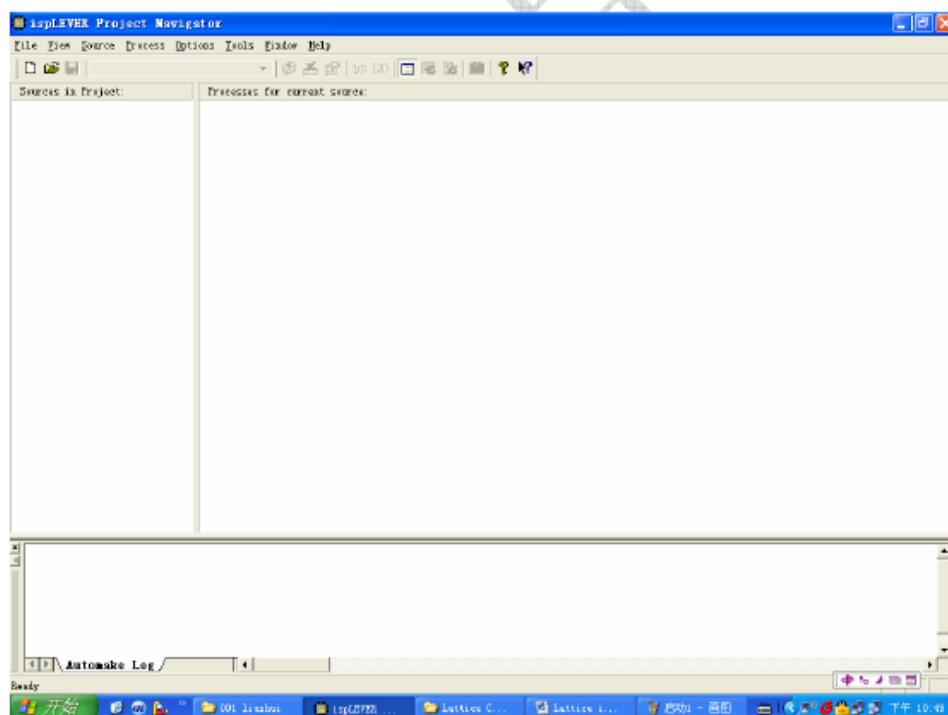


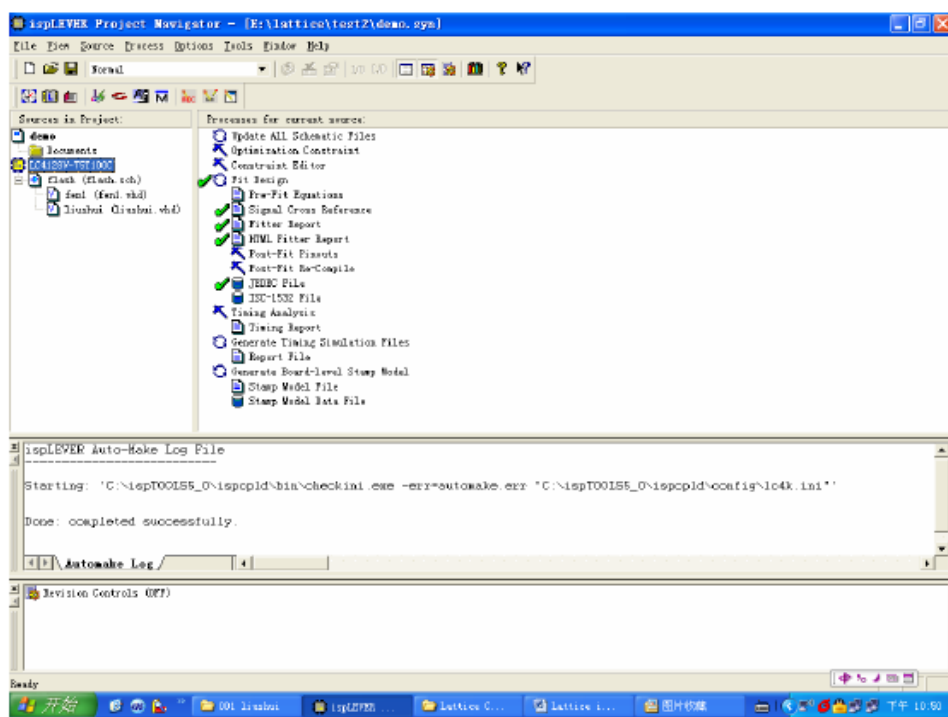
# Lattice ispLEVER 软件使用教程

## 一、建立项目

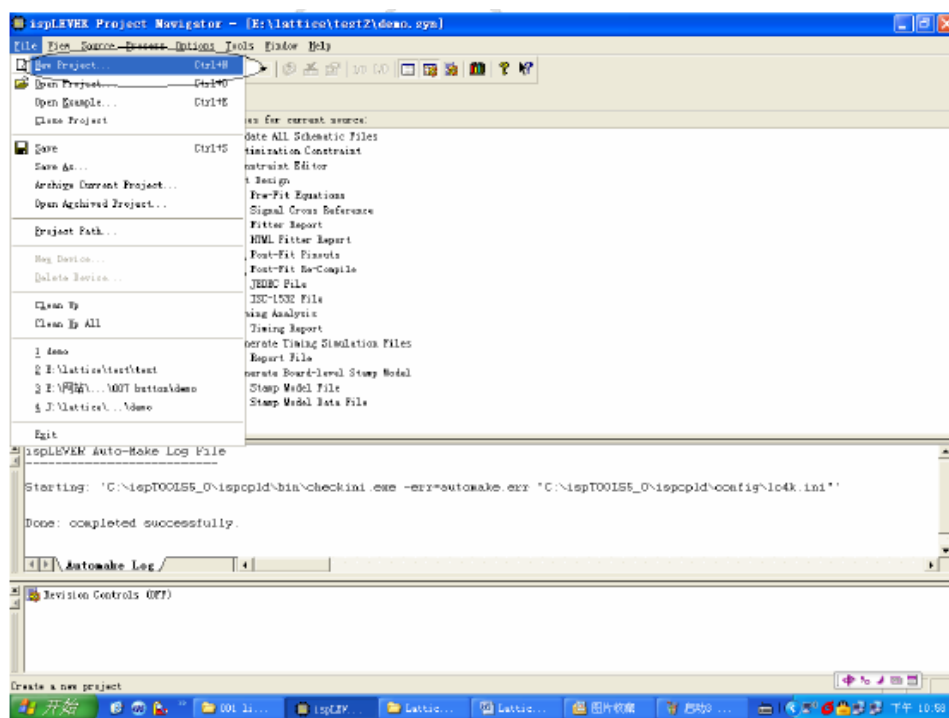
IspLEVER 软件安装后，就可以使用，双击桌面图标或通过程序选择即可打开 ISPLEVER 软件，若是第一次使用软件，则软件出现下面的启动画面。



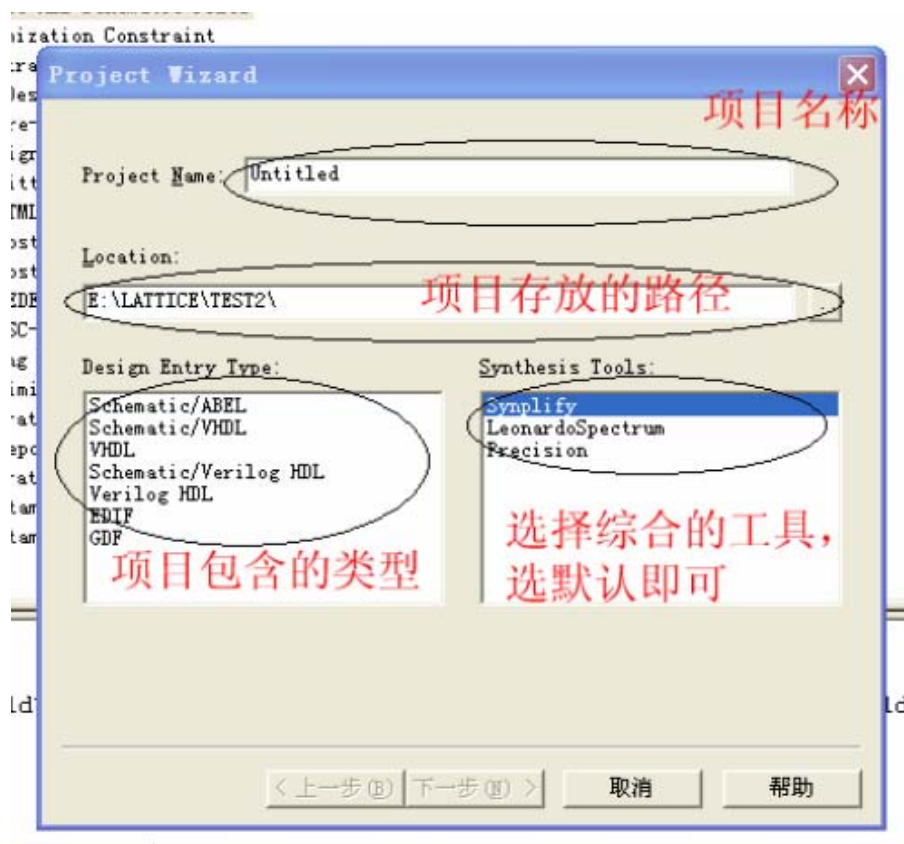
若以前使用过这个软件，则软件启动时会自动加载上次打开的项目，如下图：



现在我们需要新建一个项目，那么可以点击 File->New Project，如下图：



点击后会出现这么一个对话框：

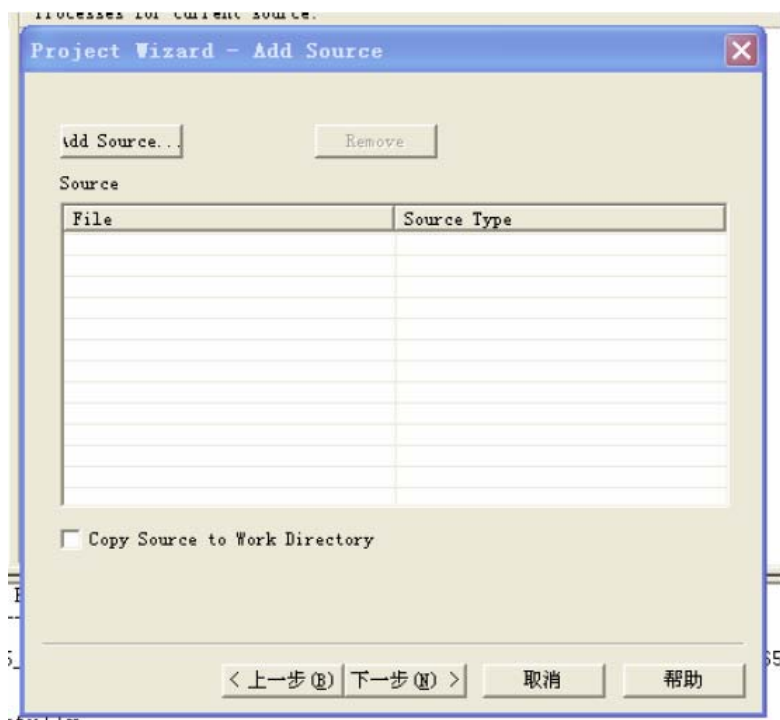


在 project name 框内是让你输入项目的名称，自己可以任意起个名字，但是

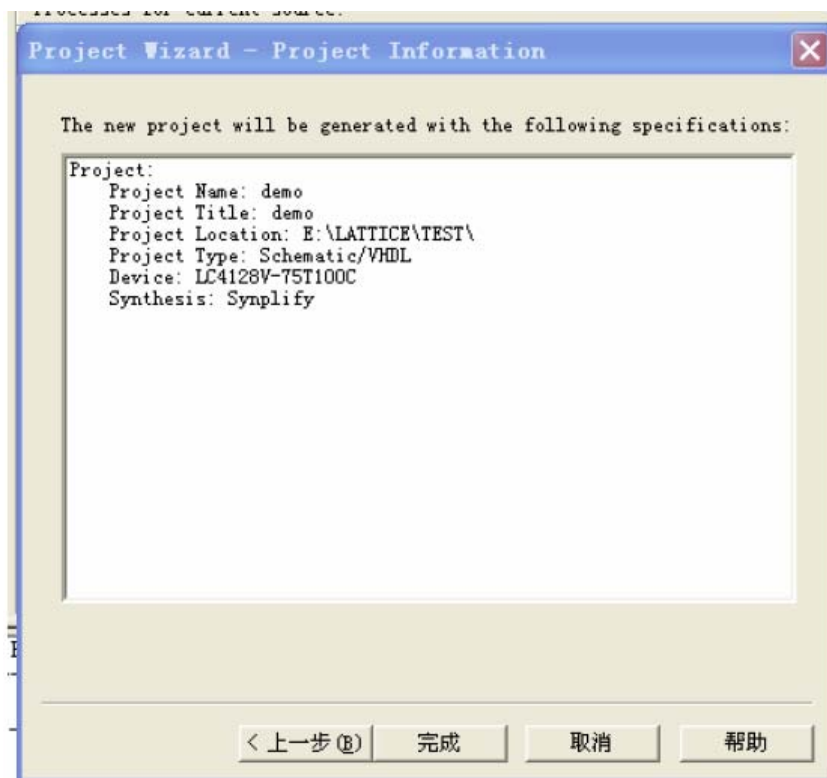
项目名称不能为中文名，在这里我们起名为“demo”；在 location 框内是让你指定这个项目存放的路径，项目名称路径中不能有中文名，必须全部为英文或数字；在 design entry type 内是让你选择一个这个项目文件的类型，若只有 VHDL 文件，那只要点第三个“VHDL”，若包含原理图和 VHDL 则可以点第二个“schematic/VHDL”，其他的依次类推，在这个项目中我们选择“schematic/VHDL”，在 synthesisi tolols 里可以为这个项目选择综合的工具，一般选择第一个“synplify”就可以，当完成上述步骤后点击下一步，会出现以下对话框：



这个对话框是为这个项目选择所用芯片的具体型号，按照上面的说明，我们这里选择 LC4128V-75T100C CPLD 芯片，这是个具有 128 个宏单元，速度等级为 7.5，TQFP100 封装商业级的芯片。选择完后继续点击下一步，出现以下的对话框：



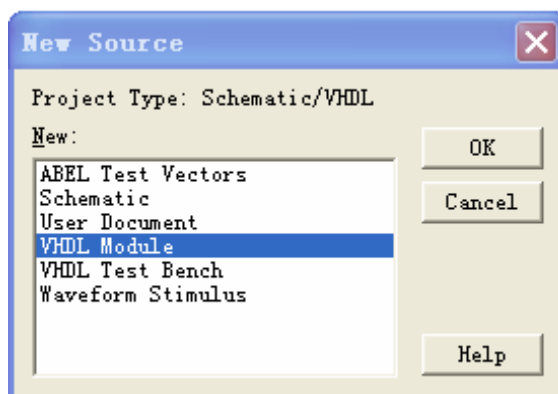
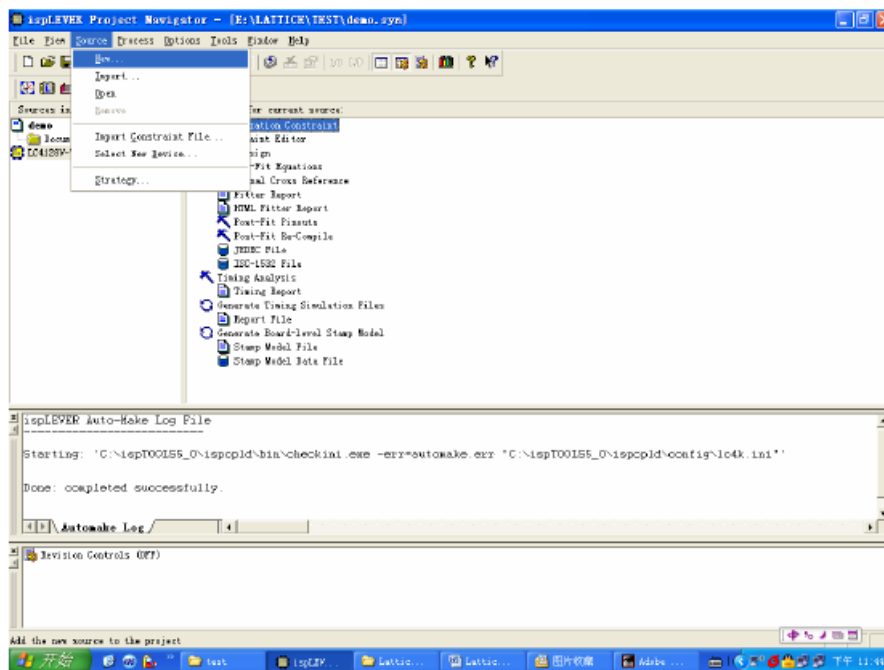
这个对话框是说，如果你这个项目用到的一些模块在其他项目中已经设计过，那么可以把这些模块加入到这个项目中，实现模块的共享，如果不需要，则直接点下一步



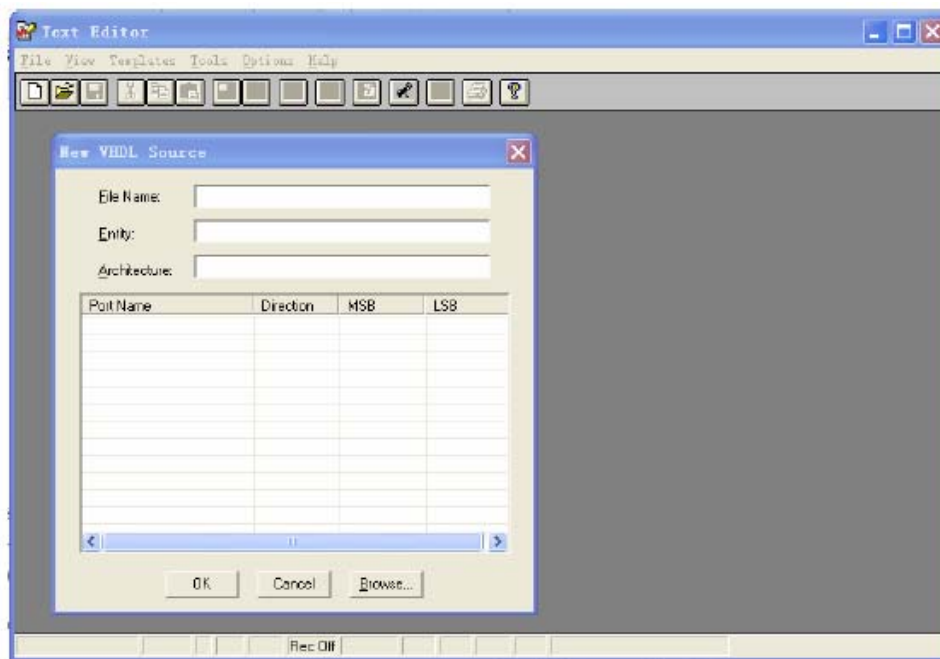
这个是建立项目的最后一个对话框，这个对话框详细列出了这个项目的设置，然后直接点击完成，就完成了一个新项目的建立。

## 二、项目设计

在建立一个新的项目后，我们就可以开始具体的设计了，点击 source->new 即可新建一个文件如下图：

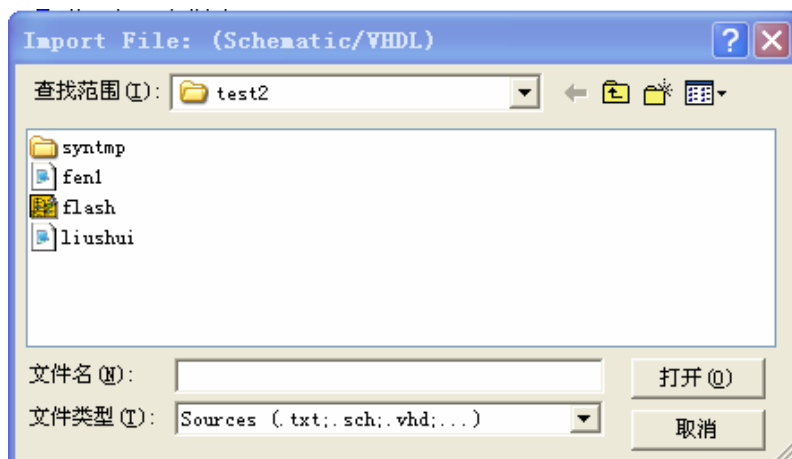


在上面这个对话框中我们可以选择文件的类型，我们选择一个 VHDL Module，也就是 VHDL 类型的文件，然后点击 OK，出现这个对话框：

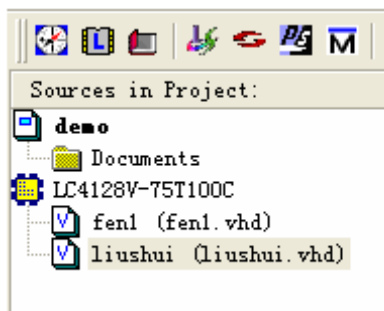


你可以把你写的 VHDL 文件的文件名、实体名、结构名按照上面的对话框写上，然后点击 OK，这样 ispLEVER 就会为你这个要写的程序框架写好，让你省点敲键盘的力气；当然你也可以点击 cancel，等你程序写好后，再存盘都是可以的。我们点击 cancel 后，然后点击 file->new，会出现一个文本编辑的界面，然后可以写好你的程序，写好后存盘。这样一个 VHDL 程序就写好了，一般一个项目不可能只有一个文件，然后你可以再点击 file->new，再建立一个文本，写第二个程序，然后存盘，依次类推，写好这个项目中所需的 VHDL 程序，然后关闭整个“text editor”。

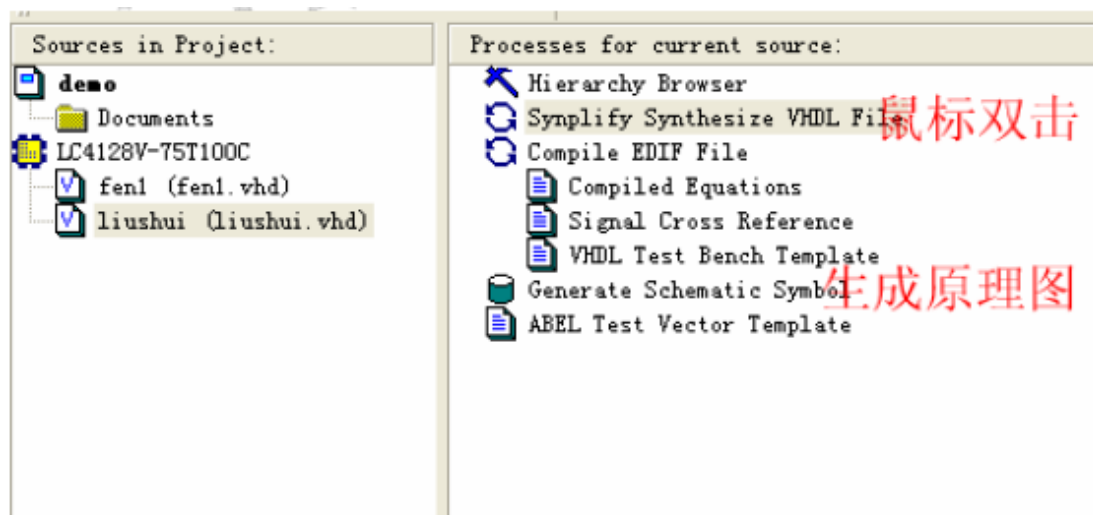
然后回到 ispLEVER project navigator 软件界面，接下来我们要做的是要把刚才所写的程序导入到这个项目中，点击 source->import，出现这么一个对话框：



我们刚才写了两个 VHDL 程序，一个是文件名为 fen1.vhd，另外一个为 liushui.vhd，这时我们把这两个文件导入进来，见下图：



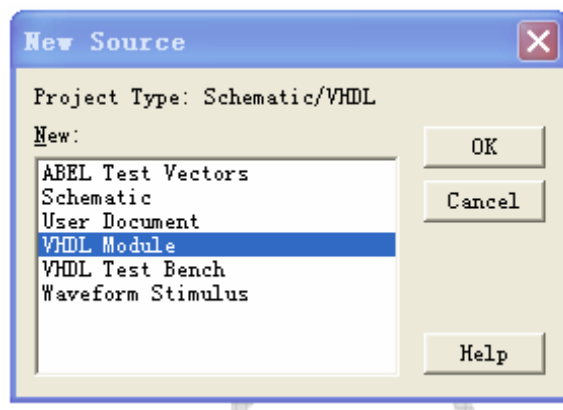
我们可以看到这两个文件已经被导入进来了，然后我们可以编译我们刚才写的程序有没有错误，鼠标左击要编译的文件，下图点击了 liushui.vhd，然后双击“synplify synthesize VHDL file”，



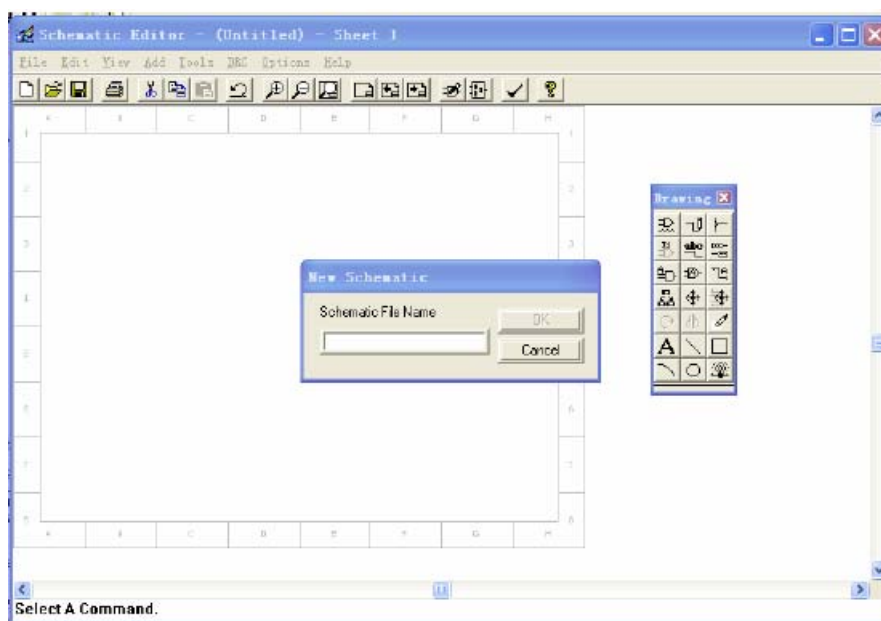
若程序错误时，编译不通过，在“synplify synthesize VHDL file”前面会打上红色的叉，然后在下面的信息框里也会提示错误信息，那就需要你到程序里面修改了，这里不再详细介绍。若程序没有错误，则会在“synplify synthesize VHDL file”前面打上绿色的勾，表示编译通过，接下来我们想要把这个程序生成原理图符号，以便在顶层的原理图中调用，那么双击：“generate schematic symbol”（见上面），符号生成后会在“generate schematic symbol”前面打上绿色的勾，其他几个 VHDL 程序也一样，先编译，然后可以生成原理图符号。

最后我们设计顶层文件，顶层文件采用原理图方式，在 source->new，出现刚才的对话框：

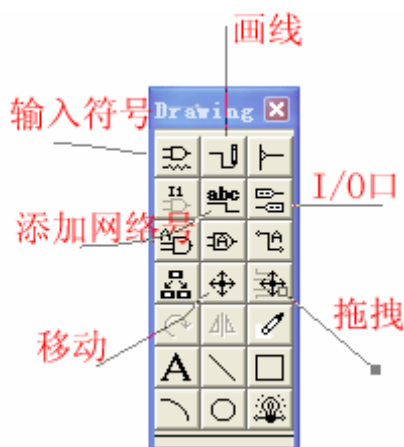




这次我们选择了 schematic，然后点击 OK，出现了一个对话框

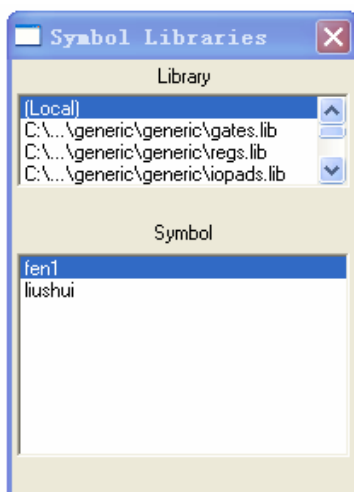


我们输入原理图文件的名字，这里我们取名 flash，然后输入原理图符号，I/O口，网络标号等，画面旁边的哪个工具栏见下图：

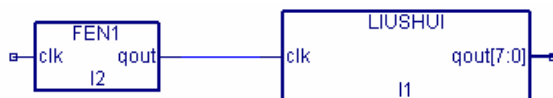


点击“输入符号”小图标

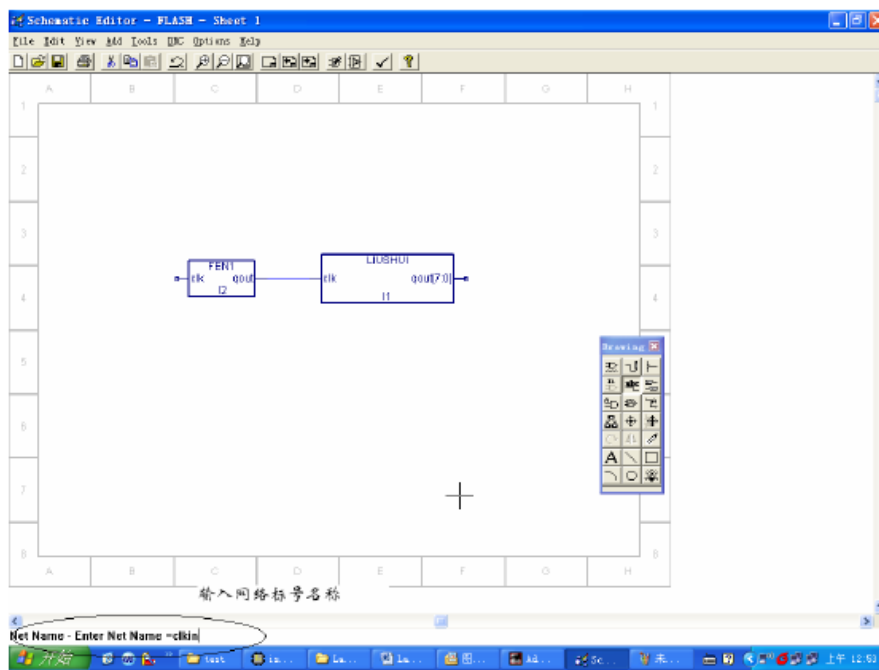


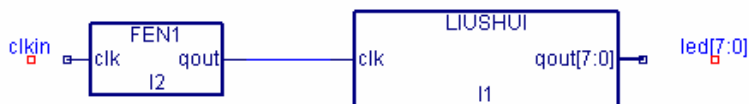


在“library”下“local”里就是用户自己生成的一些原理图符号，在其余的文件夹下，则是常用的与门、或门、非门等符号，然后单击“symbol”下的文件名，鼠标指针下就会有一个符号，然后在原理图空白的地方，鼠标右键一点，就放置了一个元件符号，其他的元件符号则是一样的放置。

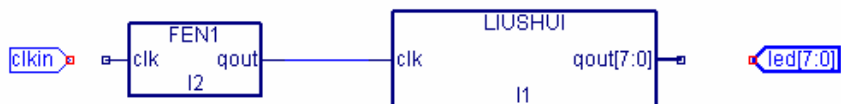


接下来要为一些线添加网络符号，点击“添加网络号”小图标，在下方输入网络标号名称，然后按回车键，鼠标指针下面会有一个刚才输入的网络标号名，然后鼠标移动到适当的位置，单击左键，即可放置一个网络标号，见下图：

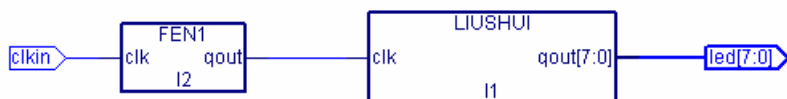




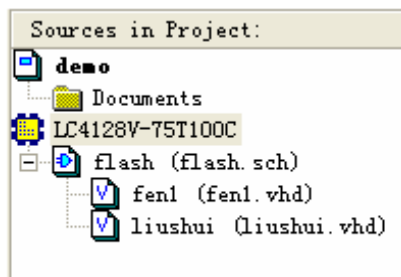
接下来就是放置输入、输出管脚符号了，单击“I/O 口”小图标，让你选择是输入口还是输出口，然后鼠标指针下面跟有一个 I/O 口，然后移动到网络标号旁边，在网络标号下面小红框上单击，这样 I/O 口就添加上了。



最后把连线连上，单击“画线”小图标，然后连接两个端点，就可连接完一条线，见下图：



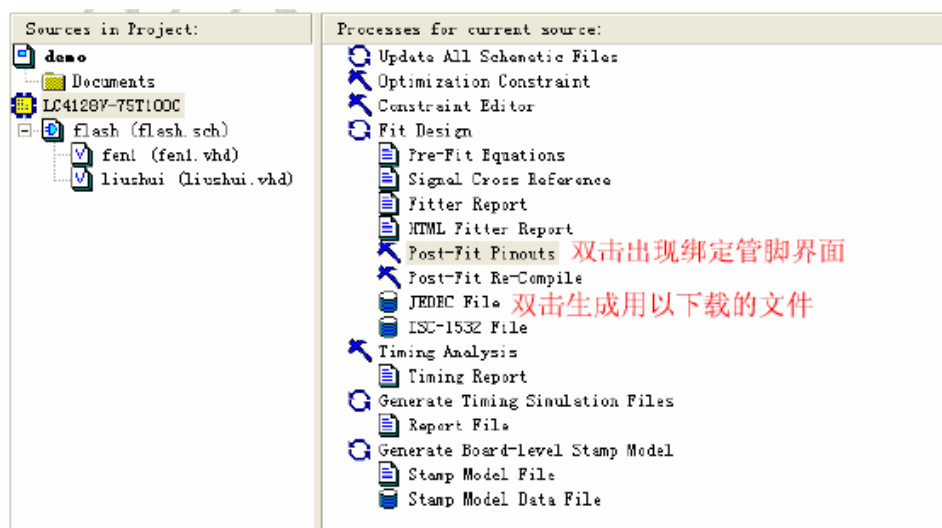
这样这个顶层原理图也画完了，存盘后关闭。至此，一个简单的项目设计完成了，回到 ispLEVER project navigator 软件界面，这时我们可以看到：



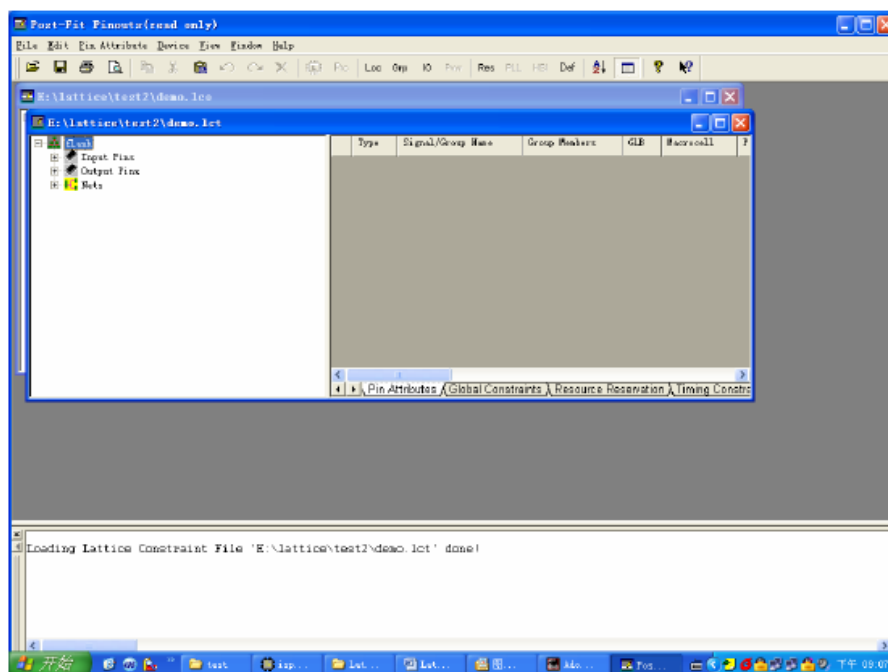
我们单击 flash 文件名，然后双击“synplify synthesize VHDL file”编译顶层文件，编译通过后，就完成了个项目的设计。

### 三、绑定管脚

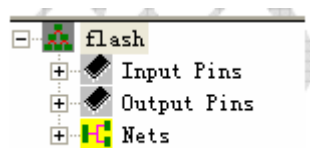
项目编译通过后，接下来就要在芯片上绑定管脚，单击“LC4128V-75T100C”，见下图：



然后双击“post-fit pinouts”，会出现一个绑定管脚的界面，见下图：



我们点击 file->open，就在项目所在的文件夹下打开后缀名为.lct 的文件，在这个项目中我们可以打开 demo.lct 的文件，然后双击下图中的“input pins”和“output pins”，这时就会在右边的框中列出我们在这个项目中使用到的输入输出口。

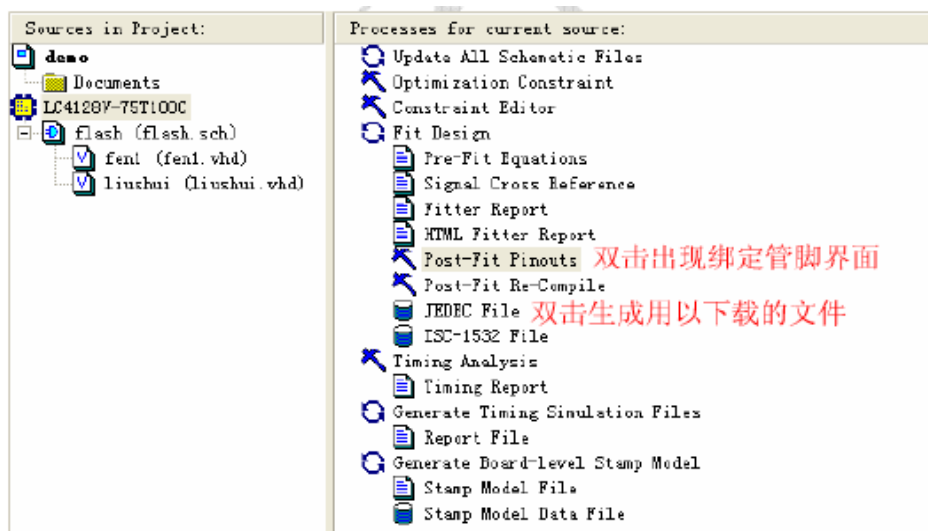


	Type	Signal/Group Name	Group Members	GLE	Macrocell	Pin	Bank	IO Types	SI
1	Clo...	clk_in	N/A		N/A			LVCNOS18	N/
2	Out...	led_0_	N/A		N/A			LVCNOS18	FA
3	Out...	led_1_	N/A		N/A			LVCNOS18	FA
4	Out...	led_2_	N/A		N/A			LVCNOS18	FA
5	Out...	led_3_	N/A		N/A			LVCNOS18	FA
6	Out...	led_4_	N/A		N/A			LVCNOS18	FA
7	Out...	led_5_	N/A		N/A			LVCNOS18	FA
8	Out...	led_6_	N/A		N/A			LVCNOS18	FA
9	Out...	led_7_	N/A		N/A			LVCNOS18	FA

然后在 pin 栏下输入管脚号，就可以为相对应信号绑定到你指定的管脚了，在 IO Types 栏下指定管脚的类型，一共有以下类型：PCI、LVTTTL、LVCNOS33、LVCNOS33\_5V、LVCNOS25、LVCNOS18 等，根据你的需要指定相应的类型，管脚绑定后见下图：

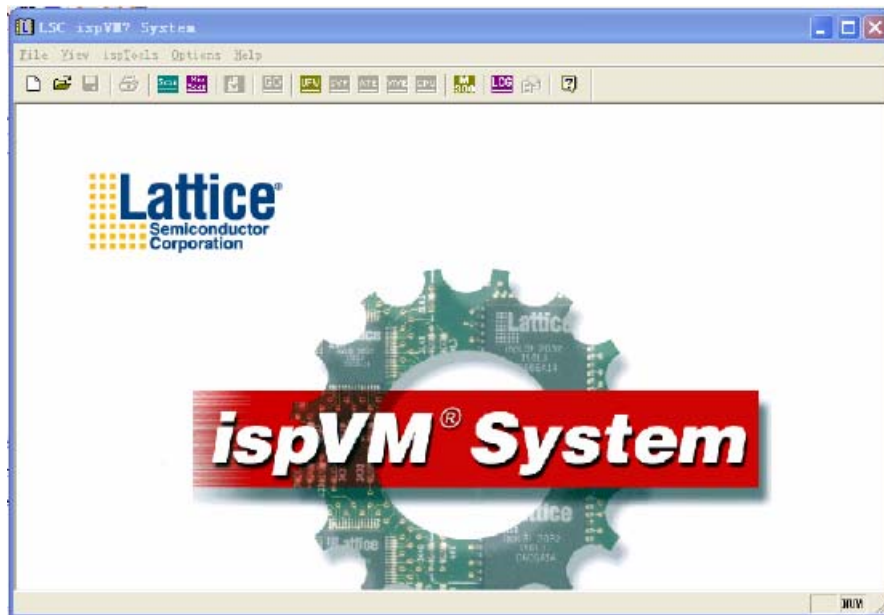
	Type	Signal/Group Name	Group Members	GLE	Macrocell	Pin	Bank	IO Types	SI
1	Clo...	clk_in	N/A		N/A	88	1	LVCNOS33	N/
2	Output	led_0_	N/A	H	2	86	1	LVCNOS33	FA
3	Output	led_1_	N/A	H	4	85	1	LVCNOS33	FA
4	Output	led_2_	N/A	H	6	84	1	LVCNOS33	FA
5	Output	led_3_	N/A	H	8	81	1	LVCNOS33	FA
6	Output	led_4_	N/A	H	10	80	1	LVCNOS33	FA
7	Output	led_5_	N/A	H	12	79	1	LVCNOS33	FA
8	Output	led_6_	N/A	H	13	78	1	LVCNOS33	FA
9	Out...	led_7_	N/A	H	0	87	1	LVCNOS33	FA

然后存盘后退出，回到 ispLEVER project navigator 软件界面，见下图，鼠标右键单击“JEDEC FILE”，然后点击“Force”，可以强制重新编译，以及生成下载类型的文件，接下来就可以把程序下载到芯片中去了。



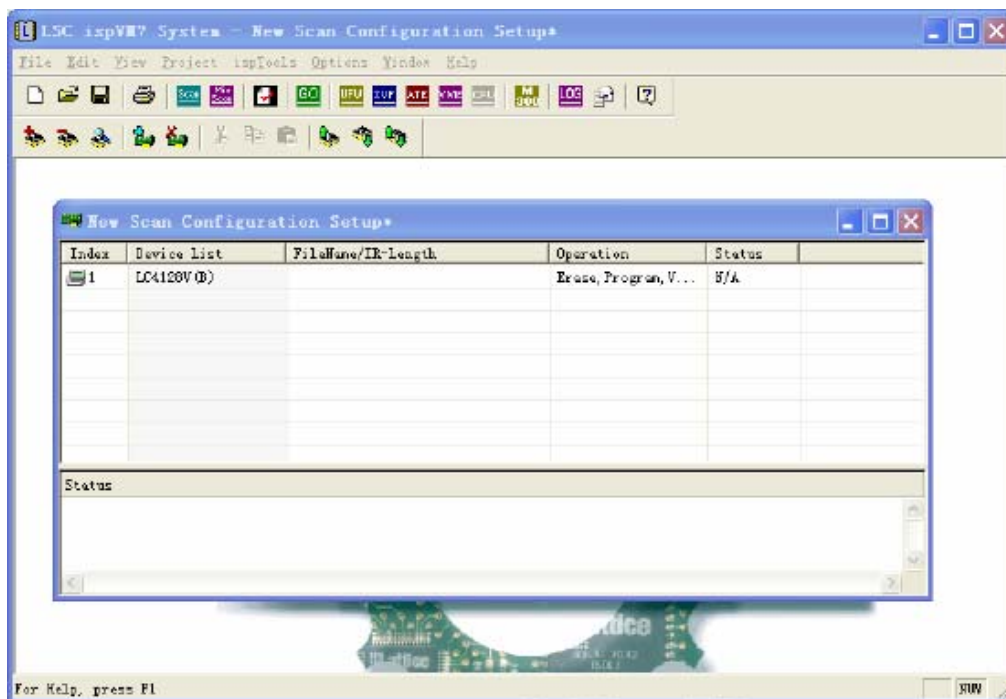
#### 四、下载程序

经过上面步骤的操作，这个项目已经生成下载类型的文件，然后我们就可以下载了，单击“tools->ispVM System”出现以下界面：

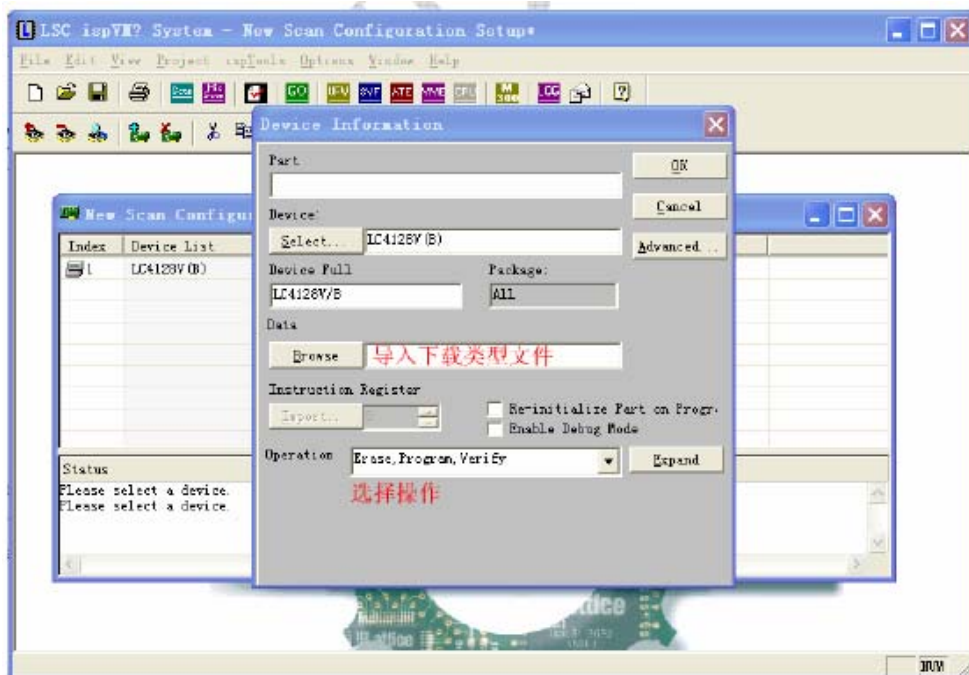


并口驱动已经在安装 ispLEVER 的时候安装好了，若软件提示你没有安装并口驱动，需要重新安装下并口驱动，在 ispTools->Install/Uninstall LSC WinNT/2000/XP/Parallel Port Driver，可以在软件安装盘中找到驱动。

连接好电脑、下载线以及开发板，打开开发板电源，然后单击 ispTools->Scan Chain，就可以扫描到芯片，见下图：

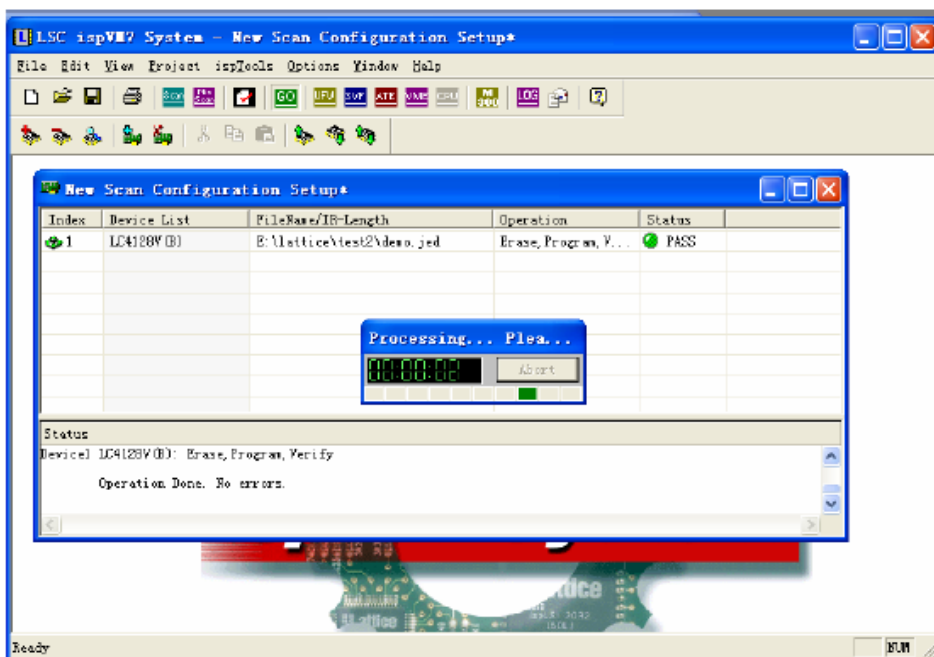


我们可以看到，已经扫描到器件 LC4128V，在 File Name/IR-Length 栏下，第一行的空白处，双击鼠标左键，会出现这个对话框：



在“导入下载类型文件”旁边的 Browse 按钮，就在项目文件夹下导入下载文件，在这里我们导入 demo.jed 文件，在“operation”下拉菜单里，选择要完成的动作，在这里可以选择“Erase,Program,Verify”即“擦除、编程、校验”，然后点击 OK，完成下载文件的导入。

最后一步，点击 Project->Download，开始下载程序：



当下载成功后，在“status”一栏显示“pass”表示下载成功了，这样芯片就可以工作了。

至此，终于完成了一个项目的建立、设计、绑定、下载，这里只是介绍了在做个项目时，在 ispLEVER 软件内必要的操作步骤。ispLEVER 是一个功能非常强大的软件。其余的操作和功能，限于篇幅，这里就不做介绍了。