

伟欣科技[®]
睿智 FPGA 2C5/2C8
开发板
用户手册 1.8 版本

提醒：发现部分客户根本不查看本手册，在使用过程中完全自己摸索，走了很多弯路！手册里对新手常见问题及注意事项都有说明，您看一看，节省的是您自己的时间。在此强烈建议您收到板子，先认真阅读本文档，优先阅读第一章、第四章、第九章内容！

更新日期：2012-11-20

请关注论坛，最新文档及资料、代码将在论坛及 QQ 群内发布，不另行单独通知！

我们的 FPGA 开发网

官方网站：www.OurFPGA.com

官方淘宝店：<http://ourfpga.taobao.com> <http://shop36398534.taobao.com/>

邮箱：OurFPGA@gmail.com

购买过的客户请加此群：

262659490（目前可加，如不可加，请到淘宝官网看最新通知）

215220426（此群已满员）

155291300（此群已满员）

（收到货后，凭保修卡号及旺旺 ID 号申请加入，由于人多，请各位想加的朋友配合遵守，否则不理睬）

提醒：请您在使用此开发板前，认真阅读此说明书，并将此电子书妥善保管，在开发学习中，可能会经常使用查询！

欢迎对 FPGA、嵌入式、单片机等感兴趣的朋友登陆网站，参加技术交流，下载技术资料。在普通会员区，提供了约 200 多门视频教程下载（截止到 2012-2-17 的数量，每周会更新添加，数量还在增加中。。），欢迎您来访！

现在立即去体验一下：www.OurFPGA.com

开发板简介

FPGA 与 EDA 技术是目前相当热门的技术，翻遍各大招聘网站电子类招聘信息，类似“精通 FPGA 技术，熟悉 Verilog VHDL 语言”等字眼已经为应聘者所熟悉；甚至，有的时候，“熟悉 FPGA”就意味着高薪！实际上，FPGA 技术已经成为目前电子行业应用最为广泛的技术之一，未来的就业和发展前景相当好！

目前有众多朋友热切的期望能学习 FPGA 知识，FPGA 技术实践性极强，拥有一块适合学习需求的 FPGA 开发板是非常有必要的！睿智 FPGA 开发板正是为此推出的一套高性价比开发板！此开发板由“伟欣科技®”精心设计开发，采用 Altera 公司的 CycloneII 系列芯片 EP2C5Q208C8、EP2C8Q208C8 作为核心处理器进行设计，CycloneII 系列芯片可以说是目前市场上性价比最高的芯片，比第一代芯片设计上、内部的逻辑资源上都有很大的改进，同时价格也可以被广大用户接受。

睿智 FPGA 开发板采用核心板与接口板分离的方式，核心板上除 FPGA、各类存储器以及用户扩展 PACK 外，还有按键、LED 及电源插座等。因为有用户扩展 PACK，核心板完全可以脱离接口板而单独使用，通过 PACK，用于自己的设计或电子设计大赛，扩展性极好。接口板上集成了最常用和经典外围接口，所有的外设经过精心分配及设计，不需要进行任何跳线设置，实验时非常方便！总之，睿智 FPGA 开发板是完全站在用户的角度精心设计开发，简约不简单！同时，睿智 FPGA 开发板的配套光盘提供相当丰富的实验代码及各种参考文档。

睿智 FPGA 开发板用户群体面向广大的高校相关专业学生、电子爱好者、科研单位、企事业单位的开发设计人员，适合于产品原型的快速开发、学生参加各种电子设计大赛、学习 FPGA 技术入门，课程设计以及毕业设计等，亦可用于系统设计前期快速评估设计方案。特别适合 FPGA、NIIOSII、SOPC 快速入门和产品开发及验证。

第一章 注意事项（必看！）

我们认为以下几条是特别重要的提醒事项，将其放在最前面，请认真阅读并牢记！

工程未使用到的 IO 请设置为三态！

任何程序（包括自己写的、别人给的、网上下的），请一定将未定义的引脚设为三态输入(As input tri-stated)，

一定不能将未使用的引脚设置为输出 (As outputs driving ground)，否则可能会造成核心板主芯片及存储芯片冲突损坏！或是造成其它意想不到的损坏！

如何设置未使用的引脚为三态？

答：Quartus--菜单 Assignments--Device---Device and Pin Options---Unused Pins---Reserve all unused pins==改为 As input tri-stated

每个例程下载前都养成习惯，先查下设置对不对，不对的话，改好，要重新编译一下（别忘记这步！）

如果程序下载进去，数码管、LED 灯都乱亮了，那肯定是没设置好，请严格遵照上面的修改好，重新编译！

我们把这条写在最前面，是希望大家重视，请不要忽略！

1.1 软件及环境

强烈建议使用 **Windows XP 中文版** 系统使用开发板及安装相关软件！本开发板配套使用的说明文档、代码、程序、文档等信息均在 XP 中文系统下测试。使用统一的电脑环境，有利于避免使用其它系统造成的未知错误，也有利于出现问题后，和技术支持人员沟通，利于 QQ 群学友沟通，利于论坛各学友沟通交流！

Altera 的软件版本在不断的升级，目前我们推荐使用最新版本 QuartusII11.0,此版本软件经作者使用，认为各方面功能均达到最好，并且支持中文注释（此功能对于初学者是很非常方便的），光盘中附带了软件安装包。**Quartus II 软件和 NIOS 软件的版本必须一致，并安装在同一个目录下面，安装目录不要有中文和空格。**详细情况请参考后面章节的安装说明。

关于 QuartusII 软件安装破解，在后面有详细步骤说明，请大家耐心阅读，并仔细认真按说明操作，不要想当然，目前发现，很多客户收到开发板，出现问题最多的就是软件的安装和破解，明明手册提供了安装详细说明，可有部分人就是不愿意看，自己在那里瞎捣腾，结果弄得软件装不好，破解了不成功，在那里急得蹦，火急火燎的找技术支持，但又听不进去建议。在这里强调一下，初学者还是认真的按说明来操作，安装与破解不是技术活，没有任何技术含量，只要按手册上认真操作，一定能成功。

1.2 JTAG 及 AS 的插拔安全

开发板带有 JTAG 和 AS 下载接口，**严禁带电拔插 JTAG 及 AS 下载线！**带电操作容易对 FPGA 芯片的内部配置部分电路造成致命损坏，损坏后是无法修复的！而且 FPGA 主芯片是最易损坏的，也是不给保修的。请牢记一条：**插拔下载线时必须断电！**这个与平时大家使用电脑主板插拔内存时差不多。这里再罗嗦几句：“别说我们没提醒大家，既然提到，请大家**严格遵照，这是为了您的利益，不要不当回事**”。平时建议大家使用 JTAG 下载插口，如图 1.1 所示，红色框的为 JTAG 下载口。AS 插口在必要时再使用。

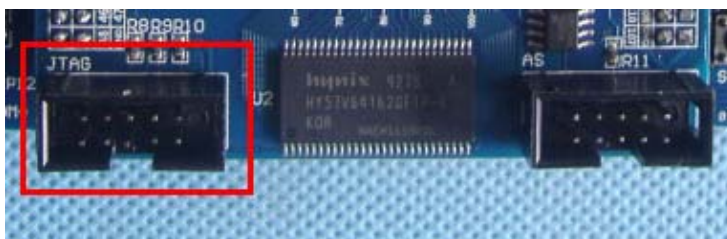


图 1.1 下载接口 JTGA 和 AS

1.3 外接电源

开发板套装里面附带标准 5V/2A 电源一条，5V 输入，内正外负。建议使用我们标配电源，**如果使用其它电源，请您一定确认输出电压是否为 5V**，以免由于电压不同造成开发板损坏！

核心板和接口板均设有 5V 电源插口，位置在 PCB 左上角。如图 1.2 所示：使用开发板时，只用随便接一个插口即可。**不必要核心板和接口板都插电源**。因为电源走线内部是相通的。

补充下：板子的电源开关在哪里？不要嫌我啰嗦，确实有不熟悉的朋友不知道，我们尽量照顾到各种基础的朋友，在这里还是写明比较好。注意图 1.2 蓝白色那个按键就是全板电源总开关，按下后即可接通电源，这时底板与核心板左上角均有电源指示灯 LED 点亮。正常情况下：板载的流水灯程序就运行了。



图 1.2 两个 5V 电源接口

1.4 核心板与接口板连接

核心板与接口板是通过三排接插口相连的，核心板上是双排座，接口板上是双排针。发货到您手中的开发板，两块 PCB 板是已经装配好的。如果不单独使用核心板，那么平时不要将核心板拔下，尽量避免频繁插拔两块板。即使要插拔时，请尽量小心谨慎些。

取下核心板时，建议一手固定接口板，另一手拿住核心板 PCB 四周，用适度的力取下，取下的过程中尽量保持核心板的平整度，不时调整四周用力力度，徐徐用力，保持每边尽量均匀受力。参考图 1.3。

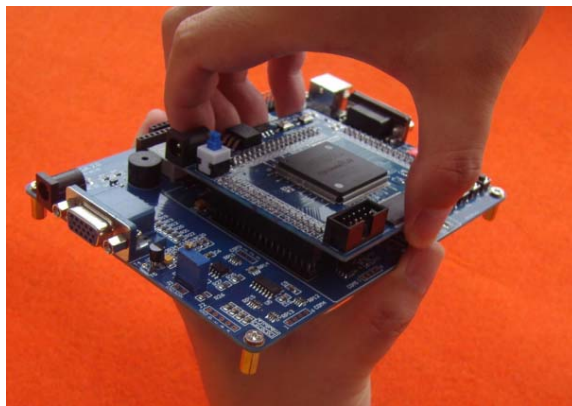


图 1.3 取下核心板

安装核心板时，注意核心板方向，要将核心板 PCB 四周边与接口板上白色示意线对齐，并且保证插座和插针一一对应，确认无误后，方可安装。参考图 1.4 示意图。

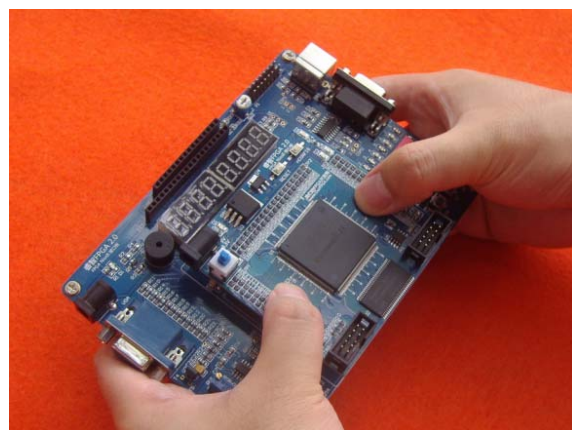


图 1.4 安装核心板

第二章 硬件资源描述

开发板套件硬件是保证实验学习的基础，这部分内容主要针对硬件部分做简单描述，可适当阅读或翻查，特别是涉及到硬件接口定义等信息时，了解这部分内容很有必要。同时，可与光盘附带的硬件原理图一并参考使用。

2.1 睿智 FPGA 开发板照片

图 2.1-2.7 为睿智 FPGA 开发板照片。**注意：**拿到手上的硬件上可能与下图略有变动，以实际产品为准。

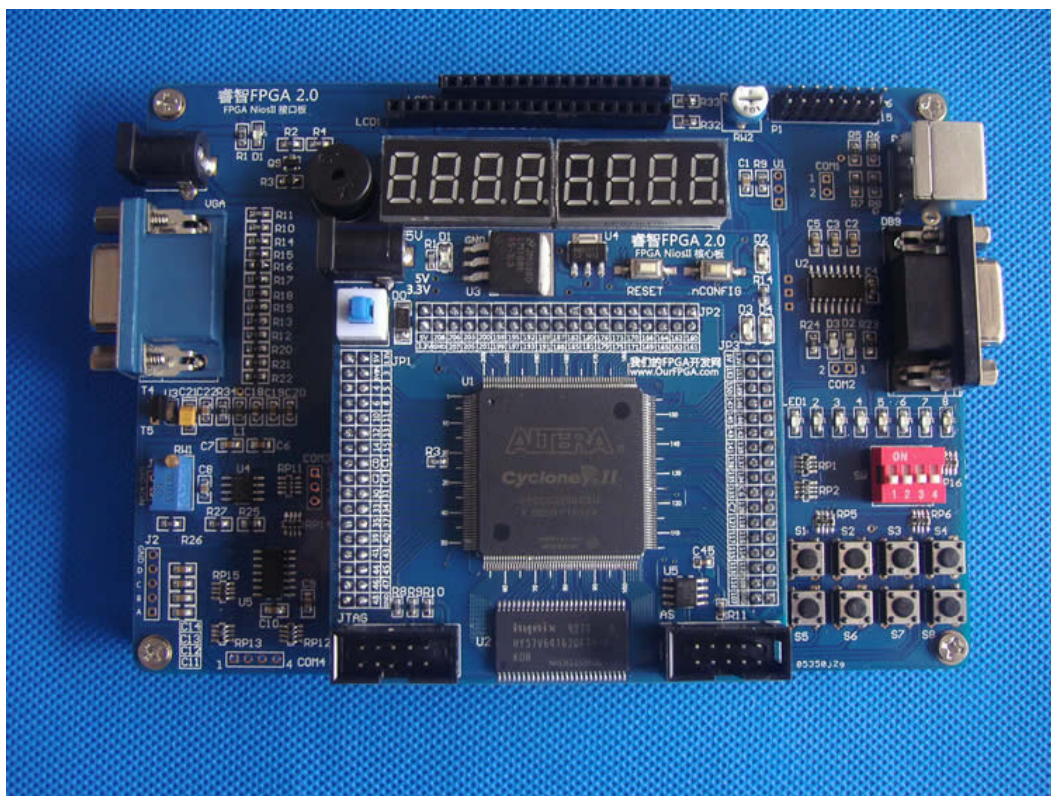


图 2.1 睿智 FPGA 开发板照片（一）

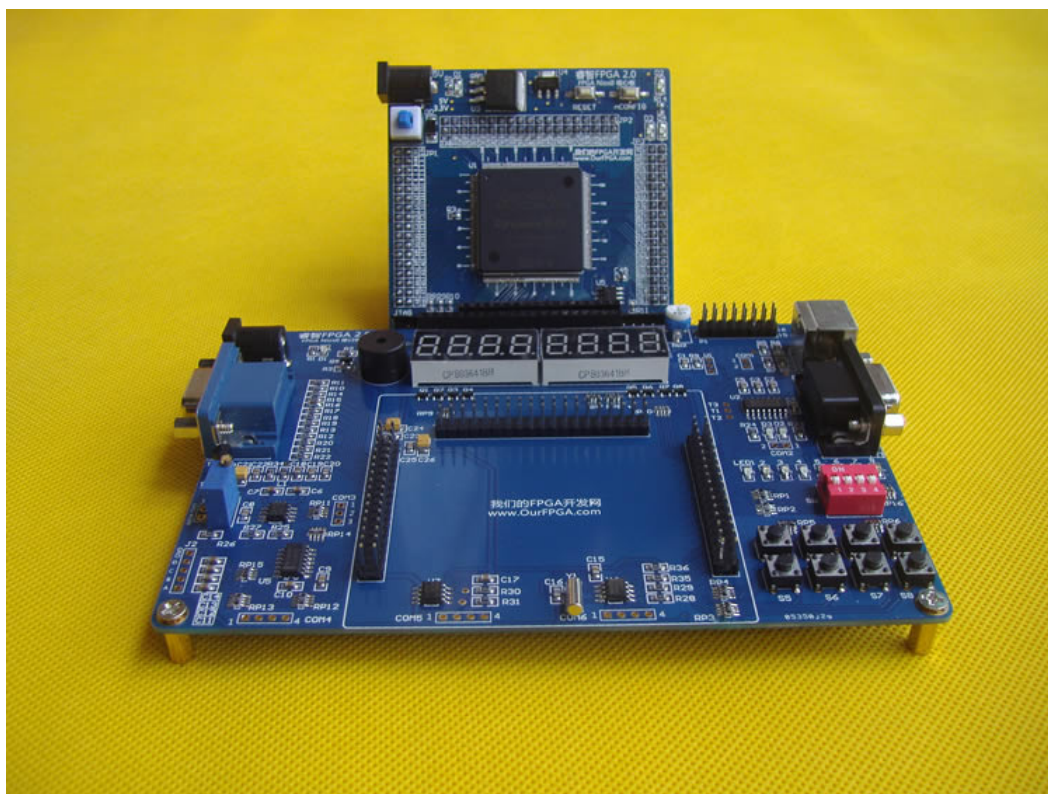


图 2.2 睿智 FPGA 开发板照片 (二)

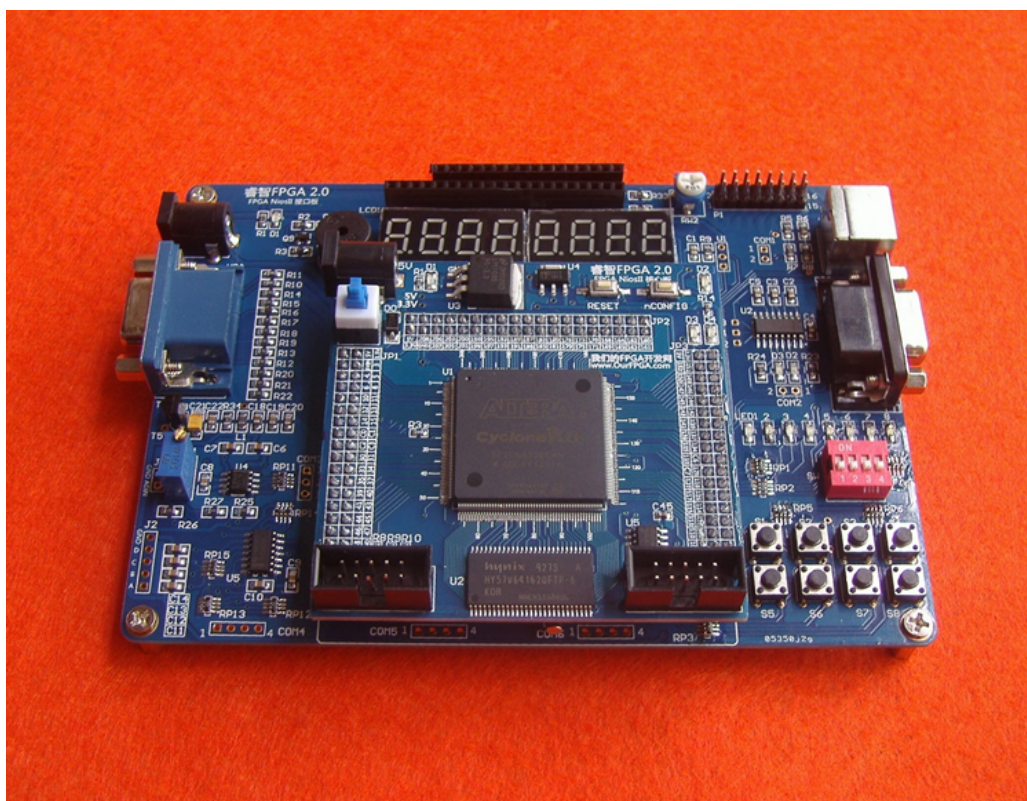


图 2.3 睿智 FPGA 开发板照片 (三)

2.2 睿智 FPGA 开发板硬件介绍

2.2.1 硬件资源图示

以下资源图，以 EP2C5Q208C8N 主芯片为例，**注意：拿到手上的硬件上可能与下图略有变动，以实际产品为准。**

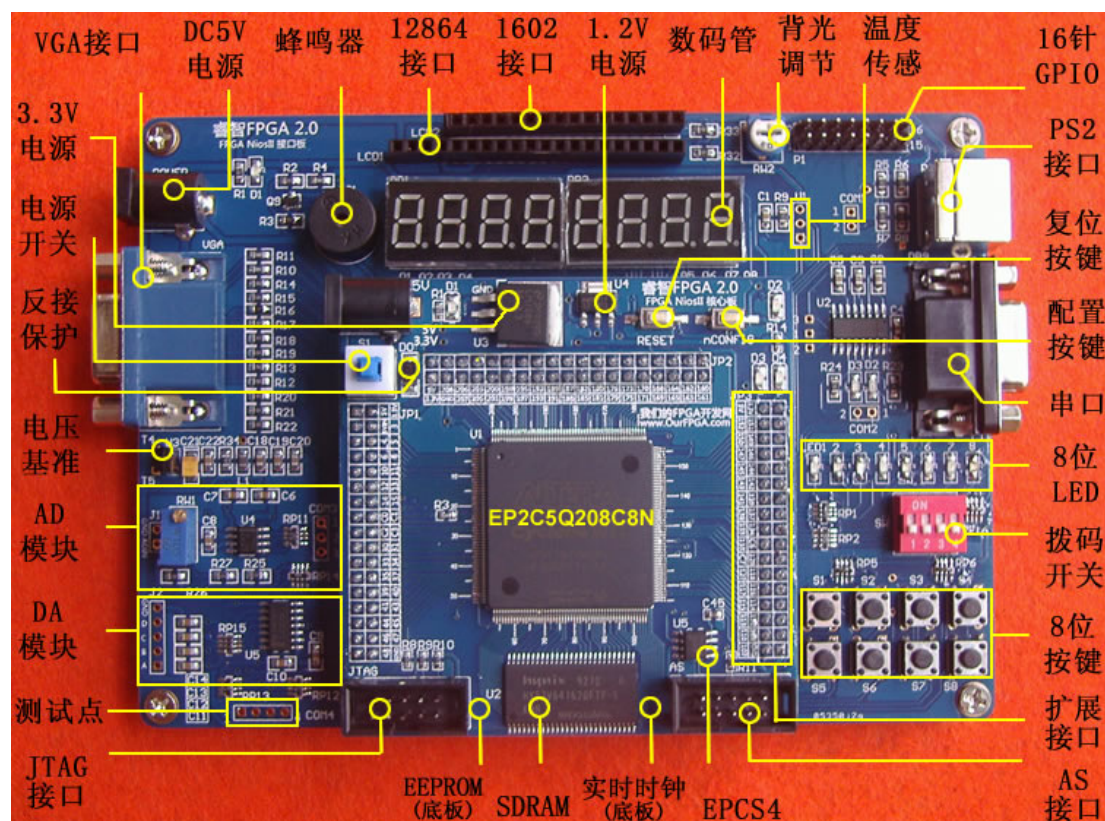


图 2.8 开发板硬件资源图示

2.2.2 核心板资源:

FPGA 主芯片采用 Altera 公司高性价比 FPGA: CycloneII 系列 EP2C5Q208C8N 或 EP2C8Q208C8N

板载 EPCS4N 或 EPCS16N 串行配置芯片，同时支持 JTAG 和 AS 模式；

采用 64Mbit 的 SDRAM，足够胜任 NIOSII 设计；

板载 50MHz 有源晶振，提供系统工作主时钟；

采用大功率 LDO 电源管理芯片 1085-3.3V，最大支持 3A 的 3.3V 电压输出；

采用 1117-1.2V 稳压芯片，提供 FPGA 内核电压；

精心的去耦设计，采用大量去耦电容，PLL 电源采用 PI 型滤波；

提供双 5V 直流电源插座，方便核心板单独使用和套装使用；

红色电源指示灯及 IN5819 高速肖特基二极管，防止电源反接；

自锁按键电源开关；

一个系统复位按键 Reset，也可做为用户输入按键；

重配置按键及配置成功指示灯；

所有输入输出精心分配，使用 3 个扩展接口插座，通用 2.54mm 间距；

JTAG 下载接口，对应下载的文件是 SOF 文件，速度快，JTAG 将程序直接下载到 FPGA 中，但是掉电程序丢失，平时学习推荐使用 JTAG 方式，最后固化程序的时候再通过 AS 方式将程序下载到配置芯片中即可；

AS 下载接口，对应下载的是 POF 文件，速度相对较慢，需要重新上电并且拔掉下载线，才能工作，操作相对麻烦，不推荐学习的时候使用。

2.2.3 接口板资源：

DC5V 接口及红色 LED 电源指示灯；

板载 8 个独立按键，可做按键控制，数字逻辑基础实验等；

板载 8 位 LED 发光二极管，做数字逻辑基础流水，显示等实验；

板载 8 位数码管，做动态或静态数码管显示实验，频率计，秒表等；

板载 4 位拨码开关，可做开关控制等实验；

板载 1 位蜂鸣器，可用作发声及音乐实验；

设有 LCD1602 液晶屏接口，做字符显示实验（不包括 LCD）；

设有 LCD12864 液晶屏接口，做汉字、字符等显示实验(不包括 LCD)；

RS232 串口，可做串口通讯实验；

PS2 接口，可做 PS/2 键盘实验；

具有温度传感器接口（标配不带温传元件），可以做温度计实验；

TLC549 AD 转换器，可以做电压表等实验；

TLC5620 DA 转换器，可做 DA 输出控制实验；

256 色 VGA 接口，可做显示器实验等；

32.768KHz 基准晶振；

TL431, 可作为 2.5V 电压基准源,

可调电位器, 调节 DA 值;

I2C 串行 EEPROMAT24C08, 做 IIC 总线实验;

PCF8563T 实时时钟;

2.3 核心板电路分析

2.3.1 FPGA 主芯片

核心板上的 FPGA 芯片采用的是 Cyclone II 系列的 EP2C5Q208C8N 或 EP2C8Q208C8N, 此芯片资源丰富, 价格适中, 非常适合 FPGA 初中级学习使用, 它的资源如图 2.9 所示。

Feature	EP2C5 (2)	EP2C8 (2)	EP2C15 (1)	EP2C20 (2)	EP2C35	EP2C50	EP2C70
LEs	4,608	8,256	14,448	18,752	33,216	50,528	68,416
M4K RAM blocks (4 Kbits plus 512 parity bits)	26	36	52	52	105	129	250
Total RAM bits	119,808	165,888	239,616	239,616	483,840	594,432	1,152,000
Embedded multipliers (3)	13	18	26	26	35	86	150
PLLs	2	2	4	4	4	4	4

图 2.9 EP2C5Q208C8N 芯片资源截图

通常, 芯片的逻辑单元和 RAM 的数量是重要的参考指标, 对于 EP2C5Q208 和 EP2C8Q208 来说, 已经足够初学者使用了。核心板上 FPGA 主芯片的原理图如图 2.10 所示。如果原理图在本文中显示不清晰, 请参考配套光盘中的 PDF 格式原理图, 可放大查阅。

EP2C5Q208 和 EP2C8Q208 除了图 2.9 的区别外, 封装上是差不多的, 仅有 4 个引脚不同。EP2C5Q208 的引脚 32、120、36、119 都是普通 IO, 而 EP2C8Q208 的 32、120 脚是接 VCCINT, 36, 119 脚接 GND。请注意这些重要信息。

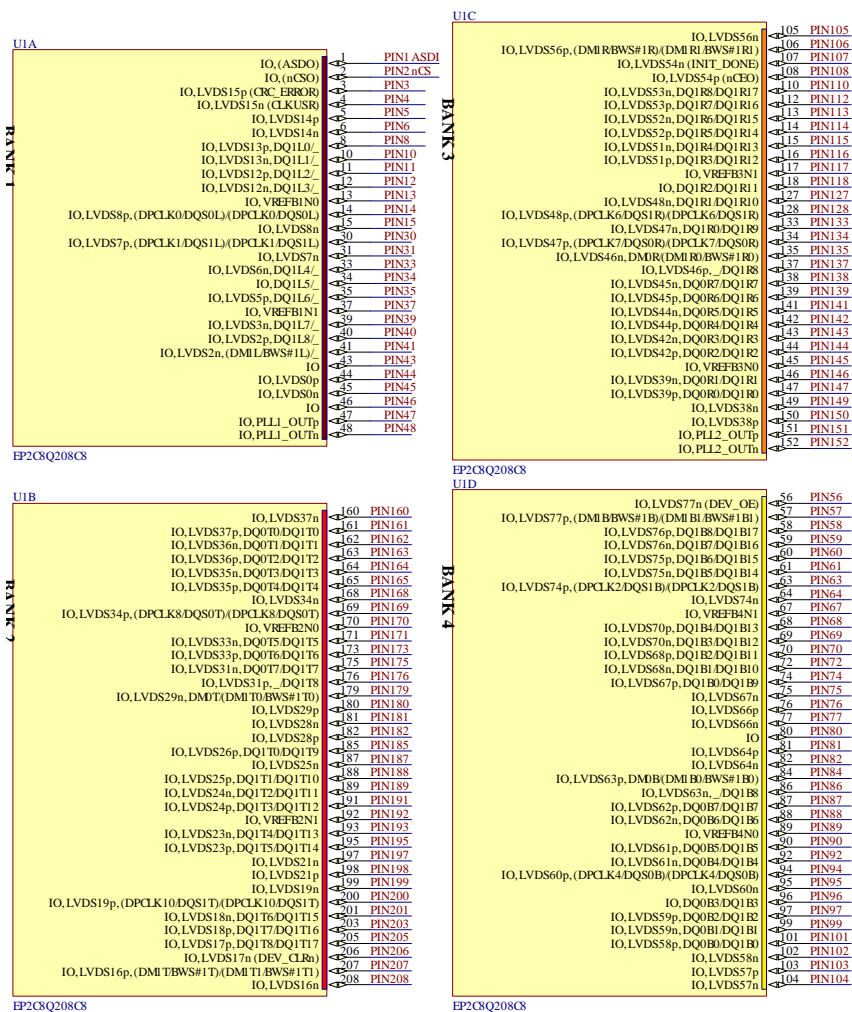


图 2.10 EP2C8Q208C8N 的 4 个 BANK 图

2.3.2 存储器 SDRAM 电路

核心板选用的 SDRAM 芯片是 HY57V641620FTP 为 64Mbit 容量，地址为 A0~A11，SDRAM 的电源部分使用多个 104 电容进行了滤波处理，保证了芯片工作的电源稳定性。SDRAM 部分原理图如图 2.11 所示：

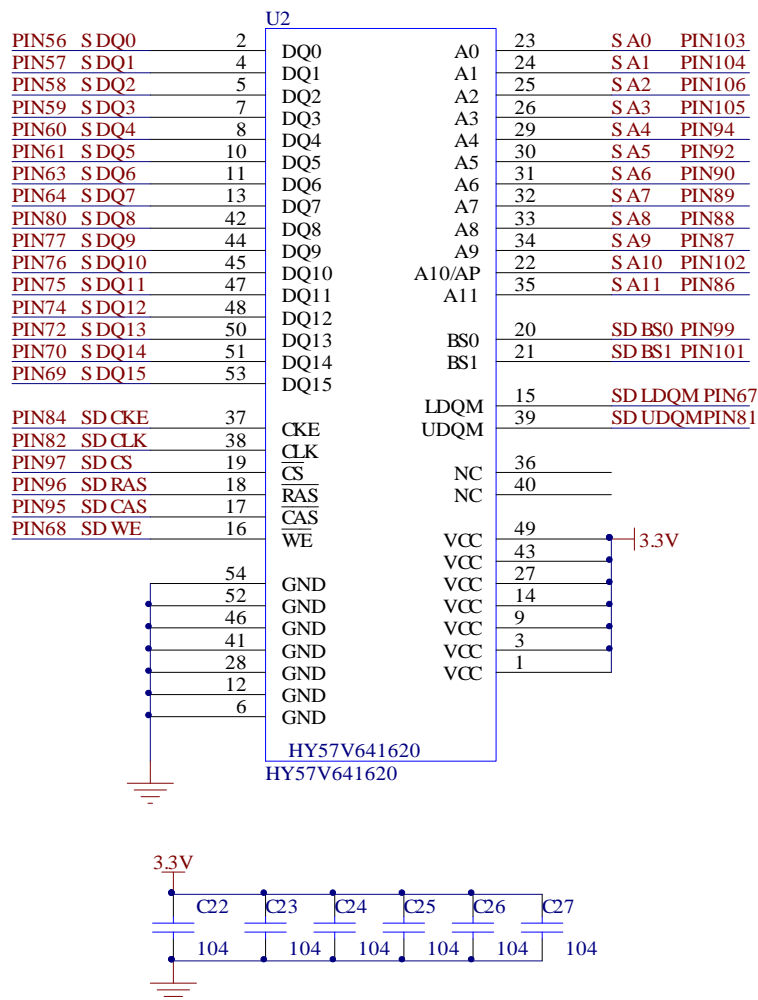


图 2.11 SDRAM 原理图

2.3.3 电源电路

电源是保证整个开发系统正常工作最重要的部分。核心板外部输入 5V 电源，经过 1085-3.3V 稳压后输出 3.3V，3.3V 主要用于给 FPGA 所有 IO 口，核心板存储器电路、串行配置器件、复位电路和 LED 等供电。然后 3.3V 送给 1117-1.2V 稳压，提供 FPGA 的内部核工作，以及锁相环部分电源工作。D1 为电源工作指示灯，在核心板的左上角，D0 为 1N5819，高速肖特基二极管，防止电源接反；同时各电源输入及输出部分设计了高质量钽电容用作电源滤波处理，提高系统电源稳定性。电源部分的原理图如图 2.12 所示。

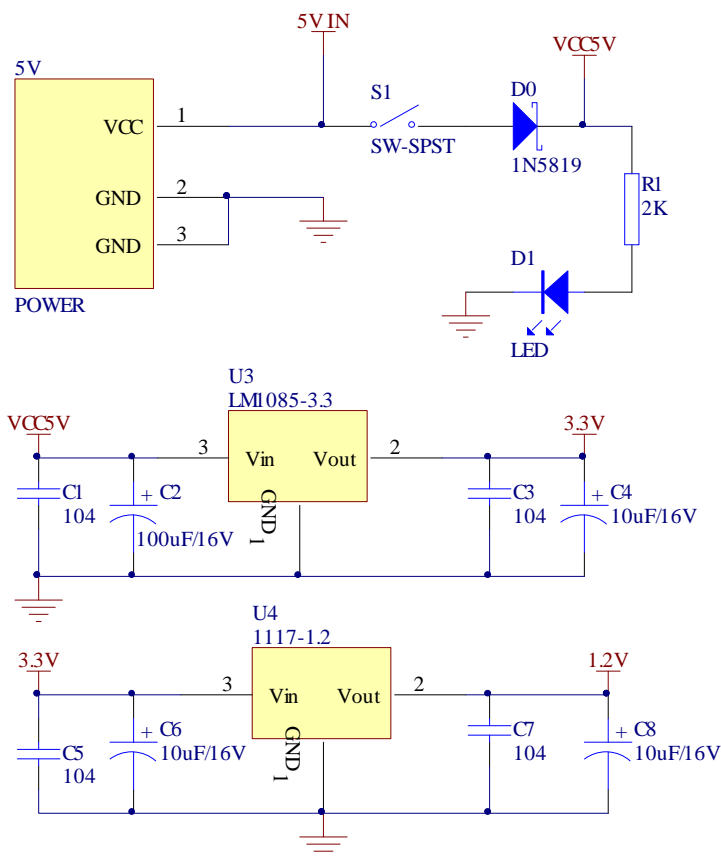


图 2.12 电源电路

图 2.13 是 FPGA 芯片 (EP2C8Q208C8N) 的电源和地处理, FPGA 内核供电为 1.2V, I/O 口供电为 3.3V。EP2C5Q208C8N 略有不同, 请参考“2.3.1 FPGA 主芯片”说明这节内容。针对 3.3V 和 1.2V 电源分别作了滤波处理, 这里要说明的是: 电源稳定性是系统非常重要的前提保证, 在电源处理上, 我们不惜成本, 使用了包括价格昂贵的钽电容在内的众多电容, (大部分电容在核心板 PCB 背面)。市面上有的开发板, 为减小成本和简单, 省掉了大量电容, 这种做法是很不可取的。

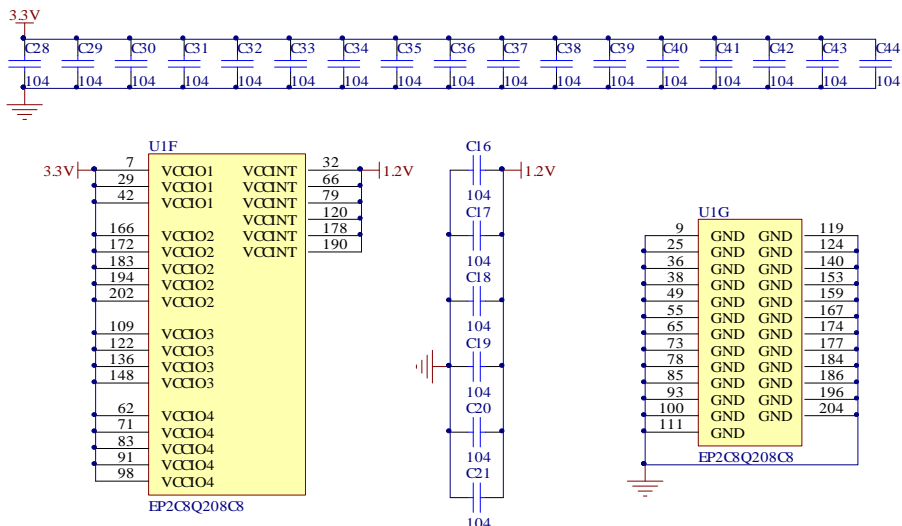


图 2.13 FPGA 供电电源电路

2.3.4 独立按键及 LED 电路

核心板上有两个 LED 灯，与接口板 LED 灯 D1, D2 引脚是共用的，位置在核心板 PCB 的右上角。在单独使用核心板时，可以用来测试基本工作情况。LED 电路如图 2.14 所示。

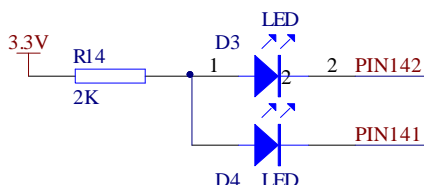


图 2.14 核心板 LED 电路

2.3.5 配置电路

这部分包括 JTAG 以及 AS 接口设计及 EPCS4 或 EPCS16 配置芯片，同时加入了下载指示电路，指示为绿色或红色 LED 灯，下载的时候，该指示灯会亮，设计了重配置按键，可以不用断电直接重新配置 FPGA。下载配置及指示电路如图 2.15 所示。

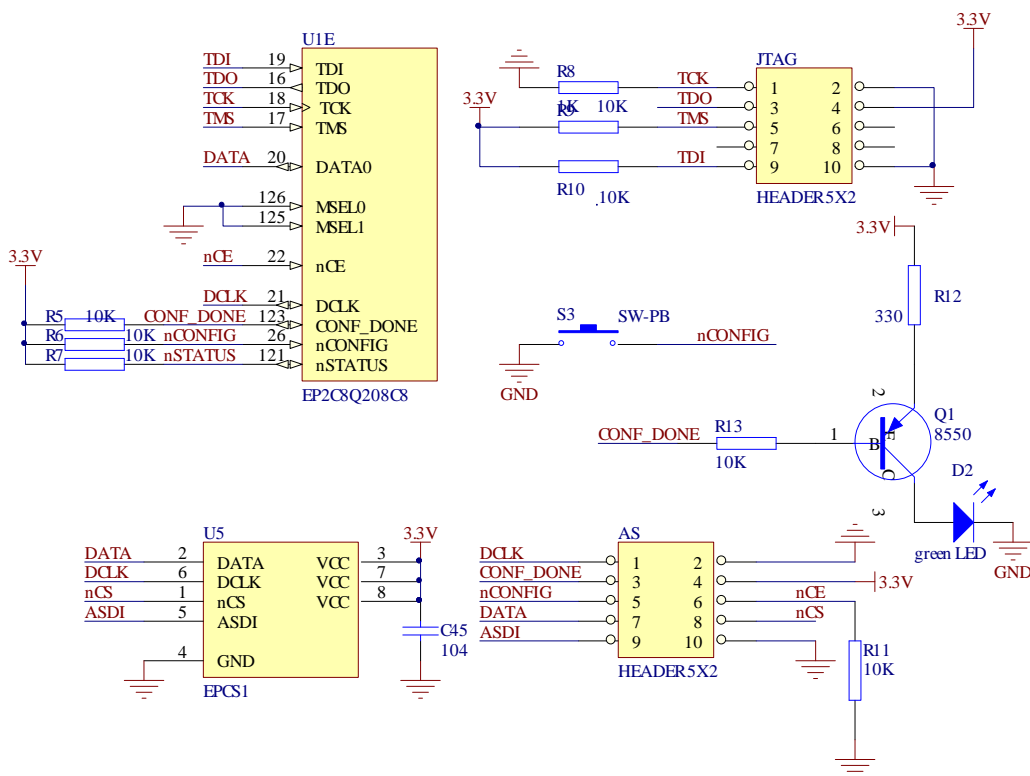


图 2.15 下载配置及指示电路

2.3.6 时钟和复位电路

开发板采用 50M 有源贴片晶体为系统提供运行时钟，时钟部分电路电源经过 π 形滤波处理，工作更加稳定可靠。时钟和复位电路电路如图 2.16 所示，开发板中对这些时钟引脚处理如下：

- (一) CLK0 和 CLK4 用作系统工作时钟，直接接入 50MHz 晶振；
- (二) CLK5 用作系统复位引脚，用户可以通过编程实现复位功能；
- (三) CLK1、CLK2、CLK3、CLK6 和 CLK7 为用户输入引脚，引出到扩展座了，供用户使用；
- (四) CLK5 未引出。

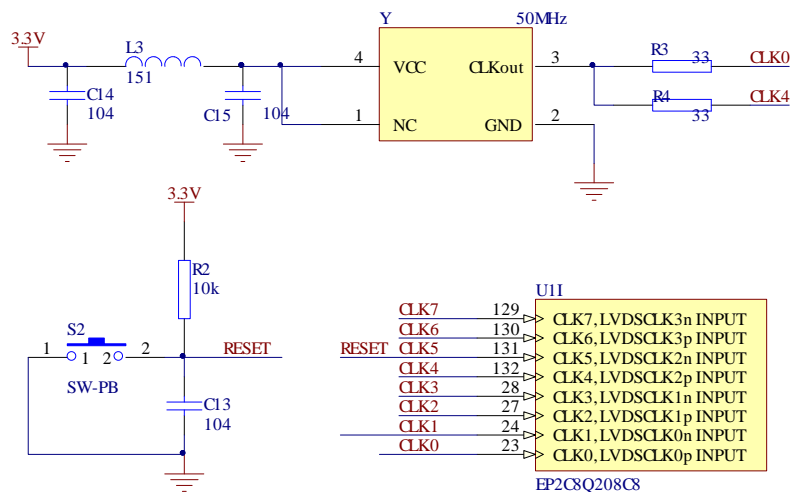


图 2.16 时钟和复位电路

2.3.7 扩展 IO 分配电路

睿智开发板采用核心板与接口板分离的形式，用户可以通过核心板外接其他电路，方便快捷，既保证了通用性，又保证了实用性。核心板 IO 通过 JP1，JP2，JP3 三个插座引出，分别对应：

- | 核心板 | 接口板 |
|-----|-------------|
| | JP1 ——— JP1 |
| | JP2 ——— JP2 |
| | JP3 ——— JP3 |

每个插座，均设置了 VCC5V、3.3V、GND 电源，方便以后单独使用核心板时，在自行开发的硬件上使用。扩展 PACK 的原理图如图 2.17 所示。

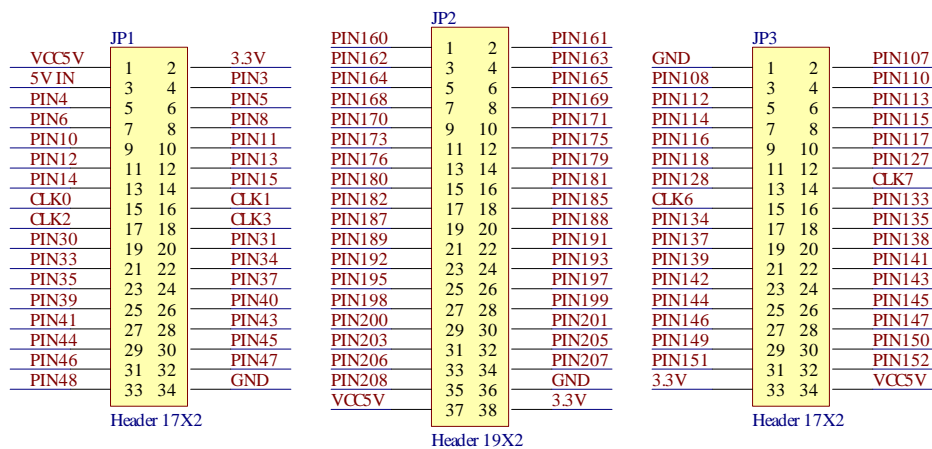


图 2.17 扩展 PACK 的原理图

2.4 接口板电路分析

2.4.1 电源电路

为方便用户，在接口板上也配了 5V 电源插口，输入 5V IN 通过核心板与接口板的互联插口相联，由核心板提供 3.3V。因此，核心板的电源是可以单独使用的，而接口板电源在没有核心板接入的情况下是无法单独使用。接口板电源部分如图 2.18 所示。

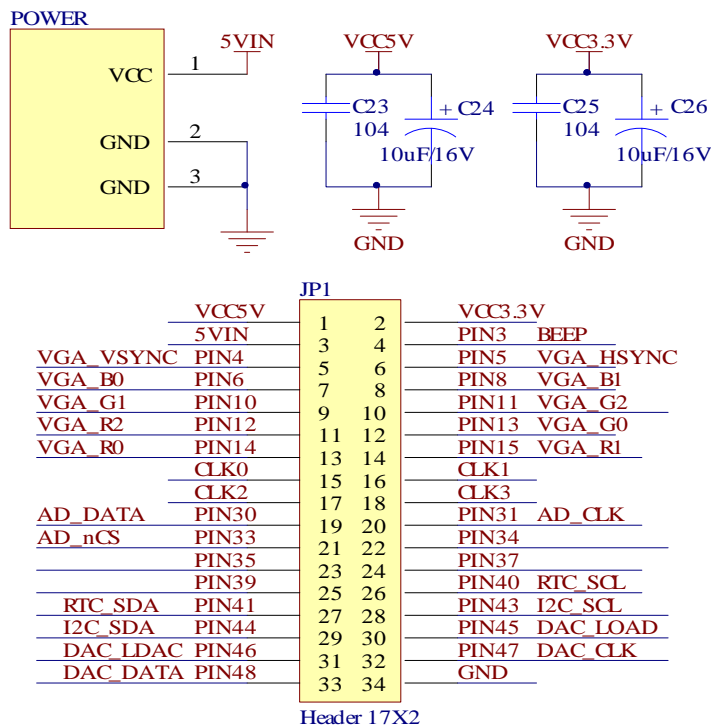


图 2.18 接口板电源部分

2.4.2 按键及 LED

接口板上具有 8 个独立按键和 8 个独立 LED，电路如图 2.19 所示，对于按键电路部分，如果输出低电平，则表示按键按下。电路中电阻 RP3,RP4 均为保护作用，以防止 FPGA 芯片 IO 设为输出且为高电平时，在按键按下时直接对地短路。

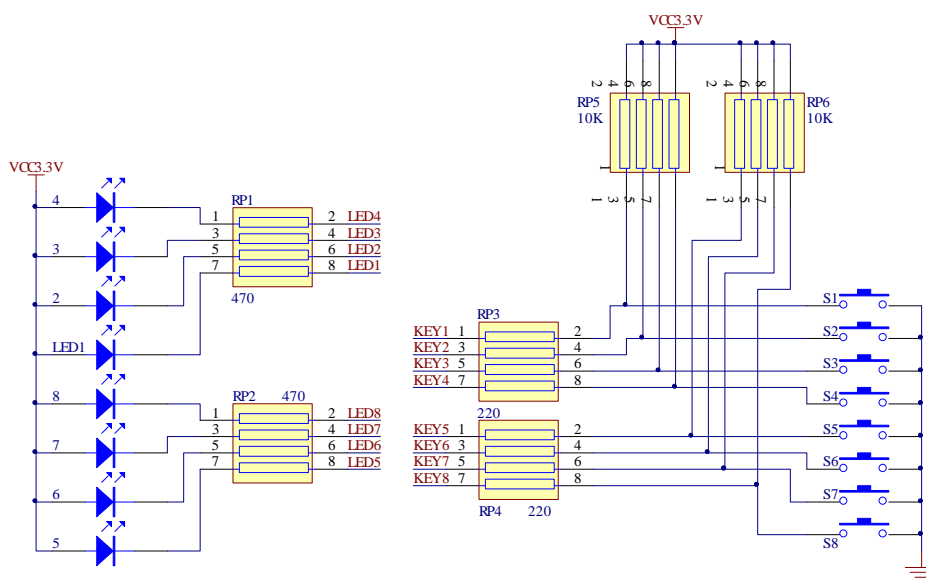


图 2.19 独立按键和 LED 电路

2.4.3 拨码开关电路

4 位拨码开关电路很简单，如图 2.20 所示。

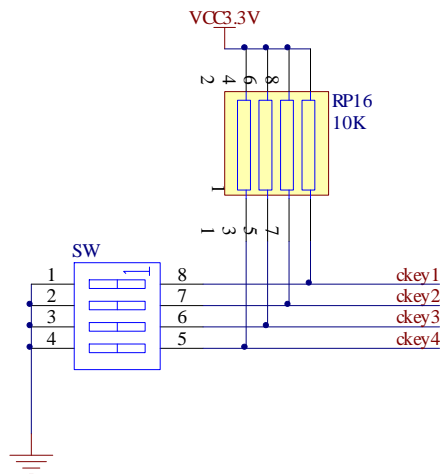


图 2.20 拨码开关电路

2.4.4 蜂鸣器电路

蜂鸣器使用 PNP 三极管驱动控制，如果在 BEEP 输入一定频率脉冲时，蜂鸣器会响，改变输入频率可以改变蜂鸣器的响声。原理图如图 2.21 所示。

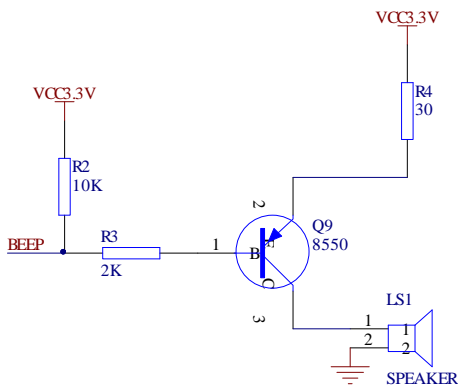


图 2.21 蜂鸣器电路

2.4.5 七段数码管显示电路

七段数码管显示电路如图 2.22 所示，数码管是共阳极的，当位码驱动信号为 0 时，对应的数码管即操作；当段码驱动信号为 0 时，对应的段码点亮。位码由于电流较大，采用了三极管驱动。

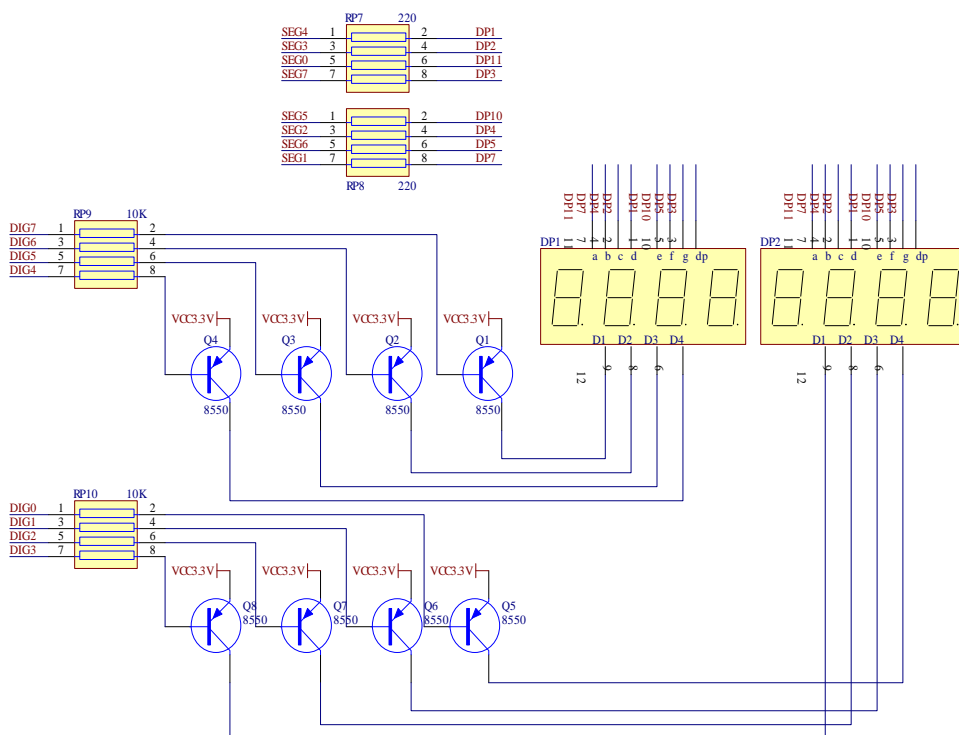


图 2.22 七段数码管显示电路

2.4.6 液晶 1602 与 12864 显示电路

接口板上设有液晶 1602 与 12864 的接口电路，如需要实验，需要自行安装选配的液晶器件，注意安装时，引脚的顺序。注意：外设板 2.0 版本 1602 液晶未加背光调节，考虑到市面上 1602 液晶混乱现状，并非所有液晶都能在不调节的情况下正常显示，2.1 版本接入了调节电阻。具体图请参考光盘内原理图。

如何判断自己手上的板子的版本呢？

最简单的方法是看开发板最右上角的扩展 IO 数量，

如果是 2X8=16 针的，则为 2.0 版本

如果是 2X10=20 针的，则为 2.1 版本

注：最新 2.2 版本外设板，右上角扩展 IO 为 40 针，请参考原理图。

2.4.8 RS232 串口电路

RS232 串口电路如图 2.25 示，由于是 3.3V 系统，所以使用了 max3232 进行 RS232 电平转换，同时，还有两个 LED 用于指示串口的工作状态。

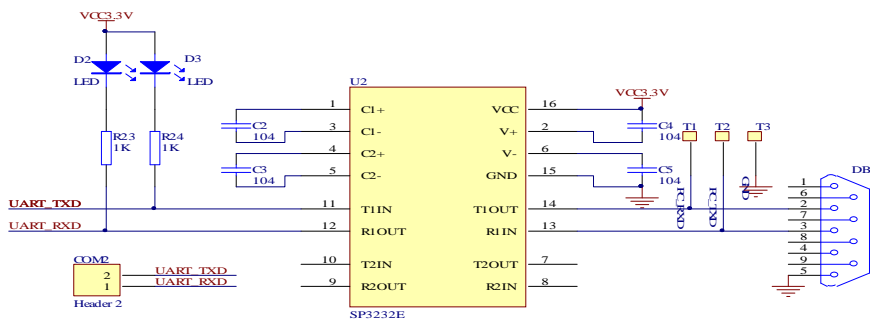


图 2.25 RS232 串口电路

2.4.9 VGA 接口电路

VGA 的接口电路如图 2.26 示，本电路采用的是电阻网络的方法来产生 VGA 所需要的不同电压信号，输入端共用了 8 个信号线，可以产生 256 色。

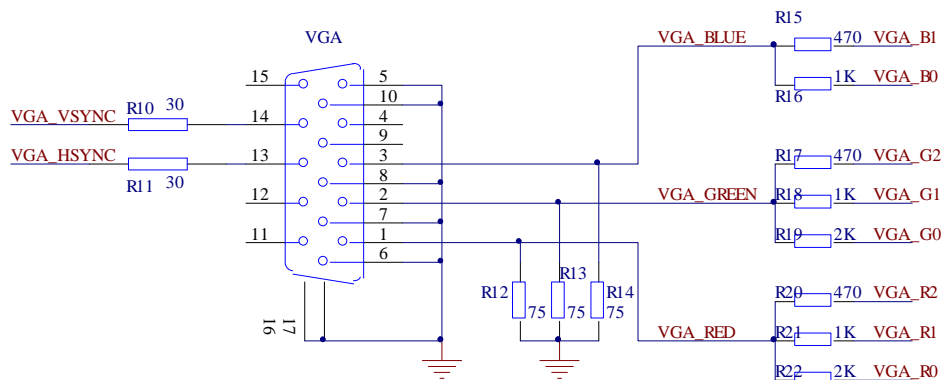


图 2.26 VGA 接口电路

2.4.10 PS/2 键盘、鼠标接口电路

PS/2 键盘鼠标接口电路原理如图 2.27 所示，使用 5V 电源供电，接口的数据线和时钟线均要接上拉电阻。

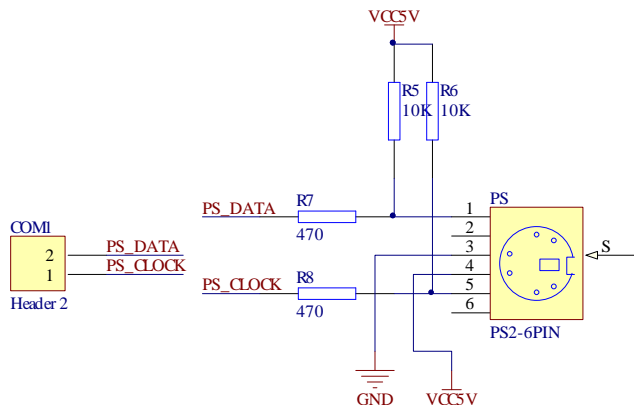


图 2.27 PS/2 键盘、鼠标接口电路

2.4.11 串行 DA、AD 电路

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

串行 AD 电路采用单通道 8 位 AD 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，AD、DA 电路如图 2.28 所示。电压基准电路由图 2.29 所示。

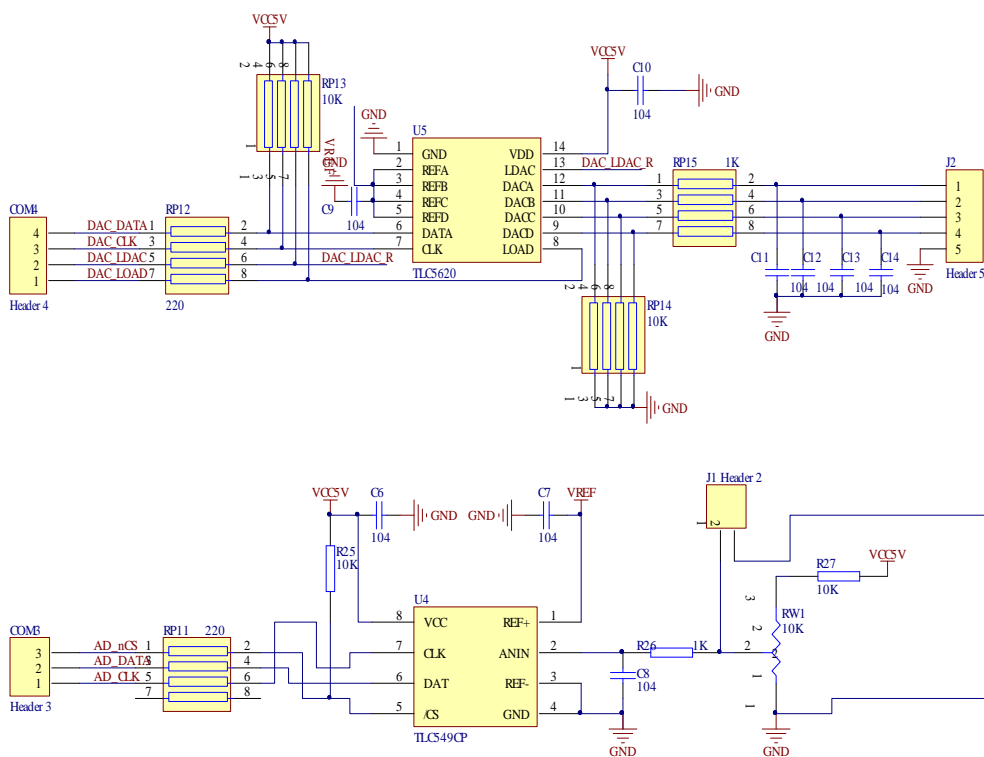


图 2.28 DA、AD 电路图

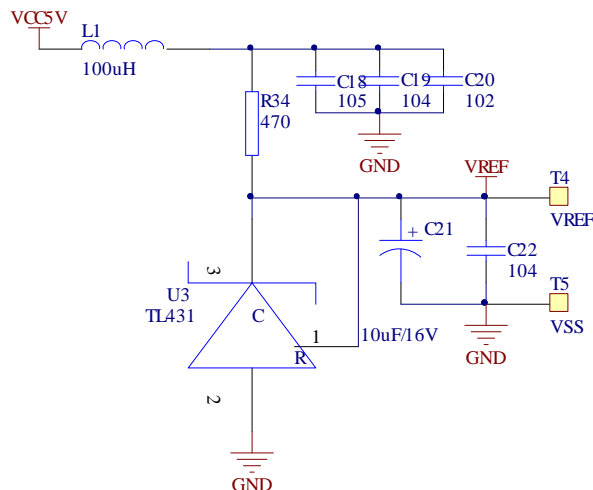


图 2.29 电压基准电路

2.4.12 实时时钟电路

实时时钟电路如图 2.30 所示。实时时钟芯片使用的是 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 总线接口串行传输。最大总线速度为 400kbps,每次读/写数据后，内嵌的字地址寄存器会自动产生增量。

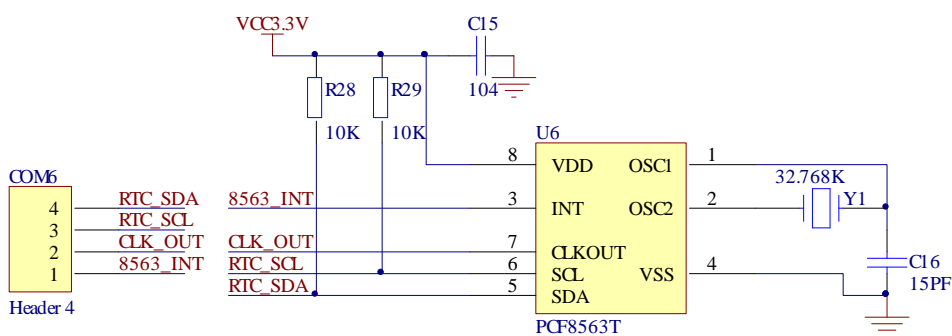


图 2.30 实时时钟电路

2.4.13 温度传感器电路

接口板上设有经典 DS18B20 温度传感器电路，原理图如图 2.31 所示，板载未焊 DS18B20 元件，为选配件。

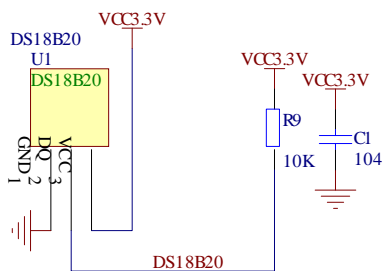


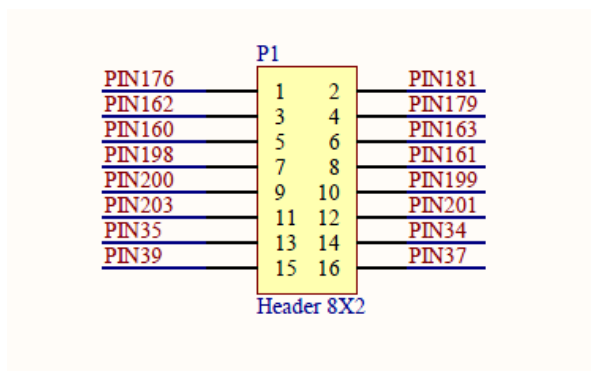
图 2.31 温度传感器电路

2.4.14 外部用户可用 IO 口

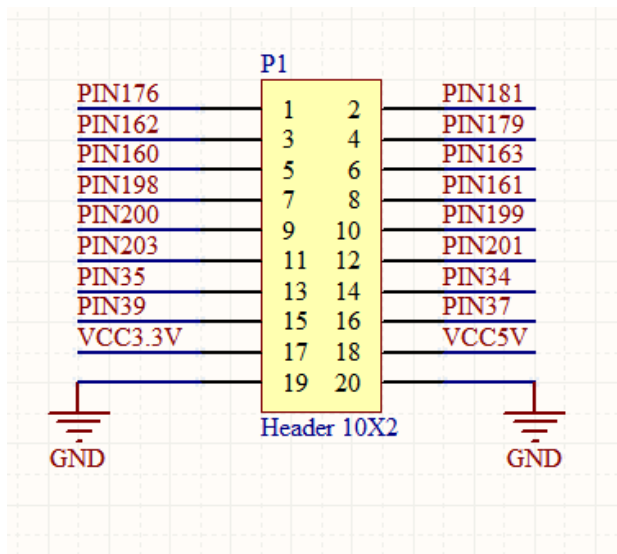
外设板扩展 IO 由于不同版本，略有不同。请参考您实际收货的版本说明使用。

如果您不使用 1602 或 12864 液晶那 11 个 IO，那么这 11 个 IO 同样也可以自行使用，如果想使用杜邦线，则可以在 1602 或 12864 的插座上插上一根单排针即可，只要发挥您的想像，板上的资源很多都可以有效利用！

#2.0 版本外扩 IO 原理图，使用 8X2 排针放在 PCB 右上角：

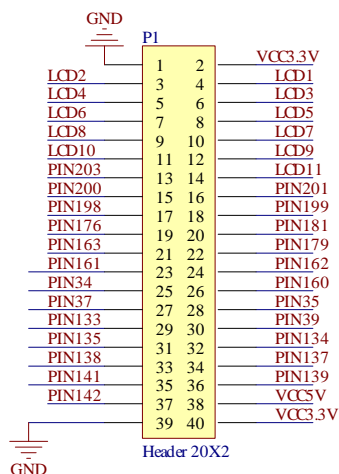


#2.1 版本外扩 IO 原理图及 PCB 示意图：





#2.2 版本外扩 IO 原理图示意图:



2.2 版本的 P1 有 35 个可用 IO，其中 16 个是板载外设从未用到的，11 个 IO 是 1602 及 12864 液晶共用 IO，另 8 个 IO 与 8 个独立 LED 共用的。在使用 P1 时，不能同时使用 12864 液晶或是 LED，在 LED1-LED8 的上方，有一个 LED 电源跳线，使用 P1 上的 IO 时，可以把短路跳帽取掉，以免受 LED 部分电路干扰。

2.4.15 关于部分外设 com 测试点的说明

2.0 版本几个外设单独引的 IO 孔 (com1-com6)，考虑工艺因素和实际用处不大，在 2.1 版本中已经去掉。如果您是 2.1 以上版本的使用者，可以忽略这部分。如果您是 2.0 版本的使用者，请参考光盘内 2.0 原理图。

第三章 配套光盘说明

睿智 FPGA 开发板配套光盘有 2DVD，提供了丰富的资源，包括最新开发软件 QuartusII 11.0 版破解版本，此版本支持中文注释，非常方便，包含全部实验代码，例程、硬件原理图，用户手册、芯片手册文档、精选 FPGA 相关电子书、常用开发工具及软件、精选视频教程等等。光盘资料还会不断更新完善，在这里就不再截图了，请购买过的朋友登陆 www.ourfpga.com，关注更新。

第四章 开发软件及软件安装

光盘中配套了相关的开发板软件，推荐使用。在技术讨论或提出问题时，建议尽量使用统一的软件版本，方便大家交流及解决技术问题。我们推荐使用的软件如下：

Quartus11.0: 主要用于编译、综合、下载；（必装）

NiosII11.0: SOPC 集成开发环境；（如果暂不学 Nios 开发的，可以不用装）

另外，提供网络下载地址连接：

下载 Quartus II 11.0 正式版

ftp://ftp.altera.com/outgoing/release/11.0_quartus_windows.exe 1.01GB

ftp://ftp.altera.com/outgoing/release/11.0_devices_windows.exe 3.92GB

ftp://ftp.altera.com/outgoing/release/11.0_DSP_builder_windows.exe 58MB

ftp://ftp.altera.com/outgoing/release/11.0_legacy_nios2_windows.exe 142MB

4.1 Quartus 11.0 安装方法

大家可以跟着一步步操作，首先安装 11.0_quartus_windows.exe,参见光盘内的文件，示意图如图 4.1 所示。

11.0_devices_windows.exe	4,114,887 KB	应用程序
11.0_dsp_builder_windows.exe	58,469 KB	应用程序
11.0_legacy_nios2_windows.exe	142,369 KB	应用程序
11.0_modelsim_ase_windows.exe	349,407 KB	应用程序
11.0_quartus_windows.exe	1,032,363 KB	应用程序

图 4.1 光盘内附带的软件安装包截图

双击后，过小会儿跳出对话框，如图 4.2 所示，提示准备解压缩，

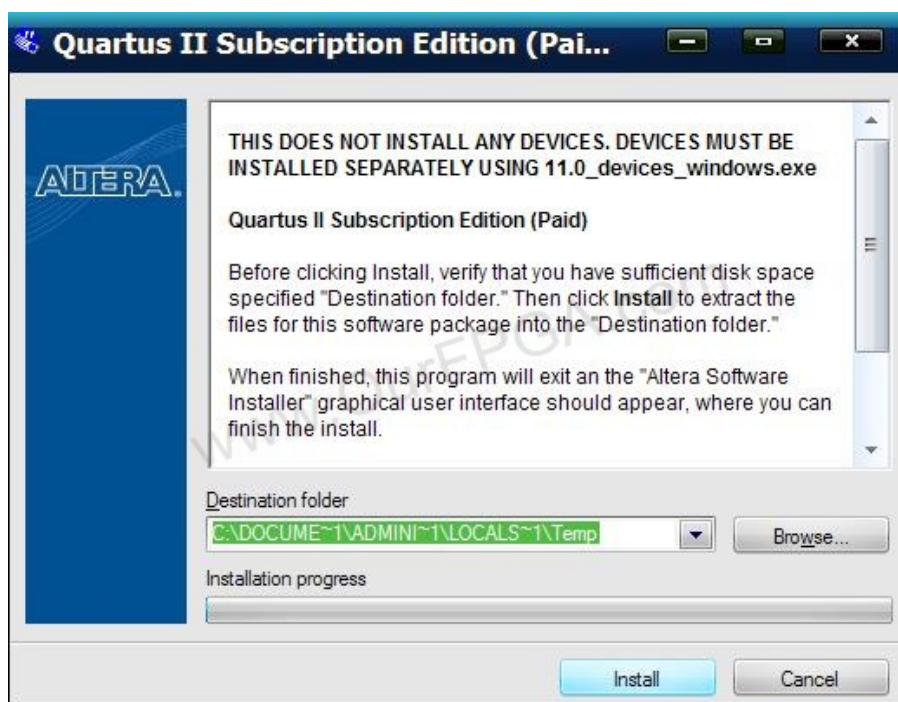


图 4.2 QuartusII 11.0 安装过程

点击 Install，开始解压，如图 4.3 所示。

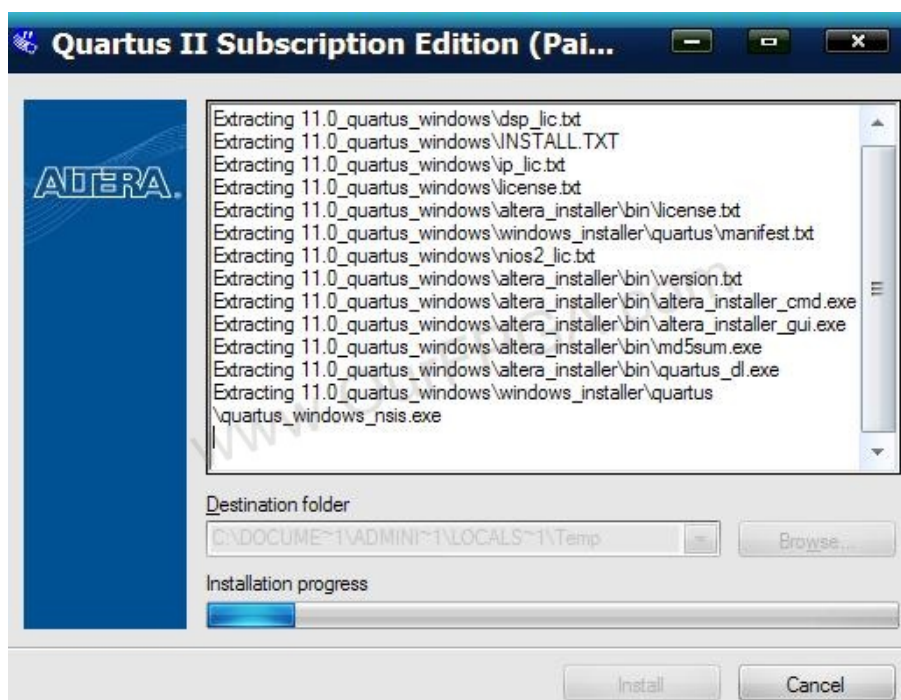


图 4.3 QuartusII 11.0 安装过程

解压完后，进行到下面一步，见图 4.4：

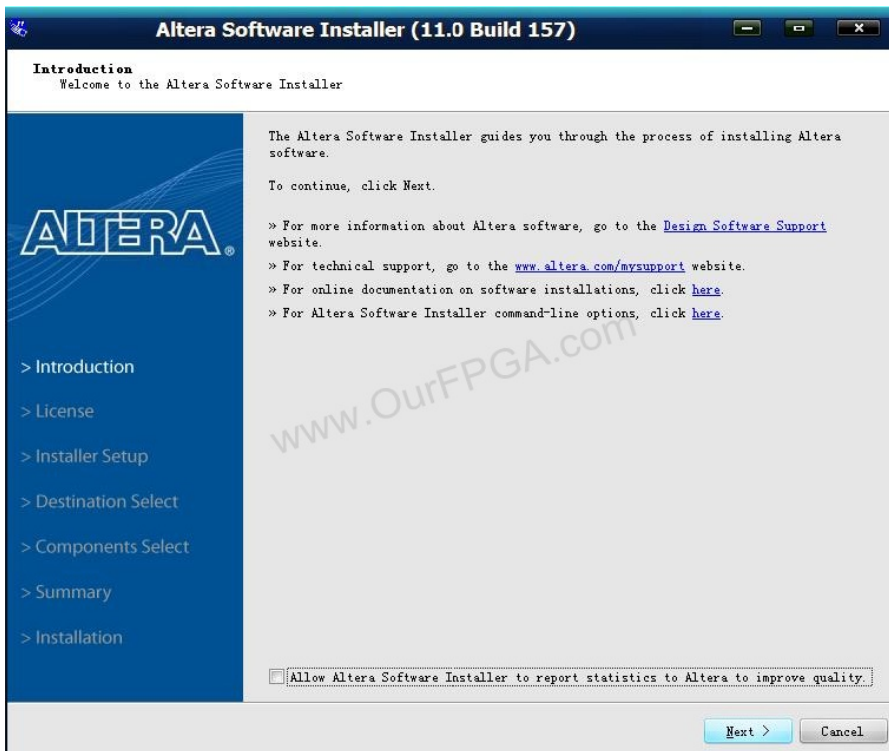


图 4.4 QuartusII 11.0 安装过程

选择 Next,勾上 “I agree.....”,再 next, 所图 4.5 所示。

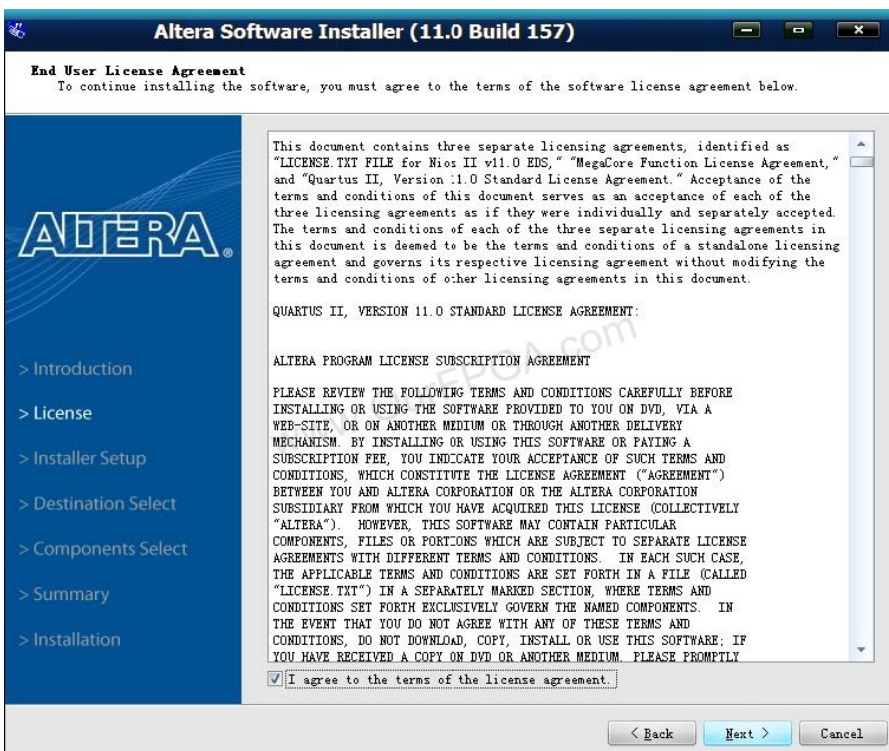


图 4.5 QuartusII 11.0 安装过程

到这一步时，注意一下：默认的目标文件夹是：c:\altera\11.0，我这里把它改成装在 D 盘了，直接将 C:改成 D:，这样软件安装在了 D:\Altera\11.0,然后一路 next,不做修改,如图 4.6 和图 4.7

所示。



图 4.6 QuartusII 11.0 安装过程



图 4.7 QuartusII 11.0 安装过程

然后就开始安装了，如图 4.8 所示，要等几分钟，可以起身出去透透气啦。



图 4.8 QuartusII 11.0 安装过程

提醒一下：安装过程中，有的杀毒软件会提示有风险，比如我电脑装的 360，就跳下面这个提示了，很显然，我们选择“允许本次操作”，如图 4.9 所示。



图 4.9 QuartusII 11.0 安装过程

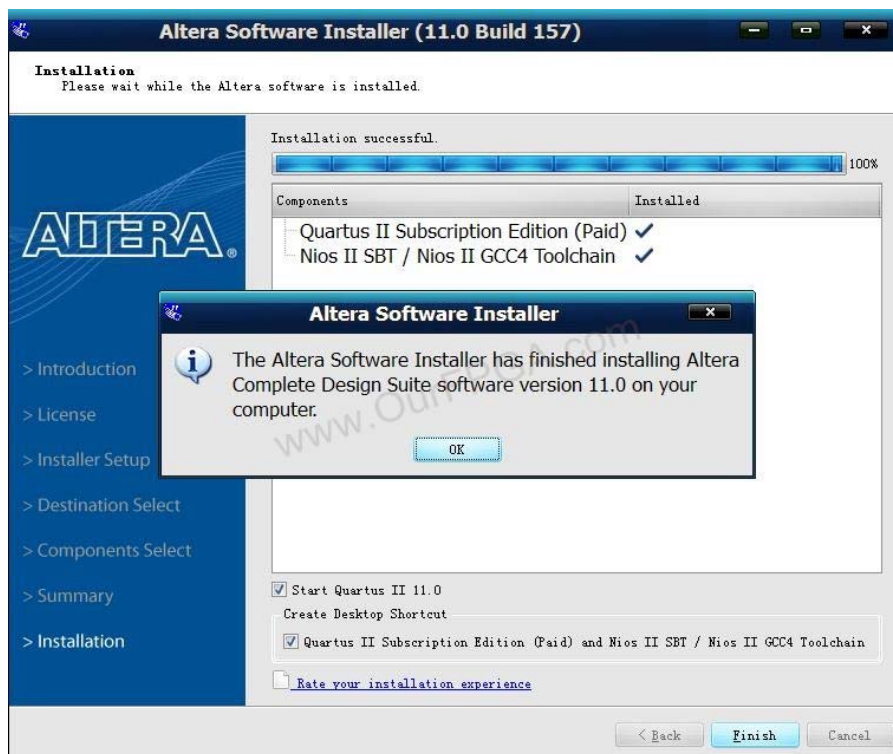


图 4.10 QuartusII 11.0 安装过程

安装好后，出现图 4.10 的提示，点确定。这里会跳出一 talkback 安装提示，一般我们不装，选择取消就行,如图 4.11 所示。

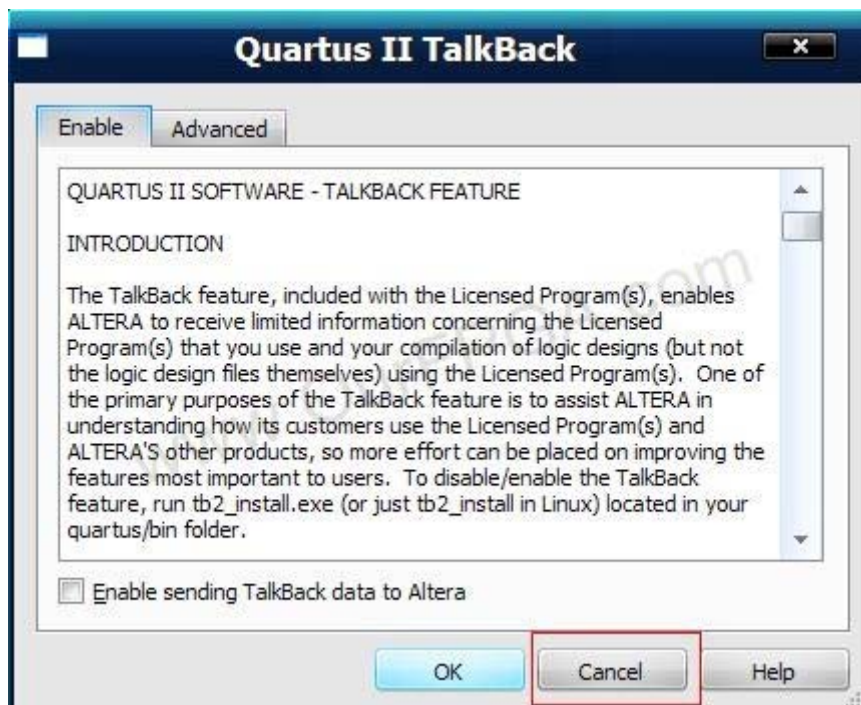


图 4.11 QuartusII 11.0 安装过程

然后到下面的这步，我们先暂时选择第二项，如图 4.12 所示，使用 30 天试用，这样连许可文件都不用了。



图 4.12 QuartusII 11.0 安装过程

这样可以打开软件了。注意，这个时候，只是装了软件，还没装 FPGA 器件文件，器件文件就是支持各种 Altera 器件的文件包，这个跟 11.0 前的版本有很大区别：以前的都是默认一起装的，现在灵活了，我们可以选择常用的器件包装，用不到的就不装了，因为有的器件可能你到老都用不上。下面我们一步步教大家装器件文件：双击 11.0_devices_windows 文件，大家发现没：后面几步和装软件差不多的，不多说，按提示一步步操作下去。好了，到了这一步，见图 4.13 所示，注意：刚才 quartus 装在哪个目录下，器件包也要装在相同目录下。我这里装在 D 盘上，修改一下。



图 4.13 QuartusII 11.0 安装过程

下面，要选择 Altera 器件家族了，见图 4.14 所示。睿智 FPGA 开发板使用的是 CycloneII 的器件，这个一定打上√，其它器件的你看着办，装也行，不装能省点硬盘空间。然后一路 next 下去，见图 4.15，图 4.16.

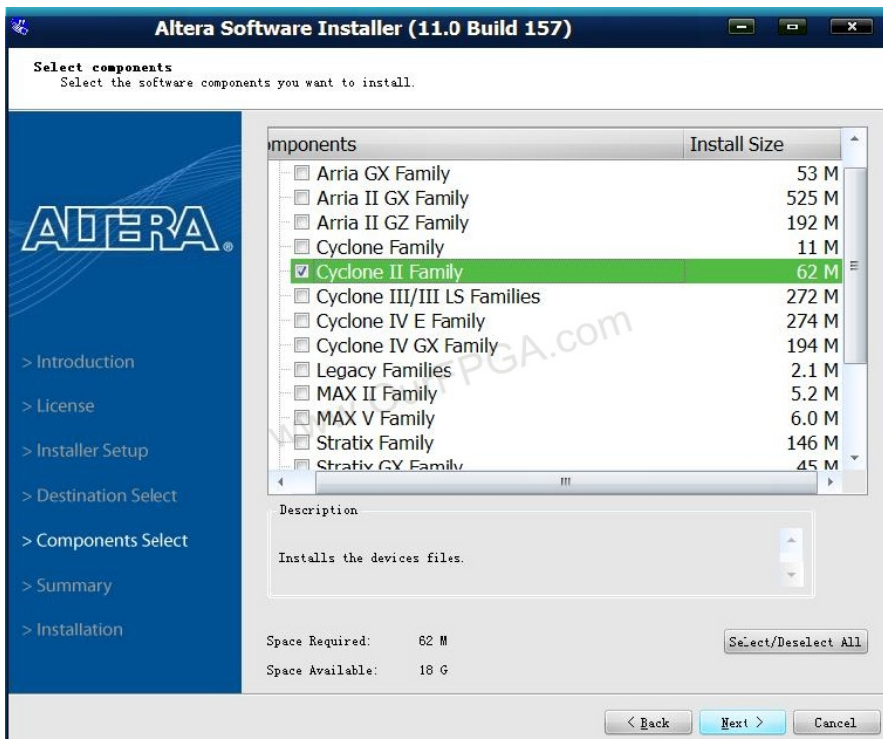


图 4.14 QuartusII 11.0 安装过程

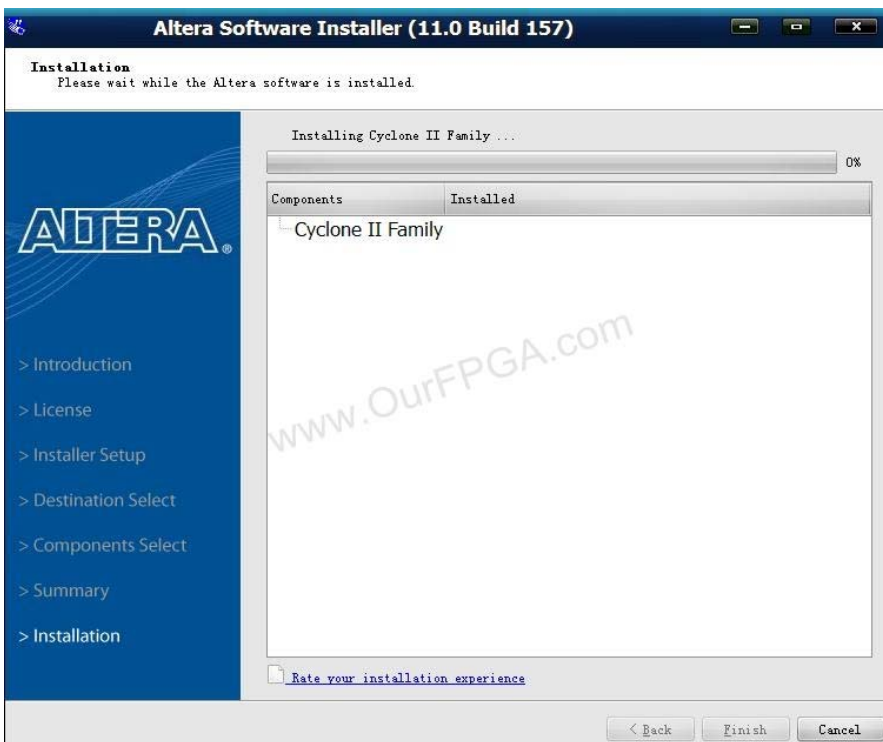


图 4.15 QuartusII 11.0 安装过程



图 4.16 QuartusII 11.0 安装过程

好了。我们装好了，但是想长期使用软件的话，就得破解。下面请各位认真操作：首先打开光盘里的 Quartus_11.0_x86 破解器(内部版)，如图 4.17 所示。瞧，360 又跳出来了，很烦，看图 4.18 所示中那个红框，选择“添加信任”，这个一定要选择哈，您电脑上如果是其它杀毒软件，可酌情处理。

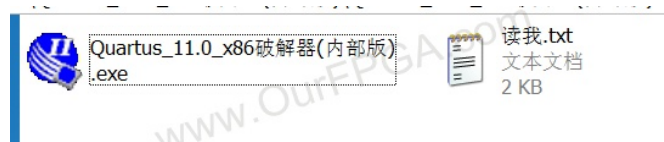


图 4.17 破解文件截图



图 4.18 破解文件使用截图

然后程序运行了，结果如图 4.19 所示：

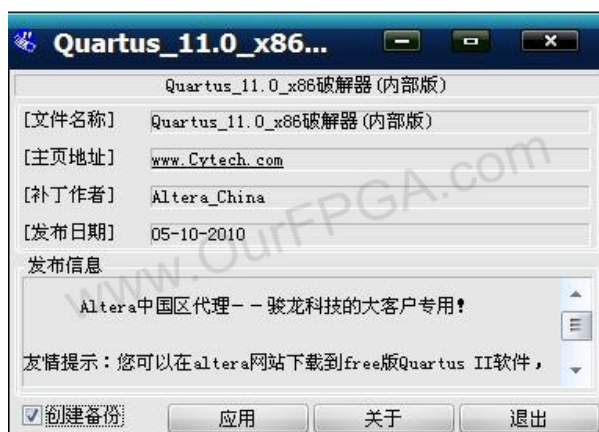


图 4.19 破解文件运行界面截图

点击“应用”后，又提示“未找到该文件。搜索该文件吗”，如图 4.20 所示，点“是”。

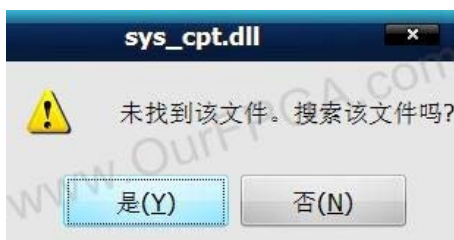


图 4.20 破解文件运行界面截图

这时候，要找到破解 D:\altera\11.0\quartus\bin 下的 sys_cpt.dll 文件，然后选中 sys_cpt.dll，点击“打开”。如图 4.21 所示。

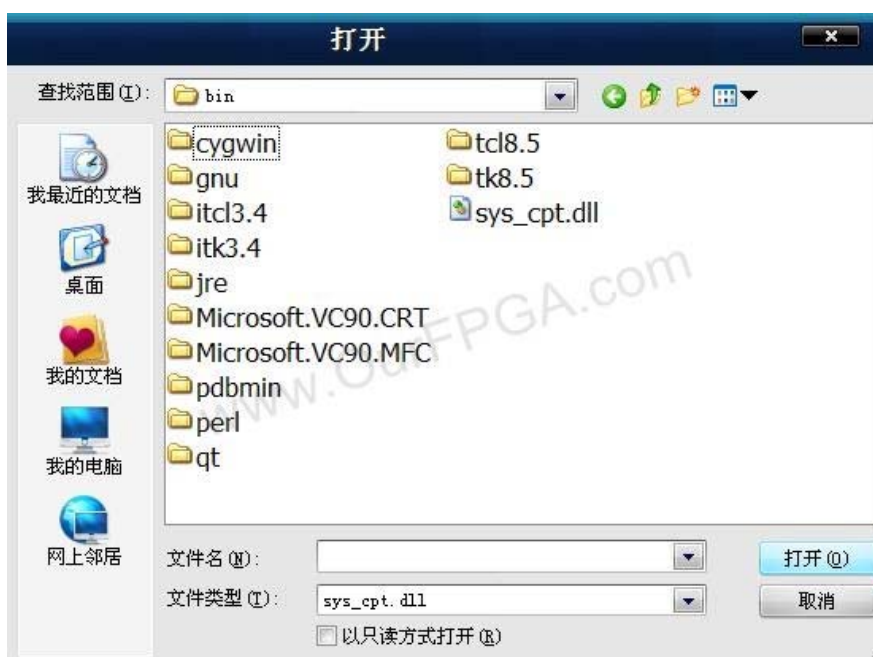


图 4.21 破解文件运行界面截图

生成 license.dat 文件，保存一下，可以直接保存在 D:\altera 下，不管保存在哪里，就是不能保存在有中文目录或有空格的目录下。

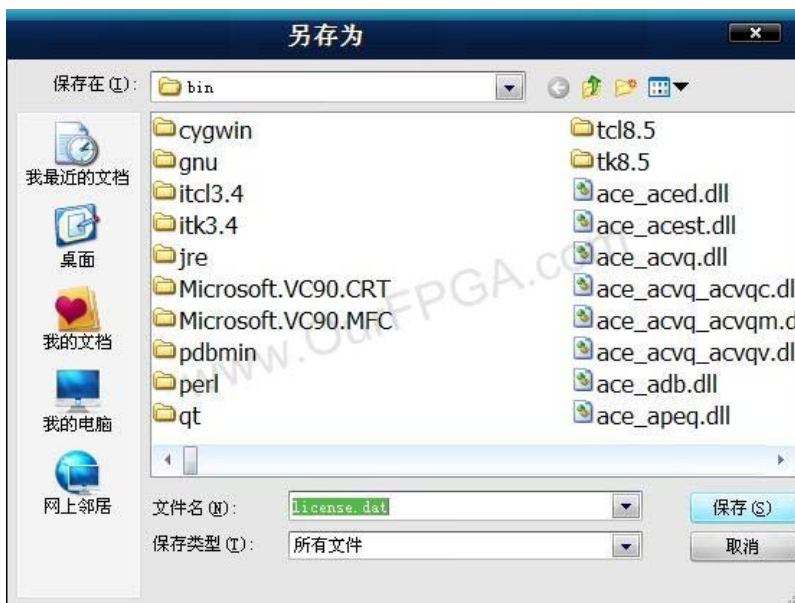


图 4.22 保存 license 文件

把 license.dat 用记事本打开，然后将里的 XXXXXXXXXXXX 用您老的网卡号替换。在 Quartus II 11.0 的 Tools 菜单下选择 License Setup，下面就有 NIC ID，如图 4.23、图 4.24、图 4.25 所示。有的电脑上有好几个 NIC ID，这时候可以多试试，总有一个是成功的。

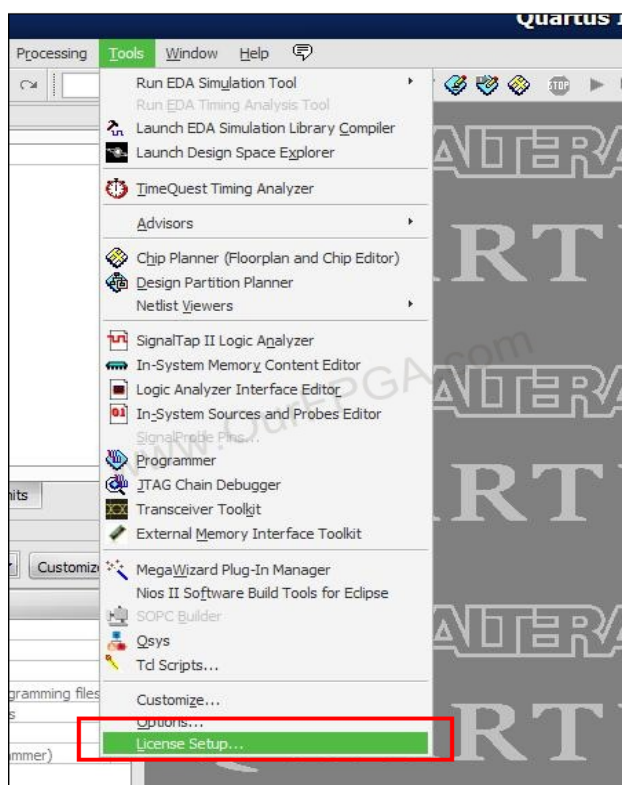


图 4.23 查看 NIC ID

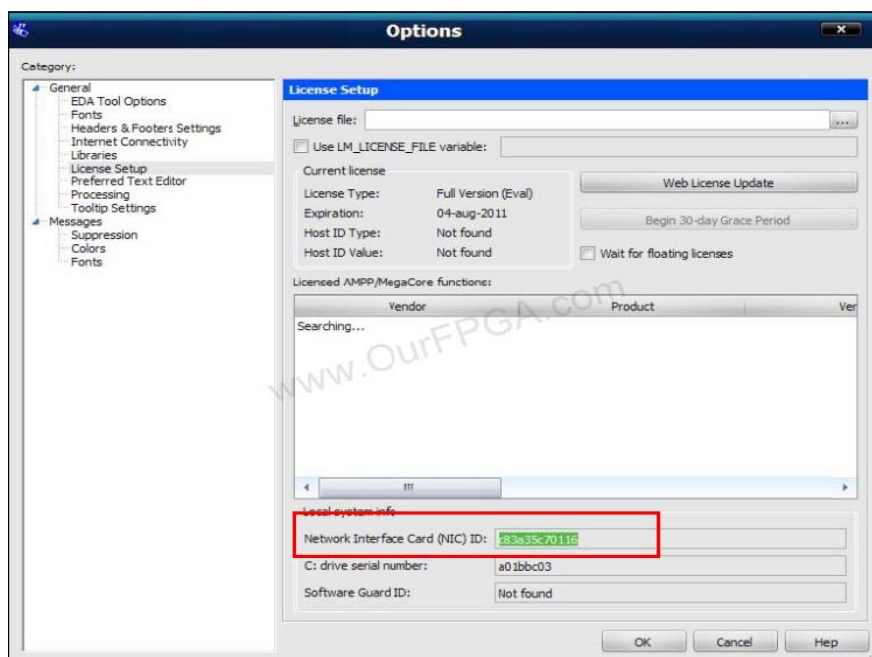


图 4.24 查看 NIC ID



图 4.25 修改网卡号

在 Quartus II 11.0 的 Tools 菜单下选择 License Setup，然后选择 License file，最后点击 OK。
 注意：license 文件存放的路径名称不能包含汉字和空格，空格可以用下划线代替。见图 4.26 所示，看到没有，下面已经是注册成功的了！注意，有的电脑是双网卡的，是有两个 NIC ID，只用复制一个去破解即可，有可能其中一个无法破解成功，那么请您再换另一个 ID 试。一定可以成功的。

此软件在 Windows XP 和 Windows 7 的 32/64 位操作系统下都验证过了，没有问题！
 Windows Vista 32/64 因为微软都放弃了，所以没有验证，理论上应该可以正常使用。

此软件已经通过了诺顿测试，在其它某些杀毒软件下，也许被误认为是“病毒”，这是杀

毒软件智能化程度不够的原因，所以只能暂时关闭之。

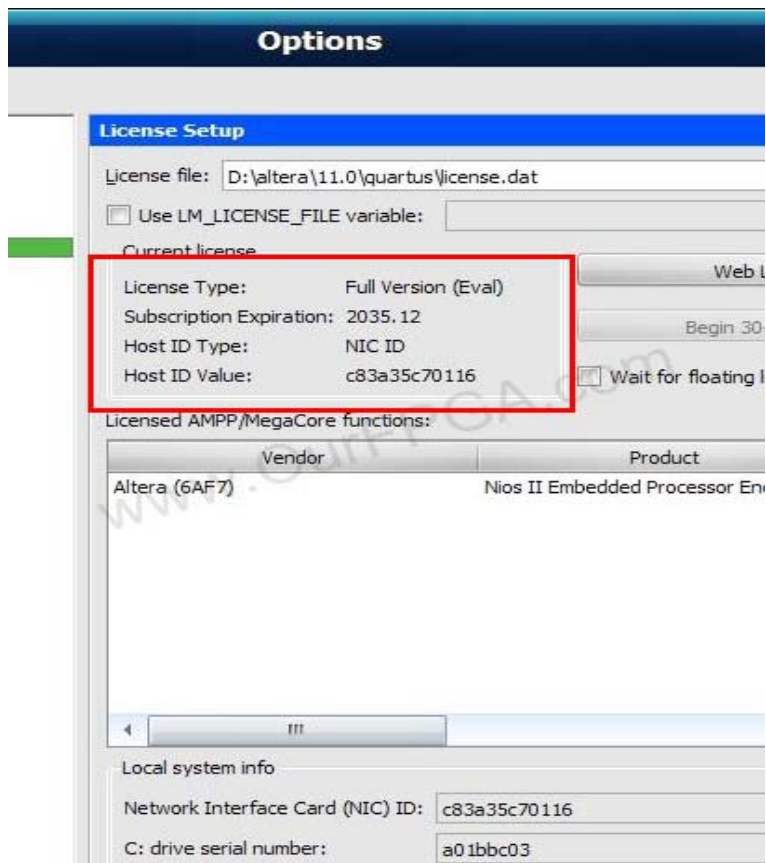


图 4.26 注册成功截图

4.2 Nios 安装方法

睿智 FPGA 开发板本身是完全支持 NIOSII 开发的，但作者认为初学者开始应该以逻辑控制为主要学习任务，NIOS 暂时可以不学。就作者本人接触到的情况来看，NIOS 在工程应用中用到的还比较少，原因有很多，我们这里不做讨论。我们一直认为，在有限的生命中，要抓紧时间学习有用的技能，既然 NIOS 用的都不多，请问干嘛要花那么多时间搞？学有余力的，可以学一下，Nios 安装很简单，装过 Quartus 后，这个就很简单了，这里不多赘述。

第五章 USB blaster 安装

5.1 Win XP 系统下安装说明

插上 USB 下载线后，等待提示发现了新的硬件，如图 5.1 所示：



图 5.1 USB blaster 安装

如果您的电脑一直没反应，可以更换一个 USB 口试试，最好是主机箱后面的 USB 口，不要用主机箱前箱面板那种外接口，建议使用 usb2.0 的口。参见图 5.2，选择“从列表或指定位置安装”。



图 5.2 USB blaster 安装

然后按图 5.3 所示，点浏览，找到相关驱动所在位置，把驱动目录指定 usb-blaster 驱动的目录，请参考图 5.4 所示。

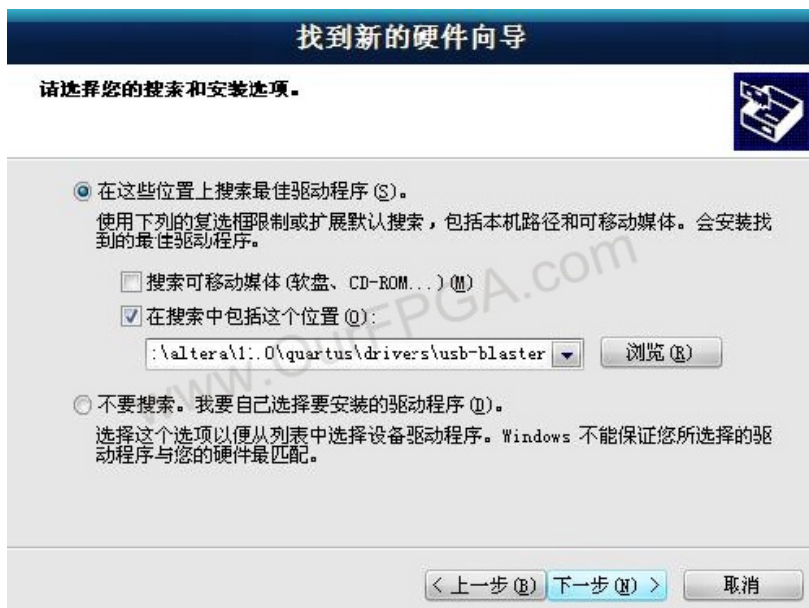


图 5.3 USB blaster 安装

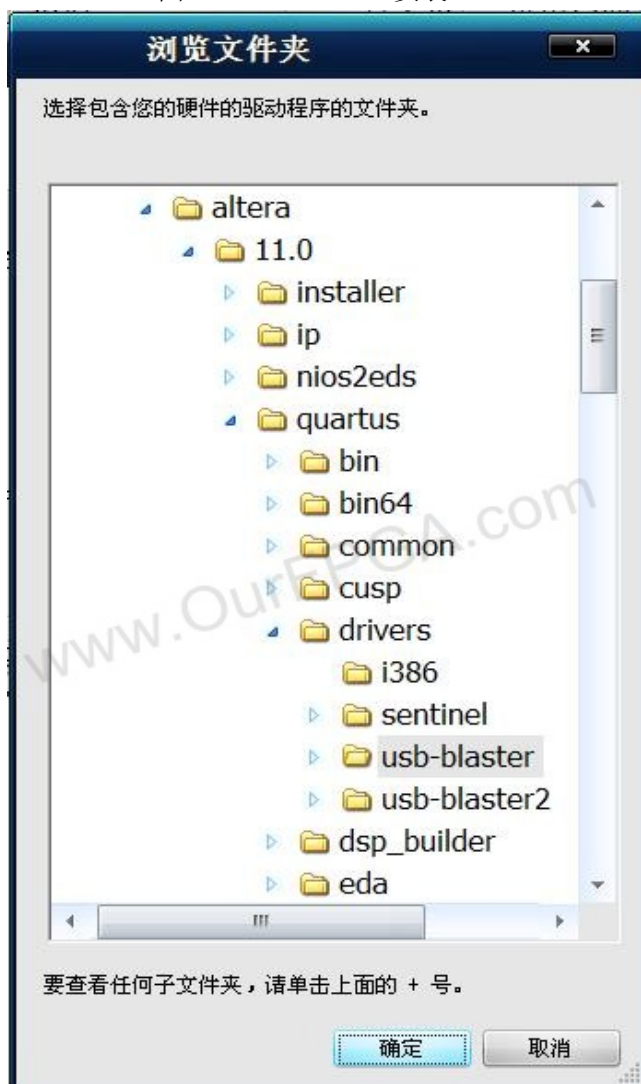


图 5.4 USB blaster 安装

然后点确定，开始安装驱动文件了，见图 5.5 所示。安装完成后，提示硬件安装成功，见图 5.6，图 5.7 所示。安装完后最好能重启下电脑。



图 5.5



图 5.6



图 5.7

5.2 Win7 系统下安装说明

Win7 系统下安装时，插上 usb blaster 后，有可能会提示如下：



有的电脑不提示，不管怎样，我们按下面步骤来操作即可：

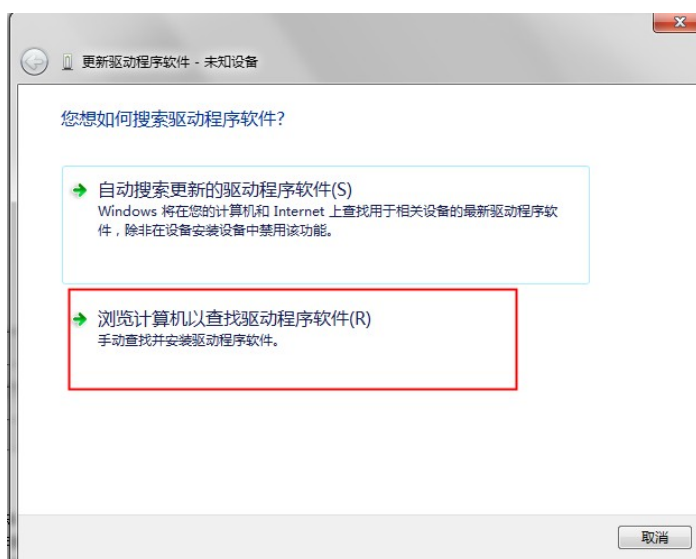
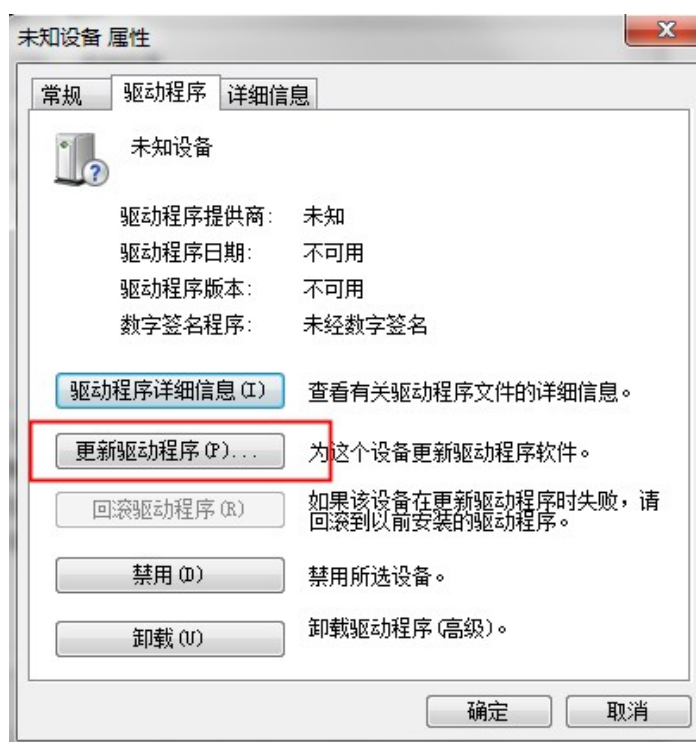
1. 第一步：打开“开始菜单”找到“设备和打印机”打开

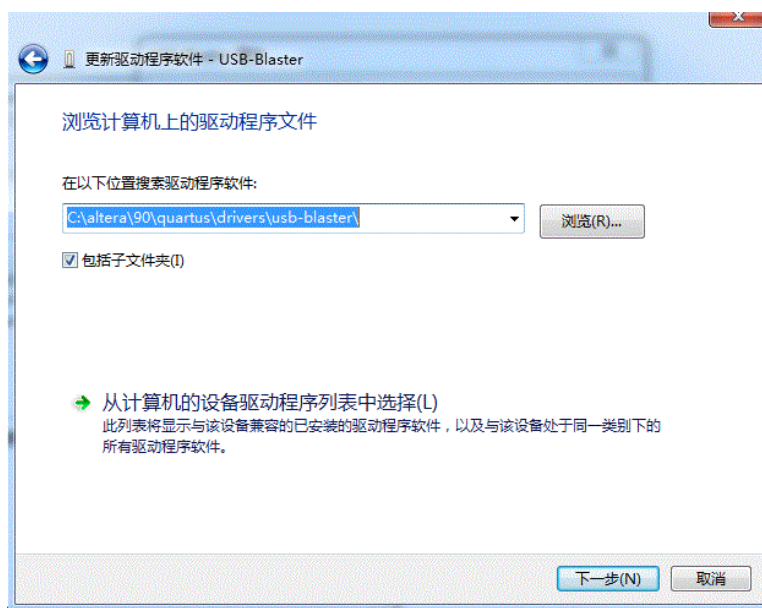


2. 点击未指定的 USB Composite Device，也有的叫“未知设备”，见红色框：



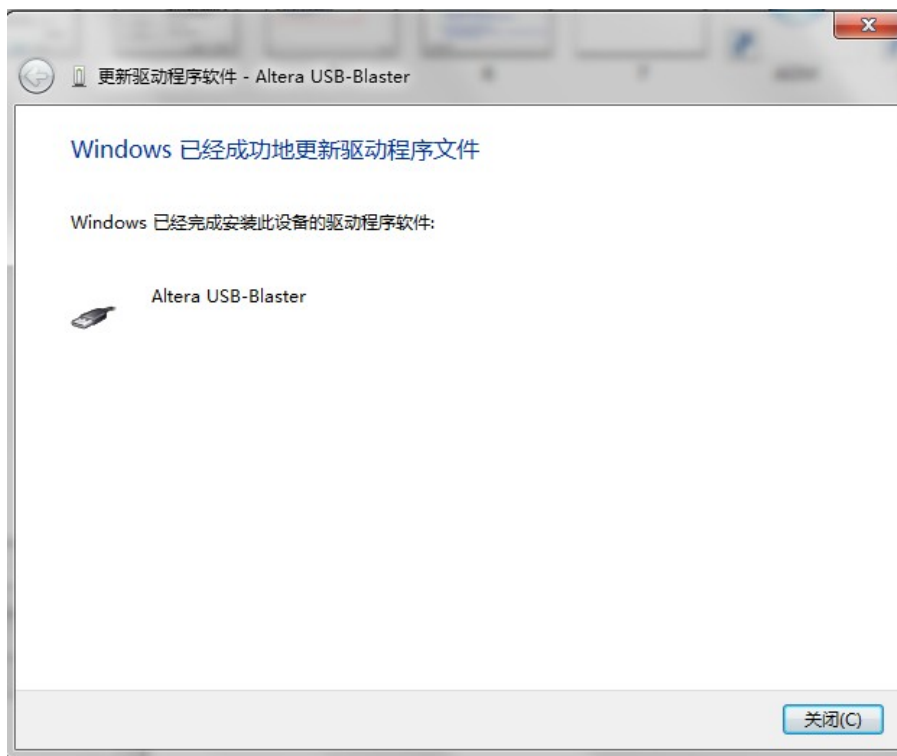
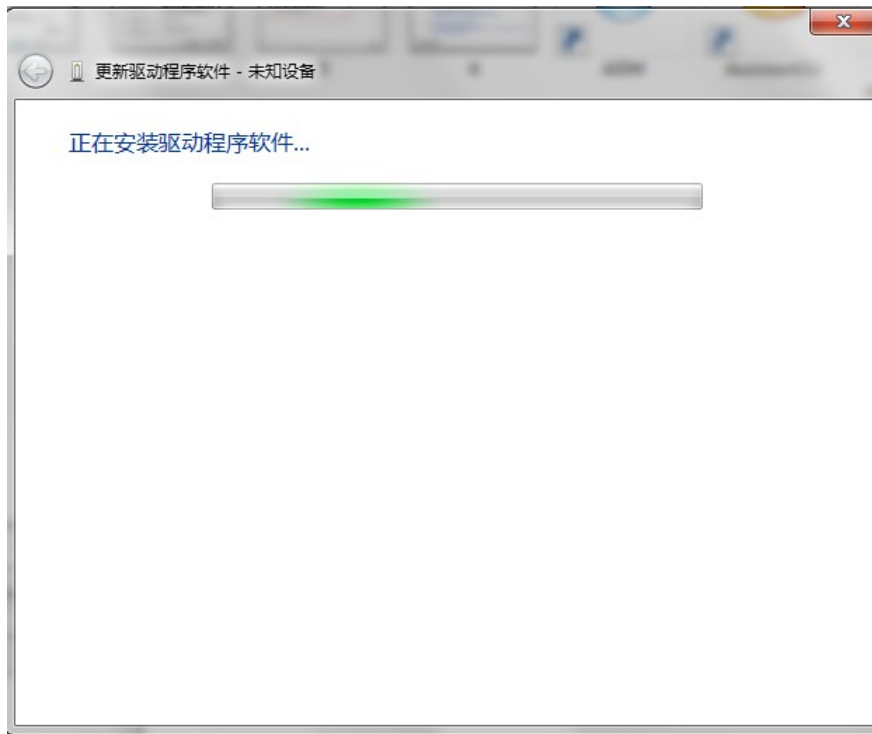
3. 双击打开“未知设备”，或是右击，点硬件>属性>驱动程序>更新驱动程序





到这一步，要找到驱动程序所在目录，根据您的机器上的情况选择一般，驱动程序在 Quartus 文件夹下的 driver>usb-blaster 里，点击下一步





到此，驱动安装成功。

第六章 USB 转串口线驱动安装

现在很多个人电脑都不带串口了,所以如果要做串口实验(开发板上的串口与电脑通讯)的话,就无法完成。USB 转串口线就是来解决问题,USB 端接电脑主机,串口端接开发板。

我们提供的为: HL340 型 USB 转串口线 (COM) 九针串口。如下图示:



安装驱动:

- 1) 插上 USB 下载线
- 2) 等待提示发现了新的硬件, 如下图示:



然后出现下图提示, 注意选择“从列表或指定位置安装”



下一步, 浏览文件夹找到光盘内附带的 340 驱动, 如下图示, 点确定



开始安装, 过一小会儿提示安装成功, 就可以正常使用了。步骤都很简单。

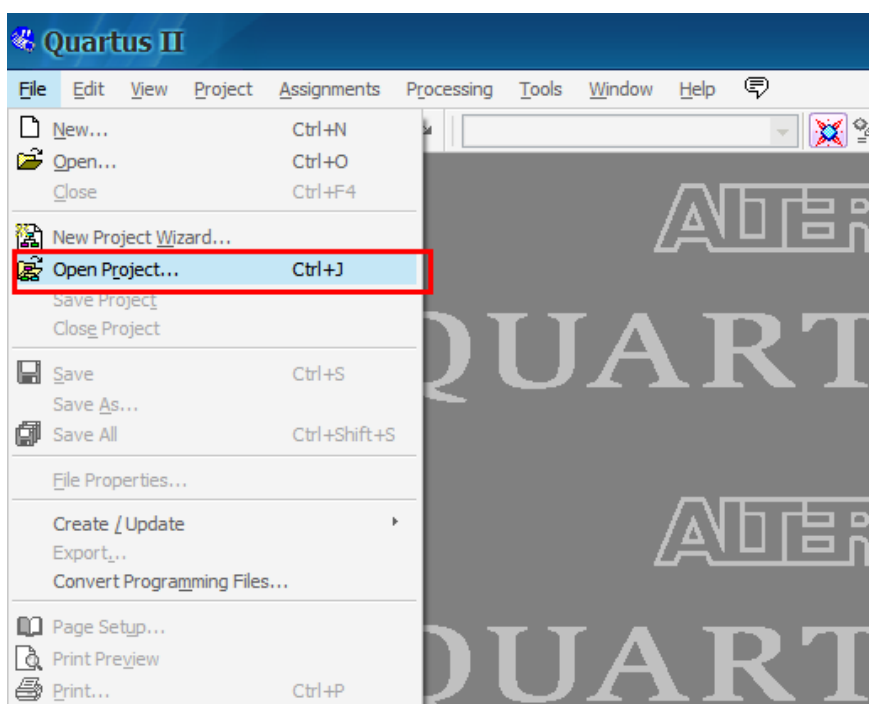


第七章 例程说明及使用、下载方法

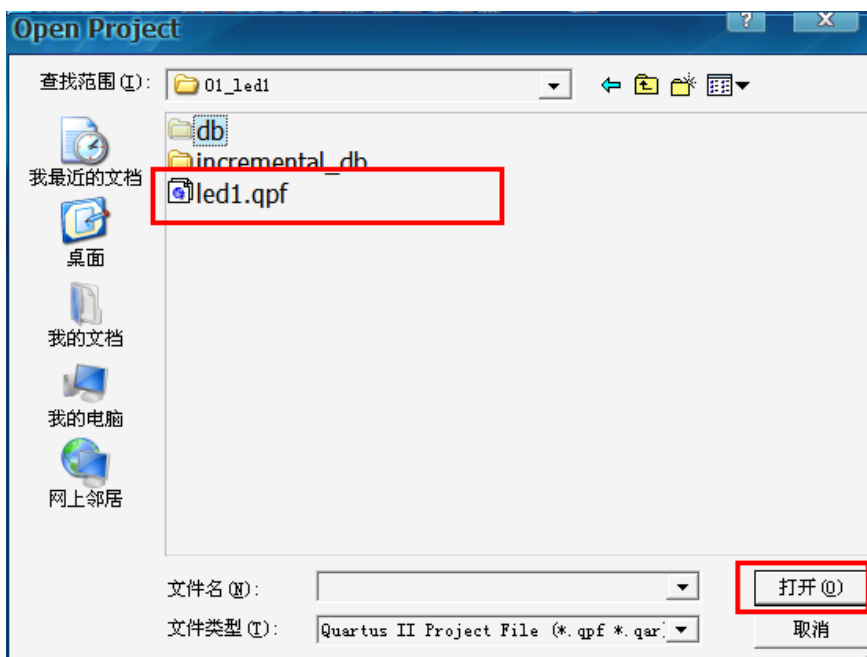
7.1 打开一个例程

开发板收到后，大家可以试着用一下光盘里配套的例程，熟悉一下相关的操作。打开一个已有的 FPGA 工程很简单：拷贝配套光盘中的实验例程包到硬盘上，解压，**注意请放在英文目录下，并且不要有空格。**

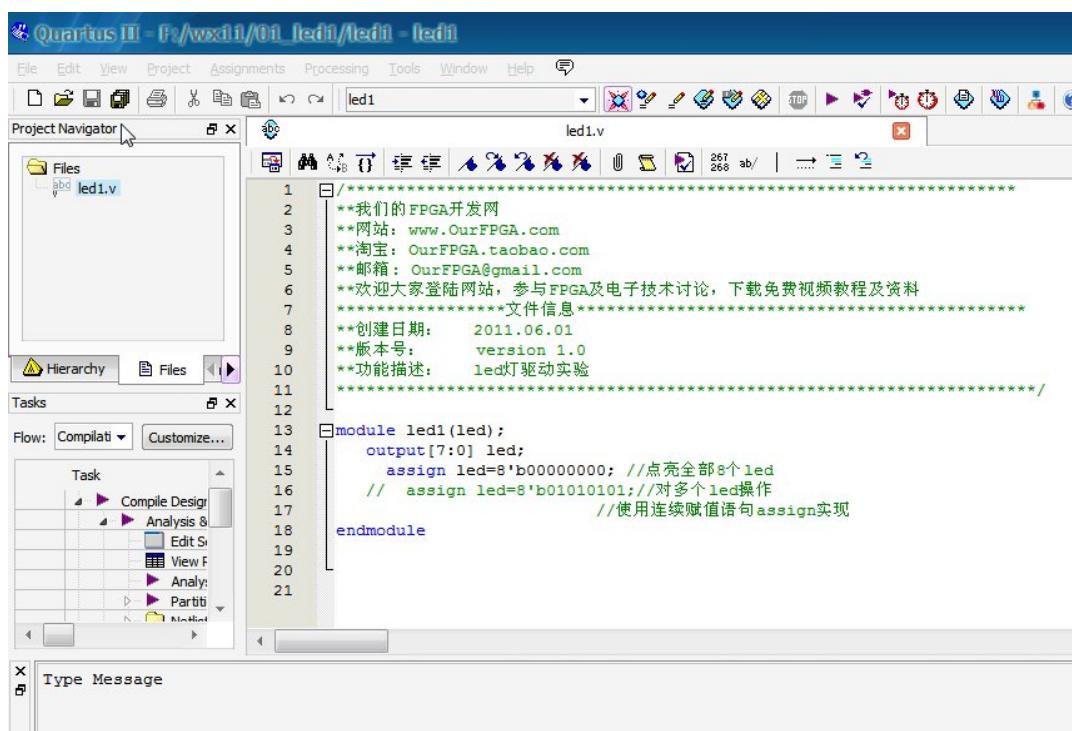
打开 QuartusII 11.0 软件，在菜单 File 里选择“Open Project”



找到实例工程包，打开 01_led1

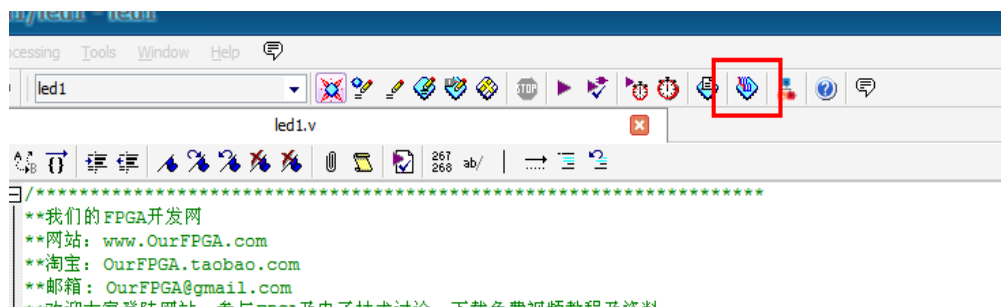


打开工程文件后，如下图所示：

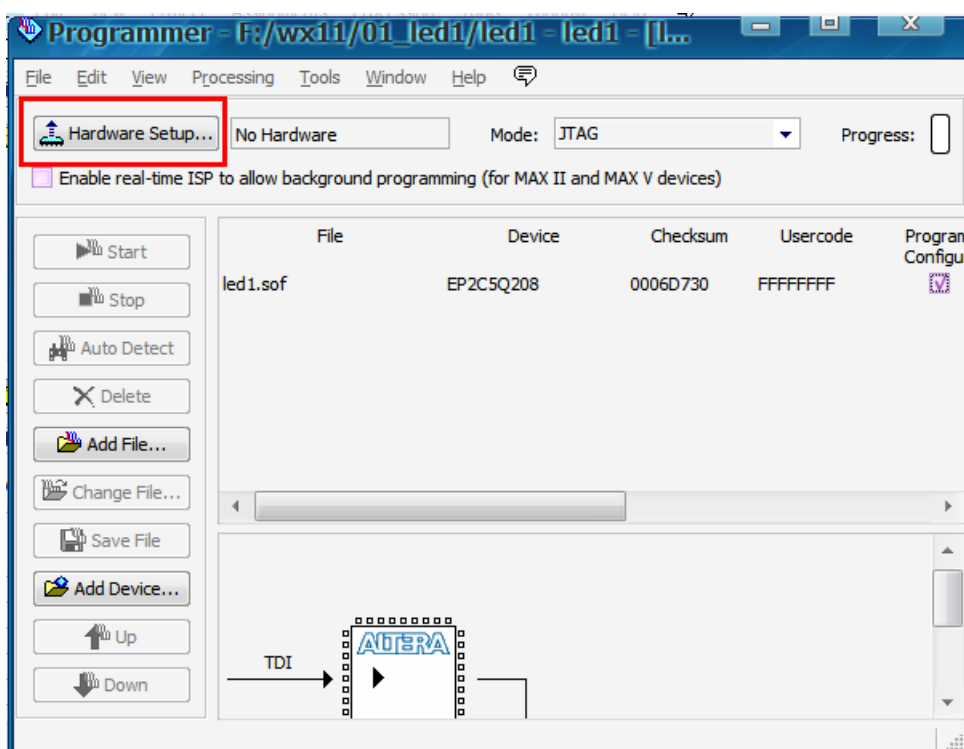


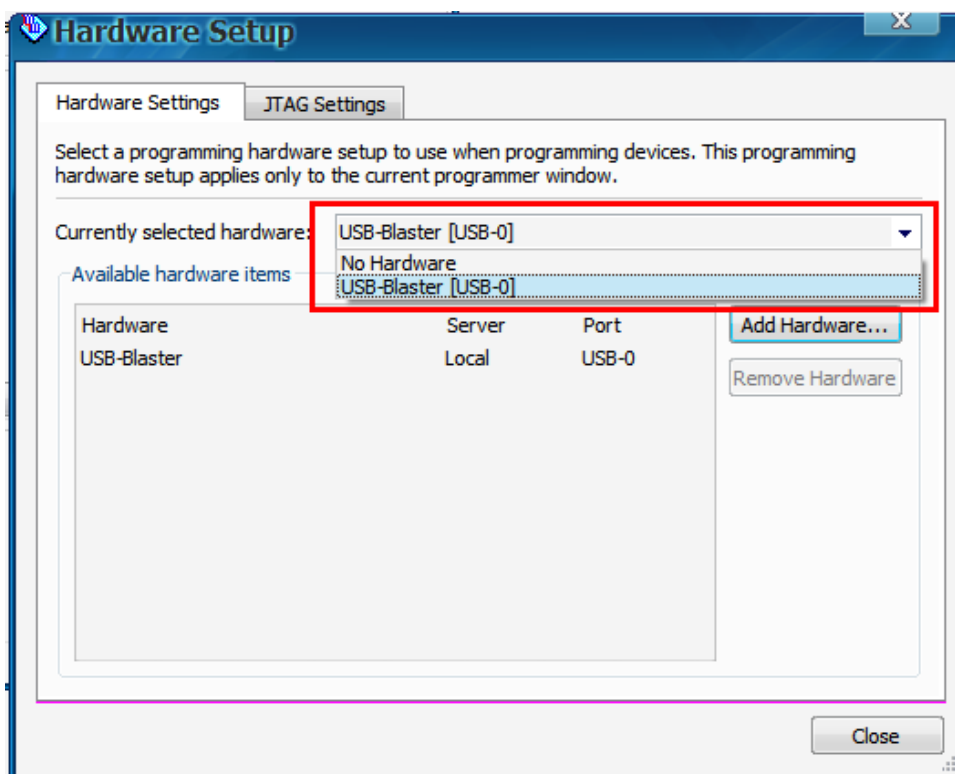
7.2 用 JTAG 下载 SOF 文件

SOF 文件是用 JTAG 口下载的，下载后掉电丢失的，主要用于程序调试过程，平时学习，使用这种模式。下载也很简单，点击工具栏上的编程选择，如下图红色框所示：

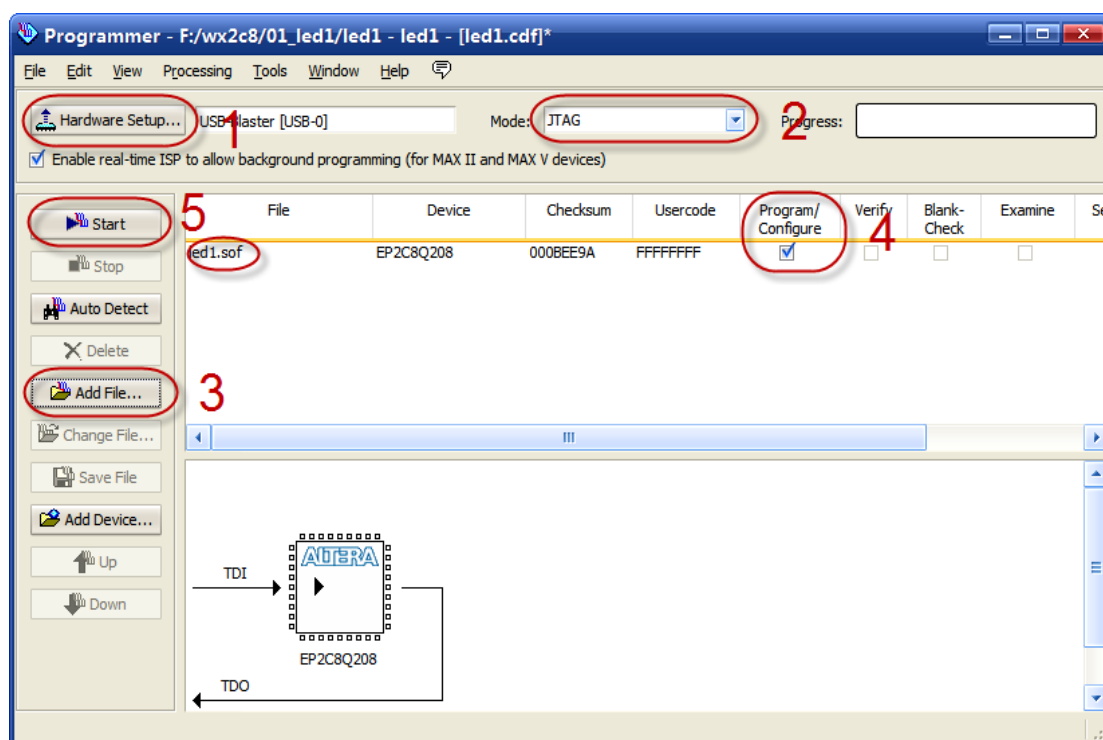


点击下图示的“Hardware Setup”,然后在选择下载硬件为：“USB blaster”。





关闭后，回到编程窗口，注意下图中各个红色框选择，



上图重点地方打了红框，并编号 1，2，3，4，5，依次如下：

- 1.选择下载器硬件设备，前面已经设好；
- 2.选择下载模式 JTAG，同时 usb blaster 下载线接在开发板的 JTAG 口上

3.选择下载文件，点击 ADD File，把.sof 的文件加进来；

4.Program configure 打√；

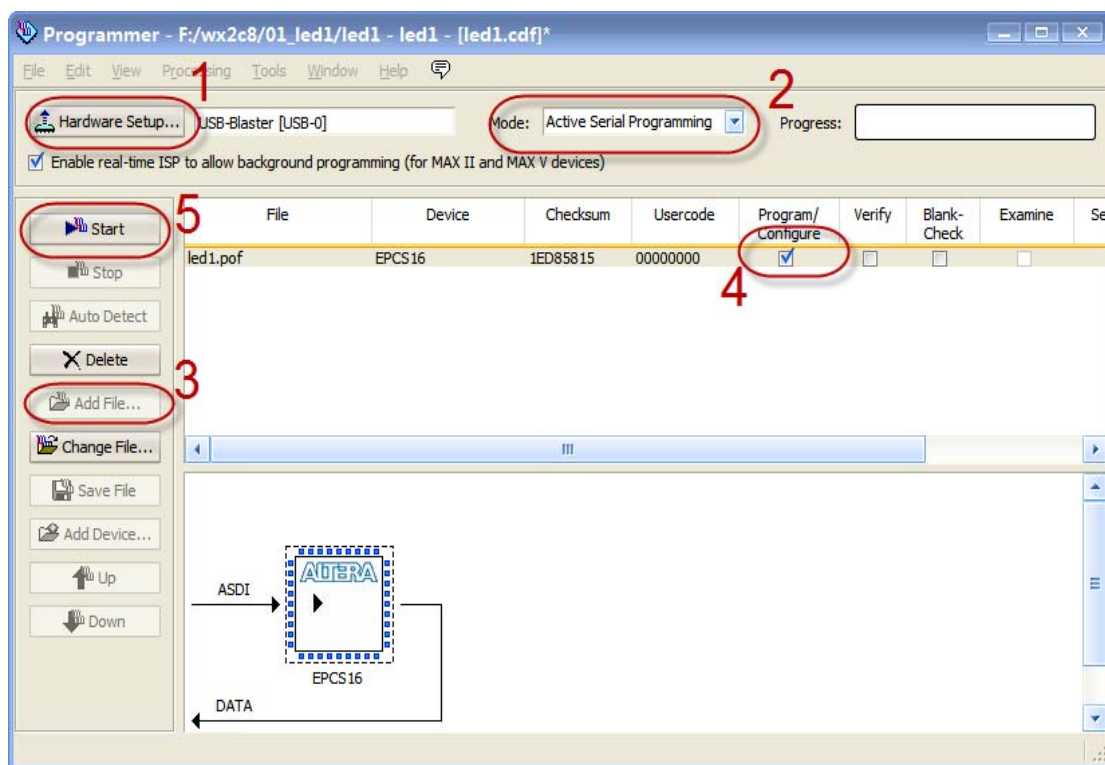
5.点 START 下载；

点"start"下载，开发板上配置指示灯会闪一下，然后程序就开运行了，8 个 LED 灯同时点亮了。

7.3 用 AS 下载配置 EPCS 芯片

AS 下载是固化程序的，代码调好，想固化在电路里面时，就用这种模式，掉电后是不丢失的，睿智 FPGA 开发板出厂前都严格测试过，最后测试一步一般烧写一个流水灯程序到 EPCS 芯片里，所以收到开发板上电后，会看流水灯效果。您可以烧写自己的程序，或擦除。

具体下载方法如下图所示：



上图重点地方打了红框，并编号 1，2，3，4，5，依次如下：

1.选择下载器硬件设备；

2 选择下载模式 AS，同时 usb blaster 下载线接在开发板的 AS 口上

3 选择下载文件，点击 ADD File，把.pof 的文件加进来；

4 Program configure 打√；

5.点 START 下载;

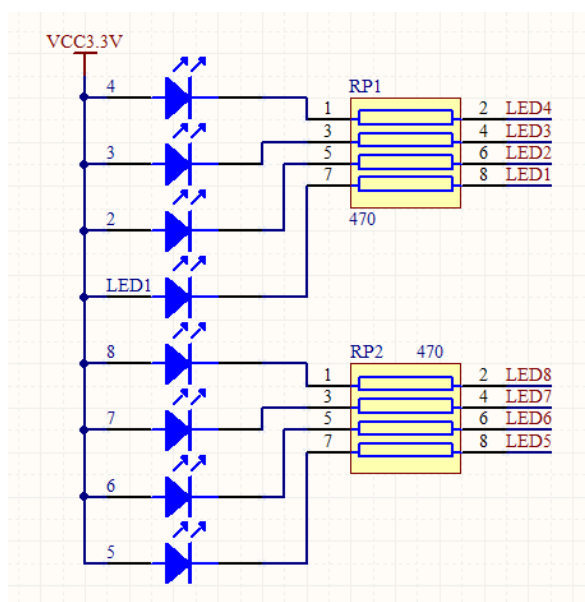
注意: 由于市面上各种下载器不同, AS 下载完后, 有的下载器需要先关掉开发板电源, 将下载器从 AS 口上拔掉, 重新给开发板上电, 才能运行。也有的下载器在下载完成后就运行。

7.4 新建工程操作说明

学习 CPLD、FPGA 知识最重要的是在掌握基本理论的基础上, 以实践和动手为主, 下面将以 LED 驱动为例, 以尽量详细的写法将 CPLD 开发的基本流程写清楚, 熟悉使用 QuartusII 软件, 通过实战来理解和掌握这些知识和技巧。

注意: 本章节内容和截图均以睿智 FPGA 开发板为例, 实际上新建工程和具体操作过程, CPLD 和 FPGA 是没有什么区别的, 在选择主芯片及引脚分配时, 请灵活修改, 选择与自己板子相符的型号或引脚!

睿智 FPGA 开发板底板上共有 8 个发光二极管 LED1-LED8, 我们先完成这样一个任务: 是要间隔点亮这 8 个发光二极管。硬件原理图如下:



可知: 如果要点亮这些 LED, 只需要把与其相连接的 FPGA 管脚输出低电平“0”就可以实现这个功能了。如果我们利用连续赋值语句 assign 实现的话, 写程序代码如下:

```
module led_light(led);
```

```
    output[7:0] led;
```

```
assign led=8'b10101010;
```

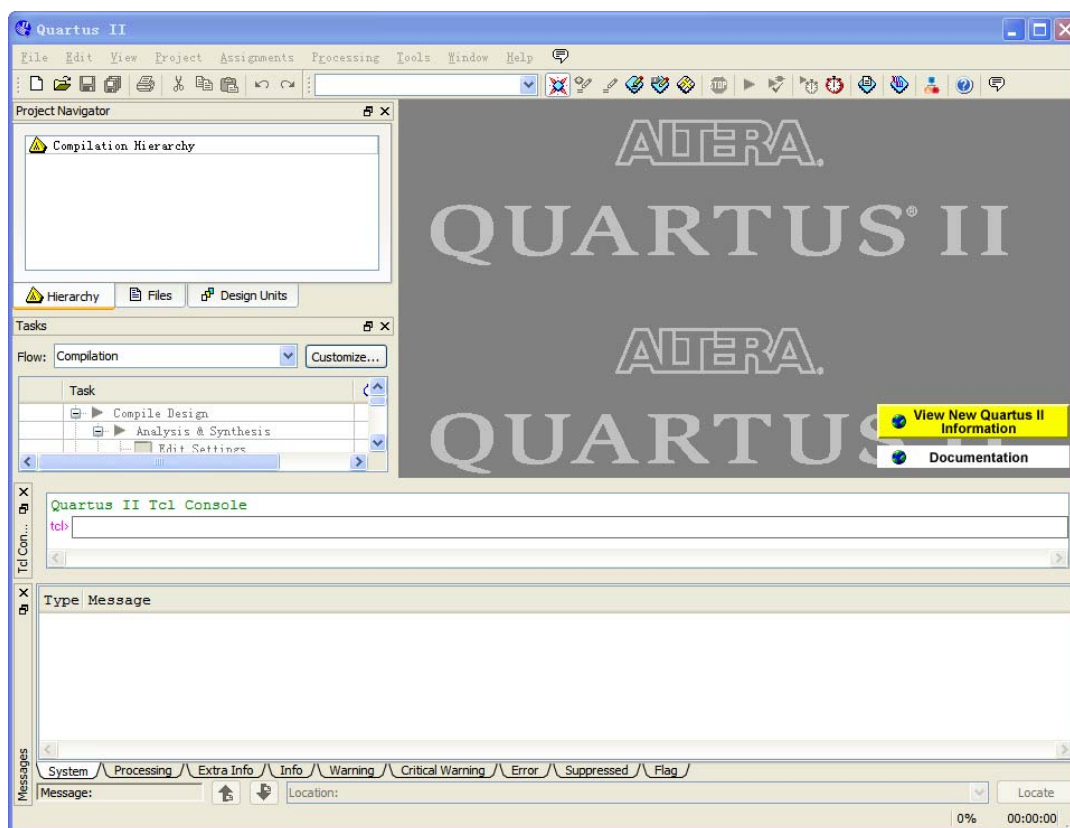
```
endmodule
```

下面我们就以这个简单的例子进行新建工程的全部过程，对于完全新手来说很有参考价值，希望大家都能熟练掌握！

启动Quartus II中任何一项设计都是一项工程，在工程设计过程中，会产生许多仅扩展名不同的同名文件，放在同一个文件夹下，便于统一管理。所以，不同的设计项目最好放在不同的文件夹中。在本例中，建立文件夹D:\FPGA_study作为工作库，以便将设计过程中的相关文件存储于此。

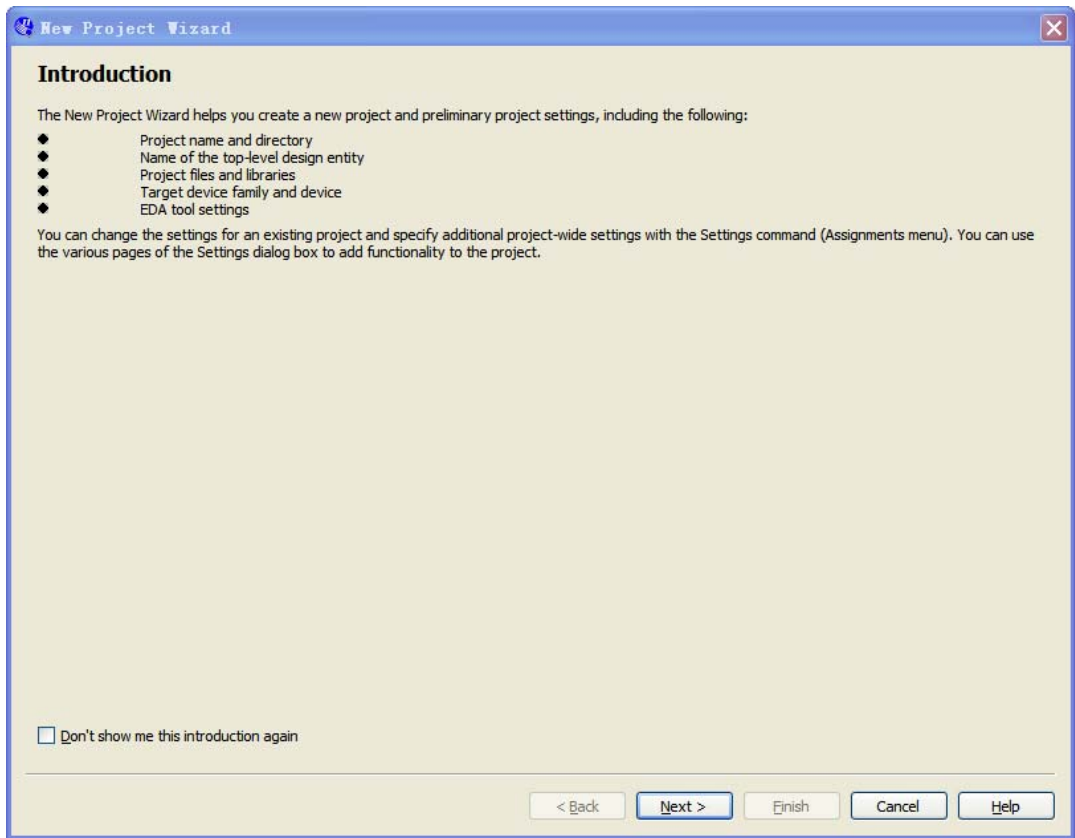
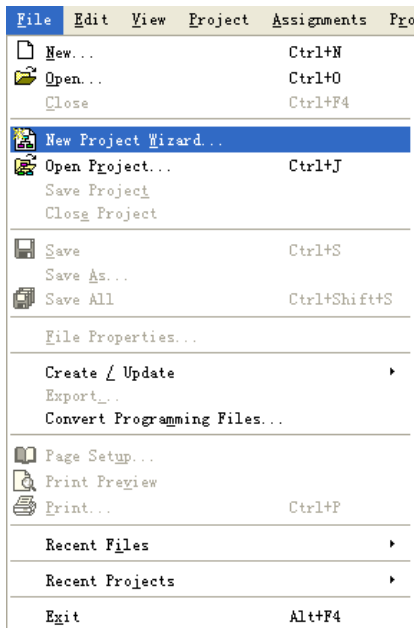
启动Quartus II：

双击桌面上的Quartus II图标或单击开始按钮，在程序菜单中选择Quartus II，可以启动Quartus II。其初始界面如图所示

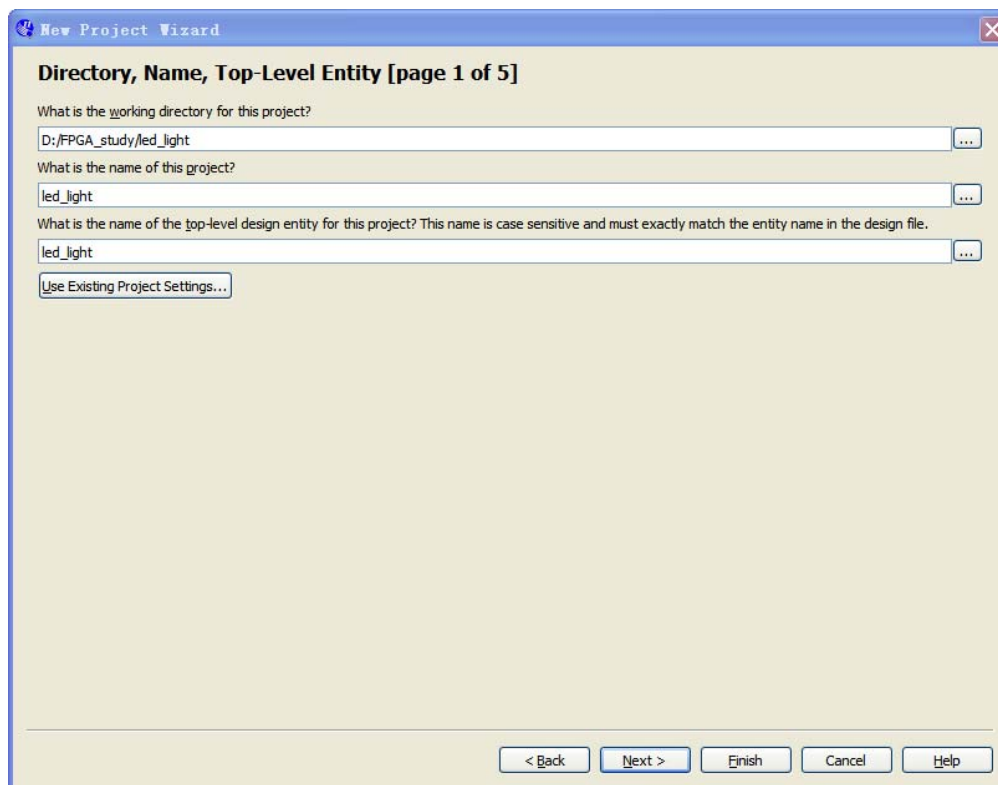


在图中，在 File 标签下点击左键，可打开如下图所示的菜单，新建工程向导点击 File->

New project wizard..新建工程项目 led_light:

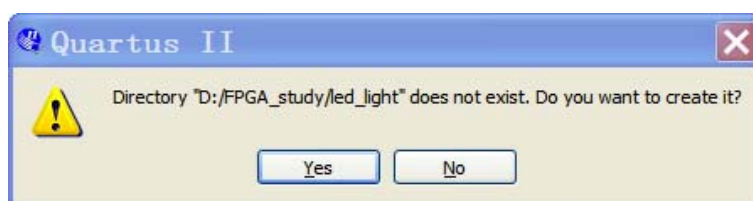


点击 Next, 进入下步, 进入基本设置对话框

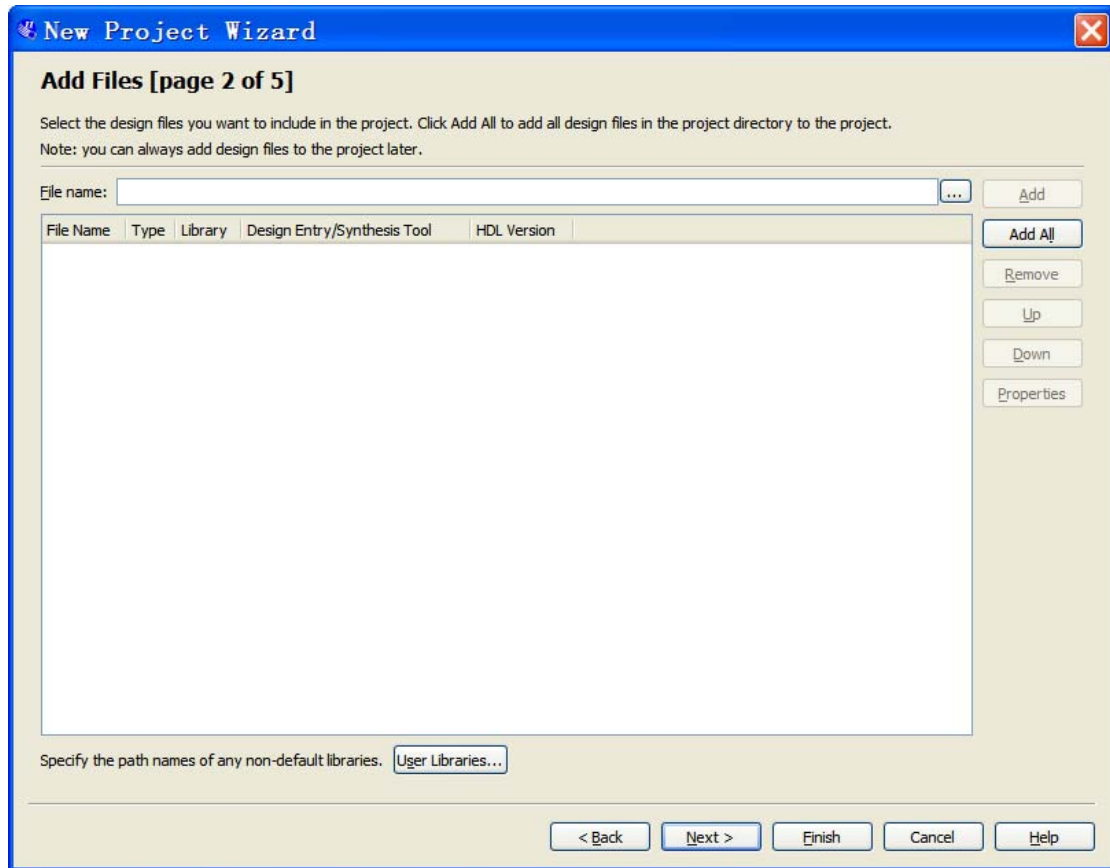


在第一栏的输入框中输入工作库文件夹的地址。第二栏的输入框要求输入该工程的名称，一般可以用顶层文件的名称作为工程名称，这里我们写led_light。第三栏的输入框要求输入顶层设计文件实体的名称，实际上，您会发现，您在写第二栏时，第三栏已经由系统自动完成了，也是led_light。完成后，直接点击Finish。

因为还没有创建文件夹，Quartus II 弹出一个对话框，询问是否新建所需文件夹，如下图所示。

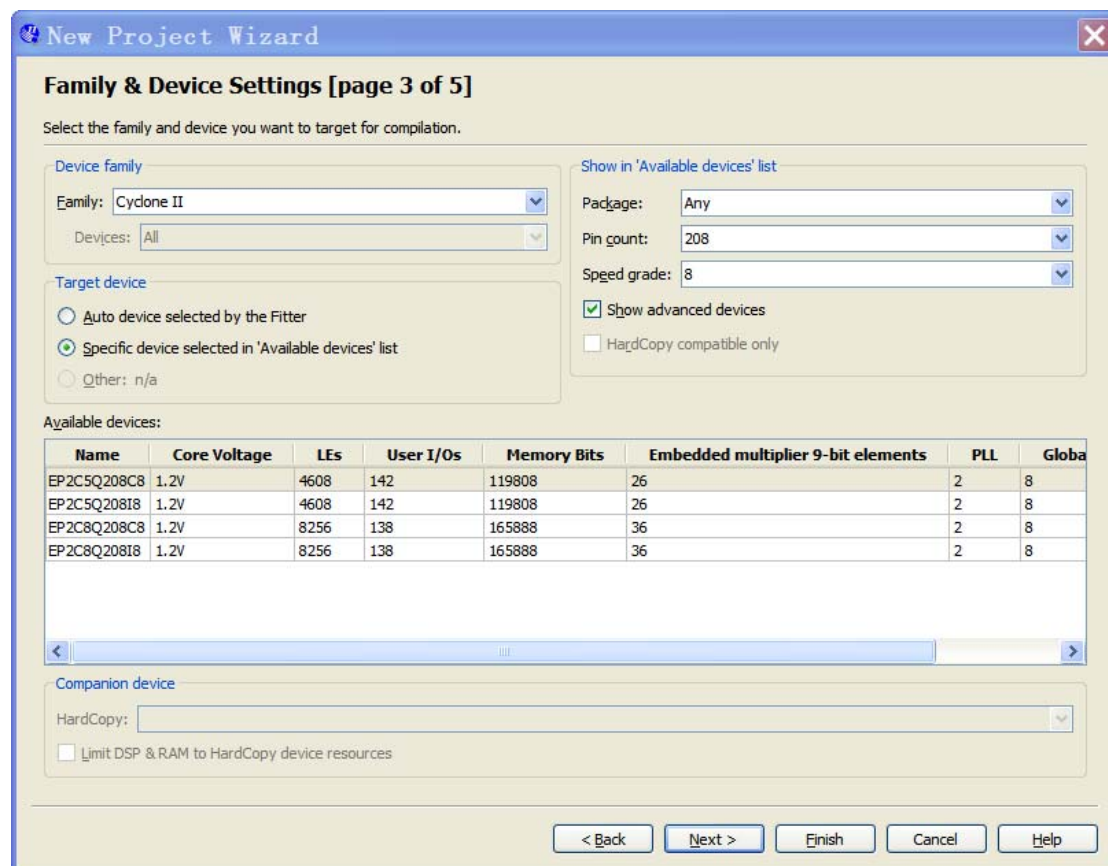


点“是”单击 Yes, 将会引出下图所示的窗口。



因为现在还没具体的设计 verilog 文件，我们先点[next]；

下面选择所用的 FPGA 器件以及进行一些必要配置。开发板使用的是 EP2C5Q208C8，（[这里 CPLD 用户请选择 MAXII](#)）点击菜单 Assignments->Device..，出现下面的对话框。



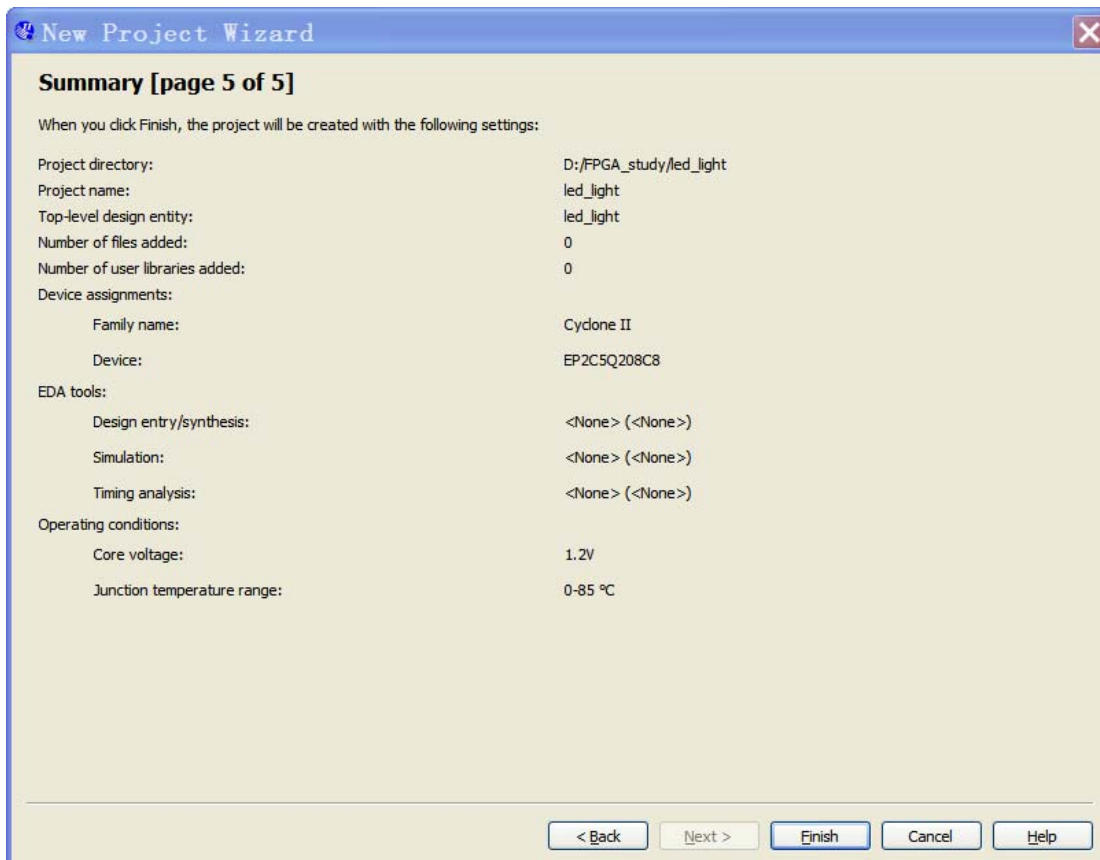
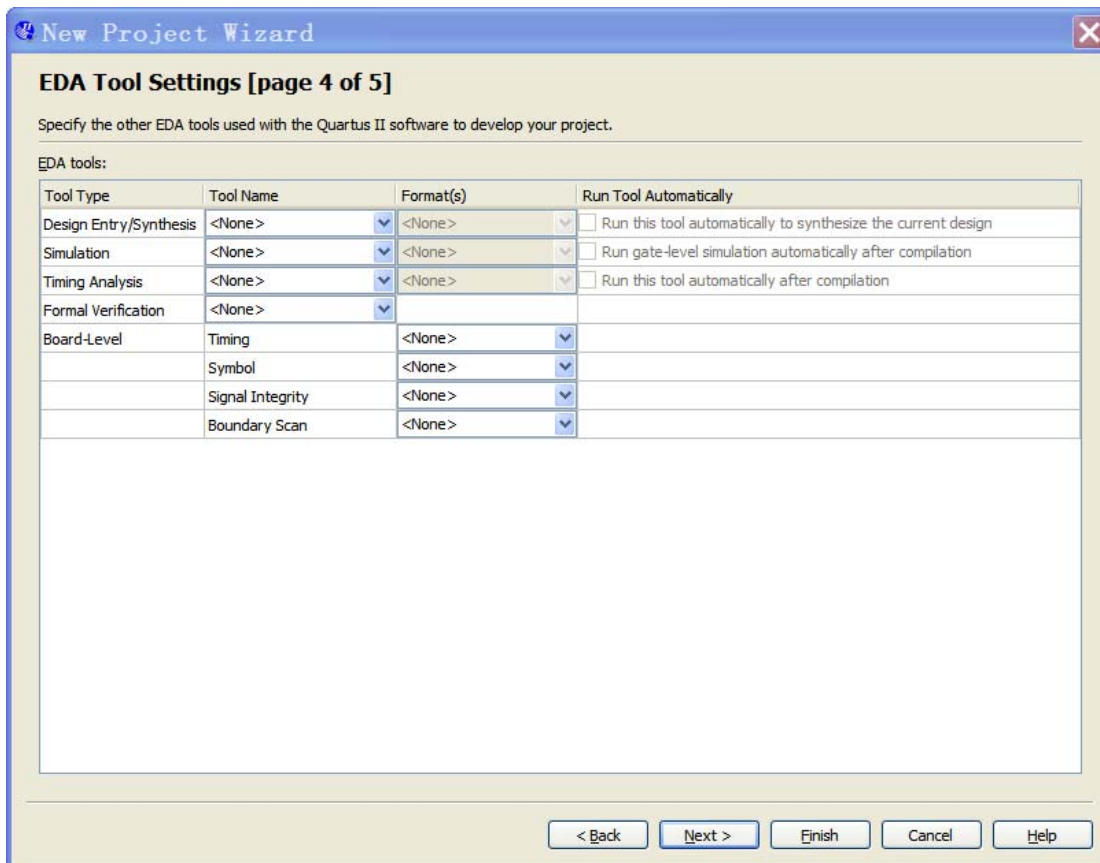
用好几个选项供筛选,我们可以随意按自己的了解选择,比如:（[这里 CPLD 用户请选择 MAXII](#)，针对自己板子的芯片进行选择，都很简单，不再说明）。

EP2C5Q208C8 属于 CycloneII 系统,在 Device Family 选择 CycloneII;

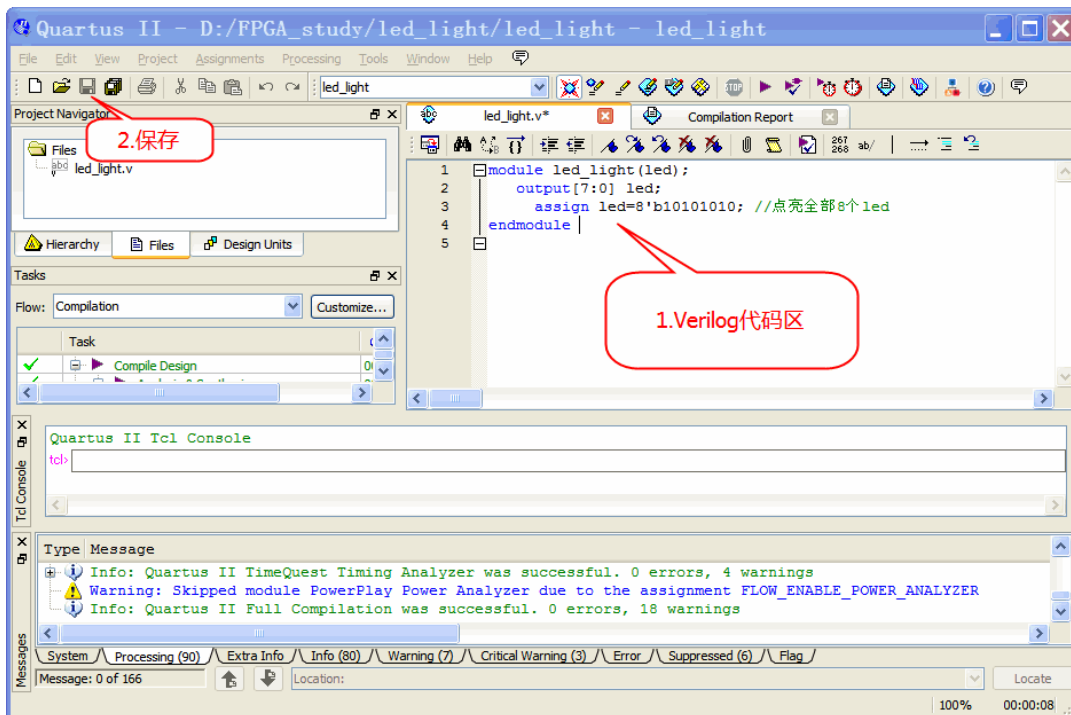
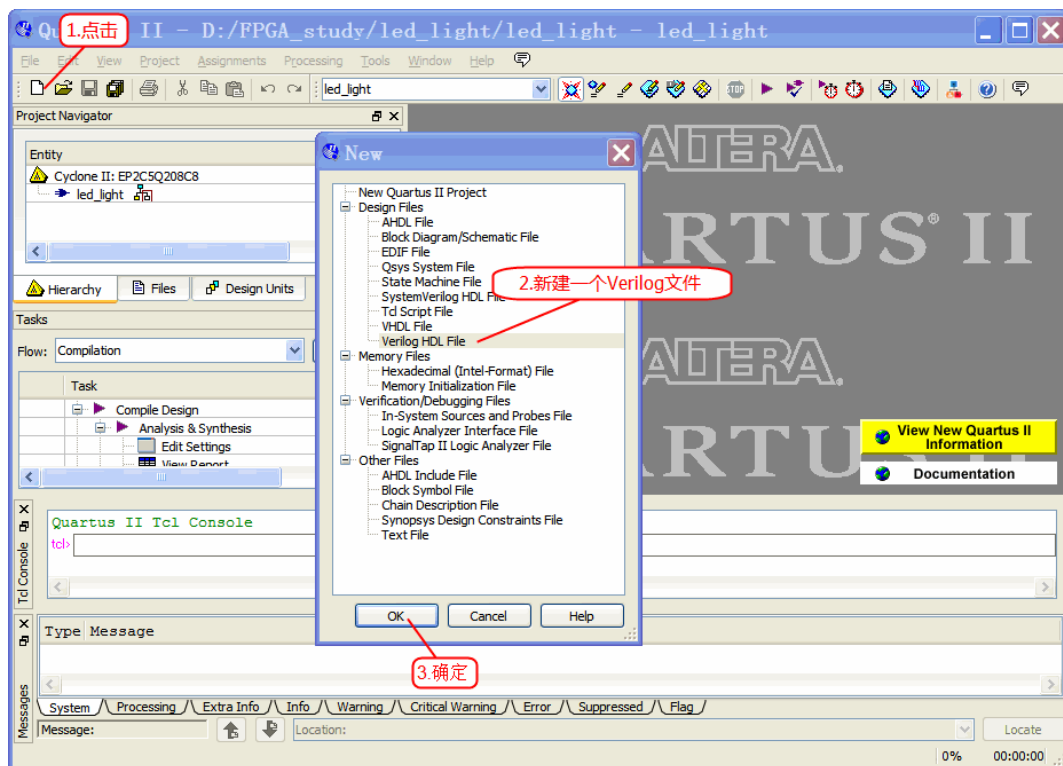
EP2C5Q208C8 引脚共 208 个, Pin count 选择 208;

EP2C5Q208C8 器件速度等级为 8,Speed grade 选择 8;

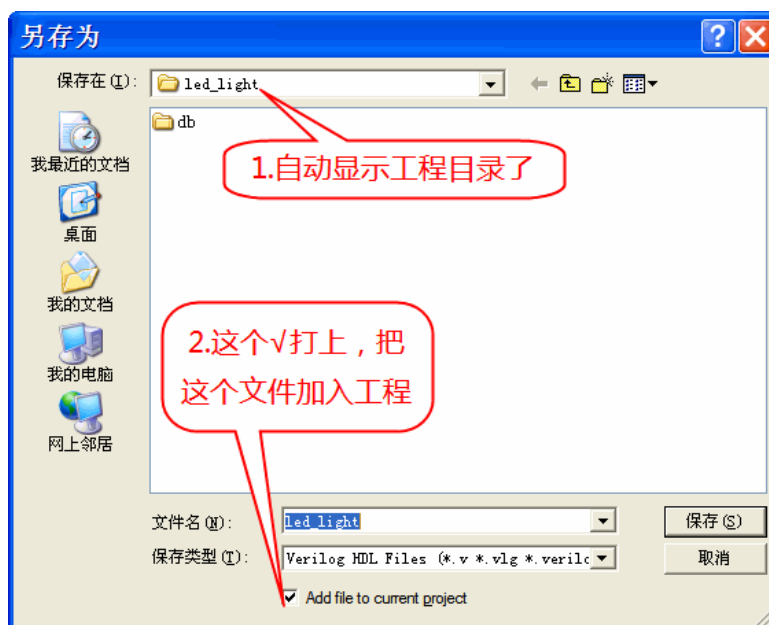
这时候在 Available device 列表已经列出了符合以上条件的器件,这时我们选择 EP2C5Q208C8,点 OK 确定



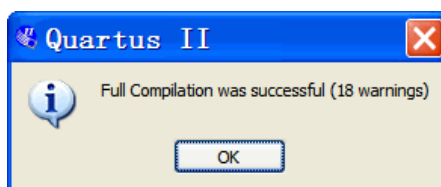
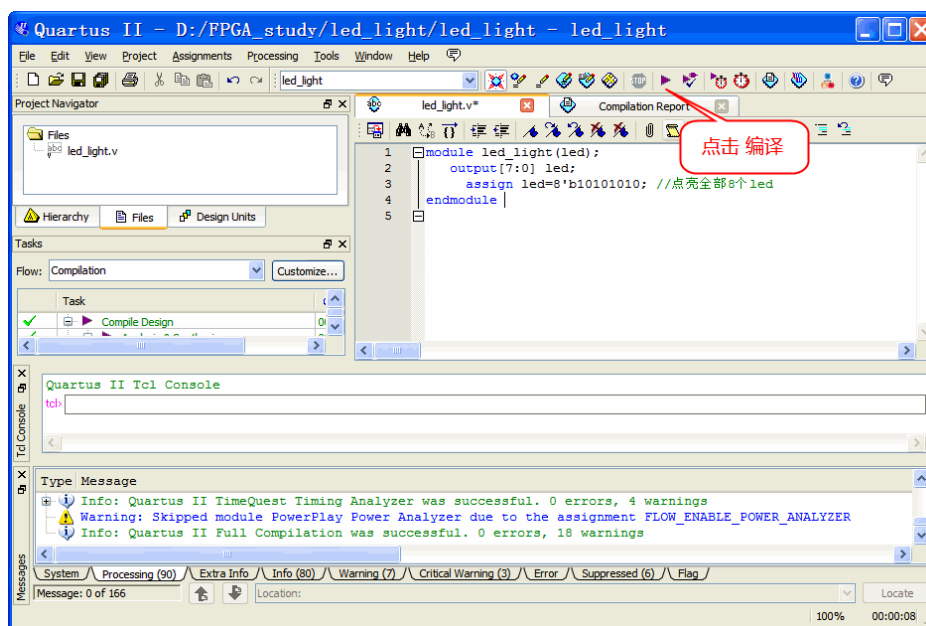
点击 File->New.. 在该项目下新建 Verilog HDL 源程序文件 led_light.v (或者直接点快捷图标, 下图部分), 输入源程序代码并保存。会弹出另存的对话框, 我们发现保存的文件目录已经在 led_light 下了, 文件名也默认为 led_light.v, 不用更改, 保存即可。

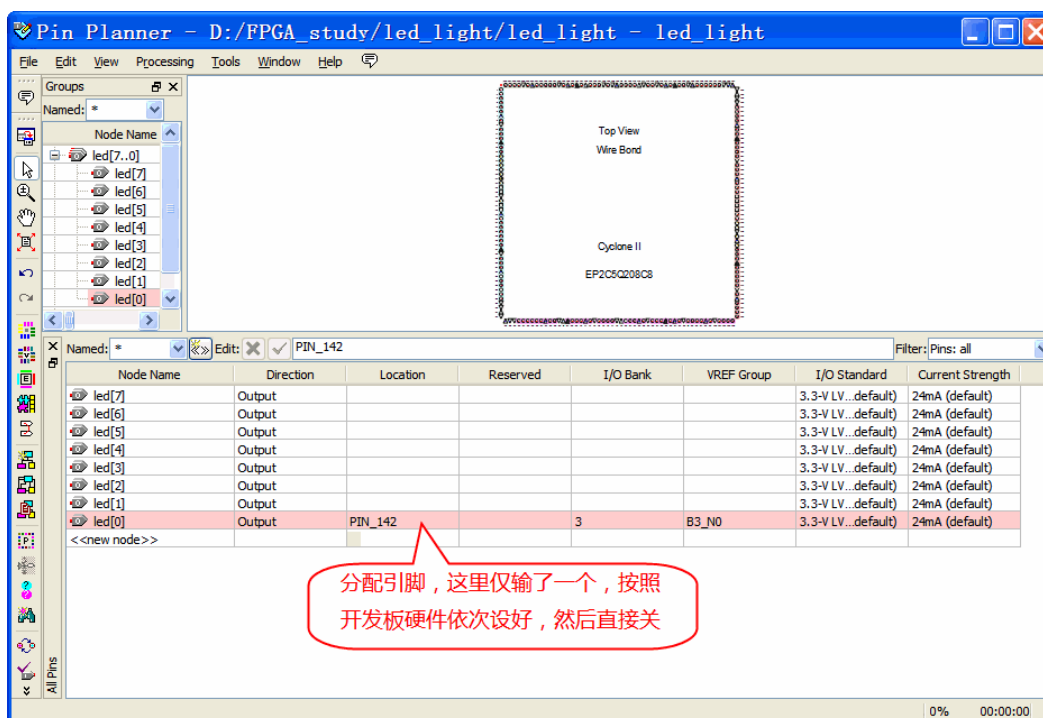
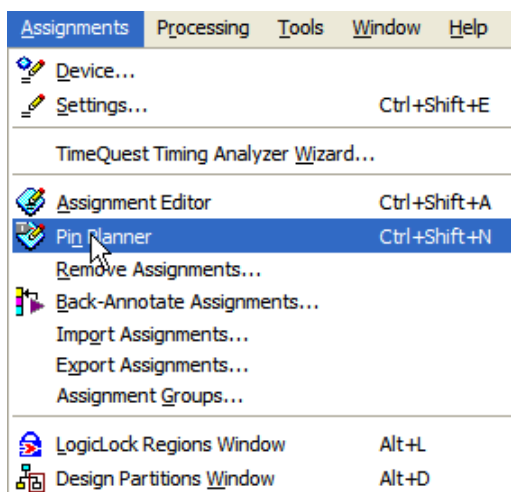


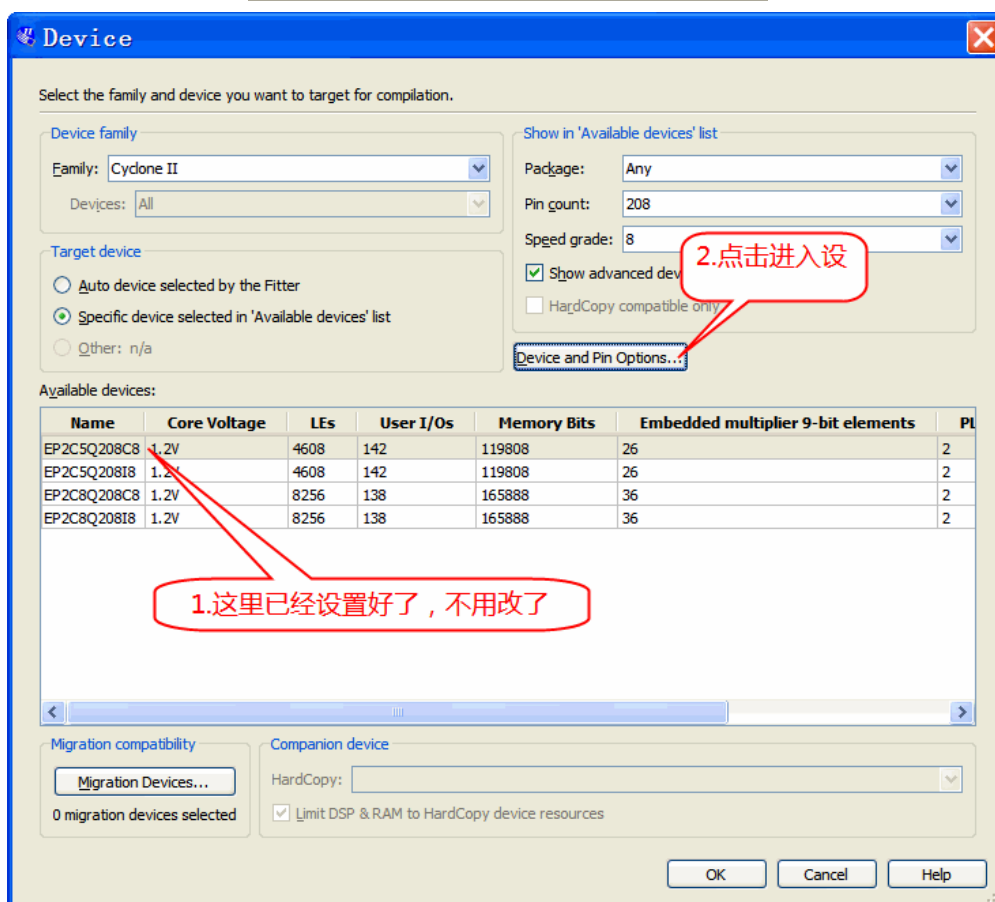
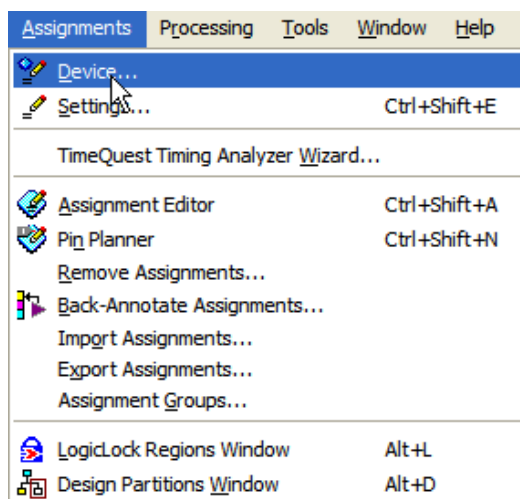
点保存后，出现下图对话框：

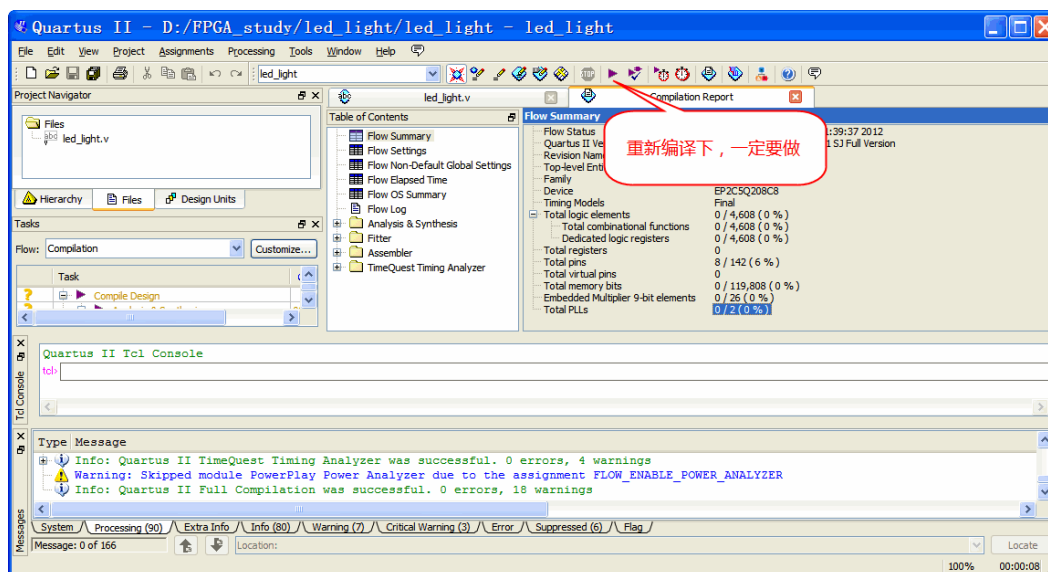
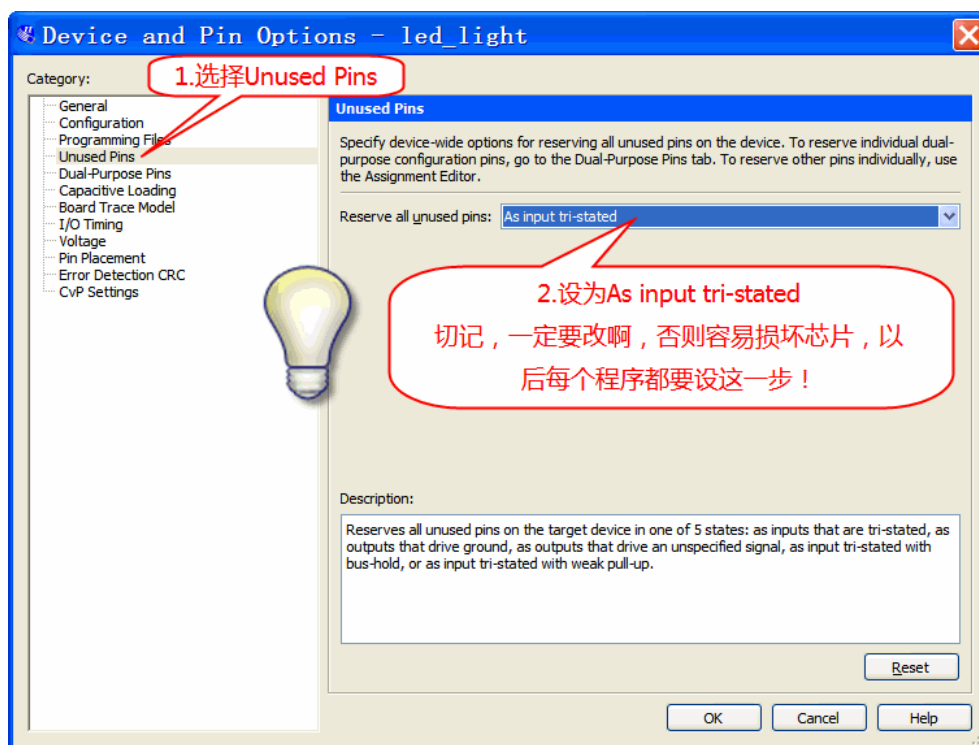


然后点编译，过一会儿，会提示编译成功：









终于完成了，可以看到板子上的 8 个 LED 间隔点亮了原来一切都这么简单，通过这个实验，基本就知道了如何使用 Quartus II 了，开发的流程也有一个比较清楚的认识，也方便进行以后的实验了。

第八章 ModelSim-Altera 6.5 实例指导

[此处默认您已经安装好 quartus11.0 软件。。。>](#)

8.1 Modelsim_Altera_ase 软件安装

睿智 FPGA 开发板配套光盘内提供了 [Modelsim_Altera_ase 的安装包](#)，ase 版本是 altera start edition，即入门版，免费使用的；ae 是 altera edition，需要破解，支持更多功能吧。

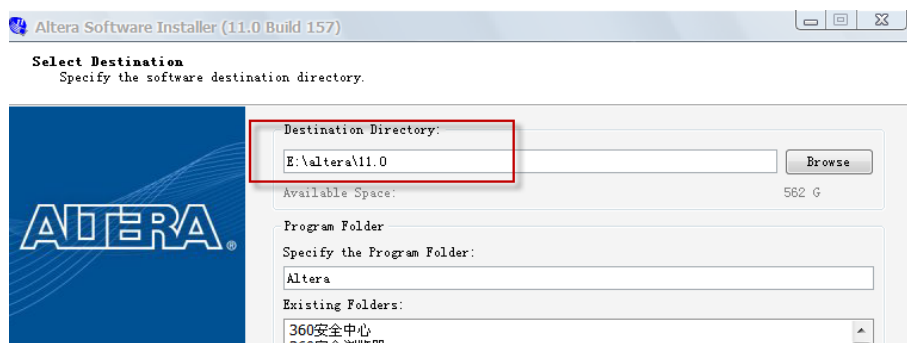
但对于我们，ase 版本已经足够了。装上就能使用。就不费破解的事了。

如想安装 ae 版本，请参考 Bingo 写过的教程，网页地址如下：

<http://www.cnblogs.com/crazybingo/archive/2011/02/21/1959893.html>

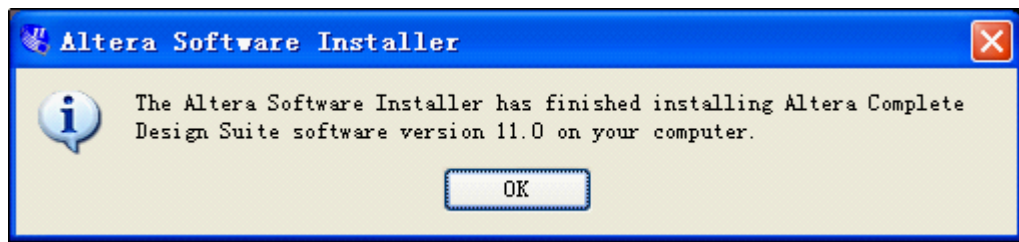
此处安装 11.0_Altera_Modelsim_ase_windows.exe，。具体步骤如下：

(1) 打开安装目录下的 setup，一路 next，直到选择路径的时候，选择与 quartus ii 安装目录相同的路径。如下图所示，我的电脑上装在 E 盘上了，您要根据您的设置来改。



(2) 继续 next，静默，等待安装完毕.....

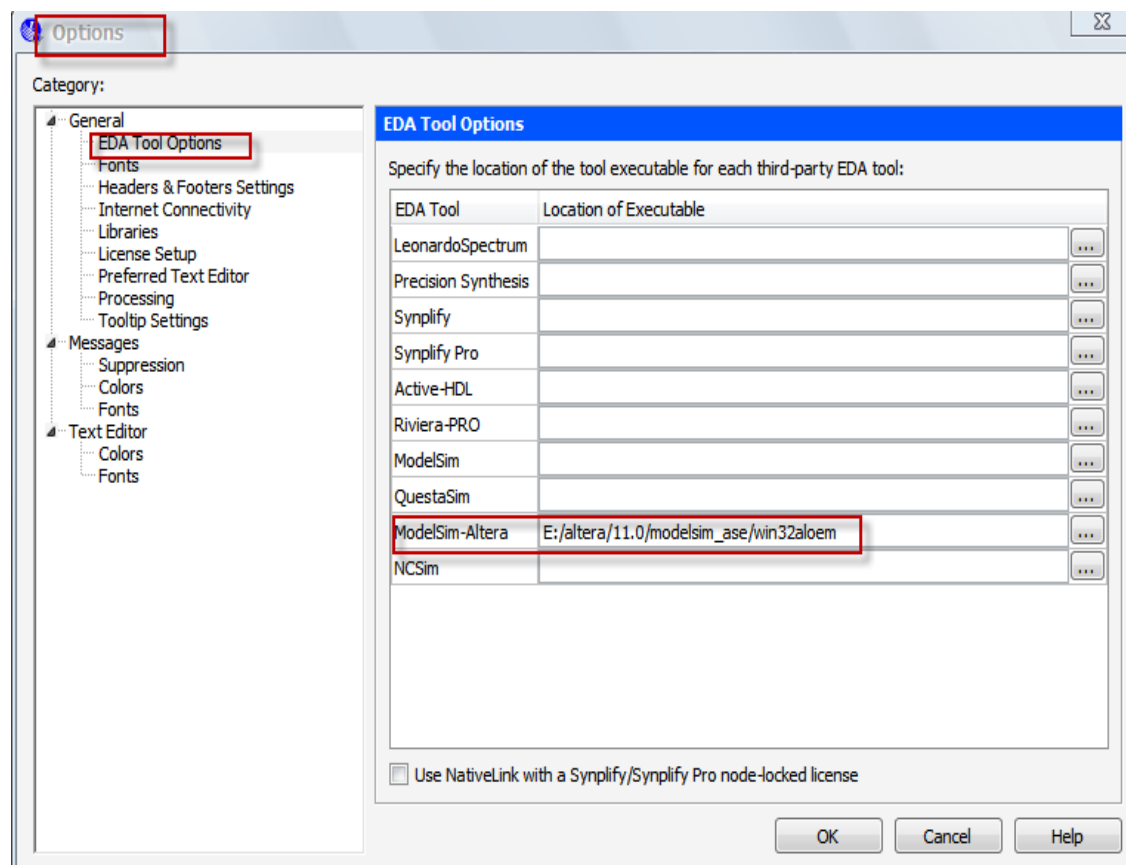
(3) 安装完毕，出现以下界面：



(4) OK, Finish!

(5) 打开 Quartus II，打开菜单 Tool-Options，在 EDA Tool Options 中的

Modelsim-Altera，选择 Modelsim-Altera 应用程序的根目录，配置 Modelsim-Altera 应用程序第三方软件路径。如下图所示：在该选项卡中下面的 ModelSim-Altera 一项指定安装路径为 E:/Altera/11.0/modelsim_ase/win32aloem（其中 E:/Altera/11.0/modelsim_ase/为我电脑中 ModelSim-Altera 6.5e 的安装路径）

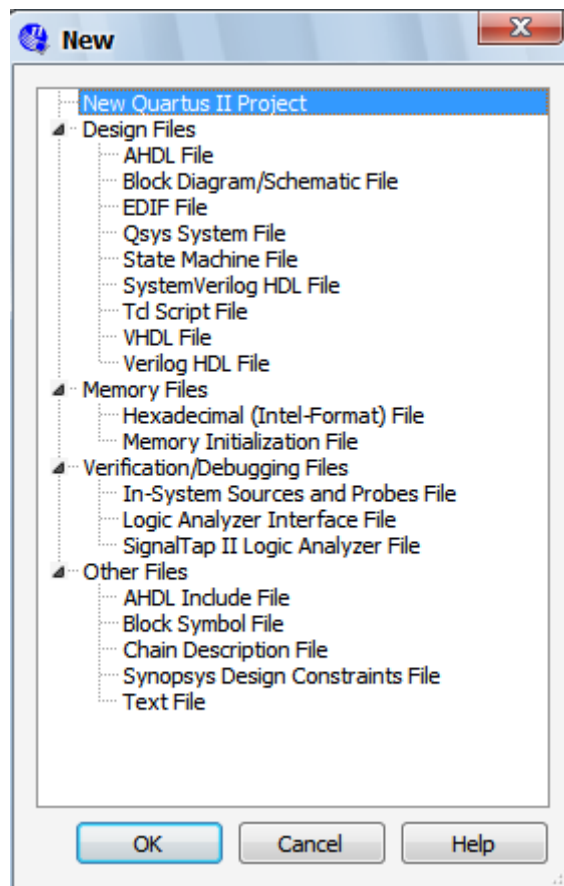
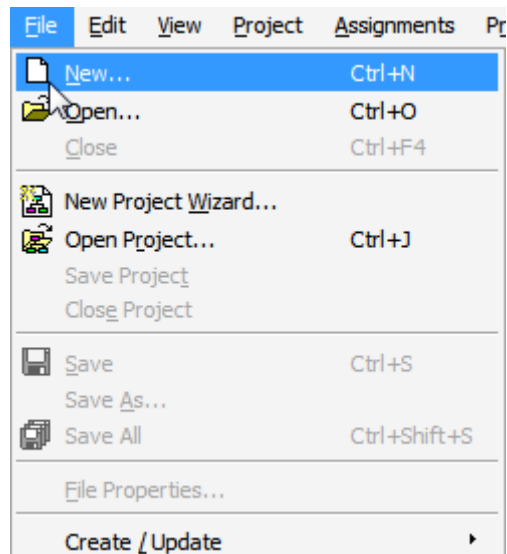


至此，Altera-Modelsim ase 版本安装完毕。

8.2 在 Quartus II 11.0 中调用 ModelSim-Altera

我们以一个简单的实例来描述整个过程：

先弄一个工程，打开 QuartusII，菜单 file---new，新建一个工程



新建一个 verilog HDL File, 代码:

```
module modelsim_test(clk,rst_n,div);
```

```
input clk;
```

```
input rst_n;
```

```
output div;
```

```
reg div;
```

```
always@(posedge clk or negedge rst_n)
```

```
if(!rst_n)div<=1'b0;
```

```
else div<=~div;
```

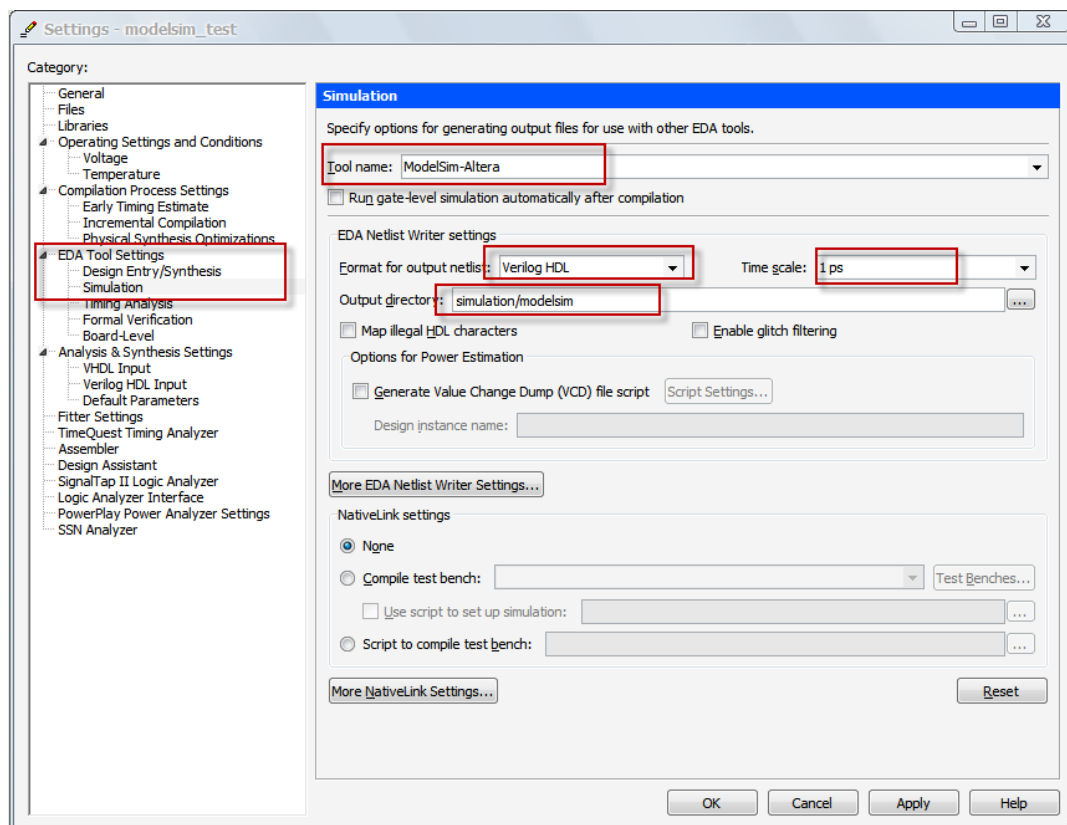
```
Endmodule
```

很简单的一个代码，是二分频电路。

我们要做什么呢，就是对这个电路进行 Modelsim 仿真，

下面，我们要设置一下：
在 Quartus II 11.0 界面菜单栏中选择 Assignments->Settings。

1. 选中该界面下 EDA Tool settings 中的 Simulation 一项；
2. Tool name 中选择 ModelSim-Altera；
3. Format for output netlist 中选择开发语言的类型 Verilog 或者 VHDL 等，
4. Time scale 指定时间单位级别
5. Output directory 指定测试文件模板的输出路径（该路径是工程文件的相对路径）。



进入到一个关键步骤：

生 成 仿 真 测 试 文 件

选择 Quartus II 11.0 开发界面菜单栏下 Processing->Start->Start Test Bench Template Writer, 提示生成成功。这个生成的仿真测试文件（`modelsim_test`工程文件下 `modelsim` 目录下找到后缀名为".vt"的文件）并根据自己需要进行编辑。下面是生成的文件原样，还没改：

```

`timescale 1 ps/ 1 ps
module modelsim_test_vlg_tst();
  // constants
  // general purpose registers
  reg eachvec;
  // test vector input registers
  reg clk;
  reg rst_n;
  // wires
  wire div;

  // assign statements (if any)
modelsim_test_i1 (
  // port map - connection between master ports and signals/registers
  .clk(clk),
  .div(div),
  .rst_n(rst_n)
);
  initial
begin
  // code that executes only once
  // insert code here --> begin

  // --> end
  $display("Running testbench");
end
  always
  // optional sensitivity list
  // @(event1 or event2 or .... eventn)
begin
  // code executes for every event on sensitivity list
  // insert code here --> begin

  @eachvec;
  // --> end
end
endmodule

```

注意：QuartusII 中 testbench 文件的后缀是.vt，产生的模板文件只是包含了端口映射，端口声明等，具体的功能还是需要设计者自己编写，下面我们在模板上修改，编写 testbench，代码如下：

```

timescale 1 ps/ 1 ps

module modelsim_test_vlg_tst();

reg clk;

reg rst_n;

wire div;

```

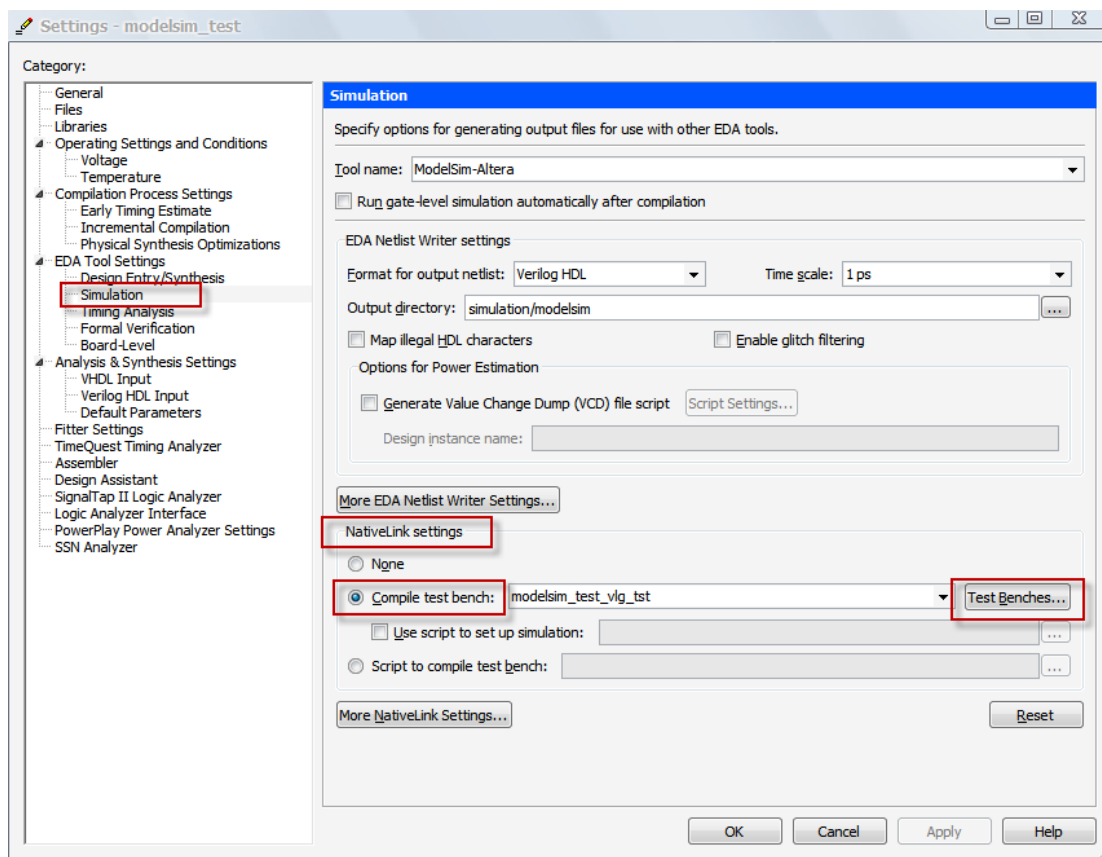
```
modelsim_test i1(  
  
    .clk(clk),  
    .div(div),  
    .rst_n(rst_n)  
);  
  
initial  
begin  
    clk=0;  
    forever  
        #10 clk=~clk;  
  
end  
  
initial  
begin  
    rst_n=0;  
    #1000 rst_n=1;  
    #1000;  
    $stop;  
  
end  
endmodule
```

代码为什么这么写，就涉及到 testbench 的编写方法学习，这个不在本文档的讨论范围，不过文档最后，附上了网上找到的一个 testbench 编写教学，大家可以参考。

下面是很关键的步骤，请一定看好，你如果设错了，就不能成功。

在 Quartus II 11.0 界面菜单栏中选择 Assignments->Settings。

1.选中该界面下 EDA Tool settings 中的 Simulation 一项；在 NativeLink settings 中选择 Compile test bench 并点击后面的 Test Benches



在 Test Benches 中点击 New

见下图：

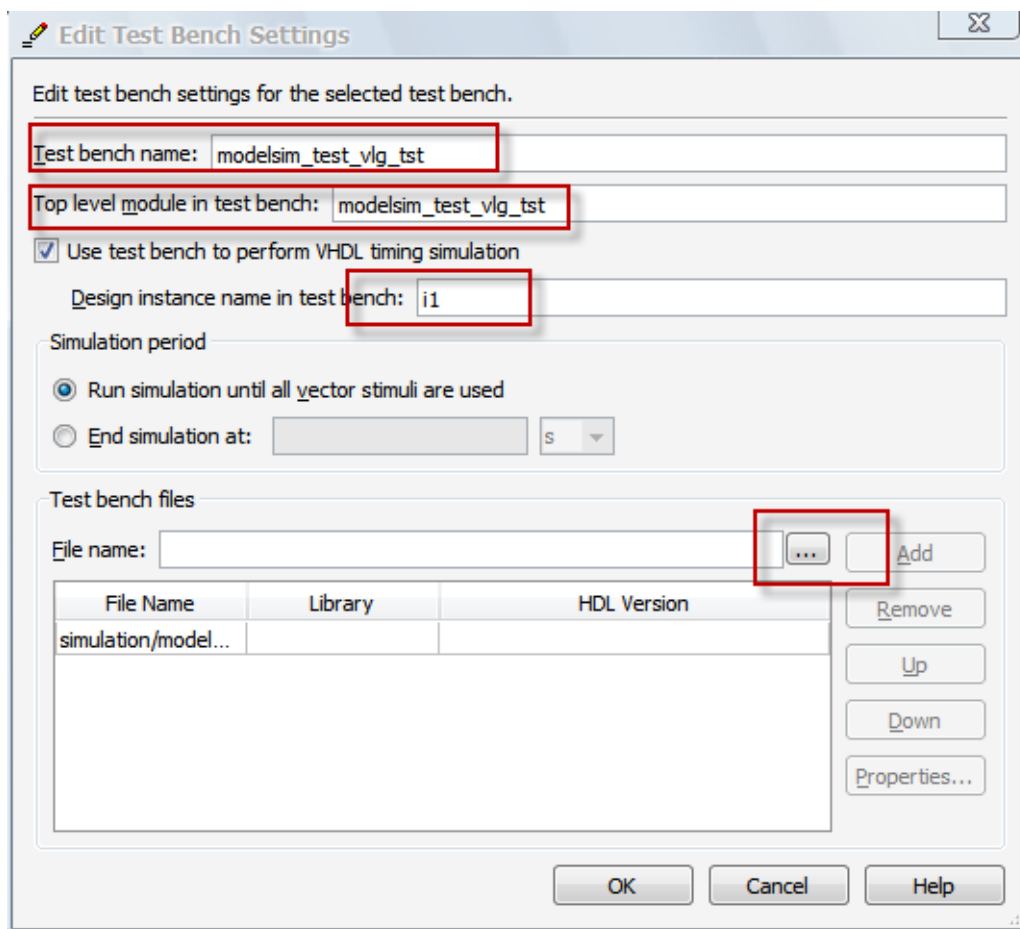
在 Test bench name 栏要填写刚刚我们创建的 testbench 文件的实体名
即 `modelsim_test_vlg_tst`；

在 Top level modele in test bench 中也填写 `modelsim_test_vlg_tst`；

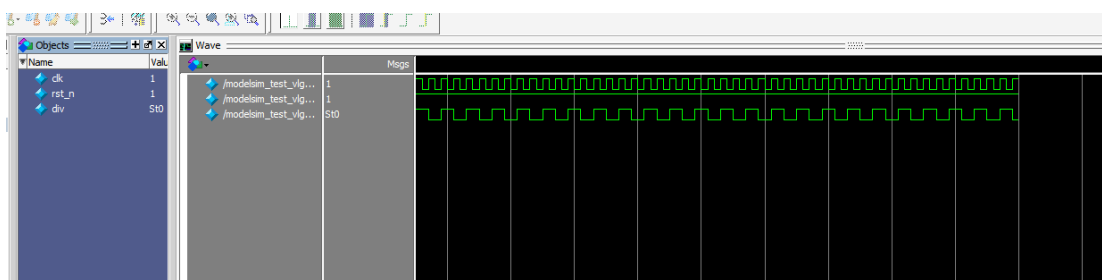
在 Design instance name in test bench 中 `i1`

这里你可以直接从 testbench 的文件里直接复制过来，避免手误写错。

然后在 Test bench files 中浏览添加 testbench 文件，然后记住点击 add,一步一步 OK。



一切准备就绪，下面在 Quartus II 11.0 界面菜单栏中选择菜单栏 Tools 中的 Run EDA Simulation Tool->EDA RTL Simulation 进行行为级仿真，接下来就可以看到 ModelSim-Altera 6.5e 的运行界面，观察仿真波形。



可以看到 div 信号是 clk 的二分频了。

第九章 技术支持与保修

9.1 关于技术支持

技术支持方式及时间:

技术支持在官方论坛 www.OurFPGA.com 提供, 请购买了开发板的朋友尽快注册, 经常访问以更新资料、参与技术讨论、寻求技术支持等。关于开发板最新消息, 最新勘误, 最新资料更新的信息都将在网站发布, 以告之您在哪里可以找到更新的资料。

支持时间:

周一至周五 9:00—17:00

我们提供的技术支持的内容仅在以下范围内:

1. 帮助客户正确或正常使用开发板硬件, 或解决硬件电路板故障;
2. 帮助客户正确或正常编译或运行我们提供的例程;
3. 对于批量客户或某单位购买的产品, 我们原则上仅针对其中一位对口技术人员代表提供技术支持, 我们不承诺对同单位下的不同人员进行一对一的技术支持及解答;

以下情况不在技术支持范围内, 仅提供技术研讨及 QQ 群讨论:

1. 代码如何理解, 怎么修改; (代码很多提供了注释, 请参考理解)
2. 与开发板本身无关的技术问题;
3. 二次开发 (包括用开发板完成自己想要的某功能、某技术) 遇到的问题
4. 自己设计或修改的代码或硬件遇到的问题

还有几点要说明下:

1. 有的人希望买一块板子, 那么所有 FPGA 的问题, 我们都得回答, 所有工作学习遇到的 FPGA 问题, 认为“我们也有义务帮你”, 抱歉, 我们不是全才, 这个我们是做不到的。这其中包括: 自己毕设, 自己搭电路, 自己写程序, 自己工作要用到, 自己项目要用的等等

问题。这些都不在我们技术支持的范围内的。

当然，你可以发帖子出来，学友们会看到，他们或是我们熟悉的或是知道的也许看到会一起讨论一下，互相学习下。

2. 属于我们工作范围内的，我们不会推，不管是正当的要求还是售后，我们都将按我们的承诺来做好。

QQ 群仅作为技术讨论，学友互助，资料更新，最新公告发布，但不做技术支持：

我们提供了 QQ 群给大家进行技术讨论，但是 QQ 群发布的消息，易丢失，易漏看，特别是对于通用问题，经常重复发布，重复的问，因此，我们还是强烈建议在论坛发帖，这样长期积累，可以有效把技术帖保存，你问过的问题，其他人可以参考，别人问过的问题，你同样可以参考。不建议大家在 QQ 群过多的提一些用户手册上都已经详细指导过的问题，当然我们推荐群友们互助，但我们不承诺在 QQ 群上对每一个问题都回答，实际上，纵观市面上出售开发板的，QQ 群都仅作为辅助，但凡承诺一对一技术支持的，都是夸下海口，却无法真正兑现。大家可以试想一下：如果有 1000 个客户，每天有 100 个人问，那么就算安排 10 个人，可能也无法一对一支持。

QQ 群内的朋友全是购买过的客户，大家使用过程中有问题可以在群内一起讨论，实际上，已经有不少学友通过 QQ 群热心网友的帮助，解决了自己的相关问题，我们提供了一个好的学习氛围的场所，希望大家能充分的利用。

关于 QQ 群的详细说明及我们的一些解释和说明，可以参考以下网页：

<http://www.ourfpga.com/forum.php?mod=viewthread&tid=10525&extra=page%3D1>

另外官方技术 QQ 群限购买过开发板的客户加入，加入前需要提供购买 ID、姓名及 VIP 卡号。不提供的我们有权拒绝加入；其他还未购买的人，如果想加 QQ 群，我们也提供了普通群，请注意官网发布的信息。

9.2 关于产品保修

伟欣科技出售的每一套开发板，均由专人严格测试，测试通过后才发出。在运输保护问题上，我们想尽了一切办法，我们不惜一切代价，定做专用珍珠棉加厚保护盒！这是迄今为止最好的一种保证方式！在这种情况下，即使进行粗暴装卸货，也能保证开发板的安全！因

此，您收到开发板后，硬件方面一般不会有问题的！我们有这样的自信。当然，什么事不是100%绝对的，我们仍然提供让您放心满意的售后服务：

- 1.从发货日开始计，7天内产品有质量问题时，可免费更换；
- 2.从发货日开始计，提供三个月保修，如果用户操作不当导致损坏，无保修；
- 3.保修期外，我们收取维修费用及元件成本费；
- 4.器件人为损坏，均不在产品质量保修范围内，FPGA 芯片，液晶屏、下载线、电源、USB 串口线为易损品，不在保修范围内。
- 5.请您妥善保管开发套件的VIP卡及编号，这是保修的重要凭证。

9.2 关于产品转让及赠送

用户转让产品（包括转让产品全部内容或部分内容）给其他人或公司，无论是收费转让，还是免费赠送，均失去所有的技术支持及保修服务，由于在转让过程造成的配件不全，资料失丢等等，我们均不提供支持或提供补全的服务。

附录 脚配置脚本文件

```
#####
```

```
# www.OurFPGA.com
```

```
#####
```

```
#-----GLOBAL-----#
```

```
set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-STATED"
```

```
set_global_assignment -name ENABLE_INIT_DONE_OUTPUT OFF
```

```
set_location_assignment PIN_23 -to clock
```

```
set_location_assignment PIN_131 -to rst_n
```

```
#-----SDRAM-----#
```

```
set_location_assignment PIN_56 -to S_DQ[0]
```

```
set_location_assignment PIN_57 -to S_DQ[1]
```

```
set_location_assignment PIN_58 -to S_DQ[2]
```

```
set_location_assignment PIN_59 -to S_DQ[3]
```

```
set_location_assignment PIN_60 -to S_DQ[4]
```

```
set_location_assignment PIN_61 -to S_DQ[5]
```

```
set_location_assignment PIN_63 -to S_DQ[6]
```

```
set_location_assignment PIN_64 -to S_DQ[7]
```

```
set_location_assignment PIN_80 -to S_DQ[8]
```

```
set_location_assignment PIN_77 -to S_DQ[9]
```

```
set_location_assignment PIN_76 -to S_DQ[10]
```

```
set_location_assignment PIN_75 -to S_DQ[11]
```

```
set_location_assignment PIN_74 -to S_DQ[12]
```

```
set_location_assignment PIN_72 -to S_DQ[13]
```

```
set_location_assignment PIN_70 -to S_DQ[14]
```

```
set_location_assignment PIN_69 -to S_DQ[15]
```

```
set_location_assignment PIN_103 -to S_A[0]
```

```
set_location_assignment PIN_104 -to S_A[1]
```

```
set_location_assignment PIN_106 -to S_A[2]
```

```
set_location_assignment PIN_105 -to S_A[3]
```

```
set_location_assignment PIN_94 -to S_A[4]
```

```
set_location_assignment PIN_92 -to S_A[5]
```

```
set_location_assignment PIN_90 -to S_A[6]
```

```
set_location_assignment PIN_89 -to S_A[7]
```

```
set_location_assignment PIN_88 -to S_A[8]
```

```
set_location_assignment PIN_87 -to S_A[9]
```

```
set_location_assignment PIN_102 -to S_A[10]
```

```
set_location_assignment PIN_86 -to S_A[11]
```

```
set_location_assignment PIN_82 -to S_CLK
```

```
set_location_assignment PIN_99 -to S_BA[0]
```

```
set_location_assignment PIN_101 -to S_BA[1]
```

```
set_location_assignment PIN_95 -to S_nCAS
```

```
set_location_assignment PIN_84 -to S_CKE
```

```
set_location_assignment PIN_96 -to S_nRAS
```

```
set_location_assignment PIN_68 -to S_nWE
```

```
set_location_assignment PIN_97 -to S_nCS
```

```
set_location_assignment PIN_67 -to S_DQM[0]
```

```
set_location_assignment PIN_81 -to S_DQM[1]
```

```
#-----LED-----#
```

```
set_location_assignment PIN_142 -to led[1]
```

```
set_location_assignment PIN_141 -to led[2]
```

```
set_location_assignment PIN_139 -to led[3]
```

```
set_location_assignment PIN_138 -to led[4]
```

```
set_location_assignment PIN_137 -to led[5]
```

```
set_location_assignment PIN_135 -to led[6]
```

```
set_location_assignment PIN_134 -to led[7]
```

```
set_location_assignment PIN_133 -to led[8]
```

```
#-----KEY-----#
```

```
set_location_assignment PIN_107 -to key[0]
```

```
set_location_assignment PIN_108 -to key[1]
```

```
set_location_assignment PIN_110 -to key[2]
```

```
set_location_assignment PIN_112 -to key[3]
```

```
set_location_assignment PIN_113 -to key[4]
```

```
set_location_assignment PIN_114 -to key[5]
```

```
set_location_assignment PIN_115 -to key[6]
```

```
set_location_assignment PIN_116 -to key[7]
```

```
#-----CKEY-----#
```

```
set_location_assignment PIN_128 -to ckey[1]
```

```
set_location_assignment PIN_127 -to ckey[2]
```

```
set_location_assignment PIN_118 -to ckey[3]
```

```
set_location_assignment PIN_117 -to ckey[4]
```

```
#-----DIG-----#
```

```
set_location_assignment PIN_150 -to dig[0]
```

```
set_location_assignment PIN_149 -to dig[1]
```

```
set_location_assignment PIN_152 -to dig[2]
```

```
set_location_assignment PIN_151 -to dig[3]
```

```
set_location_assignment PIN_205 -to dig[4]
```

```
set_location_assignment PIN_206 -to dig[5]
```

```
set_location_assignment PIN_207 -to dig[6]
```

```
set_location_assignment PIN_208 -to dig[7]
```

```
set_location_assignment PIN_170 -to seg[0]
set_location_assignment PIN_165 -to seg[1]
set_location_assignment PIN_169 -to seg[2]
set_location_assignment PIN_175 -to seg[3]
set_location_assignment PIN_173 -to seg[4]
set_location_assignment PIN_168 -to seg[5]
set_location_assignment PIN_164 -to seg[6]
set_location_assignment PIN_171 -to seg[7]
```

```
#-----BEEP-----#
```

```
set_location_assignment PIN_3 -to beep
```

```
#-----UART-----#
```

```
set_location_assignment PIN_143 -to rxd
```

```
set_location_assignment PIN_144 -to txd
```

```
#-----LCD-----#
```

```
set_location_assignment PIN_180 -to lcd[1]
```

```
set_location_assignment PIN_185 -to lcd[2]
```

```
set_location_assignment PIN_182 -to lcd[3]
```

```
set_location_assignment PIN_188 -to lcd[4]
```

```
set_location_assignment PIN_187 -to lcd[5]
```

```
set_location_assignment PIN_191 -to lcd[6]
```

```
set_location_assignment PIN_189 -to lcd[7]
```

```
set_location_assignment PIN_193 -to lcd[8]
```

```
set_location_assignment PIN_192 -to lcd[9]
```

```
set_location_assignment PIN_197 -to lcd[10]
```

```
set_location_assignment PIN_195 -to lcd[11]
```

```
#-----AD-----#  
set_location_assignment PIN_31 -to adc_clk  
set_location_assignment PIN_33 -to cs_n  
set_location_assignment PIN_30 -to sdat_in
```

```
#-----DA-----#  
set_location_assignment PIN_47 -to dac_clk  
set_location_assignment PIN_48 -to dac_data  
set_location_assignment PIN_46 -to dac_ldac  
set_location_assignment PIN_45 -to dac_load
```

```
#-----PS2-----#  
set_location_assignment PIN_145 -to ps2_clk  
set_location_assignment PIN_146 -to ps2_dat
```

```
#-----DS18B20-----#  
set_location_assignment PIN_147 -to ds18b20
```

```
#-----VGA-----#  
set_location_assignment PIN_5 -to hsync  
set_location_assignment PIN_4 -to vsync  
set_location_assignment PIN_14 -to vga_r[0]  
set_location_assignment PIN_15 -to vga_r[1]  
set_location_assignment PIN_12 -to vga_r[2]  
set_location_assignment PIN_13 -to vga_g[0]  
set_location_assignment PIN_10 -to vga_g[1]  
set_location_assignment PIN_11 -to vga_g[2]  
set_location_assignment PIN_6 -to vga_b[0]  
set_location_assignment PIN_8 -to vga_b[1]
```

```
#-----RTC-----#
```

```
set_location_assignment PIN_40 -to rtc_scl
```

```
set_location_assignment PIN_41 -to rtc_sda
```

```
#-----IIC-----#
```

```
set_location_assignment PIN_43 -to i2c_scl
```

```
set_location_assignment PIN_44 -to i2c_sda
```

```
#-----END-----#
```

附录 常见问题

1.你们的开发板适合初学者吗？

答：适合的，完全没问题！在设计之初，我们针对初学者的需求，认真研究了市面的开发板的现状，专门面向初中级用户而设计，采取的设计原则是：

- ※通用及经典资源不能少，适当拔高；
- ※使用方便，便于扩展及自行开发；
- ※配套资料详细，便于自学；
- ※重视服务理念，提供论坛技术讨论氛围，形成有效学习推动力；

因此，睿智 FPGA 开发板用户群体面向广大的高校相关专业学生、电子爱好者、科研单位、企事业单位的开发设计人员，同时也适合于产品原型的快速开发、学生参加各种电子设计大赛、学习 FPGA 技术入门，课程设计以及毕业设计、系统设计前期快速评估设计方案。

2.FPGA 好学吗？买了你们的开发板，能学会吗？

答：FPGA 相对于单片机，应该来说还有是些难度的。但是不要有畏难情绪，只要打好基础，多动手，多实践，同时希望尽快加入 www.OurFPGA.com 的讨论中去，论坛以后还会启动学习小组或是网络教学连载，融入这种学习氛围中去，坚持，不要放弃。相信您一定会有所收获。

3.收到开发板后，我该怎么开始学习，我不知道从何下手？

答：建议初学者，可以先学习一下 Verilog 或 VHDL 语法，同时通读“睿智 FPGA 开发板用户手册”，先获得对开发板硬件的必要了解，按手册说明，安装开发软件，尝试打开工程，下载文件。当然，我们强烈建议您尽快注册论坛，并经常访问，以了解最新资料更新，参与技术交流。

4.学 VHDL 好还是 Verilog 好？哪个找工作容易些

答：这两种无所谓哪个好，哪个不好，就目前我所了解，国内的公司使用 Verilog 的稍微多些，记住，是“稍微”，并非绝对的。如果您还没学过 VHDL，那我强烈建议您学习 Verilog，睿智 FPGA 开发板附带的例程，目前是 Verilog 的，VHDL 语法的例程目前也有更新了，在群共享里有下载，还会有更新，请关注。

5.你们的开发板有相关的配套指导书、资料、或是视频吗？

答：目前开发板配套了丰富的例程及专业级代码，例程绝非网上到处可以下载到的代码，并且几乎所有代码都有详细的中文注释，这对于初学者来说，非常有用！另外，我们配套了用户手册，包含了大量信息。更为详细的资料或文档，我们将在论坛上分期发布。我们计划在论坛以小篇幅文档形式分节分时发布详细指导文档，最后整理成全套的指导书，请大家关注。视频教程方面，我们店内有一套由工作人员精心整理的 FPGA EDA 工程师从入门到进阶教程包，此教程包完全包揽所有有价值的 FPGA 相关教程，大部分是网上无法获取的，都是我们近期的珍藏，教程包自推出以来，咨询者甚多，受到使用者的大量好评。需要的请联系。同时，与开发板配套的视频教程也在规划中，视频教程及指导书制作是件很费神费力的事，本着为用户负责的根本，这些还是要认真制作的，留意论坛公告。

6.买了你们的开发板，你们提供售后和技术支持吗？

答：当然要提供的，其实您花钱购买的不仅仅是一套开发板，其中也包含了售后和技术支持的。

7.听说你们有网站有保修卡，这个是做什么用的？

我们特意制作了精美的保修卡，凡是购买开发板的，均会附带上，并不需要单独花钱购买，保修卡的主要作用：保修、官网的各种活动、论坛赠送金币等等均要凭保修卡号来获得及参加，请妥善保管。

7.听说你们网站有高级会员下载区，这个是做什么用的？怎么加入？

答：高级会员下载区是针对 那些对视频教程及资料有较高热情的用户设计的，这个是要购买才可获得权限和下载币的，您有没有这个权限只取决于您购买开发板时是不是也同时购买了高级会员下载权限，如果没有购买，是无法加入的，请谅解！

如果您确实购买过高级权限，那么加入的步骤：

第一步：交易确认付款且好评（必须完成）

第二步：请提供以下相关信息给我们（发到我们的手机短信、QQ、旺旺），

我们将在 1 个工作日内完成审核并通过

提供您的旺旺 ID、收货人姓名、电话、VIP 卡号、论坛用户名）

注意：未确认收货好评之前不予开通高级会员权限，谢谢理解。

8.注册论坛、加入 QQ 群、有什么好处吗？

答：加入论坛可以参与技术讨论，您将发现，不只您一个人在“战斗”，大家一起学习不是更有效率吗？QQ 群是论坛的一个辅助方式。并且论坛里面有很多精品电子工程相关视频教程发布，目前已经有 200 多门。。。。