

文章编号:1007-2780(2012)04-0508-07

基于 SOPC 的 LCD 显示模块的设计与实现

王海霞, 武 一*

(河北工业大学 信息工程学院, 天津 300401, E-mail: hai99xia@163.com)

摘 要: 提出了一种基于 SOPC 技术来控制液晶显示的实现方法, 利用 Quartus II 和 Nios II 等开发环境进行系统设计, 完成硬件开发和软件开发, 实现点阵 LCD 显示模块的功能。并对带字库的 LCD 显示中的中文字库、英文字库原理做了详细的说明, 给出了图形显示的应用程序范例。SOPC 设计可靠性高、移植性好, 稍作改动, 就可以改变点阵 LCD 显示模块的显示, 扩充显示内容。

关 键 词: SOPC 技术; 点阵 LCD; Quartus II; Nios II IDE

中图分类号: TN402 **文献标识码:** A **DOI:** 10.3788/YJYXS20122704.0508

Design and Implementation of LCD Module Based on SOPC

WANG Hai-xia, WU Yi*

(School of Information Engineering, Hebei University of Technology, Tianjin 300401, China, E-mail: hai99xia@163.com)

Abstract: An implementation method of controlling the liquid crystal display based on the SOPC is put forward. The method used the development environment of Quartus II and NiosII to complete the design of system, hardware and software, realize the function of dot matrix LCD display module. In the LCD display with the font library, the theory of the Chinese word stock and English word stock were described. An application program example of graphical display was also provided. The design contains features such as highly reliability and easily portability. It is easy to change the display of dot matrix LCD display module and expand the displayed content by making a little change.

Key words: SOPC technology; dot matrix LCD; Quartus II; Nios II

1 引 言

随着可编程逻辑芯片 PLD(当前主要指 FPGA 与 CPLD)结构与规模的飞速发展, PLD 开发商 Altera 公司提出了一种灵活高效的片上系统解决方案, 即可编程片上系统 SOPC^[1-2], 使设计 LCD 显示模块更加灵活、方便。

SOPC 技术可以根据具体的点阵 LCD 显示模块设计对应的驱动程序, 对不同类型的点阵 LCD 显示模块完成对应的算法, 提供相应的时序

信号和数据信号。还可以根据不同的微处理器设计出对应的接口部, 适用于不同的操作, 通过一些电路的设计在很大程度上减轻微处理器的负担。应该说在 SOPC 技术面前, 点阵 LCD 显示模块的尺寸和位数将不是问题。

2 LCD 显示模块系统设计方案

系统开发主要分为 3 部分: 硬件开发、软件开发和硬件调试, 在 Quartus II 和 Nios II IDE 开发环境中进行, 完成硬件开发和软件开发后将配

收稿日期: 2012-03-24; 修订日期: 2012-05-09

基金项目: 河北省自然科学基金(No. E2010000072)

作者简介: 王海霞(1987—), 女, 河北沧州人, 硕士研究生, 主要研究方向为集成电路应用与单片机智能化技术。

* 通讯联系人, E-mail: haiqqxia@sina.com

置文件和编译好的程序下载到 FPGA 开发板上进行测试,最终实现点阵 LCD 模块的显示。

本设计采用的 FPGA 芯片是 Cyclone II 系列的 EP2C8Q208C^[3],具有 8256 个逻辑单元、36 个内部 M4K RAM 模块、165 888 bit 的 RAM,同时支持 18 个 Embedded multipliers 和 2 个 PLL,资源配备十分丰富,图 1 为系统开发流程图。

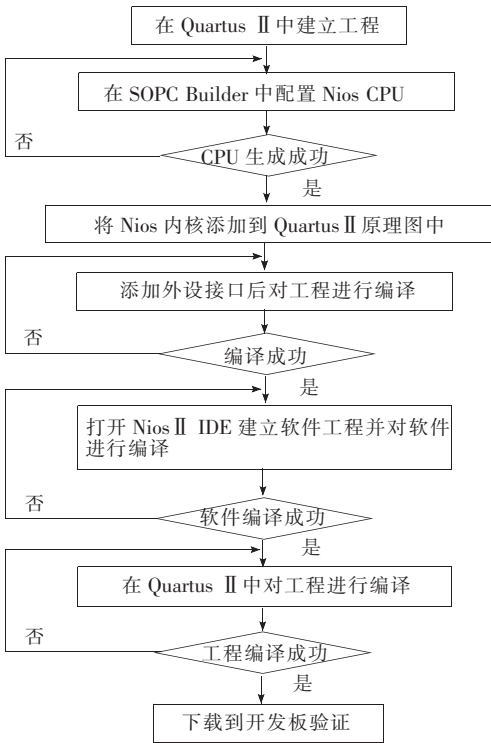


图 1 系统开发流程图
Fig. 1 System development

3 系统硬件开发

3.1 点阵 LCD 硬件电路设计

点阵 LCD 显示模块是 128×64 的 COG 液

晶,它将驱动控制液晶 IC 集成到 LCM 上,既省去了 PCB 底板又节约了空间。驱动芯片为 ST7565P,支持 3 种接口方式,内置 DC/DC 电路,通过软件调节对比度,是一种单芯片点阵 LCD 驱动器,可直接连接到微处理器总线。8 位并行或从微处理器串行显示发送的数据存储在内部显示数据处理器中。每个数据位的内部显示数据 RAM(65×132)和在液晶显示屏上对应的每个像素(65×132)为 1 对 1^[4],因此,ST7565P 既能使显示器具有高自由度,还使显示区域水平可扩展。如图 2 为硬件设计电路图。从图中可以看出,仅 4 根线就可以将点阵 LCD 显示模块与开发板相连,其中 CS 端为片选信号,低电平有效^[5]。

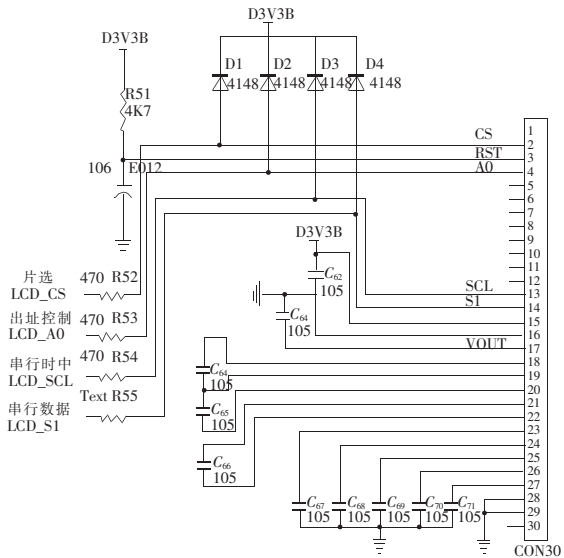


图 2 点阵 LCD 硬件设计电路图
Fig. 2 Hardware circuit of dot LCD

3.2 SOPC 系统设计

构建 SOPC 系统是硬件开发的重点部分,需

Use	Conne...	Module Name	Description	Clock	Base	End	Tags	IRQ
☒		cpu	Nios II Processor	clk_0				
☒		instruction_master	Avalon Memory Mapped Master	clk_0				
☒		data_master	Avalon Memory Mapped Master	clk_0				
☒		jtag_debug_module	Avalon Memory Mapped Slave	clk_0	0x00000800	0x00000fff	IRQ 0	IRQ 31
☒		s1	SDRAM Controller	clk_0	0x01000000	0x017fffff		
☒		epcs	Avalon Memory Mapped Slave	clk_0	0x00001000	0x000017ff		
☒		epcs_control_port	EPCS Serial Flash Controller	clk_0	0x00001840	0x00001847		
☒		sysid	Avalon Memory Mapped Slave	clk_0	0x00001848	0x0000184f		
☒		control_slave	System ID Peripheral	clk_0	0x00001800	0x0000180f		
☒		jtag_uart	JTAG UART	clk_0	0x00001810	0x0000181f		
☒		avalon_jtag_slave	Avalon Memory Mapped Slave	clk_0	0x00001820	0x0000182f		
☒		LCD_SI	PIO (Parallel I/O)	clk_0	0x00001830	0x0000183f		
☒		LCD_A0	Avalon Memory Mapped Slave	clk_0				
☒		s1	PIO (Parallel I/O)	clk_0				
☒		LCD_SCL	Avalon Memory Mapped Slave	clk_0				
☒		s1	PIO (Parallel I/O)	clk_0				
☒		LCD_CS	Avalon Memory Mapped Slave	clk_0				
☒		s1	PIO (Parallel I/O)	clk_0				

图 3 SOPC 系统
Fig. 3 SOPC system

要添加 6 个模块,分别是:CPU(中央处理器)、JTAG UART(串行通信接口)、SDRAM(同步动态随机存储器)、EPCS(串行存储器)、PIO(数据交换模块)、SYSTEM ID(标识符)。如图 3 为构

建好的 SOPC 系统,然后再通过添加输入、输出管脚,构建锁相环 PLL 模块^{[3][6]},分配管脚等步骤,完成 PLL 模块与 SOPC 系统的连接,如图 4 所示。

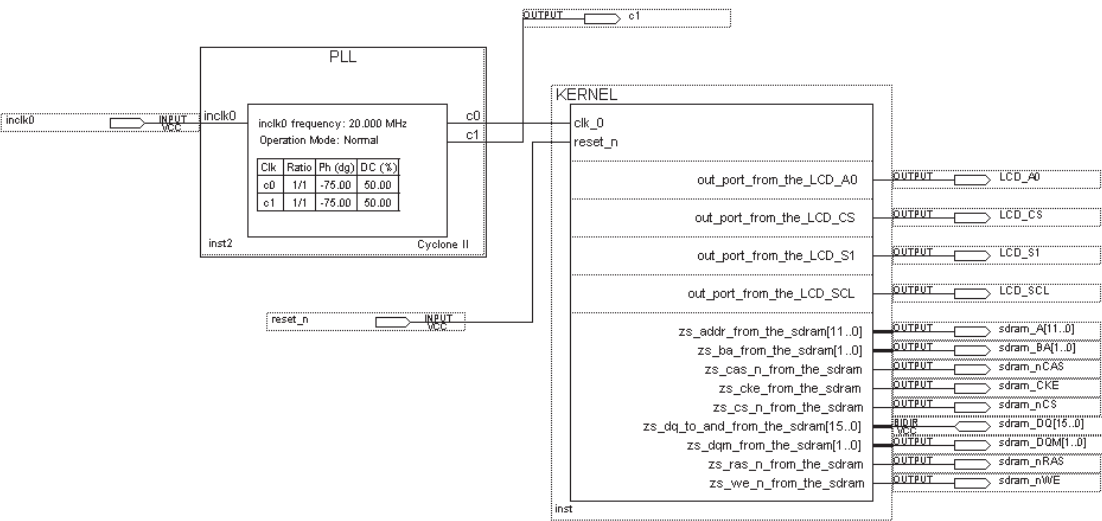


图 4 PLL 模块与 SOPC 系统连接图
Fig. 4 PLL module and SOPC system

4 系统软件开发

软件设计是在硬件系统搭建完毕的基础上,在 Nios II IDE 软件中完成的,主要采用 C 语言进行程序设计,编写的程序包括:sopc. h 宏定义函数,lcd. h 声明函数,lcd. c 驱动函数,main. c 测试函数。

其中在编写 LCD 的驱动程序时涉及到了一个放置图片的 16 进制数组,这个数组是通过字模提取软件生成的。打开字模提取软件,将要显示的图片打开,选择取模方式为 C51,获得该图片的

16 进制代码数组。如图 5 所示。

5 硬件测试及结果分析

在系统的硬件开发和软件开发结束后,将配置文件和编译好的程序下载到开发板进行测试。如图 6 为 LCD 128×64 的图片显示。



图 6 LCD 128×64 图片显示
Fig. 6 Graphical display example of LCD 128×64

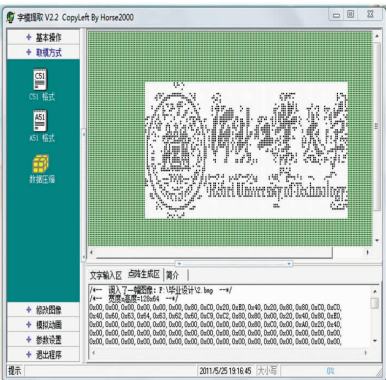


图 5 字模提取
Fig. 5 Extraction of matrix

若想显示其他图片,首先需要改变该图片的像素为 128×64 像素,然后通过字模提取软件获得该图片的 16 进制代码数组,通过改变驱动程序中的 16 进制代码数组,最后经过重新编译,实现图片的显示。

6 带字库的 LCD 显示

由于在点阵 LCD 模块上显示字符,不论是英文还是中文,都需要有字库的支持。在有些 LCD 模块中,芯片中已经包含了字库文件,这样的字库不灵活,显示的效果也不好,而且性价比比较低。本次设计选用的 COG 液晶是本身不带字库文件的液晶,利用外部字库实现字符显示,让液晶显示更漂亮的字体。

6.1 英文字库原理

英文字库使用的是 ASCII 码,起码值为 0~127,其寻址公式为:英文点阵数据在英文点阵字库中的偏移=英文的 ASCII 码×这个英文字模占用的字节数。使用的字库是 tahoma 字体,高度为 16 位。在如下的程序 tahoma_font_offset[] 中,注释中的字符与偏移量是一一对应的,即:字符宽度=它下一个字符的偏移量-它本身的偏移量。比如"#",它的地址偏移量为 19,它的宽度为25-19=6 个^[7]。

```
unsigned int tahoma_font_offset[]
={
// !"#$%&'(
0,3,7,11,19,25,36,43,45,49,
// ) * + , - . / 012
53,59,67,71,75,79,83,89,95,101,
// 3 4 5 6 7 8 9 ; : <
107,113,119,125,131,137,143,147,
152,159,
// = > ? @ A B C D E F
168,175,180,190,197,203,210,217,
223,229,
// G H I J K L M N O P
```

6.2 中文字库原理

每个汉字都是由一个矩形的点阵组成的,每个字节的位代表了这个汉字其中的一个点,0 代表没有点,1 代表有点,最常使用的是 16×16 的点阵宋体字库,即每一个汉字的显示区域为纵、横各 16 点。每一个汉字的前一个字节为该汉字的区号,后一个字节为该字的位号。每一个区记录 94 个汉字,位号则为该字在该区中的位置^[7]。因此,汉字在汉字库中的具体位置采用机内码的计算公式为:

$94 \times (\text{区号} - 0\text{XA0} - 1) + (\text{位号} - 0\text{XA0} - 1)$

这是该汉字以汉字为单位在汉字库中的位置,减 1 是因为数组是以 0 开始而区号的位号是以 1 开始的。如果想得到该汉字以字节为单位所在汉字库中的位置,只需乘以一个汉字字模占用的字节数,即:

$(94 \times (\text{区号} - 0\text{XA1}) + (\text{位号} - 0\text{XA1})) \times \text{一个汉字字模占用的字节数}$

正常来说,一个汉字字模占用的字节数为横纵点数除以 8,以 16×16 点阵为例,它的一个汉字字模占用的字节数就为 16×16/8=32。本设计使用的字库为 HZK12 的 12×12 点阵字库,它的字模每一行和每一列的位数均补齐为 8 的整数倍,该字库实际的长度是 16×16,多余的 4 位都是 0,不显示^[8]。

6.3 带字库与不带字库 LCD 显示的区别

带字库的 LCD 显示相比不带字库的 LCD 显示,在硬件方面需要在已经做好的 SOPC 系统上添加时钟模块、按键中断模块、定时器模块、数码管模块,建立一个新的 SOPC 系统。添加时钟模块时需要构建 3 个 PIO 模块,分别命名为 RTC_DATA、RTC_SCLK、RTC_nRST,对应时钟芯片 DS1302 的 RES(复位),I/O(数据线),SCLK(串行时钟)三根线,如图 7 所示为 DS1302 的电路连接图。添加按键中断模块时需要添加 5 个 PIO 模块,分别命名为 KEY1、KEY2、KEY3、KEY4、KEY5,对应开发板上的 5 个按键“上↑”,“下↓”,“左←”,“右→”,“确定”;添加动态数码管模块时需要添加两个 6 位的 PIO,一个用于数据线,一个作为开关。如图 8 所示为 SOPC 系统新添加

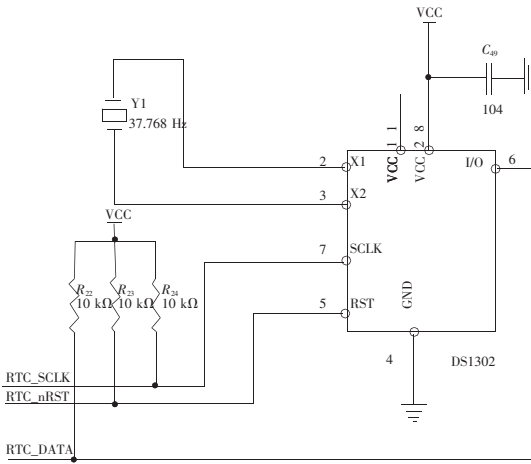


图 7 DS1302 电路连接图

Fig. 7 DS1302 circuit

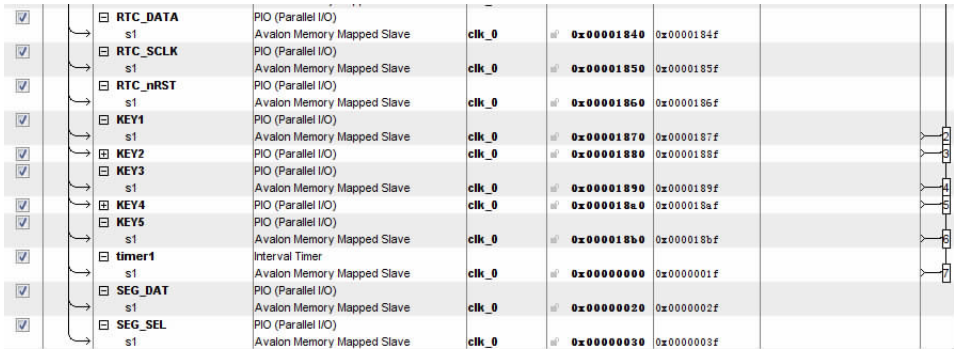


图 8 SOPC 系统新添加部分

Fig. 8 New added parts of SOPC system

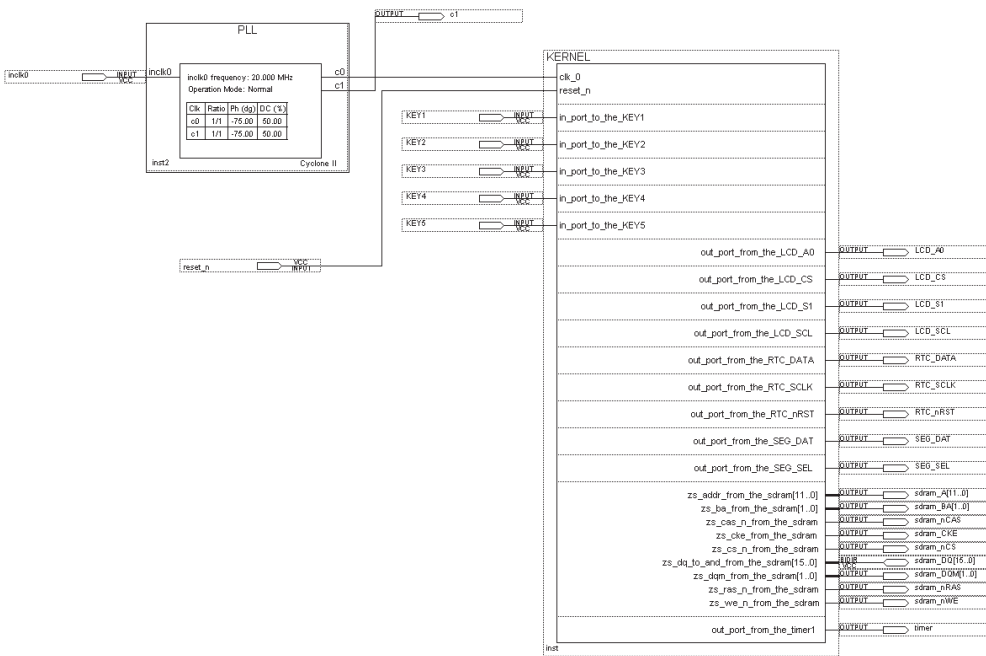


图 9 PLL 模块与 SOPC 系统新添加部分

Fig. 9 New added parts of PLL module and SOPC system

部分,图 9 为 PLL 模块与 SOPC 系统连接新添加部分。

软件开发新添加了 4 个模块和两个字库,编写的宏定义函数:ds1302. h,gui. h,hzk12. h,key. h,lcd. h,sopc. h,tahoma. h;驱动函数:ds1302. c,gui. c,hzk12. c,key. c,lcd. c;主函数:main. c^[9]。

main. c 的部分程序代码为:

定义部分代码:

```
unsigned char time[7] = {0x00,0x19,0x14,0x17,0x03,0x17,0x10};
//格式为: 秒 分 时 日 月 星期 年
alt_u8 segtab[10] = {0xc0,0xf9,0xa4,0xb0,
```

```
0x99,0x92,0x82,0xf8,0x80,0x90}; //0~9
unsigned char led_buffer[8]={0};//led 引脚初始化为 0
unsigned char bittab[6]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf};
static unsigned char cnt=0;
unsigned char ti[][7] = {"一","二","三","四","五","六","日"};
//周一~周日
unsigned char key_dir[][5]={"上↑","下↓","左←","右→","确定"};
//定义 5 个按键
```

输出部分代码:

```
string_t->x = 0;
string_t->y = 2;
string_t->reverse = 0;
printf_string(string_t,"%s",cn_str);// 第二行
输出字符
string_t->x = 0;
string_t->y = 4;
string_t->reverse = 0;
printf_string(string_t,"%s",str);// 第三行输
出符号
while(1){
string_t->x = 0;
string_t->y = 0;
string_t->reverse = 0;
if(tmp_t !=
get_key_value()){
tmp_t = get_key_value();
clear_page(string_t->y);
clear_page(string_t->y + 1);
printf_string(string_t,"您按下的按键为:%s",
key_dir[tmp_t]);
} //第一行输出按键结果
ds1302.get_time(time);.....
//采集时
string_t->x = 0;
string_t->y = 6;
string_t->reverse = 1;
printf_string(string_t,"%02d-%02d-%02d %
02d:%02d:%02d 星期%s\n",time[6],time
[4],time[3],time[2],time[1],time[0],ti
me[5]-1]);
sprintf(buffer,"20%d-%d-%d %d:%d:%d\
n",time[6],time[4],time[3],time[2],time[1],
time[0]);
} //输出日期
```

参 考 文 献:

[1] 朱亦丹,陈远帆. 基于 SOPC 技术的校园信息显示系统 [J]. 计算机与数字工程,2008,36(1):128-132.

[2] 周立功. SOPC 嵌入式系统基础教程 [M]. 北京:北京航空航天大学出版社,2006.

[3] 刘敏,戴曙光,穆平安. 采用 SOPC IP 核技术实现液晶屏显示[J]. 液晶与显示,2011,26(5):665-672.

[4] 杨军. 基于 FPGA 的 SOPC 实践教程 [M]. 北京:科学出版社,2010.

```
sprintf(buf,"%02u%02u%02u",time[2],time
[1],time[0]);
//输出时间
sprintf(buf,"%06u",j++);
for(i=0;i<6;i++){
led_buffer[i] =
buf[5-i]-'0'; //输出星期
}
} //第三行输出日期、时间、星期
```



图 10 带字库硬件测试结果

Fig. 10 Hardware test result with font library

如图 10 所示为带字库的硬件测试结果,实现了数码管、按键中断、中文字符、英文字符、时钟的显示。开发板上的 5 个按键控制点阵 LCD 显示模块第一行的显示,改变按键,第一行的显示也会做出与之对应的改变;中文字符和英文字符的改变需要通过改变 main 主程序中的字符定义;时钟部分实现了年、月、日、时、分、秒和星期的显示。

7 结 论

在 Quartus II 和 Nios II 等开发环境的基础上,基于 SOPC 技术实现了液晶显示屏的图形与字符显示。本设计可移植性好,稍作改动,就可以改变就可以改变点阵 LCD 显示模块的显示内容。SOPC 技术的灵活性使设计修改和产品升级变得非常简单,提高了电子系统的灵活性和通用能力。

[5] 宋泽琳,郑思让,马令坤. 基于 NiosII 处理器的液晶显示接口及驱动程序设计[J]. 液晶与显示,2011, 26(2): 205-209.

[6] Altera Corporation. Nios II hardware development tutorial [EB/OL]. [2011-05-04]. http://www.altera.com.cn/literature/tt/tt_nios2_hardware_tutorial.pdf.

[7] 李兰英. Nios II 嵌入式软核 SOPC 设计原理及应用 [M]. 北京:北京航空航天大学出版社,2006.

[8] 侯长宏,袁慧梅. 基于 SOPC 的 NIOS II 与液晶模块接口及程序设计[J]. 液晶与显示,2008,23(3):307-311.

[9] 高兵,陈莉平,秦俭,等. 液晶和矩阵键 SOPC 外设组件设计开发[J]. 微计算机信息,2008,24(3-2):152-154.



SI 单位的倍数单位

SI 单位的倍数单位是由 SI 词头与 SI 基本单位和 SI 导出单位构成。倍数单位的选取一般应量的数值处于 0.1~1 000 之间。

- SI 词头是用于构成倍数单位的因数符合,不能单独使用。
- 词头符号与所紧接着的单位符号应作为一个整体对待,它们共同组成一个新单位(十进倍数或分数单位),并具有相同的幂次。例如: $10\text{ hm}^2=10\times(100\text{ m})^2=10^5\text{ m}^2$; $1\text{ cm}^3=(10^{-2}\text{ m})^3=10^{-6}\text{ m}^3$ 。
- 词头不能重叠使用,即不能使用由几个词头并列组成的组合词头。例如:不能用 $\text{m}\mu\text{s}$ (毫微秒),正确的为 ns (纳秒)。
- 摄氏度、平面角和时间单位(s 除外)及 kg 等不得使用词头的倍数单位。例如:“mh”(毫小时),“hkg”(百公斤)是不规范的。

数值范围的表示法

1. 数值范围采用波浪号“~”。
2. 书写百分数范围,每个百分数后面的“%”都要重复写出。例如“10 %~20 %”不能写成“10~20 %”。
3. 书写用万或亿表示的数值范围,每个数值中的万或亿不得省略。例如 2 万~5 万,不能写成 2~5 万。
4. 书写具有相同幂次的数值范围,每个数值中的幂次都要重复写出。例如“ $3\times10^5\sim5\times10^5$ ”不能写成“ $3\sim5\times10^5$ ”,但可以写“ $(3\sim5)\times10^5$ ”。
5. 单位相同的量值范围,前一个量值的单位可以省略,只需在后一个量值上写出。例如:“5 mm~8 mm”应写作“5~8 mm”。
6. 单位不完全相同的量值范围,每个量值的单位应全部写出。例如:2 h~2 h30 min(但最好改为“2~2.5 h”)。