都必须善于进行人际联系和协同工作。此外，为了领导这样的设计组，需要培养新型的工程师，他们应该具有更加开阔的眼界和集成来自不同领域信息的能力。

由于民用市场是电子产业的最大客户，要求生产的批量大，为了减少风险希望采用标准的规格来规范民用市场，因此系统厂家会放弃设计自己独特的 ASIC 芯片，转为采用半导体厂家设计的符合专用标准的片上系统，即 ASSP 产品。但是，系统生产厂家要求能够调整 ASSP 产品的功能，使其适合厂家特定的应用目的所要求的规格，因此 ASSP 芯片应该是可编程的，必须能够利用设计软件对其所包含的功能进行编程来调整。

如果说 21 世纪初是多媒体的时代，它应该是计算机技术、以网络为代表的通信技术和 VLSI 为代表的半导体技术等有机融合，形成包括个人数字助理（PDA）、数字视频光盘（DVD）和预置盒（STB）等硬件新产品。其中的基本要素技术是识别、图象、声音、保密和通信等几大类，由于对象、方式、算法或者结构等因素，在这些大类之下，又可以区分成许多种不同的技术。这些基本要素技术都采取以中间件为代表的软件形态，而实际上又都内置于集成电路的芯片之中。集成电路的芯片未来的发展也是一种融合的状态，将 CPU、DSP、存储器和接口电路都融合在一个芯片之中，将这类集成系统性能在其中的芯片称为智能芯片，它需要采用多种多样的技术来使其付诸实现，对于微细加工技术的要求是 $0.1\mu\text{m}$ 或小于 $0.1\mu\text{m}$ ；要求工作电压在 1V 的量级；其中大多内置数十兆字节的 DRAM；封装要求 500 个引脚以上，与当前的 ASIC 芯片不同，这类智能芯片应该是内置 FPGA 的芯片，可以充分利用 FPGA 的可再编程和系统内动态配置的能力，从而允许用户自行编程序或现场编程序来适应各种各样的要求，使得多媒体的应用能够不断地花样翻新和更加地绚丽多彩。

本书着重介绍可编程 ASIC 的设计方法，作为设计电子系统的工程技术人员进行芯片集成的入门指导，前两章对可编程 ASCII 器件的结构及其设计方法进行较详细的分析和比较，在兼顾各个厂家产品的同时，以既生产 CPLD 又生产 FPGA 的 Xilinx 产品作为代表来说明设计的方法。考虑到硬件描述语言的设计方法是今后发展的方向，所以介绍了 VHDL 和 ABEL-HDL 两种硬件描述语言，正如前面提到的那样，硬件描述语言的编程必须与实现的器件相结合，才能得到最优的综合结果，所以在介绍硬件描述语言之前，较详细地介绍 XC9500 系列的 CPLD、XC4000E/EX 系列的 FPGA 和 XC6200 系列的可配置协处理器等结构，给出在设计软件中如何设置相应的参数利用这些结构上的特点。最后给出 Foundation 设计软件的操作过程，包括原理图输入、硬件描述语言设计宏单元和设计仿真等方面，使读者能够按照操作步骤通过实际操作达到掌握可编程 ASCII 的设计方法。作为可编程 ASIC 的主要特点是可以由用户自己完成设计实现，把最后的设计结果加载到器件上来验证设计的正确性，为此提供了进行实验的演示板电路和供进一步实验的设计例子。

第一章 可编程 ASIC 器件

近年来发展迅速的可编程专用集成电路 (ASIC) 包括复杂可编程逻辑器件 (CPLD) 和现场可编程门阵列 (FPGA) 等器件, 可以认为 CPLD 是将多个可编程阵列逻辑 (PAL) 器件集成到一个芯片, 具有类似 PAL 的结构; 而 FPGA 具有类似门阵列或类似 ASIC 的结构, 这两类器件都具有用户可编程的特性, 利用它们可以由用户实现其专门用途的集成化数字电路, 其中, 在门级按可编程结构实现算法的细粒度 FPGA 构成一类可编程计算逻辑阵列。本章对复杂可编程逻辑器件 (CPLD)、现场可编程门阵列 (FPGA) 和可编程计算逻辑阵列的一般特性进行介绍。

1.1 CPLD

具有固定数目输入和输出的任何组合逻辑函数可以在可编程只读存储器 (PROM) 中以输出为输入的查找表方式来实现, 许多实现组合逻辑的结构变型已从这一简单的概念引伸出来, 而利用 VLSI 的密度产生更通用的能实现 PCB 板上几个简单 PAL 互连功能的器件是 PAL/PROM 这类范例的扩展, 称为 PAL 构造的 PLD, 即复杂可编程逻辑器件—CPLD。

1.1.1 CPLD 结构

在考察 PAL 构造的 CPLD 之前, 先看一下或阵列可编程而与阵列固定的可编程只读存储器 (PROM), 作为两级可编程逻辑的较简单的概念, PROM 具有 n 个输入和 m 个输出, 2^n 行的查找表, 它之所以受到广泛关注是因为以下三个优点:

1. 给定输入和输出的数目, 在确定实际要实现的逻辑函数之前就可规定一个已知的器件, 这是十分有用的, 因为它允许在逻辑设计完成之前就开始 PCB 设计, 也允许在 PCB 设计完成之后更改 PROM 的设计。

2. 通过此可编程器件的延时是固定的, 与要实现的逻辑函数无关, 这也是十分重要的特性, 它允许将时序校验从逻辑设计中分出来。

3. 器件的功能可以在较高的级别上规定, 如用一系列逻辑方程或真值表表示, 这样做可以加快设计。

但是 PROM 的结构也有两个主要的缺点, 阻碍了它在市场上占有优势:

1. 硅片面积和由此的成本, 以及有时更重要的封装和板的面积都是由乘积项的数量所决定, n 输入的 PROM 有 2^n 个乘积项。

2. 通过 PROM 的延时正比于乘积项的数目, 所以延时性能随输入数目增加成正比地加大而变坏。

由于这些原因, PROM 仅适合必须完成输入信号译码等功能的场合, 例如, 由一个字符码变换到另一字符码的查找表。

在全定制设计中, 与阵列和或阵列都可编程的 PLA 是实现中到大尺寸任意逻辑函数的主要方法。但是, 与 PAL 等效的可编程逻辑器件并未能充满可编程逻辑市场, 这是因为可编程能力的额外层增加了延时, 可编程 PLA 不再具有实现 n 个变量所有函数的 PROM 特性。必须在器件设计之前选择一个较小的确定数目的乘积项。因此, 不能在输入和输出变

量数目的基础上单独选择器件。类似地，要实现的函数的任何改变可能不再适合所选的器件。

当前，在可编程逻辑市场上，最通常的结构是与阵列可编程而或阵列固定的 PAL，提供到每个输出变量的乘积项数目较小。例如 PAL22V10 有 22 个输入到与阵列和 16 个输出来自或阵列，器件的某些引脚是输入专用的，某些引脚是输入输出可编程的，或阵列的输出也可以反馈回与阵列。PAL 结构是在考察了大量实际设计的基础上得出的，即一般仅要求每个输出相对少的乘积项。此结果最重要的优点是可以提高速度，因为在或阵列中不再是宽的线与。与 PLA 比较，它的重要缺点是其乘积项专用于特定的输出引脚，可能要求特定乘积项的多个拷贝。一个设计尽管要求的总的乘积项数目比提供的要少，也可能因为一个特定的输出没有足够的乘积项而失败。当打算改变 PCB 板上 PAL 内的逻辑，如要求所选器件能提供更多乘积项到特定输出时，因超出其支持范围而感到很费劲。

问题是 PAL 为什么不是满足所有可编程逻辑需要的方案？这是因为具有太多输入和输出的 PAL 至少有两个问题：

1. 延时受到与阵列的影响，它由达到单个乘积项要求的连线长度和每个乘积项线与面积的宽度二者决定，因为要求输入变量的真和补两种形式，线与面积是输入数目的两倍。

2. 当实现一个大的逻辑函数时，利用几个小的 PAL 器件可能更为合理，并可以利用 PCB 布线的灵活性将不同的输入变量布线到不同的 PAL，或利用一个 PAL 的输出作为另一个 PAL 的输入等等。因此直接扩展 PAL 结构的效率不高。

PAL 构造的 CPLD 设计强调这些效率的问题，只是简单地扩展基本的 PAL 结构，而保持其速度和使用方便。所以对 PAL 的速度、设计的简单和延时的可预测等感到满意的设计者可能宁愿采用 CPLD。从概念上，CPLD 是由多个类似 PAL 的功能块组成，具有很长的固定于芯片上的布线资源，通过位于中心的开关矩阵互连起来。如图 1.1 所示。

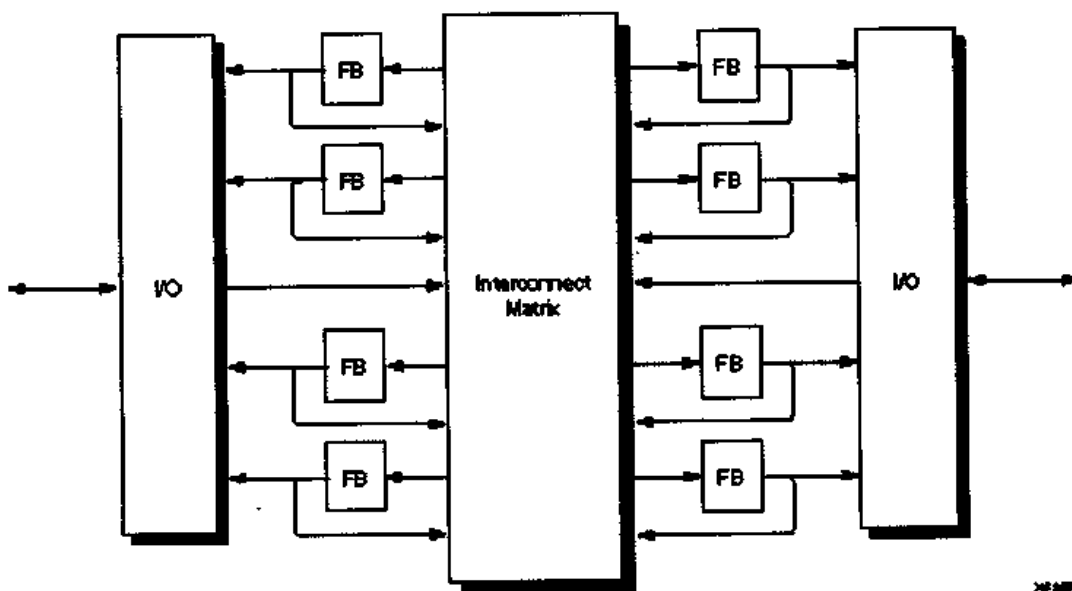


图 1.1 CPLD 的结构

与 FPGA 相比, CPLD 不采用分段互连方式, 因而具有较大的时间可预测性, 产品可以给出引脚到引脚的最大延迟时间; 此外, CPLD 具有很宽的输入结构, 适合于实现高级的有限状态机; 具有 ISP 性能的 CPLD, 可以直接在系统内对其进行编程, 因而类似于问世就具有 ISP 性能的 SRAM 查找表类型的 FPGA; CPLD 的主要缺点是功耗比较大, 15000 门以上的 CPLD 功耗要高于 FPGA、门阵列和分立器件。

1.1.2 典型 CPLD 器件

下面介绍几个厂家的一些典型的 CPLD 器件。

Altera 的 CPLD 是从基于 PLA 的传统 PLD 结构演变而来的, 它的逻辑块是由 20 个到超过 100 个输入的宽扇入的与门馈送到带有 3 到 8 个输入的或门组成的。

Altera 的与非门结构是基于可以实现与--或逻辑的与--或电路, 其输出馈送到一个异或门, 如图 1.2 所示, 图中也表示了这个基本电路可以用一个触发器和一个多路开关来扩充, 此多路开关选择锁存的或者非锁存的输出信号。异或门可以用来获得可编程的非逻辑。如果一个异或门的输入端是分离的, 它的作用和或门相同, 允许或门和异或门形成更大的或函数, 用来实现其它的算术功能, 这个可编程的形式显著增加了逻辑功能块的功能性。

Altera CPLD 的功能块具有的优点是宽与门可以用较少的功能块来形成逻辑函数, 但是很难做到有效地利用所有门的所有输入, 而导致芯片密度上的损失, 但这个损失不象一开始那样严重, 因为线与门有较高的封装密度, 逻辑连接也起布线功能的作用, 在其它结构中布线和逻辑是分开的, 未用的输入会导致同样的损失。

线与配置的缺点是利用上拉器件而消耗静态功率, 充满这些上拉器件的阵列就会消耗很大的功率, 为了减少这种功耗, MAX7000 系列中每个门可以被编程, 达到小于 60% 的功耗, 但增加约 40% 的延时, 在非临界通道中可利用这种办法来减少功耗, 它得方框图如图 1.3 所示。MAX7000 的逻辑块与 MAX5000 的类似, 只是多提供两个乘积项, 而且相邻块可以互相“借”乘积项而变得更灵活, 这是在与门和或门之间利用小的布线结构来完成的, 称为乘积项选择矩阵。

用图 1.2 和图 1.3 的 MX5000 和 MX7000 系列的逻辑单元实现一位全加器时, 当输入信号为 A_0 、 B_0 和进位输入 C_i 时, 全加器的输出 S_0 和 C_o 的逻辑方程为:

$$S_0 = (A_0 \oplus B_0) \oplus C_i = (A_0 \bar{B}_0 + \bar{A}_0 B_0) \oplus C$$

$$C_o = (A_0 \oplus B_0)C_i + A_0 B_0 = A_0 \bar{B}_0 C_i + \bar{A}_0 B_0 C_i + A_0 B_0$$

按照以上的关系式对相应的与阵列编程的结果在图 1.2 中示意给出。

CPLD 的使用取决于“适配器”(Fitter)软件, 它试图使用户设计的要求与 CPLD 提供的资源匹配, 与逻辑最小化一样, 软件必须使乘积项的数目和输出要求的输入变量与不同宏单元有效的数目匹配。这个方法可能产生不可避免的结果, 即用户的输入规定稍微改变可能导致设计不再适配实际的器件, 由于宏单元分配改变而要求引脚改变, 在 PCB 板已做好时这是用户不希望的。

CPLD 的 ISP 能力允许用户对已经固定在系统印制板上的器件编程或再编程, 它使改进样机、更新制造的流程和遥控这类系统的变更成为可能。用户对 ISP 的要求正在迅速扩大, 不仅要求器件具有可再编程的能力, 而且要求器件具有引脚锁定的能力。此外, 如果采用 JTAG 的测试存取口来实现 ISP, 作为其标准化规程, 这会给用户带来方便和好处。

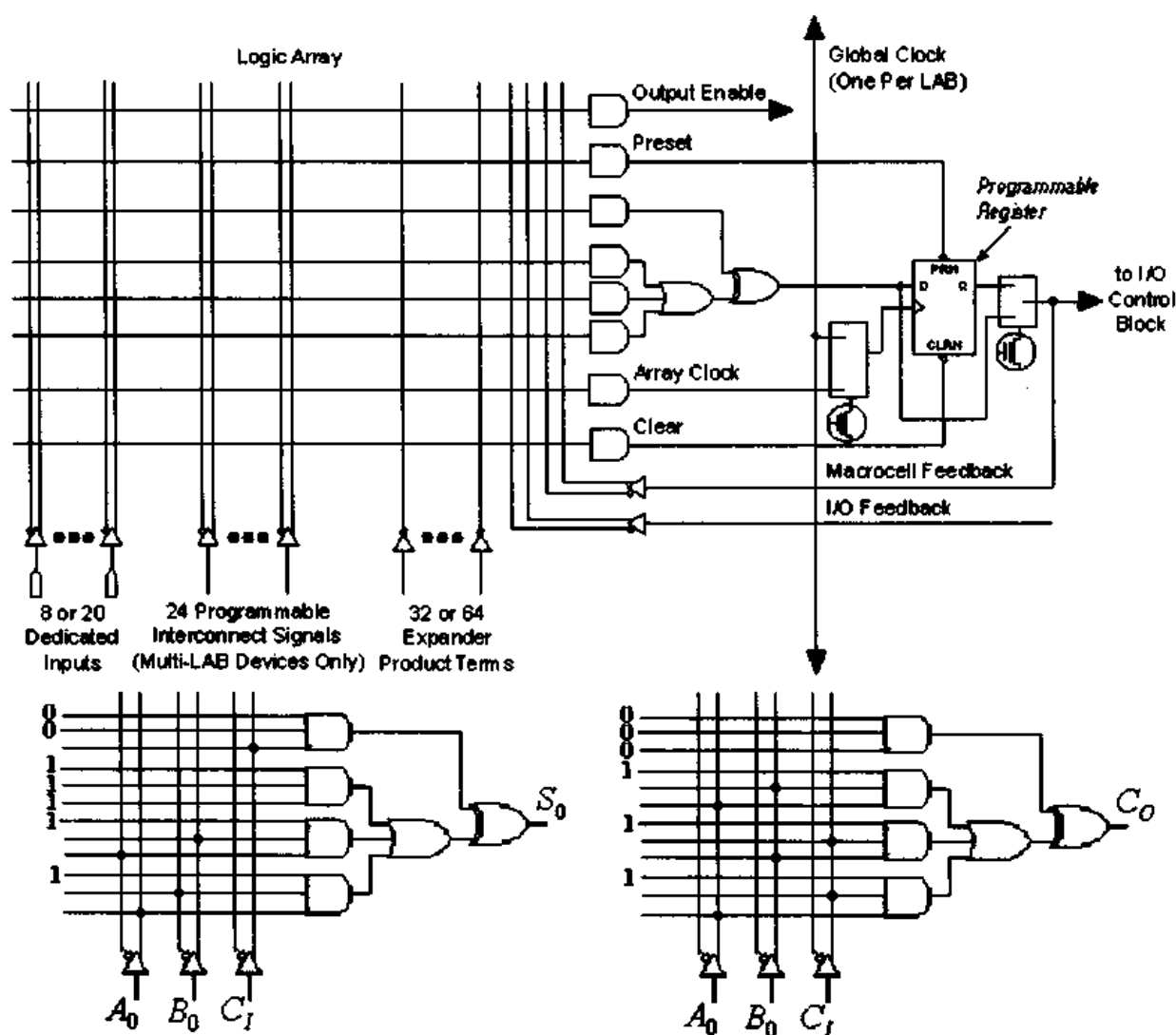


图 1.2 MAX5000 系列逻辑单元实现一位全加器原理图

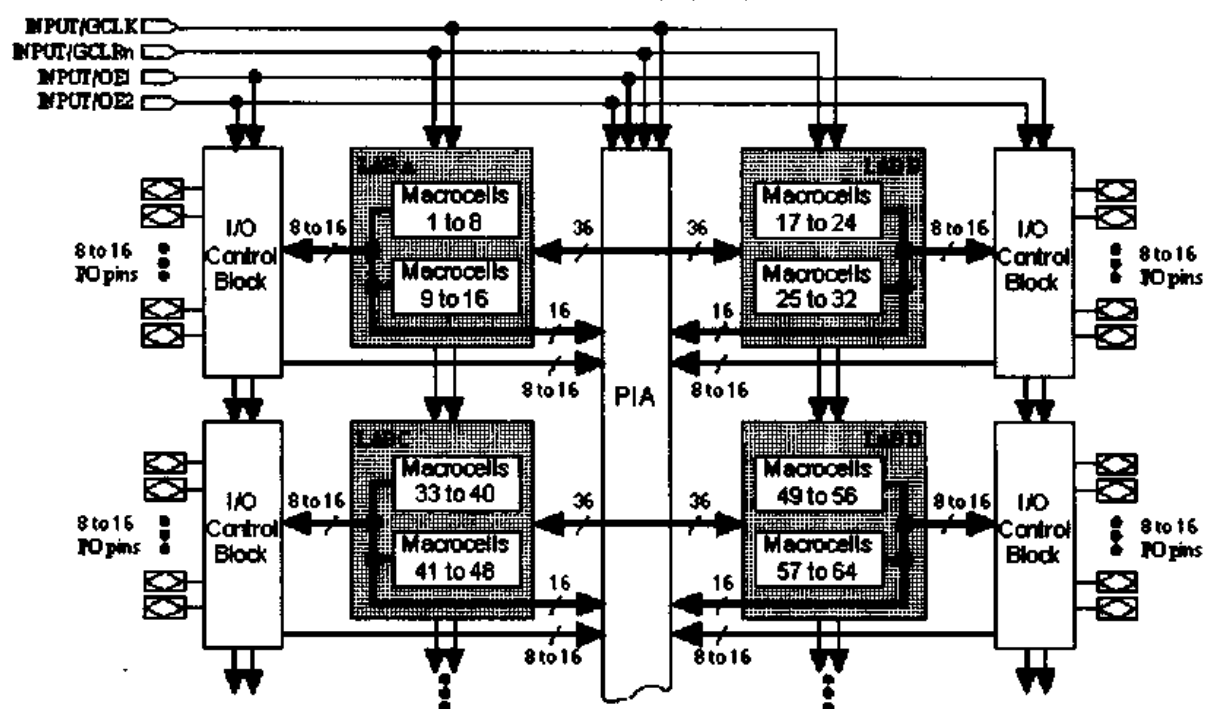


图 1.3 ALTERA 的 MAX7000 系列方框图

Lattice 的 CPLD ispLSI 系列是具有 ISP 性能的器件，但是用标准的测试程序检验时引脚锁定的能力较差，因而影响其 ISP 的性能。

Xilinx 的 XC9500 系列是采用创新的 FastFLASH 工艺制造的 CPLD，具有特殊的系统内编程（ISP）的能力，系统内编程/擦除的次数可以高达上万次，与其它可比较 CPLD 的编程/擦除的次数高一至二个数量级，这样高的耐久程度使其可用于经常要求现场更改和再配置的应用场合，它的扩展 IEEE-1149.1 边界扫描（JTAG）指令集允许器件编程模式的变更扩展和实现系统内的诊断。XC9500 系列还提供整个产品寿命期间的支持措施和最好的引脚锁定能力。在大多数 CPLD 器件中，每个 I/O 腿有宏单元通过一个 I/O 块直接驱动，当设计要求引脚锁定时，EDA 软件的适配器强迫逻辑映射到专门的宏单元来保持引脚不变。如果器件的结构受限制，在 CPLD 芯片中央的互连矩阵不具有充足的布线资源，适配器就不可能完成设计的布局和布线。有些 CPLD 器件利用一个输出布线池来补偿其初始布线资源的不足，但输出布线池引入附加的延时，不能防止适配器在布线引线时不去消耗逻辑资源，这就影响到设计的性能和资源的利用率。引脚锁定时，对逻辑提出的要求也要影响适配器对设计进行布局和布线的的能力。

对于这些引脚锁定功能带来的问题，Xilinx 的 XC9500 系列 CPLD 以丰富的布线资源、宽的功能块扇入和灵活的乘积项分配等为其特点来保证引脚锁定的实现，此系列的适配器软件也优化初始的布局使设计的引脚锁定能力最大化。包括地址译码器、数据通道和地址计数器等一系列的保证检查程序确认 XC9500 CPLD 具有优越的引脚锁定性能，这个性能对所有器件和封装类型都是一致的，宽的功能块扇入使得宽的高速逻辑函数的引脚锁定成为可能，因为不需要直通来布线，不存在由于布线拥挤的性能降低，对保持引脚锁定的设计时序的一致性与布线的布通率同样重要。

Altera 的 MAX7000、7000E 和 7000S 等器件由于缺少布线资源存在引脚锁定的问题，当利用许多宏单元的反馈和这些宏单元驱动输出引脚时出现引脚锁定的问题，这个问题在这些 Altera 器件的更高引脚数的型号时变得更差。当前的 Altera 软件没有利用逻辑直通来解决布线拥挤，代替之，当布线拥挤发生时，设计因布线而失败，这个失败可以导致不必要的重做印制板来适应设计的变化。

Lattice 的 ispLSI 器件有布线资源少和功能块扇入窄的缺点，Lattice 的适配器软件确实利用逻辑资源作为直通来尽力完成设计的布线，但是，对性能和利用率的影响较显著，甚至对一些十分简单的设计也有影响，在某些情况下，适配器软件编译时间 t_{PD} 在引脚锁定之后要比引脚锁定之前变慢达 80% 之多，宏单元数则增加 25%，所以，ispLSI 器件采用了较差的引脚锁定结构。

AMD 的 MACH5 器件似乎是具有不足的布线资源和差的适配器性能结合在一起的缺点，窄的功能在引脚锁定之后总是重新布线，但是带来由于分段延时引起的某些性能的降低。对于上述的一些典型应用的标准测试，AMD 的 MACH5 由于不能对设计布线，因而无法实现这些应用，有些甚至在初始的设计编译期间就完全失败。



1.2 FPGA

现场可编程门阵列(FPGA)是近十年加入到用户可编程技术行列中的器件,可编程门阵列在器件的选择和内部的互连上提供了更大的自由度。FPGA 的结构类似于掩膜可编程门阵列(MPGA)由逻辑功能块排列成阵列组成,并由可编程的内部连线连接这些逻辑功能块来实现不同的设计。FPGA 和 MPGA 的主要差别是 MPGA 利用集成电路制造过程进行编程来形成金属互连,而 FPGA 是利用可以编程的电子开关实现逻辑功能和互连,类似于传统的由用户进行编程的可编程逻辑器件。

FPGA 可以达到比 PLD 更高的集成度,但具有更复杂的布线结构和逻辑实现。PLD 与 FPGA 之间的主要差别是 PLD 通过修改具有固定内连电路的逻辑功能来进行编程,而 FPGA 是通过修改一根或多根分隔宏单元的基本功能块的内连线的布线来进行编程。所以 FPGA 不是建立在前面提到的可编程逻辑器件的结构上,而是在用户可编程的特性和它们的快速设计及诊断能力上类似于可编程逻辑器件。对于快速周转的样机,这些特性使得 FPGA 成为通常的选择,而且 FPGA 比 PLD 更适合于实现多级的逻辑功能。

FPGA 是由掩膜可编程门阵列和可编程逻辑器件二者演变而来的,并将它们的特性结合在一起,因此 FPGA 既有门阵列的高逻辑密度和通用性,又有可编程逻辑器件的用户可编程特性,对于 ASIC 设计来说,采用 FPGA 在实现小型化、集成化和高可靠性的同时,减少了风险,降低了成本,缩短了周期。

一个 VLSI 逻辑器件的功能是为专门应用实现时构成 ASIC,且要求的器件量必须是大量的,这意味着 ASIC 的设计必须由具有应用知识的系统工程师来进行,而不是由制造厂聘用的具有详尽工艺知识的 IC 设计者来做。完成这个转换的关键是向系统工程师们提供他们曾用过的相同的设计模型,使他们设计 IC 尽可能就象设计 PCB 一样。这导致通道型门阵列结构形式,这种形式降低了硅片的利用率,但更接近 PCB 的设计模式。靠限制使用基本工艺,使在第一块硅片就产生正确的设计变为可能,因此避免昂贵的重新设计过程。这个结构提供相同的基本逻辑功能块阵列,其连线通道可以有选择地被连接来实现所期望的功能,这是类似门阵列的具有连线通道和逻辑块结构的 FPGA。

1.2.1 FPGA 的结构

通常 FPGA 由布线资源分隔的可编程逻辑单元(或宏单元)构成阵列,又由可编程 I/O 单元围绕阵列构成整个芯片,排成阵列的逻辑单元由布线通道中的可编程内连线连结起来实现一定的逻辑功能,即分段的金属互连线可以由可编程开关以任意方式连接形成逻辑单元之间要求的信号线。一个 FPGA 包含丰富的具有快速系统速度的逻辑门、寄存器和 I/O 组成。如图 1.4 所示。

在排成阵列的可编程逻辑单元之间存在布线通道的 FPGA 构成通道型 FPGA,它们的主要结构特性为:

1. 类似门阵列的连线通道和逻辑功能块结构:逻辑资源和通信资源明显是分开的和性质不同的,反映在 CAD 系统中逻辑阵列上逻辑功能块的布局和功能块之间的布线是作为设计的不同阶段来处理的。

