

目 录

第一章 SmartSOPC+实验平台介绍	3
一、平台特点:	3
1、QuickSOPC_3C25 核心板硬件资源:	3
2、主机硬件资源:	4
3、实物图介绍:	5
二、电路介绍及使用说明	6
1、核心板电路:	6
1.1、电源电路:	6
1.2、时钟电路	6
1.3、存储电路	7
1.4、配置电路	10
1.5、扩展接口	10
1.6、独立按键及LED 电路	11
2、主板板电路:	12
2.1、电源电路	12
2.2、按键及LED电路	12
2.3、蜂鸣器电路	14
2.4、七段数码管显示电路	14
2.5、液晶显示电路	15
2.6、16*16 LED 点阵电路	15
2.7、RS232 串口电路	16
2.8、RS485 接口电路	16
2.9、红外通讯电路	17
2.10、以太网接口电路	18
2.11、USB接口电路	18
2.12、步进电机电路	19
2.13、直流电机电路	19
2.14、VGA接口电路	20
2.15、PS/2 键盘鼠标接口电路	21
2.16、串行DA、AD 电路	22
2.17、实时时钟电路	23
2.18、数字温度传感器电路	23
2.19、SD/MMC 卡接口电路	24
2.20、外设PACK 接口电路	25
3、高速AD/DA PACK 板:	25
4、红外遥控电路:	26
第二章 Quartus II使用	27
1) 使用Quartus II建立工程	27
(1) 打开Quartus II软件并建立工程	27
(2) 建立图形设计文件	31
(3) 建立文本编辑文件	33
2) Quartus II工程设计	33
(1) 在VHDL文件中编写源程序	33
(2) 从设计文件创建模块	33
(3) 添加led.bsf模块到Quartus II顶层模块	34

(4) 添加引脚和其它基本单元.....	35
(5) 选择器件型号.....	36
(6) 分配FPGA管脚.....	37
(7) 器件和管脚的其它设置.....	38
3) 设置编译选项并编译硬件系统.....	39
(1) 设置编译选项.....	39
(2) 编译硬件系统.....	39
(3) 查看编译报告.....	39
4) 下载硬件设计到目标FPGA	40
5) 波形仿真	40
附录	45

第一章 SmartSOPC+实验平台介绍

SmartSOPC+是广州周立功单片机公司生产的教学实验开发平台集众多功能于一体，是SOPC、EDA、DSP、ARM7 SOC、ARM 以及51 教学实验、科研开发的最佳选择。开发平台采用核心板加主板的结构，更换核心板即可实现不同的功能。

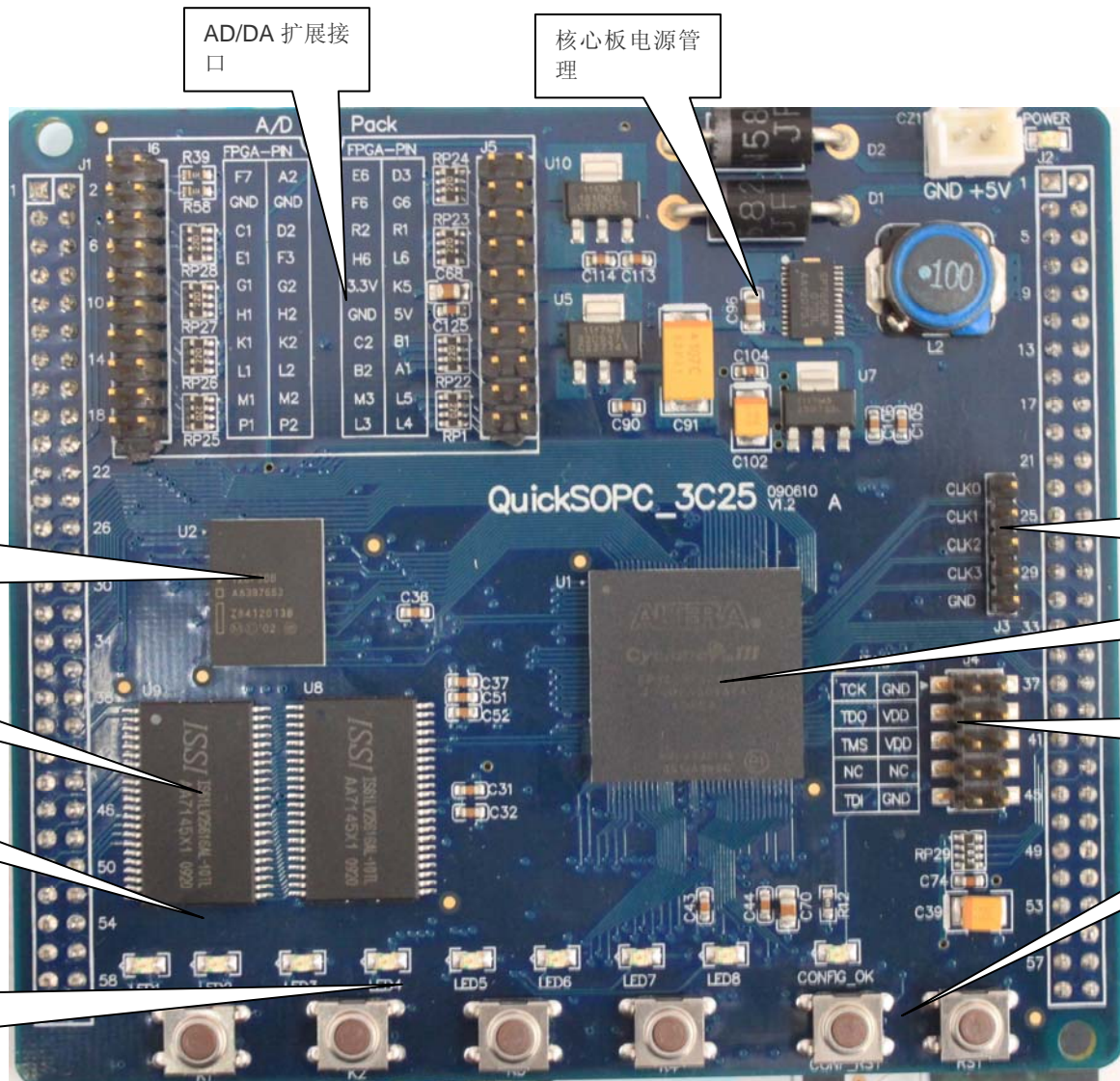
考虑到核心板对用户的实用性，核心板上除设计有FPGA、各类存储器以及用户扩展PACK 外，还设计有按键、LED（与主板共用IO 口）以及电源插座等。用户通过在PACK 上实现自己的功能电路（如主板上的某部分电路），核心板就能脱离主板而单独使用，这样核心板可用于用户自己的设计中，亦可用于电子设计大赛。与一般的实验箱不同，SmartSOPC 主板上大部分外设都不需要进行跳线设置，做实验时方便简单；此外每个外设的关键信号都设置了测试点，方便用户使用逻辑分析仪进行信号测量。总之，一切为设计为用户考虑是本开发平台的出发点。

一、平台特点：

1、QuickSOPC_3C25核心板硬件资源：

- 8层板精心设计，设计有按键、LED以及电源插座、可独立使用，用于二次开发或电子设计大赛
- FPGA : Cyclone EP3C25F324C8，215个最大用户可用管脚
- FLASH: 32 M字节Intel PC28F256P30B85，配合3C25主动并行配置
- SRAM : 两片512k字节（共1M），兼容1M字节，最大可扩充为2M字节
- SDRAM: 一片8 Mbytes，可兼容16/32字节，最大可为32M字节
- 配置器件：（用FLASH进行配置）
- JTAG接口，可下载配置到FPGA、调试程序、Flash编程
- 8个用户LED灯，4个用户按键
- 配置成功指示灯，用于指示配置的成功
- 重新配置按键
- 复位按键
- 一个33个独立IO口的用户PACK，用于扩展高速AD_DA PACK，也可用于用户自己的扩展
- 48MHz的有源晶振

核心板实物如图：

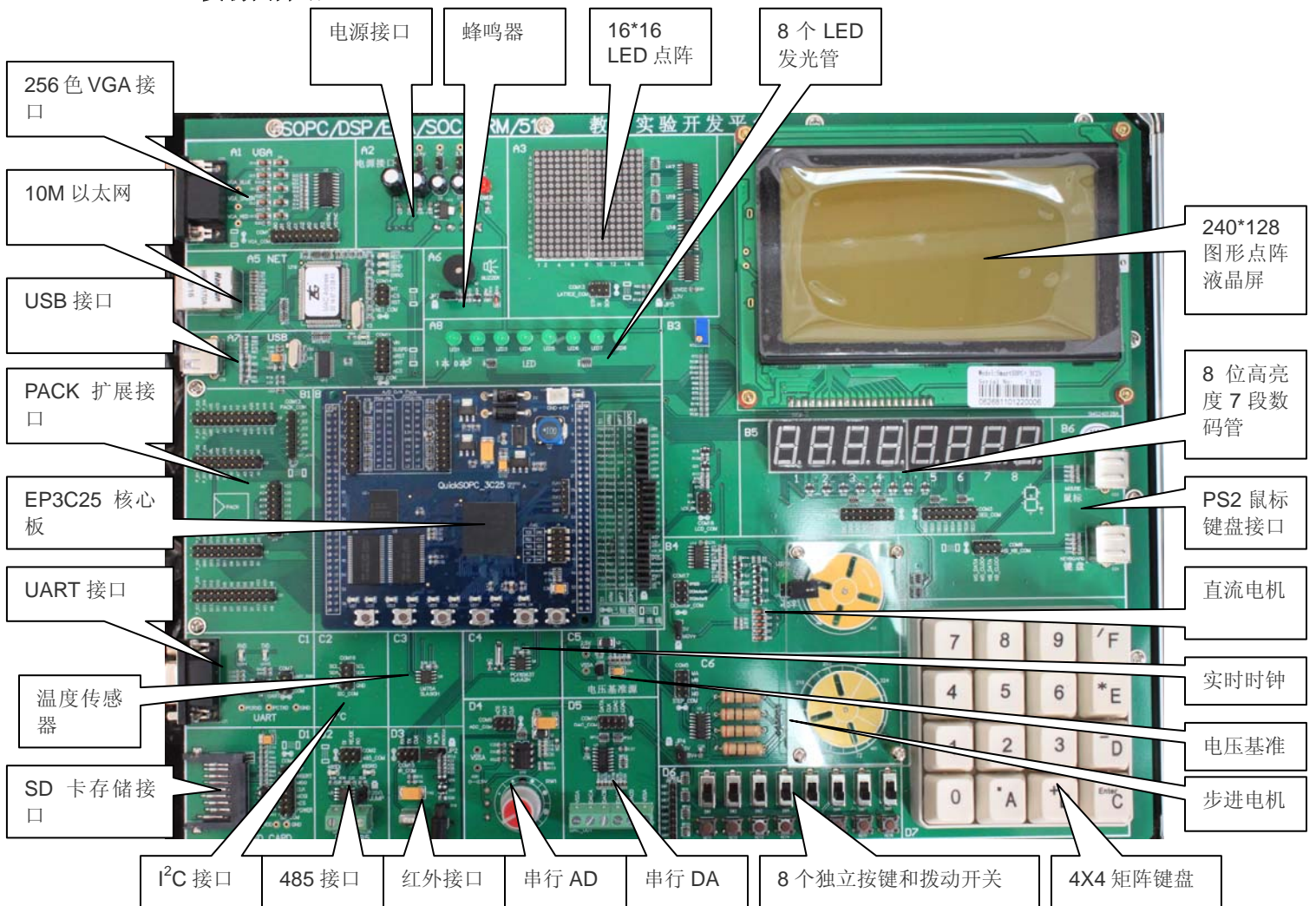


2、主机硬件资源：

输入接口	8 个独立按键	数据 采集	电压基准
	8 个拨码开关		串行 D/A
	PS/2 鼠标、键盘接口 包括完整 IPCore 及其驱动		串行 A/D
	4*4 矩阵键盘		数字温度传感器
输出接口	8 个 LED 发光管	高速数据采 集 (PACK)	实时时钟 PCF8563T
	8 位高亮度 7 段数码管		10bit 125M DA, 可升级到 2 路
	16*16 LED 点阵		1 路 8bit 40M AD
	240*128 图形点阵液晶屏		
	256 色 VGA 接口 包括完整 IPCore 及其驱动		

通讯接口	红外收发		350M 高速运放
	UART	存储设备	SD 卡
	I ² C	扩展接口	主板上总线 PACK 接口
	SPI		核心板上 IO 接口
	RS485	电源	-12V
	10M 以太网 唯一合法 MAC 地址		12V
	USB		5V
	3.3V		
常用外设	交流蜂鸣器	红外遥控器	方便做红外实验
	直流电机 (含测速模块)		
	步进电机		
	交通灯模块		
数字信号源	0.5Hz-48MHz 的多组数字时钟信号	模拟信号源	幅值可调的 10Hz-15kHz 的方波、三角波、正弦波

3、实物图介绍:



二、电路介绍及使用说明

1、核心板电路：

1.1、电源电路：

核心板有两种供电方式：1、当核心板与主板一起使用时，主板上5V 电源通过接口给核心板供电；2、当核心板单独使用时，要用户通过CZ1 外接5V 电源。电源电路如图 1.1所示。

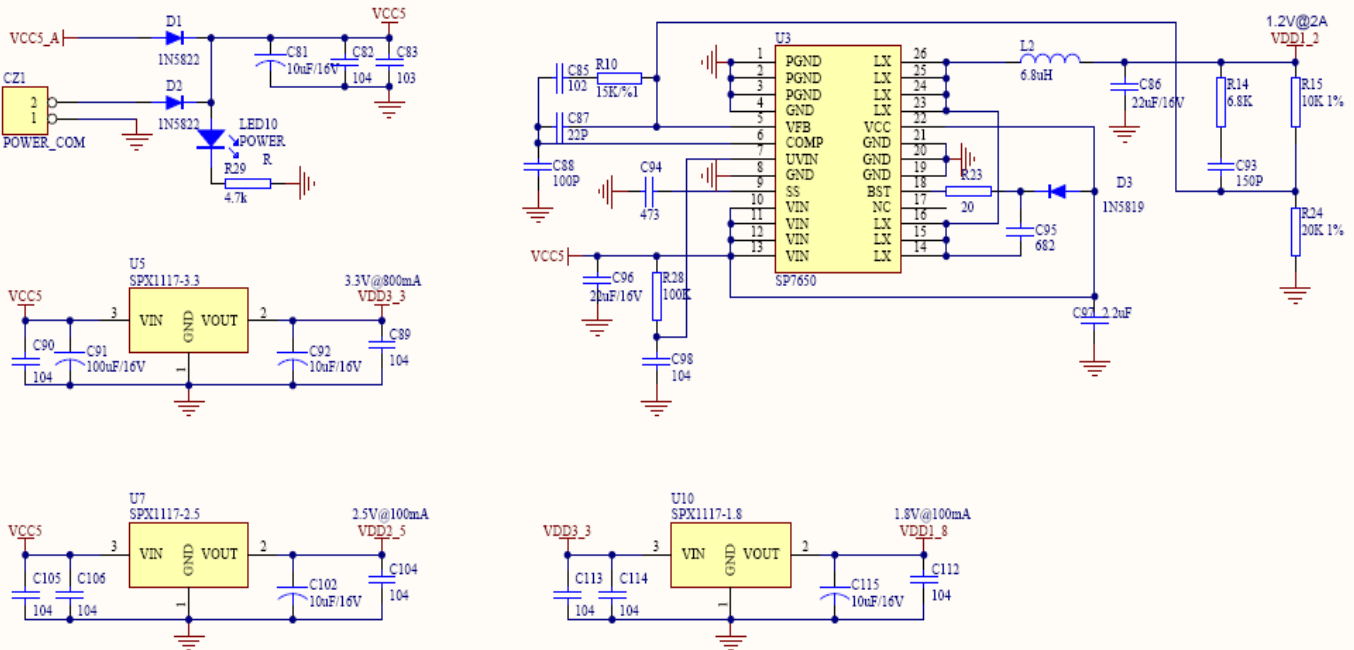


图 1.1 电源电路

3.3V 电源直接由主板提供的5V 电源经过3.3V LDO 芯片SPX1117-3.3 并且滤波以后得到。3.3V 用于给FPGA 所有IO 口、核心板上存储电路、串行配置器件、复位电路、LED 等供电。注意，3.3V 供电能力最大为800mA，用户在PACK 上扩展外设时要注意功耗要求。

SPX1117 系列LDO 芯片输出电流可达800mA，输出电压的精度在±1%以内，还具有电流限制和热保护功能。使用时，其输出端需要接一个至少10uF 的钽电容来改善瞬态响应和稳定性。

1.2、时钟电路

FPGA 内部没有振荡电路，使用有源晶振是比较理想的选择。EP1C6Q240C8 的输入的时钟频率范围为15.625MHz-387MHz，经过内部PLL 电路后可输出15.625MHz-275MHz 的系统时钟。当输入时钟频率较低时，可以使用FPGA 的内部PLL 调整FPGA 所需的系统时钟，使系统运行速度更快，关于PLL 添加和设置的内容请参考 [数字锁相环PLL 应用实验](#)。注意PLL1 使用的是CLK0 或CLK1 的时钟输入，而PLL2 使用的是CLK2 或CLK3 的时钟输入。

核心板包含一个48MHz 的有源晶振，作为系统的时钟源，电路如图 1.2 所示。为了得到一个稳定、精确的时钟频率，有源晶振的供电电源经过了LC 滤波。

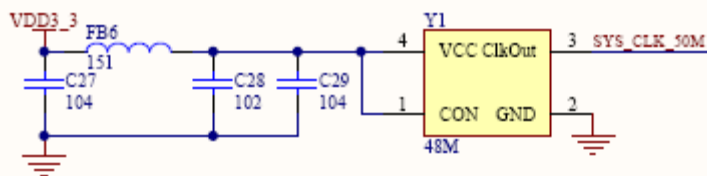
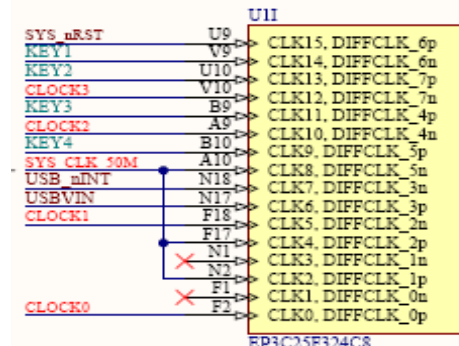


图 1.2 时钟电路



1.3、存储电路

核心板的存储器包括用于存储FPGA 配置数据并进行主动配置的并行配置器件32 M字节 Intel PC28F256P30B85、SDRAM以及SRAM。

1.3.1、Flash存储器

Flash电路如图1.3:

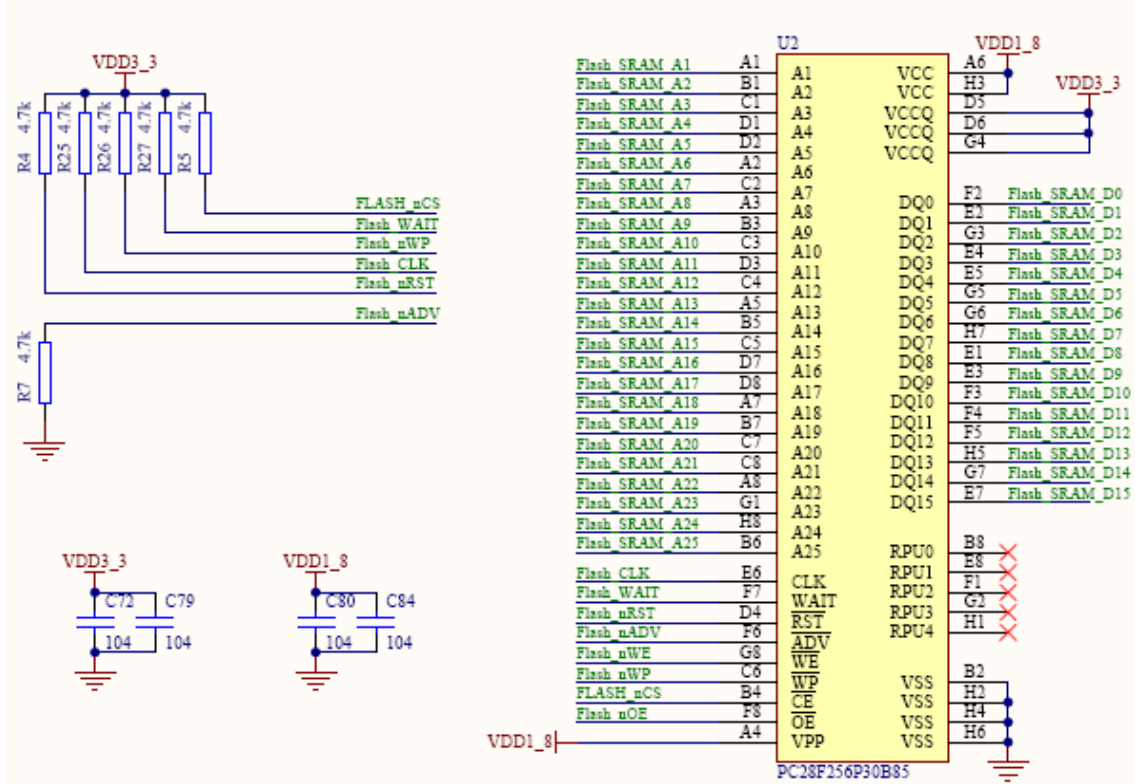


图1.3 Flash电路

QuickSOPC-3C25核心板使用Intel® P30 Flash PC28F128P30B代替EPCS，将硬件配置和软件程序都集中烧写到Flash里，通过主动并行方式 (AP) 在上电的时候对Cyclone® III器件进行配置。详细方法见附录《烧写Intel® P30 Flash》一文。

1.3.2、SRAM 存储器

核心板使用2片512kBytes 的SRAM IS61LV25616AL (256k x16bit)，每片SRAM 都可兼容1M 字节容量的IS61LV51216AL,这样SRAM 容量最大可为2M 字节。SRAM 可作为高速存储器使用，如显示缓存等。SRAM 电路如图 1.4 所示

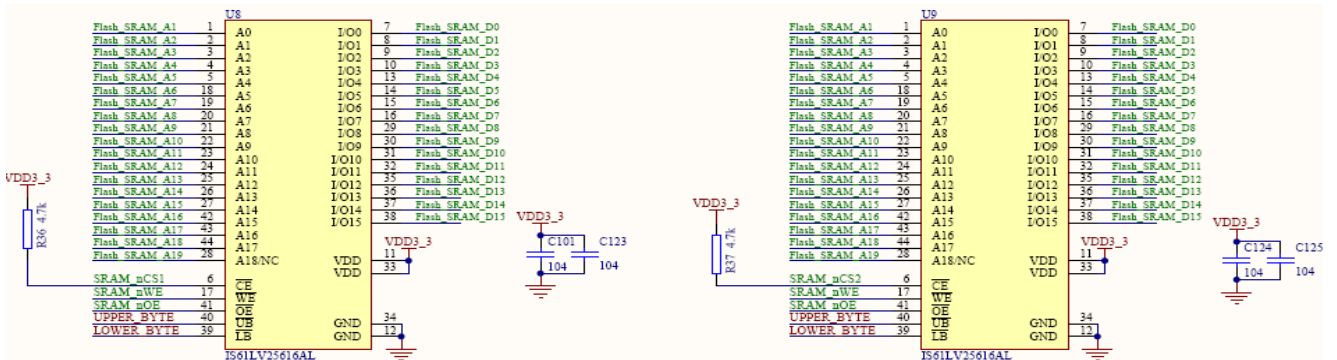


图 1.4 SRAM电路

SRAM 的28 (A18)脚用于1M 字节容量的IS61LV51216AL。电路中2 片SRAM 的片选信号独立，数据总线、地址总线、读写信号线nOE 和nWE 都与Flash 共用，并且也与所有挂在总线上的总线型外设（主板上的液晶以及外扩总线PACK）共用。2 片SRAM 的片选信号分别为SRAM_nCS1 和SRAM_nCS2。

★★2 片IS61LV25616AL的片选信号分别为SRAM_nCS1 和SRAM_nCS2。电路中对芯片的片选信号线进行了上拉（R36、R37）。SRAM 的数据总线与所有挂在总线上的总线型外设（SDRAM、FLASH、主板上的液晶以及外扩总线PACK）都是共用的。如果应用中，两个SRAM（或某一个）不使用时，用户可能不会在FPGA 中定义并设置该器件的片选管脚，片选管脚将为高阻态，不确定的电平有可能选通芯片，从而造成总线冲突。当然，为了避免总线上其它的总线型外设在不使用时因意外而造成总线冲突，应将这些外设的片选都上拉（低电平有效时）或下拉（高电平有效），具体请查看相应外设的电路。

造成总线冲突解释如下：

如图 1.5 所示，数据总线上往往不只一个设备，当FPGA 需要从器件A 中读取数据时（并不希望对器件B 操作），FPGA 发出器件A 的选通信号（nCSA 为低），随后FPGA 发送读信号有效（nOE 为低，其中器件A 与器件B 共用读写信号线），FPGA 为输入，而器件A 为输出。在不对器件B 操作时，器件B 不应该选通，但当发生nCSB 管脚在管脚分配时并没有定义，而且被驱动到地（见图 1.6），那么器件B 也将被选通，而且也为输出状态（因为相同的nOE），这样出现了数据总线上器件A 和器件B 同时输出的情况。而这时，如果一个器件输出为高（如器件A），另一个器件输出为低（器件B），就会造成总线的冲突。第一，本来FPGA 读入器件A 数据线D0 上的信号应该为1，当却被器件B 拉低，造成读数据错误；第二，会在两个输出驱动器之间形成一个电源（VCC）到地（GND）的直电路径，造成一个大电流。一般来说，短时间的冲突可能引发器件发热，长时间则会导致器件过热而烧毁。

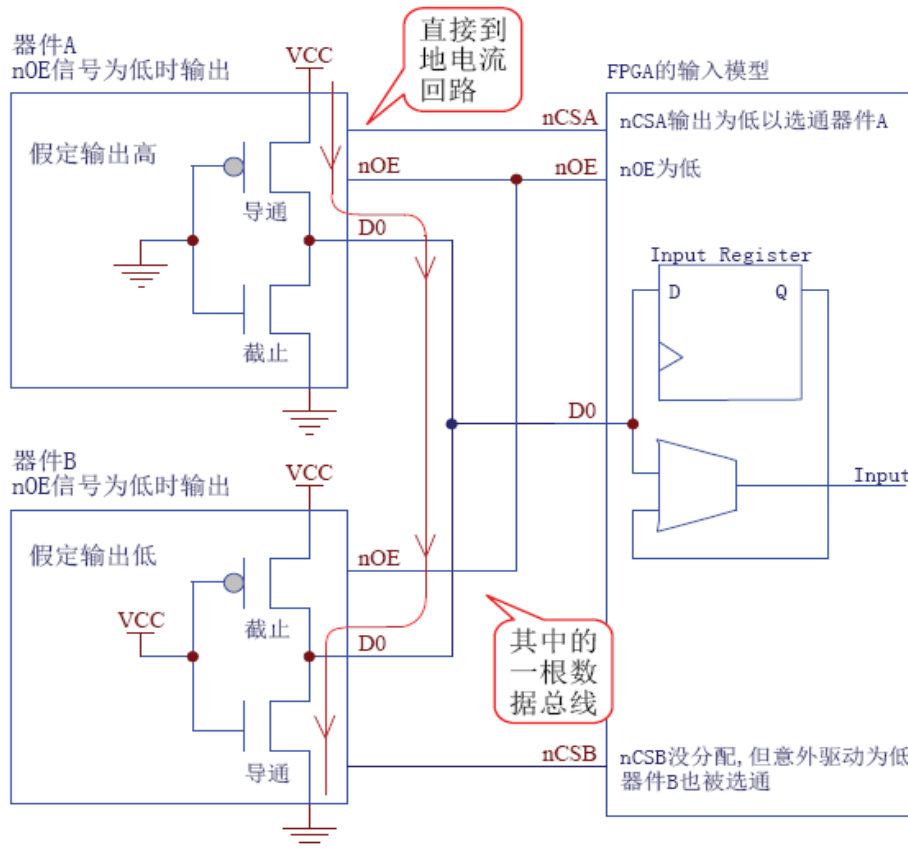


图 1.5 数据总线冲突示意图

在设计中一定要将未定义的管脚定义为三态输入，如图 1.6 所示。这样在上拉（下拉）电阻的作用下为确定的高电平（低电平）。注意一定不能将未定义（不使用）管脚模式设置为输出，并连接到地（As outputs, driving ground）。一定要设置成如图 1.6所示。否则，可能会造成连接在总线上而未使用的芯片的片选有效而长期占用总线，造成总线冲突。

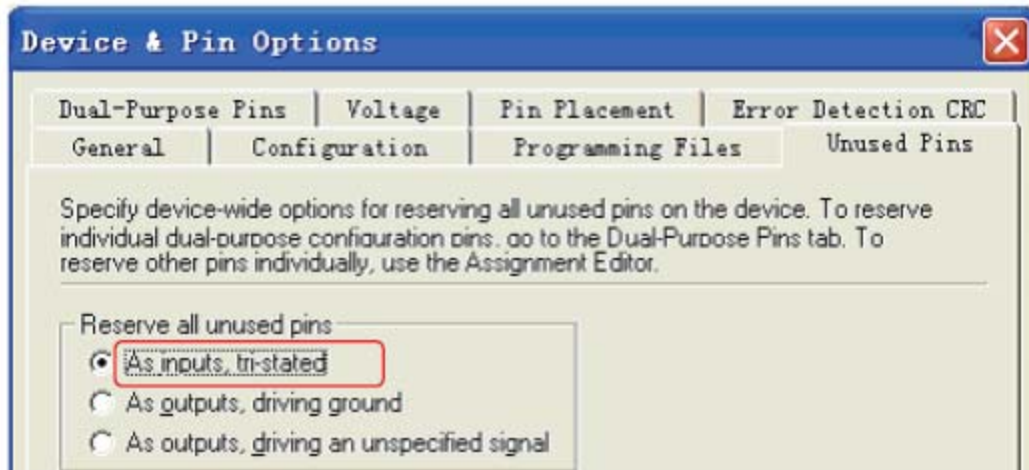


图 1.6 未定义管脚定义选择

1.3.3、SDRAM 存储器

SDRAM 通常用于需要大量存储且有成本要求的系统。SDRAM 比较便宜，但需要实现刷新操作、行列管理、不同延时和命令序列等逻辑。核心板采用 1 片 16 位总线的 8Mbytes SDRAM 器件 K4S641632H (1 MByte × 16bit × 4banks)，每片 SDRAM 都兼容 16Mbytes 的 K4S281632H (2 MByte × 16bit × 4 banks)、32Mbytes 的 K4S561632H (4 MByte × 16bit × 4 banks) 以及 64Mbytes，这样 SDRAM 的最大容量可达 128Mbytes。SDRAM 存储电路如图 1.7 所示。

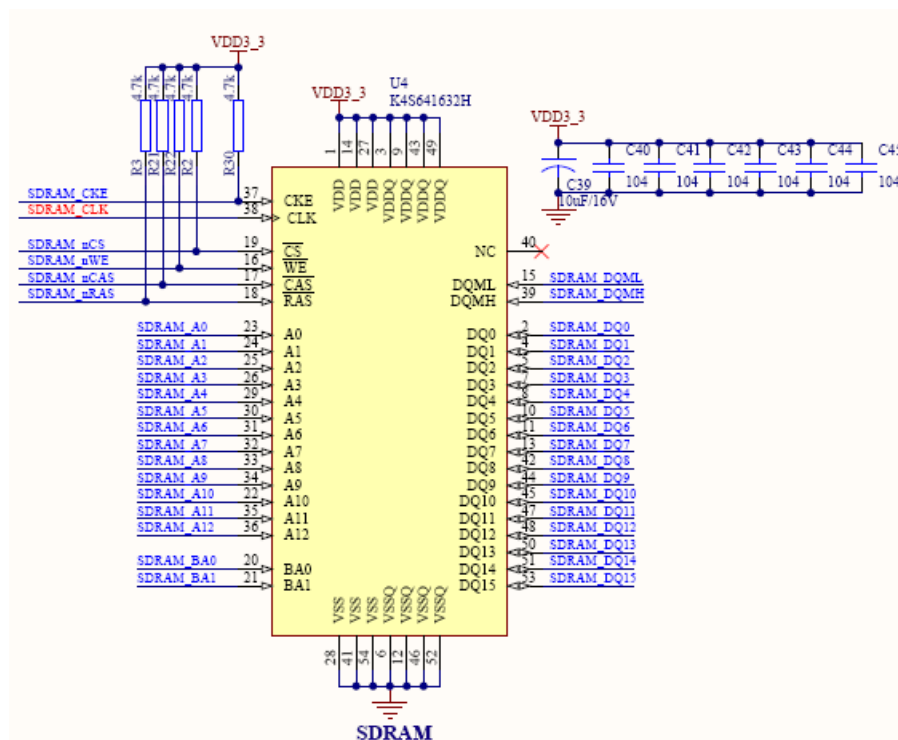


图 1.7 SDRAM 存储电路

1.4、配置电路

Cyclone FPGA 使用SRAM 单元来存储配置数据。FPGA 中的SRAM 是易失性的，每次上电之前，配置数据必须重新下载到FPGA 中。Cyclone FPGA 的配置方式包括：主动配置模式 (AS)、被动配置模式 (PS) 以及JTAG 配置模式，如表 1.1 所示。用户可以通过上述中的任意一种下载配置数据到FPGA。

配置模式	描述
主动串行配置(AS)	采用串行配置器件 (EPCS1、EPCS4、EPCS16、EPCS64)
被动配置 (PS)	1. 采用专用配置器件 (EPC1、EPC2、EPC4、EPC8、EPC16) ; 2. 采用配置控制器 (单片机、CPLD 等) 配合 Flash; 3. 下载电缆。
JTAG 配置	通过 JTAG 进行配置

表 1.1 Cyclone FPGA 配置模式表

核心板提供两种配置方法：

1. 调试时，使用运行在主计算机上的Quartus II 软件，通过JTAG 电缆连接到核心板上10 针 JTAG 接口直接下载配置数据到FPGA。用户随时可以进行JATG 模式的配置，但是要注意JTAG 模式配置是直接对FPGA 中的SRAM 单元编程，掉电后丢失，因此重新上电要重新下载。
2. 脱机运行的情况下，采用Intel® P30 Flash PC28F128P30B代替串行配置器件EPCS 进行主动配置 (AS)。主动串行配置模式(AS)是将配置数据存储在串行配置器件EPCS (或者Flash) 中，在每次系统上电时FPGA 会自动使用EPCS 中的配置数据进行配置。在脱机运行之前，要事先将配置数据通过编程器写入EPCS中。详细方法见附录《烧写Intel® P30 Flash》一文。

1.5、扩展接口

1.5.1、核心板与主板接口

核心板QuickSOPC 与主板采用2 个60 针的接口，接口定义如图 1.8 所示。

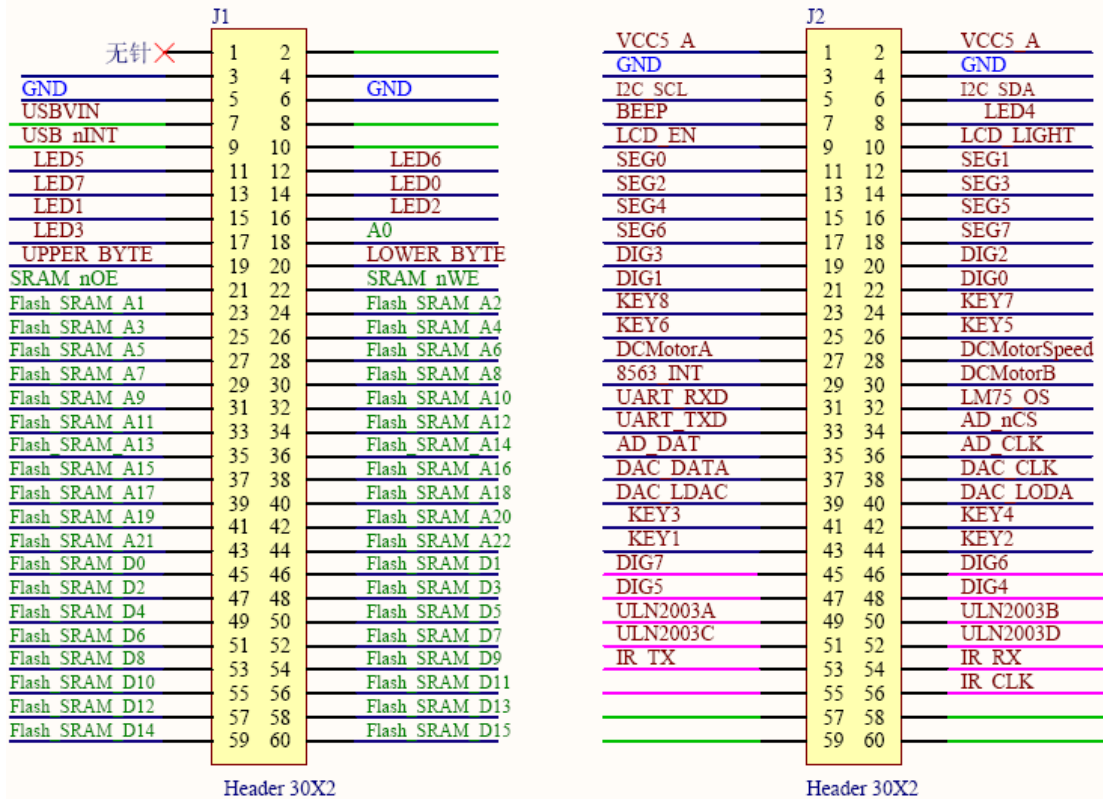


图 1.8 核心板接口

1.5.2、外扩IO 口PACK2

为了方便核心板的单独使用，在核心板上设计了一个外扩IO 口PACK2。PACK2 设计时，考虑了高速电路特性，可用户扩展高速外设，如高速AD、DA 等。当用户想单独使用核心板时，用户也可以在该PACK2 上扩展用户电路。**3C25核心板AD/DA PACK的丝印有误，J6上的A2和F7应交换，J5上的D3和E6应交换。**

PACK2 设计时采用了与主板完全兼容的引脚，并且也与周立功单片机发展有限公司的 MagicARM 2200 教学实验开发平台上的PACK 兼容，因此MagicARM2200 上所有的PACK都可使用。PACK2 的电路如图 1.9 所示。

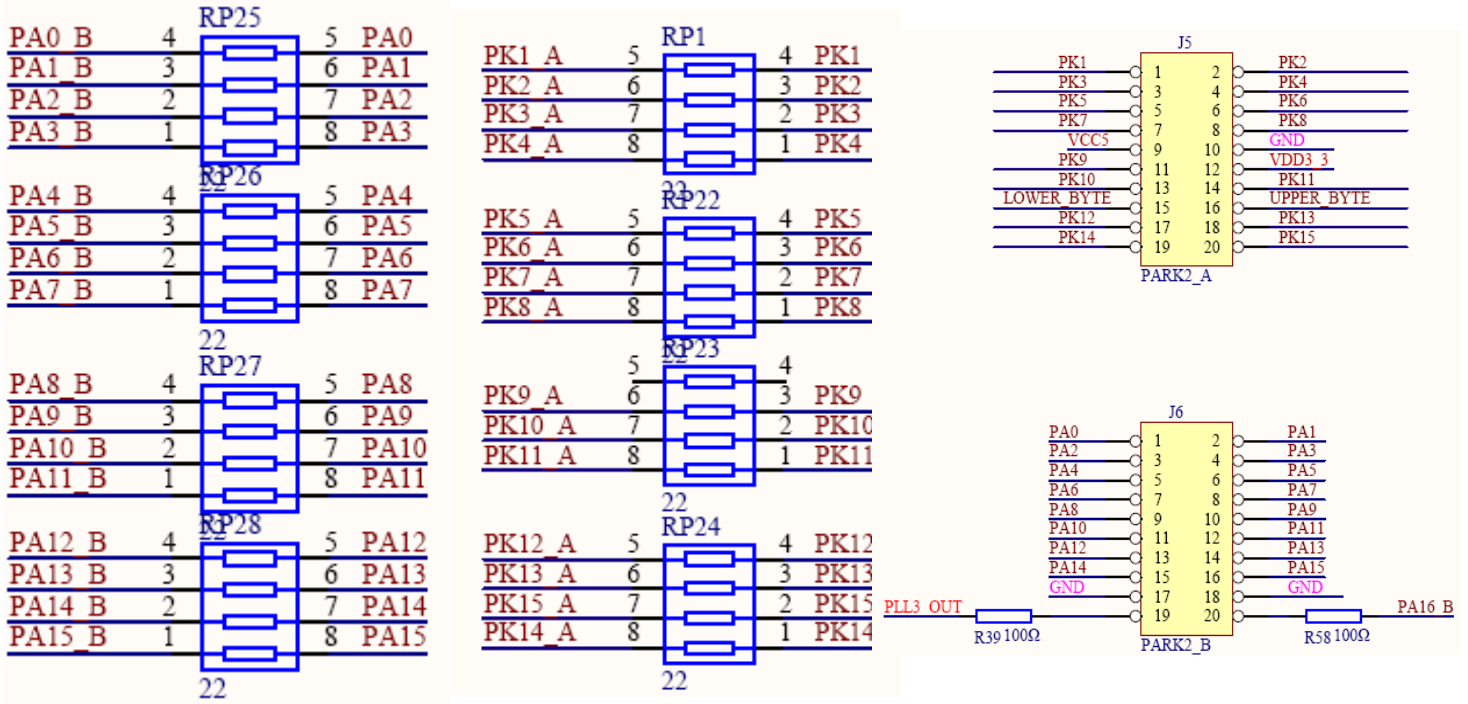


图 1.9 外扩IO 口PACK

1.6、独立按键及LED 电路

考虑到核心板对用户的实用性，核心板上除设计有FPGA、各类存储器以及用户扩展PACK 外，还设计有按键、LED（与主板共用IO 口）以及电源插座等。用户通过在扩展PACK上实现自己的功能电路（如主板上的某部分电路），核心板就能脱离主板而单独使用，这样核心板可用于用户自己的设计中，亦可用于电子设计大赛。独立按键及LED 电路如图 1.10 所示。需要注意，核心板上的按键和LED 是与主板上的一一对应的，所用的控制IO 口也是相同的。

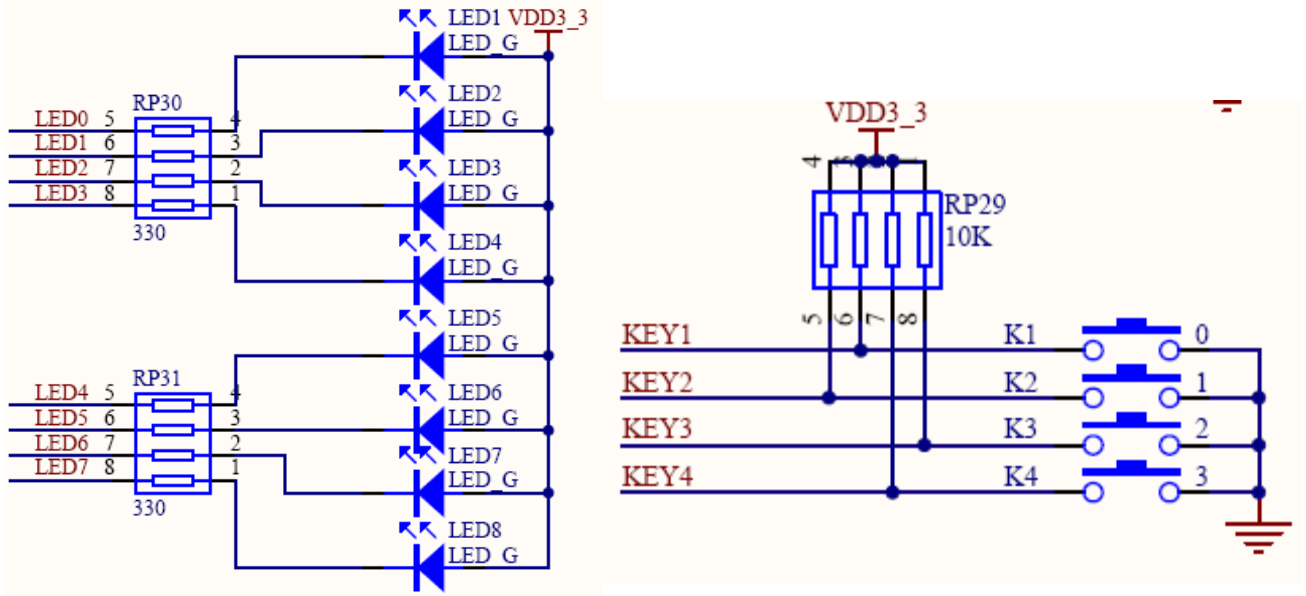


图 1.10 独立按键及LED电路

2、主板板电路：

2.1、电源电路

主板使用了一块专门的电源板进行供电，电源板所提供的电源有+5V、+12V 以及-12V，在主板上设计一个系统电源电路，使用LD0 芯片(低压差电源芯片)将5V 电源转换得到3.3V系统电源。

系统电源电路如图 1.11 所示，电源供电由J6 和J8 连接器输入实验箱主板，5V 电源经过LD0 芯片U12 稳压输出3.3V 电源。另外，主板上设计有模拟电路，需要一个5V 的模拟电源VCC5A，它通过5V 电源经过滤波得到。模拟地和数字地是通过0Ω 电阻进行隔离，以降低噪声干扰。

主板上的J1~J5 为电源输出接口，可以向用户板提供+12V、-12V、5V 和3.3V 电源，但要求负载功率不要过重，也不要与其它电源连接，否则可能导致器件损坏。

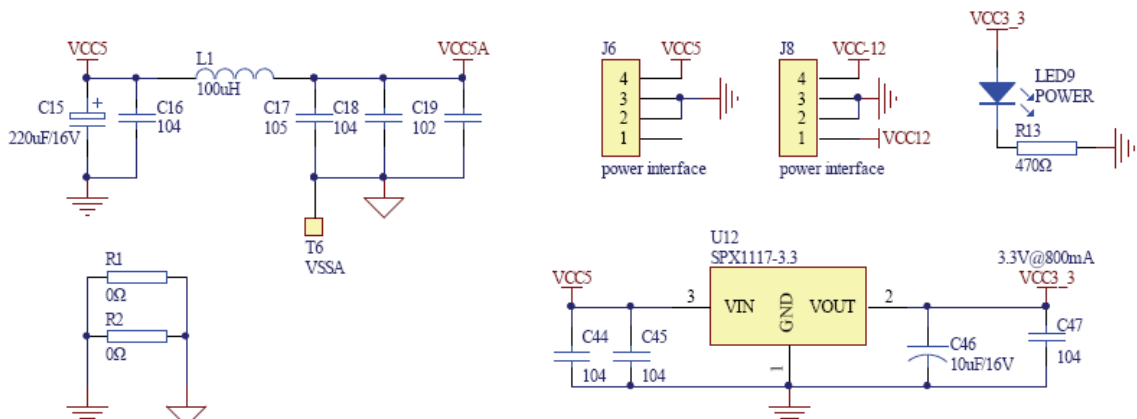


图 1.11 系统电源电路

2.2、按键及LED电路

主板上具有8 个独立按键和8 个独立LED，电路如图 1.12 所示，电路中低电平表示按键按

下，低电平点亮LED，一般LED的压降约为1.7V，LED点亮时的电流

$$I_{LED} = \frac{V_{DD} - V_{LED}}{R} = \frac{3.3 - 1.7}{220} \approx 0.00727(A), \text{即} 7.3\text{mA}.$$

电路中电阻RP9、RP10都是保护用的，防止FPGA的IO设为输出且为高电平在按键按下时直接对地短路。核心板上的按键和LED是与主板上的一一对应的，所用的控制IO口也是相同的。关于IO的分配请读者查阅附录A。

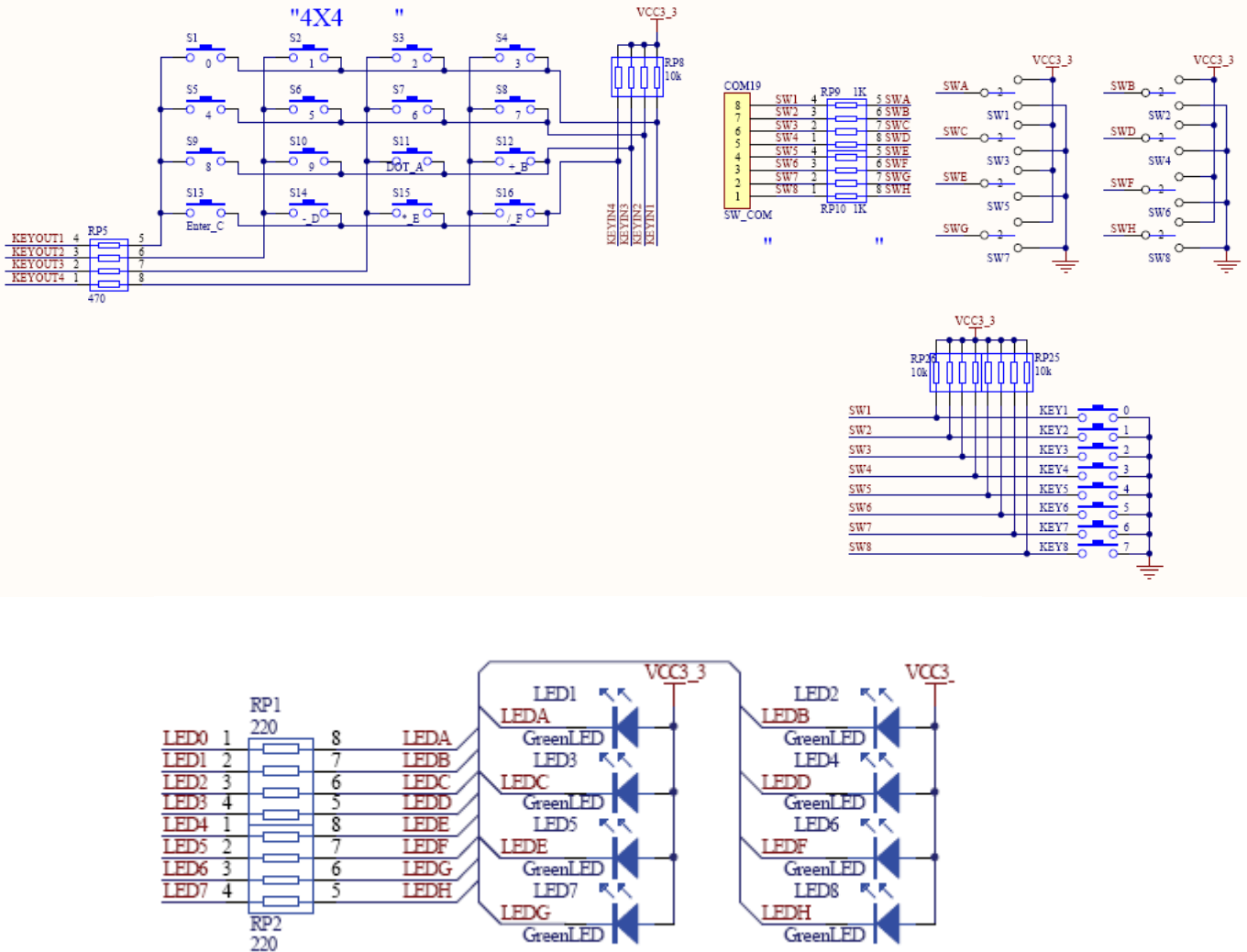


图 1.12 按键和LED电路

由于SmartSOPC+上的矩阵键盘、拨码开关、独立按键是共用I/O的,所以使用时需加注意,使用方法如下:

1、独立键盘的使用

使用独立按键时,需将SmartSOPC+主板上JP6中KEY1-KEY8的短路帽拔掉,然后用杜方线将JP6中KEY1-KEY8左边对应的排针和主板上的COM19(位于D6区)的SW1-SW8对应连接。然后确保所有的拨档开关(SW1-SW8)的拨档都向下。

2、拨档开关的使用

使用拨档开关时,需将SmartSOPC+主板上JP6中KEY1-KEY8的短路帽拔掉,然后用杜方线将JP6中KEY1-KEY8左边对应的排针和主板上的COM19(位于D6区)的SW1-SW8对应连接。使用过程中确保不按独立按键(KEY1-KEY8)。

3、矩阵键盘的使用

使用矩阵键盘时时，只需将SmartSOPC+主板上JP6 中KEY1-KEY8 的短路帽短接即可。

2.3、蜂鸣器电路

如图 1.13 所示，蜂鸣器使用PNP 三极管进行驱动控制，蜂鸣器使用的是交流蜂鸣器。当在BEEP 输入一定频率的脉冲时，蜂鸣器蜂鸣，改变输入频率可以改变蜂鸣器的响声。因此可以利用一个PWM 来控制BEEP，通过改变PWM 的频率来得到不同的声响，也可以用来播放音乐。若把JP7 断开，Q4 截止，蜂鸣器停止蜂鸣。

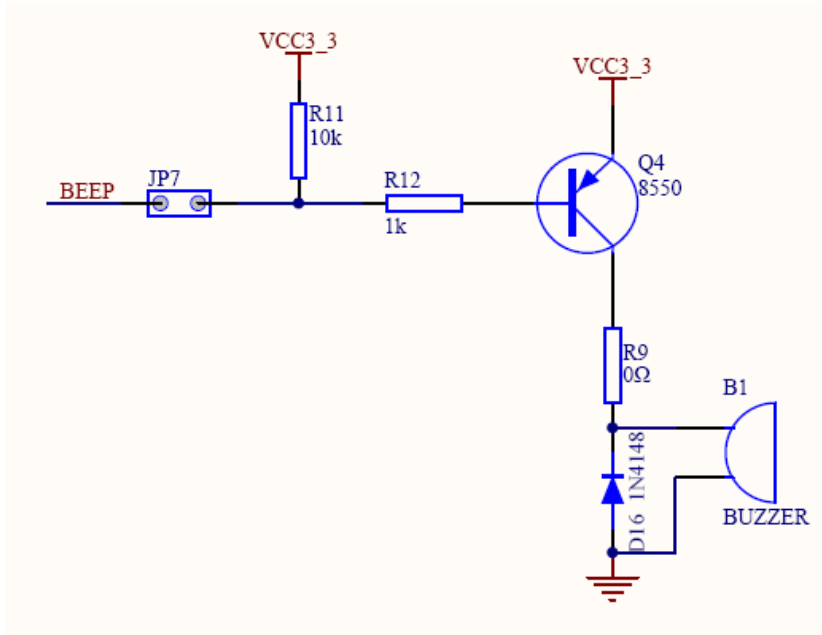


图 1.13 蜂鸣器电路

2.4、七段数码管显示电路

主板上七段数码管显示电路如图 1.14 所示，RP4 和RP6 是段码上的限流电阻，位码由于电流较大，采用了三极管驱动。从电路可以看出，数码管是共阳的，当位码驱动信号为0时，对应的数码管才能操作；当段码驱动信号为0 时，对应的段码点亮。

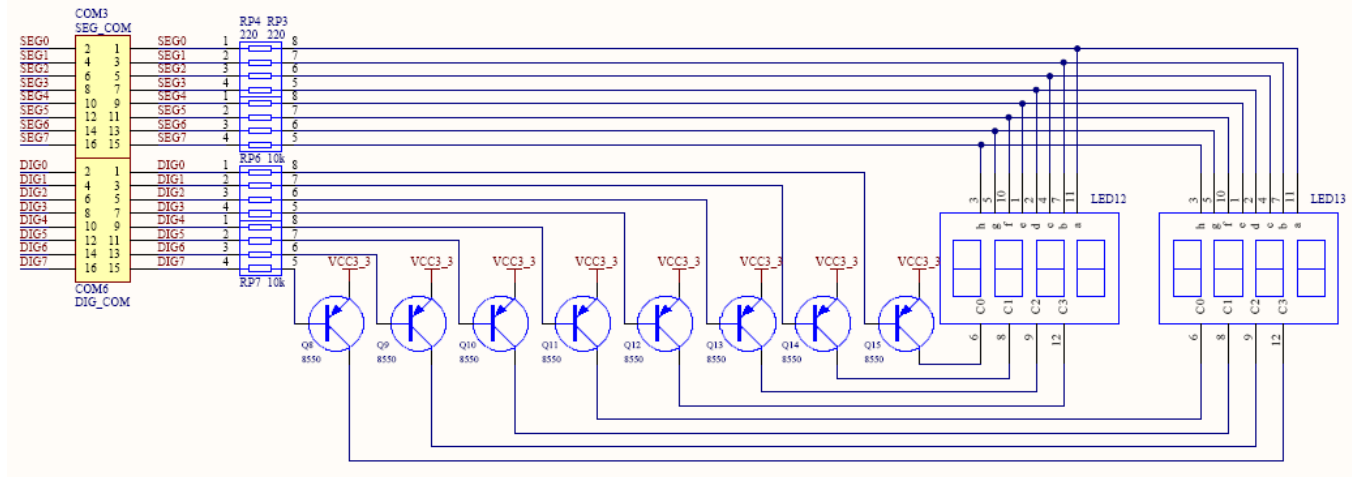


图 1.14 七段数码管显示电路

电路中的COM3(SEG_COM)和COM6 (DIG_COM) 是对外的逻辑分析仪测试点以及接口。

2.5、液晶显示电路

主板配有240×128 图形点阵液晶屏，液晶屏内部带有液晶控制器T6963。液晶驱动电路如图 1.15 所示。

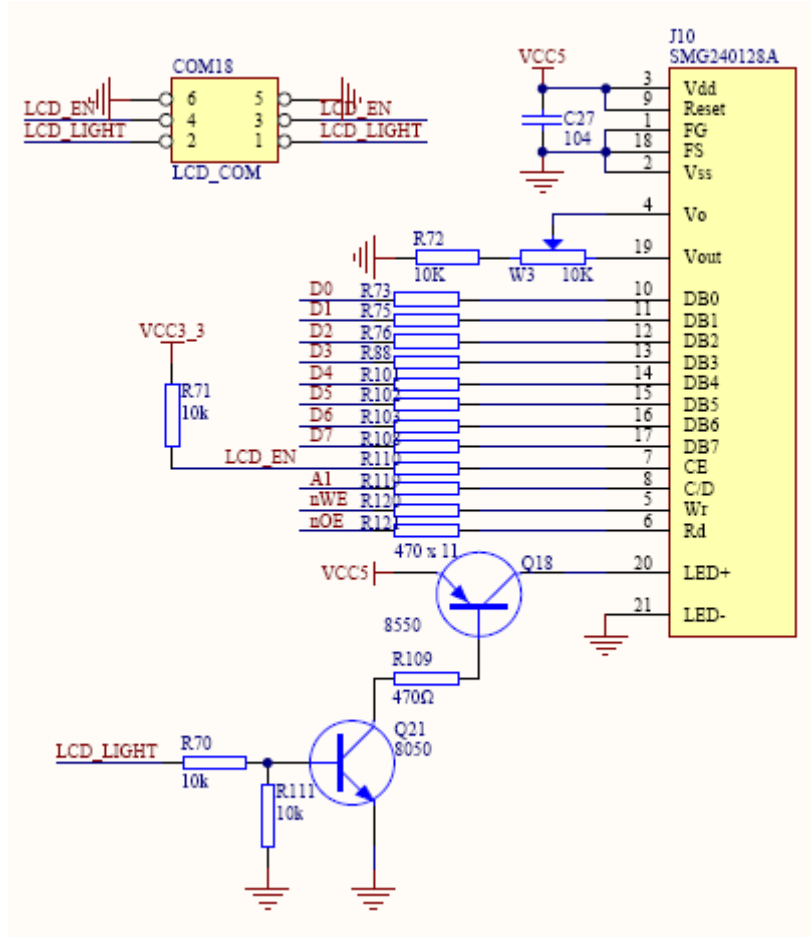


图 1.15 液晶显示电路

电路中LCD_LIGHT 是进行液晶背光控制的，当为高电平时，液晶背光打开，否则关闭背光。电路中的COM18 (LCD_COM) 是对外的逻辑分析仪测试点以及接口。

2.6、16*16 LED 点阵电路

主板上16*16LED 点阵电路如图 1.16 所示。电路中采用SPI 接口的方式对LED 点阵进行操作，LATTICE_SI 对应SPI 的MOSI，LATTICE_STR 对应SPI 的nCS，LATTICE_SCK对应SPI 的SCK。U17、U18、U19 以及U20 (74HC595) 构成一个串入并出的32bit 移位寄存器，当LATTICE_STR 低电平有效时，32bit 数据在32 个LATTICE_SCK 时钟下由LATTICE_SI 串行输入，当LATTICE_STR 由低电平变为高电平时，32bit 数据并行输出。在主板上数据输出时LDA-LDP 对应行，而LD1-LD16 对应列，最先移入的数据被当作16 列 (LD16)，最后移入的被当作第1 行 (LDA)，编程时要注意。

电路中的COM13 (LATTICE_COM) 是对外的逻辑分析仪测试点以及接口。这些信号并没有连接到相应的FPGA 管脚，因此要对LED 点阵操作必须通过连线控制。

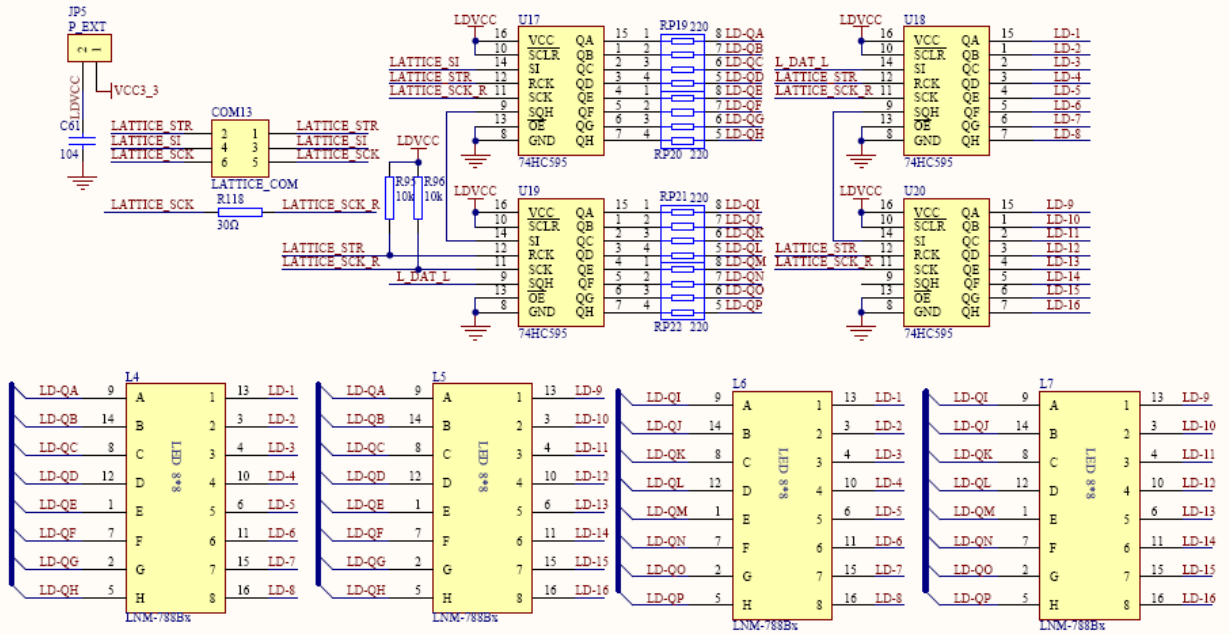


图 1.16 LED点阵电路

2.7、RS232串口电路

RS232 串口电路如图 1.17 所示，由于是3.3V 系统，所以使用了SP3232E 进行RS232电平转换，SP3232E 是3.3V 工作电压的RS232 转换芯片。LED 用于指示串口的工作状态。

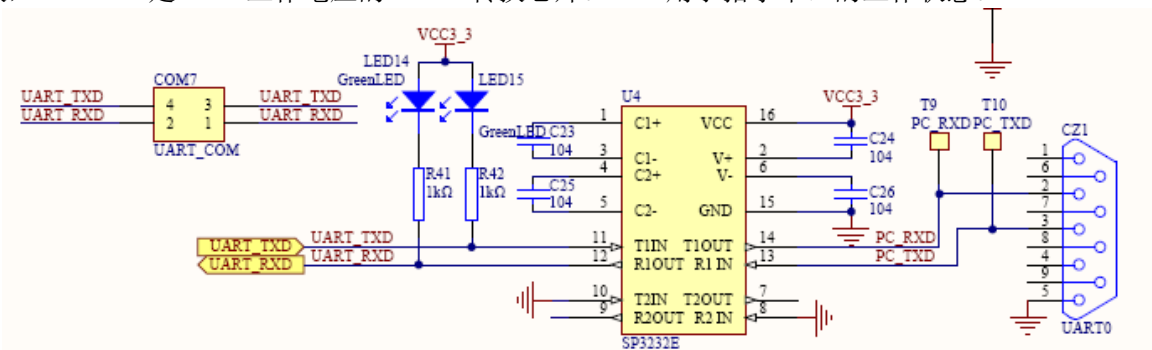


图 1.17 RS232串口电路

电路中的COM7 (UART_COM) 是对外的逻辑分析仪测试点以及接口。这些信号已经连接到相应的FPGA 管脚上，使用时不需要进行连线控制。

2.8、RS485接口电路

RS485 转换电路如图 1.18 所示。

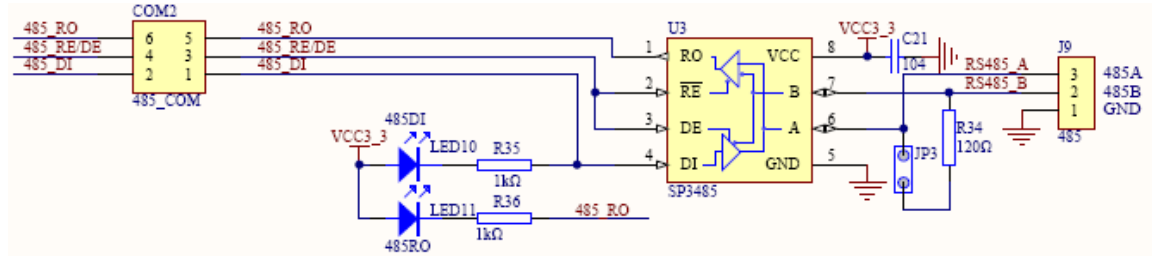


图 1.18 RS485接口电路

由于系统是3.3V系统，所以使用了SP3485进行RS485电平转换，SP3485是3.3V工作电源的半双工RS485收发器。在图1.18中，使用485_RE/DE连接到RS3485的DE引脚和RE引脚，当485_RE/DE输出低电平时，RS3485的接收器使能，RS485总线上的数据将会从RO引脚输出；当485_RE/DE输出高电平时，RS3485的驱动器使能，通过DI进行数据发送。

在RS485总线中，终端通常需要接一个终端电阻，如图1.18所示电阻R34，当需要使用此电阻时将JP3短接即可。

电路中的COM2(485_COM)是对外的逻辑分析仪测试点以及接口。这些信号没有连接到核心板的控制管脚上，使用时必须进行连线控制。

2.9、红外通讯电路

红外通信是利用950nm近红外波段的红外线作为传递信息的媒体，即通信信道。发送端采用脉冲位置调制(PPM)方式，将二进制数字信号调制成某一频率的脉冲序列，并驱动红外发射管以光脉冲的形式发送出去；接收端将接收到的光脉冲转换成电信号，再经过放大、滤波等处理后送给解调电路进行解调，还原为二进制数字信号后输出。简而言之，红外通信的实质就是对二进制数字信号进行调制与解调，以便利用红外信道进行传输；红外通信接口就是针对红外信道的调制解调器。红外收发电路如图1.19所示。

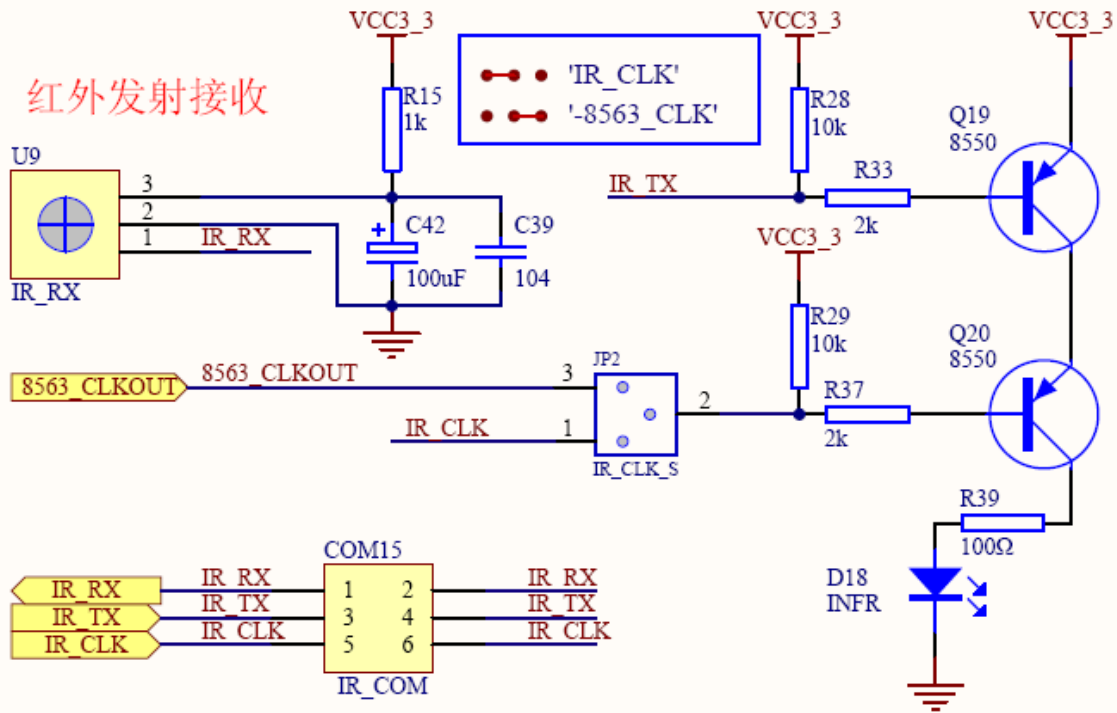


图 1.19 红外通讯电路

红外发送器电路包括调制脉冲、驱动管Q19和Q20、红外发射管D18等部分。其中调制脉冲用作载波信号，可由实时时钟芯片PCF8563的时钟输出端来产生，约32.768kHz，也可以由FPGA来产生38kHz左右的脉冲信号，使用哪种载波信号可通过JP2跳线选择。

红外发送的工作原理为：串行数据由IR_TX送出并驱动Q19管，数位“0”使Q19管导通，通过Q20管调制成38kHz的载波信号，并利用红外发射管D18以光脉冲的形式向外发送。数位“1”使Q19管截止，红外发射管D18不发射红外光。一般将传送的波特率设为1200bps，则每个数位“0”对应32个载波脉冲调制信号的时序。红外接收电路选用专用红外接收模块1838。该接收模块是一个三端元件，使用单电源+3.3V电源。

红外接收的工作原理为：首先，通过红外光敏元件U9将接收到的载波频率为38kHz的脉冲

调制红外光信号转化为电信号，再由内部电路进行放大、滤波、解调，最后由IR_RX 输出。

为保证红外接收模块接收的准确性，要求发送端载波信号的频率应尽可能接近38kHz。再有，发送的数位“0”至少要对14 个载波脉冲，这就要求传送的波特率不能超过2400bps。

电路中的COM15(IR_COM)是对外的逻辑分析仪测试点以及接口。这些信号已经连接到相应的FPGA 管脚上，使用时不需要进行连线控制。

2.10、以太网接口电路

10M 以太网接口电路如图 1.20 所示。以太网芯片是广泛使用的RTL8019AS，它是一种高度集成的以太网控制芯片，能简单地实现Plug and Play 并兼容NE2000、掉电等特性。

电路使用16 位总线方式进行控制，即数据总线D0~D15 与芯片的SD0~SD15 连接，地址线也进行相对应的连接，CPU 地址地A1 连接到RTL8019 的A0。将选线与芯片的A5 相连，RTL8019 在SA5(8019_nCS)为低电平时选通。8019_RST 为CPU 的输出引脚，用来控制RTL8019 的复位，高电平进行复位；8019_INT 为RTL8019 的中断信号，高电平中断。

由于RTL8019 是5V 供电，而与之连接的CPU (FPGA、ARM、51) 为3.3V 系统，所以数据、地址以及控制线都串了一个220Ω 的电阻。

RTL8019 的应用及连接方法的详细说明请参考RTL8019 芯片的数据手册。

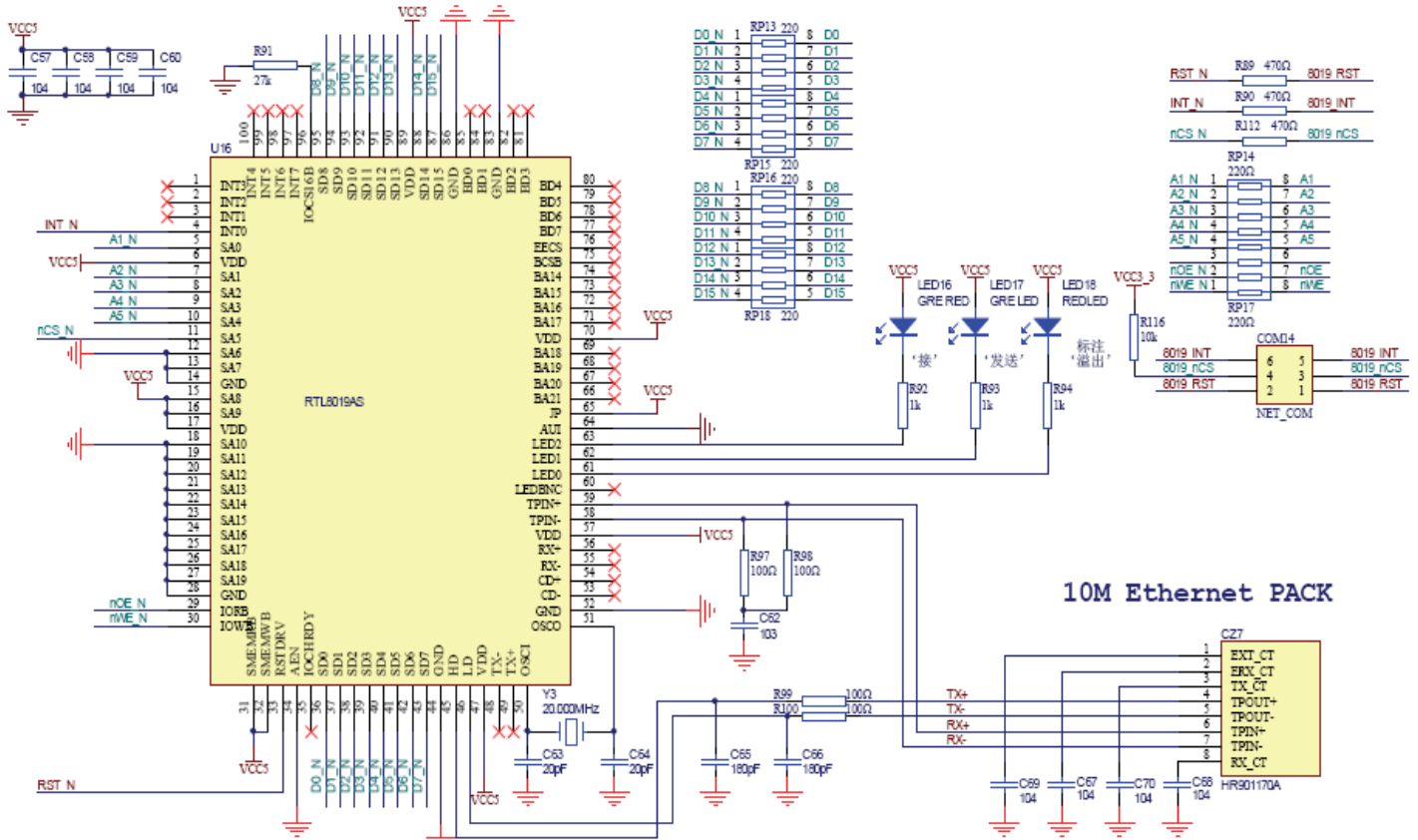


图 1.20 以太网接口电路

电路中的COM14(NET_COM)是对外的逻辑分析仪测试点以及接口。这些信号没有连接到核心板的控制管脚上，使用时必须进行连线控制。

2.11、USB接口电路

USB 接口电路如图 1.21 所示，采用PDIUSB12 芯片。PDIUSB12 是一款性价比很高的USB 器件，符合USB2.0 FullSpeed 规范。

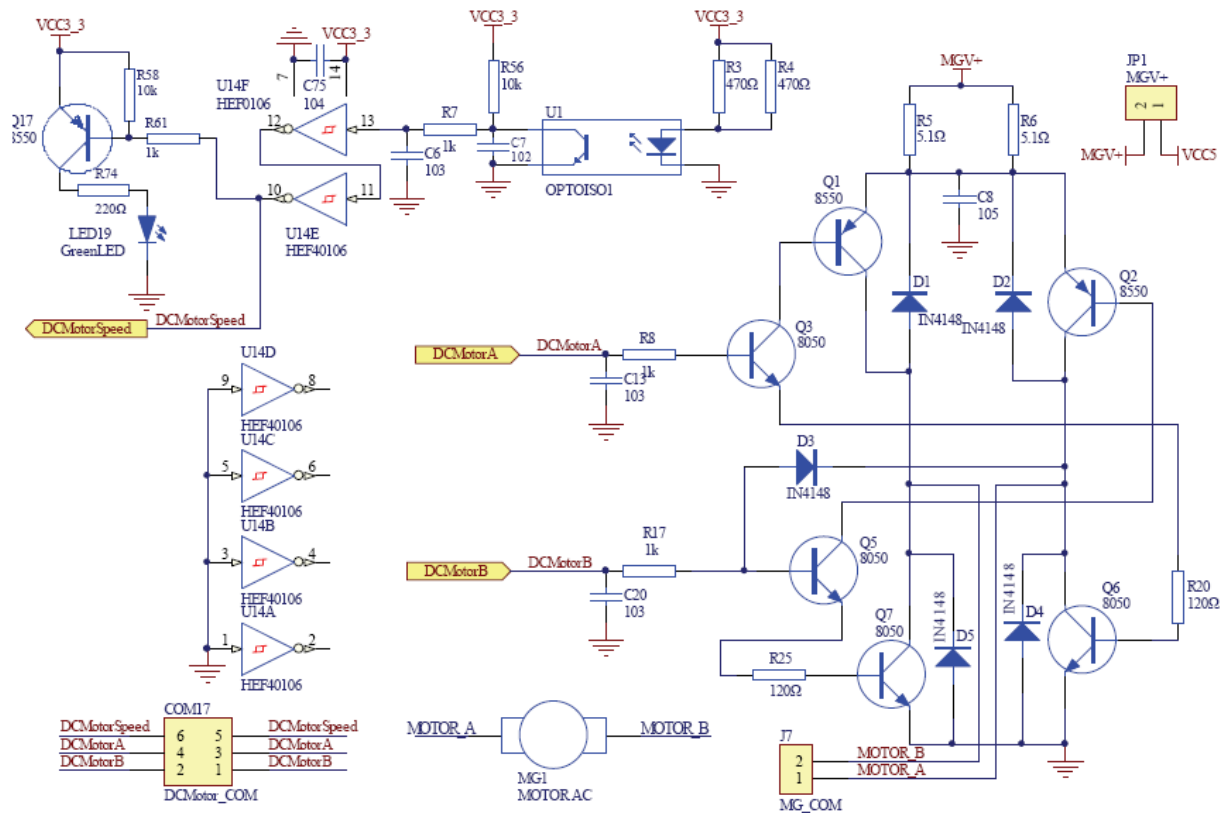


图 1.23 直流电机电路

如图 1.23 所示，直流电机控制使用了H 桥驱动电路，当DCMotorA 输出高电平时，则Q3、Q6 导通→Q1 导通→ MOTOR_B 点为MGV+，Q6 导通→ MOTOR_A 点为GND，此时直流电机将会正转。由于Q6 的集电极通过一个二极管D3 连接到H 桥的另一个控制端DCMotorB，将DCMotorB 控制端电压钳在1.0V 以下，所以不管DCMotorB 输出是高电平还是低电平，Q5、Q7 都会截止→Q2 截止，不会造成H 桥短路故障。

当DCMotorA 输出为低电平时，则Q3、Q1 截止→Q6 截止，DCMotorB 的输出电平可以控制电机反转或停机。若DCMotorB 输出高电平，则Q5、Q7 导通→Q2 导通→ MOTOR_A点为MGV+，Q7 导通→ MOTOR_B 点为GND，此时直流电机将会反转。当DCMotorB 输出低电平时，Q5、Q7 都会截止→Q2 截止，电机停机。

二极管D1、D2、D4、D5 续流二极管，用于释放电机线圈上产生的反电动势。电阻R5、R6 为限流/保护电阻。使用直流电机电路时，要将电源跳线JP1 短接。

主板上还设计有电机测速电路，这样可进行闭环调试实验。测试电路使用槽型光电开关，当电机转动时，当安装在电机转轴上转盘从槽型光电开关转过，转盘上均匀分布有四个孔，电机转动一周，光电开关工作4 次，在DCMotorSpeed 输入管脚上出现4 个低脉冲。检测1秒内DCMotorSpeed 上低脉冲的个数除以4 即可得到电机每秒的转速。U14 为施密特触发器，对光电开关输出信号进行整形处理，LED19 用于指示光电开关工作状态。

电路中的COM17 (DCMotor_COM) 是对外的逻辑分析仪测试点以及接口。

2.14、VGA接口电路

主板上VGA 的接口电路如图 1.24 所示。本电路采用电阻网络的方法来产生VGA 所需的不同电压信号，输入端共用8 个信号线，因此能产生256 色。VGA_HSYNC 是行同步信号，VGA_VSYNC 是场同步信号。电路中的二极管起过压保护作用。

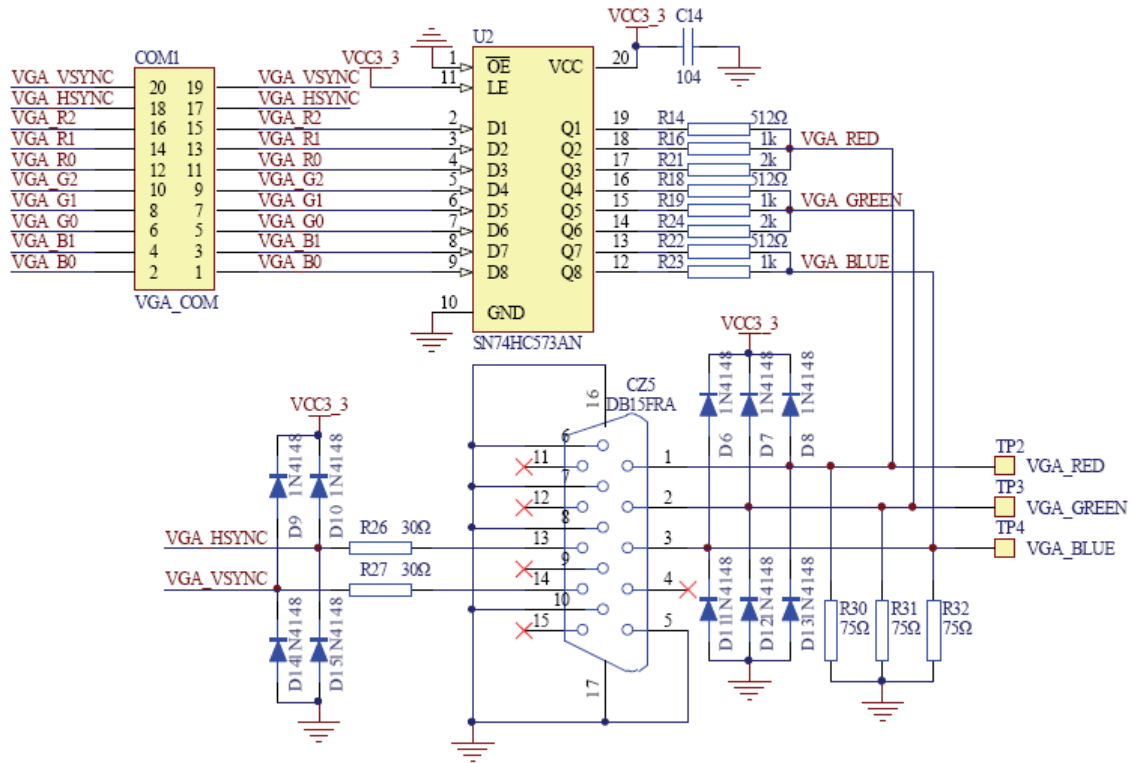


图 1.24 VGA接口电路

电路中的COM1 (VGA_COM) 是对外的逻辑分析仪测试点以及接口。这些信号并没有连接到相应的管脚上，使用时必须进行跳线控制。

2. 15、PS/2键盘鼠标接口电路

PS/2 键盘鼠标接口电路原理如图 1.25 所示，其中CZ3 接口为鼠标接口，CZ4 接口为键盘接口。根据PS/2 协议要求，PS/2 键盘、鼠标使用5V 电源供电，接口的数据线(Data)和时钟线(Clock)均要接上拉电阻。由于FPGA 的I/O 可以承受5V 电压，所以PS/2 接口的数据线(Data)和时钟线(Clock)可与FPGA 直接连接，但最好在中间串接一个小阻值的电阻。

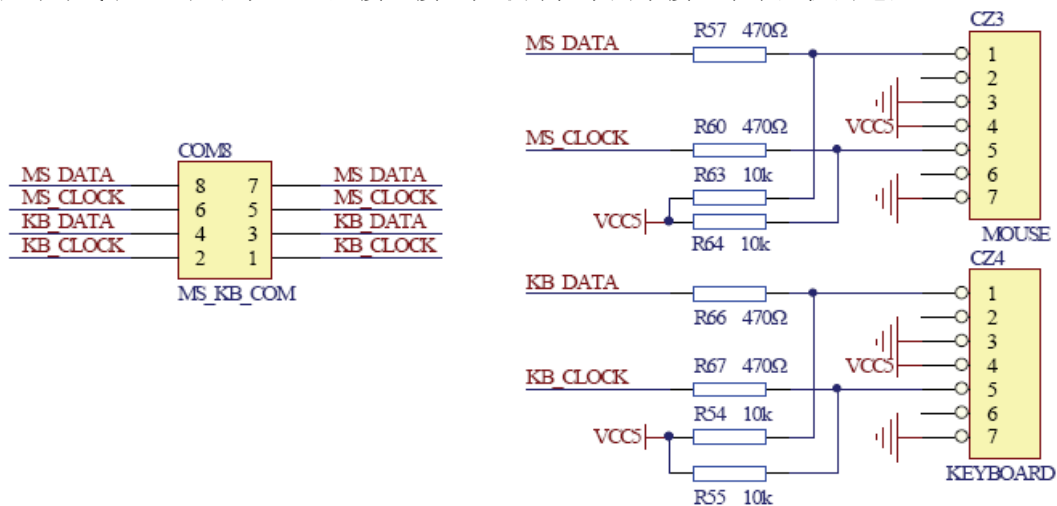


图 1.25 PS/2键盘鼠标接口电路

电路中的COM8 (MS_KB_COM) 是对外的逻辑分析仪测试点以及接口。这些信号并没有连接到相应的管脚上，使用时必须进行连线控制。

2.16、串行DA、AD 电路

实验箱主板上的DAC 电路如图 1.26 所示。

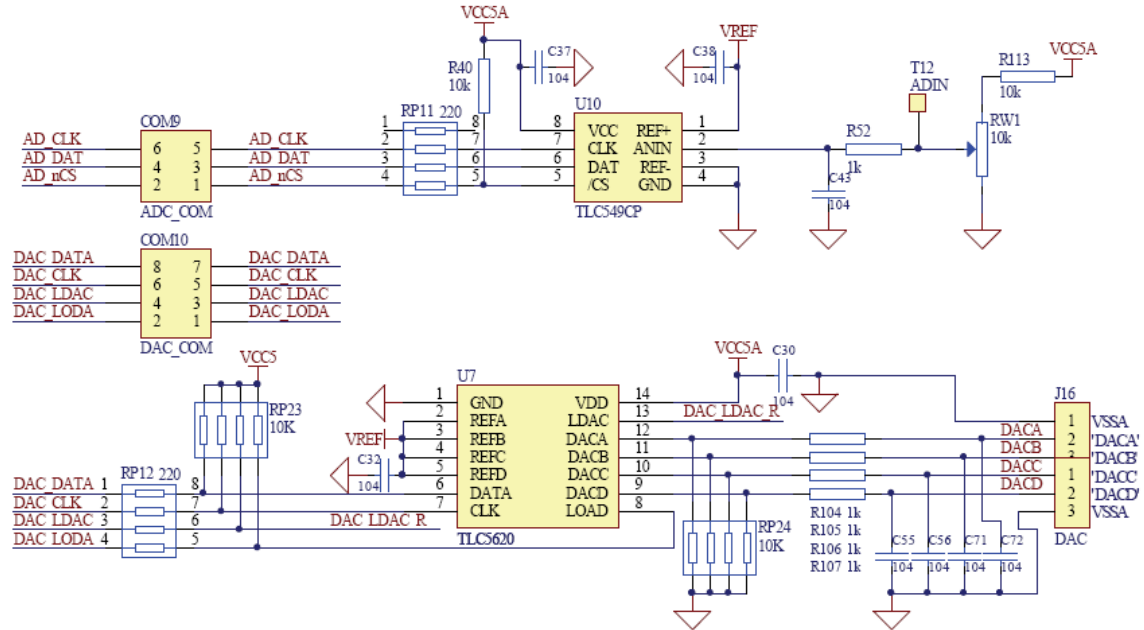


图 1.26 串行DA、AD 电路

DAC 电路使用一片串行接口的4 通道8 位D/A 转换器TLC5620, TLC5620 具有半缓冲输出功能, 可编程输出量程功能。TLC5620 的每一路D/A 通道均需要参考电源, 由REFA、REFB、REFC 和 REF D 引脚输入。

串行AD 电路采用单通道8 位A/D 转换器TLC549C, 转换所需的电压基准由REF+输入, 电压基准定为2.5V, 由图 1.36 所示的电压基准电路产生。主板上设计了一个电位器,

用于模拟转换的输入。电压基准电路如图 1.27 所示。

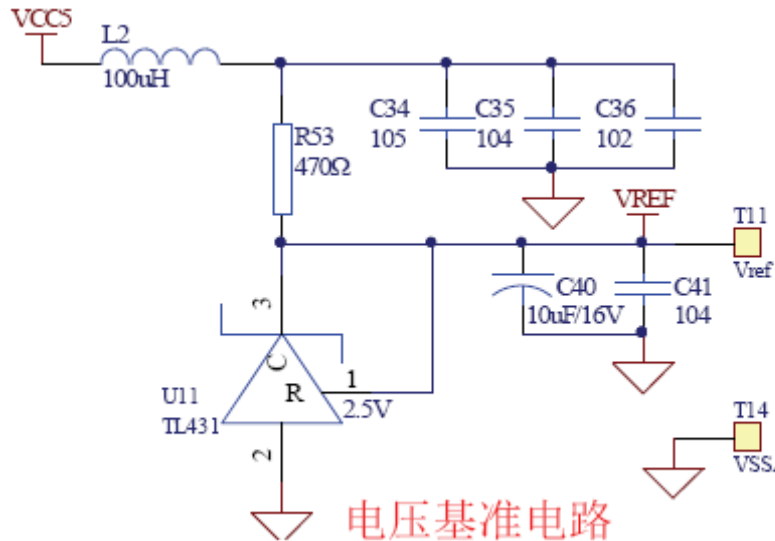


图 1.27 电压基准电路

TL431 是一个具有良好的热稳定性能的、低噪声的三端可调分流基准源(温度系统为30ppm/°C)，实验箱主板上通过TL431 产生2.5V 的参考电源，然后连接到TLC5620 的REFA、REFB、REFC 和REFD 引脚，作为D/A 转换器的参考电压。

图 1.26 所示电路中的COM9(ADC_COM)、COM10(DAC_COM)是对外的逻辑分析仪测试点以及接口。这些信号都已经连接到相应的管脚上，使用时不需要进行连线控制。

2.17、实时时钟电路

实时时钟电路如图 1.28 所示。实时时钟芯片采用的是I2C接口的低功耗的CMOS 实时时钟 / 日历芯片PCF8563T，它提供一个可编程时钟输出，一个中断输出和掉电检测器，所有的地址和数据通过I2C总线接口串行传递。最大总线速度为400kBits/s，每次读写数据后，内嵌的字地址寄存器会自动产生增量。PCF8563T具体使用方法请读者自行参考光盘中的数据手册。

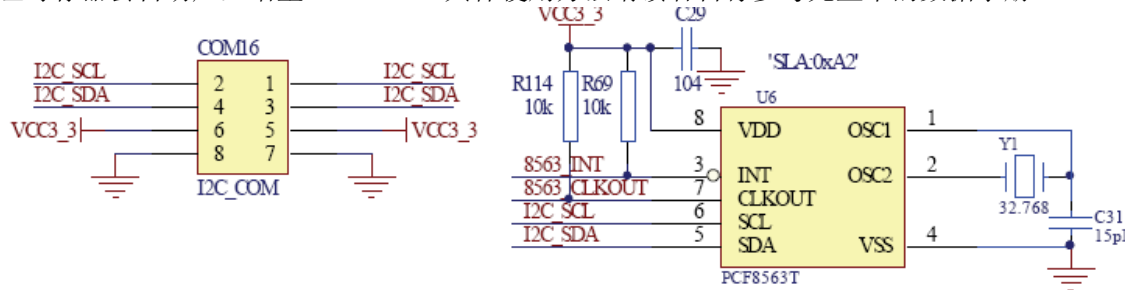


图 1.28 实时时钟电路

由于核心板上的I2C 总线已经提供了上拉电阻，所以此处没有再加上拉电阻。PCF8563T的CLKOUT 以及INT 都是开漏输出，所以都需加上拉电阻。

电路中的COM16(I2C_COM)是对外的逻辑分析仪测试点以及接口。这些信号都已经连接到相应的管脚上，使用时不需要进行连线控制。

2.18、数字温度传感器电路

数字温度传感器电路如图 1.29 所示。

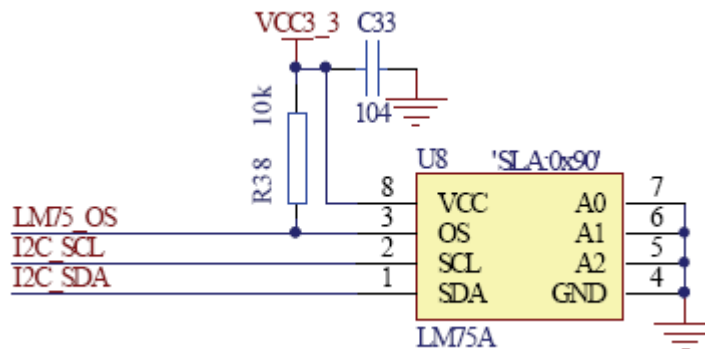


图 1.29 数字温度传感器电路

LM75A 是一个高速I2C 接口的温度传感器，可以在-55°C~+125°C的温度范围内将温度直接转换为数字信号，并可实现0.125°C的精度。MCU 可以通过I2C 总线直接读取其内部寄存器中的数据，并可通过I2C对4 个数据寄存器进行操作，以设置成不同的工作模式。LM75A有3 个可选的逻辑地址管脚，使得同一总线上可同时连接8 个器件而不发生地址冲突。LM75A 可配置成不同的工作模式。它可设置成在正常工作模式下周期性地对环境温度进行监控，或进入关断模式来将器件功耗降至最低。OS 输出有2 种可选的工作模式：OS比较器模式和OS 中断模式，OS 输出可选择高电平或低电平有效。

LM75A 工作电压为3.0V~5.5V，采用3.3V 直接供电。设计中只采用了一个LM75A，所以I2C

的地址线A0-A2 接地即可，其I2C 地址为0x90。由于核心板上的I2C 总线已经提供了上拉电阻，所以此处没有再加上拉电阻。LM75A 的OS 是开漏输出，所以都需加上拉电阻。

2.19、SD/MMC 卡接口电路

实验箱主板设计了SD 卡接口，支持SD/MMC 卡的读写，电路原理如图 1.30 所示。

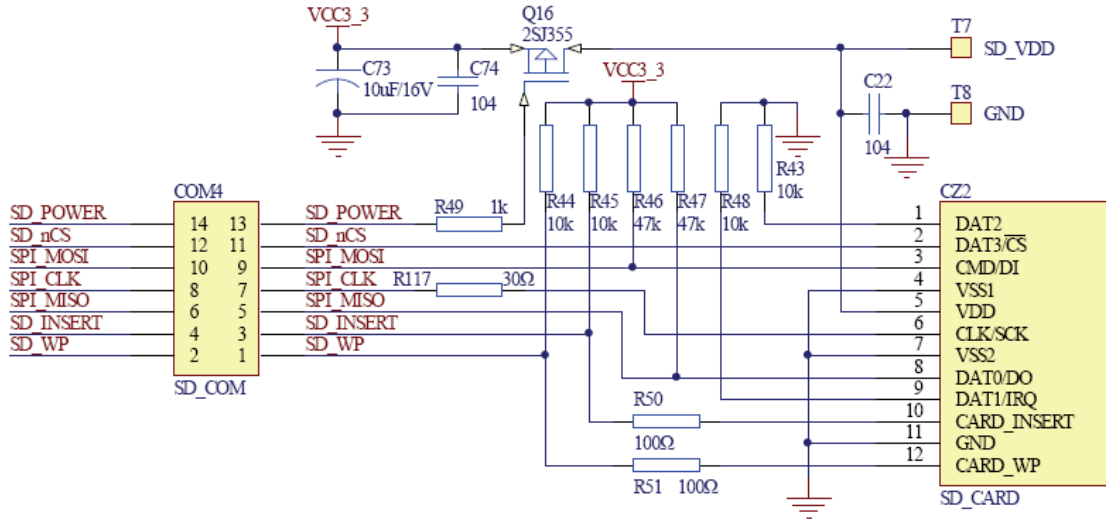


图 1.30 SD/MMC 卡接口电路

SD/MMC 卡可以采用SD 总线访问，也可以采用SPI 总线访问，考虑到Altera 提供的IP 中有SPI 接口而没有SD 总线接口，所以直接使用SPI 接口与其连接。如果采用I/O 口模拟SD 总线，不但增加了软件的开销，而且大大降低了总线数据传输的速度。

SD 卡接口电路包括以下几个部分。

- SPI 总线

如图 1.30 所示，SPI 接口的SPI_MOSI、SPI_CLK 和SPI_MISO 直接连接到卡座的相应接口，其中SPI 的两个数据线SPI_MOSI、SPI_MISO 还分别接上拉电阻，这是为了使本电路可以与MMC 卡的接口兼容。片选信号使用SD_nCS。

SPI 模式下无需用到的信号线DAT2 和DAT1 分别接下拉电阻。

- 卡供电控制

卡的供电采用可控方式，这是为了防止SD/MMC 卡进入不确定状态时，可以通过对卡重新上电使卡复位而无需拔出卡。可控电路采用P 型MOS 管2SJ355，由SD_POWER 进行控制，当SD_POWER 输出高电平时，2SJ355 关断，不给卡供电；当SD_POWER 输出低电平时，2SJ355 开通，给SD/MMC 卡供电。

采用2SJ355 的目的是当它开通时，管子上的压降比较小。2SJ355 的相关特性请见其数据手册。用户也可以采用其它P 型的MOS 管，但是要考虑管子开通时，漏极与源极之间的压降要足够小（保证SD/MMC 卡的工作电压在允许范围内），管子允许通过的电流也要满足卡的要求，一般一张SD/MMC 卡工作时的最大电流通常为45mA 左右，所以选用的MOS管要求允许通过100mA 左右的电流。

- 卡检测电路

卡检测电路包括两部分：卡是否完全插入到卡座中和卡是否写保护。

检测信号由卡座的两个引脚以电平的方式输出。当卡插入到卡座并插入到位时，由于卡座内部触点连接到GND，SD_INSERT 为低电平；当卡拔出时，该引脚由电阻R45 上拉为高电平。卡是否写保护的检测与卡是否完全插入到卡座中的检测原理是一样的。

电路中的COM4(SD_COM)是对外的逻辑分析仪测试点以及接口。这些信号并没有连接到相应的管脚上，使用时必须进行连线控制。

2.20、外设PACK 接口电路

实验箱主板上设计了外设PACK 接口，电路如图 1.31 所示，通过PACK，用户可扩展各种功能的总线型外设。PACK 接口具有23 根地址总线A0~A22，16 根数据总线D0~D15，读/写信号OE、WE、UPPER_BYTE 和LOWER_BYTE，片选信号为EXT_nCS。PACK 上还有6 个I/O 口，这样就极大的方便了与外部I/O 器件进行连接。J19 和J20 用于逻辑分析仪的测试点。

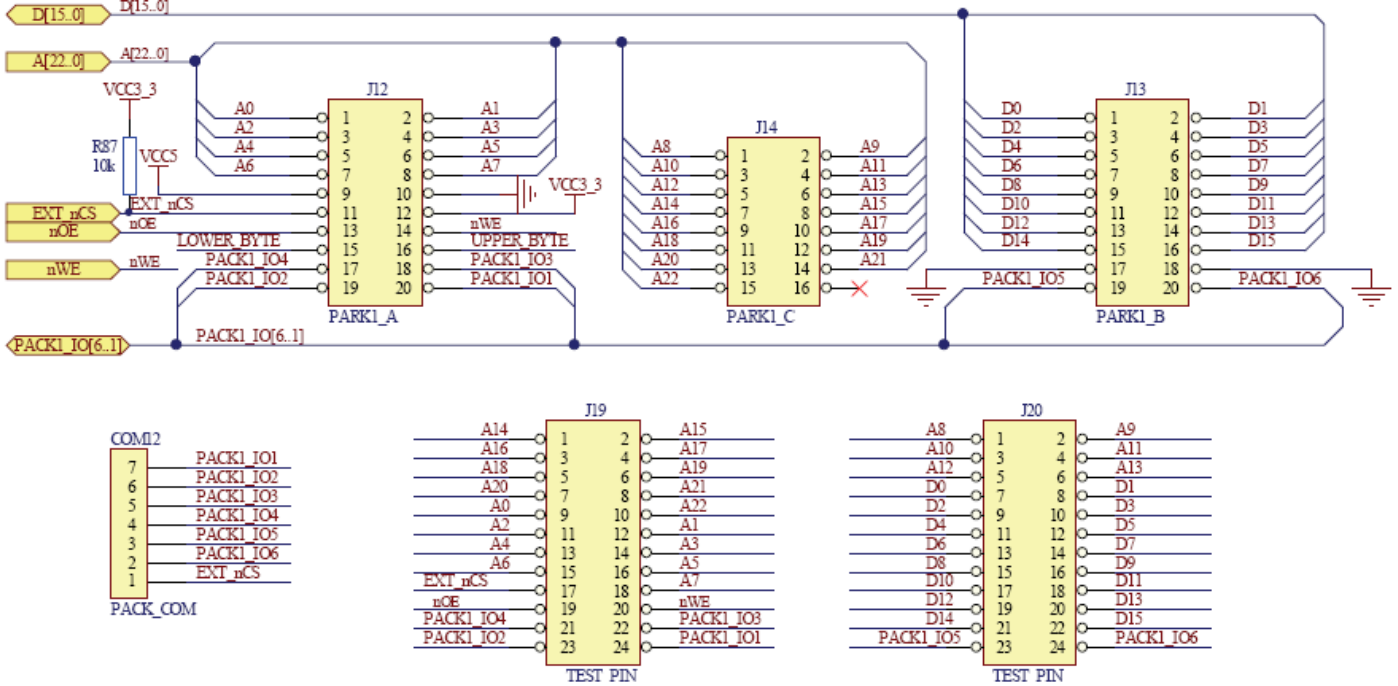


图 1.31 外设PACK 接口电路

COM12(PACK_COM)是操作PACK 的连线接口。这些信号并没有连接到相应的管脚上，使用时必须进行连线控制。PACK1 设计时采用了与主板完全兼容的引脚，并且也与周立功单片机发展有限公司的MagicARM 2200 教学实验开发平台上的PACK 兼容，因此MagicARM2200 上所有的PACK 都可使用。

3、高速AD/DA PACK 板：

高速AD/DA PACK 的电路原理图见附录B AD_DA PACK 原理图。高速AD/DA 板具有1 路DA(可升级到2 路，加一个DA 芯片即可)；1 路AD。

D/A 转换器使用的是TI 公司的125MSPS 单路10bit 器件THS5651A，该器件有管脚兼容的更高速（200MSPS）器件DAC900；A/D 转换器采用的是TI 公司的8bit 每秒20M 采样速度的器件TLC5510A，该器件具有管脚兼容的更高速（40M）器件TLC5540。运放采用的是美国模拟公司的350MHz 电压反馈运放AD8038 和AD8039，AD8038 是单路的，AD8039是双路的。以上各器件的具体参数及使用方法请读者参考光盘中提供的数据手册。

AD/DA 电路中JP1 是用于进行AD 输入前端增益设置的，以适应不同的输入信号，当短接JP1 时，增益为2；当断开JP1 时，增益为1。在AD 输入前端电路中加入了电压偏置电路(R14、R15)，偏置值为VREF/2，即2V（VREF=4V）。TLC5510A 的能测的电压范围为0~4V（VREF=4V），当JP1 断开时，对于被测模拟输入（J4）的电压幅值范围为-2V~+2V；当JP1 短接时，对于被测模拟输入（J4）的电压幅值范围为-1V~+1V。

AD/DA 电路中JP2 是用于AD 转换器的时钟源选择，当短接1、2 时AD 转换器使用板上20MHz 有源晶振提供的20M 频率；当短接2、3 时测选择核心板提供的可变频率，使用FPGA 可以产生AD 所需的任意频率。

AD 转换器的电压基准由TL431 产生， $VREF=2.5 \times ((10+3+3)/10)=4V$ ，注意电压基准电路

中R39 的阻值不能太大，要能给TL431 提供大于1mA 的电流。

AD/DA 电路中DAC 的输出采样差分方式，输出电压幅度为-2V~+2V。DAC 转换器使用内部电压基准。

注意高速AD/DA PACK 只能在核心板上的PACK 使用，不能用于主板上的PACK。此外使用AD/DA PACK 时，一定要从主板上通过连线外接一个-12V 的电源。

关于AD_DA PACK 与QuickSOPC 连接时的管脚分配请见附录。

4、红外遥控电路：

红外遥控器电路请见附件。遥控器采用Philips LPC915 单片机，电池供电，采用串口来发送数据，串口波特率为1200。

各按键所对应的码如表 1.2 所示，读者可以根据这些键值来定义功能。

红外遥控器电路中，提供了利用MiniICP 可对LPC915 进行在线编程的ICP 接口，因此用户可以编写自己的红外遥控程序并编程到LPC915。

由于红外遥控器采用电池供电，虽然采用了低功耗设计，在不使用的情况下功耗几乎为0，但为了进一步延长电池的使用寿命，建议在不使用遥控器时将电池取出进行防潮保存。

按键	键码值 (ASCII)	常用功能
S1	'U'	向上
S2	'L'	向左
短按 S3	'E'	确认
长按 S3	'S'	退出、停止
S4	'R'	向右
S5	'D'	向下

表 1.2 红外遥控器各按键对应的键值

第二章 Quartus II使用

1) 使用Quartus II建立工程

每个开发过程开始时都应建立一个Quartus II 工程，Quartus II 是以工程的方式对设计过程进行管理，QuartusII 工程中存放创建FPGA 配置文件需要的所有设置和设计文件。

(1) 打开Quartus II软件并建立工程

从【开始】>>【程序】>>【Altera】>>【Quartus II 9.0】打开Quartus II 9.0 软件，软件界面如图 2.1 所示。

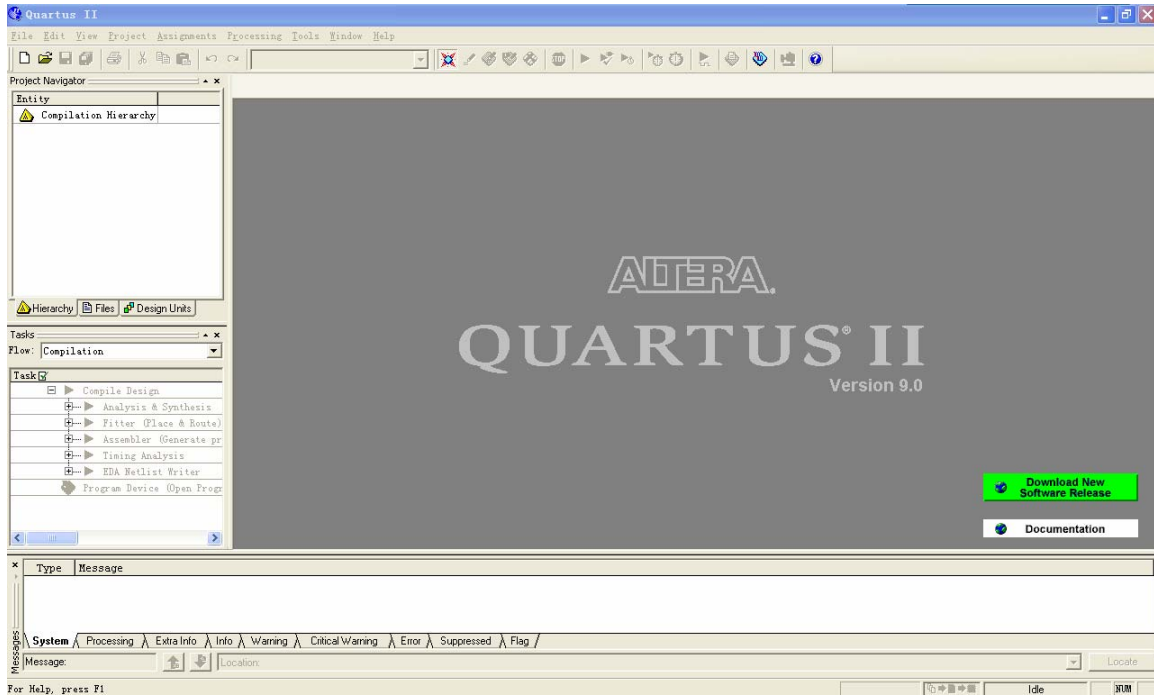


图 2.1 Quartus II软件界面

在图 2.1 中从【File】>>【New Project Wizard...】来新建一项工程，注意不要把New误认为New Project Wizard...。新建工程向导对话框如图 2.2 所示。

在如图 2.2 所示的新建工程向导说明对话框中可以了解在新建工程的过程中我们要完成哪些工作，这些工作包括：

- a. 指定项目目录、名称和顶层实体。
- b. 指定项目设计文件。
- c. 指定该设计的Altera 器件系列。
- d. 指定用于该项目的其它EDA 工具。
- e. 项目信息报告。

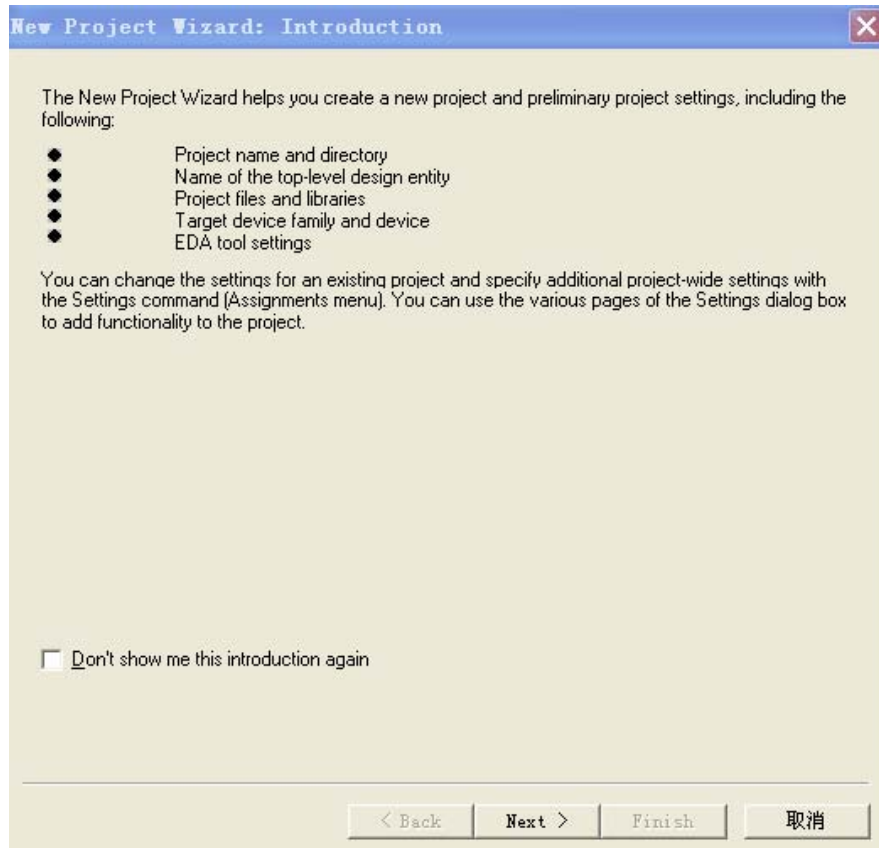


图 2.2 新建工程向导说明对话框

在图 2.2 中点击 NEXT 进入图 2.3 所示的新建工程路径、名称、顶层实体指定对话框。

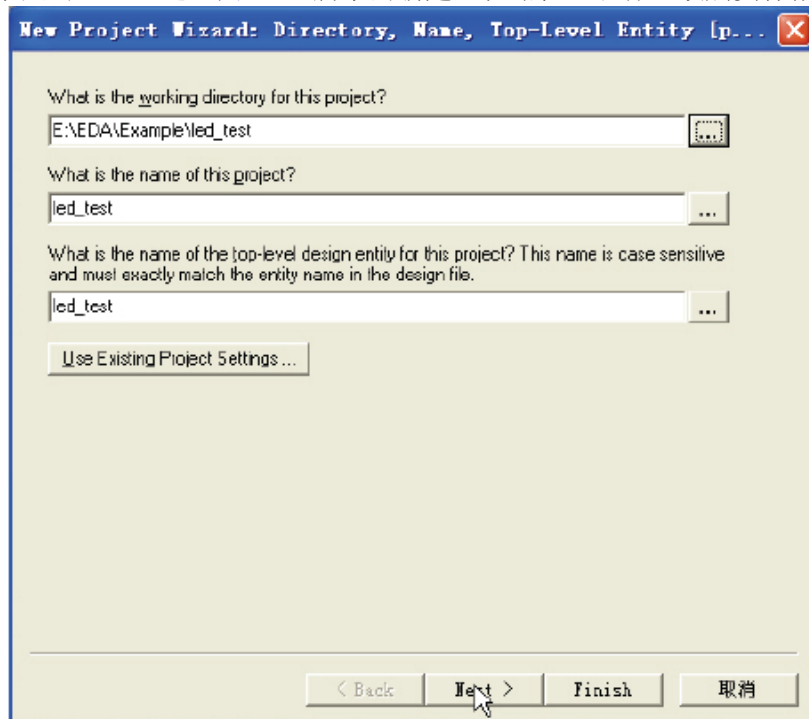


图 2.3 新建工程路径、名称、顶层实体指定对话框

任何一项设计都是一项工程(project)，必须首先为此工程建立一个放置与此工程相关的所有文件的文件夹，此文件夹将被QuartusII 默认为工作库(Work Library)。一般，不同的设计项目最好放在不同的文件夹中，而同一工程的所有文件都必须放在同一文件夹中。

不要将文件夹设在计算机已有的安装目录中，更不要将工程文件直接放在安装目录中。文件夹所在路径名和文件夹名中不能用中文，不能用空格，不能用括号（），可用下划线_，最好也不要以数字开头。

图 2.3 第一栏用于指定工程所在的工作库文件夹；第二栏用于指定工程名，工程名可以取任何名字，也可以直接用顶层文件的实体名作为工程名（建议使用）；第三栏用于指定顶层文件的实体名。例工程的路径为E:\EDA\Example\led_test 文件夹，工程名与顶层文件的实体名同名为led_test。

接着单击 NEXT 进入图 2.4 所示的添加文件对话框。

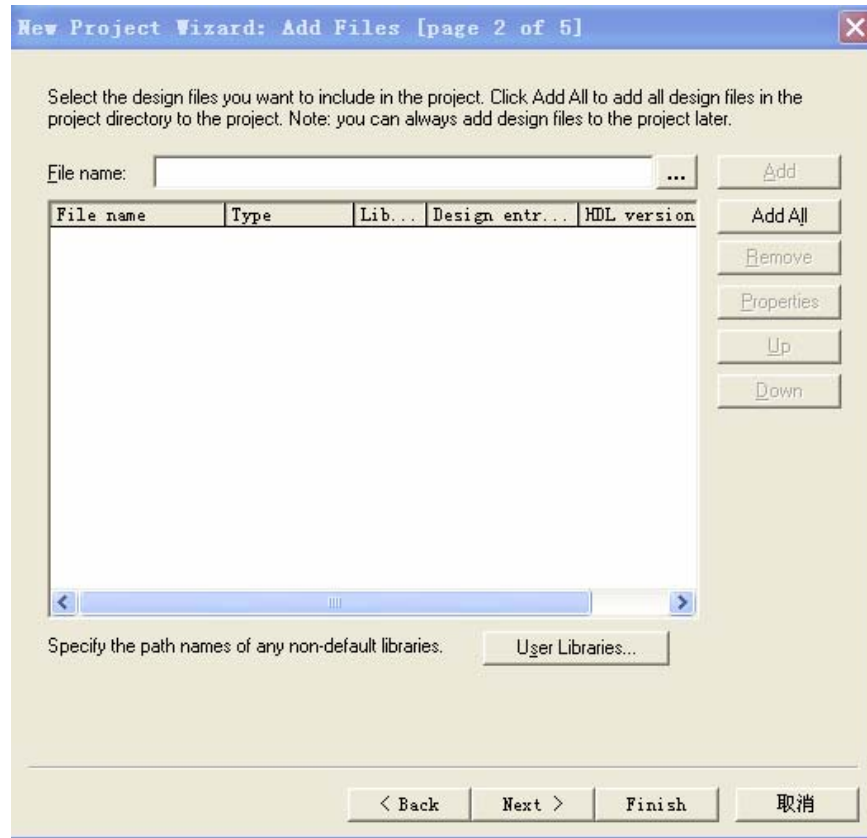


图 2.4 新建工程添加文件对话框

由于是新建工程，暂无输入文件，所以直接单击，进入图 2.5 所示的指定目标器件对话框。这里我们选择的是QuickSOPC 核心板上用的CycloneIII系列的EP3C25F324C8。

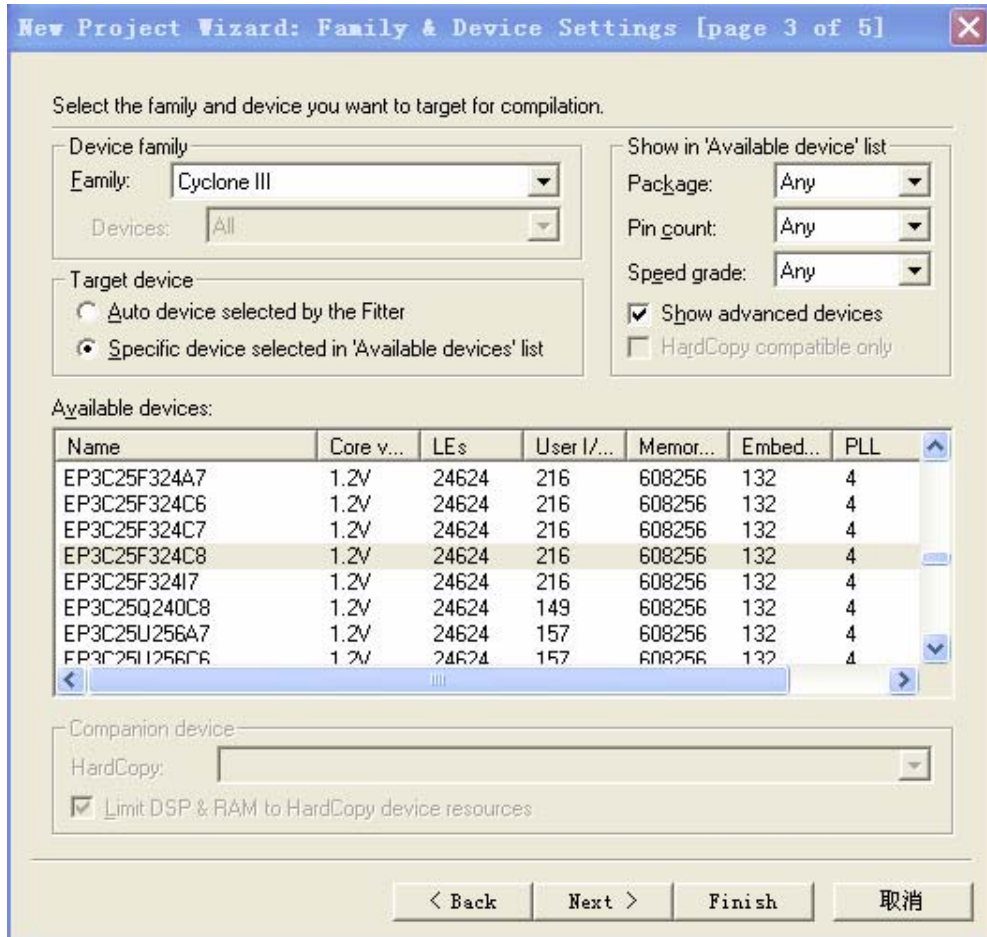


图 2.5 新建工程器件选择对话框

在图 2.5 右边的过滤器栏 (Filters) 中, 设计者可以通过指定封装、管脚数以及器件速度等级来加快器件查找的速度。

指定完器件后, 单击 NEXT 进入图 2.6 所示的指定EDA 工具对话框。

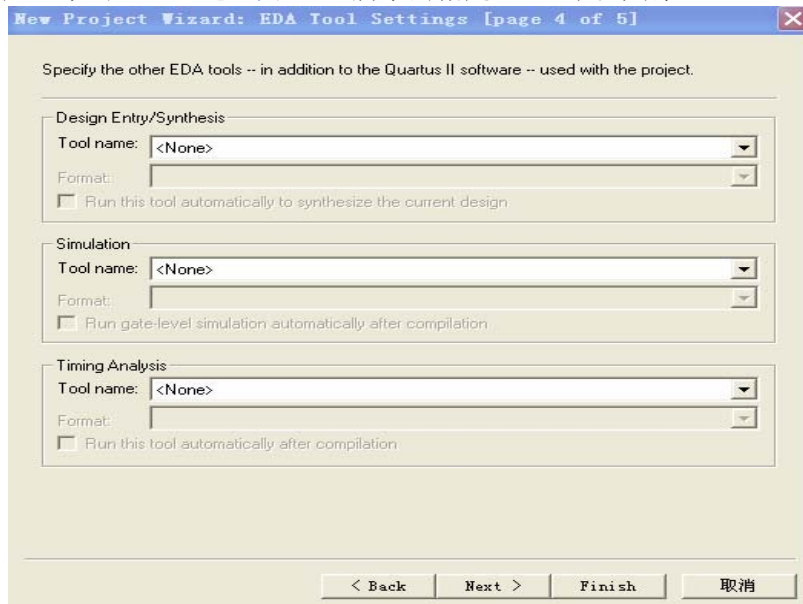


图 2.6 新建工程EDA 工具设置对话框

如果实验时仅利用QuartusII 的集成环境进行开发，不使用任何EDA 工具，因此这里不作任何改动。图 2.6 中单击 NEXT 进入图 2.7 所示的工程信息报告对话框。从工程信息报告对话框，设计者可以看到工程文件配置信息报告。点击，完成新建工程的建立。

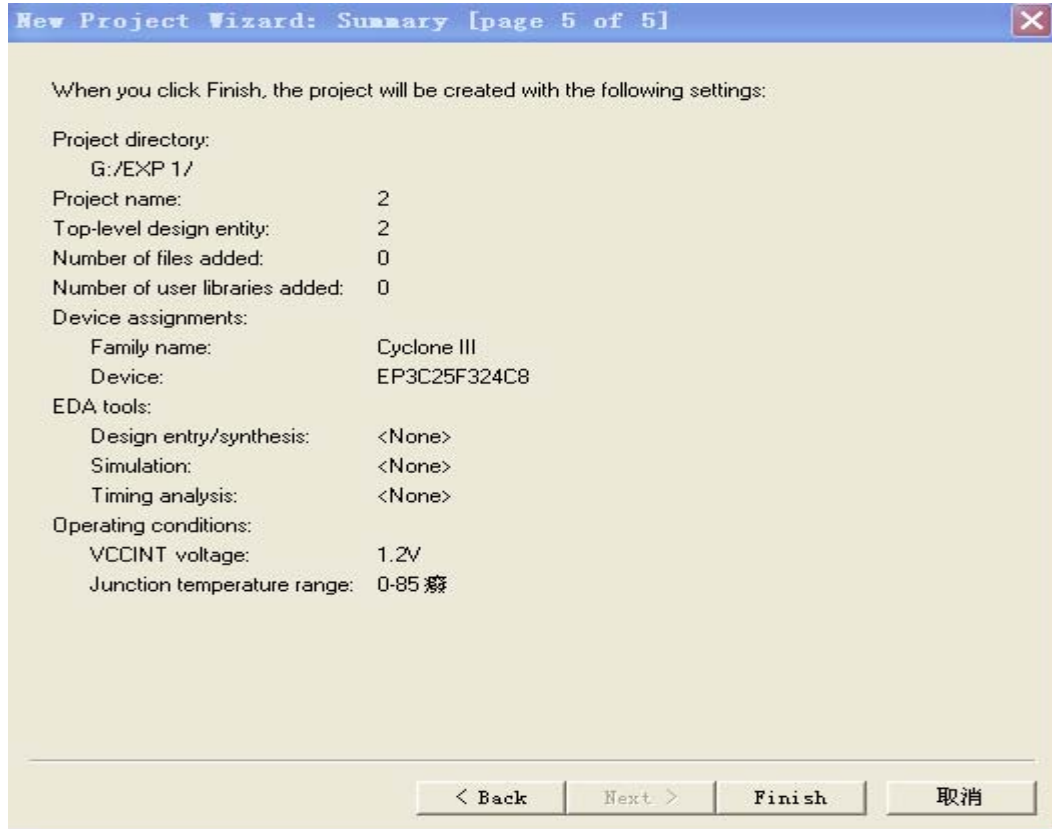



图 2.7 新建工程配置信息报告对话框

需要注意的是，建立工程后，还可以根据设计中的实际情况对工程进行重新设置，可选择【Assignments】>>【Settings...】进行设置，也可以选择工具栏上的按钮。

(2) 建立图形设计文件

Quartus II 图形编辑器也称为块编辑器(Block Editor)，用于以原理图(Schematics)和结构图(Block Diagrams)的形式输入和编辑图形设计设计信息。Quartus II 的块编辑器可以读取并编辑结构图设计文件(Block Design Files)和MAX+PLUS II 图形设计文件(Graphic DesignFiles)。可以在Quartus II 软件中打开图形设计文件并将其另存为结构图设计文件。

如图 2.8 所示，从【File】>>【New...】打开新建文件对话框如图 2.9 所示。

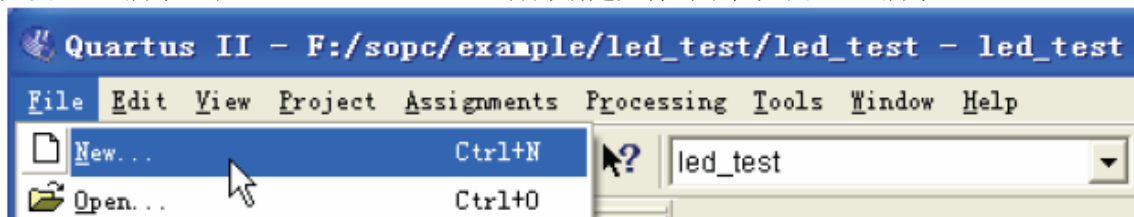


图 2.8 新建文件

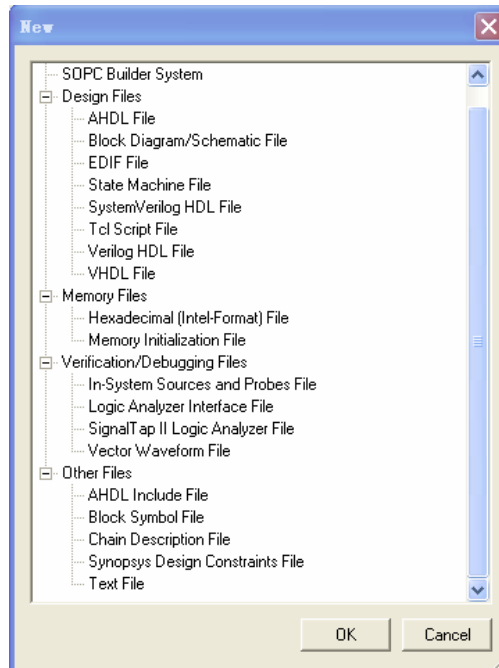


图 2.9 新建文件对话框

在如图 2.9 所示的新建文件对话框中选择 **Block Diagram /Schematic File**，按 **OK** 建立一个空的图形设计文件，缺省名为 **Block1.bdf**。如图 2.10 所示，点击 **【File】>>【Save As...】** 打开将 BDF 文件存盘的对话框，如图 2.11 所示。

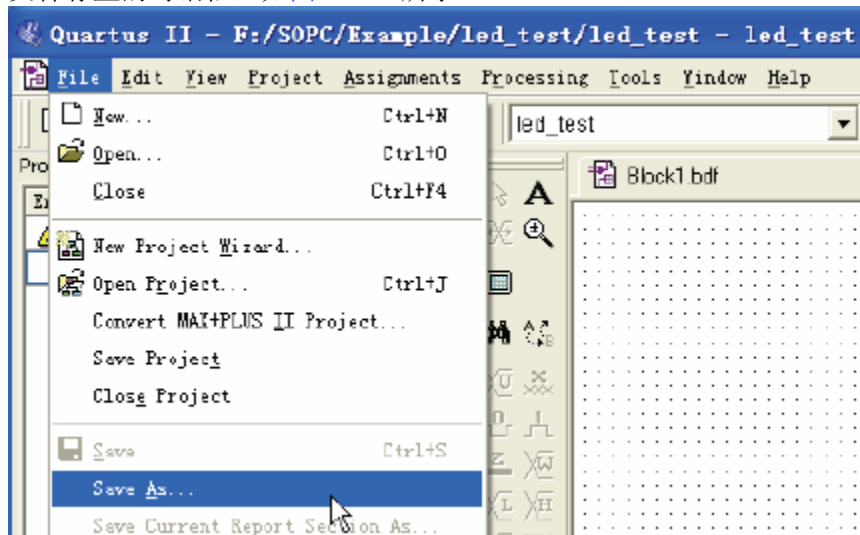


图 2.10 将 BDF 文件存盘

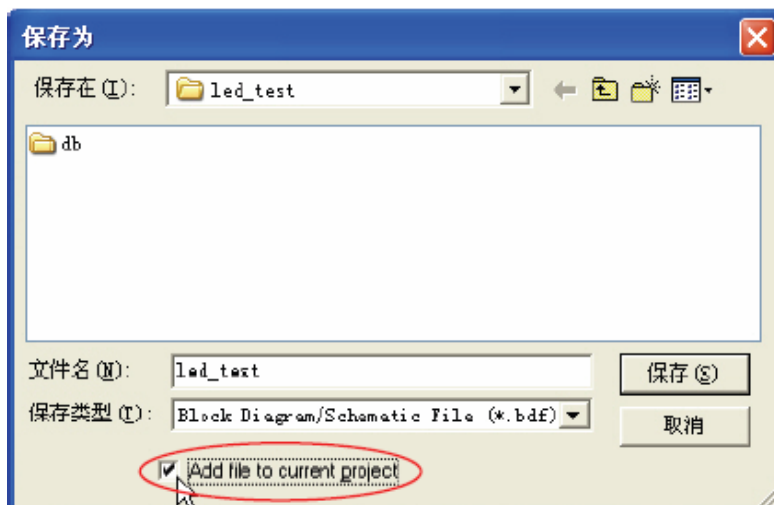


图 2.11 BDF 文件存盘对话框

在BDF 文件存盘对话框中接受默认的文件名，并默认Add file to current project 选项选中，以使该文件添加到工程中去。至此，完成了顶层模块的建立。在下面的步骤中将会将VHDL 文件生成的模块加入到顶层模块中去。

(3) 建立文本编辑文件

Quartus II 的文本编辑器是一个非常灵活的编辑工具，用于以AHDL (Altera HardwareDescription Language)、VHDL 和Verilog HDL 语言形式以及Tcl 脚本语言输入文本型设计，还可以在文本编辑器下输入、编辑和查看其他ASCII 文本文件。在这里我们要建立的是VHDL 文件，建立的流程和建立图形设计文件一样。

在创建好一个设计工程以后，如图 2.8 所示，从【File】>>【New...】打开新建文件对话框如图 2.9 所示，在新建文件对话框中选择VHDL File（若要新建Verilog HDL 文件，则选择Verilog HDL File），按 OK 建立一个空的VHDL 文件，缺省名为Vhd11.vhd (Verilog HDL 文件为Verilog1.v)。如图 2.10 所示，点击【File】>>【Save As...】如改名为LED.vhd文件并保存。

2) Quartus II 工程设计


在第1)步中已经建立好了Quartus II 工程文件，现在要对Quartus II 工程进行编程设计。

(1) 在VHDL文件中编写源程序

在新建VHDL 源程序文件输入程序代码并保存。

对该VHDL文件进行编译处理，具体操作如下：

a. 如图 2.12 所示，在Project Navigator窗口的Files标签中的led.vhd文件单击鼠标右键，在弹出的菜单中点击Set at Top-level Entity选项。将led.vhd设置为顶层实体。

b. 选择【Processing】>>【Start】>>【Start Analysis & Synthesis】进行综合编译，也可以选择工具栏上的按钮启动编译。

c. 若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。

(2) 从设计文件创建模块

在层次化工程设计中，经常需要将已经设计好的文件生成一个模块符号文件(BlockSymbol Files.bsf)作为自己的功能模块符号在顶层调用，该符号就像图形设计文件中的任何其他宏功能符号一样可被高层设计重复调用。

如图 2.12 所示，在Project Navigator窗口的Files标签中的led.vhd文件单击鼠标右键，在弹出的菜单中点击Create Symbol Files for Current File选项。之后会弹出一对话框提示

原理图文件创建成功，点 确定 按钮即可创建一个代表行文件功能的符号文件(led.bsff)。

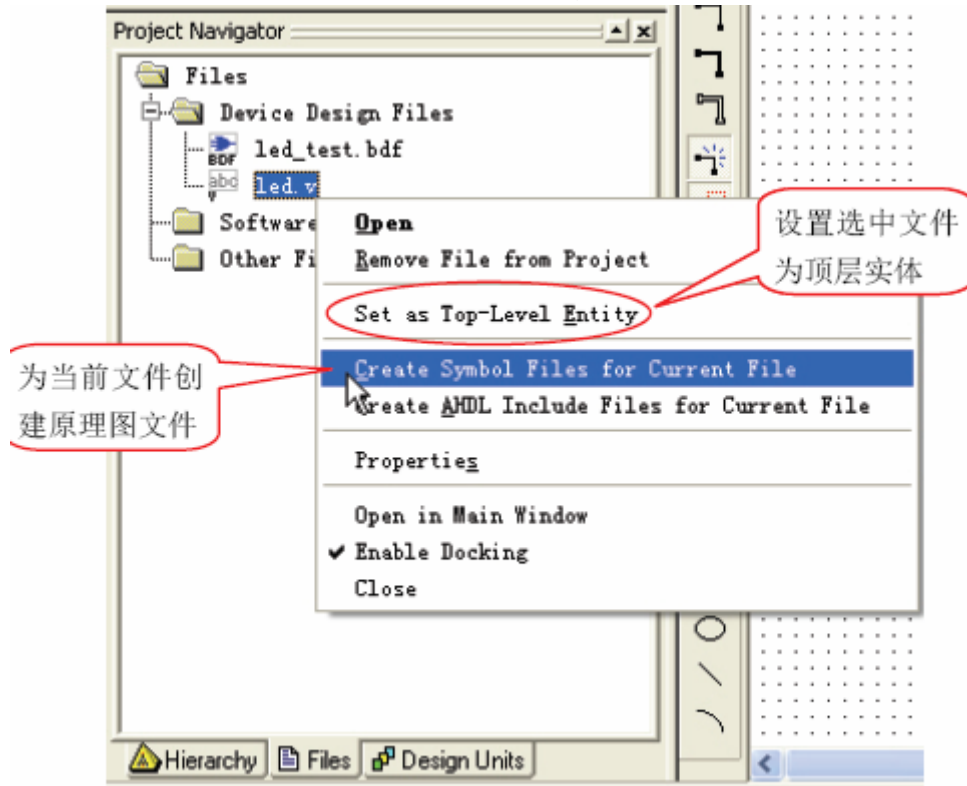


图 2.12 生成原理图文件

设计好的图形设计文件(Block Symbol Files .bdf)也可以生成一个模块符号文件(BlockSymbol Files .bsff)作为自己的功能模块符号在顶层调用。操作如下：

a. 如图2.12所示，在Project Navigator窗口的Files标签中的.bdf文件单击鼠标右键，在弹出的菜单中点击Set at Top-level Entity选项。将.bdf设置为顶层实体。

b. 在【File】>>【Create/Update】项选择Create Symbol Files for Current file，点击 确定 按钮即可创建一个代表行文件功能的符号文件(.bsff)。

(3) 添加led.bsff模块到Quartus II顶层模块

执行下列步骤将led符号(led.bsff)加入到BDF文件中（在第1)步中已经建立好的led_test.bdf）。

a. 在Quartus II中，在led_test.bdf窗口的任意处双击，弹出添加符号（Symbol）的对话框，如图 2.13 所示。

b. 在Project 下，选择led，在右边的窗口中出现一个大的符号，这就是由led.vhd 源文件生成模块符号。

c. 单击 OK 。Symbol 对话框被关闭，led 符号被附在鼠标的指针上，在led_test.bdf的窗口中的适当位置，放置该符号。

d. 选择【File】>>【Save】来保存Quartus II 顶层文件led_test.bdf。

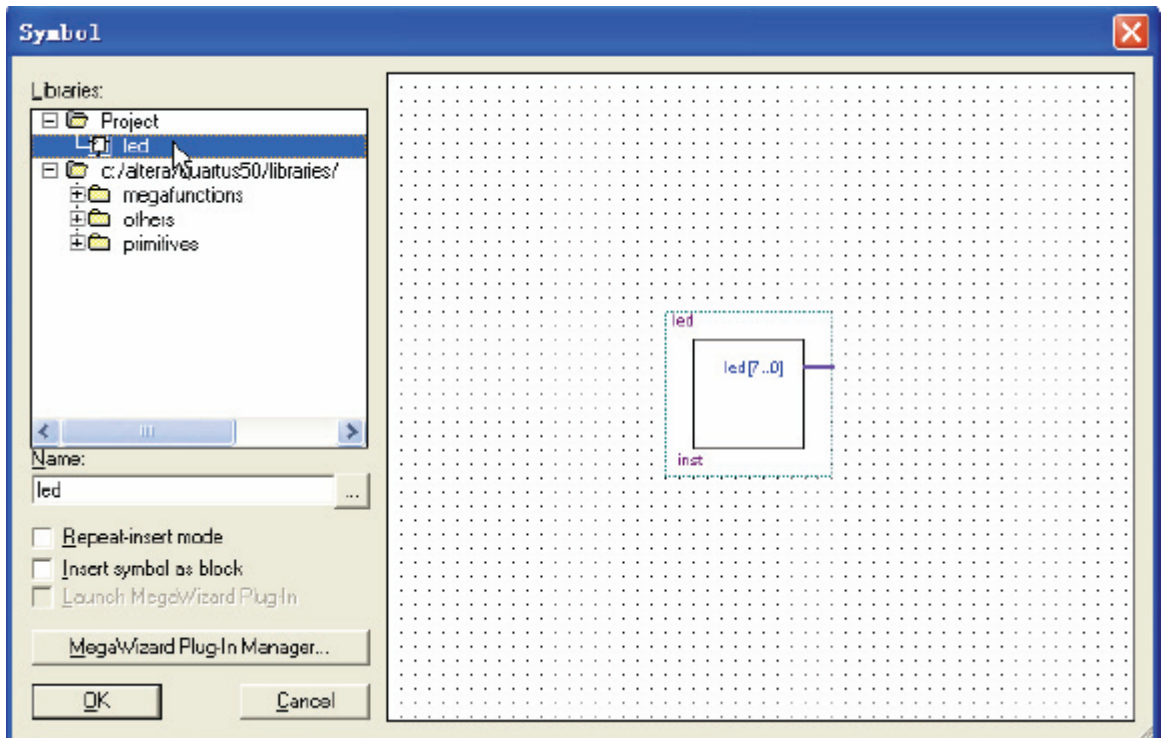


图 2.13 添加Symbol 对话框



(4) 添加引脚和其它基本单元

引脚包括输入(input)、输出(output)和双向(bidir)三种类型，在添加引脚和其它基本单元时，将使用到模块编辑工具栏中的各种工具。图 2.14 显示了工具栏中各工具的功能。



图 2.14 模块编辑工具栏

执行下面步骤为顶层模块添加输出管脚：

- a. 单击模块编辑工具栏中的  按钮，插入符号对话框如图 2.15 所示。读者可以发现，这个对话框就是图 2.13 所示的对话框，不同之处，使用该按钮时，默认重复插入模式 (Repeatinsert mode) 选中，这样可重复的插入符号。当然，前面插入led 符号也可使用  按钮。

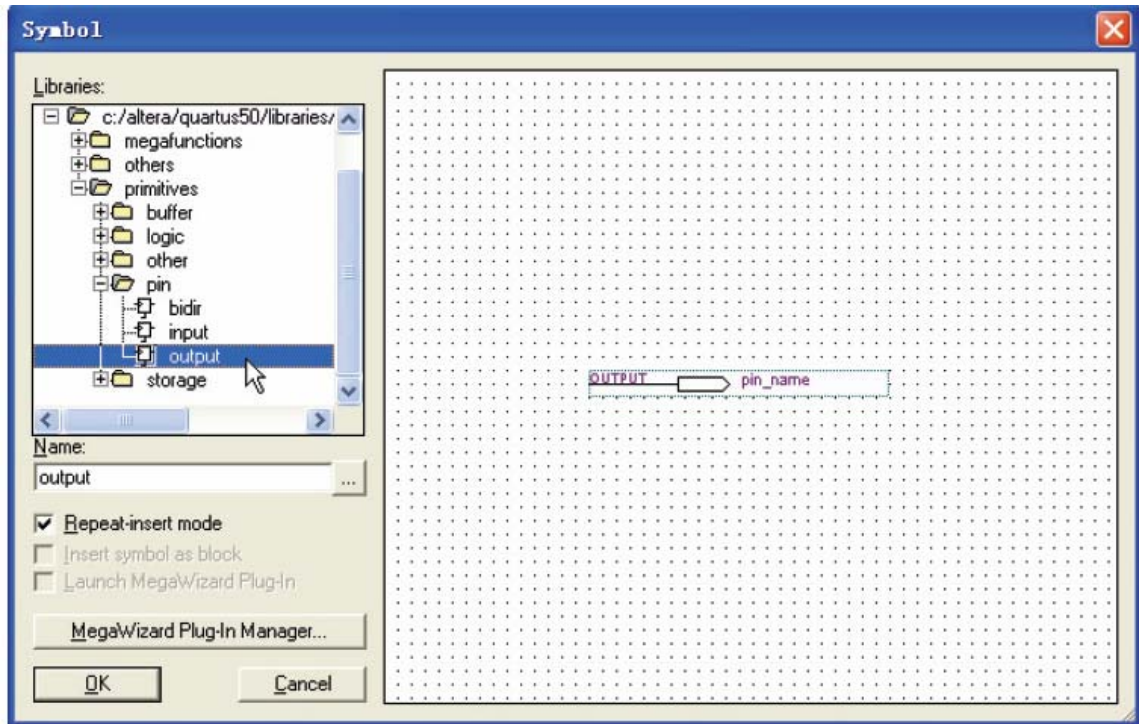


图 2.15 添加输入管脚对话框

- b. 单击“+”展开C:/altera/quartus50/libraries 文件夹，再单击“+”展开primitives 文件夹和pin 文件夹。
- c. 在pin 文件夹下选择output 组件（也可以在Name 栏目下直接输入名称）。
- d. 单击 OK 按钮。
- e. 在led 符号的左侧的空白处单击1 次，插入1 个output 引脚符号。
- f. 拖动管脚符号连接到led 符号的输出口。
- g. 双击各管脚符号，进行管脚命名。
- h. 选择【File】>>【Save】保存BDF 文件。

(5) 选择器件型号

每种型号的FPGA 芯片的管脚可能都不相同，因此在进行管脚分配之前都应选择相应目标FPGA 芯片型号。这一步其实在第1)步中已经完成，这里只是让读者了解，在QuartusII 工程创建好后仍然可以选择并修改器件型号。

如图 2.16 所示，选择【Assignments】>>【Device...】打开器件选择对话框如图 2.17 所示。在对话框中，指定所需的目標器件型号，本实验为EP3C25F324C8。

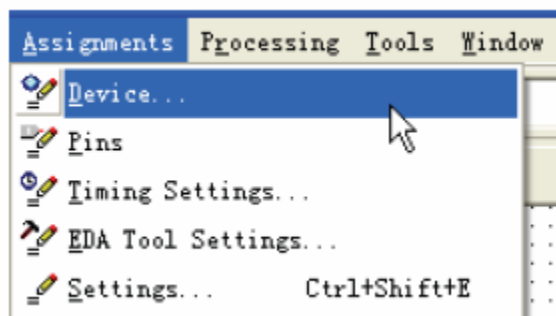


图 2.16 打开器件选择对话框

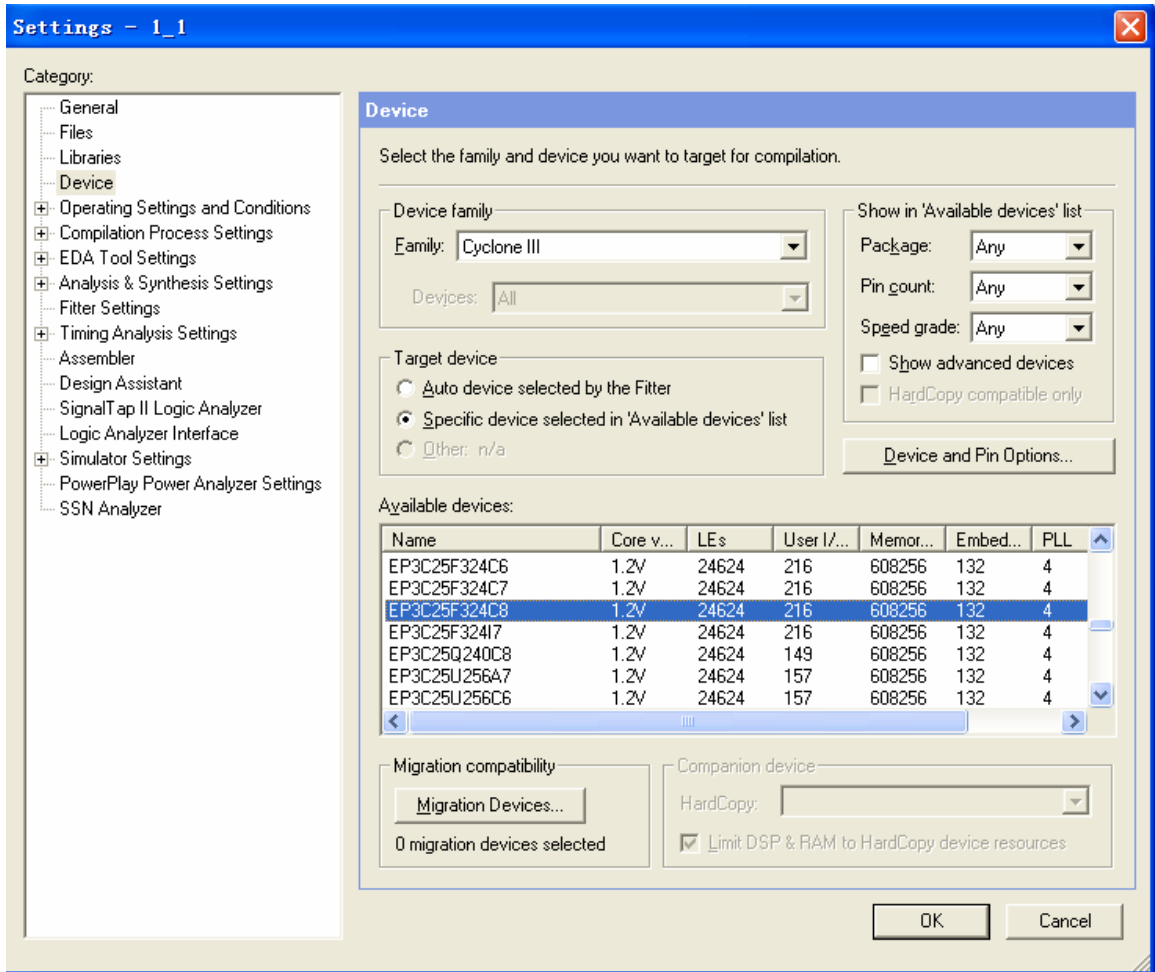


图 2.17 器件选择对话框

(6) 分配FPGA管脚

要执行FPGA 管脚的分配，可按下面步骤进行：

- 如图 2.16 所示，选择【Assignments】>>【Pins】打开管脚分配对话框如图 2.18 所示。
- 按照要求在To 栏中，输入各管脚名称，在Location 下拉选择相应的管脚，也可以在Location 下输入管脚号(如pin_50)来快速定位，最终分配的结果如图 2.18 所示。
- 选择【File】>>【Save】来保存分配，然后关闭Assignment Editor。

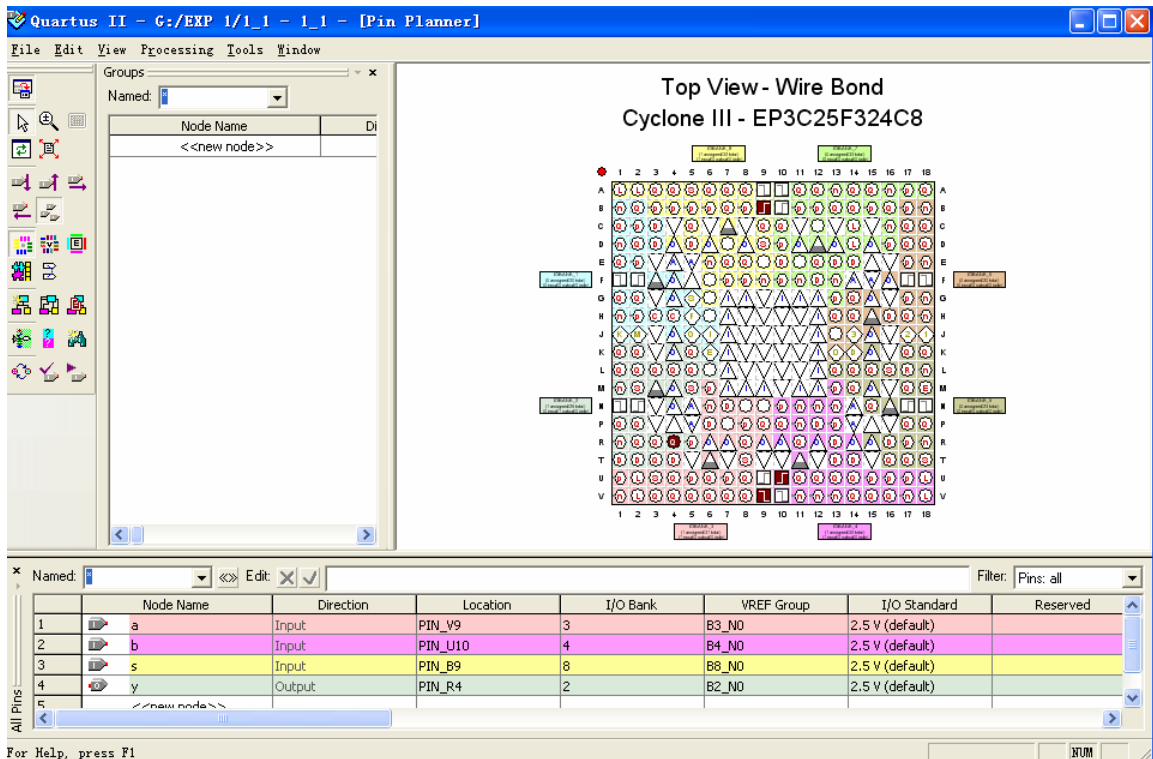


图 2.18 分配管脚对话框

(7) 器件和管脚的其他设置

点击图 2.17 中的 **Device and Pin Options...**，打开 Device & Pin Options 对话框。在 Device & Pin Options 对话框中选择 **Unused Pins** 标签页进行没有使用管脚的设置，按照图 2.19 所示设置将未使用管脚设置为高阻输入，这样上电后 FPGA 的所有不使用管脚后将进入高阻抗状态。

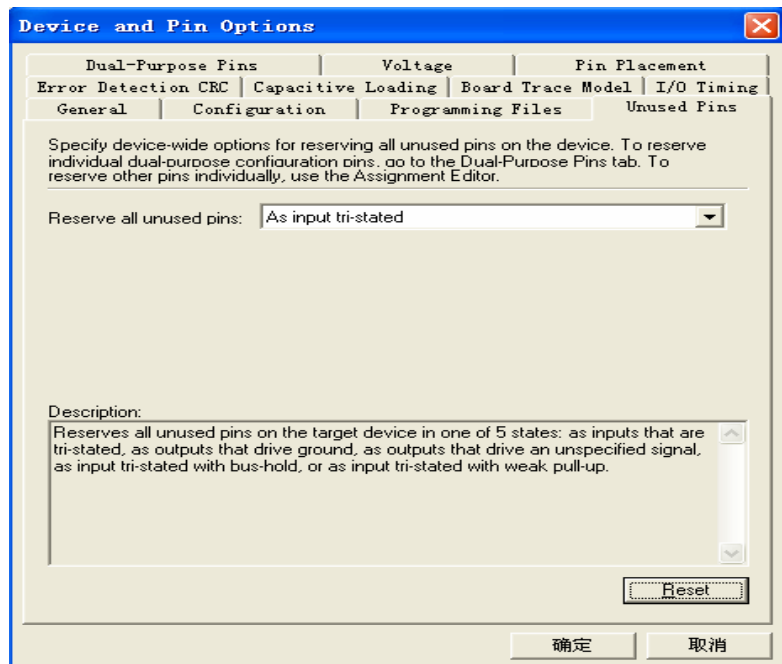


图 2.19 未用管脚设置

在设计中一定要将未定义的管脚定义为三态输入，如图 2.19 所示。注意**一定不能**将未定义（不使用）管脚模式设置为输出，并连接到地（As outputs, driving ground）。否则，可能会造成连接在核心板上的Flash、sram 等未使用的芯片冲突而损坏芯片。

3) 设置编译选项并编译硬件系统

(1) 设置编译选项

在图 2.17 中的左边框中，选择相应的设置选项进行设置，一般实验采用默认的编译设置，不进行任何修改，若要进行编译选项设置，请参考Altera 的Quartus II 使用手册。

(2) 编译硬件系统

在编译过程中，编译器定位并处理所有工程文件，生成与编译相关的消息与报告，创建SOF 文件及任何可选配置文件。

如图 2.12 所示，在Project Navigator窗口的Files标签中的led_test.bdf文件单击鼠标右键，在弹出的菜单中点击Set at Top-level Entity选项。将led_test.bdf设置为顶层实体。

选择【Processing】>>【Start Compilation】进行全程编译，也可以选择工具栏上的按钮启动编译，对该工程文件进行编译处理，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。在编译硬件系统时，状态窗口显示整个编译进程及每个编译阶段所用的时间。编译结果显示在Compilation Report 窗口中。整个编译时间大约几十秒到一分钟，这取决于计算机的性能以及编译选项设置。

在编译过程中，可能产生很多警告信息，但这些不会影响设计结果。

(3) 查看编译报告

编译结束后，对话框显示消息“Full compilation was successful.”，单击进入Compilation Report 窗口，如图 2.20 所示，包括编译报告、综合报告、适配报告、时序分析报告等。

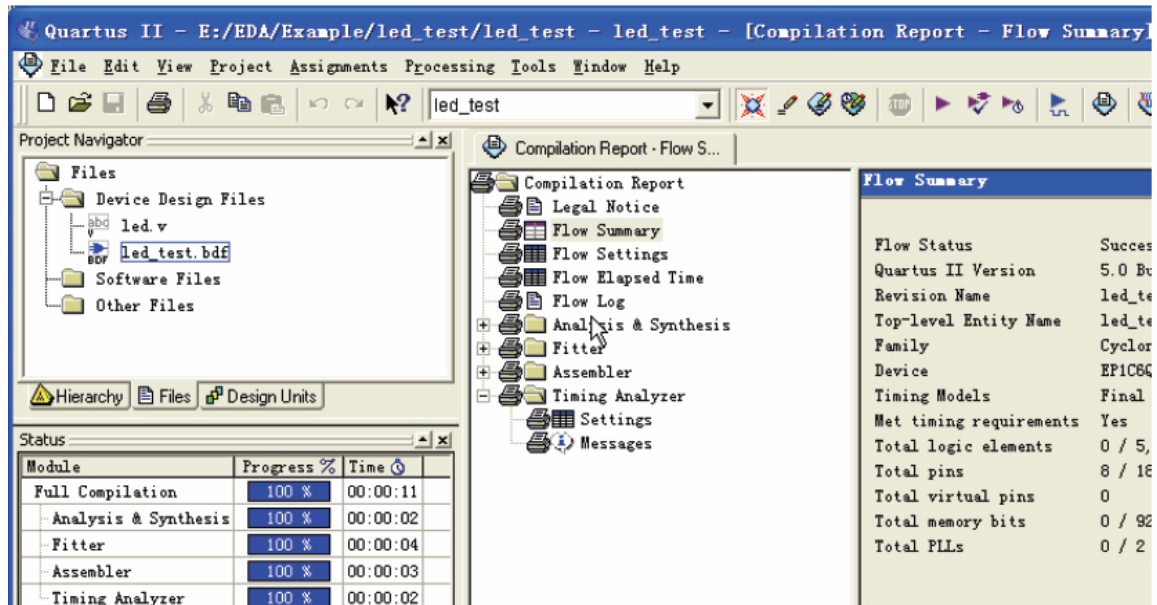



图 2.20 编译报告窗口

4) 下载硬件设计到目标FPGA

成功编译硬件系统后,将产生led_test.sof 的FPGA 配置文件输出。本步骤简单介绍将SOF 文件下载到到目标FPGA 器件的步骤。

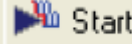
a. 按要求连接实验电路。

b. 通过ByteBlaster II 下载电缆连接实验箱JTAG 口和主计算机, **JTAG下载线接头有块突起,突起的方向朝向FPGA芯片插入连接**,然后接通实验箱电源。

c. 在QuartusII 软件中选择【Tools】>>【Programmer】,也可以按工具栏上的按钮。打开编程器窗口并自动打开配置文件(led_test.sof),如图 2.21 所示。如果没有自动打开配置文件,则需要自己添加需要编程的配置文件。

d. 确保编程器窗口左上角的Hardware Setup 栏中硬件已经安装。

e. 确保Program/Configure 下的方框选中。

f. 单击  Start

开始使用配置文件对FPGA 进行配置,Progress 栏显示配置进度。

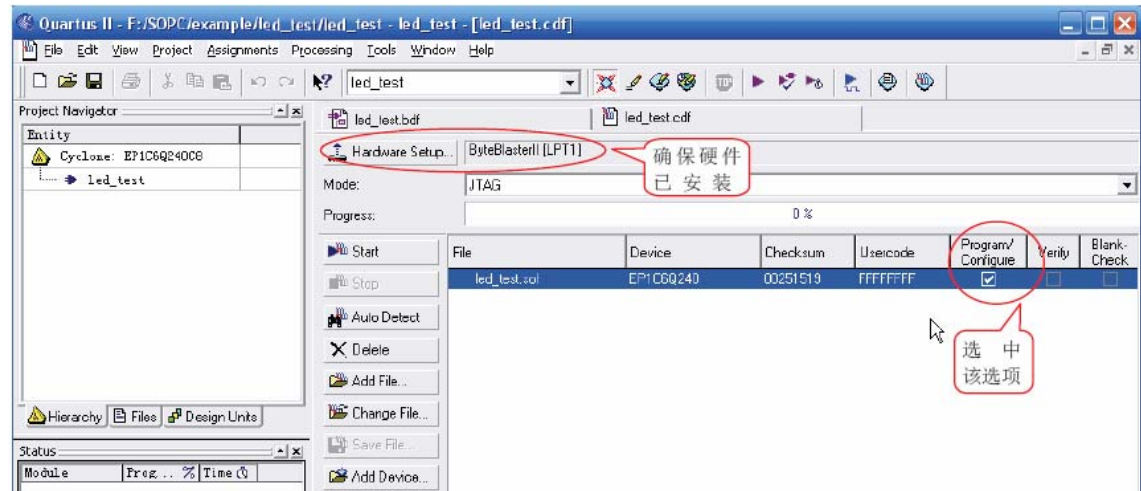


图 2.21 编程窗口

如果使用QuickSOPC 核心板,板上的配置绿色指示灯CONF 亮,说明配置成功。本例只讲述了将配置文件下载到FPGA 中,掉电后FPGA 中的配置数据将丢失。也可以将配置文件写入掉电保持的EPCS (或者配置Flash芯片),在上电时使用EPCS 对FPGA 进行配置。

Quartus II 工程的顶层文件的扩展名可以是: .bdf、.tdf、.vhd、.vhdl、.v、.vlg、.edif 或.edf。在本例中顶层文件为led_test.bdf。注:本例可以设led.vhd 为顶层文件并直接分配管脚编译下载,而无需新建led_test.bdf 和led.bsfc 文件。但为了让用户了解图形设计文件(.bdf)及图形符号文件(.bsfc)的新建过程,方便以后的设计应用。所以增设新建led_test.bdf 和led.bsfc 文件这些内容。

5) 波形仿真

(1) 选择【File】>>【New...】命令,打开新建文件对话框,在新建对话框中选择标签页,从中选择Vector Waveform File,如图 2.22 所示。按 OK 建立一个空的波形编辑器窗口,缺省名为Waveform1.vwf。点击【File】>>【Save As...】改名为full_add.vwf并保存。

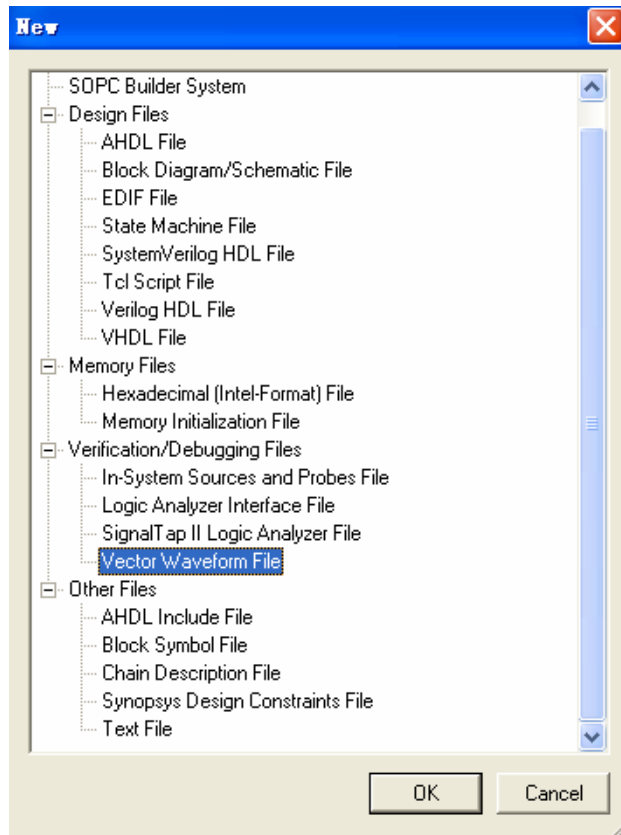


图 2.22 新建波形文件对话框

(2) 在如图2.23所示的Name 标签区域内双击鼠标左键，弹出如图 2.24 所示的添加节点对话框。

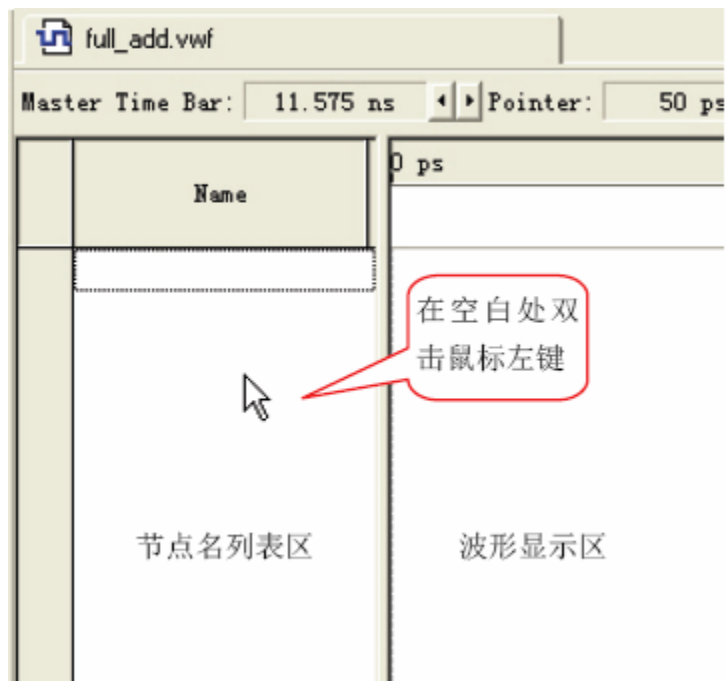


图2.23 新建波形文件界面

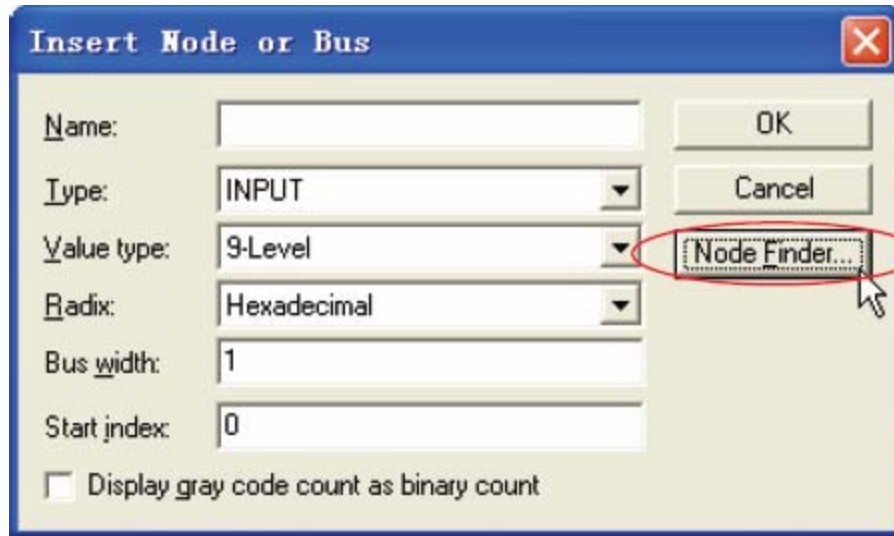


图 2.24 添加节点对话框

在如图 2.24 所示的添加节点对话框中按 **Node Finder...** 按钮，弹出如图 2.25 所示的对话框，按图 2.25 所示步骤进行选择 and 设置，按按钮后又弹出如图 2.24 所示的对话框，再按 OK 按钮完成节点添加，如图 2.26 所示。

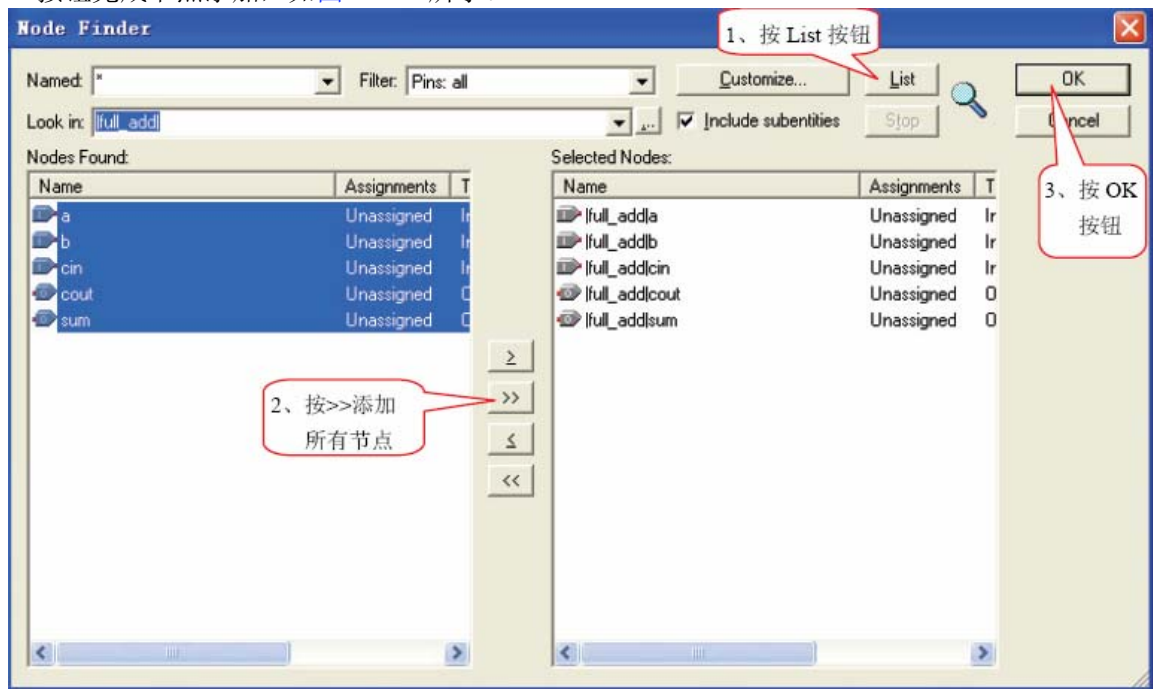


图 2.25 添加节点

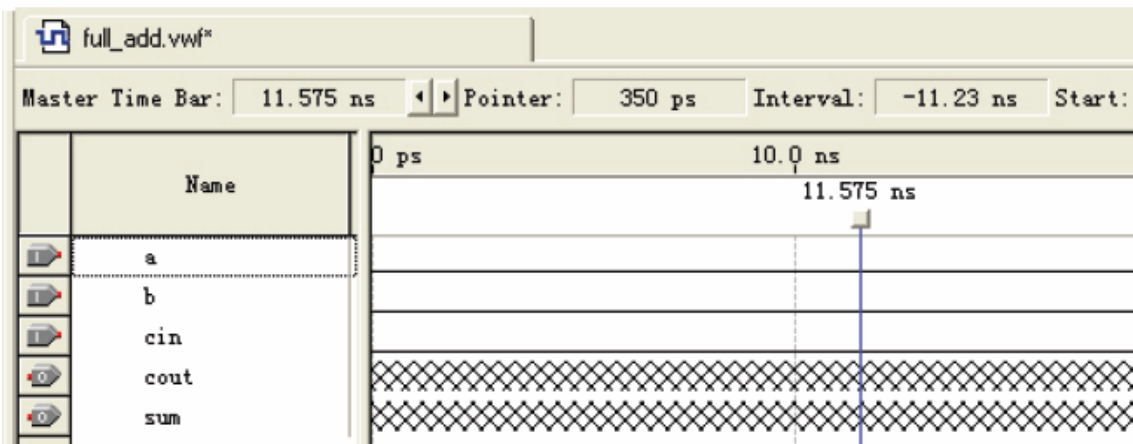


图 2.26 添加完节点的波形图

(3) 波形编辑器默认的仿真结束时间为 $1\mu\text{S}$ ，根据仿真需要，可以自由设置仿真文件的结束时间。选择【Edit】>>【End Time】命令，弹出结束时间对话框，在Time 框内输入仿真结束时间，时间单位可选为S、mS(10⁻³S)、 μS (10⁻⁶S)、nS(10⁻⁹S)、pS(10⁻¹²S)。点击 OK 按钮完成设置。在这里采用默认设置($1\mu\text{S}$)。

(4) 编辑输入节点波形。编辑时将使用到波形编辑工具栏中的各种工具。图 2.27 显示了工具栏中各工具的功能。



图 2.27 波形编辑器工具条

(5) 选择【Tools】>>【Simulator Tool】命令，弹出如图 2.28 所示的对话框。

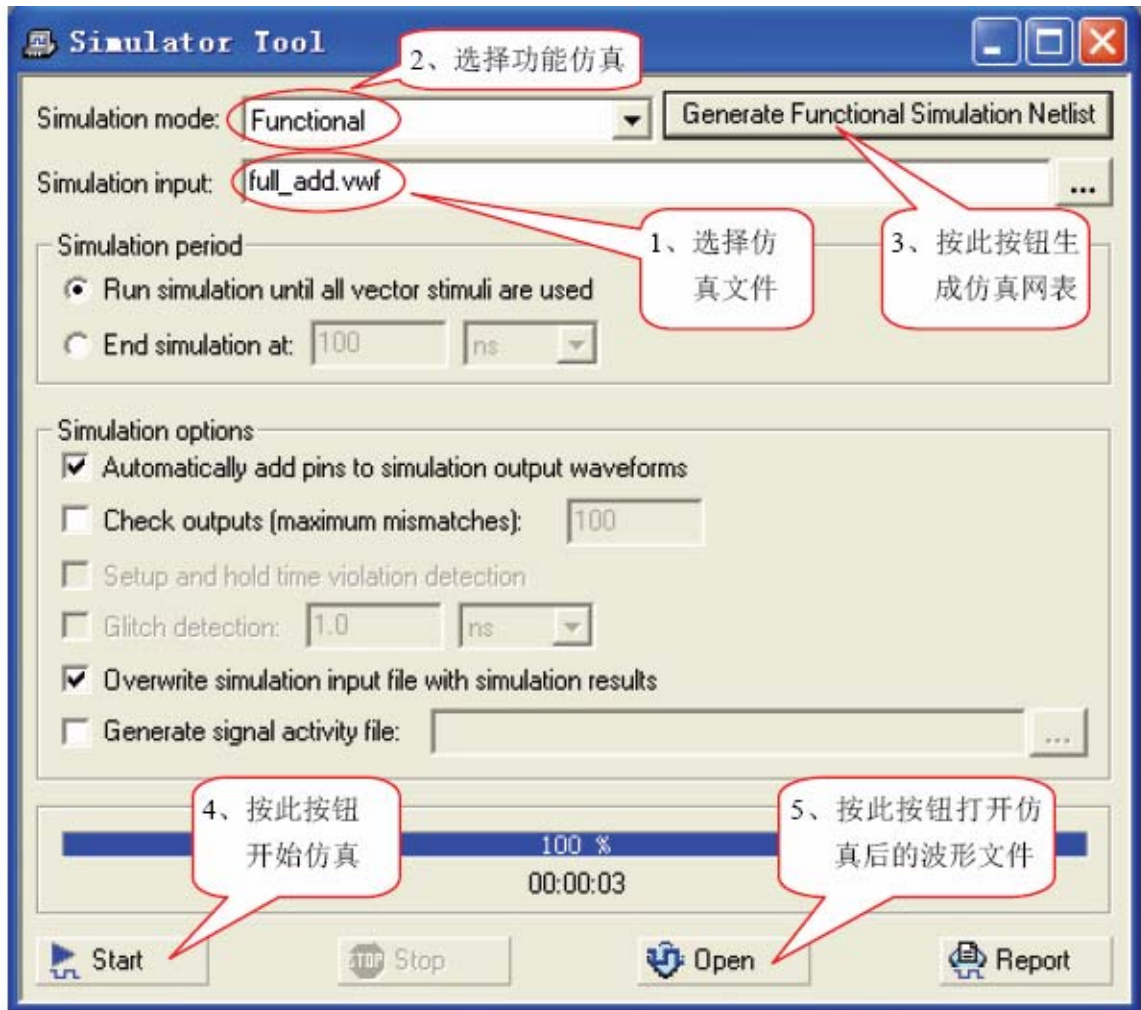


图 2.28 仿真设置对话框

(6) 验证仿真结果是否与设计相符合，如果不符合，需重新设计文件，再进行综合编译、仿真。直到仿真结果与设计相符合为止。

附 录

核心板IO分配

序号	引脚标号	FPGA 管脚	IO 口方向	功能	备注
1	SDRAM_A[0]	U6	O	SDRAM 地址线	
2	SDRAM_A[1]	V5			
3	SDRAM_A[2]	U5			
4	SDRAM_A[3]	V4			
5	SDRAM_A[4]	U1			
6	SDRAM_A[5]	V1			
7	SDRAM_A[6]	V2			
8	SDRAM_A[7]	U3			
9	SDRAM_A[8]	V3			
10	SDRAM_A[9]	U4			
11	SDRAM_A[10]	V6			
12	SDRAM_A[11]	T4			
13	SDRAM_A[12]	T6			
14	SDRAM_DQ[0]	U16	IO	SDRAM 数据线	
15	SDRAM_DQ[1]	V15			
16	SDRAM_DQ[2]	U15			
17	SDRAM_DQ[3]	V14			
18	SDRAM_DQ[4]	U14			
19	SDRAM_DQ[5]	V13			
20	SDRAM_DQ[6]	U13			
21	SDRAM_DQ[7]	V12			
22	SDRAM_DQ[8]	T11			
23	SDRAM_DQ[9]	T13			
24	SDRAM_DQ[10]	T14			
25	SDRAM_DQ[11]	V16			
26	SDRAM_DQ[12]	U17			
27	SDRAM_DQ[13]	V17			
28	SDRAM_DQ[14]	U18			
29	SDRAM_DQ[15]	V18			
30	SDRAM_BA[0]	V7	O	BA0、1 决定哪个 BANK 有效	
31	SDRAM_BA[1]	U7	O		
32	SDRAM_DQM[0]	U12	O	输入/输出屏蔽 Masked	
33	SDRAM_DQM[1]	T8	O		
34	SDRAM_NRAS	V8	O	命令的读入	
35	SDRAM_NCAS	U11	O		
36	SDRAM_NWE	V11	O		
37	SDRAM_CLK	U2	O	SDRAM 时钟	
38	SDRAM_CKE	R8	O	时钟使能	
39	SDRAM_NCS	U8	O	SDRAM 片选	
40	FLASH_SRAM_D[0]	H3	IO	Nor Flash 以及外扩的数据总线	
41	FLASH_SRAM_D[1]	D1			
42	FLASH_SRAM_D[2]	A8			
43	FLASH_SRAM_D[3]	B8			
44	FLASH_SRAM_D[4]	B7			
45	FLASH_SRAM_D[5]	C5			

序号	引脚标号	FPGA 管脚	IO 口方向	功能	备注
46	FLASH_SRAM_D[6]	E8	IO	Nor Flash 以及外扩的数据总线	
47	FLASH_SRAM_D[7]	A4			
48	FLASH_SRAM_D[8]	B4			
49	FLASH_SRAM_D[9]	E7			
50	FLASH_SRAM_D[10]	A3			
51	FLASH_SRAM_D[11]	B3			
52	FLASH_SRAM_D[12]	D5			
53	FLASH_SRAM_D[13]	B5			
54	FLASH_SRAM_D[14]	A5			
55	FLASH_SRAM_D[15]	B6			
56	A0	L15	O	Nor Flash 以及外扩的地址总线	
57	FLASH_SRAM_A[1]	E12			
58	FLASH_SRAM_A[2]	A16			
59	FLASH_SRAM_A[3]	B16			
60	FLASH_SRAM_A[4]	A15			
61	FLASH_SRAM_A[5]	B15			
62	FLASH_SRAM_A[6]	A14			
63	FLASH_SRAM_A[7]	B14			
64	FLASH_SRAM_A[8]	A13			
65	FLASH_SRAM_A[9]	B13			
66	FLASH_SRAM_A[10]	A12			
67	FLASH_SRAM_A[11]	B12			
68	FLASH_SRAM_A[12]	A11			
69	FLASH_SRAM_A[13]	B11			
70	FLASH_SRAM_A[14]	C10			
71	FLASH_SRAM_A[15]	D10			
72	FLASH_SRAM_A[16]	E10			
73	FLASH_SRAM_A[17]	C9			
74	FLASH_SRAM_A[18]	D9			
75	FLASH_SRAM_A[19]	A7			
76	FLASH_SRAM_A[20]	A6			
77	FLASH_SRAM_A[21]	B18			
78	FLASH_SRAM_A[22]	C17			
79	FLASH_SRAM_A[23]	C18			
80	FLASH_SRAM_A[24]	G14			
81	FLASH_SRAM_A[25]	B17			
82	FLASH_NWE	D18	O	Flash 写信号	
83	FLASH_NOE	D17	O	Flash 读信号	
84	FLASH_NCS	E2	O	Flash 片选	
89	SRAM_NCS1	F13	O	SRAM1 片选	
90	SRAM_NCS2	A17	O	SRAM2 片选	
91	SRAM_BE[1]	R2	O	高字节选通	
92	SRAM_BE[2]	R1	O	低字节选通	
93	SRAM_NOE	C16	O	SRAM 读信号	
94	SRAM_NWE	G13	O	SRAM 写信号	
95	I2C_SCL	C7	IO	I2C 时钟	
96	I2C_SDA	D7	IO	I2C 数据	
97	SYS_CLK	A10	I	系统时钟输入	

序号	引脚标号	FPGA 管脚	IO 口方向	功能	备注
98	SYS_CLK1	F17	I	系统时钟输入	
99	SYS_CLK2	N2	I	系统时钟输入	
100	SYS_NRST	U9	I	FPGA 复位输入	
101	CLOCK0	F2	I	时钟输入	
102	CLOCK1	F18	I	时钟输入	
103	CLOCK2	A9	I	时钟输入	
104	CLOCK3	V10	I	时钟输入	

外扩IO口分配

序号	电路图上标号	FPGA 管脚号	I/O 方向	外设名称	跳线及复用	
1	LED[0]	R4	O	8 个独立 LED 灯，主板上的 LED1-8 与核心板上 LED1-8 共用管脚	需要短接主板上 JP6 的对应跳线	
2	LED[1]	M5	O			
3	LED[2]	T2	O			
4	LED[3]	T1	O			
5	LED[4]	E9	O			
6	LED[5]	R3	O			
7	LED[6]	R5	O			
8	LED[7]	T3	O			
9	KEY1	V9	I	独立按键，核心板上 KEY1-4 与主板上 KEY1-4 共用		
10	KEY2	U10	I			
11	KEY3	B9	I			
12	KEY4	B10	I			
13	KEY5	R18	I			
14	KEY6	R17	I			
15	KEY7	P18	I			
16	KEY8	P17	I			
17	BEEP	F8	O	蜂鸣器		
18	LCD_LIGHT	F11	O	液晶显示控制		
19	LCD_EN	F9	O			
20	DCMOTORSPEED	E17	O	直流电机 注意短接 JP1 电源跳线		
21	DCMOTORA	E18	O			
22	DCMOTORB	G17	I			
23	8563_INT	G18	I	8563		
24	LM75_OS	H17	I	LM75		
25	SEG[0]	F10	O	七段数码管 段码		—
26	SEG[1]	E11	O			

序号	电路图上标号	FPGA 管脚号	I/O 方向	外设名称	跳线及复用		
27	SEG[2]	F12	O	七段数码管 段码	—		
28	SEG[3]	C12	O				
29	SEG[4]	D12	O				
30	SEG[5]	E13	O				
31	SEG[6]	C14	O				
32	SEG[7]	D14	O				
33	DIG[0]	H15	O			七段数码管 位码	—
34	DIG[1]	E14	O				
35	DIG[2]	A18	O				
36	DIG[3]	D16	O				
37	DIG[4]	L14	O				
38	DIG[5]	L16	O				
39	DIG[6]	H16	O				
40	DIG[7]	J13	O				
41	UART_RXD	H18	I	232 串口	—		
42	UART_TXD	K18	O				
43	AD_NCS	K17	O	ADC	—		
44	AD_DAT	L18	I				
45	AD_CLK	L17	O				
46	DAC_CLK	M17	O	DAC	—		
47	DAC_DATA	M18	O				
48	DAC_LDAC	T18	O				
49	DAC_LOAD	T17	O				
50	ULN2003A	M14	O	步进电机 注意短接 JP4 电源跳线	—		
51	ULN2003B	L13	O				
52	ULN2003C	N15	O				
53	ULN2003D	N16	O				
54	IR_CLK	T16	O	红外收发 通过 JP2 跳线来选择频率	—		
55	IR_TX	R16	O				
56	IR_RX	N13	I				
90	USB_NINT	N18	I	USB 接口	—		
87	USBVIN	N17	I				
88	USB_SUSPD	待定	IO				
89	USB_nRST		O				
91	USB_nCS		O				
57	RS485_DI	待定	O			485 接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用
58	RS485_RE_DE		O				
59	485_RO		I				
60	SD_WP	待定	I	SD/MMC 卡接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用		
61	SD_INSERT		I				
62	SPI_MISO		I				
63	SPI_CLK		O				
64	SPI_MOSI		O				
65	SD_nCS		O				
66	SD_POWER		O				

序号	电路图上标号	FPGA 管脚号	I/O 方向	外设名称	跳线及复用
67	VGA_VSYNC	待定	O	VGA 接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用
68	VGA_HSYNC		O		
69	VGA_R0		O		
70	VGA_R1		O		
71	VGA_R2		O		
72	VGA_G0		O		
73	VGA_G1		O		
74	VGA_G2		O		
75	VGA_B0		O		
76	VGA_B1		O		
77	LATTICE_STR	待定	O	LED 点阵接口 注意短接 JP5 电源跳线	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用
78	LATTICE_SI		O		
79	LATTICE_SCK		O		
80	MS_DATA	待定	IO	鼠标接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用
81	MS_CLK		IO		
82	KB_DATA		IO	键盘接口	
83	KB_CLK		IO		
84	RTL8019_INT	待定	I	以太网接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用
85	RTL8019_nCS		O		
86	RTL8019_RST		O		
92	P_IO1	待定	根据使用来定义	主板上 PACK1 IO 口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用
93	P_IO2				
94	P_IO3				
95	P_IO4				
96	P_IO5				
97	P_IO6				
98	EXT_nCS			O	

核心板PACK2接口插接交通灯模块管脚配置

信号	引脚	备注
A 路口绿灯	P2	低电平点亮
A 路口黄灯	P1	
A 路口红灯	M1	
B 路口绿灯	M2	
B 路口黄灯	L1	
B 路口红灯	L2	