课程报告

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** |  | **学号** |  |
| **实验时间** | 2023年11月24 | **实验地点** | 第一教研楼225 |
| **实验序号** | 13 | **实验名称** | 数码管扫描显示 |
| **课程名称** | 数字电路课程设计 | **指导教师** | 潘玉剑 |

# 方案设计

## 实验目的

1. 学习数码管的显示原理。掌握数码管动态扫描方法。

2. 掌握灵活运用 Verilog HDL 语言进行各种描述与建模的技巧与方法。

## 模块设计说明

**数码管扫描模块**

输入：clk（时钟信号），data（输入数据信号）

输出：which（片选信号），seg（段选信号）

**顶层模块**

输入：clk（时钟信号），SW（数据选择开关）

输出：which（片选信号），seg（段选信号），enable（使能信号）

**模块调用**

在顶层模块内预设数据信号 data1,data2,data3，通过 SW 数据选择开关决定数码管扫描模块对应的输入数据 data。

## 输入输出I/O

|  |  |  |
| --- | --- | --- |
|  | **信号名称** | **配置设备管脚** |
| 输入 | 时钟信号clk | HDU板卡时钟信号H4 |
| SW[1:0] | 2个逻辑开关，本代码配置的是T3,U3 |
| 输出 | 3-8译码器使能信号enable | L21 |
| 3-8译码器输入信号which[2:0] | M22,M21,N22 |
| 段选信号 seg[7:0] | H19,G20,J22,K22,  K21,H20,H22,J21 |

# 逻辑程序设计

## 逻辑程序源代码

|  |
| --- |
| module nixie\_tube(  input clk,  input [1:0]SW,  output [2:0]which,  output [7:0]seg,  output reg enable  );    reg [31:0]data1 = 32'b1000\_0111\_0110\_0101\_0100\_0011\_0010\_0001;  reg [31:0]data2 = 32'b0010\_0000\_0000\_0100\_0000\_0001\_0001\_0010;  reg [31:0]data3 = 32'b1000\_1111\_1110\_1101\_1100\_1011\_1010\_1001;  reg [31:0]data = 32'b0;    always @(\*)  begin  data = 32'b0;  case (SW)  2'b00: begin  data = 32'b0;  enable = 0;  end  2'b01: begin  enable = 1;  data=data1;  end  2'b10: begin  enable = 1;  data=data2;  end  2'b11: begin  enable = 1;  data=data3;  end  endcase  end  Display U1(clk,data,which,seg);  endmodule  module Display(  input clk, // 时钟信号  input [31:0] data, // 输入数据信号  output reg [2:0] which = 0,// 位选信号  output reg [7:0] seg // 段选信号  );  reg [14:0] count = 0; // 分频计数器  reg [2:0] which = 0;  always @(posedge clk) count <= count + 1'b1;  always @(negedge clk) if(&count)which <= which + 1'b1;    reg [3:0] data\_1; // 数据位1  always @(\*) begin  case (which) // 根据计数器的值选择数据位1的值  0: data\_1 = data[31:28];  1: data\_1 = data[27:24];  2: data\_1 = data[23:20];  3: data\_1 = data[19:16];  4: data\_1 = data[15:12];  5: data\_1 = data[11:8];  6: data\_1 = data[7:4];  7: data\_1 = data[3:0];  endcase  end  always @\* begin  case (data\_1)//十六进制转换为段选信号，段选信号控制那一条亮  4'h0:seg[7:0] <= 8'b0000\_0011;//除g\dp外全亮，显示数码0 0是不亮的那一段  4'h1:seg[7:0] <= 8'b1001\_1111;  4'h2:seg[7:0] <= 8'b0010\_0101;  4'h3:seg[7:0] <= 8'b0000\_1101;  4'h4:seg[7:0] <= 8'b1001\_1001;  4'h5:seg[7:0] <= 8'b0100\_1001;  4'h6:seg[7:0] <= 8'b0100\_0001;  4'h7:seg[7:0] <= 8'b0001\_1111;  4'h8:seg[7:0] <= 8'b0000\_0001;  4'h9:seg[7:0] <= 8'b0000\_1001;  4'hA:seg[7:0] <= 8'b0001\_0001;  4'hB:seg[7:0] <= 8'b1100\_0001;  4'hC:seg[7:0] <= 8'b0110\_0011;  4'hD:seg[7:0] <= 8'b1000\_0101;  4'hE:seg[7:0] <= 8'b0110\_0001;  4'hF:seg[7:0] <= 8'b0111\_0001;  default:seg[7:0]<=8'b1111\_1111;  endcase  end  endmodule |

## 管脚约束程序源代码

|  |
| --- |
| set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[7]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[6]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[5]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[4]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[3]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[2]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[1]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {seg[0]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {SW[1]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {SW[0]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {which[2]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {which[1]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports {which[0]}]  set\_property IOSTANDARD LVCMOS18 [get\_ports clk]  set\_property IOSTANDARD LVCMOS18 [get\_ports enable]  set\_property PULLDOWN true [get\_ports {SW[1]}]  set\_property PULLDOWN true [get\_ports {SW[0]}]  set\_property PULLDOWN true [get\_ports clk]  set\_property PACKAGE\_PIN H4 [get\_ports clk]  set\_property PACKAGE\_PIN L21 [get\_ports enable]  set\_property PACKAGE\_PIN N22 [get\_ports {which[0]}]  set\_property PACKAGE\_PIN M21 [get\_ports {which[1]}]  set\_property PACKAGE\_PIN M22 [get\_ports {which[2]}]  set\_property PACKAGE\_PIN T3 [get\_ports {SW[1]}]  set\_property PACKAGE\_PIN U3 [get\_ports {SW[0]}]  set\_property PACKAGE\_PIN H19 [get\_ports {seg[7]}]  set\_property PACKAGE\_PIN G20 [get\_ports {seg[6]}]  set\_property PACKAGE\_PIN J22 [get\_ports {seg[5]}]  set\_property PACKAGE\_PIN K22 [get\_ports {seg[4]}]  set\_property PACKAGE\_PIN K21 [get\_ports {seg[3]}]  set\_property PACKAGE\_PIN H20 [get\_ports {seg[2]}]  set\_property PACKAGE\_PIN H22 [get\_ports {seg[1]}]  set\_property PACKAGE\_PIN J21 [get\_ports {seg[0]}]  set\_property BITSTREAM.GENERAL.COMPRESS TRUE [current\_design] |

# 测试程序设计

|  |
| --- |
| module TEST3;  reg clk;  reg [1:0]SW;  // Outputs  wire [2:0] which;  wire [7:0] seg;  wire enable;    nixie\_tube uut (  clk,  SW,  which,  seg,  enable  );  initial begin  clk = 1;  SW = 0;  #5000;  SW = 1;  #10000;  SW = 2;  #15000;  SW = 3;  end  endmodule |

# 实验结果分析

## 仿真波形

电脑萤幕截图

描述已自动生成

电脑萤幕画面

描述已自动生成

电脑萤幕画面

描述已自动生成

## 仿真结果分析

**代码中的预设值与对应关系**

预设值data如下：

文本

描述已自动生成

段选信号如下：

文本

低可信度描述已自动生成

数据选择开关对应的data如下：

图片包含 图形用户界面

描述已自动生成

**仿真例子分析**

以SW[1:0] = 2’b01和SW[1:0] =2’b11的情况为例

在使能信号enable为1的条件下，数据选择开关为：SW[1:0] = 2’b01，仿真结果如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SW[1:0] = 2’b01 | | | | | enable = 1 | | | | |
| 位选信号  which[7:0] | 0 | 1 | 2 | 3 | | 4 | 5 | 6 | 7 |
| 段选信号 seg[7:0] | 00000001 | 00011111 | 01000001 | 01001001 | | 10011001 | 00001101 | 00100101 | 10011111 |

**结果与代码中的预设值及对应关系完全吻合**

在使能信号enable为1的条件下，数据选择开关为：SW[1:0] = 2’b11，仿真结果如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SW[1:0] = 2’b11 | | | | | enable = 1 | | | | |
| 位选信号  which[7:0] | 0 | 1 | 2 | 3 | | 4 | 5 | 6 | 7 |
| 段选信号 seg[7:0] | 00000001 | 01110001 | 01100001 | 10000101 | | 01100011 | 11000001 | 00010001 | 00001001 |

**结果与代码中的预设值及对应关系完全吻合**

## 实验结论

首先，数码管扫描技术是一种有效的数字显示方法，通过逐个扫描数码管的每一位，可以实现多位数的显示，提高了数字显示的效率。其次，我掌握了数码管的扫描原理和实现方法，对于理解数字电路的工作原理和设计数字系统具有重要意义。总的来说，数码管扫描实验为深入学习数字电路提供了实际操作的机会，帮助我将理论知识与实际应用相结合，提升了综合能力。

## 实验出现的问题与解决方案

问题一：数码管在显示数字时可能出现闪烁或抖动的情况

解决方案：

闪烁或抖动可能是由于扫描速率设置不当引起的。尝试调整扫描速率，确保它与数码管的响应时间相匹配。有时候，减缓扫描速率可以减轻闪烁的问题。

问题二：数码管显示的数字可能出现异常或错位

解决方案：

①确保扫描时序是正确的，每个数码管的扫描信号都按照正确的顺序发送。如果扫描时序出现问题，可能导致数码管显示的数字错位或异常。

②仔细检查逻辑电路的设计，确保逻辑电路正确连接。逻辑错误可能导致数字显示不符合预期。检查数字信号的传递路径，确保它们按照正确的顺序传递到每个数码管。

# 思考与探索

1.实验中，某同学看到数码管显示的数字半明半暗，情分析是什么原因？应该如何解决？

原因：

①数码管的连接和接线有误

②电源波动，电压不稳定

③不正确的数码管的共阳或共阴配置

解决方案：

①检查数码管的连接和接线是否良好，确保每个数码管都正确地连接到逻辑电路

②检查电源电压的稳定性。通过使用稳定的电源，并确保电源电压符合数码管的工作要求。

③调整数码管的共阳或共阴配置，确保逻辑电路与数码管的类型匹配。

# 意见和建议

通过实践板级验证，我深刻理解了数字电路设计的实际应用和复杂性。这一环节不仅加强了我对数字电路原理的理解，还培养了我解决实际问题和调试电路的能力。板级验证过程中，面对各种可能的电路问题，我学会了逐步排查和调整参数，提高了解决问题的技能。这种实际的设计和验证过程使我更加自信地应对复杂的数字电路设计挑战，并为将理论知识转化为实际项目打下了坚实的基础。