

实验 2 七段数码管动态显示电路设计

Advanced Electronics System Lab, 202309

一、实验目的

- 1、学习 8 位七段数码管动态显示的控制原理；
- 2、学习编写 Verilog 程序，实现七段数码管动态显示控制电路；
- 3、学习 Verilog Test Bench（测试平台）的编写和仿真流程；
- 4、熟练掌握 Altera FPGA 的开发环境、设计步骤和流程。

二、实验步骤

按照“实验 1_Altera FPGA 开发入门”文档中的完整设计流程进行操作，具体内容为：

- 建立设计工程（注意工程名、顶层实体名及存放文件夹等工程参数的设置）
- 添加并编写 Verilog 源代码
- 对 Verilog 源代码进行综合与分析
- 编写测试平台程序，进行仿真，观察仿真结果
- 进行全编译，生成配置文件，进行硬件验证
- 查看 RTL 级的电路结构

三、七段数码管动态显示电路设计

1、七段数码管的工作原理

七段数码管是用七个不同形状的发光二极管（LED）摆放在不同的位置构成的，如图 1 所示。通常为了方便使用，七段数码管还会多出一段，用来表示“小数点”，因此七段数码管实际上是八段，但习惯上仍称为“七段数码管”。

一个七段数码管的内部原理如图 2 所示，分为共阴极和共阳极两种封装形式。若七段数码管中所有 LED 的正极连接在一起进制控制，所有的 LED 负极单独控制，则称为“共阳极”数码管，反之则称为“共阴极”数码管。本实验中 FPGA 实验板上用到的是“共阳极”七段数码管，如图 2 所示。

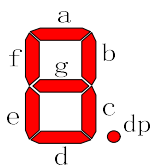


图 1 七段数码管外形图

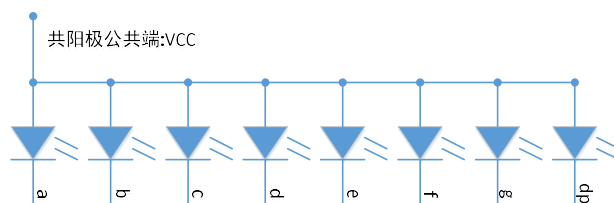


图 2 共阳极七段数码管原理图

2、8 位七段数码管的动态连接方式

如上所述,每个七段数码管需要 9 根信号线控制,才能完成字符的正确显示。这 9 根线分别是共阳极端和阴极端 a、b、c、d、e、f、g、dp。当配备多个数码管时(如 8 个),如果每个数码管都要使用 9 个控制信号,则对硬件资源占用量过大,是不可取的方式。

通常是将 8 个七段数码管的相同段的控制端连接在一起,通过一根控制信号线进行控制。例如 8 个七段数码管的 8 个 a 端连接在一起,用一个信号 A 控制,8 个 b 端连接在一起,用一个信号 B 控制,依次类推,这样对多个数码管而言,控制数码管负极端的信号只要 8 根线,如图 3 所示。而 8 个七段数码管中的每一个阳极则分别由一根控制信号线单独控制,如图中的 LED_S0~LED_S7,这样连接起来的 8 位七段数码管就是动态的连接方式。LED_S0~LED_S7 叫做位控制信号,A、B、C、D、E、F、G、DP 叫做段控制信号。

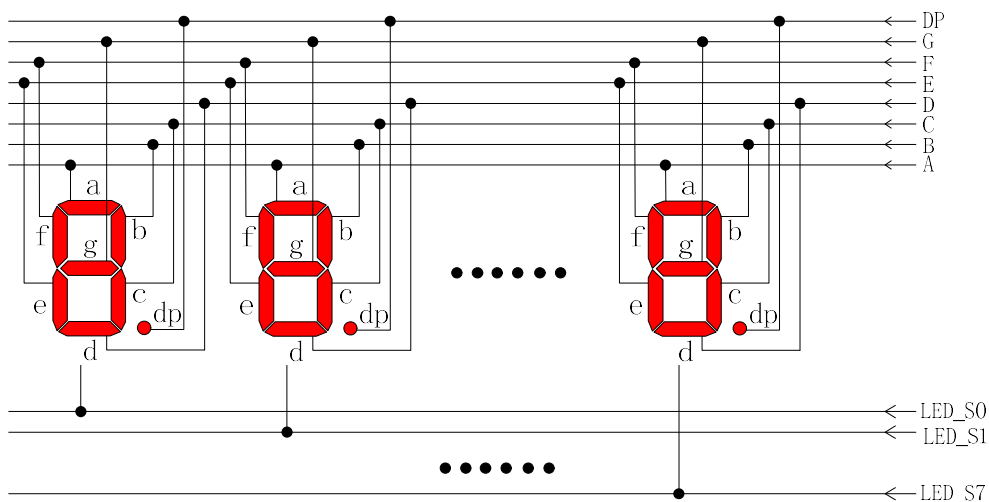


图 3 多位共阳极七段数码管的动态连接方式

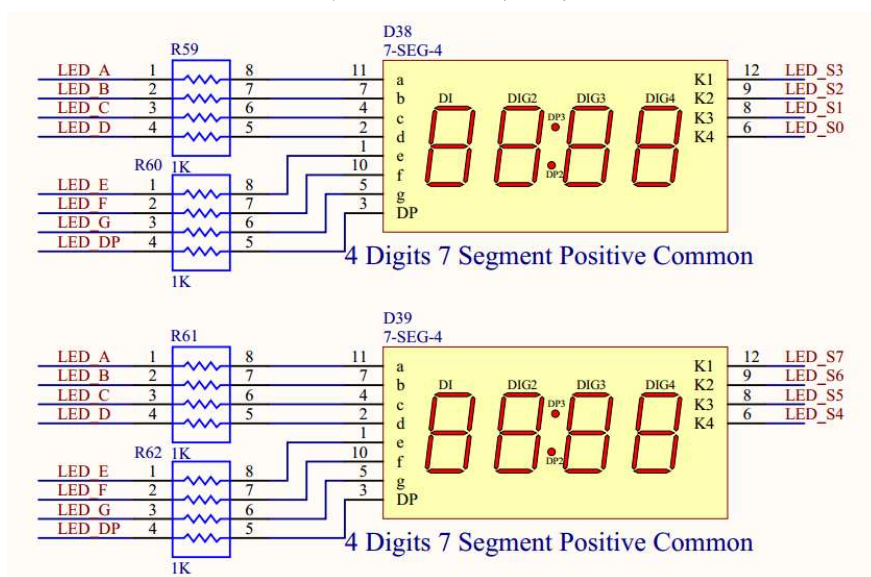


图 4 七段数码管与 FPGA 的硬件连接

图 4 是实际的 8 个七段数码管与 FPGA 的硬件连接电路，从图中可见，4 个数码管是一个整体组件，对外有一组段控制信号线和 4 根位控制信号线。两个组件的段控制线连接在一起，但位控制信号是独立的。

3、实验内容

1、采用 Verilog 编程方式设计一个七段数码管动态扫描控制电路

实现在 8 位七段数码管上同时显示不同的数字（如 0~9，A~F），如图 5 所示，并可以通过拨码开关改变数码管上显示的数字。

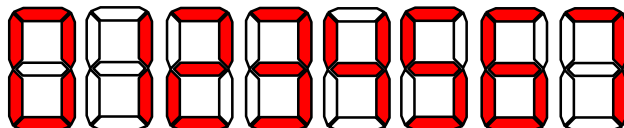


图 5 8 个数码管显示不同的数字

七段数码管动态扫描控制电路的设计思路可以参考讲义 6.3.2 小节，电路基本结构如图 6 所示。

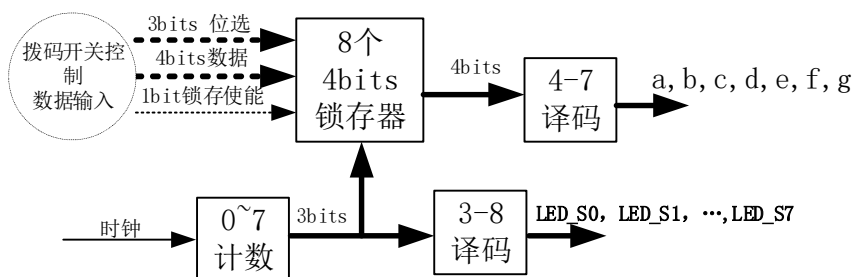


图 6 数码管动态扫描电路结构

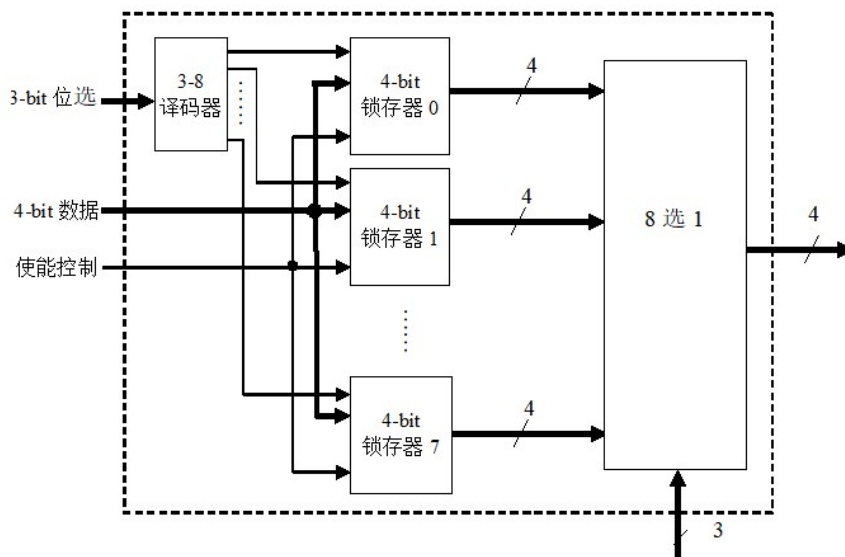


图 7 8 位锁存器的结构框图

在 8 位七段数码管上同时显示不同的数字，需要把 8 个 4bits 数字依次输入 FPGA，并用锁存器保存起来。采用拨码开关从 FPGA 外部依次输入 8 个数字，

并设计 8 个锁存器保存 8 个数字。采用拨码开关选定数字要存入的锁存器，并采用使能信号控制数字的锁存。锁存器里锁存的数字经过经过 4-7 译码，转换成驱动七段数码管的段码后，送给 a、b、c、d、e、f、g、dp。

8 位七段数码管共有 8 个共阳极端（LED_S0~LED_S7），其控制来自于 0~7 计数值经 3-8 译码后的输出（注意：这里的 3-8 译码输出为高电平有效）。

图 6 中的 8 个锁存器可以有不同的实现方式，图 7 给出了一种参考结构。

2、编写 Test Bench 程序并完成功能仿真

针对以上设计的七段数码管动态显示电路，编写 Verilog Test Bench 进行仿真。因为本设计中的模块比较多，因此只要求对图 7 所示部分进行仿真，其他模块的仿真可自选。

需要注意的是，在编写包含时钟端口的 Test Bench 程序时，时钟信号要单独产生，其它的输入信号另外产生。比如：

```
initial clk = 0;
always #1 clk = ~clk; //产生时钟信号
.....
initial begin //产生其他信号
    #0 a = 0;
    #2 a = 1;
    #2 a = 0;
    #1 $stop;
end
```

若时间单位为 10ns，以上语句产生了周期为 20ns 的时钟信号，先 10ns 低电平，后 10ns 高电平，然后不断地反复。

3、查看 RTL 电路结构图

在仿真结束后，通过“Tools”菜单“Netlist Viewers”子菜单下的“RTL Viewer”命令查看设计电路的 RTL 级的电路结构。

4、为设计工程分配管脚

通过查阅“实验指导书 altera_cyclone5.pdf”文档中的“附件一 FPGA 管脚约束”，分别为设计中的输入和输出选定管脚并添加到 Quartus II 工程中。

5、完成硬件验证

分配管脚之后，对设计文件进行全编译，生成配置文件，并将配置文件配置到 FPGA 中。观察数码管，查看显示是否正确，并通过拨码开关改变数码管的显示数字。

四、结束

至此，我们完成了 Altera FPGA 设计的基本流程和步骤。如果你结束了今天的实验，请备份实验数据和实验记录，关闭所有打开的软件并关闭计算机；关闭实验箱的电源，并拆除所有相关的连线，恢复到实验之前的整齐样子，别留下任何物品。O(∩_∩)O 谢谢！

