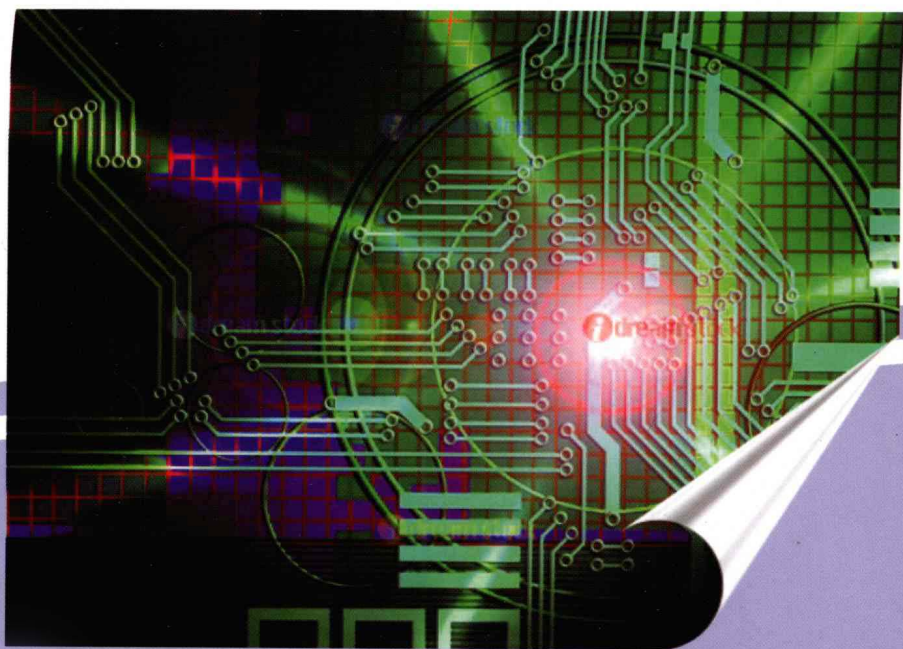


理工科电子信息类 **DIY** 系列丛书

硬件描述语言 实验指导

● 主编 曲波



理工科电子信息类 DIY 系列丛书

硬件描述语言实验指导

主 编 曲 波
副主编 黄秋萍 金慧敏

苏州大学出版社

图书在版编目(CIP)数据

硬件描述语言实验指导/曲波主编. —苏州:苏州大学出版社,2004.10

(理工科电子信息类 DIY 系列丛书)

ISBN 7-81090-332-2

I. 硬… II. 曲… III. 硬件描述语言—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 084257 号

内容简介

本书是硬件描述语言 VHDL 和 Verilog-HDL 的配套实验指导书。全书分三个部分:第一部分是基础验证实验,每个实验都给出了完整的 VHDL 和 Verilog-HDL 两个参考程序。目的是让读者掌握硬件描述语言程序设计的基本思想和方法,熟悉硬件描述语言的上机环境。第二部分为设计部分,每个实验项目都由多个程序文件组成,目的是让读者掌握模块化程序设计的思想和方法,提高分析问题和解决问题的能力。第三部分为综合部分,每个实验都涉及了与外部器件的连接,如 A/D、D/A、PS/2 键盘等,通过接口控制程序的编写,提高读者的综合应用能力。书中的附录部分给出了两种实验教学系统的使用说明和 MAXplusII 与 ispEXPERT 的使用指导。

书中的每一个实验都有明确的实验目的、任务和要求,并且给出了一种设计提示。本书既可作为学习硬件描述语言 VHDL 和 Verilog-HDL 的上机操作指导书,也可作为学习 VHDL 和 Verilog-HDL 语言的参考书,或作为教师的参考书。

硬件描述语言实验指导

曲 波 主编

责任编辑 薛华强

苏州大学出版社出版发行

(地址:苏州市干将东路 200 号 邮编:215021)

常熟高专印刷厂印装

(地址:常熟市元和路 98 号 邮编:215500)

开本 787mm×1092mm 1/16 印张 10.5 字数 248 千

2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

ISBN 7-81090-332-2/TP·25(课) 定价:18.00 元

苏州大学版图书若有印装错误,本社负责调换

苏州大学出版社营销部 电话:0512-67258835

电工电子实验教材编委会

主 编：胡人杰

副主编：赵鹤鸣 汪一鸣

编 委：赵德安 仲嘉霖 杨季文 翁桂荣

邹丽新 徐大诚 周 江

序 言

为了进一步加强高等学校综合实力,整合和共享教学资源,充分发挥理工科专业实验实践教学环节对于创新型人才培养的重要作用,从根本上提高本科教学的质量,提高大学生继续深造及创业就业的竞争力,近年来,各大学纷纷建立面向全校甚至所在城市的校级实验教学中心。实验中心的建立,统一和规范了实验仪器配备、实验教学要求,加强了实验教学的人才和梯队建设。

实验教学质量的提高,也离不开实验教材的建设。在苏州大学出版社的大力支持下,苏州大学电工电子实验教学中心学术委员会聘请校内外专家,成立了电工电子实验教材编委会,将陆续组织出版一套理工科电子信息类 DIY 系列丛书。这套丛书包括《数字电子技术实验指导》、《电子线路实验指导》、《硬件描述语言实验指导》、《电路与信号系统实验指导》、《微型计算机原理及应用实验指导》、《电工电子技术基础实验指导》等共六本。丛书将着力考虑符合实验教学大纲的要求,同时对不同专业留有充分的选择余地,可以灵活组合。希望丛书的出版能够改变实验教学不够规范,随意性较大,相同的课程分散在多处做实验,要求不同、过程不同、设备不一的现状。

编委会将在丛书出版之后,广泛听取各方面的反映和意见,不断完善丛书的内容,提高丛书的学术水平和质量,更好地满足高等学校电工电子类实验教学的需求。

电工电子实验教材编委会

2004. 6

前 言

随着现代电子技术的迅速发展,数字系统的硬件设计正朝着速度快、体积小、容量大、重量轻的方向发展。推动该潮流迅猛发展的就是日趋进步和完善的 ASIC (Application Specific Integrated Circuit) 技术。目前,数字系统的设计可以直接面向用户需求,根据系统的行为和功能要求,自上而下地逐层完成相应的描述、综合、优化、仿真与验证,直至生成器件系统。其中绝大部分设计过程可以通过计算机自动完成,也就是做到了电子设计自动化(EDA: Electronic Design Automation)。

EDA 的关键技术之一就是要求用形式化方法来描述数字系统的硬件电路,即用硬件描述语言来描述硬件电路。我国许多高校在相关的专业教学中设有硬件描述语言即 VHDL 或 Verilog 课程,并建有 EDA 实验室。硬件描述语言以其强大的系统描述能力、规范的程序设计结构、灵活的语句表达风格和多层次的仿真测试手段,已成为软、硬件工程师们的共同语言,也是电子信息类专业的学生必须要掌握的知识。

本书共分三章,第一章是基础验证实验,每个实验都给出了完整的 VHDL (Very High speed integrated circuit hardware Description Language) 和 Verilog - HDL 两个参考程序,通过这些实验使读者掌握 VHDL 或 Verilog 硬件描述语言程序设计的基本思想和方法。第二章主要是设计型实验,通过这些实验来掌握数字系统的设计方法,使读者掌握模块化程序设计的思想和方法,提高分析问题和解决问题的能力。第三章是综合型实验,每个实验都涉及了与外部器件的连接,如 A/D、D/A、PS/2 键盘等,通过这些实验来提高读者设计数字系统的能力和综合应用能力。

书中的每个实验都给出了一个设计提示和参考方案,这些方案只是许多方案中的一种,仅供参考,读者可以自己设计其他的方案。

本书在编写过程中引用了诸多学者和专家的著作与研究成果,在这里向他们表示衷心的感谢。另外,姚庆明、张洁颖、陈婷玉等几位同学对本书中的一些程序进行了验证,为本书的编写做了许多工作,在此也表示感谢。

由于作者水平有限,且时间仓促,错误和不当之处敬请读者不吝赐教。

编 者
2004 年 6 月

Contents

目录

第一章 基础验证实验

- 实验 1 译码器及译码显示电路设计 (1)
- 实验 2 编码器及数据选择器设计 (5)
- 实验 3 加法器设计 (9)
- 实验 4 乘法器设计 (12)
- 实验 5 触发器设计 (19)
- 实验 6 寄存器设计 (24)
- 实验 7 移位寄存器设计 (27)
- 实验 8 计数器设计 (30)
- 实验 9 可预置数的计数器设计 (33)
- 实验 10 模可变 16 位计数器设计 (36)
- 实验 11 七人表决器设计 (39)
- 实验 12 序列检测器设计 (41)

第二章 设计型实验

- 实验 1 数字秒表设计 (44)
- 实验 2 频率计设计 (45)
- 实验 3 多功能数字钟设计 (47)
- 实验 4 彩灯控制器设计 (49)
- 实验 5 交通灯控制器设计 (50)
- 实验 6 可控脉冲发生器设计

- (51)
- 实验 7 数控脉宽可调信号发生器设计 (52)
- 实验 8 密码锁设计 (53)
- 实验 9 出租车计费器设计 (54)
- 实验 10 VGA 彩色信号发生器设计 (55)

第三章 综合型设计

- 实验 1 万年历设计 (57)
- 实验 2 数字电压表设计 (57)
- 实验 3 波形发生器设计 (59)
- 实验 4 PS/2 键盘接口逻辑设计 (60)
- 实验 5 可编程并行 I/O 接口设计(8255) (60)

附录 1 GW48 - CK 型 EDA 教学实验系统使用说明

- 1.1 概述 (62)
- 1.2 系统主板结构与使用方法 (63)
- 1.3 实验电路结构图说明及使用方法 (68)
- 1.4 GW48 - CK 系统结构图信号名与芯片引脚对照表

.....	(79)		(112)
■.....		3.5 图形的层次化设计及 BUS 使用	(114)
附录 2 NH - TIV 型 EDA 实验开发 系统使用说明		3.6 语言描述输入法	(119)
2.1 概述	(83)	3.7 混合设计输入	(120)
2.2 系统主板结构	(83)	3.8 LPM 使用及 FLEX10K 中的 RAM 使用	(121)
2.3 下载板结构和使用方法	(84)	3.9 常见错误及处理方法 ...	(125)
2.4 实验板功能介绍	(85)	■.....	
2.5 单片机部分	(89)	附录 4 ispLEVER 3.0 使用指导	
2.6 管脚锁定	(91)	4.1 概述	(128)
2.7 跳线、地址开关使用说明	(93)	4.2 原理图输入	(128)
■.....		4.3 设计的编译与仿真	(134)
附录 3 MAX + plusII 使用指导		4.4 ABEL 语言和原理图混合 输入	(143)
3.1 概述	(94)	4.5 ispEXPERT 系统中 VHDL 和 Verilog 语言的设计方法	(150)
3.2 MAX + plusII 的设计过程	(94)	4.6 在系统编程的操作方法	(153)
3.3 图形输入的设计过程	(95)	主要参考文献	(158)
3.4 工具条和常用菜单选项说明			



第一章 基础验证实验

实验1 译码器及译码显示电路设计

一、实验目的

1. 熟悉硬件描述语言软件的编辑、仿真、下载等的使用。
2. 学习译码器和七段译码显示电路的设计。

二、实验原理

1. 译码器是数字系统中常用的组合逻辑电路,常用于地址译码。74LS138 是最常用的一种小规模集成电路,它有 3 个二进制输入端和 8 个译码输出端。表 1.1 是它的真值表。

表 1.1 3-8 译码器真值表

选通输入			二进制输入			译码输出							
G1	G2A	G2B	D2	D1	D0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

2. 普通的 LED 数码管由 7 段和一个点组成,使用它进行显示,需要译码驱动。本实验实现一个 7 段 LED 显示译码电路。为了实验方便,在译码之前加入一个四位二进制加法计数器,当低频率的脉冲信号输入计数器后,由 7 段译码器将计数值译为对应的十进制码,并由数码管显示出来。图 1.1 为此电路的原理图。

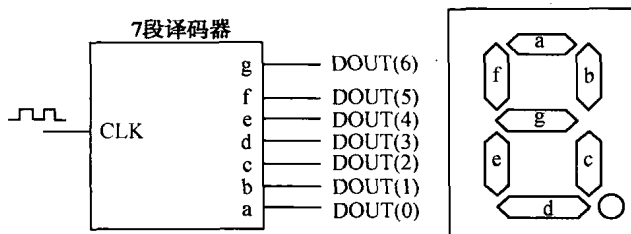


图 1.1 7 段 LED 译码显示电路



*** 三、实验内容

1. 设计一个 4-16 译码器。
2. 设计轮流显示表 1.2 所示字符的程序。

表 1.2 字母显示真值表

段 字符	a	b	c	d	e	f	g
A	1	1	1	0	1	1	1
B	0	0	1	1	1	1	1
C	1	0	0	1	1	1	0
D	0	1	1	1	1	0	1
E	1	0	0	1	1	1	1
F	1	0	0	0	1	1	1
H	0	1	1	0	1	1	1
P	1	1	0	0	1	1	1
L	0	0	0	1	1	1	0

3. 通过仿真,观察设计的正确性。
4. 下载、验证设计的正确性。

*** 四、设计提示

对于字符轮流显示,可以通过计数器计数进行显示,也可以通过状态机的编码方式来实现。

*** 五、实验报告要求

1. 分析电路的工作原理。
2. 写出所有的源程序。
3. 画出仿真波形。
4. 书写实验报告时要结构合理,层次分明,在分析描述的时候,注意语言的流畅。

*** 六、参考程序

1. 3-8 译码器的 VHDL 参考程序。

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY decoder3_8 IS
    PORT( d0, d1, d2, g1, g2a, g2b :IN STD_LOGIC;
          Y : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END decoder3_8;
ARCHITECTURE rtl OF decoder3_8 IS

```



```
SIGNAL indata : STD_LOGIC_VECTOR(2 DOWNTO 0);
BEGIN
```

```
    indata <= d2 & d1 & d0;
```

```
    PROCESS(indata, g1, g2a, g2b)
```

```
    BEGIN
```

```
        IF (g1 = '1' AND g2b = '0' AND g2a = '0') THEN
```

```
            CASE indata IS
```

```
                WHEN "000" => Y <= "11111110";
```

```
                WHEN "001" => Y <= "11111101";
```

```
                WHEN "010" => Y <= "11111011";
```

```
                WHEN "011" => Y <= "11110111";
```

```
                WHEN "100" => Y <= "11101111";
```

```
                WHEN "101" => Y <= "11011111";
```

```
                WHEN "110" => Y <= "10111111";
```

```
                WHEN "111" => Y <= "01111111";
```

```
                WHEN OTHERS => NULL;
```

```
            END CASE;
```

```
        ELSE
```

```
            Y <= "11111111";
```

```
        END IF;
```

```
    END PROCESS;
```

```
END rtl;
```

2. 7 段译码显示电路的 VHDL 参考程序。

```
LIBRARY IEEE;
```

```
USE IEEE.STD_LOGIC_1164.ALL;
```

```
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
ENTITY decled IS
```

```
    PORT (clk:IN STD_LOGIC;
```

```
          DOUT:OUT STD_LOGIC_VECTOR(6 DOWNTO 0));
```

```
END DECLED;
```

```
ARCHITECTURE behav OF decled IS
```

```
    SIGNAL cnt4b : STD_LOGIC_VECTOR(3 DOWNTO 0);
```

```
BEGIN
```

```
    PROCESS (clk)
```

```
    BEGIN
```

```
        IF clk'EVENT AND clk = '1' THEN
```

-- 4 位二进制计数器

```
            cnt4b <= cnt4b + 1;
```

```
        END IF;
```

```
    END PROCESS;
```



```
PROCESS( cnt4b)
BEGIN
    CASE cnt4b IS
        -- 译码电路
        WHEN "0000" => DOUT <= "0111111"; -- 显示 0
        WHEN "0001" => DOUT <= "0000110"; -- 显示 1
        WHEN "0010" => DOUT <= "1011011"; -- 显示 2
        WHEN "0011" => DOUT <= "1001111"; -- 显示 3
        WHEN "0100" => DOUT <= "1100110"; -- 显示 4
        WHEN "0101" => DOUT <= "1101101"; -- 显示 5
        WHEN "0110" => DOUT <= "1111101"; -- 显示 6
        WHEN "0111" => DOUT <= "0000111"; -- 显示 7
        WHEN "1000" => DOUT <= "1111111"; -- 显示 8
        WHEN "1001" => DOUT <= "1101111"; -- 显示 9
        WHEN "1010" => DOUT <= "1110111"; -- 显示 A
        WHEN "1011" => DOUT <= "1111100"; -- 显示 B
        WHEN "1100" => DOUT <= "0111001"; -- 显示 C
        WHEN "1101" => DOUT <= "1011110"; -- 显示 D
        WHEN "1110" => DOUT <= "1111001"; -- 显示 E
        WHEN "1111" => DOUT <= "1110001"; -- 显示 F
        WHEN OTHERS => DOUT <= "0000000"; -- 必须有此项
    END CASE;
END PROCESS;
END behav;
```

3. 3-8 译码器 verilog HDL 参考程序。

```
module Decoder3_8(G1,G2A,G2B, D, Y);
output[7:0] Y;
input[2:0] D;
input G1,G2A,G2B;
assign Y[0] = ~((G1)&( ~G2A)&( ~G2B)&( ~D[2])&( ~D[1])&( ~D[0]));
assign Y[1] = ~((G1)&( ~G2A)&( ~G2B)&( ~D[2])&( ~D[1])&(D[0]));
assign Y[2] = ~((G1)&( ~G2A)&( ~G2B)&( ~D[2])&(D[1])&( ~D[0]));
assign Y[3] = ~((G1)&( ~G2A)&( ~G2B)&( ~D[2])&(D[1])&(D[0]));
assign Y[4] = ~((G1)&( ~G2A)&( ~G2B)&(D[2])&( ~D[1])&( ~D[0]));
assign Y[5] = ~((G1)&( ~G2A)&( ~G2B)&(D[2])&( ~D[1])&(D[0]));
assign Y[6] = ~((G1)&( ~G2A)&( ~G2B)&(D[2])&(D[1])&( ~D[0]));
assign Y[7] = ~((G1)&( ~G2A)&( ~G2B)&(D[2])&(D[1])&(D[0]));
endmodule
```

4. 7 段译码显示电路的 verilog HDL 参考程序。

```
module Decoder4_7 (EN,clock,a, b, c, d, e, f,g);
```



```

input EN, clock;
output a, b, c, d, e, f, g;
reg [4:1] in;
reg a, b, c, d, e, f, g;
always @ (posedge clock)
if ( !EN)
    in = 0;
else begin in = in + 1;                                //4 位二进制计数
    case(in)                                           //译码电路
        4'b0000: {g,f,e,d,c,b,a} = 8'b0111111;      //显示 0
        4'b0001: {g,f,e,d,c,b,a} = 8'b0000110;      //显示 1
        4'b0010: {g,f,e,d,c,b,a} = 8'b1011011;      //显示 2
        4'b0011: {g,f,e,d,c,b,a} = 8'b1001111;      //显示 3
        4'b0100: {g,f,e,d,c,b,a} = 8'b1100110;      //显示 4
        4'b0101: {g,f,e,d,c,b,a} = 8'b1101101;      //显示 5
        4'b0110: {g,f,e,d,c,b,a} = 8'b1111101;      //显示 6
        4'b0111: {g,f,e,d,c,b,a} = 8'b0000111;      //显示 7
        4'b1000: {g,f,e,d,c,b,a} = 8'b1111111;      //显示 8
        4'b1001: {g,f,e,d,c,b,a} = 8'b1101111;      //显示 9
        4'b1010: {g,f,e,d,c,b,a} = 8'b1110111;      //显示 A
        4'b1011: {g,f,e,d,c,b,a} = 8'b1111100;      //显示 B
        4'b1100: {g,f,e,d,c,b,a} = 8'b0111001;      //显示 C
        4'b1101: {g,f,e,d,c,b,a} = 8'b1011110;      //显示 D
        4'b1110: {g,f,e,d,c,b,a} = 8'b1111001;      //显示 E
        4'b1111: {g,f,e,d,c,b,a} = 8'b1110001;      //显示 F
        default: {g,f,e,d,c,b,a} = 8'b0000000;      //必须有此项
    endcase
end
endmodule

```

实验 2 编码器及数据选择器设计

* 一、实验目的

1. 进一步熟悉硬件描述语言软件的使用。
2. 学习编码器、选择器的设计。
3. 掌握组合逻辑电路的设计方法。



二、实验原理

74/54 系列 148/348 是优先编码器,低电平有效,图 1.2 是它的功能图,表 1.3 是它的真值表。

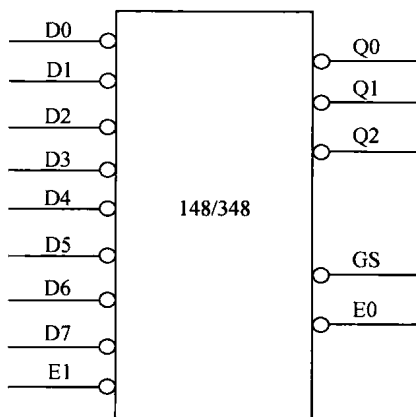


图 1.2 优先编码器功能图

表 1.3 优先编码器真值表

输 入									输 出				
E1	D0	D1	D2	D3	D4	D5	D6	D7	Q0	Q1	Q2	GS	E0
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

三、实验内容

1. 根据真值表,写出源程序。
2. 设计一个四选一选择器。
3. 仿真、下载验证设计的正确性。

四、设计提示

IF、CASE 语句是顺序语句,只可以在进程内部使用。



五、实验报告要求

1. 分析电路的工作原理。
2. 写出优先编码器、四选一选择器的源程序。
3. 比较顺序语句和并行语句的异同。
4. 画出仿真波形。

六、参考程序

1. 优先编码器 148/348 的 VHDL 参考程序。

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY prioryencoder IS
    PORT(d:IN STD_LOGIC_VECTOR (7 DOWNTO 0);
         E1:IN STD_LOGIC;
         GS, E0:OUT STD_LOGIC;
         Q:OUT STD_LOGIC_VECTOR(2 DOWNTO 0));
END prioryencoder;
ARCHITECTURE encoder OF prioryencoder IS
BEGIN
    P1: PROCESS (d)
    BEGIN
        IF (d(7) = '0' AND E1 = '0') THEN
            Q <= "000";
            GS <= '0';
            E0 <= '1';
        ELSIF (d(6) = '0' AND E1 = '0') THEN
            Q <= "001";
            GS <= '0';
            E0 <= '1';
        ELSIF (d(5) = '0' AND E1 = '0') THEN
            Q <= "010";
            GS <= '0';
            E0 <= '1';
        ELSIF (d(4) = '0' AND E1 = '0') THEN
            Q <= "011";
            GS <= '0';
            E0 <= '1';
        ELSIF (d(3) = '0' AND E1 = '0') THEN

```



```
Q <= "100";
GS <= '0';
E0 <= '1' ;
ELSIF(d(2) = '0' AND E1 = '0') THEN
Q <= "101";
GS <= '0';
E0 <= '1';
ELSIF(d(1) = '0' AND E1 = '0') THEN
Q <= "110";
GS <= '0';
E0 <= '1';
ELSIF(d(0) = '0' AND E1 = '0') THEN
Q <= "111";
GS <= '0';
E0 <= '1';
ELSIF(E1 = '1') THEN
Q <= "111";
GS <= '1';
E0 <= '1';
ELSIF(d = "11111111" AND E1 = '0') THEN
Q <= "111";
GS <= '1';
E0 <= '0';
END IF;
END PROCESS P1;
```

```
END encoder;
```

2. 优先编码器 148/348 的 verilog HDL 参考程序。

```
module encoder8_3(E1, D, Q,GS,E0);
output GS, E0;
output [0:2] Q;
input [0:7] D;
input E1;
reg [2:0] Q;
reg GS, E0;
always @ (D or E1)
begin
casex({E1,D[0:7]})
9'b1xxxxxxx: {Q,GS,E0} = 5'b11111;
9'b01111111: {Q,GS,E0} = 5'b11110;
```




```

9'b0xxxxxx0: {Q,GS,E0} = 5'b00001;
9'b0xxxxxx01: {Q,GS,E0} = 5'b00101;
9'b0xxxxxx011: {Q,GS,E0} = 5'b01001;
9'b0xxxxxx0111: {Q,GS,E0} = 5'b01101;
9'b0xxx01111: {Q,GS,E0} = 5'b10001;
9'b0xx011111: {Q,GS,E0} = 5'b10101;
9'b0x0111111: {Q,GS,E0} = 5'b11001;
9'b001111111: {Q,GS,E0} = 5'b11101;
default:      {Q,GS,E0} = 5'bxxxxx;
endcase
end
endmodule

```

实验3 加法器设计

一、实验目的

1. 学习利用组合电路设计加法器。
2. 掌握利用结构描述设计程序的方法。

二、实验原理

加法器是数字系统中的基本逻辑器件,多位加法器的构成有两种方式:并行进位和串行进位方式。并行进位加法器设有并行进位产生逻辑,运算速度较快;串行进位方式是将全加器级联构成多位加法器。并行进位加法器通常比串行级联加法器占用更多的资源,随着位数的增加,相同位数的并行加法器与串行加法器的资源占用差距快速增大。因此,在工程中使用加法器时,要在速度和容量之间寻找平衡。

表 1.4 是 1 位全加器的真值表,通过串行级联的方法可以构成多位全加器。

表 1.4 1 位全加器的真值表

输 入			输 出	
A	B	CI	CO	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1