

VHDL 语言与FPGA设计

—— 基于 Protel DXP 开发平台

刘瑞新 主编

胡 健 高明远 等编著



Protel DXP 电路设计丛书

VHDL 语言与 FPGA 设计

——基于 Protel DXP 开发平台

刘瑞新 主编

胡 健 高明远 等编著



机械工业出版社

本书以 Protel DXP 为开发平台, 详细介绍了 VHDL 语言的语法基础、组合逻辑电路设计、译码器设计、时序逻辑电路设计、状态机设计、CPU 及外围器件设计, 以及在 Protel DXP 环境下如何进行基于原理图的 FPGA 设计、基于 VHDL 语言的 FPGA 设计、原理图与 VHDL 的混合设计, 并结合实例讲述了如何在 Protel DXP 环境下实现测试平台程序设计、设计仿真与调试、逻辑综合等内容。

本书可作为大中专院校电子类、计算机类、自动化类、机电技术类专业的教材或教学参考书, 也可供数字电路设计人员、大规模专用集成电路设计人员参考。

图书在版编目 (CIP) 数据

VHDL 语言与 FPGA 设计——基于 Protel DXP 开发平台/刘瑞新主编.

—北京: 机械工业出版社, 2004.4

(Protel DXP 电路设计丛书)

ISBN 7-111-14154-7

I. V... II. 刘... III. 硬件描述语言, VHDL—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 019325 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 李馨馨

责任印制: 洪汉军

三河市宏达印刷有限公司印刷·新华书店北京发行所发行

2004 年 4 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·23.25 印张·573 千字

0001—5000 册

定价: 35.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

前 言

Protel 在国内已流行多年,其功能强大、资源丰富、操作简便、设计输出与其他 EDA (电子设计自动化)工具的兼容性最好,是电子工程师进行电子设计的首选工具,也是大中专院校 EDA 教学的必选软件。

Protel DXP 是 Altium 公司于 2002 年 8 月推出的,基于 Windows 2000/XP 环境的桌面 EDA 开发工具。Protel DXP 是第一个将所有设计工具集于一身的板级设计系统,包括原理图编辑输入系统模块、电路板零件放置及布线系统模块、类神经网络自动布线及推挤系统模块、模拟/数字混合式 Spice 电路仿真系统模块、可编程逻辑系统模块、CAMTastic DXP 专业 CAM 系统模块等。Protel DXP 不但兼容以前所有版本的 Protel,而且在原理图绘制、PCB 布局布线、电路仿真、PLD 设计等方面均有很大的加强。此外 Protel DXP 还具有 Windows XP 的稳定性。

Protel DXP 同前面各种版本的 Protel 的最大区别,就在于该版本完全支持用于 FPGA 设计的图形描述、VHDL 语言描述及图形与语言的混合描述,并与 Altera 及 Xilinx 建立了良好的接口,提供了大量的 FPGA 设计宏单元。

其中,VHDL (Very High Speed Integration Circuits HDL,超高速集成电路硬件描述语言)硬件描述语言是数字电路设计者、大规模专用集成电路(ASIC)设计者与电子设计自动化(EDA)工具之间的接口语言,是现代电子设计的基础语言。Protel DXP 可作为 FPGA 设计的前端工具,完全支持 IEEE 93 及 IEEE 87 标准的 VHDL 语言,并且可以产生 EDIF 格式的输出文件,也可以导入到 Max+Plus-II 及 Quartus 等各种芯片厂家所提供的编译器,实现对芯片的下载烧录。

本书以 Protel DXP 为开发平台,详细介绍了 VHDL 语言的语法基础、组合逻辑电路设计、译码器设计、时序逻辑电路设计、状态机设计、CPU 及外围器件设计,以及在 Protel DXP 环境下如何进行基于原理图的 FPGA 设计、基于 VHDL 语言的 FPGA 设计、原理图与 VHDL 的混合设计,并结合实例讲述了如何在 Protel DXP 环境下实现测试平台程序设计、设计仿真与调试、逻辑综合等内容。

本书可作为大中专院校电子类、计算机类、自动化类、机电技术类专业的教材或教学参考书,也可供数字电路设计人员、大规模专用集成电路设计人员参考。

本书由刘瑞新主编,胡健、高明远编著,参加编写的还有刘玉宾、范文军、穆杰、吴全文、沈淑娟、周星、孙洪玲、臧顺娟、彭守军、崔瑛。由于计算机信息基础发展迅速,书中不足和谬误之处,恳请广大师生批评指正。

作 者

目 录

前言

第 1 章 概述	1
1.1 PLD、CPLD 及 FPGA	1
1.2 CPLD/FPGA 结构与原理	2
1.2.1 基于乘积项 (Product-Term) 的 CPLD 结构	2
1.2.2 乘积项结构 CPLD 的逻辑实现原理	4
1.2.3 查找表 (Look-Up-Table) 的原理与结构	5
1.2.4 基于查找表 (LUT) 的 FPGA 的结构	5
1.2.5 查找表结构的 FPGA 逻辑实现原理	6
1.3 数字系统的设计方法	8
1.3.1 传统的硬件设计方法	8
1.3.2 EDA 设计	8
1.4 PLD 系统设计方式	9
1.4.1 基于原理图的 PLD 设计	9
1.4.2 基于 HDL 的 PLD 设计	10
1.4.3 HDL 与原理图的混合设计	11
1.5 硬件描述语言	12
1.5.1 硬件描述语言的分类	12
1.5.2 VHDL 语言的特点	13
1.5.3 PLD 开发工具	13
1.6 可编程逻辑器件的最新发展趋势	13
1.7 思考与练习	15
第 2 章 认识 Protel DXP	16
2.1 Protel DXP 的系统需求	16
2.2 初识 Protel DXP	16
2.2.1 设置系统参数	17
2.2.2 Protel DXP 文档组织	24
2.3 面板的操控	25
2.4 工具栏的摆放	27
2.4.1 启动原理图编辑器	28
2.4.2 摆放工具栏	29
2.5 关闭文档及编辑窗口	30
2.6 思考与练习	30
第 3 章 基于原理图的 FPGA 设计	31
3.1 认识 Protel DXP_Sch 设计界面	31

3.2	创建 FPGA 设计项目	33
3.2.1	创建项目组文档	33
3.2.2	FPGA 项目文档	34
3.2.3	设置 FPGA 项目选项	36
3.3	创建原理图源文件	37
3.3.1	图纸创建及设置	37
3.3.2	厂家选择及单元库的装载	38
3.3.3	放置元件	39
3.3.4	放置端口	40
3.3.5	创建连接	42
3.4	设计配置	45
3.5	产生 EDIF-FPGA 网表	47
3.6	后向标注 FPGA 项目	48
3.7	后向标注 PCB 项目	49
3.8	仿真分析	49
3.8.1	创建测试平台文档	49
3.8.2	仿真 FPGA 设计	52
3.9	思考与练习	55
第 4 章	VHDL 语言程序结构	57
4.1	VHDL 程序的设计单元	57
4.1.1	实体的组织和设计	58
4.1.2	实体说明	59
4.1.3	构造体	64
4.2	构造体的描述方式	66
4.2.1	构造体的行为描述方式	66
4.2.2	构造体的数据流描述方式	68
4.2.3	构造体的结构描述方式	69
4.3	构造体的子结构	71
4.3.1	构造体的多模块描述	72
4.3.2	构造体的多进程描述	75
4.3.3	构造体的子程序描述	77
4.4	库、包集合及包集合体	81
4.4.1	库	81
4.4.2	包集合	83
4.4.3	包集合体	84
4.5	配置	85
4.6	在 Protel DXP 环境下编辑 VHDL 源文件	86
4.6.1	创建源文件	87
4.6.2	设计仿真	90

4.7 思考与练习	90
第 5 章 对象、数据类型及运算符	91
5.1 信号的使用	91
5.1.1 信号的初始化	92
5.1.2 信号的使用	92
5.2 变量的使用	94
5.3 常数及常量的使用	95
5.3.1 常数	95
5.3.2 常量	96
5.4 数据类型及子类型	97
5.4.1 标量类型	98
5.4.2 复合类型	101
5.4.3 存取及不完全类型	104
5.4.4 文件类型	105
5.5 类型转换及类型标记	107
5.5.1 显式类型转换	107
5.5.2 类型转换函数	108
5.5.3 多义常量类型	110
5.5.4 决断和非决断类型	111
5.6 了解 VHDL 的运算符	114
5.6.1 逻辑运算符	116
5.6.2 关系运算符	117
5.6.3 adding 运算符	118
5.6.4 符号运算符	119
5.6.5 multiplying 运算符	119
5.6.6 其他运算符	120
5.7 思考与练习	122
第 6 章 标准逻辑的使用	123
6.1 IEEE 1164 标准	123
6.1.1 IEEE 1164 标准的优点	124
6.1.2 使用标准逻辑包	124
6.1.3 Std_logic_vector 和 Std_ulogic_vector	126
6.2 类型转换及标准逻辑	126
6.3 IEEE 1164 标准逻辑数据类型	128
6.3.1 Std_ulogic 类型	128
6.3.2 Std_ulogic_vector 类型	128
6.3.3 Std_logic 类型	128
6.3.4 Std_logic_vector 类型	129
6.4 标准逻辑运算符	129

6.4.1	逻辑运算符	129
6.4.2	数组逻辑运算符	129
6.5	标准逻辑类型转换函数	130
6.5.1	类型转换函数	130
6.5.2	强制类型转换函数	130
6.6	边沿检测及其他函数	131
6.6.1	边沿检测函数	131
6.6.2	其他函数	131
6.7	1076.3 标准	131
6.7.1	数值型数据类型的使用	132
6.7.2	数值型标准运算符	133
6.7.3	数字大小调整函数	136
6.7.4	数值类型转换函数	136
6.7.5	数值匹配函数	136
6.7.6	数值变换函数	137
6.8	思考与练习	137
第 7 章	并发语句	138
7.1	并发域	138
7.2	信号代入语句	139
7.2.1	并发信号代入语句	140
7.2.2	条件信号代入语句	140
7.2.3	选择信号代入语句	141
7.2.4	条件与选择信号代入语句的对比	143
7.3	并发过程调用语句	144
7.4	生成语句	145
7.4.1	for-generate 语句	145
7.4.2	if-generate 语句	146
7.5	并发进程语句	148
7.6	元件例化语句	149
7.7	端口及类属的映射语句	151
7.7.1	端口映射语句	151
7.7.2	类属映射语句	152
7.8	延时说明子句	153
7.8.1	惯性延时语句	154
7.8.2	传输延时语句	155
7.9	信号驱动	156
7.10	思考与练习	157
第 8 章	顺序语句	158
8.1	进程中的顺序语句	158

8.1.1	用进程语句描述组合逻辑	158
8.1.2	用进程语句描述寄存器逻辑	160
8.1.3	用进程语句描述状态机	161
8.1.4	用进程描述测试激励	164
8.2	子程序中的顺序语句	165
8.3	信号及变量代入语句	166
8.4	If 语句	168
8.4.1	简单 if 语句结构	168
8.4.2	if 语句的双路选择结构	169
8.4.3	if 语句的多路选择结构	170
8.5	Case 语句	174
8.6	Loop 语句	176
8.6.1	For loop 结构	176
8.6.2	While loop 语句	178
8.6.3	无限循环语句	179
8.6.4	Exit 语句	180
8.6.5	Next 语句	181
8.7	Wait 语句	182
8.7.1	wait on 结构	182
8.7.2	wait-until 结构	183
8.7.3	wait for 结构	184
8.8	Assert 语句	184
8.9	Null 语句	185
8.10	思考与练习	186
第 9 章	对象的属性描述	187
9.1	数值类属性	187
9.1.1	一般数据的数值属性	187
9.1.2	数组的数值属性	189
9.1.3	块的数值属性	190
9.1.4	实体的数值属性	191
9.2	函数类属性	191
9.2.1	数据类型的属性函数	191
9.2.2	数组的属性函数	194
9.2.3	信号的属性函数	195
9.3	信号类属性	198
9.3.1	S'delayed[(t)]属性	198
9.3.2	S'stable[(t)]属性	200
9.3.3	S'quiet[(t)]属性	201
9.3.4	S' transaction[(t)]属性	201

9.4	数据类型类属性	202
9.5	区间类属性	202
9.6	自定义属性	203
9.7	思考与练习	204
第 10 章	配置	205
10.1	默认配置	205
10.2	元件建模与元件配置	206
10.2.1	元件建模	207
10.2.2	元件配置	211
10.3	Generic 配置	212
10.4	Block 的配置	215
10.5	构造体的配置	217
10.6	思考与练习	218
第 11 章	测试平台的描述	219
11.1	一个简单的测试台	219
11.2	用断言语句显示复杂字符串	224
11.3	使用循环语句及多进程	226
11.4	编写测试向量	227
11.5	使用 text I/O	231
11.5.1	读出/写入表格数据	231
11.5.2	非表格数据的读出	235
11.6	使用 Protel DXP 测试平台程序模板	235
11.6.1	创建 FPGA 项目文档及源文件	235
11.6.2	为设计创建测试平台程序	237
11.7	思考与练习	239
第 12 章	VHDL 与原理图的混合设计	244
12.1	创建 VHDL 设计文档	245
12.2	创建 VHDL 顶级原理图文档	249
12.3	FPGA 项目设置	254
12.4	设计编译	255
12.5	创建 VHDL 测试平台程序	258
12.6	设计仿真	261
12.7	思考与练习	266
第 13 章	组合逻辑电路的 VHDL 设计	267
13.1	门电路设计	267
13.1.1	四 2 输入与非门 (74HCT00)	267
13.1.2	四 2 输入或非门 (74HCT32)	268
13.1.3	N 输入与门	268
13.1.4	异或门	269

13.2	比较器的设计	271
13.2.1	4 位二进制数大小比较器 (74HC85)	271
13.2.2	8 位相等比较器 (74HCT688)	272
13.3	译码器的设计	273
13.3.1	7 段 BCD 码显示译码器 (74LS48)	273
13.3.2	双 2-4 译码器 (74HCT139)	274
13.3.3	3-8 线译码器 (74LS138)	275
13.3.4	地址译码器 (M68008)	276
13.4	数据选择器	277
13.5	编码器	278
13.5.1	8-3 优先编码器 (74LS148)	278
13.5.2	汉明纠错码编码器	279
13.6	运算器	281
13.6.1	半加器	281
13.6.2	全加器	282
13.6.3	N 位加法器	283
13.6.4	全减器	284
13.6.5	4 位加/减器	285
13.6.6	求补运算器	286
13.7	总线器件的设计	286
13.7.1	8 位总线收发器 (74HCT245)	286
13.7.2	三态总线	287
13.8	编码转换器	287
13.8.1	BCD 码转换为二进制数	287
13.8.2	8421BCD 码转换为余 3 码	289
13.8.3	BCD 码转换为移位码	290
13.8.4	BCD 码转换为格雷码	291
13.8.5	二进制转换成 BCD 码	292
13.9	奇偶校验位发生器设计	293
13.10	思考与练习	294
第 14 章	时序逻辑电路设计	295
14.1	触发器的设计	295
14.1.1	D 触发器设计	295
14.1.2	JK 触发器设计	298
14.2	计数器设计	299
14.2.1	同步计数器设计	299
14.2.2	异步计数器设计	310
14.3	寄存器设计	311
14.3.1	锁存寄存器设计	311

14.3.2 移位寄存器	312
14.4 思考与练习	317
第 15 章 有限状态机 (FSMs)	318
15.1 反馈机制	318
15.1.1 反馈信号	318
15.1.2 反馈变量	319
15.2 Moore 状态机	320
15.2.1 输出寄存器	321
15.2.2 输入寄存器	322
15.3 Mealy 状态机	322
15.4 状态机的设计步骤	323
15.5 状态机设计举例	326
15.5.1 Mealy 状态机设计	326
15.5.2 Moore 状态机设计	331
15.6 思考与练习	334
第 16 章 CPU 及外围器件设计	335
16.1 存储器设计	335
16.1.1 只读存储器 (ROM)	335
16.1.2 随机读写存储器 (RAM)	337
16.2 模数及数模转换器	340
16.2.1 模数转换器	340
16.2.2 数模转换器	343
16.3 简单微处理器设计	343
16.3.1 CPU 指令定义包集合	344
16.3.2 运算函数包集合	344
16.3.3 256 字节 ROM 设计	347
16.3.4 16 字节 RAM 设计	348
16.3.5 8 位 CPU 模型设计	349
16.3.6 微处理器系统结构描述	352
16.4 可编程并行接口 8255A 的设计	353
16.5 思考与练习	357
参考文献	358

第1章 概述

Protel DXP 是 Altium 公司于 2002 年 8 月推出的, 基于 Windows 2000/XP 环境的新一代电子设计自动化 (EDA) 开发工具, 是第一个集所有设计工具于一身的板级设计系统。Protel DXP 同以前 Protel 产品的最大区别, 就在于该版本完全支持用于 FPGA 设计的图形描述、VHDL 语言描述及图形与 VHDL 语言的混合描述, 并与 Altera 及 Xilinx 建立了良好的接口, 提供了大量的 FPGA 设计宏单元。

Protel DXP 的 FPGA 设计工具支持 EDIF 格式的输出文件, 并且可以导入到 Max+Plus-II 及 Quartus 编译器中。

本章主要介绍 PLD 的分类、CPLD/FPGA 结构与原理、PLD 的最新发展趋势、FPGA 的设计过程、VHDL 语言开发环境及硬件平台等内容。有关 Protel DXP 的基本操作及原理图的详细设计过程请参阅参考文献 1。

1.1 PLD、CPLD 及 FPGA

由于所有的逻辑电路都可以用“积之和”的布尔方程来表示, 所以出现了由与阵列和或阵列构成的可编程逻辑器件 (PLD, Programmable Logic Device), 两级逻辑阵列的一个阵列或全部阵列可以通过编程来实现所要求的逻辑功能。只有或阵列可编程时, 称为可编程只读存储器 (PROM); 只有与阵列可编程时, 称为可编程阵列逻辑 (PAL, Programmable Array Logic); 两个阵列都可编程时, 称为可编程逻辑阵列 (PLA, Programmable Logic Array)。

随着 IC 规模的不断扩大, PLD 器件的集成度也在不断提高, 由此出现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)。由于初期它们只能实现粘附逻辑的功能, 所以通称为 PLD。另外, 还有些公司 (如 Altera) 把 CPLD 和 FPGA 产品都称为 CPLD, 而 FPGA 被分类为 SRAM PLD。

CPLD 和 FPGA 器件在结构上是完全不同的。由于这些器件具有可编程性, 利用它们可以实现用户的专门功能, 所以国内也称这些器件为可编程 ASIC (Application Specific Integrated Circuits, 即专用集成电路, 又称单元基)。

复杂可编程逻辑器件 (CPLD) 可以看作宏单元的多个 PAL 器件的集成, 一个典型的 CPLD 通常包含 18~256 个宏单元, 可以取代 2~62 块 SPLD 或几十片 74 系列的标准逻辑器件。其内包含一个很大的布线阵列来实现宏单元以及 I/O 之间的连接, 所以 CPLD 不是 PAL 器件的简单扩充。但是, 相对于 FPGA 来说, CPLD 的宏单元扇入大, 器件拥有的触发器少, 器件给出的引脚到引脚的延时是确定的, 一般通过 JTAG (IEEE1149) 实现对器件的编程。为了弄清 CPLD 的结构, 必需了解以下三个问题:

- 1) 一个宏单元里有多少个乘积项。
- 2) 一个宏单元里的乘积项是否能被分配或借给另一个宏单元。

3) 开关矩阵是全部内连还是部分内连。

在一般情况下, 用宏单元的个数来描述 CPLD 的性能, 而不沿用传统的门阵列数进行描述。另外, CPLD 采用 EPROM、EEPROM 和 FLASH 这三种制造工艺技术。

FPGA 容量通常大于 CPLD, 其内部结构与 CPLD 完全不同。FPGA 具有类似门阵列的结构, 通常是可编程的逻辑单元排成阵列, 位于器件的中央, 四周有可编程 I/O 焊盘围绕, 每行和每列之间有可编程的互连资源实现逻辑单元以及 I/O 之间的连接, 以便实现各种复杂的逻辑运算。一般的 FPGA 都含有成千上万的逻辑单元 (又称逻辑块), 还有一定数量的触发器。FPGA 的逻辑块有粗粒和细粒之分。粗粒结构的逻辑块较大, 含有两个以上的查询表和触发器。这种 FPGA 器件比较适合高性能应用。细粒结构的逻辑块相对较小, 含有一个 2 输出功能逻辑块、一个 4 至 1 多路器和一个触发器。

这种器件具有时钟功能, 常用于可综合逻辑设计。高密度的 FPGA 按逻辑功能块和编程方式可划分为两大类:

1) 用查找表作逻辑单元, 用 SRAM 进行编程, 称为 SRAM 查找表结构。器件的编程是掉电丢失的, 但可以无限次反复编程。这种结构的 FPGA 的每个逻辑单元通常由一个 4 输入查找表和一个触发器组成。随着 SRAM 技术的迅速发展, 这类 FPGA 的门数已接近千万门 (如 Xilinx 公司的 Virtex II 系列 XC2V8000)。

2) 用多路选择器作为逻辑单元, 利用反熔丝多路开关结构, 集成了一次可编程 (OTP) 技术, 因此其内连效率较高。

相对 CPLD 来说, FPGA 的逻辑单元扇入小, 触发器数量大, 布局和布线的延时是不确定的, 但是可行的。

PLD 领域中 Xilinx 和 Altera 双峰对峙, Xilinx 称霸 FPGA, Altera 则垄断 PLD。

1.2 CPLD/FPGA 结构与原理

CPLD 内部多采用乘积项结构, 而 FPGA 内部多采用查找表结构。本节简单介绍这两种器件的结构及逻辑实现。

1.2.1 基于乘积项 (Product-Term) 的 CPLD 结构

采用这种结构的 CPLD 芯片有: Altera 的 MAX7000 及 MAX3000 系列 (EEPROM 工艺), Xilinx 的 XC9500 系列 (Flash 工艺) 和 Lattice、Cypress 的大部分产品 (EEPROM 工艺)。

以 MAX7000 为例, 这种 CPLD 的总体结构如图 1-1 所示。其他型号的结构与此相似。

这种结构的 CPLD 包含三个主要结构模块: 宏单元 (Macrocell)、可编程连线阵列 (PIA) 和 I/O 控制块。宏单元是 PLD 的基本结构, 由它来实现基本的逻辑功能。图 1-1 中 LAB A 至 LAB D 是多个宏单元的集合。可编程连线负责信号传递, 连接所有的宏单元。I/O 控制块负责输入输出的电气特性控制, 比如可以设定集电极开路输出, 三态输出等。INPUT/GCLK1, INPUT/GCLKn, INPUT/OE1, INPUT/OE2 是全局时钟、清零和输出使能信号, 这几个信号有专用连线与 PLD 的每个宏单元相连, 信号到每个宏单元的延时相同并且延时最小。

宏单元的具体结构如图 1-2 所示。

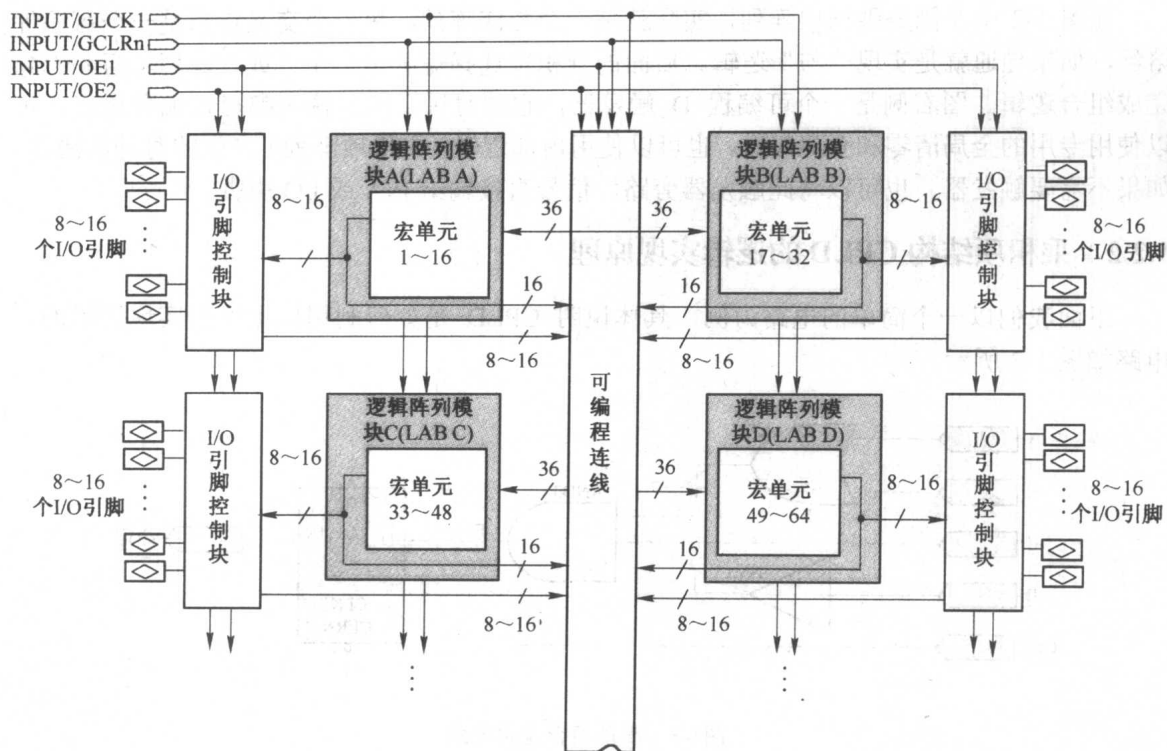


图1-1 基于乘积项的CPLD内部结构

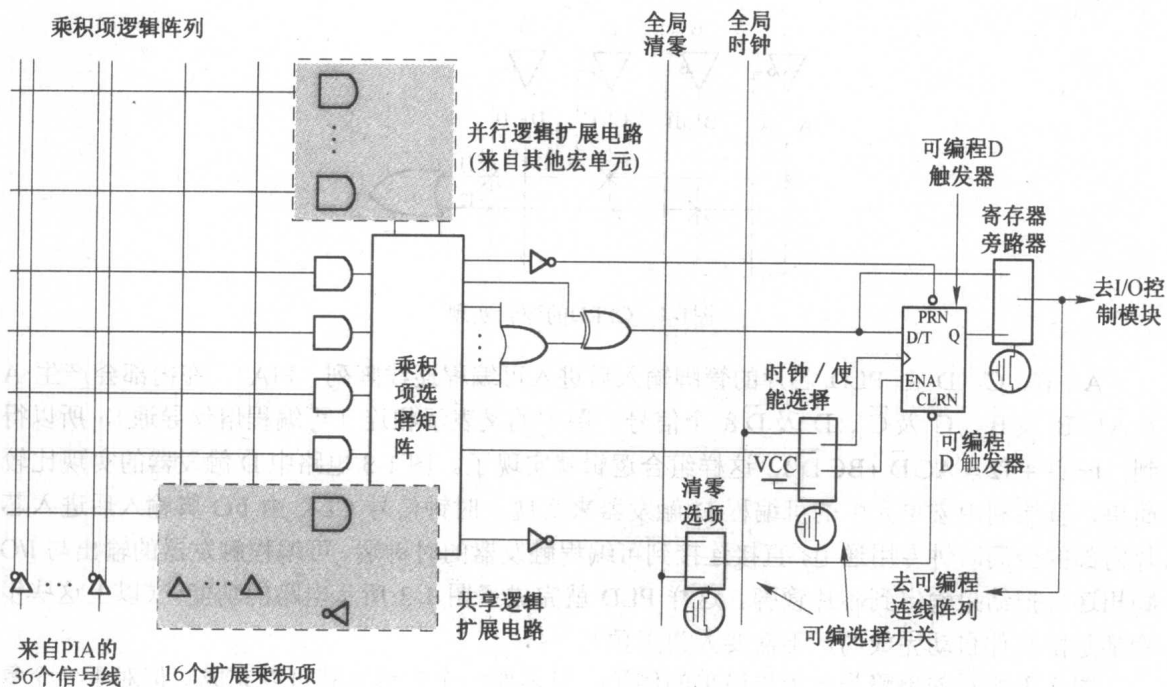


图1-2 宏单元结构

在图 1-2 中左侧是乘积项阵列，实际就是一个与或阵列，每一个交叉点都是一个可编程熔丝，如果导通就是实现“与”逻辑。后面的乘积项选择矩阵是一个“或”阵列。两者一起完成组合逻辑。图右侧是一个可编程 D 触发器，它的时钟、清零输入都可以编程选择，可以使用专用的全局清零和全局时钟，也可以使用内部逻辑（乘积项阵列）产生的时钟和清零。如果不需要触发器，也可以将此触发器旁路，信号直接输给 PIA 或 I/O 引脚。

1.2.2 乘积项结构 CPLD 的逻辑实现原理

下面我们以一个简单的电路为例，具体说明 CPLD 是如何利用以上结构实现逻辑的，电路如图 1-3 所示。

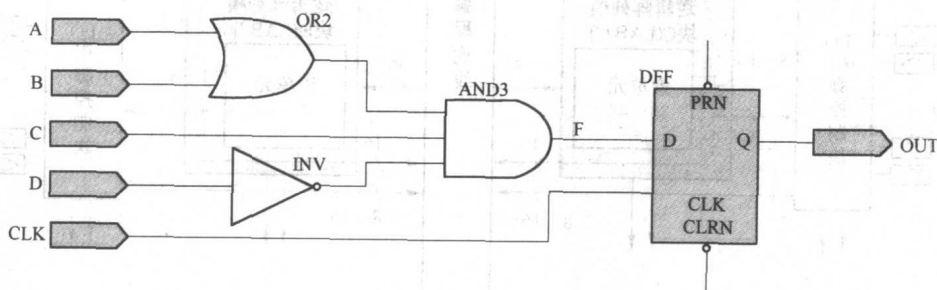


图1-3 简单组合逻辑电路

假设组合逻辑（AND3）的输出为 F，则 $F = (A+B)C\bar{D} = AC\bar{D} + BC\bar{D}$
PLD 将以图 1-4 所示方式来实现组合逻辑 F。

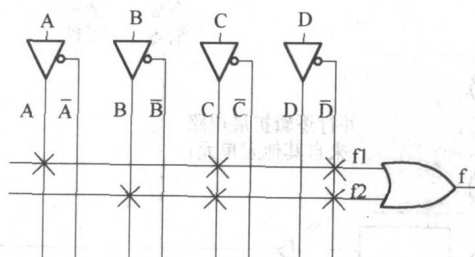


图1-4 CPLD的逻辑实现

A、B、C、D 由 PLD 芯片的管脚输入后进入可编程连线阵列（PIA），在内部会产生 A 及 $\bar{A}$ 、B 及 $\bar{B}$ 、C 及 $\bar{C}$ 、D 及 $\bar{D}$ 8 个信号。图中的叉表示相连（可编程熔丝导通），所以得到： $f = f1 + f2 = AC\bar{D} + BC\bar{D}$ ，这样组合逻辑就实现了。图 1-3 电路中 D 触发器的实现比较简单，直接利用宏单元中的可编程 D 触发器来实现。时钟信号 CLK 由 I/O 脚输入后进入芯片内部的全局时钟专用通道，直接连接到可编程触发器的时钟端。可编程触发器的输出与 I/O 脚相连，把结果输出到芯片管脚。这样 PLD 就完成了图 1-3 所示电路的功能。（以上这些步骤都是由软件自动完成的，不需要人为干预）

图 1-3 所示的电路是一个很简单的例子，只需要一个宏单元就可以完成。但对于一个复杂的电路，一个宏单元是不能实现的，这时就需要通过并联扩展项和共享扩展项将多个宏单元相连，宏单元的输出也可以连接到可编程连线阵列，作为另一个宏单元的输入。这样 PLD

就可以实现更复杂逻辑。

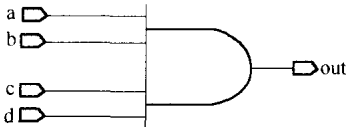
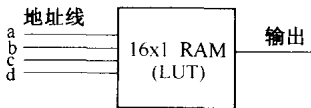
1.2.3 查找表（Look-Up-Table）的原理与结构

采用这种结构的 FPGA 有 Altera 的 FLEX、ACEX、APEX 系列和 Xilinx 的 Spartan、Virtex 系列。

查找表（Look-Up-Table）简称为 LUT，LUT 本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT，所以每一个 LUT 可以看成是一个有 4 位地址线的 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，PLD/FPGA 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运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。

表 1-1 是一个 4 输入与门的例子。

表 1-1 4 输入与门的查找表实现

实际逻辑电路		LUT 的实现方式	
			
a、b、c、d 输入	逻辑输出	地址	RAM 中存储的内容
0000	0	0000	0
0001	0	0001	0
....	0	...	0
1111	1	1111	1

1.2.4 基于查找表（LUT）的 FPGA 的结构

1. Xilinx 产品的内部结构

以 Xilinx Spartan-II 为例，其内部结构如图 1-5 所示。

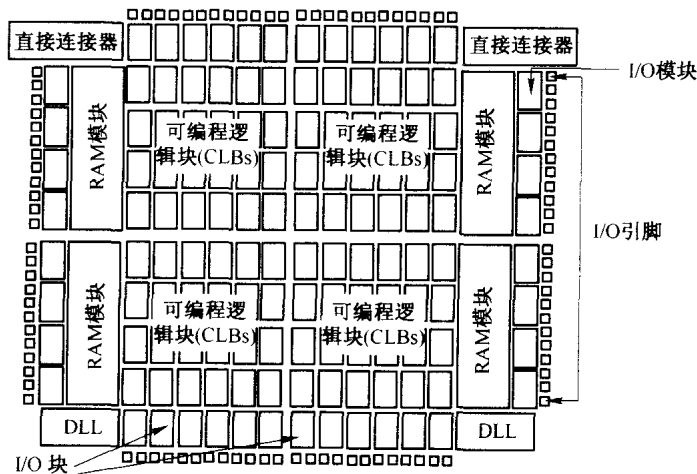


图1-5 Xilinx Spartan-II的内部结构