北京航空航天大學
BEIHANG UNIVERSITY

实验一、初识实验板——数码管实验

2018 年 11 月 5 日

序号	组长标记	教学班	学号	姓名	签名
1	*	162321	16231275	刘瀚骋	
提交日期	2018-11-12		签收：	评价日期	
评阅 1	评阅 2		评阅 3	评阅 4	平均（百分制）

高等理工学院

目录

实验一、初识实验板——数码管实验（实验指导书部分）

1 实验指导

通过一个实际的数字逻辑电路的实现案例，熟悉 EELAB-FPGACORE2 实验硬件实验平台（以下简称“实验板”）的 FPGA 和串行 PROM 等核心硬件，以及按键、拨码开关、七段数码管等外设；练习通过 Verilog HDL 语言实现 FPGA 可综合代码，以及运用测试程序进行功能仿真的方法；熟悉综合和 FPGA 加载的过程，并实践相应的软硬件调试。

1.1 实验板的硬件资源

实验板核心的 FPGA 芯片的规模达到 50 万门，具有约 1 万个可配置逻辑单元（CLB），360kb（“b”表示比特，“B”表示字节，下同）的块 RAM，以及约 70kb 分布式 RAM。实验板上集成了 JTAG 编程芯片，通过 JTAG 以菊花链的形式连接 PROM 和 FPGA 芯片，可以通过 USB 串口直接编程。

对于实验板编程有两种方式，(1)直接下载*.bit 文件对 FPGA 编程，这种情况下如果掉电则配置丢失；(2)将*.bit 文件转换为*.mcs 镜像文件，对 PROM 编程，形成非易失实现。

在本次实验的基本部分，用到的外设包含 2 路按键、2 路拨码开关、2 位七段数码管。在本次实验的扩展部分，建议试用实验板的 UART（通过 Micro USB）功能，形成具有交互式输入输出的工作状态。

1.3 基本实验要求

1）熟悉实验板资源，熟悉 ISE 操作环境；

2）完善数码管显示定义；

3）完善按键与数码管的配合操作；

4）采用门级电路方式实现数目管编码控制。

1.4 已有软件资源

1.4.1 时钟分频

实验板的主频为 50MHz 时钟，过快的时钟会过多耗能，因此设置了时钟分频的过程块，可供改写使用。

1.4.2 数码管

两个数码管采用动态刷新，需要用寄存器将它们显示的内容寄存起来，通过选通端 COM[1:0]，交替刷新显示。

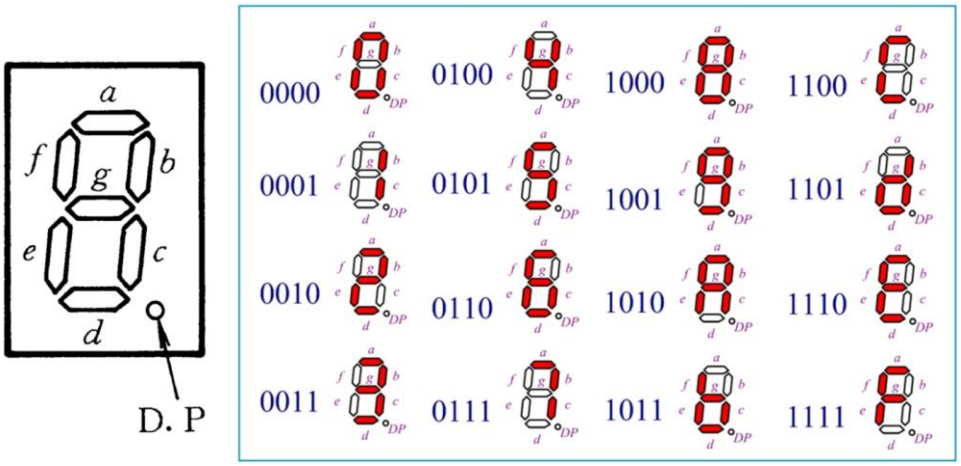

图 1 数码管分段定义

1.4.3 按键去抖

按键需要去抖，提供按键去抖模块 Key 可供利用。

1.4.4 已有程序框架

为了便于入门，提供“预置”模式下按键“加”、“减”计数及其电亮 4 个 LED 灯的程序。

数码管显示也提供模块 Nixietube，但部分字模不全（即：A、b、C、d、E、F），请补充。

1.4.5 引脚和 I/O 命名

附录 A 提供实验板 FPGA 引脚速查，请结合《EELAB-FPGACORE2 使用简介》查对；附录 B 提供一个可供参考的 UCF 定义，Verilog HDL 编程时可参考其 I/O 变量命名。

1.4.6 实验板的 UART

提供一个只能每次读写一个字节的 UART 的样例源代码，具有固定的 9600 波特率、8 位数据位、偶校验、1 位停止位。更好的 UART（注：更稳定可靠、波特率可配置、有 Telnet 等上层协议）需要用户自行开发。

1.4.7 实验板的其它资源

具有 12 位的 ADC 芯片 ADS7951，由于它和 FPGA 采用 SPI 串行接口，写控制指令和读取数据都要根据 SPI 总线的协议，在实验中心给出的例子程序中具有 4 位 ADC 测量的样例。（但由于注释较少，可读性较差）

在实验板上，ADC 的输入范围是 0 到 2.5 伏，请注意使用，避免短路或烧毁。

用户 I/O 资源可以根据引脚定义进行实验，请注意电平的范围，避免短路或击穿。

实验一、初识实验板——数码管实验（实验报告部分）

2 实验报告

2.1 实验背景与需求分析

本次实验需要使用 Verilog HDL 设计电路，实现通过按键控制数码管交替循环显示十六进制数字 0 – F。每次按下按键显示的数值增加 1，且相邻的数字不使用同一个数码管显示。

2.2 系统设计

2.2.1 总体设计思路

系统共分为按键去抖，数据生成，位选择及数码管译码器四部分。

2.2.2 接口设计

系统顶层文件的输入接口为时钟信号，硬件复位信号，按键输入信号；输出为数码管段选信号，数码管位选信号。

2.2.3 按键去抖模块

按键去抖模块用于消除按键输入信号的机械抖动。消除抖动的主要实现思路是通过对系统时钟进行分频，获得一个周期大约在 100ms 量级的按键采样时钟；比较相邻两次按键的电位，如果两次电位相同，则认为按键被按下（或者释放）。

2.2.4 数据生成模块

数据生成模块用于产生用于显示的数据。基本实现思路是时一个长度为 4 的 reg 型输出变量（硬件表示为一组寄存器）在每次按键输入信号的上升沿时自增 1，由于长度限制，该变量的值将会在 0x00 到 0x0F 间循环。

2.2.5 位选择模块

位选择模块用于实现控制数码管的交替显示。基本实现思路是通过输入数据的最低位生成数码管位选信号。当输入最低位为 0 时选中第一个数码管，否则选中第二个数码管，实现交替显示。

2.2.6 数码管译码模块

数码管译码模块用于将输入数据转换成数码管段选信号。该模块按照要求，使用门级描述。

2.3 功能仿真测试

2.3.1 测试程序设计

针对每一个模块分别编写 Testbench，并通过观察其输出波形来判断其是否按照预期工作。

2.3.2 功能仿真过程

首先编写测试文件，给定激励，其次在 Isim 软件中进行仿真。

2.3.2 实验关键结果及其解释

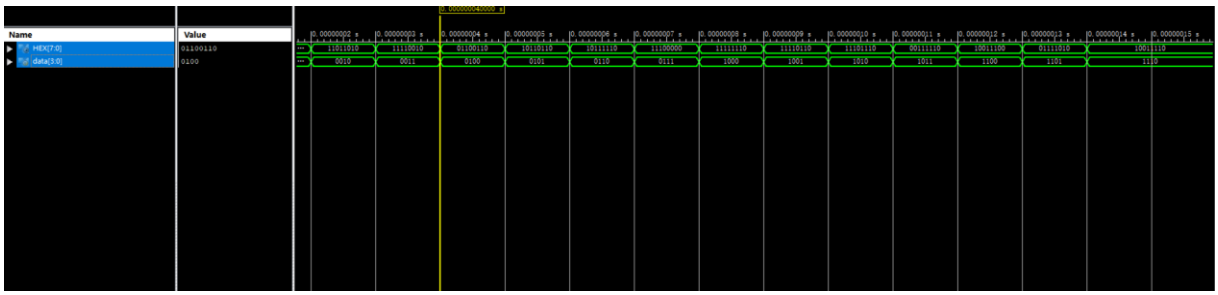

图 1 数码管译码模块测试结果

数码管译码模块测试结果如图 1，其将给定的数据输入转换成数码管段选信号。

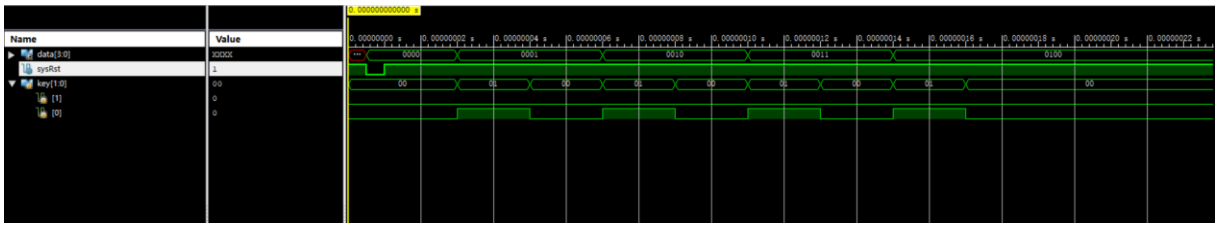

图 2 数据生成模块测试

数据生成模块测试结果如图 2，该模块输出的数据在按键输入的上升沿时自增 1。

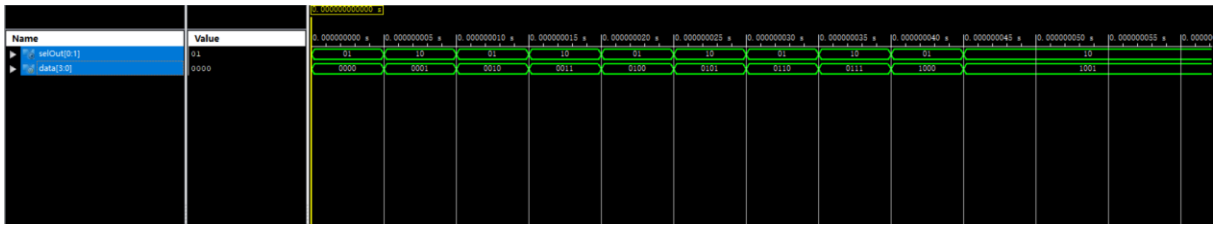

图 3 位选择模块仿真结果

位选择模块测试结果如 3，该模块输出的根据输入数据的奇偶性，输出不同的位选信号，实现交替显示。

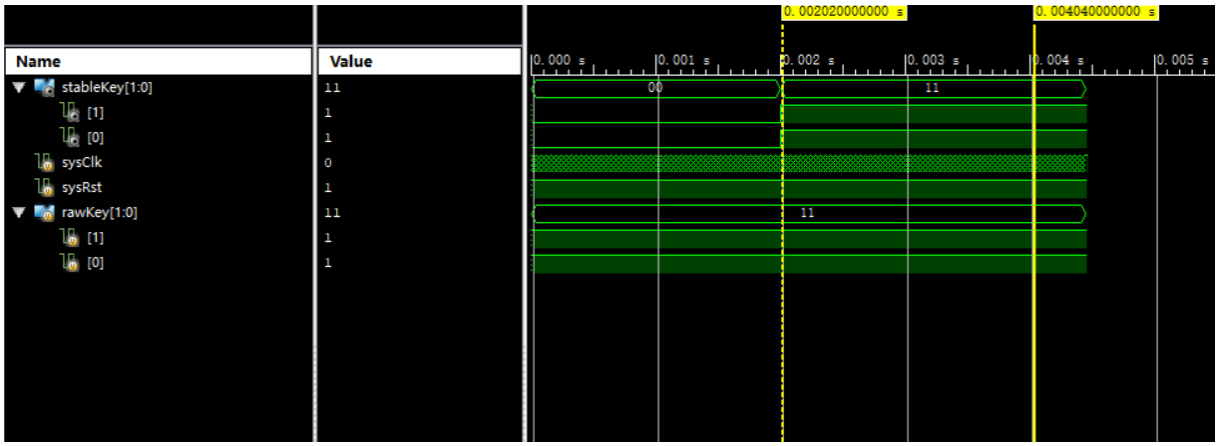

图 4 按键去抖模块仿真结果 1

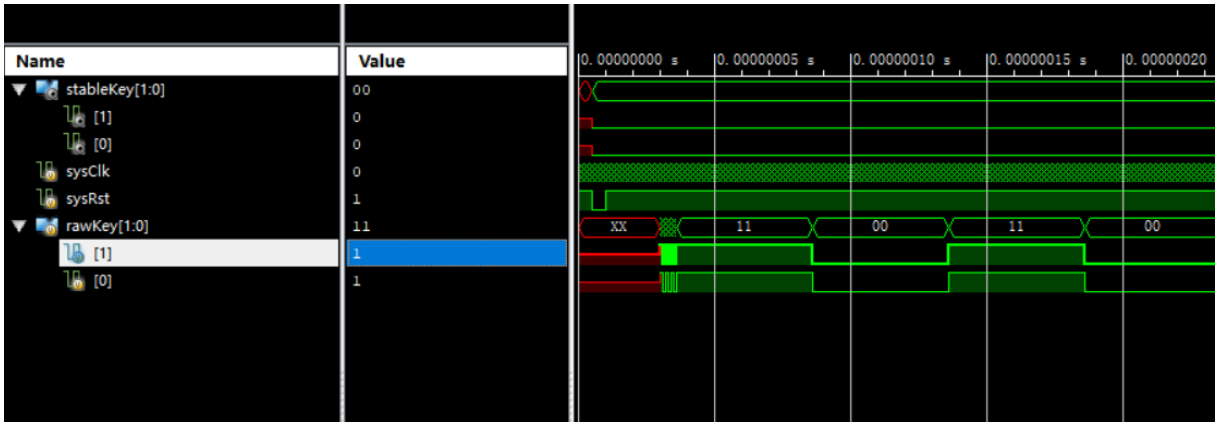

图 5 按键去抖模块仿真结果 2

按键去抖模块测试结果如图 4，图 5。图 4 中对于一个高电平稳定时间超过 10ms 的信号，该模块跟随输出按键信号；图 5 中对于若干个持续时间较短的脉冲输入，按键去抖模块的输出保持先前状态不变。

2.4 设计实现

2.4.1 综合和下载过程

代码编写完成后，点击 ISE 程序主界面左侧的 Synthesis 按钮进行综合与测试。测试完成后，点击 Generating Programming File 按钮生成下载所需的配置文件。

下载时，首先连接实验板，通过 ISE iMPACT 工具配置 Daisy Chain。在测试阶段通过.bit 文件直接编程 FPGA; 测试完成后, 通过 bit 文件生成 MCS 文件, 编程实验板上的非易失性存储器。

2.4.2 实验关键结果及其解释

实验板通电后，因为数据生成模块响应了硬件复位信号，输出清零，此时数码管 1 显示“0”；按下任意一个按键一次后，由于数据位自增一，位选择模块改变了位选信号，此时数码管 1 熄灭，数码管 2 点亮并显示“1”;之后每次按下按键，显示内容会增加 1，两个数码管交替显示；当显示为“F”时再次按下按键，则数据生成模块内部寄存器上溢而清零，回到初始状态。

2.5 小结

本次实验使用 Verilog HDL 设计了使用按键控制数码管交替显示的电路，完成了实验要求。

参考文献

附录 A 实验板 FPGA 引脚速查表

FPGA 引脚序号	原理图引脚定义	引脚说明	备注（参考命名 等）
P36	“Sys_CLK”	系统时钟	主频：50MHz
P26	COM1	“位”选信号——十位	"COM[0]"
P27	COM2	“位”选信号——个位	"COM[1]"
P23	LED_A	“段”选信号——A，共阴极	"SEG[7]"
P18	LED_B	“段”选信号——B，共阴极	"SEG[6]"
P15	LED_C	“段”选信号——C，共阴极	"SEG[5]"
P16	LED_D	“段”选信号——D，共阴极	"SEG[4]"
P17	LED_E	“段”选信号——E，共阴极	"SEG[3]"
P22	LED_F	“段”选信号——F，共阴极	"SEG[2]"
P24	LED_G	“段”选信号——G，共阴极	"SEG[1]"
P12	LED_DP	“段”选信号——小数点，共阴极	"SEG[0]"
P32	SW1	拨码开关第 1 路	"Switch[0]"
P33	SW2	拨码开关第 2 路	"Switch[1]"
P79	LED0	LED 第 1 路	"LED[0]"
P83	LED1	LED 第 2 路	"LED[1]"
P84	LED2	LED 第 3 路	"LED[2]"
P85	LED3	LED 第 4 路	"LED[3]"
P62	CS	片选输入，以 ADS7951 芯片为判断方向	"AD_CS"
P58	SDO	ADC 串行数据输出， 以 ADS7951 芯片为判断方向	"AD_SDO"
P60	SDI	ADC 串行数据输入， 以 ADS7951 芯片为判断方向	"AD_SDI"
P61	SCLK	串行时钟输入， 以 ADS7951 芯片为判断方向	"AD_SCLK"
P2	KEY0	按键 0	"Key[0]"
P3	KEY1	按键 1	"Key[1]"
P11	RESET	复位按键	上拉，"Sys_RST"
P9	TX	USB 串口发送， 以核心 FPGA 为判断方向	NET "Uart_Rx" LOC = P9;
P10	RX	USB 串口接收， 以核心 FPGA 为判断方向	NET "Uart_Tx" LOC = P10;
P63	D0	可扩展 IO_0	电路板正面
P65	D1	可扩展 IO_1	同上
P66	D2	可扩展 IO_2	同上
P67	D3	可扩展 IO_3	同上
P68	D4	可扩展 IO_4	同上
P70	D5	可扩展 IO_5	同上
P71	D6	可扩展 IO_6	同上

P78	D7	可扩展 IO_7	同上
P86	D8	可扩展 IO_8	同上
P90	D9	可扩展 IO_9	同上
P91	D10	可扩展 IO_10	同上
P92	D11	可扩展 IO_11	同上
P94	D12	可扩展 IO_12	同上
P95	D13	可扩展 IO_13	同上
P98	D14	可扩展 IO_14	同上
P99	D15	可扩展 IO_15	同上
P34	D16	可扩展 IO_16	同上

说明：

- ADC 采用 SPI 串行数据总线读入读出数据，最高 SPI 速度可以被配置为 20MHz，ADC 采样率为 1MHz。

附录 B UCF 文件中的定义（Verilog HDL 编程时参考的 I/O 变量命名）

```
NET "Sys_CLK" LOC = P36;    /* EEBUAA 实验板的主频为 50MHz */
NET "SEG[0]" LOC = P12;     /* LED_DP 数码管的小数点 */ /* “数码管”的英文为 nixietube*/
NET "SEG[1]" LOC = P24;     /* LED_G */
NET "SEG[2]" LOC = P22;     /* LED_F */
NET "SEG[3]" LOC = P17;     /* LED_E */
NET "SEG[4]" LOC = P16;     /* LED_D */
NET "SEG[5]" LOC = P15;     /* LED_C */
NET "SEG[6]" LOC = P18;     /* LED_B */
NET "SEG[7]" LOC = P23;     /* LED_A */
NET "COM[0]" LOC = P26;     /* “位”选信号 COM1，“十”位 */
NET "COM[1]" LOC = P27;     /* “位”选信号 COM1，“个”位 */
NET "Switch[0]" LOC = P32;     /* 拨码开关第 1 路 */
NET "Switch[1]" LOC = P33;     /* 拨码开关第 2 路 */
NET "LED[0]" LOC = P79;     /* LED0 */
NET "LED[1]" LOC = P83;     /* LED1 */
NET "LED[2]" LOC = P84;     /* LED2 */
NET "LED[3]" LOC = P85;     /* LED3 */
NET "AD_CS" LOC = P62;      /* CS */
NET "AD_SDO" LOC = P58;     /* SDO */
NET "AD_SDI" LOC = P60;     /* SDI */
NET "AD_SCLK" LOC = P61;    /* SCLK */
NET "Key[0]" LOC = P2;      /* KEY0 */
NET "Key[1]" LOC = P3;      /* KEY1 */
NET "Sys_RST" LOC = P11;    /* RESET */
NET "Sys_RST" PULLUP;
NET "Uart_Tx" LOC = P10;    /* Uart_Tx 是以外部设备为判断方向，RX 以本地设备为判断方向 */
NET "Uart_Rx" LOC = P9;     /* Uart_Rx 是以外部设备为判断方向，TX 以本地设备为判断方向 */
```