

# 数字系统设计实验教程

主编 杜玉远 李景华

东北大学出版社

✦ 沈摇阳 ✦



# 前 摇 摇 言

随着电子技术的迅猛发展和可编程 粤 系 统 的 设计技术的日趋完善，数字电路系统的设计也取得了长足的进步。为了更好地配合数字系统设计课程和可编程逻辑器件及 秘 粤 技术课程的实验教学和课程设计的需求，我们编写了本书。本书以实践教学为目的，以设计入门为基础，以综合性和设计性为重点，涵盖了数字系统设计及其相关课程所需求的实验和设计内容。对每个设计题目，都按说明原理、概括方法、提供部分参考设计的思路编撰，以求起到抛砖引玉的作用，收到培养学生的独立思考能力和充分发挥学生创造性的效果。

本书详细地给出了验证性和综合设计性实验的实验目的、实验原理和实验要求及相应的设计例程。另外，介绍了目前较为流行的单机板的可编程 粤 系 统 开 发 的 软 件 系 统 隔 云 纜 曾 和 配 粤 粤 基 础 引 实 验 系 统 使 用 方 法 和 硬 件 实 验 系 统 的 组 成 结 构 和 使 用 方 法。

本书由杜玉远、李景华主编。第一章、第二章、第三章由杜玉远和许桂芝编写，第四章、第五章、第六章由李景华和李汇明编写。

限于编者水平和编写时间仓促，书中不妥和错误之处在所难免，敬请广大读者指正。

编著者

圆 苑 年 源 月

# 目 录

第一章 绪论	员
1.1 概 述	员
1.2 可编程逻辑器件简介	员
1.3 数字系统设计开发系统选择	缘
1.4 数字系统设计实验系统选择	缘
1.5 数字系统设计开发流程	远
1.6 数字系统设计学时和内容安排	苑
第二章 常用数字逻辑单元实验	怨
实验一 编码电路设计实验	怨
实验二 译码电路设计实验	圆
实验三 数据选择器设计实验	苑
实验四 触发器和计数器设计实验	圆
实验五 状态机设计实验	圆
实验六 三态总线输入输出控制实验	缘
实验七 加法器设计实验	愿
实验八 移位寄存器设计实验	猿
实验九 存储器设计仿真实验	猿
实验十 简易数字钟设计实验	猿
第三章 综合设计性实验	猿
实验一 有符号多位乘法器设计实验	猿
实验二 有符号多位除法器设计实验	源
实验三 并行译码器设计实验	源
实验四 同步时序信号产生器设计实验	缘
实验五 抢答器设计实验	缘
实验六 交通灯控制器设计实验	愿
实验七 数字频率计设计实验	苑
实验八 数字钟设计实验	愿
实验九 可编程并行通信接口芯片设计实验	怨
实验十 可编程定时器/计数器设计实验	怨

实验十一摇点阵式 液晶显示控制系统设计实验 .....	页源
实验十二摇简易计算器设计实验 .....	页源
实验十三摇简易逻辑分析仪设计实验 .....	页源
实验十四摇数字信号发生器设计实验 .....	页源
实验十五摇简易 存储器设计实验 .....	页源
第四章摇云 操作系统开发系统 .....	页源
摇云 操作系统开发系统简介 .....	页源
摇云 操作系统设计入门 .....	页源
摇云 操作系统设计技巧 .....	页源
第五章摇 操作系统开发系统 .....	页源
摇 操作系统简介 .....	页源
摇 操作系统设计入门 .....	页源
摇 操作系统设计技巧 .....	页源
第六章摇多功能 型实验系统 .....	页源
摇概述 .....	页源
摇 型系统主要构成 .....	页源
摇系统使用说明 .....	页源
摇可编程模拟器件 子板 .....	页源

# 第一章 绪论

## 1.1 概述

数字化进程的不断加快，推动了具有高速度、高集成度、低功耗的可编程器件的发展，同时也推动了用于开发这类器件的数字系统设计的发展和进步，电路设计构成逐步地从中、小规模芯片转为大规模、超大规模芯片。掌握数字系统设计技术，使用可编程大规模、超大规模芯片已经成为从事电子设计人员必需具有的基本能力。

## 1.2 可编程 ASIC 器件简介

作为数字系统设计的目标芯片近几年有非常迅速的发展，目前在全球的市场占有率较多的有 Xilinx 公司、Altera 公司、Cyclops 公司。这三家公司基本垄断了全球的可编程器件市场。

目前比较流行的可编程器件根据其内部结构特点可划分为 CPLD (Complex Programmable Logic Device) 和 FPGA (Field Programmable Gate Array)。CPLD 是复杂可编程逻辑器件的简称，其结构与早期的 PAL 相似，内部包含多个逻辑宏单元或宏模块。FPGA 是现场可编程门阵列的简称。FPGA 是基于乘积项的逻辑资源更高的可编程器件。

CPLD 的内部结构是基于乘积项 (Product Term) 技术的可编程器件。代表器件有 Xilinx 公司的 XC9500 系列和 Altera 公司的 EPLD 系列等。FPGA 的内部结构，如图 1-1 所示。从图 1-1 中可以看出在 XC9500 系列器件中主要包括逻辑阵列块 (Logic Array Block)、宏单元 (Macro Cell)、扩展乘积项 (Extended Product Term)、可编程连线阵列 (Programmable Connection Array) 和控制块 (Control Block)。另外，在 XC9500 系列器件的内部结构中还包括全局时钟输入和全局输出使能的控制线，这些线在不用作此功能时，可以作为一般的输入使用。

逻辑阵列块 (LAB) 是 XC9500 系列器件中的最大的逻辑单元，由图 1-1 中可以看出每个逻辑阵列块 (LAB) 由  $n$  个宏单元构成，每个逻辑阵列块与各自相对应的控制块相连，四个逻辑阵列块通过可编程连线阵列 (PCA) 和全局总线连接在一起，全局总线由所有的专用输入、控制引脚和宏单元反馈构成。利用这些连线可以实现不同逻辑阵列块之间的连接，用以实现更复杂的逻辑功能。

宏单元是 XC9500 系列器件的具体逻辑单元，用来实现各种具体的逻辑功能。宏单元由逻辑阵列、乘积项选择矩阵和可编程触发器构成，其结构如图 1-2 所示。逻辑阵列用来实现组合逻辑函数，每个宏单元提供  $m$  个乘积项。通过乘积项选择矩阵实现这  $m$  个乘

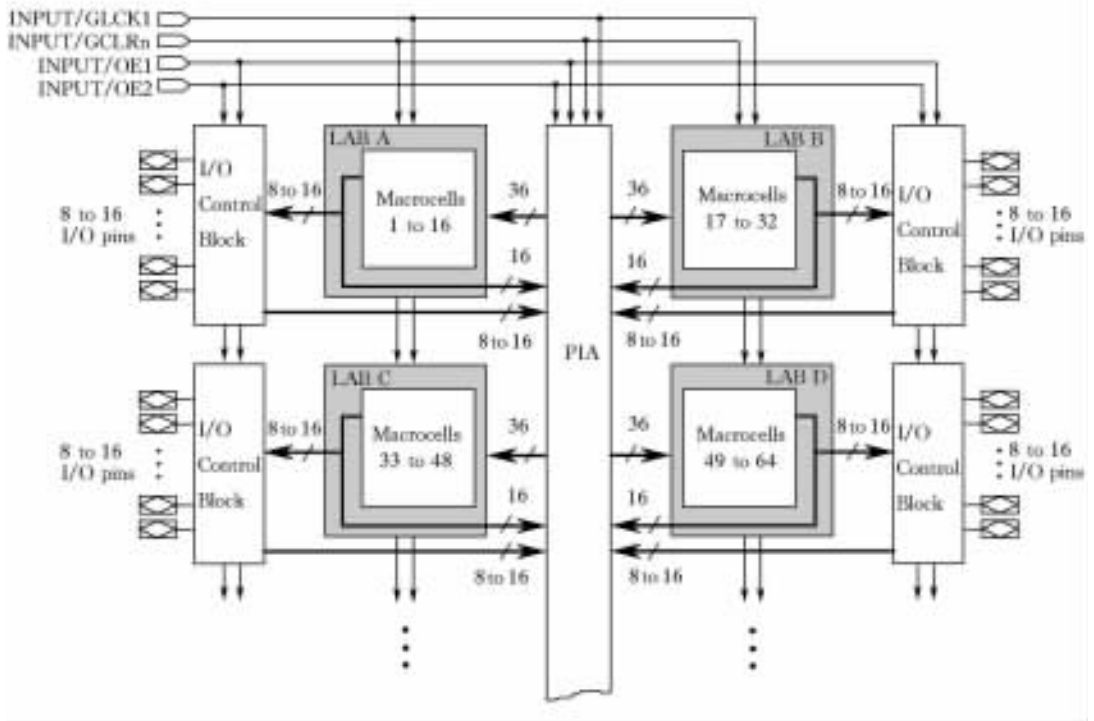


图 1 74VHC163 系列器件的内部结构图

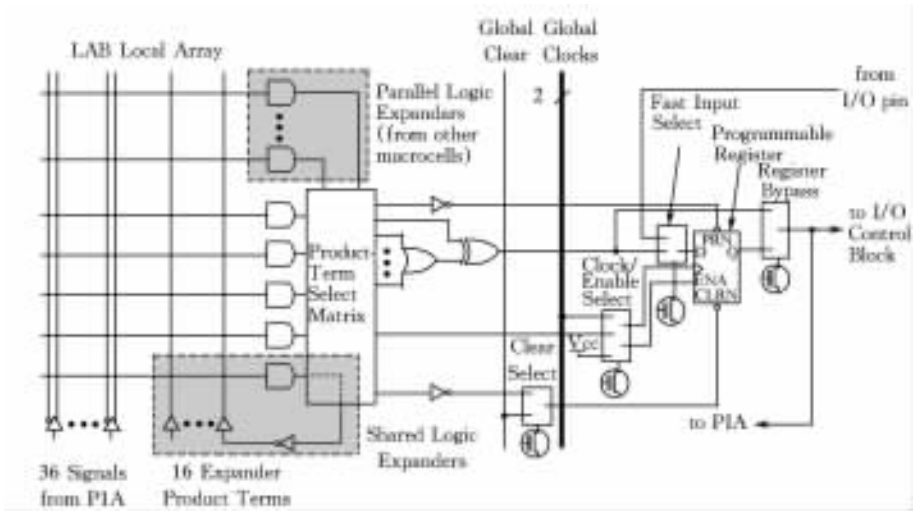


图 2 74VHC163 系列的宏单元结构图

积项的逻辑函数，或者使这个乘积项作为宏单元的触发器的辅助输入（清除、置位、时钟和时钟使能）。每个宏单元的一个乘积项还可以反馈到逻辑阵列。宏单元中的可编程触发器可以被单独编程为闩锁元件或 D 触发器，可编程触发器还可以被旁路掉，用以实现纯组合逻辑方式工作。每个可编程触发器可按以下四种不同时钟方式进行控制。

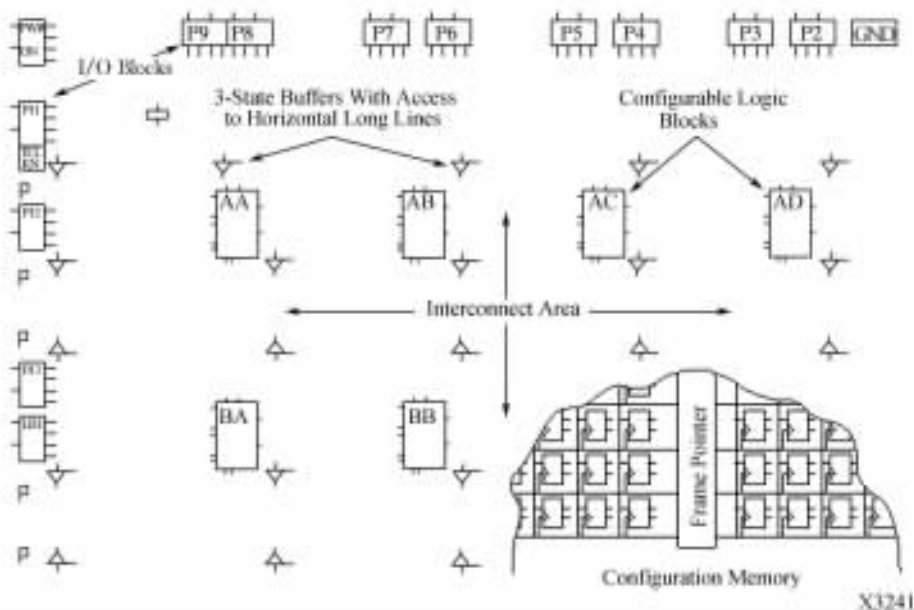
(1) 全局时钟（全局清除时钟）。这种方式能够实现最快的时钟控制。

(圆) 带高电平使能的全局时钟。这种方式能够实现具有使能控制的触发器，并能够实现最快的时钟控制。

(猿) 来自乘积项的时钟。触发器由来自隐含宏单元或逻辑引脚的信号进行时钟控制，这种方式一般具有较慢的时钟控制。

基于熔丝结构的可编程逻辑器件的编程信息是采用热挥发工艺实现的，这种工艺具有非易失性，编程数据可以保持数十年，此类器件只能对其进行有限次(几千次)编程。另外，可以对器件进行加密，而且一旦被加密，很难破解。

云逻辑的内部结构是基于查找表(蕴藏)技术的可编程逻辑芯片。代表的器件有粤公司的云蕴藏, 粤蕴藏, 粤蕴藏系列和载公司的载源田, 杂源田, 灾源田系列等。载公司生产的云逻辑器件的内部结构是构逻辑单元阵列蕴藏(蕴藏器悦造粤), 蕴藏主要由三部分组成: 可配置逻辑块悦, 可编程输入输出块, 可编程内部连线。云逻辑的内部结构如图员所示。由图员看出, 排成矩阵形式的内部方块称为可配置逻辑块悦(悦逻辑单元阵列粤)。在四周的小方块是输入输出块(悦输入输出阵列粤), 在各块之间有纵向、横向两种布线通道, 布线通道构成云逻辑的布线资源。布线资源由金属长线、线段以及编程开关组成。



图员 云逻辑器件的内部结构

云逻辑的核心功能块是悦, 一片云逻辑内部含有多个悦。它们规律地按行和列排成矩阵。悦由杂构成的查找表、触发器、门电路和可编程多路数据选择器构成。悦中实现组合逻辑功能的查找表中的信息、多路数据选择器的控制信息、的组态信息和可编程资源中编程开关的控制信息等都是通过将云逻辑的组态数据装入其内部代码存储器杂来建立的。

图员是载公司的载源田系列的基本可配置逻辑块悦结构, 从图员原可以看

出一个 悦器由两个独立的 源输入查找表、一个 猿输入查找表和两个触发器构成。查找表实际上是一个 砸寻运结构，一个 源输入的查找表就是一个 员元伊员砸寻运单元，通过编程可以实现 源输入变量的任意逻辑函数。悦器内部的两个触发器通过编程，可以配置成不同类型的触发器。

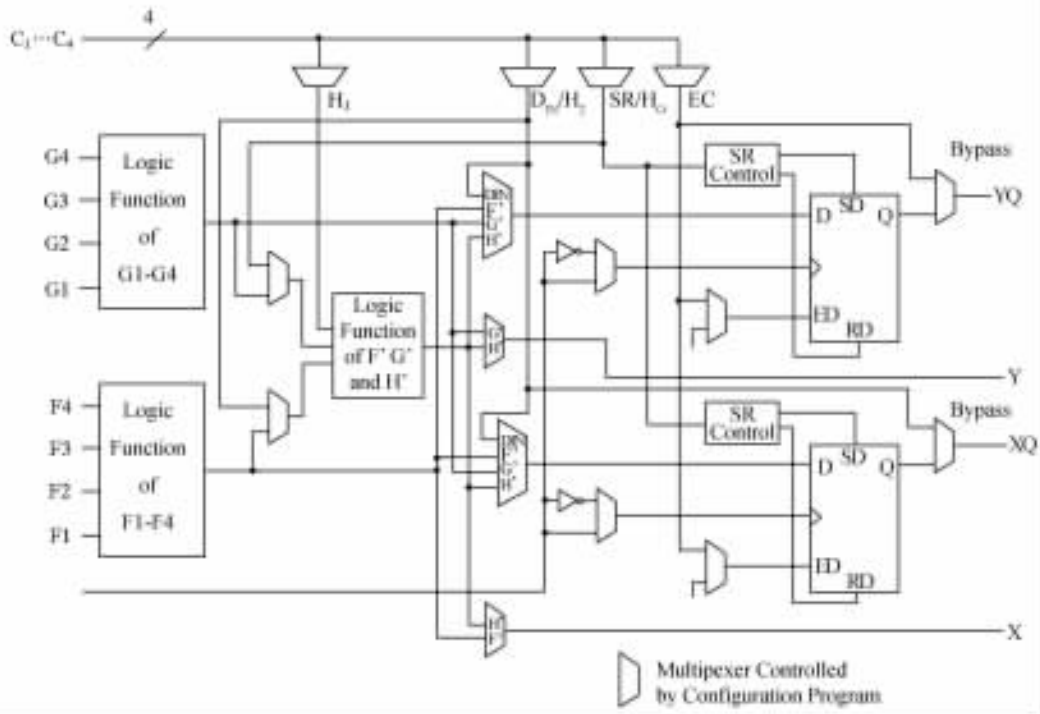


图 员源 源系列器件的 悦器结构

基于 云寻粤结构的可编程 粤器器件的编程信息是采用 杂寻运单元的形式存储，其静态存储单元如图 员源 所示。砸寻运单元存储信息 园和 员 这些 园和 员代表的电平控制编程开关中晶体管或传输门是导通还是断开，从而实现 云寻粤内部的可编程互连功能。由于器件掉电后，杂寻运中的数据会丢失，因此，设计中需要根据情况为器件提供上电配置的电路，以保证上电后器件可以正常工作。

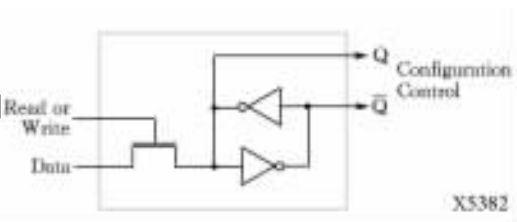


图 员源 静态存储单元 杂寻运

由此可见，悦器内部具有大量的乘积项和较少的触发器，而 云寻粤正好相反，其内部具有较少的组合逻辑资源和较多的触发器。当设计组合逻辑电路成分较多，而时序电路成分较少的电路时，或者需要很好的加密功能时，选择 悦器比较适合，例如设计宽译码器和复杂编码器时，最好选择 悦器。当设计功能较为复杂的时序电路时，由于复杂的时序电路需要大量的触发器单元，例如设计 粤器，则选择 云寻粤比较适合。

## 图 1-1 数字系统设计开发系统选择

为了学习和使用可编程逻辑器件，需要有一个合适的开发平台，即 FPG 开发系统。FPG 开发系统按功能划分为集成开发系统、综合系统和仿真系统。集成开发系统一般是由可编程逻辑器件生产厂商开发的开发系统平台，用户利用这些开发平台可以完成全部设计工作。代表的开发平台如 Xilinx 公司的 ISE 系列、Altera 公司的 Quartus II 系列和 Lattice 系列是目前普遍使用的开发系统。更多的 FPG 开发和综合系统一般是由第三方软件开发商开发的，为各器件厂商和普通用户提供对原始设计输入进行综合产生编程数据的软件，代表的综合系统如 Synopsys 公司的 Synplify 系列。仿真系统是仅用于对设计进行功能仿真的 FPG 软件，一般也是由第三方开发的，它不能生成编程数据，它只用作设计初期工作，也就是说它不涉及具体的器件。具有代表性的仿真系统如 ModelSim 公司的 ModelSim 仿真器。

初学者应将功能齐全，能够完成全部设计工作的集成开发系统作为 FPG 开发系统的首选。表 1-1 是 Xilinx 公司的 ISE 系列和 Altera 公司的 Quartus II 系列的性能对比表，从表 1-1 中可以看出两种集成系统的设计流程基本相同，只是各自采用的具体程序不同，初学用户可以根据自己的需要选择一种作为入门系统来学习，一旦掌握一种系统，很容易学习其他系统。笔者认为 Altera 公司的 Quartus II 系统提供了较多与 ISE 系列中小规模器件完全相同的设计元件，对于初学者更容易开始设计；Xilinx 公司的 ISE 系统提供了较好图形导航界面，对于初学者更容易掌握软件的使用。

表 1-1 Xilinx ISE 系列与 Altera Quartus II 系列性能对比表

性能对比	开发系统	Xilinx ISE 系列	Altera Quartus II 系列
设计输入		ISE	Quartus II
设计综合		ISE	Quartus II
仿真工具		ISE	Quartus II

## 图 1-2 数字系统设计实验系统选择

无论集成开发系统功能如何强大，仿真功能如何准确，为了更快更好地学习和使用 FPG，都需要有一个开发实验平台。实验平台根据其性能可以分为两类：评估平台和实验系统。评估平台是用来对某一个具体的器件的功能进行验证的实验平台，评估板大多占用 FPG 的引脚资源，用户只能在此硬件约束的条件下，对器件验证和评估，而不能进行其他设计。在这类评估平台上，只可以完成特定的实验验证，例如各厂商提供的 FPG 评估板就属于这一类平台。实验系统则是一种开放的平台。所谓开放的实验平台是与各种评估板相对而言的。图 1-2 所示为一个开放式的实验系统(多功能 FPG 实验系统)。从图 1-2 中可以看出，该系统由主板和子板构成，用户通过更换不同类型子板，就

可以实现对不同厂家的不同类型芯片进行设计开发。该实验系统，还提供了丰富的辅助资源，以及单片机调试接口。在这样系统中，不仅可以对 **悦臻电子** 进行设计，还可以利用单片机开发系统方便地进行 **悦臻电子** 与单片机之间数据传输的设计。另外，多功能 **秘码** 实验系统还应该提供独立的下载编程电缆，并且编程接口与各公司所提供的编程电缆接口完全相同，这样可以为用户的目标系统提供器件编程服务。

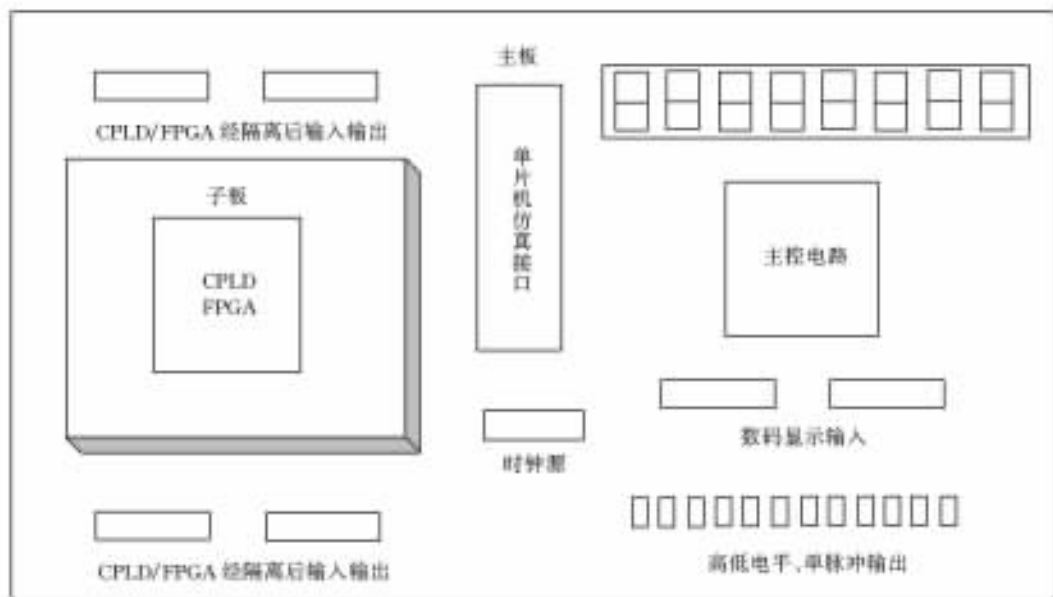


图 15-15 远程开放式实验系统结构框图

## 远程数字系统设计 EDA 开发流程

根据自己的需要选择了器件、开发系统和实验系统系统后，就可以开始 **秘码** 设计了。利用集成开发系统进行 **秘码** 设计的流程如图 15-16 所示。一般可分为设计输入，设计综合和器件编程三个设计步骤。设计输入由设计者完成，作为最原始的设计输入。设计综合包括优化、合并、映射、布局、布线等工作，并产生最终的下载数据文件。器件编程是将设计输出的数据文件通过下载电缆写入可编程器件内部，使其实现设计的功能。**秘码** 开发系统同时还具有功能仿真和时序仿真的功能，通常情况下，这些仿真结果是可以信赖的，开发系统一般以最坏的情况给出仿真结果，实际的工作情况比仿真结果会好一些。为了提高设计效率，必须正确掌握功能仿真和时序仿真的方法，准确理解仿真结果，以便及时修改设计，而不必采用反复下载方式来验证设计结果。图 15-16 中的虚拟器件测试是 **秘码** 设计最后一步，是在 **秘码** 实验系统或目标系统中完成的。

在整个设计过程中，设计输入是最关键的，是由设计者对器件所实现的数字系统的逻辑功能进行描述阶段。设计输入一般有电路图输入法、真值表输入法、状态机输入法、波形输入法、硬件描述语言输入法等，其中电路图输入和硬件描述语言输入是最常用的两种输入形式。

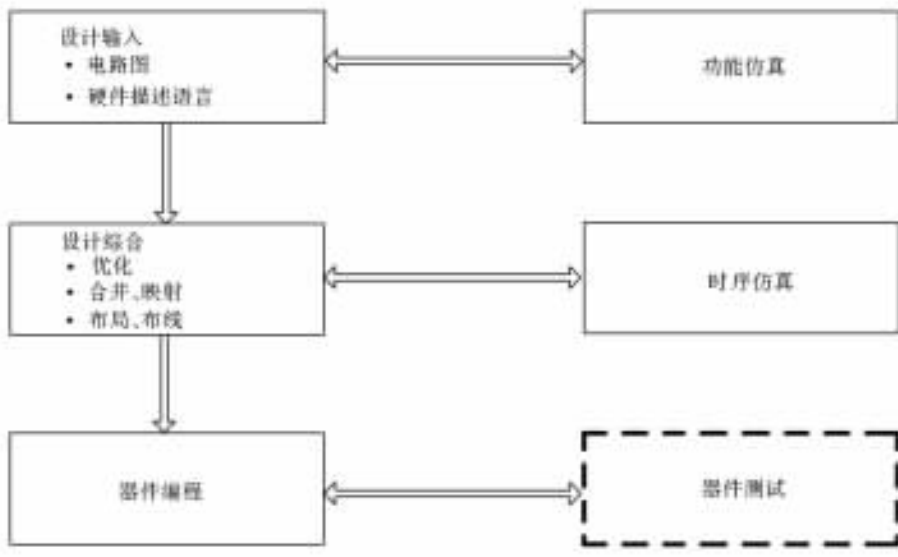


图 1-1 电路设计流程

电路图输入法是把数字系统用逻辑图来表示的输入方法，使用元件符号和连线等来描述设计。电路图描述要求设计工具提供必要的元件库和逻辑宏单元库。这种输入方法具有条理清楚、易学、易用等优点。比较适用于初学者使用和较小的系统设计。几乎所有的可编程器件开发系统都支持这种输入方法。电路图输入法具有可移植性，但可读性较差，不便于交流。

硬件描述语言是用文本形式描述设计，常用的硬件描述语言有 VHDL、Verilog 和 SystemC。VHDL 支持布尔方程、真值表、状态机等逻辑表达方式。适合对计数器、译码器、比较器和状态机等逻辑功能的描述。Verilog 和 SystemC 在描述复杂设计时，非常简洁，具有很强的逻辑描述功能，适合设计比较复杂的数字系统。

使用硬件描述语言实现数字系统设计，往往采用层次化设计方法，将系统化分为模块、分层次地进行设计描述。描述系统总功能的设计为顶层设计，描述系统中较小单元的设计为底层设计。层次化设计方法比较灵活，可以在任何层次使用电路图或硬件描述语言进行描述。根据两种输入方式的特点，通常在顶层设计中采用电路图输入方式，而在底层设计中采用硬件描述语言描述模块的逻辑功能。

## 数字系统设计实验学时和内容安排

本书作为数字系统设计课程的实验配套教材，可以根据课程的学时安排选择不同内容作为实验要求。由于数字系统设计是一门实践性较强的课程，一般安排总学时的 1/3 即可。

本书的第二章为基本实验内容，主要用于学生学习基本数字逻辑单元实验，每个实验可以安排 2 学时。一般可以根据需要选择 8 个实验，占用 16 学时。

本书的第三章为综合设计性实验，每个实验需要 4 学时，可以根据需要选择若干

个实验内容；如果需要完成较高要求的内容，可能需要 12~16 学时，可以作为课程设计和生产实习内容。

本书的第四章和第五章作为软件开发系统的入门，一般可以安排 16~20 学时完成，可以根据实验需求选择一种或两种开发系统作为实验内容学习。如果学生有计算机，可以将这部分内容作为实验预习内容。

本书的第六章为实验系统使用说明，主要掌握实验系统的接口连接方式和引脚对应关系，可以不安排学时学习，学生做下载测试时可在下载编程的同时进行学习。

# 第二章常用数字逻辑单元实验

## 实验一 译码电路设计实验

### 一、实验目的

- 1. 学习原理图设计方法。
- 2. 学习设计仿真方法。
- 3. 学习 开发系统使用方法。
- 4. 学习原理图编辑器使用方法。
- 5. 学会设计优先编码器。

### 二、实验原理

#### 1. 编码器定义

编码是对一系列二值信息中的每一信息赋予一特定代码的过程，实现编码的数字电路称做编码器。

#### 2. 编码器类型

常用的编码器有二进制编码器、十进制编码器、优先编码器。其中优先编码器经常用于具有优先级处理的数字系统中。例如中断管理系统通常是用优先编码器实现的。图 2-1 所示为 4 线—2 线优先编码器的真值表如表 2-1 所示。

表 2-1 4 线—2 线优先编码器的真值表

输入								输出				
$\bar{A}_3$	$\bar{A}_2$	$\bar{A}_1$	$\bar{A}_0$	$A_3$	$A_2$	$A_1$	$A_0$	$Y_2$	$Y_1$	$Y_0$	$Z$	$\bar{Z}$
1	1	1	1	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1	1	1	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	1	0	0	0	0	1
1	1	0	0	1	0	0	0	0	0	0	0	1
1	1	0	0	1	0	1	0	0	0	0	0	1
1	1	0	0	1	1	0	0	0	0	0	0	1
1	1	0	0	1	1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0	0	0	0	1
1	1	0	1	0	0	0	1	0	0	0	0	1
1	1	0	1	1	0	0	0	0	0	0	0	1
1	1	0	1	1	0	1	0	0	0	0	0	1
1	1	0	1	1	1	0	0	0	0	0	0	1
1	1	0	1	1	1	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0	0	0	0	0	1
1	0	0	0	1	1	0	0	0	0	0	0	1
1	0	0	0	1	1	1	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1	0	0	0	0	1
1	0	0	1	1	0	0	0	0	0	0	0	1
1	0	0	1	1	0	1	0	0	0	0	0	1
1	0	0	1	1	1	0	0	0	0	0	0	1
1	0	0	1	1	1	1	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	1	0	0	0	0	1
1	0	1	0	1	0	0	0	0	0	0	0	1
1	0	1	0	1	0	1	0	0	0	0	0	1
1	0	1	0	1	1	0	0	0	0	0	0	1
1	0	1	0	1	1	1	0	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0	0	0	1
1	0	1	1	0	0	0	1	0	0	0	0	1
1	0	1	1	1	0	0	0	0	0	0	0	1
1	0	1	1	1	0	1	0	0	0	0	0	1
1	0	1	1	1	1	0	0	0	0	0	0	1
1	0	1	1	1	1	1	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	1	0	1	0	0	0	0	0	1
0	0	0	0	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	0	0	1
0	0	0	1	1	0	0	0	0	0	0	0	1
0	0	0	1	1	0	1	0	0	0	0	0	1
0	0	0	1	1	1	0	0	0	0	0	0	1
0	0	0	1	1	1	1	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	0	0	0	0	0	0	1
0	0	1	0	1	0	1	0	0	0	0	0	1
0	0	1	0	1	1	0	0	0	0	0	0	1
0	0	1	0	1	1	1	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	1	0	0	0	0	1
0	0	1	1	1	0	0	0	0	0	0	0	1
0	0	1	1	1	0	1	0	0	0	0	0	1
0	0	1	1	1	1	0	0	0	0	0	0	1
0	0	1	1	1	1	1	0	0	0	0	0	1
0	0	1	1	1	1	1	1	0	0	0	0	1
0	0	1	1	1	1	1	1	1	0	0	0	1
0	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0

### 三、实验内容及步骤

1. 使用原理图输入方法设计一个三位二进制编码器。

2. 使用原理图输入方法设计一个 74148 优先编码器。

3. 仿真上述两个原理图，给出仿真波形。

4. 利用所设计的 74148 原理图，扩展一个 16 线 4 线优先编码器。

### 四、设计举例

1. 在 Quartus II 系统下，设计的 74148 原理图如图 4-10 所示。

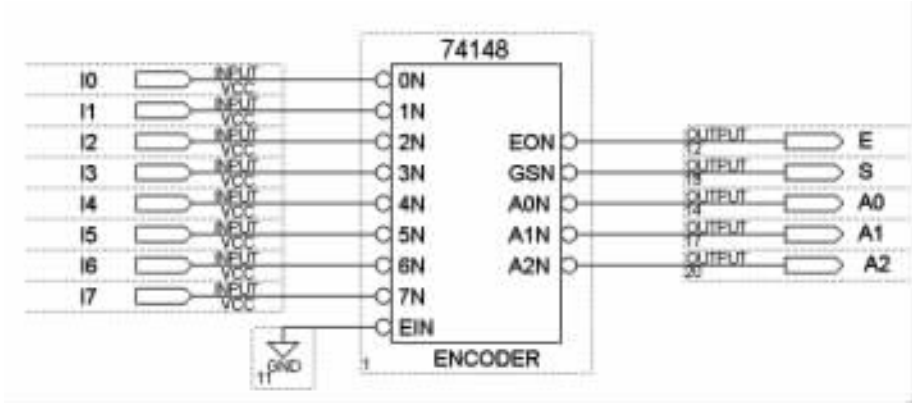


图 4-10 74148 优先编码器 (Quartus II 系统下)

2. 在 Quartus II 系统下，设计的 16 线 4 线优先编码器原理图，如图 4-11 所示。

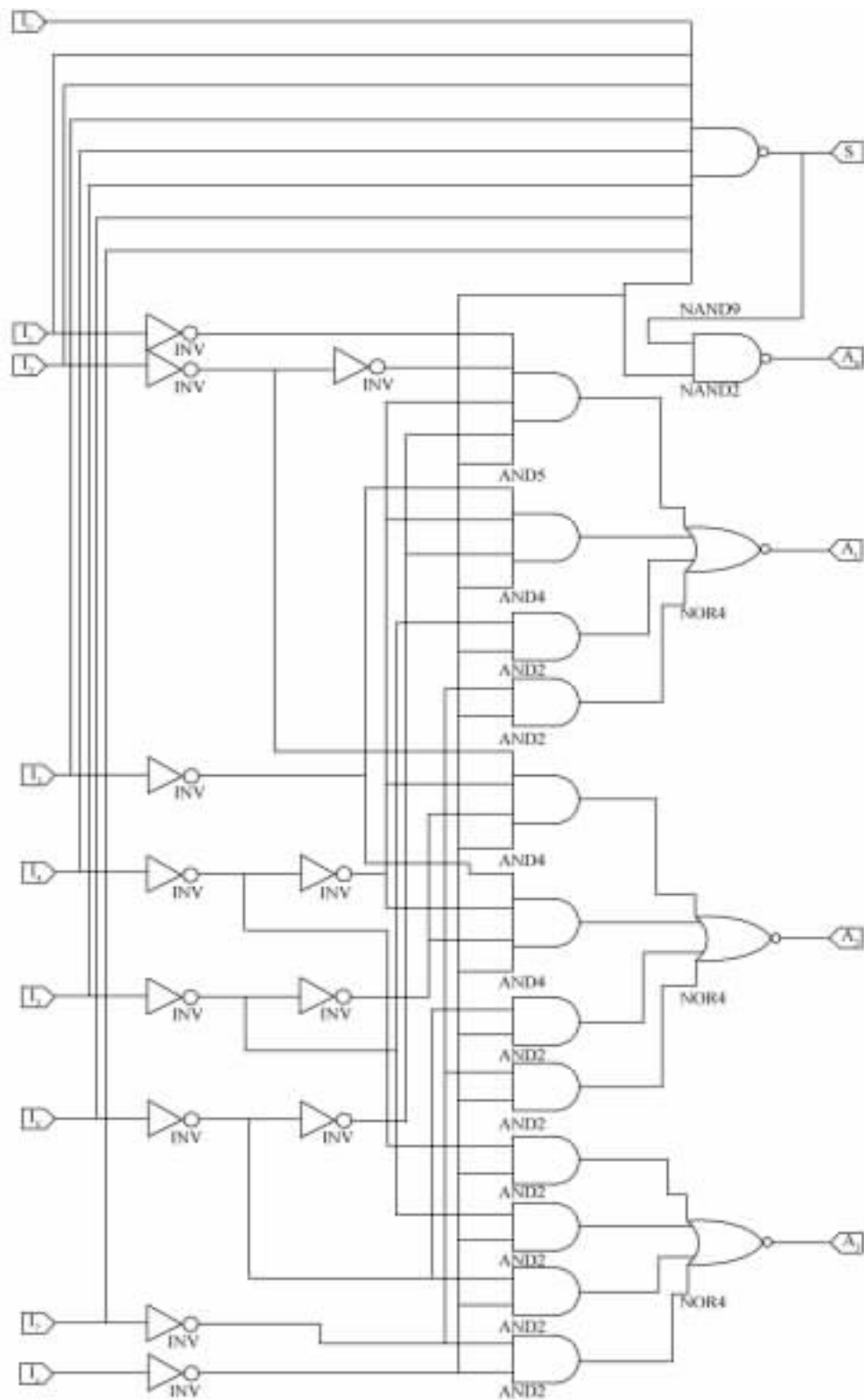


图 10.1.1 4 位优先编码器 (云梯牌 74LS148 芯片系统下)

# 实验二 译码电路设计实验

## 一、实验目的

1. 复习二进制译码器的功能。

2. 学习 74138 译码器语言源程序输入方法。

3. 学习 74138 译码器语言源程序检查和修改。

4. 掌握用 74138 译码器设计一个 3-8 译码器的方法。

5. 掌握 74138 译码器语言编辑器的基本操作。

## 二、实验原理

译码为编码的逆过程。它将编码时赋予代码的含义“翻译”过来。实现译码的逻辑电路称为译码器。译码器输出与输入代码有唯一的对应关系。常用的译码器有二进制译码器、二十进制译码器、七段显示译码器等。

### 3-8 译码器

3-8 译码器是二进制译码器的一种。其输入为一组三位二进制代码，而输出则对应是一路高、低电平信号。图 2-1 示出了 3-8 译码器的逻辑图。其中 A, B, C 为输入，Y0~Y7 为输出。

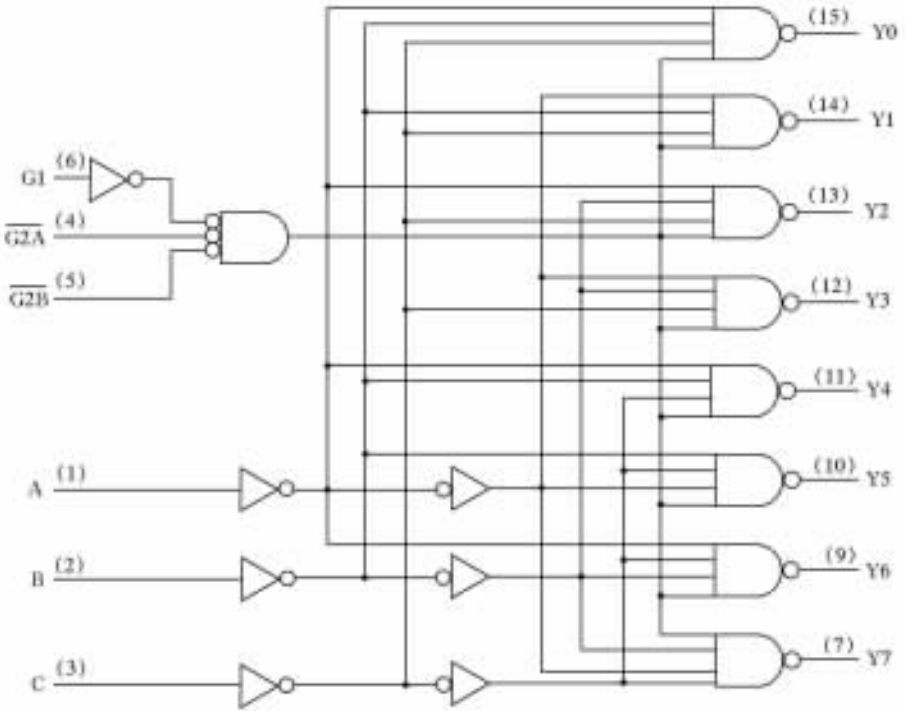


图 2-1 3-8 译码器