# 实验2 简单组合电路设计

## 实验内容

实现一个“静态7段数码管显示”模块。

**要求：**

* SW[3:0]输入待显示的二进制数。
* SW[15:13]输入点亮的数码管编号（0-7号）。
* LED[15:0]显示SW的开关状态。



图1-1 实验最终结果示例

SW[15:13] = 010，SW[3:0] = 0110；2号数码管显示数字“6“。

## 实验目的

1. 掌握Verilog语言的组合逻辑电路设计、实现、仿真、调试方法。
2. 熟悉Nexys4开发板，学习、掌握编写引脚约束文件、FPGA编程过程。
3. 学习、掌握设计组合逻辑电路的方法，实现简单的组合逻辑电路。

## 实验步骤

## 直接驱动7段数码管（7-Seg Display）

**实验内容：**

* 使用“数据流描述”设计“7段数码管直接驱动”模块。

模块定义：

|  |
| --- |
| module \_7Seg\_Driver\_Direct(SW, CA, CB, CC, CD, CE, CF, CG, DP, AN, LED);  input [15:0] SW; // 16位拨动开关  output CA, CB, CC, CD, CE, CF, CG, DP; // 7段数码管驱动，低电平有效  output [7:0] AN; // 7段数码管片选信号，低电平有效  output [15:0] LED; // 16位LED显示，高电平有效  endmodule |

* 用SW[7:0]直接驱动7段数码管的CA-CG、DP显示单元。
* 用SW[15:8]选择被驱动的7段数码管。
* 用LED[15:0]显示SW的状态。

表3-1 拨动开关与7-Seg Display对应关系

|  |  |
| --- | --- |
| **拨动开关编号** | **7-Seg Display** |
| SW7 | CA |
| SW6 | CB |
| SW5 | CC |
| SW4 | CD |
| SW3 | CE |
| SW2 | CF |
| SW1 | CG |
| SW0 | DP |

**实验原理：**

* “7段数码管显示原理”请提前预习“数字逻辑”相关章节。
* 请参考“nexys4ddr\_rm.pdf”（Nexys 4 DDR Manual）第10节 —— Basic I/O。



图3-1 直接驱动7段数码管原理框图

**知识点：**

* 熟悉vivado开发环境的使用。
* 学习使用“数据流描述”设计一个模块。
* 掌握“引脚约束”文件的编写方法。
* 熟悉Nexys4的开发步骤。
* 了解7段数码管的工作原理和直接驱动方法。

## 译码驱动7段数码管

**实验内容：**

* 使用“行为描述”设计“7段数码管译码驱动”模块。

模块定义：

|  |
| --- |
| module \_7Seg\_Driver\_Decode(SW, SEG, AN, LED);  input [15:0] SW; // 16位拨动开关  output [7:0] SEG; // 7段数码管驱动，低电平有效  output [7:0] AN; // 7段数码管片选信号，低电平有效  output [15:0] LED; // 16位LED显示  endmodule |

* 用SW[3:0]输入待显示的单个数字。
* 输入数字与显示对应表如表3-2。
* 用SW[15:8]选择被驱动的7段数码管。
* 用LED[15:0]显示SW的状态。

表3-2 输入与7-Seg Display对应关系及显示码

|  |  |  |
| --- | --- | --- |
| **输入数字** | **7-Seg Display** | **显示码** |
| 0000 | 0 | 11000000 |
| 0001 | 1 | 11111001 |
| 0010 | 2 | 10100100 |
| 0011 | 3 | 10110000 |
| 0100 | 4 | 10011001 |
| 0101 | 5 | 10010010 |
| 0110 | 6 | 10000010 |
| 0111 | 7 | 11111000 |
| 1000 | 8 | 10000000 |
| 1001 | 9 | 10011000 |
| 1010 | A | 10001000 |
| 1011 | b | 10000011 |
| 1100 | C | 11000110 |
| 1101 | d | 10100001 |
| 1110 | E | 10000110 |
| 1111 | F | 10001110 |

**实验原理：**

* 尝试使用“case”结构，或者“if-else”结构。参见教材：p.122相关章节。



图3-2 译码驱动7段数码管原理框图

**知识点：**

* 学习使用“行为描述”设计一个模块。
* 了解7段数码管的译码驱动原理。

## 译码选择7段数码管

**实验内容：**

* 使用“行为描述”设计一个“7段数码管译码选择”模块。
* 使用“结构描述”连接“译码显示模块”和“译码选择模块”，完成最终的实验要求（第一章）。

模块定义：

|  |
| --- |
| module \_7Seg\_Driver\_Choice(SW, SEG, AN, LED);  input [15:0] SW; // 16位拨动开关  output [7:0] SEG; // 7段数码管驱动，低电平有效  output [7:0] AN; // 7段数码管片选信号，低电平有效  output [15:0] LED; // 16位LED显示  endmodule |

**实验原理：**



图3-3 译码选择7段数码管原理框图

**知识点：**

* 学习使用“结构描述”设计一个模块。
* 学习调用模块。

## 进阶——“二进制-BCD码转换器”

选做，请尝试完成以下功能：

**实验内容：**

* SW[3:0]输入待转换位BCD码的二进制数。
* SW[14:12]选择用几号数码管显示。
* SW[15]选择显示BCD码的十位还是个位。

SW[15] = 0：显示个位；

SW[15] = 1：显示十位。

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 05/15/2018 10:55:17 AM

// Design Name:

// Module Name: main

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module main(sw, seg, dp, an, led);

input [15:0] sw; //16位拨动开关

output [6:0] seg; //7段数码管驱动,低电平有效

output dp;

output [7:0] an; //7段数码管片选信号,低电平有效

output [15:0] led; //16位led显示,高电平有效

\_7Seg\_Driver\_Decode a(sw, seg, dp, an, led);

endmodule

//////////////////////////////////

// 模块功能：7段数码显示管直接驱动模块

//////////////////////////////////

module \_7Seg\_Driver\_Direct(sw, seg, dp, an, led);

input [15:0] sw; //16位拨动开关

output [6:0] seg; //7段数码管驱动,低电平有效

output dp;

output [7:0] an; //7段数码管片选信号,低电平有效

output [15:0] led; //16位led显示,高电平有效

assign led[15:0] = sw[15:0]; //将led赋值为相应的开关

assign seg[6:0] = sw[6:0];

assign dp = sw[7];

assign an[7:0] = ~sw[15:8]; //将sw[15:8]赋值给数码管片选信号AN[7:0]

endmodule

//////////////////////////////////

// 模块功能：译码驱动7段数码管

//////////////////////////////////

module \_7Seg\_Driver\_Decode(sw, seg, dp, an, led);

input [15:0] sw; //16位拨动开关

output [6:0] seg; //7段数码管驱动,低电平有效

output [7:0] an; //7段数码管片选信号,低电平有效

output [15:0] led; //16位led显示

output dp;

reg [6:0] seg;

reg [3:0] num;

reg dp;

assign led[15:0] = sw[15:0];

assign an[7:0] = ~sw[15:8];

initial

begin

num[3:0] = seg[3:0];

dp = 1'b1;

end

always

@(sw[15:0])begin

num[3:0] = seg[3:0];

case(num)

4 'b0000:

seg [6:0] = 7'b1000000;

4 'b0001:

seg [6:0] = 7'b1111001;

4 'b0010:

seg [6:0] = 7'b0100100;

4 'b0011:

seg [6:0] = 7'b0110000;

4 'b0100:

seg [6:0] = 7'b0011001;

4 'b0101:

seg [6:0] = 7'b0010010;

4 'b0110:

seg [6:0] = 7'b0000010;

4 'b0111:

seg [6:0] = 7'b1111000;

4 'b1000:

seg [6:0] = 7'b0000000;

4 'b1001:

seg [6:0] = 7'b0011000;

4 'b1010:

seg [6:0] = 7'b0001000;

4 'b1011:

seg [6:0] = 7'b0000011;

4 'b1100:

seg [6:0] = 7'b1000110;

4 'b1101:

seg [6:0] = 7'b0100001;

4 'b1110:

seg [6:0] = 7'b0000110;

4 'b1111:

seg [6:0] = 7'b0001110;

endcase

end

endmodule