**《脉冲电路与数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | **魏和琬 陈炫瑾** | | | **年级** | **2017** |
| **学号** | | **20174393 20174417** | | | **专业、班级** | **信安2班 物联1班** |
| **实验名称** | **七段数码管显示** | | | | | |
| **实验时间** | **5.17** | | **实验地点** | **DS1410** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 □设计性 □综合性** | | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  评语：  评价教师签名（电子签名）： | | | | | | |
| 一、实验目的  1、掌握 7 段数码管的显示原理，学会设计八进制数、十六进制数的七段数  码管显示；  2、学会使用显示模块和多路选择器设计组合逻辑电路。 | | | | | | |
| 二、实验项目内容  1、将拨码开关 SW3~SW0 以十六进制数显示在 1 个七段数码管上。  2、将拨码开关 SW11~SW9、SW8~SW6、SW5~SW3、SW2~SW0 分别以八进制数显示  在 4 个七段数码管上。  3、设计一个 3 位的 4 选 1 多路选择器，并利用拨码开关 SW15~SW14 控制选  择将拨码开关 SW11~SW9、SW8~SW6、SW5~SW3、SW2~SW0 以八进制数显示在  1 个七段数码管上。 | | | | | | |
| 三、实验过程或算法(包括真值表、原理图、关键步骤、核心代码）  共阳极七段数码管各个16进制数对应真值表    通过对应真值表，实现八、十六进制数显示模块,将拨码开关SW2~SW0、SW3~SW0 分别以八进制数、十六进制数显示在1 个七段数码管上。  实现一个 3 位的 4 选 1 多路选择器，并使所输出的值控制七段数码管，实现将 3 位的 4 选 1 多路选择器的 3 位输出接到八进制数显示模块上，最终实现利用拨码开关 SW15~SW14 控制选择将拨码开关SW11~SW9、SW8~SW6、SW5~SW3、SW2~SW0 以八进制数显示在 1 个七段数码管上。  **代码：**  **十六进制数显示模块：**  module Five\_1(sw0,sw1,sw2,sw3,segs,an);  input sw0;  input sw1;  input sw2;  input sw3;  output reg [6:0] segs;  output wire [7:0] an;  wire[3:0] i;  assign i[0]=sw0;  assign i[1]=sw1;  assign i[2]=sw2;  assign i[3]=sw3;  assign an[0]=0;  always@(\*)  case(i)  4'h0:segs = 7'b000\_0001;  4'h1:segs = 7'b100\_1111;  4'h2:segs = 7'b001\_0010;  4'h3:segs = 7'b000\_0110;  4'h4:segs = 7'b100\_1100;  4'h5:segs = 7'b010\_0100;  4'h6:segs = 7'b010\_0000;  4'h7:segs = 7'b000\_1111;  4'h8:segs = 7'b000\_0000;  4'h9:segs = 7'b000\_1100;  4'ha:segs = 7'b000\_1000;  4'hb:segs = 7'b110\_0000;  4'hc:segs = 7'b111\_0010;  4'hd:segs = 7'b100\_0010;  4'he:segs = 7'b011\_0000;  4'hf:segs = 7'b011\_1000;  default:segs = 7'b111\_1111;  endcase  endmodule  **八进制数显示模块：**  module Five\_1(sw0,sw1,sw2,sw3,segs,an);  input sw0;  input sw1;  input sw2;  input sw3;  output reg [6:0] segs;  output wire [7:0] an;  wire [2:0] oct;  assign an[0]=0;  assign oct[0]=sw0;  assign oct[1]=sw1;  assign oct[2]=sw2;  always@(\*)  case(oct)  3'h0: segs = 7'b000\_0001;  3'h1: segs = 7'b100\_1111;  3'h2: segs = 7'b001\_0010;  3'h3: segs = 7'b000\_0110;  3'h4: segs = 7'b100\_1100;  3'h5: segs = 7'b010\_0100;  3'h6: segs = 7'b010\_0000;  3'h7: segs = 7'b000\_1111;  default: segs = 7'b111\_1111;  endcase  endmodule  **3 位的 4 选 1 多路选择器的 3 位输出接到八进制数显示模块：**  **3 位的 4 选 1 多路选择器：**  module Change(oct0,oct1,oct2,oct3,ch,out);  input [2:0] oct0,oct1,oct2,oct3;  input [1:0] ch;  output out;  reg [0:2] out;  always@(ch)  begin  case(ch)  2'b00:out=oct0;  2'b01:out=oct1;  2'b10:out=oct2;  2'b11:out=oct3;  endcase  end  endmodule  **八进制数显示模块：**  module Five\_3(oct0,segs,an);  input [2:0]oct0;  output reg [6:0]segs;  output [7:0] an;  assign an[0]=0;  assign an[1]=1;  assign an[2]=1;  assign an[3]=1;  assign an[4]=1;  assign an[5]=1;  assign an[6]=1;  assign an[7]=1;  always@(oct0)  begin  case(oct0)  3'b000:segs = 7'b000\_0001;  3'b001:segs = 7'b100\_1111;  3'b010:segs = 7'b001\_0010;  3'b011:segs = 7'b000\_0110;  3'b100:segs = 7'b100\_1100;  3'b101:segs = 7'b010\_0100;  3'b110:segs = 7'b010\_0000;  3'b111:segs = 7'b000\_1111;  default:  segs = 7'b111\_1111;  endcase  end  endmodule  **顶层模块：**  module Top(oct0,oct1,oct2,oct3,ch,segs,an);  input [2:0] oct0,oct1,oct2,oct3;  input [1:0] ch;  output [6:0] segs;  output [7:0] an;  wire [2:0]out;  Change ch1(oct0,oct1,oct2,oct3,ch,out);  Five\_3 show(out,segs,an);  always@(\*)  begin    end  endmodule  **管脚：**  set\_property PACKAGE\_PIN U13 [get\_ports {an[7]}]  set\_property PACKAGE\_PIN K2 [get\_ports {an[6]}]  set\_property PACKAGE\_PIN T14 [get\_ports {an[5]}]  set\_property PACKAGE\_PIN P14 [get\_ports {an[4]}]  set\_property PACKAGE\_PIN J14 [get\_ports {an[3]}]  set\_property PACKAGE\_PIN T9 [get\_ports {an[2]}]  set\_property PACKAGE\_PIN J18 [get\_ports {an[1]}]  set\_property PACKAGE\_PIN J17 [get\_ports {an[0]}]  set\_property PACKAGE\_PIN V10 [get\_ports {ch[1]}]  set\_property PACKAGE\_PIN U11 [get\_ports {ch[0]}]  set\_property PACKAGE\_PIN M13 [get\_ports {oct0[2]}]  set\_property PACKAGE\_PIN L16 [get\_ports {oct0[1]}]  set\_property PACKAGE\_PIN J15 [get\_ports {oct0[0]}]  set\_property PACKAGE\_PIN T18 [get\_ports {oct1[2]}]  set\_property PACKAGE\_PIN R17 [get\_ports {oct1[1]}]  set\_property PACKAGE\_PIN R15 [get\_ports {oct1[0]}]  set\_property PACKAGE\_PIN T8 [get\_ports {oct2[2]}]  set\_property PACKAGE\_PIN R13 [get\_ports {oct2[1]}]  set\_property PACKAGE\_PIN U18 [get\_ports {oct2[0]}]  set\_property PACKAGE\_PIN T13 [get\_ports {oct3[2]}]  set\_property PACKAGE\_PIN R16 [get\_ports {oct3[1]}]  set\_property PACKAGE\_PIN U8 [get\_ports {oct3[0]}]  set\_property PACKAGE\_PIN T10 [get\_ports {segs[6]}]  set\_property PACKAGE\_PIN R10 [get\_ports {segs[5]}]  set\_property PACKAGE\_PIN K16 [get\_ports {segs[4]}]  set\_property PACKAGE\_PIN K13 [get\_ports {segs[3]}]  set\_property PACKAGE\_PIN P15 [get\_ports {segs[2]}]  set\_property PACKAGE\_PIN T11 [get\_ports {segs[1]}]  set\_property PACKAGE\_PIN L18 [get\_ports {segs[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {an[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {ch[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {ch[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct0[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct0[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct0[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct1[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct1[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct1[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct2[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct2[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct2[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct3[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct3[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {oct3[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {segs[0]}] | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程  拨码开关 SW3~SW0 以十六进制数显示在1 个七段数码管上。  拨码开关 SW2~SW0 以八进制数显示在 1 个七段数码管上    将 3 位的 4 选 1 多路选择器的 3 位输出接到八进制数显示模块上。  通过拨码开关sw14~15来选择拨码开关 SW11~SW9、SW8~SW6、SW5~SW3、SW2~SW0，从而显示出数字。    五、小组分工情况说明  魏和琬：内容讨论，代码和仿真部分完成。  陈炫瑾：内容讨论，状态转移图表绘制，脚管设置和上板。 | | | | | | |
| 六、个人总结  通过四选一选择器的原理来控制拨码开关，从而将八进制数显示在数码管上。运用顶层模块将两个选择器模块与显示模块结合。 | | | | | | |