# 硬件描述语言与系统仿真

# 实验报告

班级：\_\_\_\_\_\_\_\_\_

组号：\_\_\_\_\_\_\_\_\_

组员：\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_

## 实验八 基于FPGA开发板的直流电压测量装置

### 实验目的

学习模数转换器ADC的相关知识

串行（I2C接口）ADC芯片的PCF8591的驱动设计

学习二进制数转换BCD码的设计方法

LDC显示驱动方法

完成简易电压表设计实现

### 实验内容

基于STEP-MXO2 V2 FPGA核心板和STEP BaseBoard V2.0底板完成简易电压表设计并观察调试结果。通过底板上的串行模数转换器ADC芯片测量可调电位计输出电压，并将电压信息显示在核心板或LCD屏上。

基本要求内容：通过点亮8个LED显示电压的相对强度。

扩展要求内容：通过底板上连接的4个7位数码管以二进制的形式（0-255）或直流电压的方式（0-3.300）显示；通过扩展板上连接的LCD显示屏，在LCD上显示电压值[1]。

### 设计思路

3.1基本原理

系统连接如图8.1所示。0-3.3V的直流电压加在串行ADC的模拟输入端，串行ADC将直流电压转换为8位的数字量，0-3.3V的直流模拟电压得到0-255的数字量；小脚丫FPGA通过内部产生的I2C时序将ADC转换的数据读取到FPGA内部的寄存器，并将串行的二进制数据转换成8位并行的数据。转换后的数据以LED/数码管/LCD显示屏三种方式显示[1]。

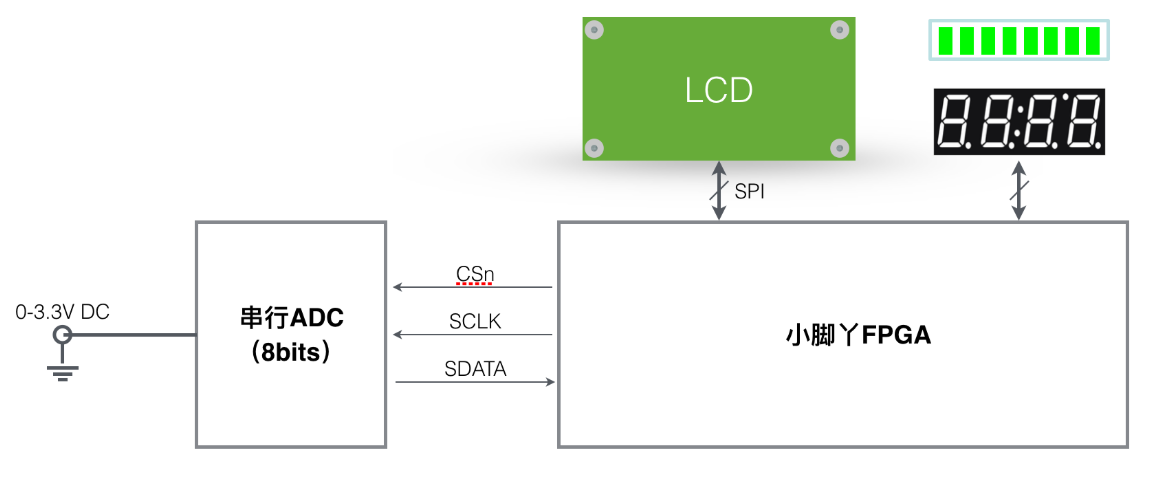


图8.1 测量装置模块示意图

对于不同模块的设计思路分别详述如下。

3.2 ADC采样模块(ADC.vhd)

按照基本原理3.1中所述的内容，ADC芯片PCF8591将0-3.3V直流电压转为8位的数字量，即对应为十进制的0~255，而FPGA的ADC模块就应当通过I2C接口接收并存储这8位的数据。

查阅相关技术手册[2][3]，我们得到了如下信息：

1）PCF8591芯片本质上是对0~3.3V的模拟电压进行256级量化，得到8位的ADC码，量化电平取为分层电平的中间值，如图8.2所示。

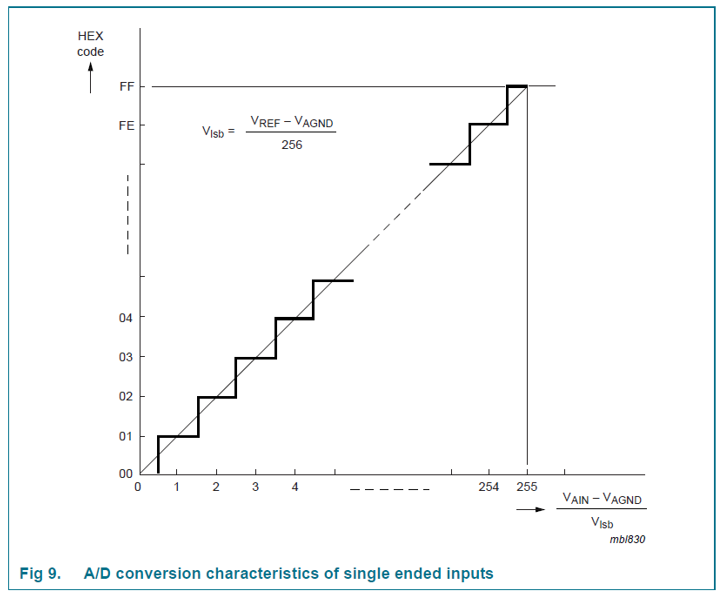


图8.2 PCF8591芯片单端采样时的采样值与模拟电压关系[2][3]

2） FPGA对PCF8591寻址的地址由固定地址和可编程地址组成，如图8.3所示。由于外设底板已将可编程地址 A0、 A1、 A2 接地，加上最低位的读写控制位，故FPGA给PCF8591写数据的寻址地址为90H，从PCF8591读数据的寻址地址为91H。

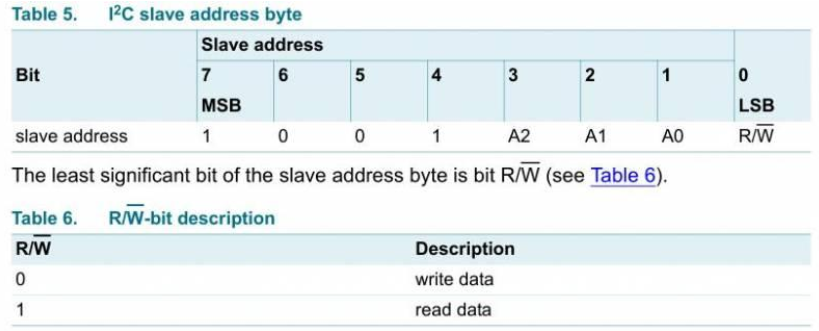
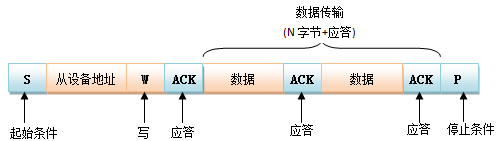
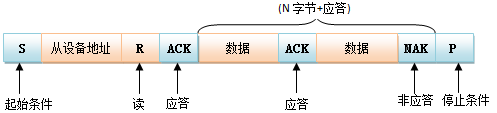


图8.3 PCF8591的I2C从地址[2]

3）为了使I2C总线与PCF8591通信，需要满足I2C总线的通信协议，包括总线时序，时钟频率，通信的地址格式，起止条件，应答等内容。一个典型的I2C总线时序图如图8.4所示。



主设备往从设备中写数据



主设备从从设备中读数据

图8.4 一个典型的I2C总线时序图

根据参考资料[3]，本设计中我们需要两次通信：

第一次为配置数据，具体为：开始–写寻址–读响应–写配置数据–读响应–结束。

第二次为读ADC数据，具体为：开始–读寻址–读响应–[读ADC数据–写响应–]循环读-结束。读出的ADC数据为一个8bits整数，经过处理后使用数码管，LCD显示屏等器件进行显示。

具体地，我们通过SCL（串行时钟信号）和SDA（串行数据信号）两根线完成上述通信。

在芯片手册中关于两次通信的叙述如图8.5和图8.6所示。

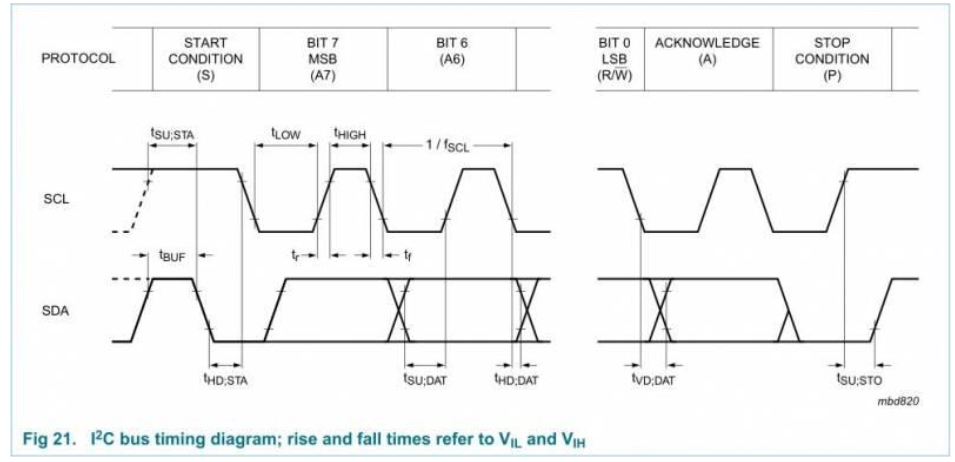


图8.5 I2C通信初始化配置时序[2]

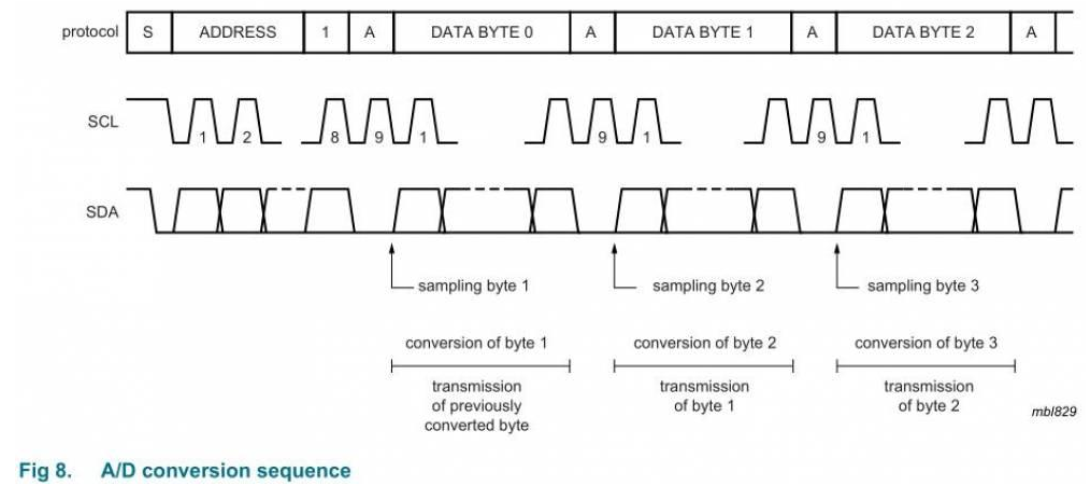


图8.6 I2C通信读取时序[2]

由于原始时钟信号为12MHz，而I2C通信的最高频率为100kHz，分频后每4个周期传输16bit，故至少应分频30倍，即分到400kHz。但在实际过程中发现这样得到的电压显示仍不稳定，即有跳变，我们认为是工作频率仍然较高，故通过多次实验测试最后将分频数增大到了20000，即分到了600Hz，得到的显示数据较为稳定。

根据I2C时序，ADC模块中设置了6个状态来进行通信，即：

IDLE：软件自复位状态，用于程序失控后的复位处理；

MAIN：主状态，包含上述两次通信过程；

START：I2C通信时序中的起始START；

WRITE：I2C通信时序中的写操作WRITE和相应的判断操作ACK；

READ：I2C通信时序中的读操作READ和返回ACK；

STOP：I2C通信时序中的结束STOP。

3.3模式切换模块(mode\_shift.vhd)

模式切换模块通过读取模式切换按键信号，切换系统显示状态。状态mode=0时数码管和LCD屏显示实际电压0.0000~3.3000V；状态mode=1时数码管和LCD屏显示量化编码的十进制值0~255。若有复位键按下，则系统立即复位（异步复位）。本模块中包含有按键消抖模块。

3.4二进制码转BCD码模块(binary\_to\_BCD.vhd)

BCD码中4位只有0~9共十个数，但是在四位二进制中4位有0~15共十六个数，因此需要经过转换算法，将二进制数转换成BCD码后才能在数码管上显示。如图8.7所示，具体来讲可以采用加三移位法，步骤如下[3]：

1、左移要转换的二进制码1位

2、左移之后，BCD码分别置于百位、十位、个位

3、如果移位后所在的BCD码列大于或等于5，则对该值加3

4、继续左移的过程直至全部移位完成。

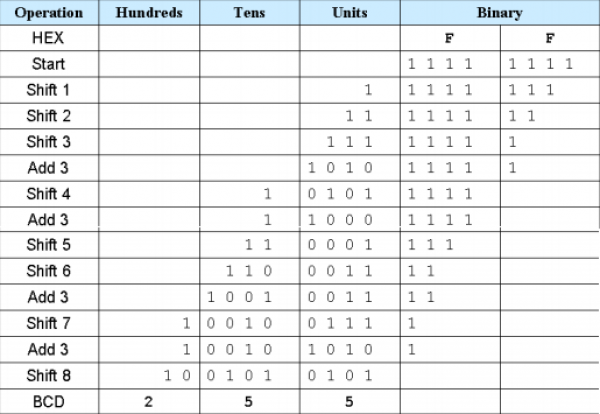


图8.7 二进制数转换BCD码步骤示意图[3]

也可以由BCD码得到量化编码的十进制值。由于量化编码十进制值到实际电平的映射为0~255到0~3.3V，故有

当然，我们实际的显示精度为0.0001V，故实际的由量化编码的十进制值到实际电平的放大倍数为129=128+1，若均用二进制表示，则可将量化编码左移7位，再加上原始量化编码值，就可以得到实际电平的二进制值。

3.5 74HC595控制数码管显示的控制字模块(ctrlword\_595.vhd)

根据参考资料[3]，74HC595的控制字长为16bit，高8位用于位选（使能某个数码管），低8位用于段选（7段数码管具体显示值）。在该模块中，我们将6个数码管的控制字依次逐位串行输入，每读满一个数码管的16bit控制字后并行输出到数码管显示。由上述过程可知，每一轮只能让一个数码管显示数据，但由于显示刷新频率较高，故人肉眼无法分辨，总体上会呈现五个数码管一起显示的效果。

在该模块中，首先将系统时钟12MHz作6分频得到2MHz时钟作为串行读取时钟信号SCK，在SCK下降沿时将当前位数据写入din，在其上升沿时移位并将din数据读入74HC595，传完16bit控制字后，令RCK=1从而并行输出，接着再读下一次的控制字，直到读完所有数码管控制字为止。

3.6 LCD屏显示模块(LCD\_display.vhd)

根据参考资料[3]，在本实验板上集成了Z180SN009液晶屏，液晶屏为1.8寸，128RGB160像素。使用SPI总线进行FPGA主板与ST7735S的通信，包括CSX (chip enable), D/CX (data/command flag), SCL (serial clock) and SDA (serial data input/output)四个接口，分别对应硬件连接图中的D/C，SCL与SDA，图中CS接口接地，因此始终处于允许通信状态。图中RES管脚控制LCD显示屏的复位与否，低电平关闭。BL管脚控制LCD显示屏背光的开启与否，低电平关闭，高电平开启。LCD模块电路连接如图8.8所示。

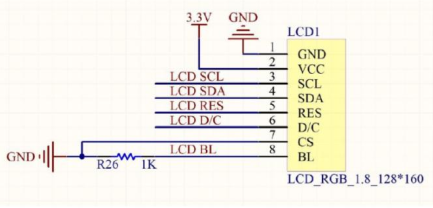


图8.8 LCD模块电路连接[3]

LCD液晶屏上的显示方法采用12\*12数字点阵显示，即在一个12\*12的像素点矩阵用0、1两种像素点灰度值来显示一个数据，例如数字0的显示如图8.9所示。由于本次实验主要只显示数字、小数点和字母V，故需要的led点阵个数其实不多。

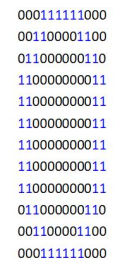


图8.9 数字0的12\*12点阵显示示意

在使用SPI总线进行通信时，需要符合SPI总线与器件的通信标准。经查阅相关资料[4]，LCD显示模块的总线时序与ADC模块类似，如图8.10所示，故我们也设置了6个状态(IDLE,MAIN,INIT,SCAN,WRITE,DELAY)。

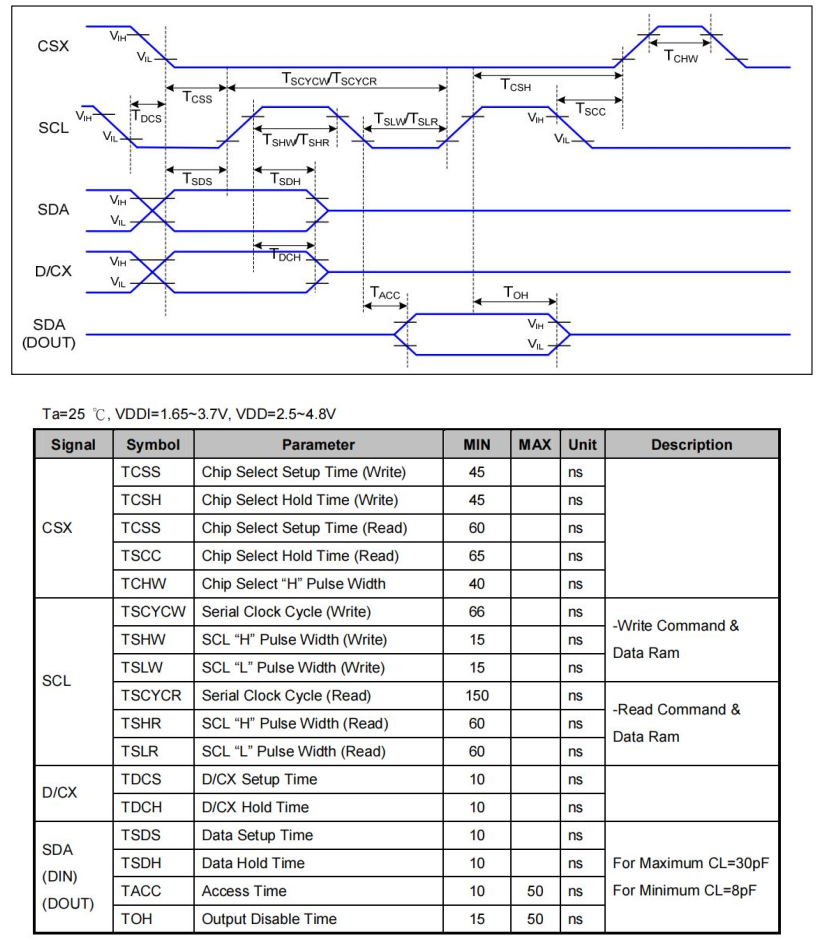


图8.10 SPI四线通信时序[4]

在向ST7735S进行写操作时，典型的时序图如图8.11所示。在CSX由高转低时开启通信，根据SCL写入8bits，在第八个上边沿采样D/CX上数据，0代表写入命令，1代表写入数据。由于本板卡CSX始终为低，故SPI总线要求发送完该字节数据后，下一字节数据需要紧跟着下一个时钟上边沿采样。因此我们在设计时需要合理安排时钟信号的变化，当不发送数据时将SCL停止发送上边沿，防止写入不必要的数据导致设计出错。

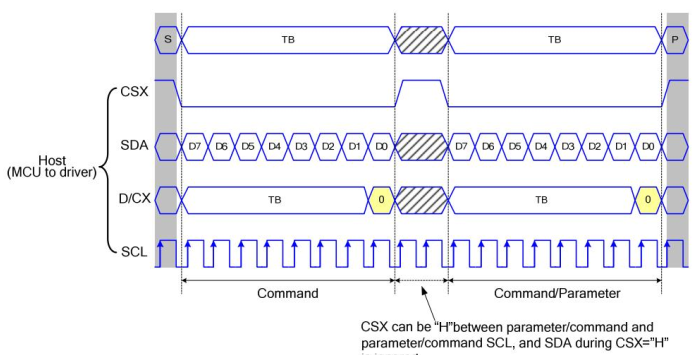


图8.11 SPI四线通信写操作经典时序[3]

3.7 顶层模块(top.vhd)

顶层模块其实相当于一个包装函数，它调用了上述的所有子模块，并且将ADC模块得到的ADC编码送led显示（注意小脚丫的led灯为共阳极管，即低电平显示，故需要先对ADC编码逐位取反再送显示）。

### 仿真及板级测试过程说明

考虑到lab8为针对模拟电压的ADC测量和显示实验，我们组经讨论后认为需要以实际硬件调试为重，故没有编写软件方面的testbench文件。

初始情况下，默认右侧五个数码管和LCD显示真实电压值，而led则显示8位ADC编码。通过旋转FPGA 底板上的A/D、D/A模块的旋钮可调节ADC测量电压，数码管和LCD显示值会相应发生变化，观察发现其变化范围为0.0000~3.2895V。按小脚丫模块最上方按钮(K1)可实现复位，数码管与LCD液晶屏刷新。

按小脚丫模块最下方按钮(K4)可切换系统模式，右侧三个数码管和LCD显示ADC编码值，而led仍显示8位ADC编码。通过旋转FPGA底板上A/D、D/A模块的旋钮可调节ADC编码，数码管与LCD显示值会相应发生变化，观察发现其变化范围为0~255。按小脚丫模块最上方按钮可实现复位，数码管与LCD 恢复真实电压显示模式。

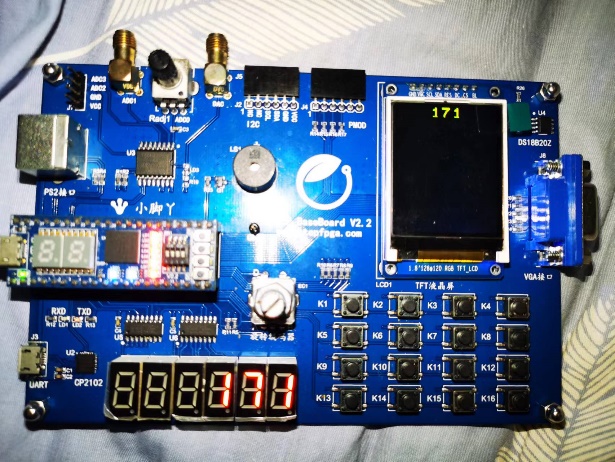
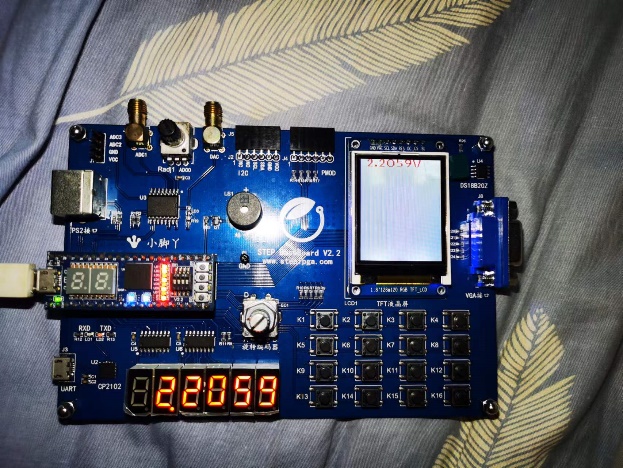


图8.12 板级测试现象图

如图8.12所示，实验现象与预期基本一致，符合实验要求。

### 设计验证过程的问题及解决方案

#### 问题：数码管及LCD屏幕显示跳变过快

一开始将ADC读取频率设为对系统时钟作30分频时，发现数码管以及LCD示数跳变过快。我们分析，这是由于ADC采样对处在两个量化电平中间的电压判决存在误差，故采样值在两个量化电平间快速跳动。我们降低其读取频率，使得跳变能够肉眼分辨，最后发现做20000分频时，即使用600Hz的时钟时控制ADC采样时，显示达到了相对稳定的结果。

### 主要的VHDL源代码列表说明

表8.1 VHDL源代码列表

|  |  |
| --- | --- |
| 模块名 | VHDL源代码 |
| 顶层模块 | top.vhd |
| ADC模块 | ADC.vhd |
| 模式切换模块 | mode\_shift.vhd |
| 二进制码转BCD码模块 | binary\_to\_BCD.vhd |
| 74HC595控制数码管显示的控制字模块 | ctrlword\_595.vhd |
| LCD屏显示模块 | LCD\_display.vhd |

### 总结

本实验作为拓展实验，难度比前几个实验都要高出不少，不仅是在于代码方面，更在于更多的硬件实现细节，例如I2C通信时序、数码管和LCD显示时序和算法、SPI通信时序等等，对于我们这种初学者而言都是不小的挑战。当然，我们通过查阅老师提供的参考资料，最终将所有问题逐一解决，最后完成了本实验，也算有所成就。通过本实验，我们对于VHDL语言、一些通信协议以及算法都有了进一步的了解，大家也都有所收获。

### 参考资料

[1] 实验八-进阶实验直流电压测量装置-实验要求.doc

[2] PCF8591datasheet.pdf

[3] 直流电压测量装置设计指导.doc

[4] ST7735S\_V1.1\_20111121datasheet.pdf

### 附录

9.1 Netlist Analyzer工具综合生成的电路模块图

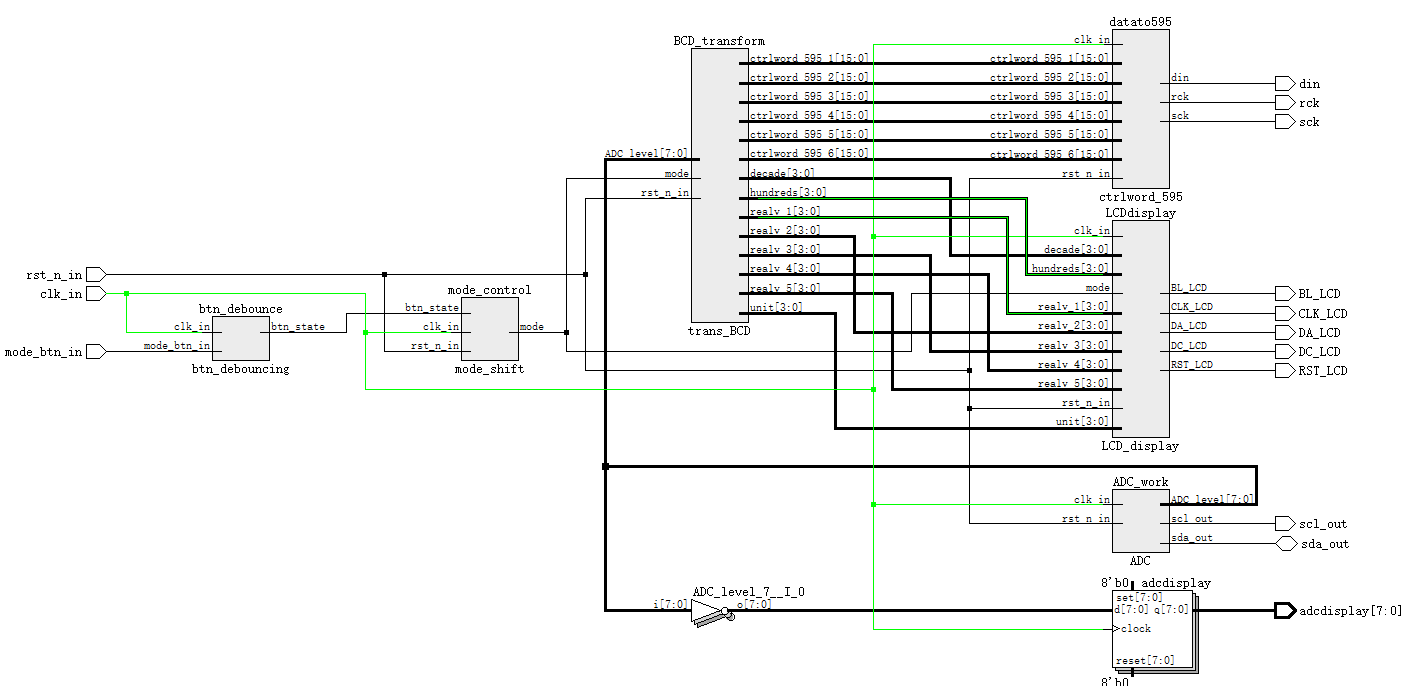


图8.13 综合电路图

9.2 FPGA硬件资源使用情况

Design Summary

Number of registers: 259 out of 4635 (6%)

PFU registers: 259 out of 4320 (6%)

PIO registers: 0 out of 315 (0%)

Number of SLICEs: 783 out of 2160 (36%)

SLICEs as Logic/ROM: 783 out of 2160 (36%)

SLICEs as RAM: 0 out of 1620 (0%)

SLICEs as Carry: 65 out of 2160 (3%)

Number of LUT4s: 1559 out of 4320 (36%)

Number used as logic LUTs: 1429

Number used as distributed RAM: 0

Number used as ripple logic: 130

Number used as shift registers: 0

Number of PIO sites used: 21 + 4(JTAG) out of 105 (24%)

Number of block RAMs: 0 out of 10 (0%)

Number of GSRs: 1 out of 1 (100%)

EFB used : No

JTAG used : No

Readback used : No

Oscillator used : No

Startup used : No

POR : On

Bandgap : On

Number of Power Controller: 0 out of 1 (0%)

Number of Dynamic Bank Controller (BCINRD): 0 out of 6 (0%)

Number of Dynamic Bank Controller (BCLVDSO): 0 out of 1 (0%)

Number of DCCA: 0 out of 8 (0%)

Number of DCMA: 0 out of 2 (0%)

Number of PLLs: 0 out of 2 (0%)

Number of DQSDLLs: 0 out of 2 (0%)

Number of CLKDIVC: 0 out of 4 (0%)

Number of ECLKSYNCA: 0 out of 4 (0%)

Number of ECLKBRIDGECS: 0 out of 2 (0%)

9.3 管脚定义

BLOCK RESETPATHS ;

BLOCK ASYNCPATHS ;

IOBUF PORT "scl\_out" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "sda\_out" IO\_TYPE=LVCMOS33 DIFFDRIVE="NA" ;

LOCATE COMP "scl\_out" SITE "C8" ;

LOCATE COMP "sda\_out" SITE "B8" ;

IOBUF PORT "clk12MHz" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "rst\_n\_in" IO\_TYPE=LVCMOS33 ;

LOCATE COMP "clk12MHz" SITE "C1" ;

LOCATE COMP "rst\_n\_in" SITE "L14" ;

IOBUF PORT "led[0]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[1]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[2]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[3]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[4]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[5]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[6]" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "led[7]" IO\_TYPE=LVCMOS33 ;

LOCATE COMP "led[0]" SITE "N13" ;

LOCATE COMP "led[1]" SITE "M12" ;

LOCATE COMP "led[2]" SITE "P12" ;

LOCATE COMP "led[3]" SITE "M11" ;

LOCATE COMP "led[4]" SITE "P11" ;

LOCATE COMP "led[5]" SITE "N10" ;

LOCATE COMP "led[6]" SITE "N9" ;

LOCATE COMP "led[7]" SITE "P9" ;

LOCATE COMP "sck" SITE "J2" ;

LOCATE COMP "rck" SITE "H3" ;

LOCATE COMP "din" SITE "G3" ;

IOBUF PORT "BL\_LCD" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "CLK\_LCD" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "DA\_LCD" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "DC\_LCD" IO\_TYPE=LVCMOS33 ;

IOBUF PORT "RST\_LCD" IO\_TYPE=LVCMOS33 ;

LOCATE COMP "BL\_LCD" SITE "J13" ;

LOCATE COMP "CLK\_LCD" SITE "K13" ;

LOCATE COMP "DA\_LCD" SITE "K14" ;

LOCATE COMP "DC\_LCD" SITE "J14" ;

LOCATE COMP "RST\_LCD" SITE "K12" ;

IOBUF PORT "mode\_btn\_in" IO\_TYPE=LVCMOS33 ;

LOCATE COMP "mode\_btn\_in" SITE "N14" ;

LOCATE COMP "col[3]" SITE "N6" ;

LOCATE COMP "col[2]" SITE "P6" ;

LOCATE COMP "col[1]" SITE "N5" ;

LOCATE COMP "col[0]" SITE "L3" ;

LOCATE COMP "row[3]" SITE "P7" ;

LOCATE COMP "row[2]" SITE "N7" ;

LOCATE COMP "row[1]" SITE "P8" ;

LOCATE COMP "row[0]" SITE "N8" ;