

电子设计 自动化技术 (Verilog HDL版)

张平华 黄秀亮 徐红丽 肖成 孙小进 ©编著

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

项目1 三人表决器设计

【项目描述】

【知识准备】

1. 1 EDA技术及其发展

1. 1. 1 EDA技术概念

1. 1. 2 EDA技术的主要特征

1. 1. 3 EDA技术发展历程

1. 2 可编程逻辑器件

1. 2. 1 可编程逻辑器件概述

1. 2. 1. 1 可编程逻辑器件的历史演变

1. 2. 1. 2 可编程逻辑器件的基本结构与分类

1. 2. 1. 3 PLD相对于MCU的优势所在

1. 2. 2 FPGA和CPLD

1. 2. 2. 1 基于乘积项的PLD结构

1. 2. 2. 2 乘积项结构PLD的逻辑实现原理

1. 2. 2. 3 查找表的原理与结构

1. 2. 2. 4 基于查找表的FPGA结构

1. 2. 2. 5 查找表结构的FPGA逻辑实现原理

1. 2. 2. 6 其他类型的FPGA和PLD

1. 2. 3 可编程逻辑器件的应用

1. 2. 3. 1 PLD在专用集成电路设计中的应用

1. 2. 3. 2 基于EDA工具的PLD应用

1. 2. 3. 3 FPGA和CPLD开发应用选择

1. 3 面向CPLD／FPGA的EDA设计流程

1. 3. 1 源程序的编辑与编译

1. 3. 2 逻辑综合和优化

1. 3. 3 目标器件的布线／适配

1. 3. 4 目标器件的编程／下载

1. 3. 5 设计过程中的有关仿真

- 1. 3. 6 硬件仿真／硬件测试
- 1. 4 基于CPLD／FPGA的常用EDA工具
 - 1. 4. 1 AJtera EDA软件工具Quartus II 简介
 - 1. 4. 2 Quanus II 主要设计流程
 - 1. 4. 3 Quartus II 详细设计流程
 - 1. 4. 4 其他CPLD／FPGA常用工具
- 1. 5 硬件描述语言
 - 1. 5. 1 常用硬件描述语言简介
 - 1. 5. 1. 1 Verilog概述
 - 1. 5. 1. 2 Verilog与c语言
 - 1. 5. 1. 3 其他常用硬件描述语言
 - 1. 5. 1. 4 Verilog程序设计举例
 - 1. 5. 2 Verilog基本语句结构与语法知识
 - 1. 5. 2. 1 模块表达
 - 1. 5. 2. 2 端口语句、端口信号名和端口模式
 - 1. 5. 2. 3 内部信号说明
 - 1. 5. 2. 4 功能定义
 - 1. 5. 2. 5 关键字
 - 1. 5. 2. 6 标识符
 - 1. 5. 2. 7 注释符号
 - 1. 5. 2. 8 规范的程序书写格式
 - 1. 5. 2. 9 文件取名和存盘

【任务实现】

任务1. 1 Quartus II 软件安装

任务1. 2 三人表决器设计

【考核评价】

【拓展提高】

项目2 四位加法器设计

【项目描述】

【知识准备】

2.1 四位加法器原理

2.1.1 半加器逻辑功能

2.1.2 四位加法器原理图

2.2 Quanus II 原理图输入设计方法

【任务实现】

任务2.1 利用原理图输入法设计3—8译码器

任务2.2 利用原理图输入法设计四位加法器

【考核评价】

【拓展提高】

项目3 数字电子钟设计

【项目描述】

【知识准备】

3.1 Verilog语言要素

3.1.1 Verilog文字规则

3.1.2 Verilog数据类型

3.1.3 Verilog操作符

3.2 Verilog语句

3.2.1 赋值语句和块语句

3.2.1.1 赋值语句

3.2.1.2 块语句

3.2.2 条件语句

3.2.2.1 if_else语句

3.2.2.2 case语句

3.2.2.3 条件语句使用不当

3.2.3 循环语句

3.2.3.1 forever语句

3. 2. 3. 2 repeat语句

3. 2. 3. 3 while语句

3. 2. 3. 4 for语句

3. 2. 4 结构描述语句

【任务实现】

任务3. 1 数字电子钟设计方案分析

任务3. 2 数字电子钟的Verilog程序设计

【考核评价】

【拓展提高】

项目4 交通管理器设计

【项目描述】

【知识准备】

4. 1 Verilog与原理图混合设计方式

4. 1. 1 四位二进制计数器的Verilog设计

4. 1. 2 八段显示译码器的Verilog设计

4. 1. 3 顶层文件原理图设计

【任务实现】

任务4. 1 交通管理器设计方案分析

任务4. 2 交通管理器设计

【考核评价】

【拓展提高】

项目5 简易电子琴设计

【项目描述】

【知识准备】

5. 1 结构语句

5. 1. 1 结构说明语句

5. 1. 1. 1 initial语句

- 5. 1. 1. 2 always语句
- 5. 1. 2 task和function说明语句
- 5. 2 Verilog设计中LPM函数的应用
- 5. 2. 1 常用LPM兆功能块
- 5. 2. 2 通过MegaWizardPlug-In Manager例化LPM
- 5. 2. 2. 1 计数器LPM模块文本文件的调用
- 5. 2. 2. 2 LPM计数器程序与参数传递语句
- 5. 2. 2. 3 创建工程与仿真测试

【任务实现】

任务5. 1 简易电子琴设计方案分析

任务5. 2 简易电子琴Verilog程序设计

【考核评价】

【拓展提高】

项目6 步进电动机控制器设计

【项目描述】

【知识准备】

- 6. 1 有限状态机设计
- 6. 1. 1 概述
- 6. 1. 2 一般有限状态机的设计
- 6. 1. 3 Moore型状态机的设计
- 6. 1. 4 Mealy型有限状态机的设计
- 6. 2 步进电动机控制技术
- 6. 2. 1 步进电动机的工作原理
- 6. 2. 2 步进电动机的工作方式

【任务实现】

任务6. 1 步进电动机控制器设计方案分析

任务6. 2 步进电动机控制器设计

【考核评价】

【拓展提高】

项目7 数字温度计设计

【项目描述】

【知识准备】

7.1 温度传感器

7.2 单总线器件

7.2.1 单总线的工作原理

7.2.2 单总线器件概述

7.2.2.1 DS18B20简介

7.2.2.2 DS18B20的主要特性

7.2.2.3 DS18B20的内部结构

7.2.2.4 DS18B20的工作原理

7.2.2.5 DS18B20的4个主要数据部件

7.2.2.6 高速暂存存储器

7.2.2.7 控制命令

【任务实现】

任务7.1 数字温度计设计方案分析

任务7.2 数字温度计设计

【考核评价】

【拓展提高】

附录 本书所使用开发板功能简介

附录A 开发板硬件介绍

A.1 硬件资源图示

A.2 核心板资源

附录B 开发板硬件原理图

B.1 核心板管脚示意图

B.2 接口板原理图

项目 1

三人表决器设计

【项目描述】

三人表决器完成的逻辑功能：当同意人数为多数时（两人或两人以上），表决结果为通过；当不同意人数为多数时（两人或两人以上），表决结果为不通过。

电路功能描述如下：三个人分别用拨码开关 KD1、KD2、KD3 来表示自己的意愿，如果对某决议同意，将对应的拨码开关拨到高电平（上方），若不同意就把对应的拨码开关拨到低电平（下方）。表决结果用 LED 显示，如果决议通过那么实验板上 LED 点亮；如果不通过那么实验板上 LED 不亮；如果对某项决议有任意两到三人同意，那么此决议通过，LED 点亮；如果对某项决议只有一人或没人同意，那么此决议不通过，LED 不亮。

本项目利用 Verilog 语言，在 PLD（Programmable Logic Device，可编程逻辑器件）开发平台上设计一个三人表决器，并用 FPGA（Field Programmable Gate Array，现场可编程门阵列）器件为载体最终实现硬件电路。通过三人表决器设计项目训练及相关知识的学习，达到如下目标。

- (1) 了解 EDA 技术及其发展。
- (2) 了解 PLD 的发展概况及结构特点。
- (3) 掌握学习开发板的硬件组成。
- (4) 掌握 EDA 的开发流程。
- (5) 掌握 Quartus II 软件的文本输入设计方法。
- (6) 能使用 Quartus II 软件对设计电路进行功能仿真。
- (7) 掌握 Verilog 语言的基本结构。
- (8) 能利用学习开发板对设计电路进行下载与调试。

(9) 能按照现场管理要求（整理、整顿、清扫、清洁、素养、安全、环保、节能）安全文明生产。

- (10) 会查找相关资料。
- (11) 会撰写项目报告并答辩。
- (12) 具有团队合作精神,具有一定的组织协调能力。

【知识准备】

1.1 EDA 技术及其发展

1.1.1 EDA 技术概念

EDA 是电子设计自动化 (Electronic Design Automation) 缩写,是 20 世纪 90 年代初从 CAD (Computer Aided Design, 计算机辅助设计)、CAM (Computer Aided Manufacturing, 计算机辅助制造)、CAT (Computer Aided Testing, 计算机辅助测试) 和 CAE (Computer Aided Engineering, 计算机辅助工程) 的概念发展而来的。

一般而言,EDA 技术的概念分狭义和广义两种。

狭义的 EDA 技术,就是以计算机为工作平台,以 EDA 软件工具为开发环境、以硬件描述语言为系统逻辑描述的主要表达方式、以 ASIC (Application Specific Integrated Circuit, 专用集成电路) 为实现载体的电子产品自动化设计过程。狭义的 EDA 技术或称为 IES/ASIC 自动设计技术,通过使用有关的开发软件,自动完成电子系统设计的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真,以及对特定目标芯片的适配编译、逻辑映射、编程下载等工作,最终形成集成电子系统或专用集成芯片。本书讨论的对象专指狭义的 EDA 技术。

典型的 EDA 工具中必须包含两个特殊的软件包,即综合器和适配器。综合器的功能就是将设计者在 EDA 平台上完成的针对某个系统项目的 HDL、原理图或状态图形描述,针对给定的硬件系统组件,进行编译、优化、转换和综合,最终获得人们欲实现功能的描述文件。综合器在工作前,必须给定所要实现的硬件结构参数,它的功能就是将软件描述与给定的硬件结构用一定的方式联系起来。也就是说,综合器是软件描述与硬件实现的一座桥梁。综合过程就是将电路的高级语言描述转换为低级的、可与目标器件 FPGA/CPLD 相映射的网表文件。

适配器的功能是将由综合器产生的网表文件配置到指定的目标器件中,产生最终的下载文件,如 JED 文件。适配器所选定的目标器件 (FPGA/CPLD 芯片) 必须属于在综合器中已指定的目标器件系列。

硬件描述语言 (Hardware Description Language, HDL) 是相对于一般的计算机软件语言而言的,如 C、PASCAL。HDL 语言是用于设计硬件电子系统的计算机语言,它能描述电子系统的逻辑功能、电路结构和连接方式。设计者可利用 HDL 程序来描述所希望的电路系统,规定器件结构特征和电路的行为方式;然后利用综合器和适配器将此程序编译成能控制 FPGA 和 CPLD 的内部结构,并实现相应逻辑功能的门级或底层的结构网表文件或下载文件。目前,就 FPGA/CPLD 开发来说,比较常用和流行的 HDL 主要有 ABEL - HDL、AHDL、Verilog HDL 和 VHDL。

广义的 EDA 技术,除了包括狭义的 EDA 技术外,还包括计算机辅助分析 (Computer Aided Analysis, CAA),如 PSpice、EWB、MATLAB 等,印制电路板计算机辅助设计 PCB - CAD 技术 (如 Protel、OrCAD 等)。在广义的 EDA 技术中,CAA 技术和 PCB - CAD 技术不具备逻辑综合和逻辑适配的功能,因此它并不能称为真正意义上的 EDA 技术。

EDA 技术的 3 个层次。

- (1) EWB、PSpice、Protel 作为 EDA 的初级内容。
- (2) 利用硬件描述语言完成对 CPLD/FPGA 的开发等作为中级内容。
- (3) ASIC 的设计作为高级内容。

1.1.2 EDA 技术的主要特征

EDA 技术主要有以下特征。

- (1) 高层次综合与优化的理论与方法取得了很大的进展,其结果大大缩短了复杂的 ASIC 设计周期,同时提高了设计质量。
- (2) 采用硬件描述语言来描述 10 万门以上的设计,形成了国际通用的 Verilog 等硬件描述语言。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,并可建立独立的工艺设计文档,便于设计重用。
- (3) 开放式的设计环境 (各厂家均适合)。
- (4) “自顶向下”的算法。
- (5) 丰富的元器件模型库。
- (6) 建立并行设计工程框架结构的集成化设计环境,以适应当今 ASIC 的特点,即规模大而复杂,数字与模拟电路并存,硬件与软件并存,产品上市更新快。

1.1.3 EDA 技术发展历程

EDA 技术是伴随着计算机、集成电路、电子系统技术发展起来的,至今已有 40 多年的历程,大致可以分为三个发展阶段。

第一阶段为 20 世纪 70 年代的 CAD (计算机辅助设计) 阶段。这一阶段的主要特征是利用计算机辅助进行电路原理图编辑、PCB 布线,使得设计师从传统高度重复繁杂的绘图劳动中解脱出来。

第二阶段为 20 世纪 80 年代的 CAED (计算机辅助工程设计) 阶段。这一阶段的主要特征是以逻辑模拟、定时分析、故障仿真、自动布局布线为核心,重点解决电路设计的功能检测等问题,使设计能在产品制作之前预知产品的功能与性能。

第三阶段为 20 世纪 90 年代的 EDA (电子设计自动化) 阶段。这一阶段的主要特征是以高级描述语言、系统仿真和综合技术为特点,采用自上而下的设计理念,将设计前期的许多高层次设计由 EDA 工具来完成。

1.2 可编程逻辑器件

数字电路由早期的电子管、晶体管、中小规模集成电路,发展到超大规模集成电路以及

许多具有特定功能的专用集成电路 ASIC。在 20 世纪 80 年代初，随着微电子技术的高速发展，特别是 ASIC 的市场需求，出现了一系列生命力强、应用广泛、发展迅猛的新型集成电路，即可编程逻辑器件 (Programmable Logic Devices, PLD)。它们是一种由用户根据自己要求来构造逻辑功能的数字集成电路，一般可利用计算机辅助设计，即用原理图、状态机、布尔方程、硬件描述语言 HDL 等方法来表示设计思想，经一系列编译或转换程序，生成相应的目标文件，再由编程器或下载电缆将设计文件配置到目标文件中，这时可编程器件就可作为满足用户要求的专用集成电路使用了。PLD 适宜于小批量生产的系统，或在系统开发研制过程中采用。因此在计算机硬件、自动化控制、智能化仪表、数字电路系统等领域中得到了广泛的应用。它的应用和发展不仅简化了电路设计，降低了成本，提高了系统的可靠性和保密性，而且给数字设计方法带来了重大变化。其中应用最广泛的是现场可编程门阵列 (Field Programmable Gate Array, FPGA) 和复杂可编程逻辑器件 (Complex Programmable Logic Devices, CPLD)。

1.2.1 可编程逻辑器件概述

1.2.1.1 可编程逻辑器件的历史演变

最早的可编程逻辑器件出现在 20 世纪 70 年代初，主要是可编程只读存储器 (Programmable Read Only Memory, PROM) 和可编程逻辑阵列 (Programmable Logic Array, PLA)，也称为现场可编程逻辑阵列 (Field Programmable Logic Array, FPLA)；20 世纪 70 年代末出现了可编程阵列逻辑 (Programmable Array Logic, PAL) 器件。

20 世纪 80 年代初，美国 Lattice 半导体公司推出通用阵列逻辑 (Generic Array Logic, GAL) 器件，具有可擦除、可重复编程和可加密等特点。

20 世纪 80 年代中期，美国 Xilinx 公司率先推出了现场可编程阵列 (FPGA) 器件。

20 世纪 90 年代初，Lattice 公司推出了在系统可编程大规模集成电路 (In - System Programming Large - Scale Integrated Circuits, ISPLSIC)。

20 世纪 90 年代末，Lattice 公司推出了模拟可编程逻辑器件 (In - System Programming Ability Analog Circuits, ISPPAC)，它可实现三种功能：信号调理、信号处理及信号转换。

1.2.1.2 可编程逻辑器件的基本结构与分类

1. 可编程逻辑器件的基本结构

可编程逻辑器件的基本结构是由“与”阵列和“或”阵列、输入缓冲电路和输出电路组成，其框图如图 1 - 1 所示。其中“与”阵列和“或”阵列是核心，“与”阵列用来产生乘积项，“或”阵列用来产生乘积项之和形式的函数。输入缓冲电路可以产生输入变量的原变量和反变量，输出结构可以是组合输出、时序输出或是可编程输出，输出信号还可以通过内部通道反馈到输入端。

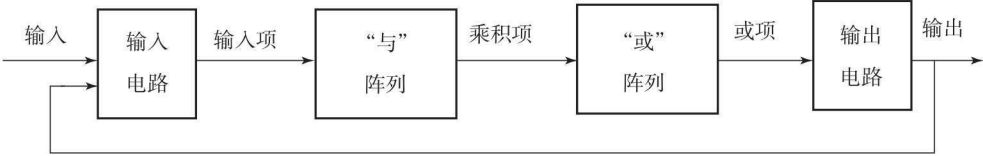


图 1 - 1 PLD 基本结构框图

2. 可编程逻辑器件的分类

可编程器件按集成度来区分可分为简单 PLD（SPLD）和复杂 PLD（CPLD），如图 1-2 所示。

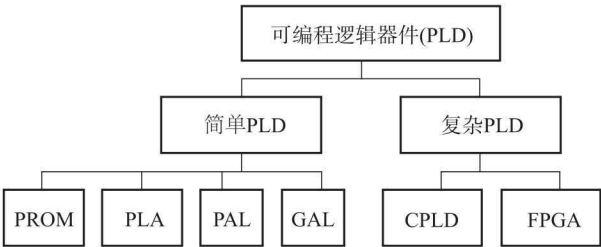


图 1-2 PLD 发展结构示意图

1) 简单可编程逻辑器件

简单可编程逻辑器件（Simple Programmable Logic Devices, SPLD）是编程器件的早期产品，包括可编程只读存储器（Programmable Read Only Memory, PROM）、可编程逻辑阵列（Programmable Logic Array, PLA）、可编程阵列逻辑（Programmable Array Logic, PAL）和通用阵列逻辑（Generic Array Logic, GAL）。SPLD 的典型结构是由与门阵列、或门阵列组成，能够以“积之和”的形式实现布尔逻辑函数。因为任意一个组合逻辑都可以用“与-或”表达式来描述，所以 SPLD 能够完成大量的组合逻辑功能，并且具有较高的速度和较好的性能。

最早的可编程逻辑器件是熔丝编程的只读存储器 PROM。当与阵列固定（在结构图中常用·表示），或阵列可编程时（在结构图中常用 X 表示），称之为可编程只读存储器（PROM），其结构如图 1-3 所示。这种可编程逻辑器件一般用作存储器，其输入为存储器的地址，输出为存储单元的内容。由于与阵列采用全译码器，随着输入的增多，阵列规模按输入的 2^n 增长。当输入的数目太大时，器件功耗增加，而巨大的阵列开关时间也会导致其速度缓慢。但 PROM 价格低，易于编程，同时没有布局、布线问题，性能完全可以预测。此外，它不可擦除、不可重写的局限性也由于 EPROM、E²PROM 的出现得到解决，所以还是具有一定的应用价值。

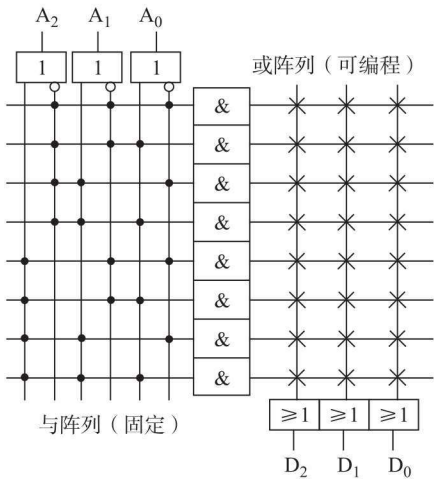


图 1-3 PROM 基本结构

当与阵列和或阵列都可编程时，称之为可编程逻辑阵列（PLA），其结构如图 1-4 所示。由于与阵列可编程，使得 PROM 中由于输入增加而导致规模增加的问题不复存在，从而有效地提高了芯片的利用率。PLA 用于含有复杂的随机逻辑置换的场合是较为理想的，PLA 器件既有现场可编程的，也有掩膜可编程的，虽然利用率高，但运行速度较慢，仅适用于小规模逻辑，同时相对高的价格也妨碍了它被广泛使用。

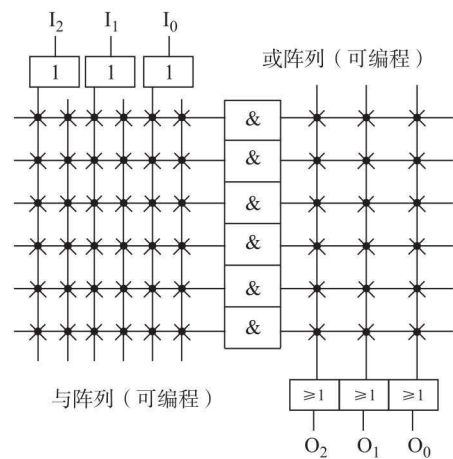


图 1-4 PLA 结构

随后出现了可编程阵列逻辑 PAL。当或阵列固定，与阵列可编程时，称之为可编程阵列逻辑 PAL，其结构如图 1-5 所示，其或门的输出可以通过触发器有选择地被置为寄存状态。与阵列的可编程特性使输入项可以增多，而固定的或阵列又使器件得到简化。在这种结构中，每个输出都是若干乘积项之和，其中乘积项的数目是固定的。PAL 的这种基本门阵列结构对于大多数逻辑函数是有很有效的，因为大多数逻辑函数都可以方便地化简为若干个乘积项之和，即与或表达式，同时这种结构也提供了较高的性能和速度，所以一度成为 PLD 发展史上的主流。PAL 有几种固定的输出结构，不同的输出结构对应不同的型号，设计时可以根据实际需要进行选择。PAL 器件是现场可编程的，它的实现工艺有反熔丝技术、EPROM 技

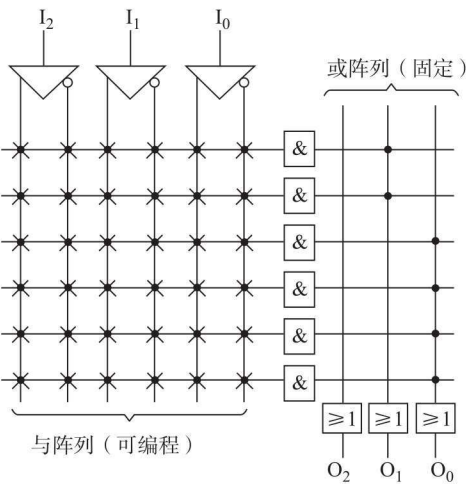


图 1-5 PAL 结构

术和 E²PROM 技术,但由于市场上类型和结构较多,反而给用户带来了使用的不方便和修改不方便,目前已被淘汰。

在 PAL 的基础上,随后发展出了一种通用阵列逻辑 GAL。它采用 E²PROM 工艺,彻底解决了熔丝型可编程器件的一次可编程问题,实现了电可擦除、电可改写、可重复编程、可设置加密的功能。GAL 的输出可由用户来定义,它的每个输出端都集成着一个可编程的输出逻辑宏单元 (Output Logic Macro Cell, OLMC)。如图 1-6 所示, GAL16V8 的逻辑框图中,在 12~19 号管脚内就各有一个 OLMC。

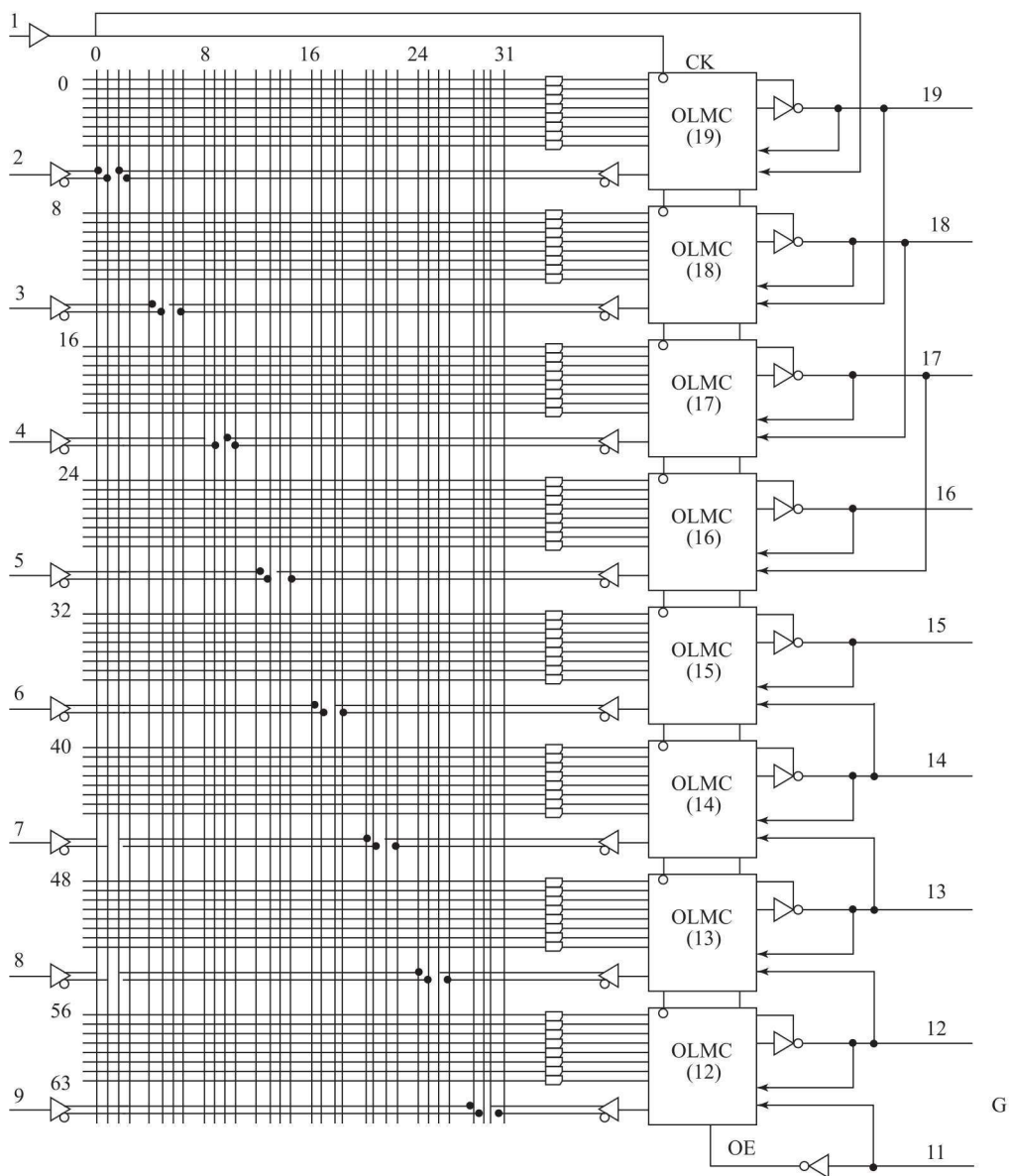


图 1-6 GAL16V8 逻辑框图

GAL22V10 的 OLMC 内部结构如图 1-7 所示。从图中可以看出,OLMC 中除了包含或门阵列和 D 触发器之外,还有两个多路选择器 (MUX);其中 4 选 1 MUX 用来选择输出方

式和输出极性, 2 选 1 MUX 用来选择反馈信号。这些选择器的状态都是可编程控制的, 通过编程改变其连线可以使 OLMC 配置成多种不同的输出结构, 完全包含了 PAL 的几种输出结构。普通 GAL 器件只有少数几种基本型号就可以取代数十种 PAL 器件, 因而 GAL 是名副其实的通用可编程逻辑器件。由于 GAL 的设计具有很强的灵活性, 所以至今仍被许多人使用。GAL 的缺点是规模较小, 对于较为复杂的逻辑电路显得力不从心。

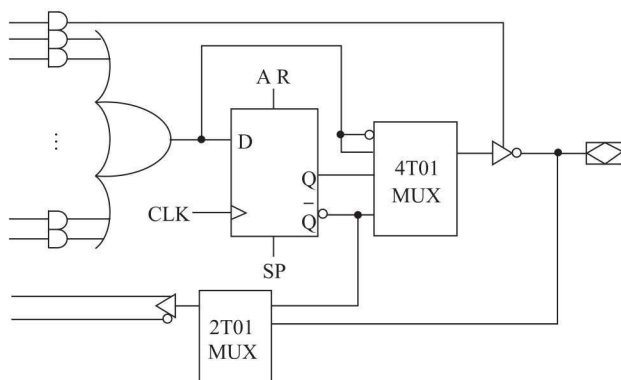


图 1-7 GAL22V10 的 OLMC 结构框图

2) 复杂可编程逻辑器件

复杂可编程逻辑器件 (Complex Programmable Logic Devices, CPLD) 出现在 20 世纪 80 年代末期, 其结构区别于早期的简单 PLD, 最基本的特点在于: 简单 PLD 是逻辑门编程, 而复杂 PLD 为逻辑板块编程, 即以逻辑宏单元为基础, 加上内部的与或阵列和外部的输入/输出模块, 不但实现了除简单逻辑控制之外的时序控制, 又扩大了在整个系统中的应用范围和扩展性。

3) 现场可编程门阵列

现场可编程门阵列 (Field Programmable Gate Array, FPGA) 是一种可由用户自行定义配置的高密度专用集成电路, 它将定制的超大规模集成电路 (Very Large Scale Integrated Circuits, VLSI) 的单片逻辑集成优点和用户可编程逻辑器件的设计灵活、工艺实现方便、产品上市快捷的长处结合起来。器件采用逻辑单元阵列结构, 静态随机存储工艺, 设计灵活, 集成度高, 可重复编程, 并可现场模拟调试验证。

1.2.1.3 PLD 相对于 MCU 的优势所在

1. MCU 经常面临的难题

MCU (Micro controller Unit, 微控制单位) 逻辑行为上的普适性, 常会引导人们认为 MCU 是无所不能的, 任何一个电子系统设计项目, MCU 无可置疑地成为主角。但不深入考察 MCU 的优势和弱点, 事事都以 MCU 越俎代庖、勉为其难, 将严重影响系统设计的最佳选择和性价比的提高。

1) 运行速度

从理论上讲, MCU 几乎可以解决任何逻辑的实现, 但 MCU 是通过内部的 CPU 逐条执行软件指令来完成各种运算和逻辑功能, 无论多么高的工作时钟频率和多么好的指令时序方式, 在排队式串行指令执行方式 (DSP 处理器也不能逃避这种工作方式) 面前, 其工作速度和效率必将大打折扣。因此, MCU 在实时仿真、高速工控或高速数据采样等许多领域尤

显力不从心，速度是 MCU 及其系统面临的最大挑战。

2) 复位

复位工作方式是 MCU 的另一致命弱点，任何 MCU 工作初始都必须经历一个复位过程，否则将无法进行正常工作。MCU 的复位必须满足一定的电平条件和时间条件（长达毫秒级）。在工作电平有某种干扰性突变时，MCU 不可缺少的复位设置将成为系统不可工作的重要因素。而且这种产生于复位的不完全性，构成了系统不可靠工作的隐患，其出现方式极为随机和动态，一般难于检测。一些系统在工作中出现的“假复位”和不可靠性复位带来的后果是十分严重的。尽管人们不断提出了种种改善复位的方法及可靠复位的电路，市场上也有层出不穷的 MCU 复位监控专用器件，但到目前为止，复位的可靠性问题仍然未能得到根本解决。

3) 程序“跑飞”

在强干扰或某种偶然因素下，任何 MCU 的程序指针都极可能越出正常的程序流程“跑飞”，这已是不争的事实。事实证明，无论多么优秀的 MCU，无论多么良好的抗干扰措施，包括设置任何方式的内外硬件看门狗，在受强干扰特别是强电磁干扰情况下，MCU 都无法保证仍能正常工作而不进入不可挽回的“死机”状态。尤其是当程序指针“跑飞”与复位不可靠因素相交错时，情况将变得尤为复杂。

2. CPLD/FPGA 的优势

基于 CPLD/FPGA 器件的开发应用可以从根本上解决 MCU 所遇到的问题。与 MCU 相比，CPLD/FPGA 在某些领域的优势是多方面的和根本性的。

1) 高速性

CPLD/FPGA 的时钟延迟仅为纳秒级，结合其并行工作方式，在超高速应用领域和实时测控方面有非常广阔的应用前景。

2) 高可靠性

在高可靠应用领域，MCU 的缺憾为 CPLD/FPGA 的应用留下了很大的用武之地。除了不存在 MCU 所特有的复位不可靠与程序“跑飞”等固有缺陷外，CPLD/FPGA 的高可靠性还表现在几乎可将整个系统下载于同一芯片中，从而大大缩小了体积，易于管理和屏蔽。

3) 编程方式

采用 JTAG 在系统配置编程方式，可对正在工作的系统上的 CPLD/FPGA 进行在系统编程，这在工控、智能仪器仪表、通信和军事上有特殊用途，同时为系统的调试带来极大的方便。

4) 标准化设计语言

CPLD/FPGA 的设计开发工具，通过符合国际标准的硬件描述语言（如 VHDL 或 Verilog HDL）来进行电子系统设计和产品开发。由于开发工具的通用性、设计语言的标准化以及设计过程几乎与所用的 CPLD/FPGA 器件的硬件结构没有关系，所以设计成功的各类逻辑功能块软件都有很好的兼容性和可移植性。

在电子应用系统设计中，充分了解系统的需求，认识 MCU 与 CPLD/FPGA 各自的优势所在，利用 MCU 与 CPLD/FPGA 在功能和性能上的互补性，在构成系统的功能模块中合理选择 MCU 与 CPLD/FPGA，充分发挥 MCU 与 CPLD/FPGA 的所长，使应用系统实现最佳的技术配合。

1.2.2 FPGA 和 CPLD

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。为了弥补这一缺陷,20 世纪 80 年代中期,Altera 和 Xilinx 分别推出了类似于 PAL 结构的扩展型 CPLD 以及与标准门阵列类似的 FPGA,它们都具有体系结构和逻辑单元灵活、集成度高以及适用范围宽等特点。这两种器件兼容了 PLD 的通用门阵列的优点,可实现较大规模的电路,编程也很灵活。与门阵列等其他 ASIC 相比,它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无须测试、质量稳定,以及可实时在线检验等优点,因此被广泛应用于产品的原型设计和产品设计(一般在 10 000 以下)之中。几乎所有应用门阵列、PLD 和中小规模通用数字集成电路的场合均可以应用 FPGA 和 CPLD 器件。

不过,不同厂家的叫法不尽相同,Xilinx 把基于查找表技术,SRAM 工艺,要外挂配置用的 E²PROM 的 PLD 称为 FPGA;把基于乘积项技术,FLASH (类似 E²PROM 工艺)工艺的 PLD 称为 CPLD;Altera 把自己的 PLD 产品,MAX 系列(乘积项技术,E²PROM 工艺),FLEX 系列(查找表技术,SRAM 工艺)都称为 CPLD。由于 FLEX 系列也是 SRAM 工艺,基于查找表技术,要外挂配置用的 EPROM,用法和 Xilinx 的 FPGA 一样,所以很多人把 Altera 的 FLEX 系列产品也称为 FPGA。CPLD 是可编程逻辑器件的简称,FPGA 是现场可编程门阵列的简称,两者的功能基本相同,只是实现原理略有不同。对用户而言,CPLD 与 FPGA 的内部结构稍有不同,但用法一样,所以大多数情况下,不加以区分,统称为可编程逻辑器件或 CPLD/FPGA。

PLD 是电子设计领域中最具活力和发展前途的一项技术,它的影响丝毫不亚于 20 世纪 70 年代单片机的发明和使用。PLD 能做什么呢?可以毫不夸张地讲,PLD 能完成任何数字器件的功能,上至高性能的 CPU,下至简单的 74 电路,都可以用 PLD 来实现。PLD 如同一张白纸或是一堆积木,工程师可以通过传统的原理图输入法,或是硬件描述语言自由地设计一个数字系统。通过软件仿真,可以事先验证设计的正确性。在 PCB 完成以后,还可以利用 PLD 的在线修改能力,随时修改设计而不必改动硬件电路。使用 PLD 来开发数字电路,可以大大缩短设计时间,减少 PCB 面积,提高系统的可靠性。PLD 的这些优点使得 PLD 技术在 20 世纪 90 年代以后得到飞速的发展,同时也大大推动了 EDA 软件和硬件描述语言(HDL)的进步。如何使用 PLD 呢?其实 PLD 的使用很简单,学习 PLD 要比学习单片机容易得多,有数字电路基础,会使用计算机,就可以进行 PLD 的开发。

1.2.2.1 基于乘积项的 PLD 结构

采用基于乘积项(Product-Term)结构的 PLD 芯片有 Altera 的 MAX7000、MAX3000 系列(E²PROM 工艺),Xilinx 的 XC9500 系列(FLASH 工艺)和 Lattice, Cypress 的大部分产品(E²PROM 工艺)。先看一下这种 PLD 的总体结构(以 MAX7000 为例,其他型号的结构与此类似),如图 1-8 所示。

这种 PLD 可分为三部分:宏单元(Macrocell),可编程连线阵列(Programmable Interconnect Array, PIA)和 I/O 控制块(I/O Control Blocks)。宏单元是 PLD 的基本结构,由它来实现基本的逻辑功能。图 1-8 中灰色部分是多个宏单元的集合(因为宏单元较多,没有一一画出)。可编程连线阵列负责信号传递,连接所有的宏单元。I/O 脚控制块负责输入/输出的电气