

高等学校电子信息学科“十二五”规划教材

基于FPGA的 数字电路实验指导书

冯建文 章复嘉 包健 李二涛 编著

西安电子科技大学出版社
<http://www.xduph.com>

高等学校电子信息学科“十二五”规划教材

基于 FPGA 的数字电路实验指导书

冯建文 章复嘉 包健 李二涛 编著

西安电子科技大学出版社

内 容 简 介

本书从数字电路与FPGA的基本概念出发,系统地介绍了数字电路实验课程所必需的FPGA开发的基础知识和开发流程,并设计了26个课程实验项目。

全书共分7章,第1章介绍了数字电路、FPGA与HDL的基本概念;第2~4章分别介绍了Verilog语言基础、Nexys3 FPGA实验板结构和FPGA开发工具软件ISE;第5~7章设计了15个基础实验、6个扩展实验和5个综合实验项目。

本书所列实验项目设计由浅入深、循序渐进,注重培养学生的综合设计和灵活应用能力。每个实验包含了实验目的、实验内容与原理、实验要求、实验步骤和思考与探索5个部分,内容详尽,要求明确,指导适度,适合作为本科院校学生的课程实验及课后自学用书,也适合Xilinx FPGA开发的初学者使用。

图书在版编目(CIP)数据

基于FPGA的数字电路实验指导书/冯建文等编著.

—西安:西安电子科技大学出版社,2016.4

高等学校电子信息学科“十二五”规划教材

ISBN 978-7-5606-4035-8

I. ①基… II. ①冯… III. ①数字电路—可编程序逻辑阵列—高等学校—教材 IV. ①TN790.2

中国版本图书馆CIP数据核字(2016)第060495号

策 划 陈 婷

责任编辑 陈 婷 马 静

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2016年4月第1版 2016年4月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 11

字 数 256千字

印 数 1~2000册

定 价 22.00元

ISBN 978-7-5606-4035-8/TN

XDUP 4327001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

数字电路是电子信息、计算机及其相关专业的一门重要的专业基础课程，主要讲述了数字逻辑的基本概念和数字电路的基本设计方法。同时它也是高年级本科生硬件课程的前导课，其后续课程有“计算机组成原理”、“计算机接口技术”、“单片机原理与应用”、“计算机系统结构”、“数字系统设计与自动化”等。作为一门经典的传统硬件基础课程，“数字电路”不仅有着坚实的系统理论，而且现代数字技术的发展也为其提供了先进的实践手段。

本实验指导书以 Verilog HDL 语言为编程工具，使用可编程逻辑器件 FPGA 来设计所有实验项目，可使学生通过实验真正掌握各种组合逻辑和时序逻辑电路的分析与设计方法，以巩固“数字电路”课程所学习的理论知识。

本书中的实验项目以美国 Digilent 公司研制的 Nexys3 教学开发板为硬件实验平台，它采用了 Xilinx 公司的 Spartan6 系列 FPGA 芯片，开发软件工具为 Xilinx 公司的 ISE Design Suite 和 Digilent 公司的 Adept 软件，编程语言为国际标准的硬件描述语言 Verilog HDL。

本书分为 7 章，内容安排如下：

第 1 章概括性地介绍了数字电路的基本概念，从实现数字电路的器件入手，引出 FPGA 器件及其相关的两种硬件描述语言——VHDL 和 Verilog HDL 语言。

第 2 章详细讲述了 Verilog HDL 语言的程序结构、语法、语句等基本编程要素，重点介绍了 Verilog HDL 的三种建模方式。

第 3 章介绍了硬件实验平台 Nexys3 教学开发板，重点介绍它拥有的各个硬件部件或者接口，以及这些部件与 FPGA 芯片的管脚连接。

第 4 章主要介绍相关的系统开发平台，包括 ISE Design Suite 和 Adept 软件的安装，并以一个具体的实例为样板，重点阐述了开发流程，即如何在 Nexys3 教学开发板上完整地设计一个实验项目。

第 5 章包含数字电路课程的基础实验项目，含组合逻辑实验 7 个，时序逻辑实验 8 个，每个实验含实验目的、实验内容与原理、实验要求、实验步骤和

思考与探索。

第6章包含6个扩展实验项目，可以作为“计算机接口技术”课程的实验项目。

第7章包含5个综合设计实验，要求明确，设计方案可灵活多变，锻炼学生的数字系统综合设计能力。

本书的主要宗旨是：由浅入深，通过各个实验项目的编程、设计和板级实验的调试、操作，培养学生的基本实验规范和技能，巩固课堂教学内容，使学生牢固地掌握数字逻辑电路的基本概念、原理和设计方法，不仅锻炼学生灵活运用理论知识解决实际问题的能力，也为后续课程的学习打下坚实的基础。

编者

2015年12月

目 录

| | | | |
|---------------------------------|----|--------------------------------------|----|
| 第 1 章 数字电路、FPGA 与 HDL | 1 | 2.6.3 系统函数 \$time 和 \$realtime | 32 |
| 1.1 数字电路 | 1 | 2.6.4 系统任务 \$finish 和 \$stop | 33 |
| 1.2 FPGA | 3 | 2.6.5 系统任务 \$readmem | 33 |
| 1.2.1 可编程逻辑器件(PLD) | 3 | 2.6.6 系统任务 \$random | 34 |
| 1.2.2 FPGA 的基本结构 | 4 | 2.7 Verilog HDL 建模方式 | 34 |
| 1.3 硬件描述语言(HDL) | 5 | 2.7.1 建模方式概述 | 34 |
| 1.3.1 VHDL | 6 | 2.7.2 结构建模方式 | 36 |
| 1.3.2 Verilog HDL | 7 | 2.7.3 数据流建模方式 | 40 |
| 第 2 章 Verilog HDL 基础 | 8 | 2.7.4 行为建模方式 | 42 |
| 2.1 Verilog HDL 概述 | 8 | 第 3 章 实验系统 | 53 |
| 2.1.1 数字电路的设计方法 | 8 | 3.1 Nexys3 教学开发板简介 | 53 |
| 2.1.2 Verilog HDL 程序结构 | 9 | 3.2 Nexys3 实验板硬件系统 | 54 |
| 2.2 Verilog HDL 的模块 | 9 | 3.2.1 主芯片 | 54 |
| 2.2.1 模块的结构 | 9 | 3.2.2 FPGA 配置电路 | 55 |
| 2.2.2 模块的声明与内容 | 10 | 3.2.3 存储模块 | 56 |
| 2.2.3 模块实例与调用 | 12 | 3.2.4 电源模块 | 57 |
| 2.2.4 时间单位与时延 | 13 | 3.2.5 时钟模块 | 58 |
| 2.3 词法约定 | 15 | 3.2.6 USB PROG 接口 | 58 |
| 2.3.1 标识符 | 15 | 3.2.7 USB A 型接口 | 59 |
| 2.3.2 关键字 | 15 | 3.2.8 USB-UART 接口(串口) | 60 |
| 2.3.3 注释 | 16 | 3.2.9 基本 I/O 设备 | 60 |
| 2.3.4 格式 | 16 | 3.2.10 以太网口 | 62 |
| 2.4 数据类型 | 16 | 3.2.11 VGA 接口 | 63 |
| 2.4.1 常量 | 17 | 3.2.12 Pmod 连接器 | 64 |
| 2.4.2 变量 | 19 | 3.2.13 VHDC 连接器 | 64 |
| 2.4.3 标量与向量 | 22 | 第 4 章 系统开发平台 | 66 |
| 2.4.4 参数 | 22 | 4.1 开发平台简介 | 66 |
| 2.5 表达式与操作符 | 23 | 4.2 ISE Design Suite | 66 |
| 2.5.1 表达式与操作数 | 23 | 4.2.1 ISE 简介 | 66 |
| 2.5.2 操作符 | 23 | 4.2.2 ISE Design Suite 功能 | 67 |
| 2.5.3 操作符优先级 | 28 | 4.2.3 ISE Design Suite 安装 | 67 |
| 2.6 系统任务和函数 | 29 | 4.3 Digilent Adept 调试工具 | 73 |
| 2.6.1 \$display 和 \$write | 29 | 4.3.1 Digilent Adept 简介 | 73 |
| 2.6.2 系统任务 \$monitor | 31 | | |

| | | | | | |
|--------------|-------------------------|-----------|--------|-------------------------|-----|
| 4.3.2 | Digilent Adept 安装..... | 73 | 5.5.1 | 实验目的 | 99 |
| 4.4 | 项目开发流程 | 75 | 5.5.2 | 实验内容与原理 | 99 |
| 4.4.1 | 创建工程 | 75 | 5.5.3 | 实验要求 | 100 |
| 4.4.2 | 编写代码 | 78 | 5.5.4 | 实验步骤 | 100 |
| 4.4.3 | 语法检查 | 80 | 5.5.5 | 思考与探索 | 101 |
| 4.4.4 | 编写测试代码 | 81 | 5.6 | 实验 6: 加法器设计 | 101 |
| 4.4.5 | 仿真 | 83 | 5.6.1 | 实验目的 | 101 |
| 4.4.6 | 逻辑综合 | 84 | 5.6.2 | 实验内容与原理 | 101 |
| 4.4.7 | 查看电路 | 85 | 5.6.3 | 实验要求 | 101 |
| 4.4.8 | 配置管脚 | 87 | 5.6.4 | 实验步骤 | 102 |
| 4.4.9 | 下载代码 | 89 | 5.6.5 | 思考与探索 | 102 |
| 4.4.10 | 板卡实验 | 91 | 5.7 | 实验 7: 加/减法运算器设计 | 102 |
| 第 5 章 | 基础实验项目 | 92 | 5.7.1 | 实验目的 | 102 |
| 5.1 | 实验 1: 五输入表决器设计 | 92 | 5.7.2 | 实验内容与原理 | 103 |
| 5.1.1 | 实验目的 | 92 | 5.7.3 | 实验要求 | 103 |
| 5.1.2 | 实验内容与原理 | 92 | 5.7.4 | 实验步骤 | 103 |
| 5.1.3 | 实验要求 | 92 | 5.7.5 | 思考与探索 | 104 |
| 5.1.4 | 实验步骤 | 93 | 5.8 | 实验 8: 基本 RS 触发器设计 | 105 |
| 5.1.5 | 思考与探索 | 93 | 5.8.1 | 实验目的 | 105 |
| 5.2 | 实验 2: 多路数据选择器设计 | 94 | 5.8.2 | 实验内容与原理 | 105 |
| 5.2.1 | 实验目的 | 94 | 5.8.3 | 实验要求 | 105 |
| 5.2.2 | 实验内容与原理 | 94 | 5.8.4 | 实验步骤 | 106 |
| 5.2.3 | 实验要求 | 94 | 5.8.5 | 思考与探索 | 106 |
| 5.2.4 | 实验步骤 | 94 | 5.9 | 实验 9: D 触发器设计 | 106 |
| 5.2.5 | 思考与探索 | 95 | 5.9.1 | 实验目的 | 106 |
| 5.3 | 实验 3: 译码器设计 | 95 | 5.9.2 | 实验内容与原理 | 106 |
| 5.3.1 | 实验目的 | 95 | 5.9.3 | 实验要求 | 107 |
| 5.3.2 | 实验内容与原理 | 95 | 5.9.4 | 实验步骤 | 107 |
| 5.3.3 | 实验要求 | 96 | 5.9.5 | 思考与探索 | 108 |
| 5.3.4 | 实验步骤 | 96 | 5.10 | 实验 10: JK 触发器设计 | 108 |
| 5.3.5 | 思考与探索 | 97 | 5.10.1 | 实验目的 | 108 |
| 5.4 | 实验 4: 二进制优先级编码器设计 | 97 | 5.10.2 | 实验内容与原理 | 108 |
| 5.4.1 | 实验目的 | 97 | 5.10.3 | 实验要求 | 109 |
| 5.4.2 | 实验内容与原理 | 97 | 5.10.4 | 实验步骤 | 109 |
| 5.4.3 | 实验要求 | 98 | 5.10.5 | 思考与探索 | 109 |
| 5.4.4 | 实验步骤 | 99 | 5.11 | 实验 11: 同步二进制计数器设计 | 110 |
| 5.4.5 | 思考与探索 | 99 | 5.11.1 | 实验目的 | 110 |
| 5.5 | 实验 5: 数值比较器设计 | 99 | 5.11.2 | 实验内容与原理 | 110 |
| | | | 5.11.3 | 实验要求 | 111 |

| | | | |
|---------------------------|------------|----------------------------------|------------|
| 5.11.4 实验步骤..... | 111 | 6.2.5 思考与探索..... | 126 |
| 5.11.5 思考与探索..... | 111 | 6.3 实验 18: 数码管扫描显示..... | 126 |
| 5.12 实验 12: 十进制计数器设计..... | 112 | 6.3.1 实验目的..... | 126 |
| 5.12.1 实验目的..... | 112 | 6.3.2 实验内容与原理..... | 126 |
| 5.12.2 实验内容与原理..... | 112 | 6.3.3 实验要求..... | 129 |
| 5.12.3 实验要求..... | 112 | 6.3.4 实验步骤..... | 130 |
| 5.12.4 实验步骤..... | 113 | 6.3.5 思考与探索..... | 130 |
| 5.12.5 思考与探索..... | 113 | 6.4 实验 19: 数字钟设计..... | 130 |
| 5.13 实验 13: 可逆计数器设计..... | 113 | 6.4.1 实验目的..... | 130 |
| 5.13.1 实验目的..... | 113 | 6.4.2 实验内容与原理..... | 131 |
| 5.13.2 实验内容与原理..... | 113 | 6.4.3 实验要求..... | 131 |
| 5.13.3 实验要求..... | 114 | 6.4.4 实验步骤..... | 131 |
| 5.13.4 实验步骤..... | 114 | 6.4.5 思考与探索..... | 132 |
| 5.13.5 思考与探索..... | 115 | 6.5 实验 20: 小键盘扫描实验..... | 132 |
| 5.14 实验 14: 基本寄存器设计..... | 115 | 6.5.1 实验目的..... | 132 |
| 5.14.1 实验目的..... | 115 | 6.5.2 实验内容与原理..... | 132 |
| 5.14.2 实验内容与原理..... | 115 | 6.5.3 实验要求..... | 135 |
| 5.14.3 实验要求..... | 116 | 6.5.4 实验步骤..... | 136 |
| 5.14.4 实验步骤..... | 116 | 6.5.5 思考与探索..... | 136 |
| 5.14.5 思考与探索..... | 116 | 6.6 实验 21: 串口通信实验..... | 137 |
| 5.15 实验 15: 移位寄存器设计..... | 117 | 6.6.1 实验目的..... | 137 |
| 5.15.1 实验目的..... | 117 | 6.6.2 实验内容与原理..... | 137 |
| 5.15.2 实验内容与原理..... | 117 | 6.6.3 实验要求..... | 144 |
| 5.15.3 实验要求..... | 118 | 6.6.4 实验步骤..... | 145 |
| 5.15.4 实验步骤..... | 118 | 6.6.5 思考与探索..... | 146 |
| 5.15.5 思考与探索..... | 119 | | |
| 第 6 章 扩展实验项目..... | 120 | 第 7 章 综合实验项目..... | 147 |
| 6.1 实验 16: 定时与分频实验..... | 120 | 7.1 实验 22: 寄存器堆设计..... | 147 |
| 6.1.1 实验目的..... | 120 | 7.1.1 实验目的..... | 147 |
| 6.1.2 实验内容与原理..... | 120 | 7.1.2 实验内容与原理..... | 147 |
| 6.1.3 实验要求..... | 122 | 7.1.3 实验要求..... | 148 |
| 6.1.4 实验步骤..... | 122 | 7.1.4 实验步骤..... | 148 |
| 6.1.5 思考与探索..... | 122 | 7.1.5 思考与探索..... | 148 |
| 6.2 实验 17: 流水灯实验..... | 123 | 7.2 实验 23: 按键消抖与单脉冲 产生实验..... | 149 |
| 6.2.1 实验目的..... | 123 | 7.2.1 实验目的..... | 149 |
| 6.2.2 实验内容与原理..... | 123 | 7.2.1 实验内容与原理..... | 149 |
| 6.2.3 实验要求..... | 125 | 7.2.3 实验要求..... | 151 |
| 6.2.4 实验步骤..... | 125 | 7.2.4 实验步骤..... | 151 |

| | |
|---------------------------|-----|
| 7.2.5 思考与探索 | 152 |
| 7.3 实验 24: 出租车计费器设计 | 152 |
| 7.3.1 实验目的 | 152 |
| 7.3.2 实验内容与原理 | 152 |
| 7.3.3 实验要求 | 153 |
| 7.3.4 实验步骤 | 153 |
| 7.3.5 思考与探索 | 153 |
| 7.4 实验 25: 交通灯设计 | 154 |
| 7.4.1 实验目的 | 154 |
| 7.4.2 实验内容与原理 | 154 |
| 7.4.3 实验要求 | 155 |
| 7.4.4 实验步骤 | 155 |
| 7.4.5 思考与探索 | 156 |
| 7.5 实验 26: 自动售货机设计 | 156 |
| 7.5.1 实验目的 | 156 |
| 7.5.2 实验内容与原理 | 156 |
| 7.5.3 实验要求 | 157 |

| | |
|-------------------|-----|
| 7.5.4 实验步骤 | 158 |
| 7.5.5 思考与探索 | 158 |

| | |
|-----------------------------|-----|
| 附录 A Nexys4-DDR 实验板卡 | 159 |
| A.1 Nexys4-DDR 实验板卡概况 | 159 |
| A.2 Nexys4-DDR 部件 | 160 |
| A.2.1 电源供电 | 160 |
| A.2.2 FPGA 配置 | 161 |
| A.2.3 基本 I/O 设备 | 161 |
| A.2.4 USB-UART 接口 | 162 |
| A.2.5 Pmod 连接器 | 162 |
| A.2.6 VGA 接口 | 163 |
| A.2.7 振荡器/时钟 | 164 |
| A.2.8 以太网接口 | 164 |
| A.2.9 温度传感器 | 164 |
| 附录 B 实验报告模板 | 165 |
| 参考文献 | 168 |

第1章 数字电路、FPGA 与 HDL

本章首先介绍数字电路相关的基本概念、发展历程及电路元件，然后引出最为常用的可编程逻辑器件 FPGA，最后对 FPGA 的硬件设计描述语言 HDL 做概括性说明。

1.1 数字电路

用数字信号对数字量进行算术运算和逻辑运算的电路称为数字电路或数字系统。数字电路采用二进制码制，1 和 0 两种数码正好对应于两种逻辑状态——真和假，因此可以方便地进行逻辑运算和逻辑处理。逻辑运算是数字电路最基本的运算形式，因此数字电路也称为数字逻辑电路。

数字电路中，实现基本逻辑运算的电路称为门电路，如用于实现“与”运算的与门、实现“或”运算的或门等。采用少量的几个门电路还可以很容易地构成一种能够存储并记忆 1 位二进制信息的逻辑部件，称为触发器。门电路和触发器是数字电路最基本的电路单元。

数字电路的发展经历了电子管、晶体管分立器件电路以及现在广泛应用的半导体集成电路。

早期的门电路是用导线将晶体管、电阻等独立的元件连接形成的，称为分立元件门电路。20 世纪 60 年代，出现了以双极型工艺制成的小规模数字集成逻辑器件，它将构成门电路或触发器的电子电路制作于一块半导体芯片中，形成了目前广泛应用的集成电路(IC, Integrated Circuit)。

集成电路的一个重要指标是集成度。集成度指每块芯片或芯片每单位面积中包含的晶体管的数量，通常用于表示集成电路的规模。集成门电路和集成触发器的结构简单，只包含几个少量的晶体管，集成度低，规模小，因此被称为小规模集成电路(SSIC, Small Scale Integrated Circuit)。此后，随着集成电路技术的迅速发展，更多的晶体管，甚至是整个电路都能够被制作于一块芯片中，形成了规模更大的集成电路。按照摩尔定律，芯片的集成度每 18 个月就能提高一倍。目前，数十亿级别的晶体管处理器已经得到商用。

按照集成度的高低，数字电路分为小规模集成电路(SSIC)、中规模集成电路(MSIC, Middle Scale Integrated Circuit)、大规模集成电路(LSIC, Large Scale Integrated Circuit)、超大规模集成电路(VLSIC, Very Large Scale Integrated Circuit)、甚大规模集成电路(ULSIC, Ultra Large Scale Integrated Circuit)和巨大规模集成电路(GSIC, Giga Scale Integration Circuit)等类别。

按照制造工艺, 数字电路又可分为 TTL 型和 CMOS 型。TTL 电路采用双极型晶体管作为主要电子器件, 问世较早, 在长期的使用过程中逐渐演化为一种电路标准。CMOS 电路采用 NMOS 和 PMOS 两种互补的金属-氧化物-半导体场效应晶体管作为主要电子器件, 具有显著的低功耗、高密度等特性, 使 CMOS 电路逐渐取代了 TTL 电路的主导地位, 发展成为主流的电路形式。

经过半个多世纪的发展, 数字电路的应用几乎延伸到了所有的领域。

近年来, 可编程逻辑器件(PLD), 特别是现场可编程门阵列(FPGA)的飞速发展, 使数字电子技术开创了新局面。这些器件不仅规模大, 而且将硬件与软件相结合, 使其功能更加完善, 使用更灵活。

设计开发数字电路, 通常可以选择以下 3 类芯片。

1) 标准芯片

标准芯片指具有通用、固定逻辑功能的集成电路器件, 如集成门电路、集成触发器、加法器、译码器、数据选择器、计数器等。这一类集成电路有大量的产品可供选择使用, 虽然具体产品可能来自于不同的厂商, 但一般都遵循统一的命名规则, 相同编号的芯片具有相同的逻辑功能和引脚排列。标准芯片集成度低, 逻辑功能简单、固定。20 世纪 80 年代之前, 一般采用标准芯片设计数字电路, 其缺点是: 所需要的芯片个数多, 占用电路板体积大, 功耗大, 可靠性差, 难于实现复杂的逻辑功能; 逻辑功能固定, 一旦完成设计, 就很难再进行更改。

2) 可编程逻辑器件(PLD)

可编程逻辑器件(PLD, Programmable Logic Device)是 20 世纪 70 年代开始发展起来的一类集成电路器件。与标准芯片类似, PLD 具有通用的逻辑结构, 可以按通用的集成电路器件进行批量生产。不同的是, PLD 内部包含大量的可编程开关, 用户编程配置这些开关为不同的状态, 就能实现不同的逻辑功能; 而且这样的编程配置过程可以由最终的用户借助编程工具实现, 而不必由芯片制造厂商来完成。

PLD 的优点有:

(1) PLD 作为通用芯片, 可批量生产, 成本低, 但又可通过编程配置设计不同的电路来实现专用集成电路(ASIC, Application Specific Integrated Circuit)的功能。

(2) 大多数的 PLD 器件允许多次编程, 便于系统的修改、升级和维护。

(3) PLD 的集成度高, 与标准芯片相比, 可以实现更复杂的逻辑电路, 且具有功耗低、体积小、可靠性高等优点。

基于以上优势, PLD 器件获得了广泛的应用, 成为设计数字系统的主流器件。

3) 定制芯片

使用 PLD 设计数字系统, 能够满足大多数应用的需求。但其内部的可编程开关在带来可编程定制、便于修改升级等优势的同时, 也带来了一些缺点, 如可编程开关耗费了芯片空间, 限制了可实现电路的规模, 增加了器件的成本, 降低了速度性能, 增大了功耗。

在一些对集成度、速度、功耗等性能要求较高的系统中, 可以将设计好的电路交付给半导体器件制造厂商, 由厂商选择合适的技术来生产满足特定性能指标的芯片。这样的芯片可依据用户的要求定制生产, 因此称为定制芯片。由于生产的芯片主要用于一些特定的

应用场合，因此也称为专用集成电路(ASIC)。

定制芯片的主要优点在于：针对特定的应用需求生产，能够根据特定的任务进行优化。相对于标准芯片和 PLD，定制芯片具有更好的性能，能够实现更大规模的电路。定制芯片的缺点是：设计和开发周期长，产品投放市场时间长；生产过程中可能要经过多次反复的尝试，成本高，风险大。为降低成本，通常需要生产足够的数量，以降低每片芯片的平均价格。

1.2 FPGA

1.2.1 可编程逻辑器件(PLD)

可编程逻辑器件(PLD)一般是由输入缓冲电路、与阵列、或阵列、输出缓冲电路 4 部分构成的，如图 1.1 所示。

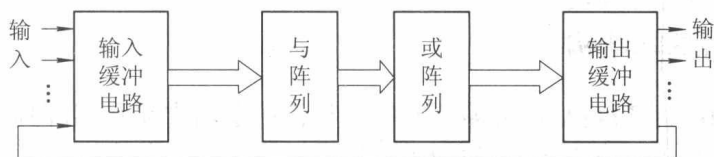


图 1.1 PLD 结构框图

PLD 按照集成度可以分为简单 PLD 和复杂 PLD。

简单 PLD 又称为低密度 PLD(低于 700 门/片)，常见的有 4 种：可编程只读存储器(PROM, Programmable ROM)、现场可编程逻辑阵列(FPLA, Field Programmable Logic Array)、可编程阵列逻辑(PAL, Programmable Array Logic)和通用可编程阵列逻辑(GAL, General Array Logic)。它们的电路结构特征如表 1.1 所示。

表 1.1 简单 PLD 的电路特征

| 类型 | 与阵列 | 或阵列 | 输出方式 |
|------|-----|-----|------|
| PROM | 固定 | 可编程 | 固定 |
| FPLA | 可编程 | 可编程 | 固定 |
| PAL | 可编程 | 固定 | 固定 |
| GAL | 可编程 | 固定 | 可组态 |

复杂 PLD 有现场可编程门阵列(FPGA, Field Programmable Gate Array)和复杂可编程逻辑器件(CPLD, Complex Programmable Logic Device)，它们都是从 PAL、GAL 发展起来的阵列型高密度 PLD 器件。

CPLD 主要是由可编程逻辑宏单元(LMC, Logic Macro Cell)通过互连组成的，内部大多数采用了 CMOS、EPROM、EEPROM 和快速闪存等编程技术，因此有的可编程 1 次，有的可反复编程，有的可实现在线编程，有的则需要编程器进行编程。CPLD 比 FPGA 速度快，性能更可靠，有较好的可预测性，但集成度比 FPGA 低。

目前应用最广泛的复杂 PLD 器件是 FPGA，它是作为专用集成电路(ASIC)领域中的一种半定制电路而出现的，既解决了定制电路的不足，又克服了原有可编程器件门电路数有限的缺点。其集成度可达到千万级以上晶体管/片，甚至可以将整个系统制作于一个芯片内。

FPGA 内部一般采用 SRAM 结构，可实现在线编程。

FPGA 和 CPLD 的特点对比如表 1.2 所示。

表 1.2 FPGA 和 CPLD 的特点

| 比较项目 | CPLD | FPGA |
|-----------|--------------------------|--------------------------------------|
| 组合逻辑的实现方法 | 乘积项, 查找表 | 查找表 |
| 编程元素 | Flash, EEPROM | SRAM |
| 特点 | 非易失性 立即上电 可在单芯片上运行 | 需要外部配置 ROM 内建高性能硬宏功能 高集成度, 高性能 |
| 应用范围 | 偏向于简单的控制通道应用 以及组合逻辑 | 偏向于较复杂且高速的控制 通道应用以及数据处理 |
| 集成度 | 小规模至中规模 | 中规模至大规模 |

1.2.2 FPGA 的基本结构

现场可编程门阵列 FPGA 是可编程器件，与传统逻辑电路和门阵列(如 PAL、GAL 及 CPLD 器件)相比，FPGA 具有不同的结构。

FPGA 采用了逻辑单元阵列(LCA, Logic Cell Array)结构，内部包括可配置逻辑模块(CLB, Configurable Logic Block)、可编程输入/输出模块(IOB, Input Output Block)和可编程内部连接(PIC, Programmable Interconnect)三个部分。FPGA 利用小型查找表(16 × 1 RAM)来实现组合逻辑，每个查找表连接到一个 D 触发器的输入端，触发器再来驱动其他逻辑电路或驱动 I/O，由此构成了既可实现组合逻辑功能又可实现时序逻辑功能的基本逻辑单元模块，这些模块利用金属连线互相连接或连接到 I/O 模块。FPGA 内部结构如图 1.2 所示。

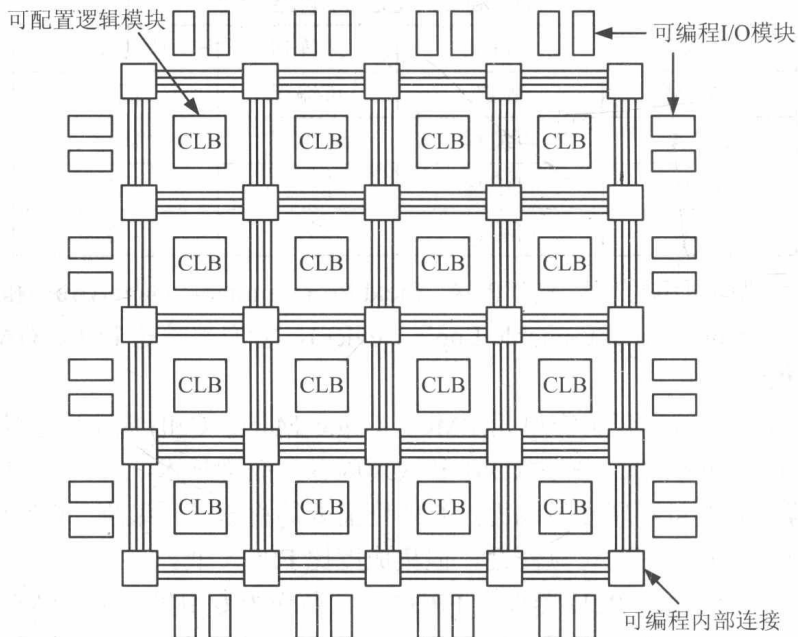


图 1.2 FPGA 内部结构图

1) 可配置逻辑模块(CLB)

CLB 是 FPGA 的基本逻辑单元。CLB 的实际数量和特性会根据器件的不同而不同,但是每个 CLB 都由一个可配置开关矩阵、若干多路选择器和触发器组成。开关矩阵一般具有 4~6 个输入,是高度灵活的,可以对其进行配置以便处理组合逻辑、移位寄存器或 RAM。在 Xilinx 公司的 FPGA 器件中,CLB 由多个(一般为 4 个或 2 个)相同的 Slice 和附加逻辑构成。每个 CLB 模块不仅可以用于实现组合逻辑、时序逻辑,还可以配置为分布式 RAM 和分布式 ROM。

2) 可编程输入/输出模块(IOB)

可编程输入/输出模块简称 I/O 单元,是芯片与外界电路的接口部分,可满足不同电气特性下对输入/输出信号的驱动与匹配要求。FPGA 内的 I/O 按组分类,每组都能够独立地支持不同的 I/O 标准。通过软件的灵活配置,IOB 可适配不同的电气标准与 I/O 物理特性,可以调整驱动电流的大小,可以改变上、下拉电阻。每个 IOB 可以编程为三种不同的功能:输入、输出和双向。目前,I/O 口的工作频率也越来越高,一些高端的 FPGA 通过 DDR 寄存器技术可以支持高达 2 Gb/s 的数据速率。

3) 可编程内部连接(PIC)

可编程内部连接主要由金属线组成,它分布在 CLB 阵列周围,通过 SRAM 控制可编程开关,实现系统逻辑的布线。

目前,FPGA 芯片的主要生产厂商有 Xilinx、Altera 和 Actel 公司,其中全球最大的生产厂商是 Xilinx。对 Xilinx FPGA 芯片进行设计开发,要使用 Xilinx 公司的开发软件平台——集成软件环境(ISE, Integrated Software Environment),在其中用硬件描述语言 VHDL 或者 Verilog 语言编程,具体安装、使用方法及 FPGA 开发流程见本书第 4 章。

1.3 硬件描述语言(HDL)

大规模可编程逻辑器件 PLD 的出现,引起了数字系统设计领域的革命性变革。传统的基于标准芯片的“搭积木”式的数字系统设计技术正成为历史,而基于可编程逻辑器件、硬件描述语言和计算机开发平台的 EDA 技术则蓬勃发展并日趋完善。EDA 技术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,通过有关的开发软件,自动实现用软件方式设计电子系统和硬件系统。现代 EDA 技术的主要特征是:引入硬件描述语言(HDL),支持高层次的抽象设计,具有逻辑综合、行为综合、系统综合的能力。

HDL 以文本形式来描述数字系统硬件结构和行为,是一种用形式化方法来描述数字电路和系统的语言,可以从上层到下层来逐层描述自己的设计思想,即用一系列分层次的模块来表示复杂的数字系统,并逐层进行验证仿真,再把具体的模块组合由综合工具转化成门级网表,接下去再利用布局布线工具把网表转化为具体电路结构。目前,这种自顶向下的设计方法已被广泛使用。概括地讲,HDL 具有以下主要特征:

(1) HDL 既包含一些高级程序设计语言的结构形式,同时也兼顾描述硬件线路连接的

具体结构。

(2) 通过使用结构级行为描述,可以在不同的抽象层次描述设计。HDL 采用自顶向下的数字电路设计方法,主要包括 3 个领域 5 个抽象层次。

(3) HDL 是并行处理的,具有同一时刻执行多任务的能力。这和一般高级设计语言(如 C 语言等)串行执行的特征是不同的。

(4) HDL 具有时序的概念。一般的高级编程语言是没有时序概念的,但在硬件电路中从输入到输出总是有延时存在的,为了描述这一特征,需要引入时延的概念。HDL 不仅可以描述硬件电路的功能,还可以描述电路的时序。

随着 EDA 技术的发展,使用硬件语言设计 PLD/FPGA 已成为一种趋势。目前最主要的硬件描述语言是(VHDL, Very-High-Speed Integrated Circuit Hardware Description Language)和 Verilog HDL。

VHDL 发展得较早,语法严格,而 Verilog HDL 是在 C 语言的基础上发展起来的一种硬件描述语言,语法较自由。VHDL 和 Verilog HDL 两者相比,VHDL 的书写规则比 Verilog 繁琐一些,但 Verilog 自由的语法也容易让少数初学者出错。

从 EDA 技术的发展上看,已出现用于 CPLD/FPGA 设计的硬件 C 语言编译软件,虽然还不成熟,应用极少,但它有可能会成为继 VHDL 和 Verilog 之后,设计大规模 CPLD/FPGA 的又一种手段。

1.3.1 VHDL

VHDL 诞生于 1982 年。1987 年年底,VHDL 被 IEEE 和美国国防部确认为标准硬件描述语言。自 IEEE 公布了 VHDL 的标准版本 IEEE-1076(简称 87 版)之后,各 EDA 公司相继推出了自己的 VHDL 设计环境,或宣布自己的设计工具可以和 VHDL 接口。此后,VHDL 在电子设计领域得到了广泛的应用,并逐步取代了原有的非标准的硬件描述语言。1993 年,IEEE 对 VHDL 进行了修订,从更高的抽象层次和系统描述能力上扩展 VHDL 的内容,公布了新版本的 VHDL,即 IEEE 标准的 1076-1993 版本(简称 93 版)。现在,VHDL 和 Verilog 作为 IEEE 的工业标准硬件描述语言,又得到众多 EDA 公司的支持,在电子工程领域,已成为名副其实的通用硬件描述语言。有专家认为,在 21 世纪中,VHDL 与 Verilog 语言将承担起大部分的数字系统设计任务。

VHDL 主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外,VHDL 的语言形式、描述风格和句法类似于一般的计算机高级语言。VHDL 的程序结构特点是将一项工程设计(或称设计实体,可以是一个元件、一个电路模块或一个系统)分成外部和内部。外部又称可视部分或端口;内部又称不可视部分,它涉及实体的内部功能和算法实现部分。在对一个设计实体定义了外部界面后,一旦其内部开发完成后,其他的设计就可以直接调用这个实体。这种将设计实体分成内外部分的概念是 VHDL 系统设计的基本点。

应用 VHDL 进行工程设计的优点是多方面的,具体如下:

(1) 与其他的硬件描述语言相比,VHDL 具有更强的行为描述能力,从而决定了它成为系统设计领域最佳的硬件描述语言。强大的行为描述能力是避开具体的器件结构,从逻辑行为上描述和设计大规模电子系统的重要保证。

(2) VHDL 具有丰富的仿真语句和库函数,使得在任何大系统的设计早期就能查验设计系统的功能可行性,随时可对设计进行仿真模拟。

(3) VHDL 语句的行为描述能力和程序结构决定了它具有支持大规模设计的分解和已有设计的再利用功能。大规模系统高效、高速的完成必须有多人甚至多个研发组共同并行工作才能实现, VHDL 的特性符合市场的需求。

(4) 对于用 VHDL 完成的一个确定的设计,可以利用 EDA 工具进行逻辑综合和优化,并自动地把 VHDL 描述设计转变成门级网表。

(5) VHDL 对设计的描述具有相对独立性,设计者可以不了解硬件的结构,也不必知道最终设计实现的目标器件是什么,而进行独立的设计。

1.3.2 Verilog HDL

Verilog HDL 是在应用最广泛的 C 语言的基础上发展起来的一种硬件描述语言,它是由 GDA(Gateway Design Automation)公司的 Phil Moorby 在 1983 年末首创的,最初只设计了一个仿真与验证工具,之后又陆续开发了相关的故障模拟与时序分析工具。1985 年 Moorby 推出它的第三个商用仿真器 Verilog-XL,并获得了巨大的成功,从而使得 Verilog HDL 迅速得到推广应用。1989 年 CADENCE 公司收购了 GDA 公司,使得 Verilog HDL 成为了该公司的独家专利。1990 年 CADENCE 公司公开发表了 Verilog HDL,并成立 OVI(Open Verilog International)组织以促进 Verilog HDL 的发展。1995 年,Verilog HDL 成为 IEEE 标准,即 IEEE Standard 1364-1995。1998 年,模拟和数字都适用的 Verilog HDL 标准公开发表。目前,最新的 Verilog 语言版本是 2001 年 IEEE 公布的 Verilog 2001 标准,其大幅度提高了系统级和可综合性能。

Verilog HDL 的主要优点是:简洁、高效、功能强、易学易用,其语法与 C 语言有许多相似之处。其特点如下:

- (1) 既能进行面向综合的电路设计,也可用于电路的模拟仿真。
- (2) 设计能够在多个层次上加以描述,从开关级、门级、寄存器传送级(RTL)到算法级,包括进程和队列级。
- (3) 设计的规模可以是任意的,该语言不对设计的规模(大小)施加任何限制。
- (4) 有三种电路描述与建模方式:行为描述方式——使用过程化结构建模;数据流描述方式——使用连续赋值语句方式建模;结构描述方式——使用门和模块实例语句描述建模。
- (5) 具有混合方式建模能力,即在一个设计中每个模块均可以在不同设计层次上建模。
- (6) 可使用内置基本逻辑门,在门级对设计完整建模。
- (7) 可使用内置开关级原语,在开关级对设计完整建模。
- (8) 用户可灵活创建自定义原语(UDP)。
- (9) 允许通过使用编程语言接口(PLI)与外部函数交互。

第 2 章 Verilog HDL 基础

2.1 Verilog HDL 概述

硬件描述语言 Verilog HDL 用于从算法级、门级到开关级的多种抽象设计层次的数字系统建模。被建模的数字系统对象的复杂性可以介于简单的门和完整的电子数字系统之间。数字系统能够按层次描述，并可在相同描述中显式地进行时序建模。

Verilog HDL 语言具有下述描述能力：设计的行为特性、设计的数据流特性、设计的结构组成以及包含响应监控和设计验证方面的时延和波形产生机制，所有这些都使用同一种建模语言。此外，Verilog HDL 语言提供了编程语言接口，通过该接口可以在模拟、验证期间设计从外部访问，包括模拟的具体控制和运行。

Verilog HDL 语言不仅定义了语法，而且对每个语法结构都定义了清晰的模拟、仿真语义。因此，用这种语言编写的模型能够使用 Verilog 仿真器进行验证。Verilog HDL 语言从 C 编程语言中继承了多种操作符和结构，另外还提供了扩展的建模能力。Verilog HDL 语言的核心子集非常易于学习和使用，这对大多数建模应用来说已经足够。同时，完整的硬件描述语言足以对从最复杂的芯片到完整的电子系统进行描述。

2.1.1 数字电路的设计方法

Verilog HDL 既是一种行为描述语言，也是一种结构描述语言。如果按照一定的规则和风格编写代码，就可以将功能行为模块通过工具自动转化为门级互连的结构模块。这意味着利用 Verilog 语言所提供的功能，就可以构造一个模块间的清晰结构来描述复杂的大型设计，并对所需的逻辑电路进行严格的设计。下面就数字电路的设计方法加以讨论。

1) 自下而上的设计方法

自下而上的设计是传统的设计方法，是从基本单元出发，对设计进行逐层划分的过程。这种设计与用电子元件在模拟实验板上建立一个系统的步骤非常相似。这种方法的优点是：设计人员对这种设计方法比较熟悉，实现各个子模块所需的时间较短。缺点是：对系统的整体功能把握不足；由于必须先对多个子模块进行设计，因此实现整个系统的功能所需的时间长；另外，对设计人员之间相互协作也有较高的要求。

2) 自上而下的设计方法

自上而下的设计是从系统级开始，把系统划分为基本单元，然后再把基本单元划分为下一层次的基本单元，直到可用 EDA 元件实现为止。

这种方法的优点是：在设计周期开始就做好了系统分析；由于设计的主要仿真和调试