ispLEVER 培 训 教 程

上 海 莱 迪 思 半 导 体 公 司 市 场 部

2002 年 12 月

英文网址： http://www.latticesemi.com
中文网址： http://www.latticesemi.com.cn **或** http://www.lattice.com.cn

目 录

第 一 节 ispLEVER 简 介

ispLEVER 是 Lattice 公司最新推出的一套 EDA 软件。设计输入可采用原理图、硬件描述语言、混合输入三种方式。能对所设计的数字电子系统进行功能仿真和时序仿真。编译器是此软件的核心，能进行逻辑优化，将逻辑映射到器件中去，自动完成布局与布线并生成编程所需要的熔丝图文件。软件中的 Constraints Editor 工具允许经由一个图形用户接口选择 I/O 设置和引脚分配。软件包含 Synolicity 公司的“Synplify”综合工具和 Lattice 的 ispVM 器件编程工具。ispLEVER 软件提供给开发者一个简单而有力的工具，用于设计所有莱迪思可编程逻辑产品。软件支持所有 Lattice 公司的 ispLSI 、MACH、ispGDX、ispGAL、GAL 器件。ispLEVER 工具套件还支持莱迪思新的 ispXPGA™和 ispXPLD™产品系列，并集成了莱迪思 ORCA Foundry 设计工具的特点和功能。这使得 ispLEVER 的用户能够设计新的 ispXPGA 和 ispXPLD 产品系列，ORCA FPGA/FPSC 系列和所有莱迪思的业界领先的 CPLD 产品而不必学习新的设计工具。

软件主要特征：

1. 输入方式
   * 原理图输入
   * ABEL-HDL 输入
   * VHDL 输入
   * Verilog-HDL 输入
   * 原理图和硬件描述语言混合输入

2. 逻辑模拟
   * 功能模拟
   * 时序模拟

3. 编译器
   * 结构综合、映射、自动布局和布线

4. 支持的器件
   * 含有支持 ispLSI 器件的宏库及 MACH 器件的宏库、TTL 库
   * 支持所有 ispLSI、MACH、ispGDX、ispGAL、GAL、ORCA FPGA/FPSC、ispXPGA 和 ispXPLD 器件

5. Constraints Editor 工具
   * I/O 参数设置和引脚分配

6. ispVM 工具
   * 对 ISP 器件进行编程

软件支持的计算机平台：

PC: Windows 98/NT/2000/XP

第二节 ispLEVER 开发工具的原理图输入

I. 启动 ispLEVER(按 Start=>Programs=>Lattice Semiconductor=>ispLEVER 菜单)

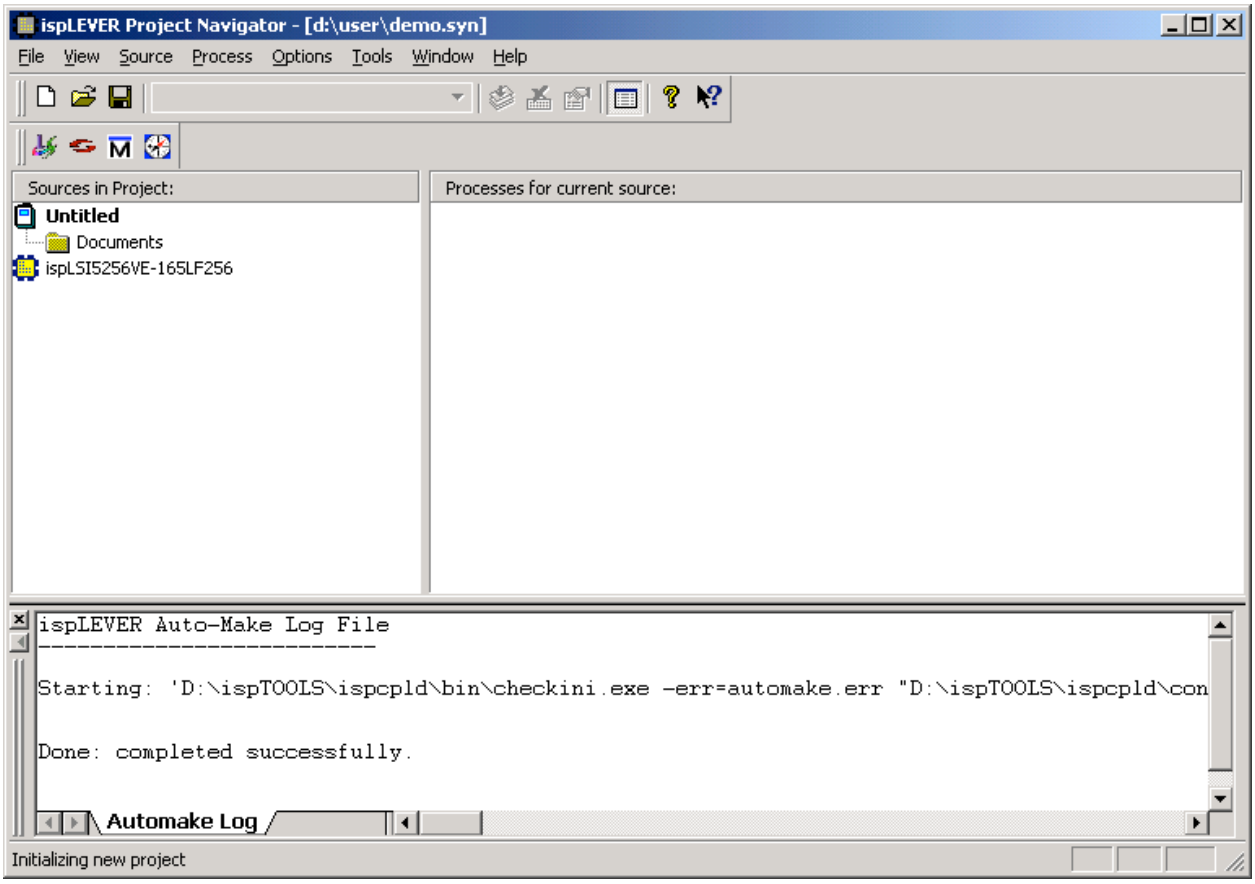
II. 创建一个新的设计项目

A. 选择菜单 File。

B. 选择 New Project...。

C. 在 Create New Project 对话框的 Project Name 栏中，键入项目名 d:\user\demo.syn。在 Project type 栏中选择 Schematic/ABEL（ispLEVER 软件支持 Schematic/ABEL、Schematic/VHDL、Schematic/Verilog 等的混合设计输入，在此例中，仅有原理图输入，因此可选这三种中的任意一种）。

D. 你可以看到默认的项目名和器件型号: Untitled and ispLSI5256VE-165LF256。



III. 项目命名

A. 用鼠标双击 Untitled。

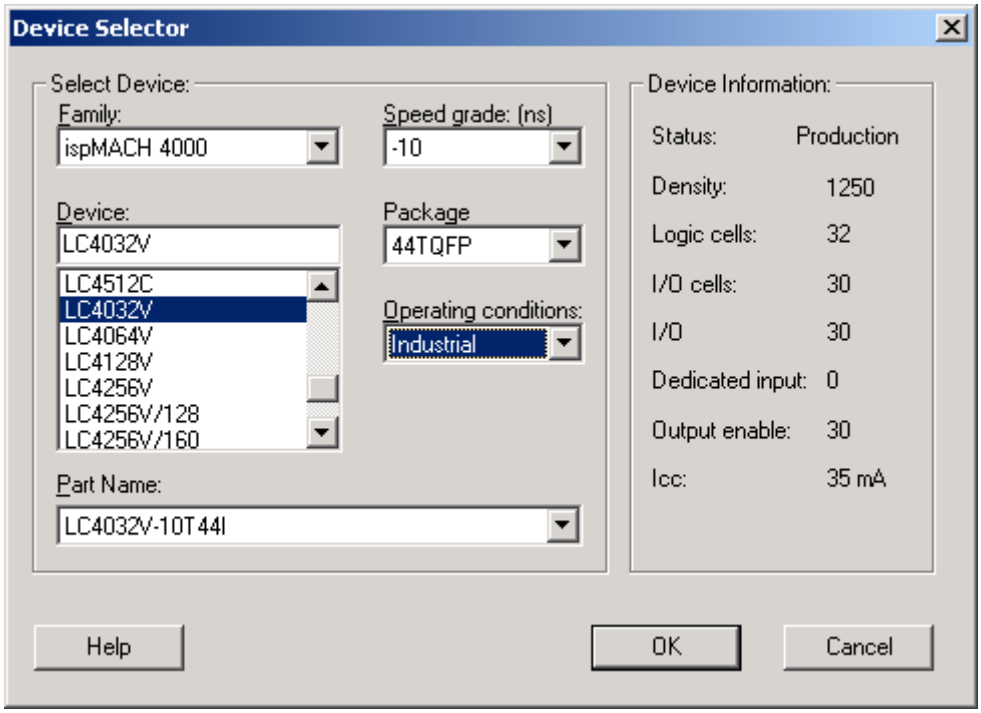
B. 在 Title 文本框中输入“Demo Project”,并选 OK。

IV. 选择器件

A. 双击 ispLSI5256VE-165LF256, 你会看到 Device Selector 对话框(如下图所示)。

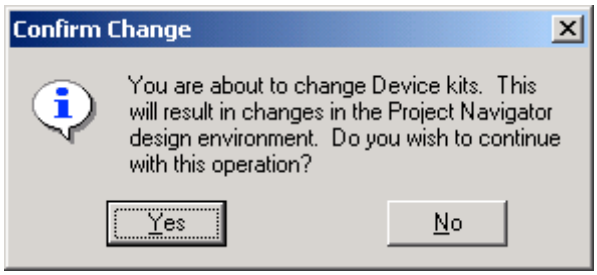
B. 在 Select Device 窗口中选择 ispMACH 4000 项。

C. 按动器件目录中的滚动条，直到找到并选中器件 LC4032V-10T44I。

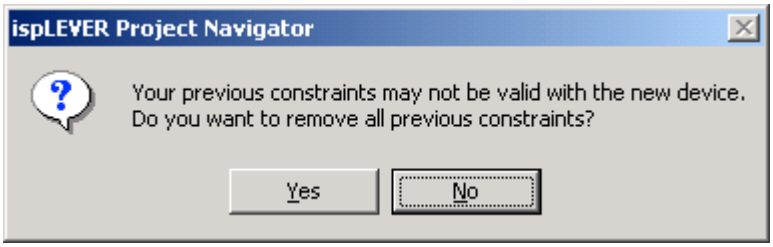


D. 撳 OK 按钮，选择这个器件。

E. 在软件弹出的如下图显示的 Confirm Change 窗口中，按 Yes 按钮。



F. 因改选器件型号后，先前的约束条件可能对新器件无效，因此在软件接着弹出的如下图显示的 ispLEVER Project Navigato 窗口中，按 Yes 按钮,以用来去除原有的约束条件。



V. 在设计中增加源文件

一个设计项目由一个或多个源文件组成。这些源文件可以是原理图文件(*.sch)、ABEL HDL 文件(*.abl)、VHDL 设计文件(*.vhd)、Verilog HDL 设计文件(*.v)、测试向量文件(*.abv)

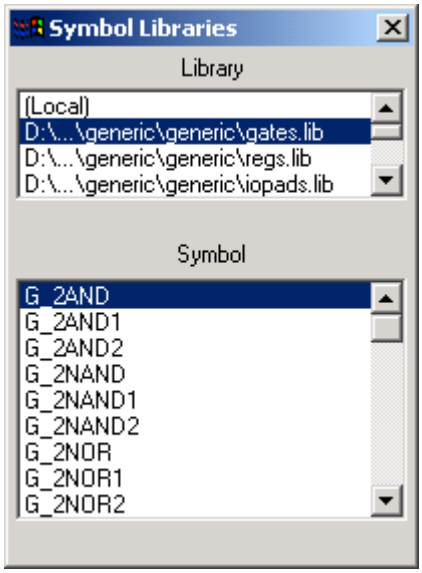
或者是文字文件(*.doc, *.wri, *.txt)。在以下操作步骤中，你要在设计项目中添加一张空白的原理图纸。

A. 从菜单上选择 Source 项。
B. 选择 New... 。
C. 在对话框中，选择 Schematic(原理图)，并按 OK。
D. 输入文件名 demo.sch。
E. 确认后撳 OK。

VI. 原理图输入

你现在应该进入原理图编辑器。在下面的步骤中，你将要在原理图中画上几个元件符号，并用引线将它们相互连接起来。

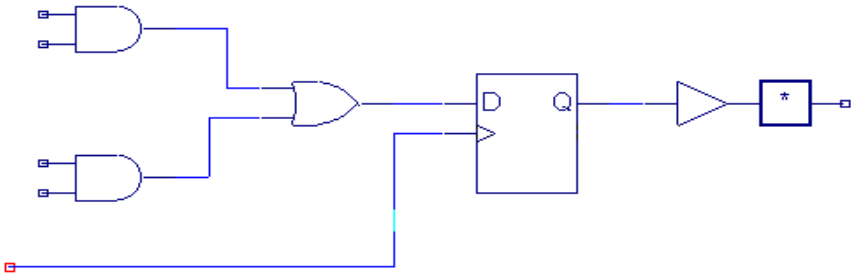
A. 从菜单栏选择 Add, 然后选择 Symbol，你会看到如下图所示的对话框：



B. 选择 GATES.LIB 库，然后选择 G_2AND 元件符号。
C. 将鼠标移回到原理图纸上，注意此刻 AND 门粘连在你的光标上，并随之移动。
D. 单击鼠标左键，将符号放置在合适的位置。
E. 再在第一个 AND 门下面放置另外一个 AND 门。
F. 将鼠标移回到元件库的对话框，并选择 G_2OR 元件。
G. 将 OR 门放置在两个 AND 门的右边。
H. 现在选择 Add 菜单中的 Wire 项。
I. 单击上面一个 AND 门的输出引脚，并开始画引线。
J. 随后每次单击鼠标，便可弯折引线（双击便终止连线)。
K. 将引线连到 OR 门的一个输入脚。
L. 重复上述步骤，连接下面一个 AND 门。

VII. 添加更多的元件符号和连线

A. 采用上述步骤，从 REGS.LIB 库中选一个 g_d 寄存器，并从 IOPADS.LIB 库中选择 G_OUTPUT 符号。
B. 将它们互相连接，实现如下的原理图：



VIII. 完成你的设计

在这一节，通过为连线命名和标注 I/O Markers 来完成原理图。

当要为连线加信号名称时，你可以使用 ispLEVER 的特点，同时完成两件事-----同时添加连线和连线的信号名称。这是一个很有用的特点，可以节省设计时间。I/O Markers 是特殊的元件符号，它指明了进入或离开这张原理图的信号名称。注意连线不能被悬空(dangling)，它们必需连接到 I/O Marker 或逻辑符号上。这些标记采用与之相连的连线的名字，与 I/O Pad 符号不同，将在下面定义属性(Add Attributes)的步骤中详细解释。

A. 为了完成这个设计，选择 Add 菜单中的 Net Name 项。

B. 屏幕底下的状态栏将要提示你输入的连线名，输入‘A’并按 Enter 键，连线名会粘连在鼠标的光标上。

C. 将光标移到最上面的与门输入端，并在引线的末连接端(也即输入脚左端的红色方块)，按鼠标左键，并向左边拖动鼠标。这可以在放置连线名称的同时，画出一根输入连线。

D. 输入信号名称现在应该是加注到引线的末端。

E. 重复这一步骤，直至加上全部的输入‘B’,’C’,’D’和‘CK’，以及输出‘OUT’。

F. 现在选择 Add 菜单的 I/O Marker 项。

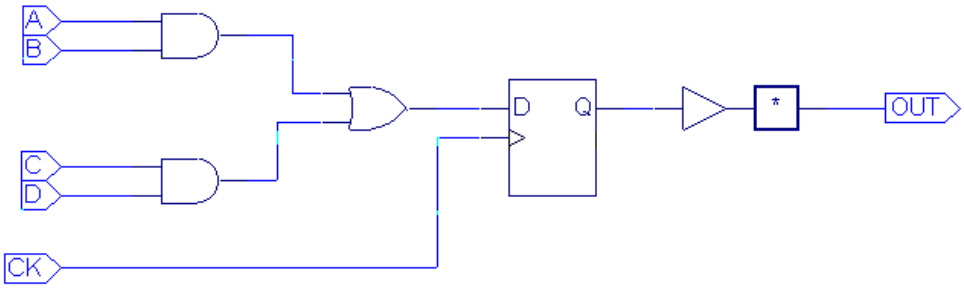
G. 将会出现一个对话框，请选择 Input。

H. 将鼠 标 的光标移至输入连线的末端(位于连线和连线名之间)，并单击鼠标的左键。这时回出现一个输入 I/O Marker，标记里面是连线名。

I. 鼠标移至下一个输入，重复上述步骤，直至所有的输入都有 I/OMarker。

J. 现在请在对话框中选择 Output,然后单击输出连线端，加上一个输出 I/O Marker。

K. 至此原理图就基本完成，它应该如下图所示。



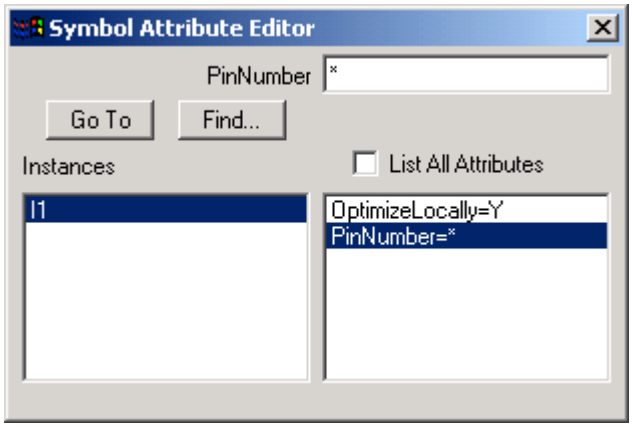
IX. 定义器件的属性(Attributes)

你可以为任何一个元件符号或连线定义属性。在这个例子中，你可以为输出端口符号添加引脚锁定 LOCK 的属性。请注意，在 ispLEVER 中，引脚的属性实际上是加到 I/O Pad 符号上，而不是加到 I/O Marker 上。同时也请注意，只有当你需要为一个引脚增加属性时，才需要 I/O Pad 符号，否则，你只需要一个 I/O Marker。

A. 在菜单条上选择 Edit =>Attribute =>Symbol Attribute 项，这时会出现一个 Symbol Attribute Editor 对话框。

B. 单击需要定义属性的输出 I/O Pad。

C. 对话框里会出现一系列可供选择的属性。



D. 选择 PinNumber 属性，并且把文本框中的‘*’替换成‘4’（ ‘4’ 为器件的引脚号）。这样，该 I/O Pad 上的信号就被锁定到器件的第四个引脚上了。

E. 关闭对话框。

F. 请注意，此时数字‘4’出现在 I/O Pad 符号内。

X. 保存已完成的设计

从菜单条上选择 File，并选 Save 命令。再选 Exit 命令。

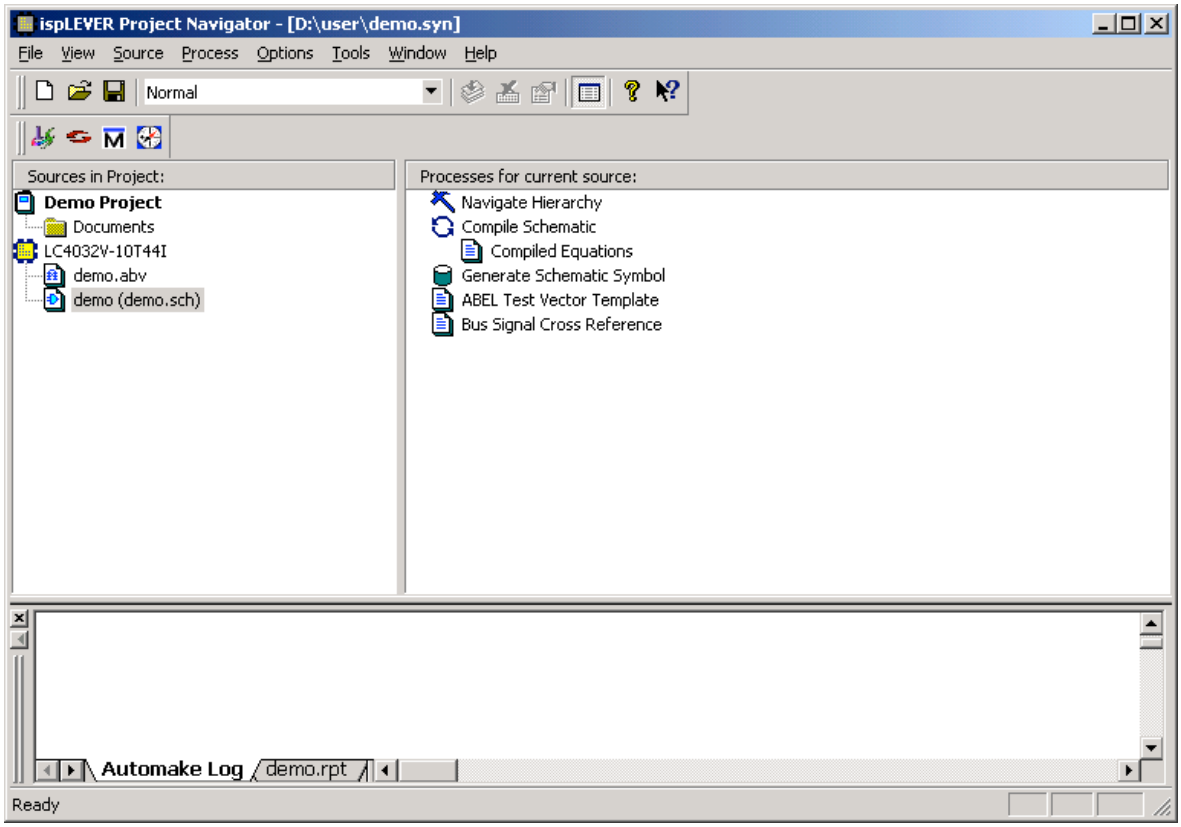
第三节　设计的编译与仿真

I.　建立仿真测试向量(Simulation Test Vectors)

A.　在已选择 LC4032V-10T44I 器件的情况下，选择 Source 菜单中的 New...命令。
B.　在对话框中，选择 ABEL Test Vectors 并按 OK。
C.　输入文件名 *demo.abv* 作为你的测试向量文件名。
D.　按 OK。
E.　文本编辑器弹出后，输入下列测试向量文本：
F.　完成后，选择 File 菜单中的 Save 命令，以保留你的测试向量文件。
G.　再次选择 File，并选 Exit 命令。

```
module demo;

c,x = .c.,.x.;

CK,A,B,C,D,OUT      PIN;

TEST_VECTORS
([CK,   A,     B,   C,   D]->[OUT])
 [  c  ,   0   , 0 ,  0 , 0 ]->[ x ];
 [  c  ,   0   , 0 ,  1 , 0 ]->[ x ];
 [  c  ,   1   , 1 ,  0 , 0 ]->[ x ];
 [  c  ,   0   , 1 ,  0 , 1 ]->[ x ];
 END
```

H.　此时你的项目管理器(Project Navigator)应如下图所示。



II. 编译原理图与测试向量

现在你已为你的设计项目建立起所需的源文件，下一步是执行每一个源文件所对应的处理过程。选择不同的源文件，你可以从项目管理器窗口中观察到该源文件所对应的可执行过程。在这一步，请你分别编译原理图和测试向量。

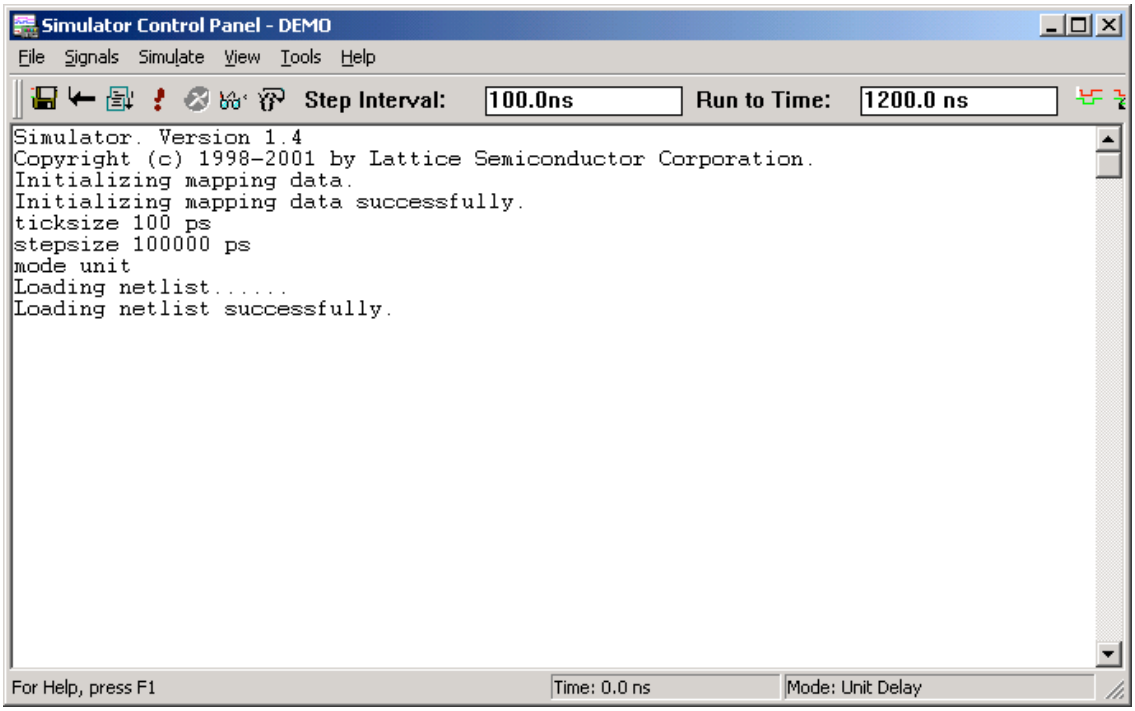
A. 在项目管理器左边的项目源文件(Sources in Project)清单中选择原理图(demo.sch)。
B. 双击原理图编译(Compile Schematic)处理过程。
C. 编译通过后，Compile Schematic 过程的左边会出现一个绿色的查对记号，以表明编译成功。编译结果将以逻辑方程的形式表现出来。
D. 然后从源文件清单中选择测试向量源文件(demo.abv)。
E. 双击测试向量编译(Compile Test Vectors)处理过程。

III. 设计的仿真

ispLEVER 开发系统不但可以进行功能仿真(Functional Simulation)，而且可以进行时序仿真(Timing Simulation)。在仿真过程中还提供了单步运行、断点设置功能。

IV. 一、功能仿真

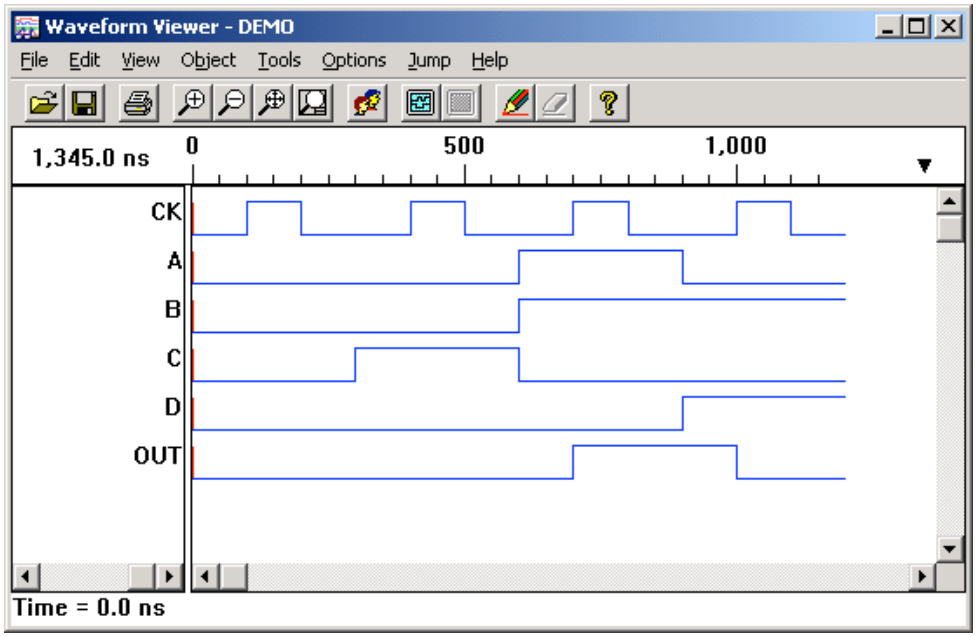
A. 在 ispLEVER Project Navigator 的主窗口左侧，选择测试向量源文件(demo.abv)，双击右侧的 Functional Simulation 功能条。将弹出如下图所示的仿真控制窗口(Simulator Control Panel)。



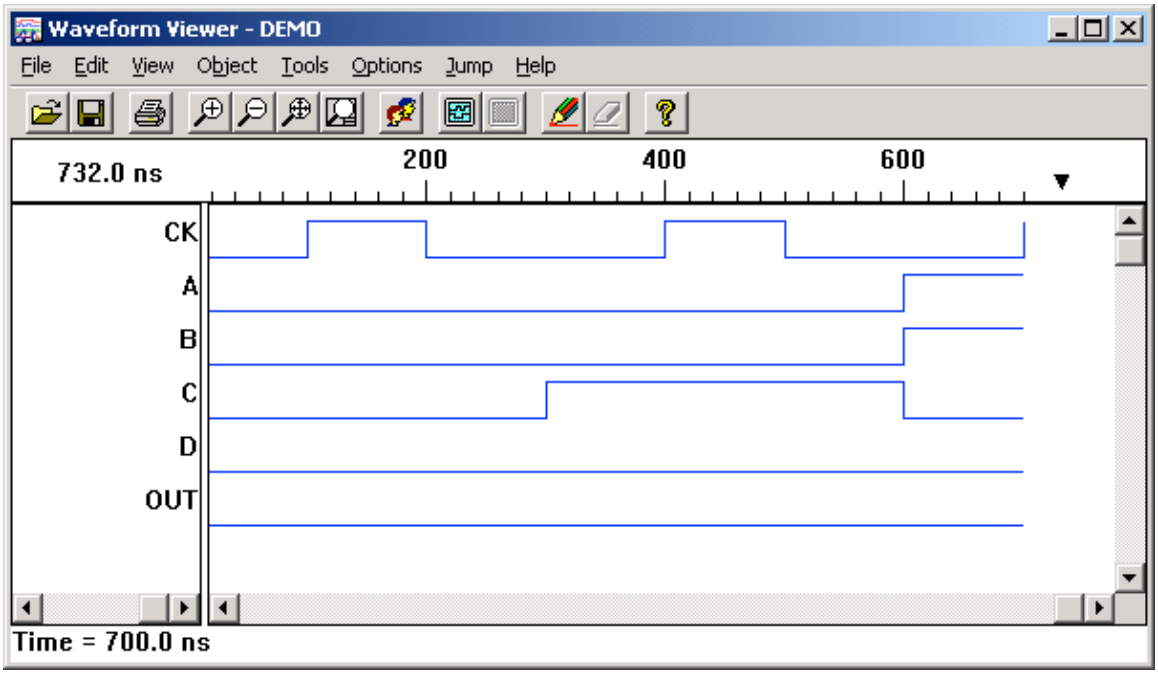
B. 在 Simulator Control Panel 中，将根据(*.abv)文件中所给出的输入波形进行一步到位的仿真。

在 Simulator Control Panel 中，按 Simulate=>Run,再按 Tools => Waveform Viewer 菜单，将打开波形观察器 Waveform Viewer 如下图所示。

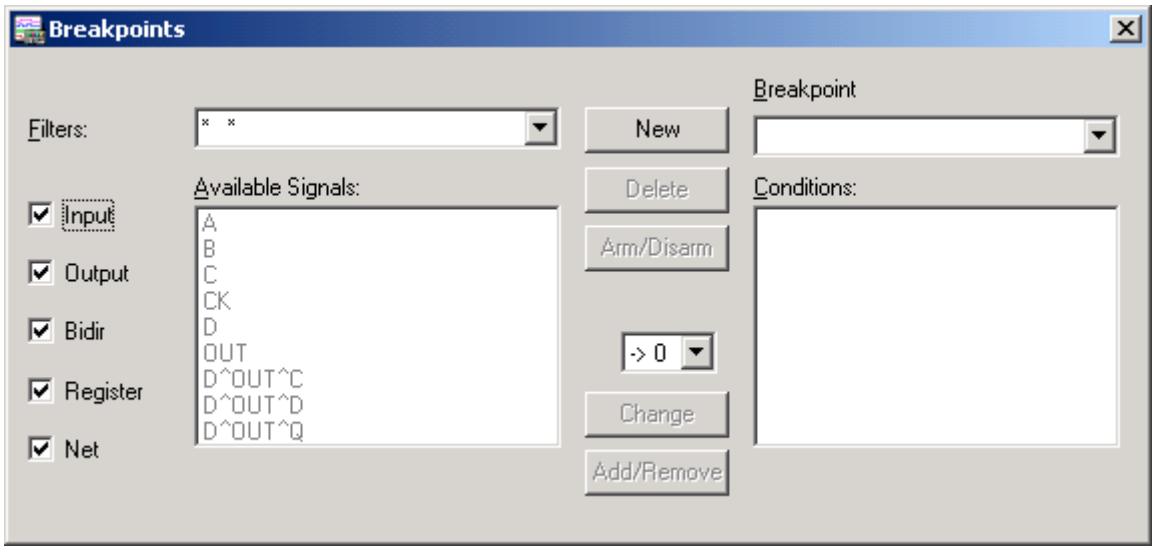
C. 波形现在都显示在波形观察器的窗口中，如下图所示：



D. 单步仿真。选 Simulator Control Panel 窗口中的 Simulate=>Step 可对您的设计进行单步仿真。ispLEVER 中仿真器的默认步长为 100ns，您可根据需要在按 Simulate=>Settings 菜单所激活的对话框(Setup Simulator)中重新设置您所需要的步长。按 Simulator Control Panel 窗口中的 Simulate=>Reset 菜单，可将仿真状态退回至初始状态(0 时刻)。随后，每按一次 Step，仿真器便仿真一个步长。下图是按了七次 Step 钮后所显示的波形(所选步长为 100ns)。



E. 设置断点(Breakpoint)。在 Simulator Control Panel 窗口中，按 Signal=>Breakpoints 菜单，会显示如下图所示的断点设置控制的 Breakpoint 窗口。



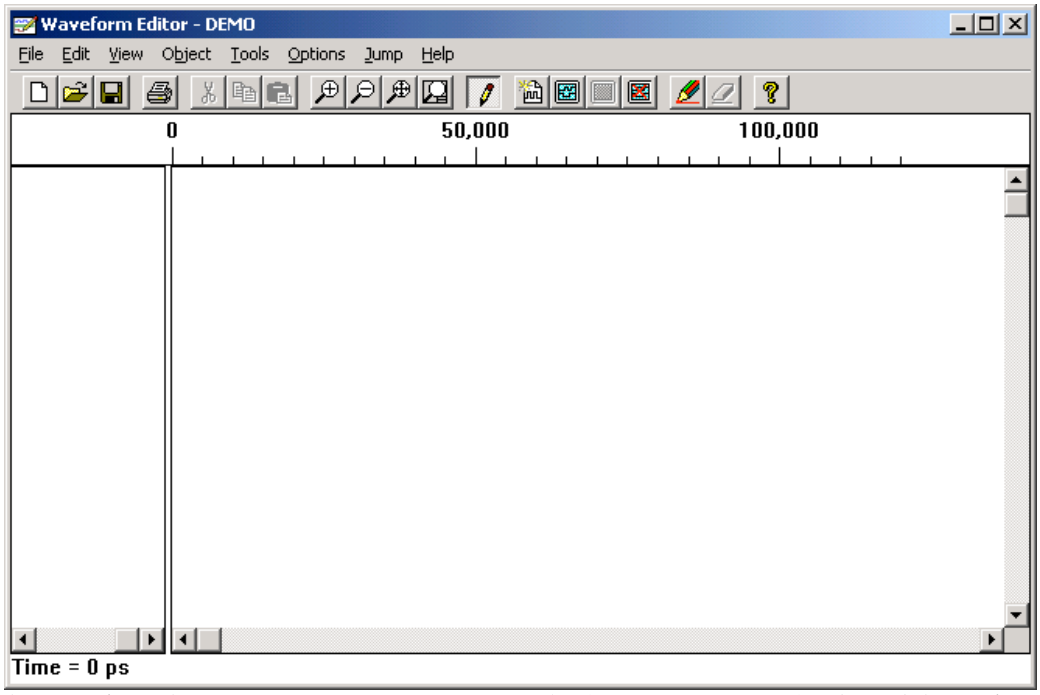
在该窗口中按 New 按钮，开始设置一个新的断点。在 Available Signals 栏中单击鼠标选择所需的信号，在窗口中间的下拉滚动条中可选择设置断点时该信号的变化要求，例如：->0，指该信号变化到 0 状态；!=1，指该信号处于非 1 状态。一个断点可以用多个信号所处的状态来作为定义条件，这些条件在逻辑上是与的关系。最后在 Breakpoints 窗口中,先选中 ADD，再按 Arm 按钮使所设断点生效。本例中选择信号

OUT->?作为断点条件，其意义是指断点条件成立的条件为 OUT 信号发生任何变化(变为 0，1，Z 或 X 状态)。这样仿真过程中在 0ns,700ns,1000ns 时刻都会遇到断点。

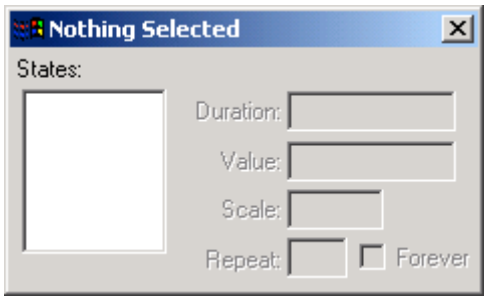
F. 波形编辑(Waveform Edit)。

除了用*.abv 文件描述信号的激励波形外，ispLEVER 还提供了直观的激励波形的图形输入工具—Waveform Editor。以下是用 Waveform Editor 编辑激励波形的步骤(仍以设计 demo.sch 为例)：

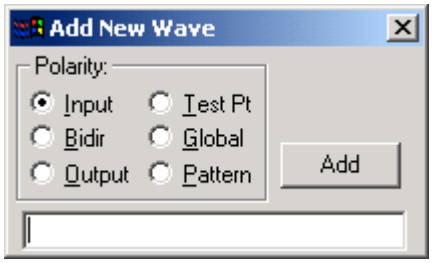
1. 在 Simulator Cotrol Panel 窗口中，按 Tools=>Waveform Editor 菜单，进入波形编辑器窗口(Waveform Editing Tool)，如下图所示：



2. 在上述窗口中按 Object=>Edit Mode，将弹出如下图所示的波形编辑子窗口：

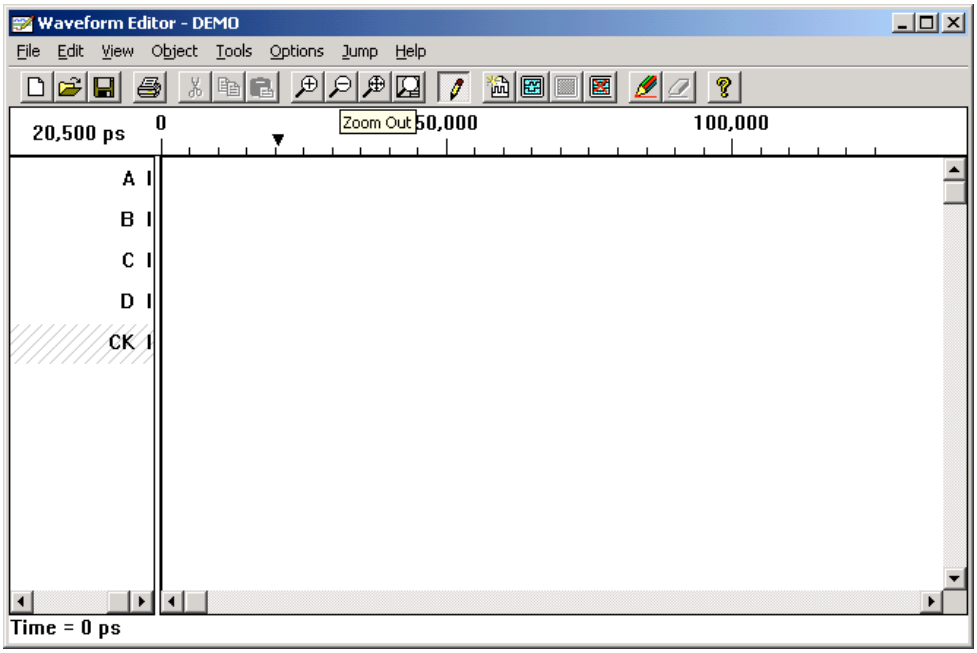


3. 在 Waveform Editing Tool 窗口中按 Edit=>New Wave 菜单，弹出如下窗口：

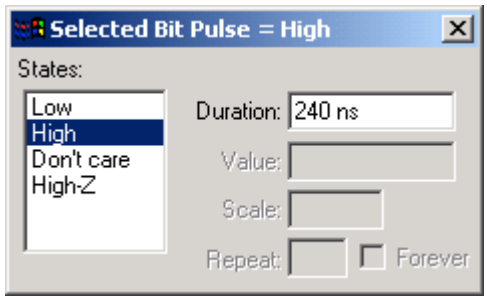


在该窗口中的 Polarity 选项中选择 Input，然后在窗口下部的空格中输入信号名：A，B，C，D，CK。每输完一个信号名按一次 Add 钮。

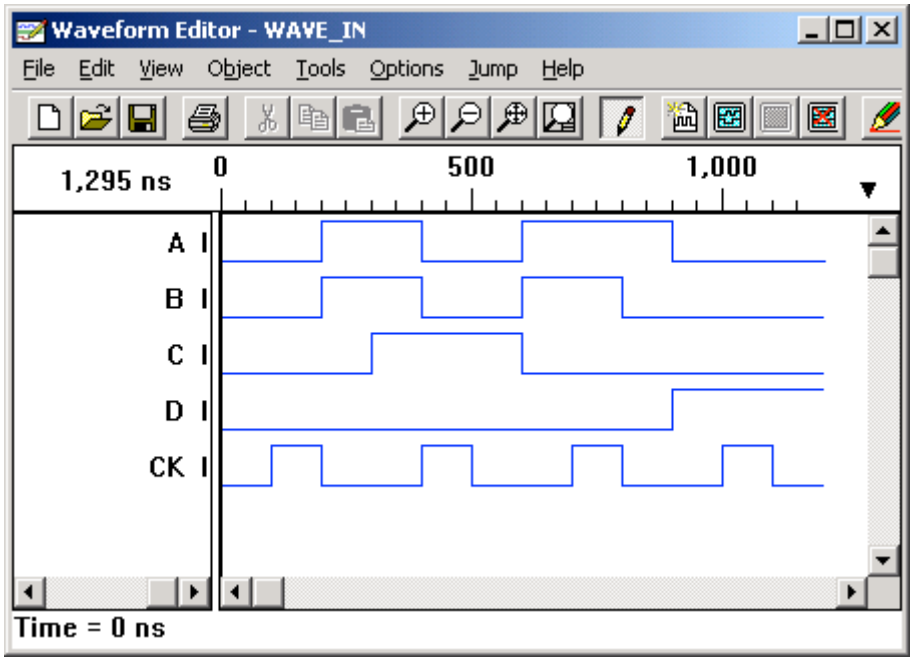
4. 在完成上述步骤 3 以后，Waveform Editing Tool 窗口中有了 A，B，C，D，CK 的信号名，如下图所示：



单击窗口左侧的信号名 A，开始编辑 A 信号的激励波形。单击 0 时刻右端且与 A 信号所处同一水平位置任意一点，波形编辑器子窗口中将显示如下信息：



在 States 栏中选择 Low，在 Duration 栏中填入 200ns 并按回车键。这时，在 Waveform Editing Tool 窗口中会显示 A 信号在 0-200ns 区间为 0 的波形。然后在 Waveform Editing Tool 窗口中单击 200ns 右侧区间任一点，可在波形编辑器的子窗口中编辑 A 信号的下一个变化。重复上述操作过程，编辑所有输入信号 A，B，C，D，CK 的激励波形，并将它存盘为 wave_in.wdl 文件。完成后，Waveform Editing Tool 窗口如下图所示：



5. 在 Waveform Editing Tool 菜单中，按 File=>Consistency Check 菜单，检测激励波形是否存在冲突。在该例中，错误信息窗口会提示 No Errors Dected。
6. 至此，激励波形已描述完毕，剩下的工作是调入该激励文件(wave_in.wdl)进行仿真：

   回到 ispLEVER Project Navigator 主窗口，按 Source=>Import 菜单，调入激励文件 wave_in.wdl。在窗口左侧的源程序区选中 Wave_in.wdl 文件，双击窗口右侧的 Functional Simulation 栏进入功能仿真流程，以下的步骤与用*.abv 描述激励的仿真过程完全一致，在此不再赘述。

二、时序仿真(Timing Simulation)

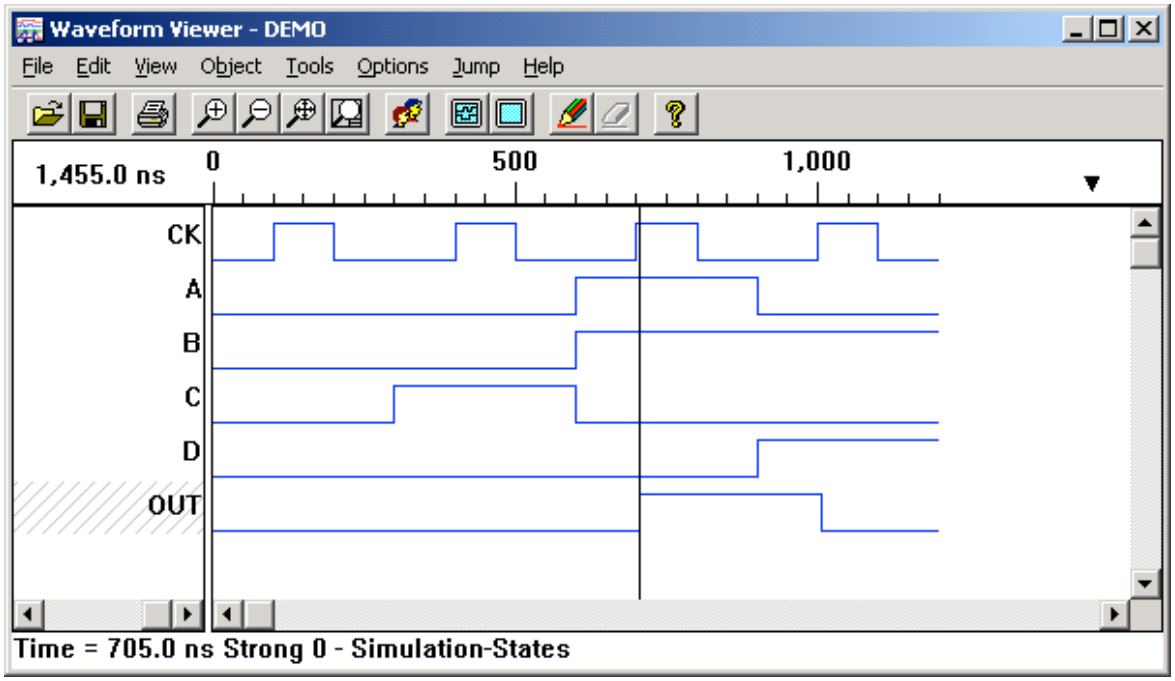
时序仿真的操作步骤与功能仿真基本相似，以下简述其操作过程中与功能仿真的不同之处。

仍以设计 Demo 为例，在 ispLEVER Project Navigator 主窗口中，在左侧源程序区选中 Demo.abv，双击右侧的 Timing Simulation 栏进入时序仿真流程。由于时序仿真需要与所选器件有关的时间参数，因此双击 Timing Simulation 栏后，软件会自动对器件进行适配，然后打开与功能仿真时间相同的 Simulator Control Panel 窗口。

时序仿真与功能仿真操作步骤的不同之处在于仿真的参数设置上。在时序仿真时，打开 Simulator Control Panel 窗口中的 Simulate=>Settings 菜单，产生 Setup Simulator 对话框。在此对话框中可设置延时参数(Simulation Delay)最小延时(Minimun Delay)、典型延时(Typical Delay)、最大延时(Maximun Delay)和 0 延时(Zero Delay)。最小延时是指器件可能的最小延时时间，0 延时指延时时间为 0。

在 Setup Simulator 对话框中，仿真模式(Simulation Mode)可设置为两种形式：惯性延时(Inertial Mode)和传输延时(Transport Mode)。

将仿真参数设置为最大延时和传输延时状态，在 Waveform Viewer 窗口中显示的仿真结果如下图所示：



由图可见，与功能仿真不同的是：输出信号 OUT 的变化比时钟 CK 的上升沿滞后了 5ns。

IV. 建立元件符号(Symbol)

ispLEVER 工具的一个非常有用的特点是能够迅速地建立起一张原理图的符号。通过这一步骤，你可以建立一个可供反复调用的逻辑宏元件，以便放置在更高一层的原理图纸上。下一节将指导你如何调用。这里仅教你如何建立元件符号。

A. 双击原理图的资源文件 demo.sch，把它打开。
B. 在原理图编辑器中，选择 File 菜单。
C. 从下拉菜单中，选择 Matching Symbol 命令。
D. 关闭原理图。
E. 至此，这张原理图的宏元件符号已经建立完毕，并且被加到元件表中。你可以在下一节中调用这个元件。

第四节　硬件描述语言和原理图混合输入

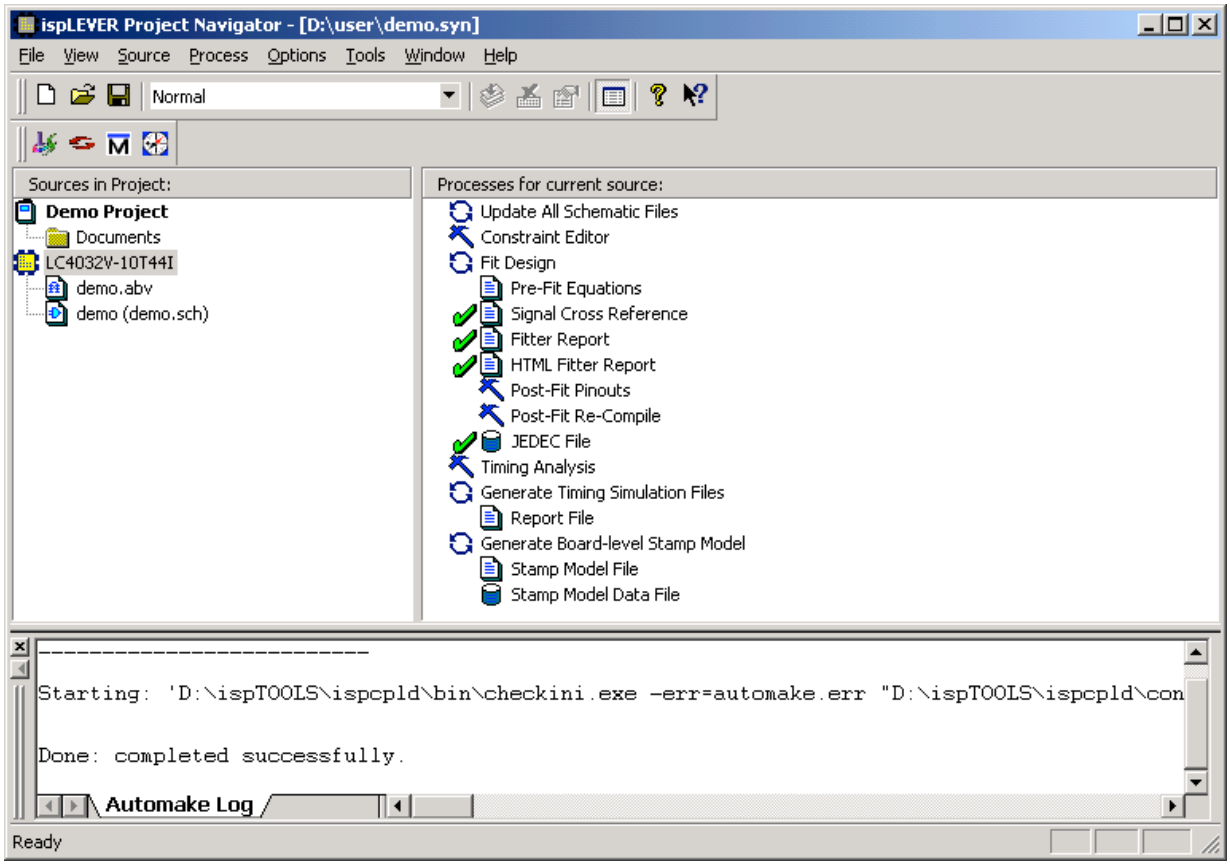
ispLEVER 软件支持 ABEL/原理图、VHDL/原理图、Verilog/原理图的混合输入。这一节，我们以 ABEL/原理图为例，介绍硬件描述语言和原理图混合输入的方法。

现在，你要建立一个简单的 ABEL HDL 语言输入的设计，并且将其与上一节中完成的原理图进行合并，以层次结构的方式，画在顶层的原理图上。然后对这个完整的设计进行仿真、编译，最后适配到器件中。

现在我们就开始吧！

I　启动 ispLEVER

如果你在上一节的练习后退出了 ispLEVER，点击 Start => Programs => LatticeSemiconductor => ispLEVER 菜单，屏幕上你的项目管理器应该如下图所示。



II　建立顶层的原理图

A.　仍旧选择 LC4032V-10T44I 器件，从菜单条上选 Source。

B.　选择 New...

C.　在对话框中选 Schematic，并按 OK。

D.　在文本框中输入文件名 top.sch，并按 OK。

E.　现在你就进入了原理图编辑器。

F. 调用上节中创建的元件符号。选择 Add 菜单中的 Symbol 项，这时会出现 Symbol Libraries 对话框，选择 Local 的库，你会注意到在下部的文本框中有一个叫 demo 的元件符号，这就是你在上一节中自行建立的元件符号。

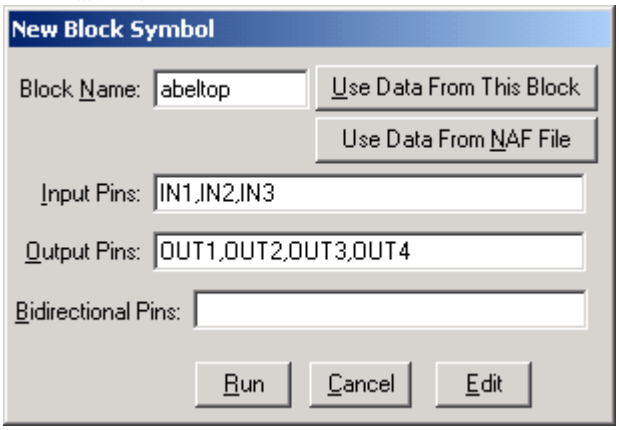
G. 选择 demo 元件符号，并放到原理图上的合适位置。

II. 建立内含 ABEL 语言的逻辑元件符号

现在你要为 ABEL HDL 设计文件建立一个元件符号。只要知道了接口信息，你就可以为下一层的设计模块创建一个元件符号。而实际的 ABEL 设计文件可以在以后再完成。

A. 在原理图编辑器里，选择 ADD 菜单里的 New Block Symbol...命令。

B. 这时候会出现一个对话框，提示你输入 ABEL 模块名称及其输入信号名和输出信号名。请按照下图所示输入信息：

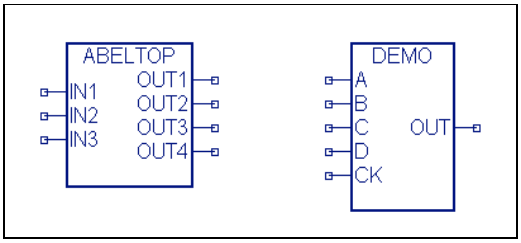


C. 当你完成信号名的输入，撳 Run 按钮，就会产生一个元件符号，并放在本地元件库中。同时元件符号还粘连在光标上，随之移动。

D. 把这个符号放在 demo 符号的左边。

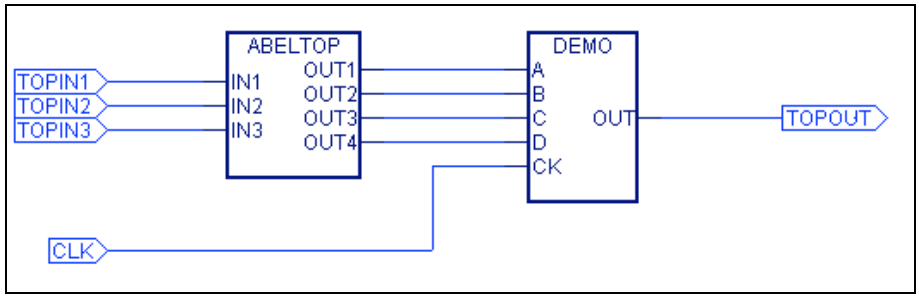
E. 单击鼠标右键，就会显示 Symbol Libraries 的对话框。
请注意 abeltop 符号出现在 Local 库中。

F. 关闭对话框。你的原理图应该如下图所示：



III. 完成原理图

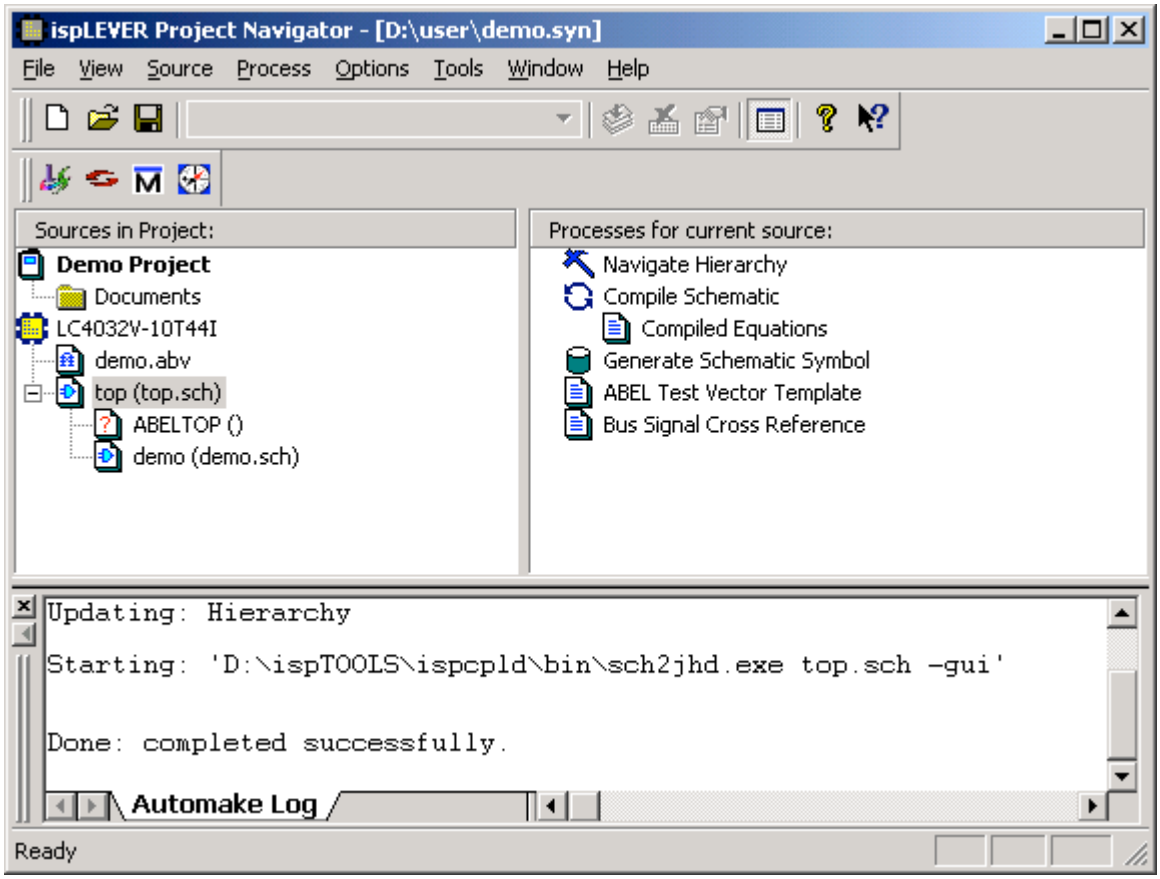
现在请你添加必需的连线，连线名称，以及 I/O 标记，来完成顶层原理图，使其看上去如下图所示。如果你需要帮助，请参考第二节中有关添加连线和符号的指导方法。当你画完后，请存盘再退出。



IV. 建立 ABEL-HDL 源文件

现在你需要建立一个 ABEL 源文件，并把它链接到顶层原理图对应的符号上。项目管理器使这些步骤简化了：

A. 你当前的管理器应该如下图所示：

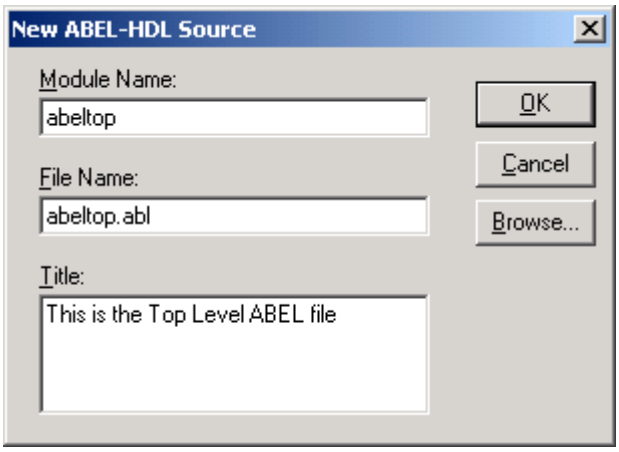


B. 请注意 abeltop 左边的红色“?”图标。这意味着目前这个源文件还是个未知数，因为你还没有建立它。同时也请注意源文件框中的层次结构，abeltop 和 demo 源文件位于 top 原理图的下面并且偏右，这说明它们是 top 原理图的底层源文件。这也是 ispLEVER 项目管理器另外一个有用的特点。

C. 为了建立所需的源文件，请选择 abeltop，然后选择 Source 菜单中的 New...命令。

D. 在 New Source 对话框中，选择 ABEL-HDL Module 并按 OK。

E. 下一个对话框会问你模块名，文件名，以及模块的标题。为了将源文件与符号相链接，模块名必须与符号名一致，而文件名没有必要与符号名一致。但为了简单，你可以给它们取相同的名字。按下图所示，填写相应的栏目：



F. 按 OK。你就进入了 Text Editor，而且可以看到 ABEL HDL 设计文件的框架已经呈现在你的面前。

G. 输入下列的代码。确保你的输入代码位于 TITLE 语句和 END 语句之间。

```
MODULE abeltop

TITLE 'This is the Top Level ABEL file'

" Inputs
IN1,IN2,IN3 pin;

"Outputs
OUT1,OUT2,OUT3,OUT4            pin;

Equations
OUT1=IN1 & !IN3;
OUT2=IN1 & !IN2;
OUT3=!IN1 & IN2 & IN3;
OUT4=IN2 & IN3;
END
```

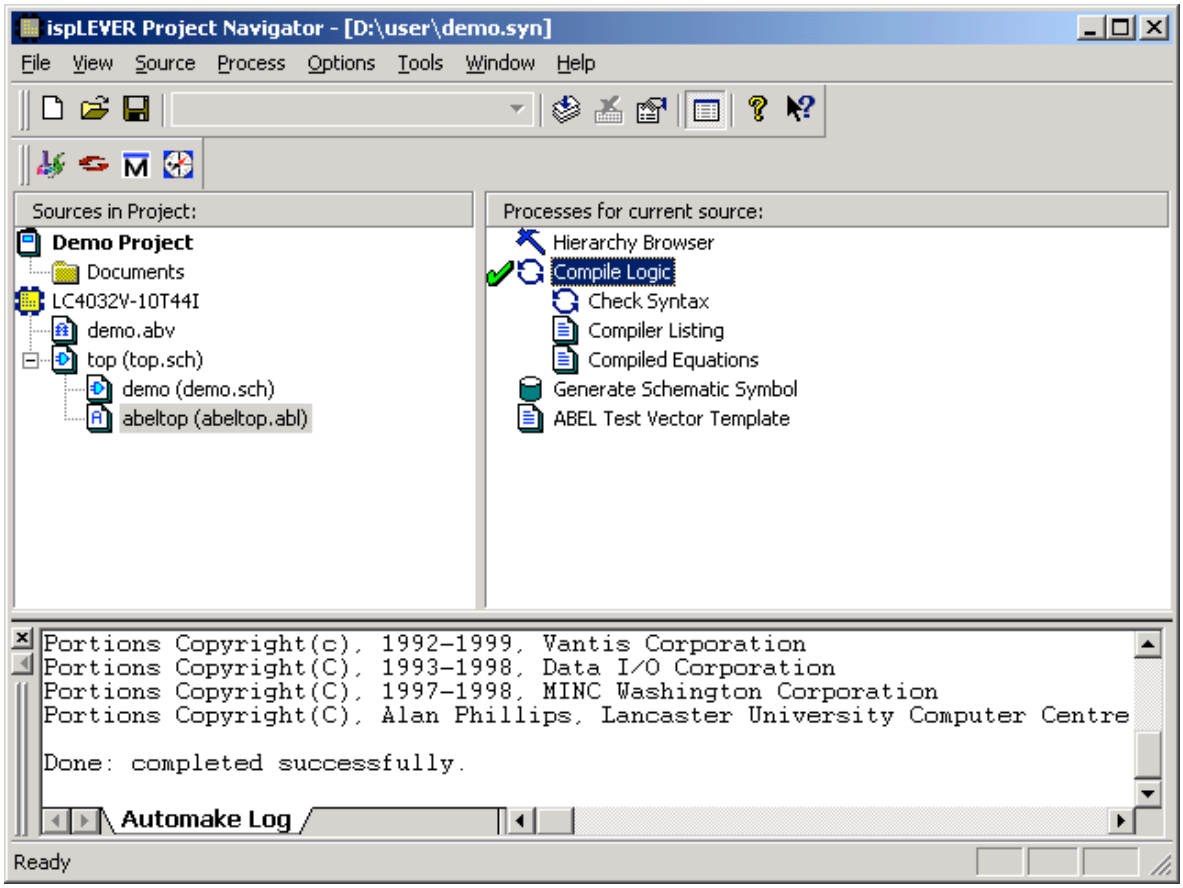
H. 当你完成后，选择 File 菜单中的 Save 命令。

I. 退出文本编辑器。

J. 请注意项目管理器中 abeltop 源文件左边的图标已经改变了。这就意味着你已经有了一个与此源文件相关的 ABEL 文件，并且已经建立了正确的链接。

V. 编译 ABEL HDL

A. 选择 abeltop 源文件。



B. 在处理过程列表中，双击 Compile Logic 过程。当处理过程结束后，你的项目管理器应该如上图所示。

VII. 仿真

你现在可以对整个设计进行仿真。为此，你需要一个新的测试矢量文件。在这个例子中你只需要修改当前的测试矢量文件。

A. 双击 demo.abv 源文件，就会出现文本编辑器。

B. 按照下图修改测试矢量文件：

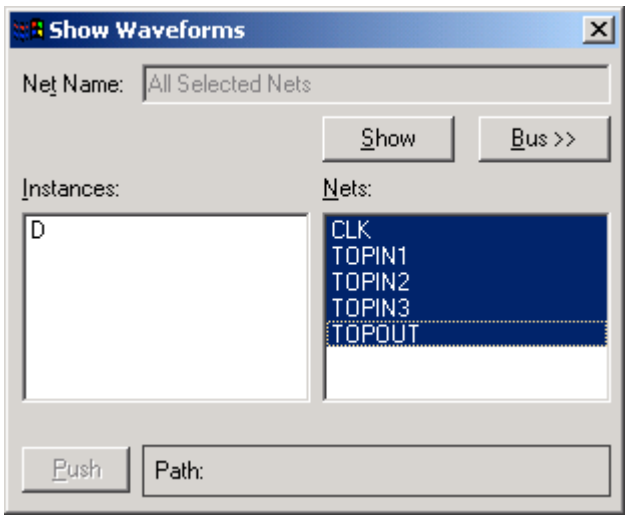
```
module demo;

c,x = .c.,.x.;

CLK,TOPIN1,TOPIN2,TOPIN3,TOPOUT   PIN;

TEST_VECTORS
([CLK,TOPIN1,TOPIN2,TOPIN3]->[TOPOUT])
[ c ,  0  ,0 , 0  ]->[ x ];
[ c ,  0  ,0 , 1  ]->[ x ];
[ c ,  0  ,1 , 0  ]->[ x ];
[ c ,  0  ,1 , 1  ]->[ x ];
[ c ,  1  ,0 , 0  ]->[ x ];
[ c ,  1  ,0 , 1  ]->[ x ];
[ c ,  1  ,1 , 0  ]->[ x ];
[ c ,  1  ,1 , 1  ]->[ x ];
END
```

C. 完成后，存盘退出。

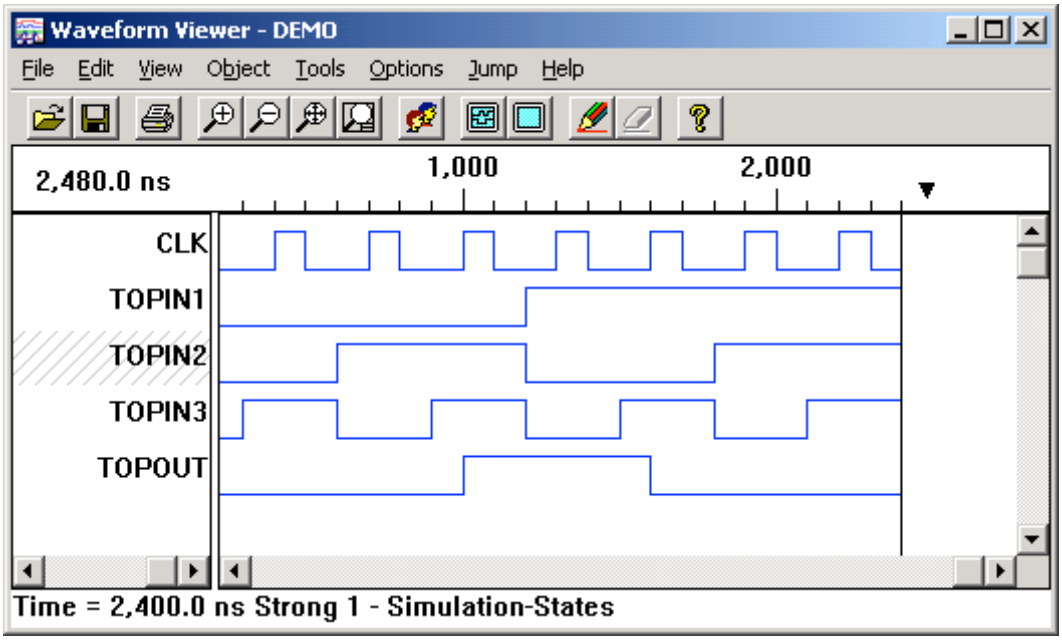
D. 仍旧选择测试矢量源文件，双击 Functional Simulation 过程，进行功能仿真。

E. 现进入 Simulation Control Panel 窗口。按 Tools=> Waveform Viewer 窗口，打开波形观测器准备查看仿真结果。

F. 为了看波形，你必须在 Waveform Viewer 窗口中按 Edit=>Show 菜单，弹出如下 Show Waveforms 窗口：



G. Show Waveforms 窗口中选择 CLK, TOPIN1, TOPIN2, TOPIN3 和 TOPOUT 信号，并且按 Show 钮。然后按 File=>Save 菜单。这些信号名都可以在波形观测器中观察到。再按 Run 钮进行仿真，其结果如下图所示：



H. 在步骤 D 中，如双击 Timing Simulation 过程，即可进入时序仿真流程，以下仿真步骤与功能仿真相同。

VIII. 把设计适配到 Lattice 器件中

现在你已经完成了原理图和 ABEL 语言的混合设计及其仿真。剩下的步骤只是将你的设计放入器件中。因为你已经在第二节中选择了器件，你可以直接执行下面的步骤:

A. 在源文件窗口中选择 LC4032V-10T44I 器件作为编译对象，并注意观察对应的处理过程。

B. 双击处理过程 Fit Design。这将迫使项目管理器完成对源文件的编译，然后连接所有的源文件，最后进行逻辑分割，布局和布线，将设计适配到所选择的 Lattice 器件中。

C. 当这些都完成后，你可以双击 HTML Fitter Report，查看一下设计报告和有关统计数据。

D. 祝贺!!你现在已经完成了设计例子，并且掌握了 ispLEVER 的主要功能。

IX. 层次化操作方法

层次化操作是 ispLEVER 项目管理器的重要功能，它能够简化层次化设计的操作。

a) 在项目管理器的源文件窗口中，选择最顶层原理图“top.sch”.此时在项目管理器右边的操作流程清单中必定有 Navigation Hierarchy 过程。

b) 双击 Navigation Hierarchy 过程，即会弹出最顶层原理图“top.sch”。

c) 选择 View 菜单中的 Push/Pop 命令，光标就变成十字形状。

d) 用十字光标单击顶层原理图中的 abeltop 符号，即可弹出描述 abeltop 逻辑的文本文件 abeltop.abl。此时可以浏览或编辑 ABELHDL 设计文件。浏览完毕后用 File 菜单中的 Exit 命令退回顶层原理图。

e) 用十字光标单击顶层原理图中的 demo 符号，即可弹出描述 demo 逻辑的底层原理图 demo.sch。此时可以浏览或编辑底层原理图。

f) 若欲编辑底层原理图，可以利用 Edit 菜单中的 Schematic 命令进入原理图编辑器。编译完毕后用 File 菜单中的 Save 和 Exit 命令退出原理图编辑器。

g) 底层原理图浏览完毕后用十字光标单击图中任意空白处即可退回上一层原理图。

h) 若某一设计为多层次化结构，则可在最高层逐层进入其底层，直至最底一层；退出时亦可以从最底层逐层退出，直至最高一层。

i) 层次化操作结束后用 File 菜单中的 Exit 命令退回项目管理器。

注意：　　将 Y1 端口定义成时钟输入端的方法

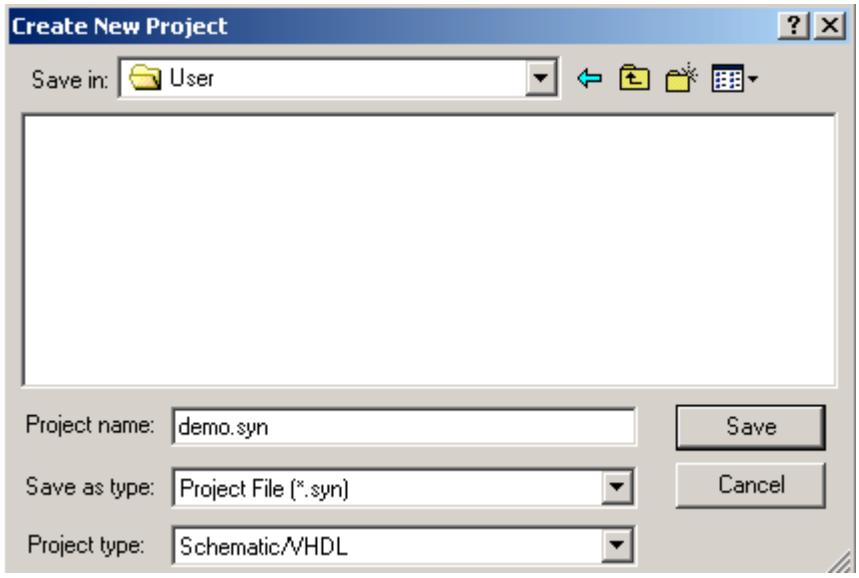
ispLSI1016 和 ispLSI2032 两种器件的 Y1 端是功能复用的。如果不加任何控制，适配软件在编译时将 Y1 默认为是系统复位端口(RESET)。若欲将 Y1 端用作时钟输入端，必须通过编译器控制参数来进行定义。

第五节　ispLEVER 工具中 VHDL 和 Verilog 语言的设计方法

用户的 VHDL 或 Verilog 设计可以经 ispLEVER 系统提供的综合器进行编译综合，生成 EDIF 格式的网表文件，然后可进行逻辑或时序仿真，最后进行适配，生成可下载的 JEDEC 文件。

VHDL 设计输入的操作步骤

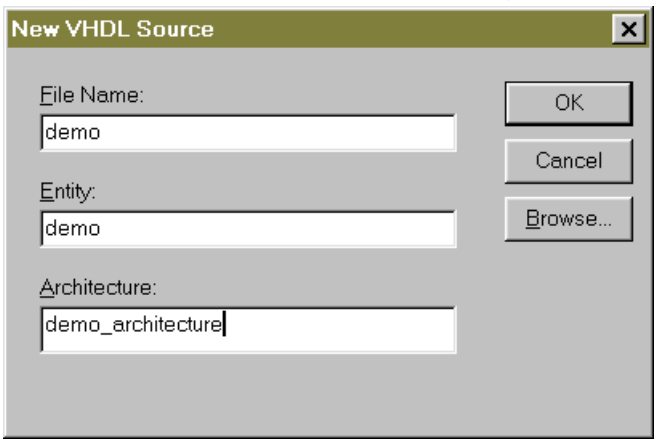
在 ispLEVER System Project Navigator 主窗口中，按 File=>New Project 菜单建立一个新的工程文件，此时会弹出如下图所示的对话框。请注意：在该对话框中的 Project Type 栏中，必须根据设计类型选择相应的工程文件的类型。本例中，选择 VHDL 类型。若是 Verilog 设计输入，则选择 Verilog HDL 类型。



将该工程文件存盘为 demo.syn。

在 ispLEVER System Project Navigator 主窗口中，选择 Source=>New 菜单。在弹出的 New Source 对话框中，选择 VHDL Module 类型。

此时，软件会产生一个如下图所示的 New VHDL Source 对话框：



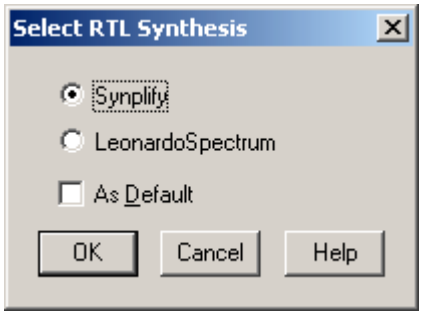
在对话框的各栏中，分别填入如上图所示的信息。按 OK 钮后，进入文本编辑器-Text Editor 编辑 VHDL 文件。

在 Text Editor 中输入如下的 VHDL 设计，并存盘。

```
library ieee;
use ieee.std_logic_1164.all;

entity demo is
        port ( A, B, C, D, CK: in std_logic;
                OUTP: out std_logic);
end demo;

architecture demo_architecture of demo is
signal INP: std_logic;
begin
        Process (INP, CK)
        begin
                if (rising_edge(CK)) then
                        OUTP <= INP;
                end if;
        end process;
        INP <= (A and B) or (C and D);
end demo_architecture;
```

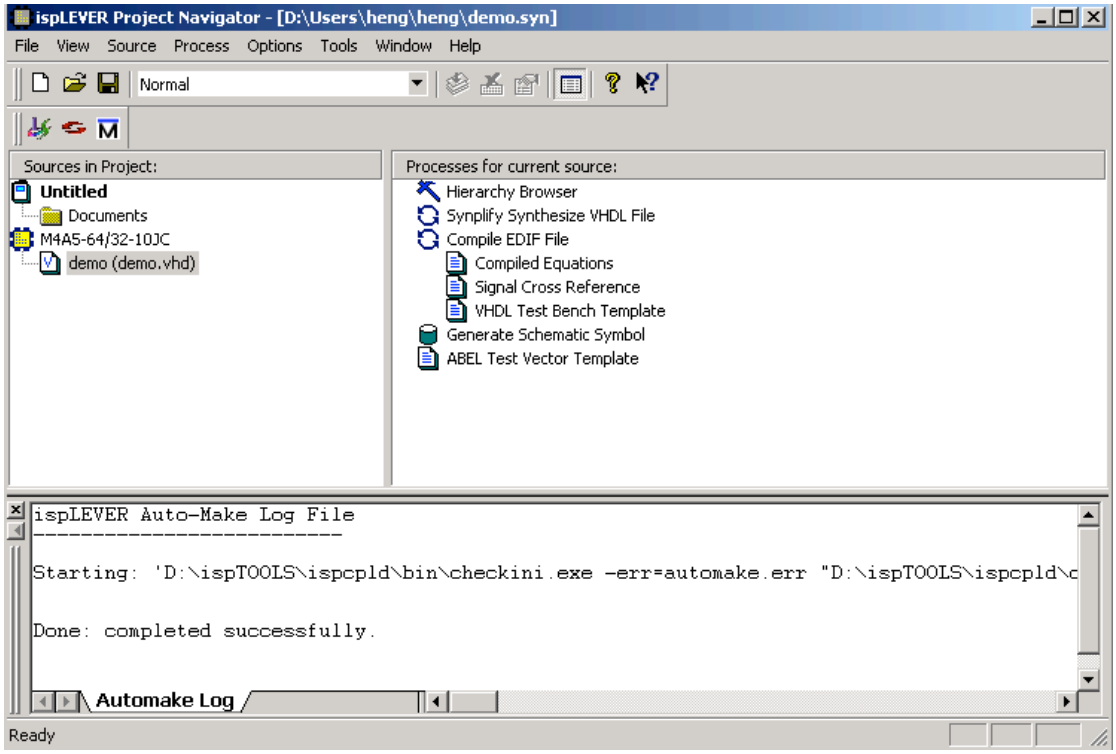
此 VHDL 设计所描述的电路与 5.2 节所输入的原理图相同，只不过将输出端口 OUT 改名为 OUTP(因为 OUT 为 VHDL 语言保留字)。

此时，在 ispLEVER System Project Navigator 主窗口左侧的源程序区中，demo.vhd 文件被自动调入。选择器件 ispMACH4A5－64/32-10JC，并启动 Options=>Select RTL Synthesis 菜单，显示如下对话框：

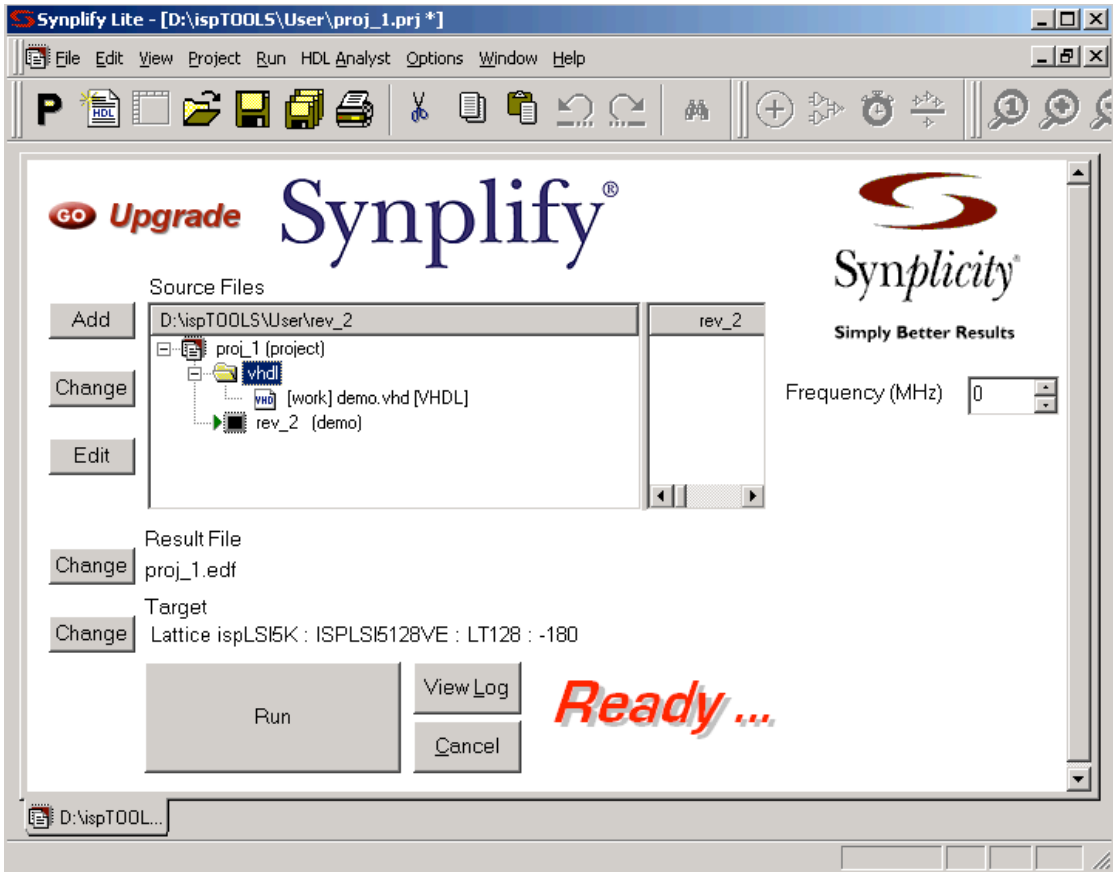


在该对话框选择 Synplify，即采用 Synplify 工具对 VHDL 设计进行综合。

此时的 ispLEVER System Project Navigator 主窗口如下图所示：



双击 Processes 窗口的 Synplify Synthesize VHDL File 进行编译、综合。或者选择菜单 Tools=> Synplify Synthesis 产生如下窗口。选 Add 调入 demo.vhd，然后对 demo.vhd 文件进行编译、综合。

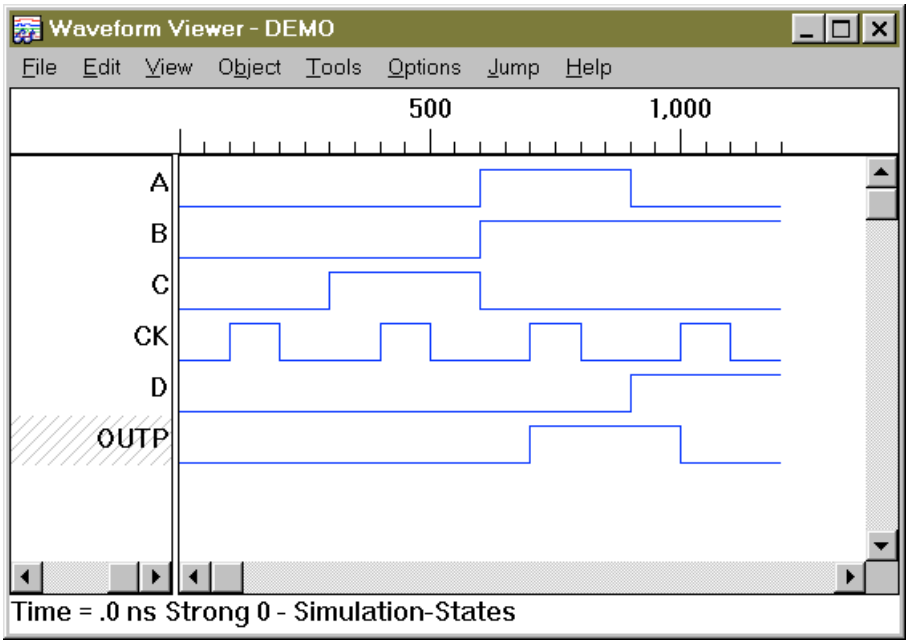


若整个编译、综合过程无错误，该窗口在综合过程结束时会自动关闭。若在此过程中出错，双击上述 Synplify 窗口中 Source Files 栏中的 demo.vhd 文件进行修改并存盘，然后按 RUN 钮重新编译。

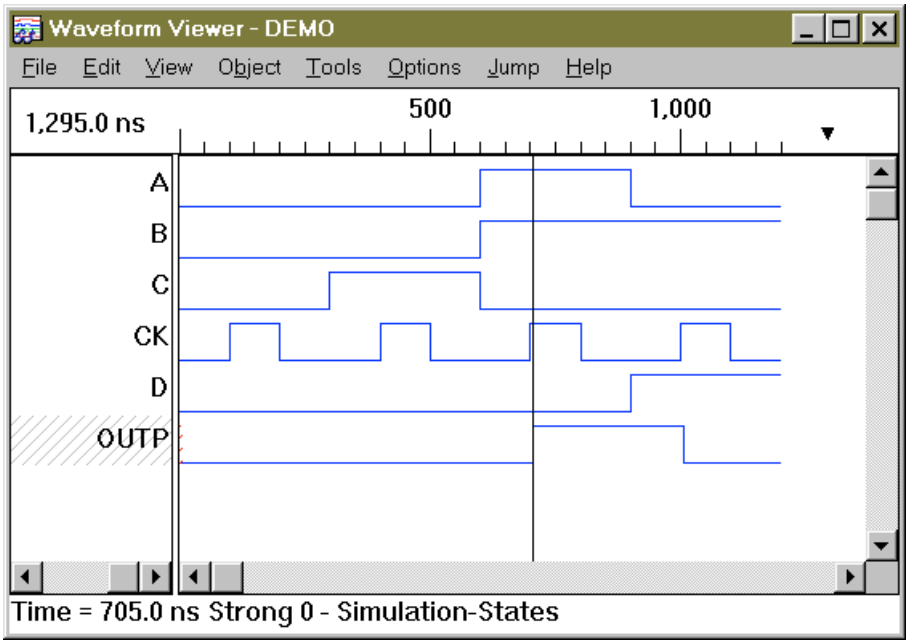
在通过 VHDL 综合过程后，可对设计进行功能和时序仿真。在 ispLEVER System Project Navigator 主窗口中按 Source=>New 菜单，产生并编辑如下的测试向量文件 demo.abv：

```
module demo;

c,x = .c.,.x.;

CK,A,B,C,D,OUTP     PIN;

TEST_VECTORS
([CK,  A,    B,  C,  D]-
>[OUTP])
[  c  ,   0    , 0  ,  0  , 0 ]->[ x ];
[  c  ,   0    , 0  ,  1  , 0 ]->[ x ];
[  c  ,   1    , 1  ,  0  , 0 ]->[ x ];
[  c  ,   0    , 1  ,  0  , 1 ]->[ x ];
END
```

在 ispLEVER System Project Navigator 主窗口中选中左侧的 demo.abv 文件，双击右侧的 Functional Simulation 栏，进行功能仿真。在 Waveform Viewer 窗口中观测信号 A，B，C，CK，D 和 OUTP，其波形如下图所示：



在 ispLEVER System Project Navigator 主窗口中选中左侧的 demo.abv 文件，双击右侧的 Timing Simulation 栏，进行时序仿真。选择 Maximum Delay，在 Waveform Viewer 窗口中观测信号 A，B，C，CK，D 和 OUTP，其波形如下图所示：



在 ispLEVER System Project Navigator 主窗口中选中左侧的 ispMACH 器件，双击右侧的 Fit Design 栏，进行器件适配。该过程结束后会生成用于下载的 JEDEC 文件 demo.jed。

Verilog 设计输入的操作步骤

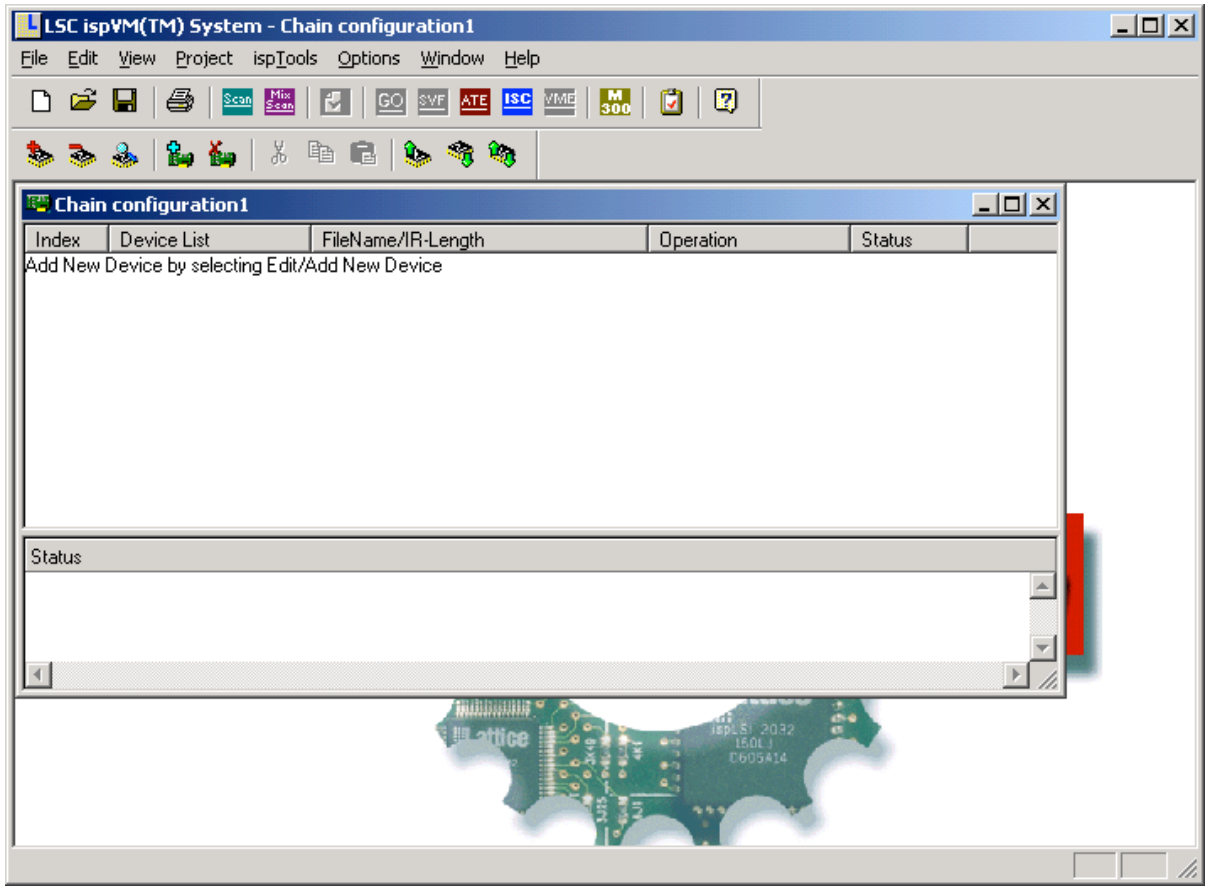
Verilog 设计输入的操作步骤与 VHDL 设计输入的操作步骤完全一致，在此不再赘述。需要注意的是在产生新的工程文件时，工程文件的类型必须选择为 Verilog HDL。

第六节 ispVM System－在系统编程的软件平台

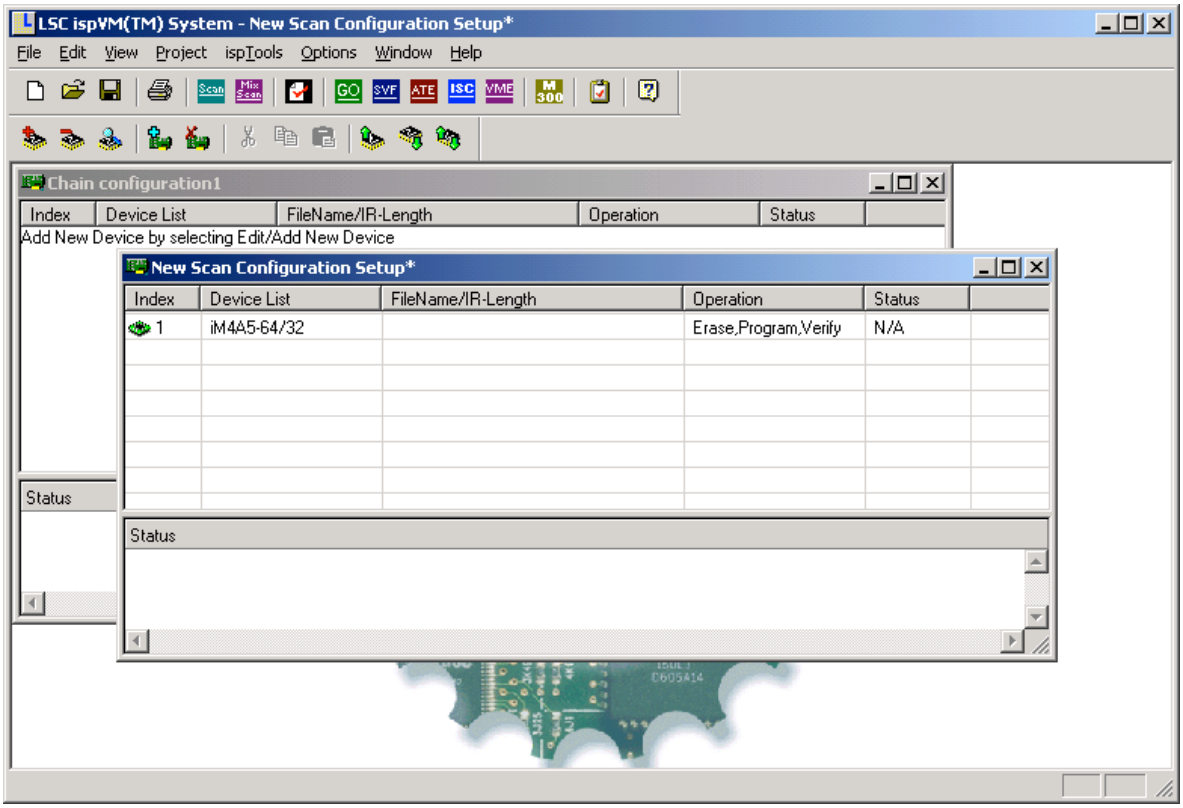
Lattice 器件的在系统编程是借助 ispVM System 软件来实现的。ispVM System 软件集成在 ispLEVER 软件中，它同时也可以是一个独立的器件编程软件。ispVM System™ 是一个综合的将设计下载到器件的软件包。该软件提供一种有效的器件编程方式，即采用由莱迪思半导体公司或其他公司的设计软件所生成的 JEDEC 文件来对 ISP 器件编程。这一完整的器件编程工具允许用户快速简便地通过 ispSTREAM™ 将设计烧写到器件上。它还拥有简化 ispATE™、ispTEST™ 及 ispSVF™ 编程的功能。在此仅介绍最常用的基于 PC 机 Windows 环境的 ispVM System，其使用方法如下：

在启动 ispVM System 前，先将 Lattice 下载电缆连接在 PC 机的并行口和待下载的印刷电路板上，并打开印刷电路板的电源。

在 Windows 中，按 Start=>Programs=>Lattice Semiconductor=>ispVM System 菜单启动 ispVM System，如下图所示。

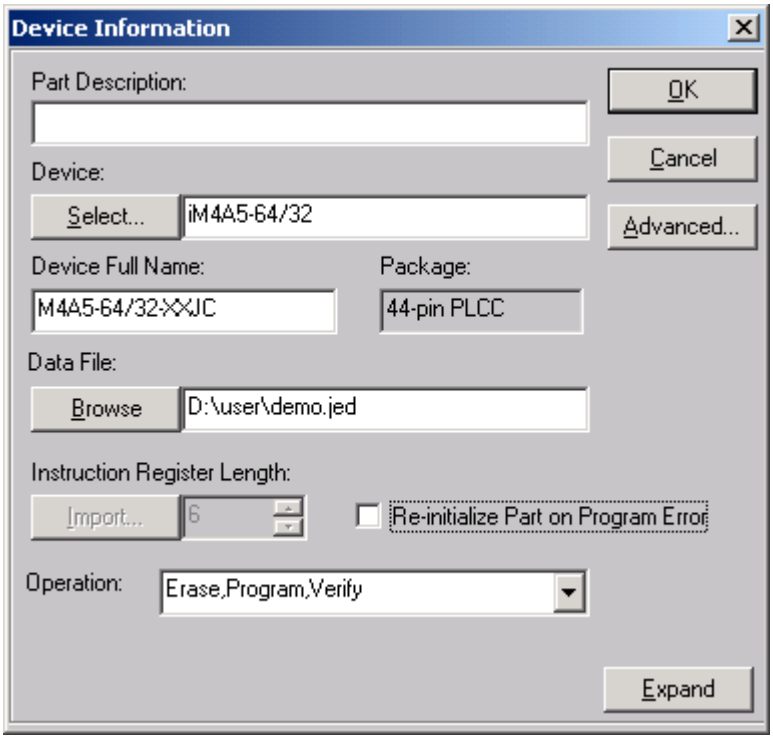


在 LSC ispVM™ System 窗口中，按 ispTools=>Scan Chain 菜单，ispVM System 软件会自动检测 JTAG 下载回路，找到回路中所有的器件型号。在本例中，印刷电路板上的 JTAG 下载回路中仅有一片 M4A5-64/32-10JC 器件，因此，Scan Chain 后的窗口如下图所示。

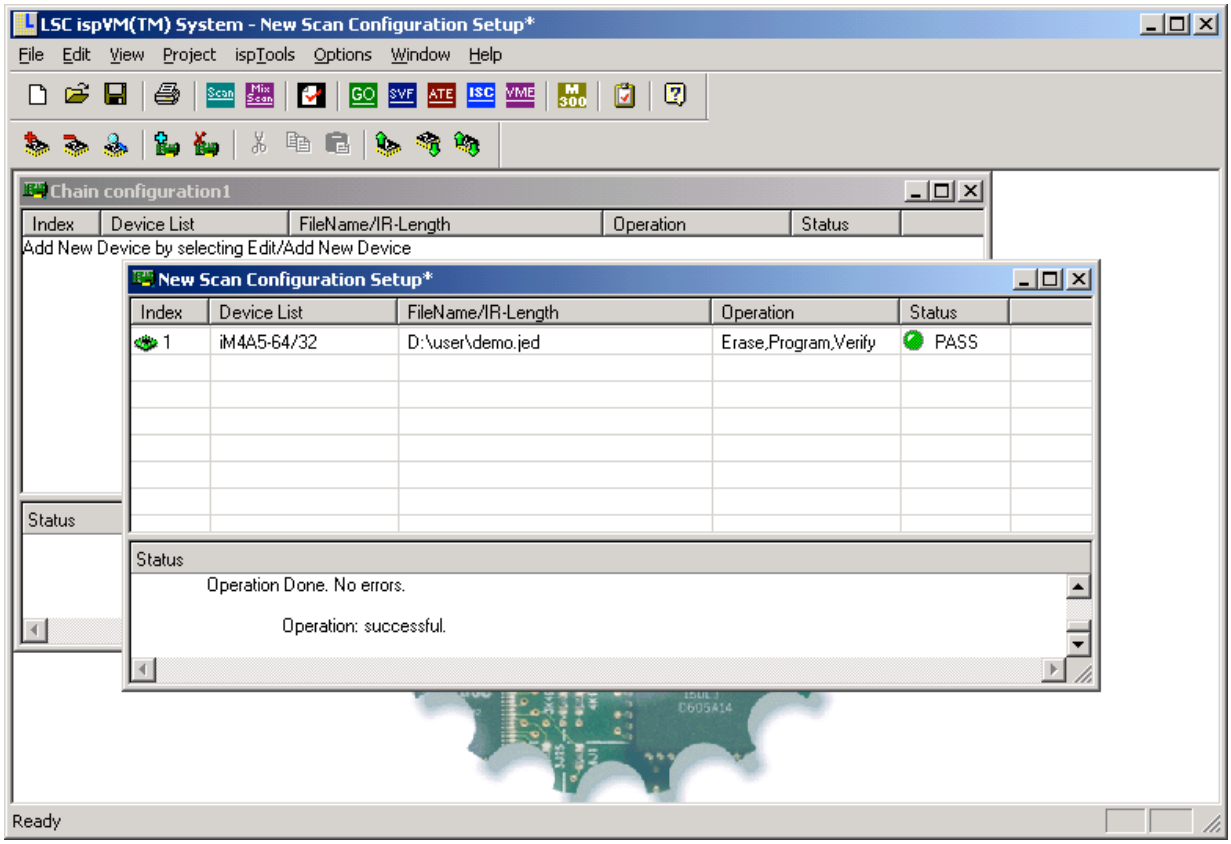


为得到可供下载到 M4A5-64/32-10JC 器件中的 JED 文件，我们可以将第四节设计实例中的器件型号改选为 M4A5-64/32-10JC，重新做编译和适配，得到基于 M4A5-64/32-10JC 器件的 JED 文件。

在 LSC ispVM™ System 窗口中，双击 New Scan Configuration Setup 子窗口中的 iM4A5-64/32 栏，弹出 Device Information 对话框。在该对话框中的 Data File 栏里，选择需要下载的 JED 文件 D:\user\demo.jed；在该对话框中的 Operation 栏里，选择所需的编程操作，这里选 Erase,Program,Verify，对器件进行擦除、编程、校验。完成这些操作后，Device Information 对话框如下图所示。按 OK 钮，关闭该对话框。



在 LSC ispVM™ System 窗口中，按 Project=>Download 菜单启动下载操作。数秒钟后，下载完成，这时 New Scan Configuration Setup 子窗口中的 Status 栏显示 PASS，并有一个绿色的圆点，参见下图。



器件回读与加密

器件回读

运用 ispVM System 软件，可以将已下载过的、未经加密的器件中的熔丝信息回读出来，并存储为新的 JED 文件共复制相同设计的器件。其操作方法是：在 Device Information 对话框中的 Operation 栏里，选择 Read and Save JEDEC 操作，同时，在 Data File 栏里，输入将要存放熔丝信息的文件名（JED 文件）。在 LSC ispVM™ System 窗口中，按 Project=>Download 菜单启动回读操作。

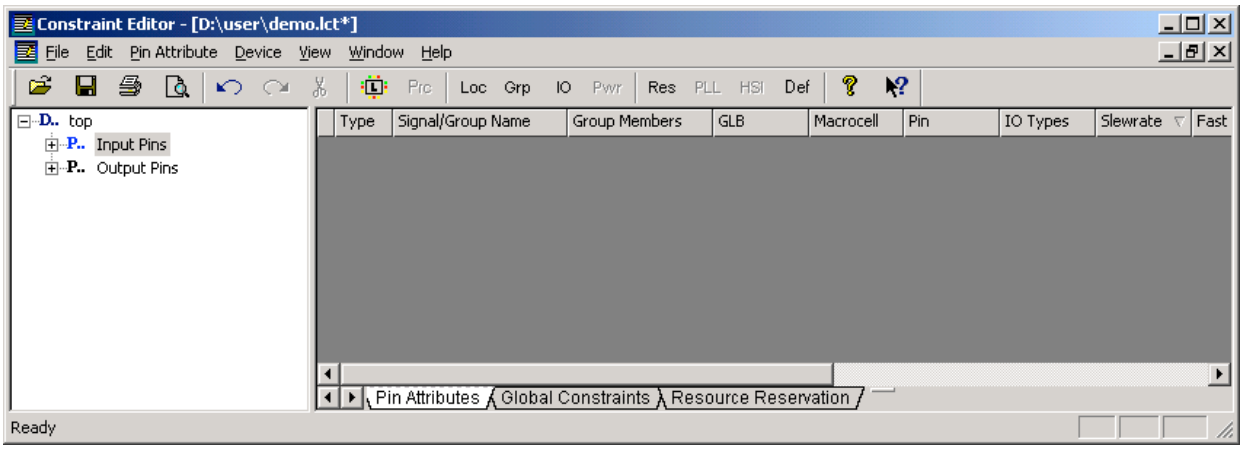
器件加密

为防止自己的设计被非法回读，设计者可以在下载设计的时候对器件进行加密。其操作方法是：在 Device Information 对话框中的 Operation 栏里，选择 Erase,Program,Verify,Secure 操作。在 LSC ispVM™ System 窗口中，按 Project=>Download 菜单启动加密下载操作。如果对加密后的器件进行回读操作，那么可以看到回存的 JED 文件中，熔丝信息均为 0。

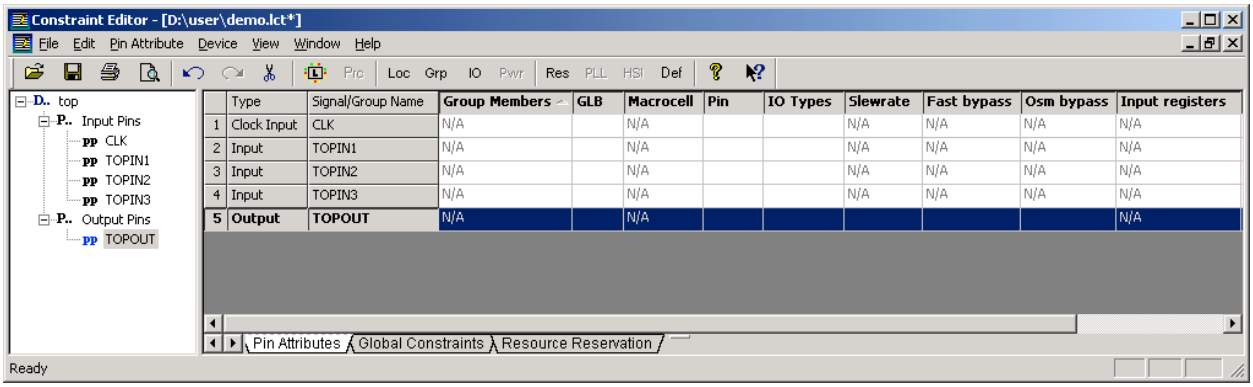
第七节　约束条件编辑器（Constraint Editor）的使用方法

ispLEVER 软件中的 Constraints Editor 是一个功能强大的、集成的设计参数设置工具，其可以设置 Pin Attributes，Global Constraints，Resource Reservation 等参数。根据用户所选器件型号的不同，可供选择的参数也不尽相同。以下我们仍以第四节中的设计为例，说明其使用方法。

在 ispLEVER Project Navigator 的主窗口左侧，选中器件型号栏（LC4032V-10T44I），双击右侧的 Constraint Editor 功能条，打开 Constraint Editor，如下图所示。

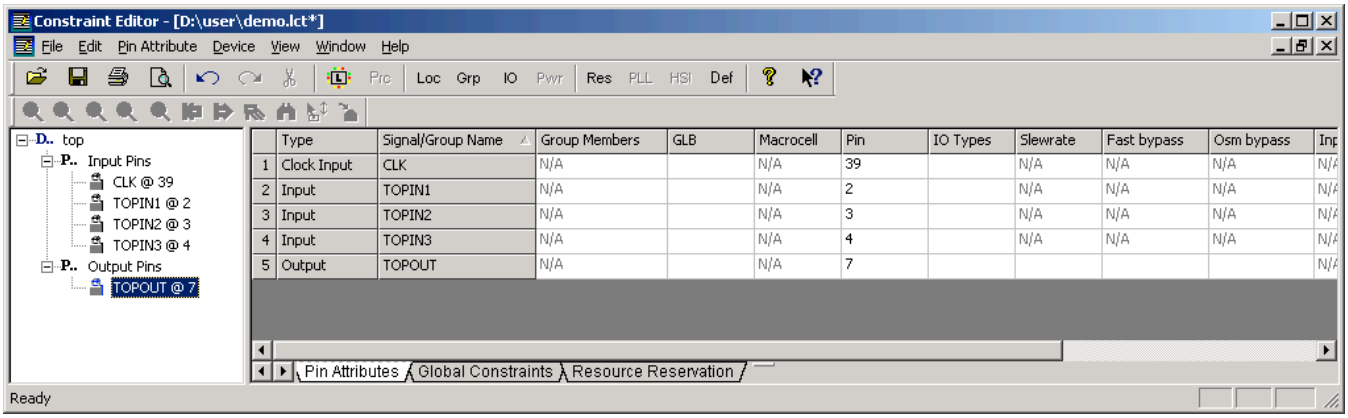


点击窗口左侧 Input Pins 和 Output Pins 左边的⊞，展现所有的输入信号：CLK，TOPIN1，TOPIN2，TOPIN3 以及输出信号 TOPOUT。双击这些信号名，在窗口右侧会出现对应于每个信号的参数行，如下图所示。



在参数行中，可以单独设置每个信号的 Group Members，GLB，Macrocell，Pin，I/O Types，Slewrate，Fast bypass，Osm bypass，Input registers，Register powerup 等参数。在这些参数中，最常用的是用于引脚锁定的参数 Pin，其设置方法如下：

双击每个信号参数行的 Pin 这一格，输入该信号需要锁定的引脚序列号。假定信号 CLK，TOPIN1，TOPIN2，TOPIN3 和 TOPOUT 需要锁定的引脚号分别为 39,2,3,4,7，分别输入这些引脚号，结果如下：



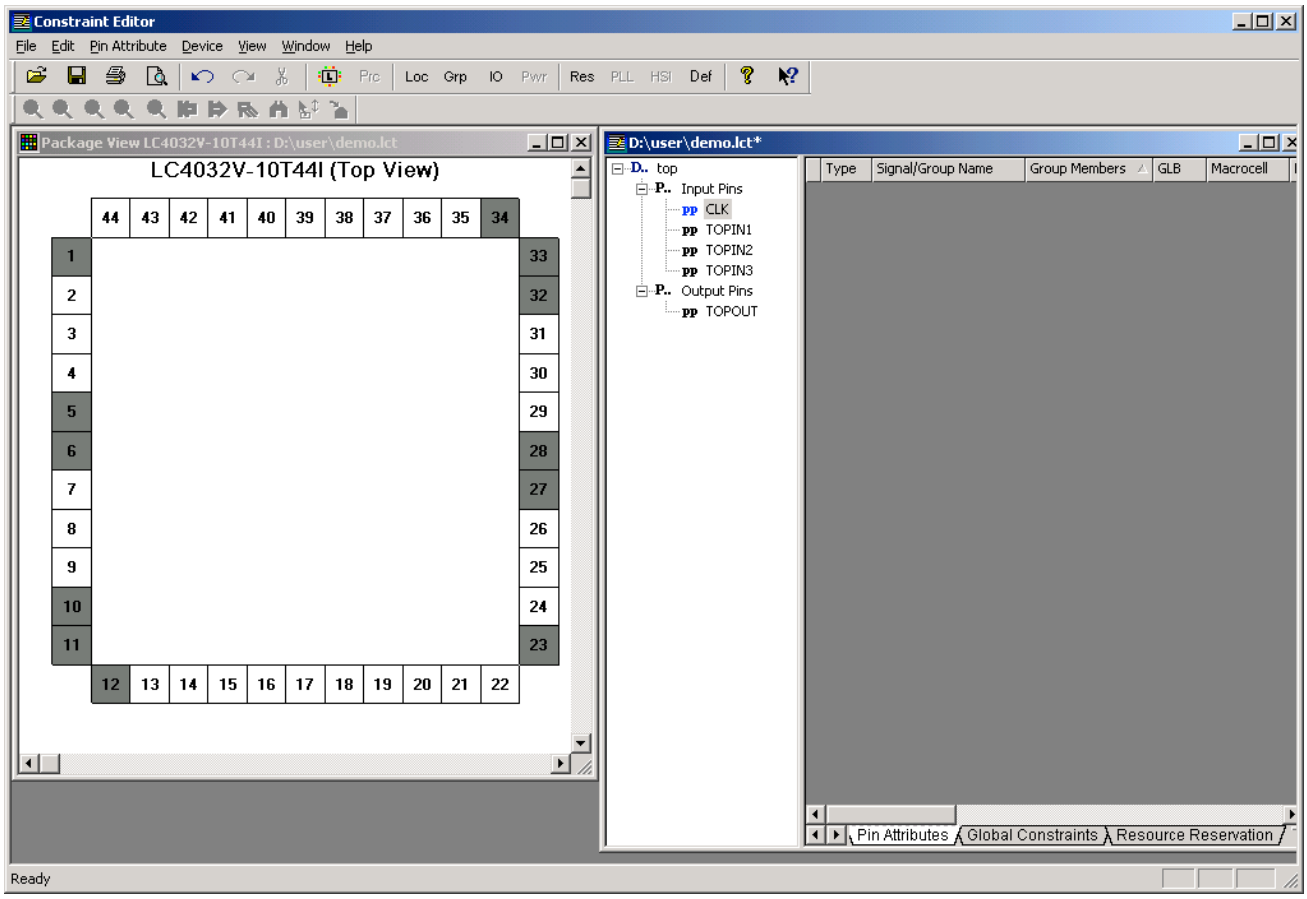
设置完成后，按 File=>Save 菜单存盘保存设置。无论是原理图还是用 HDL 做的设计，都可以采用这种方法设定器件的引脚。

若需设置 Global Constraints，Resource Reservation 的参数，可以按窗口右下方的 Global Constraints 和 Resource Reservation 菜单。

引脚锁定的另一种方法：

引脚锁定是参数设置中最常用的，以下介绍另一种直观的引脚锁定方法：

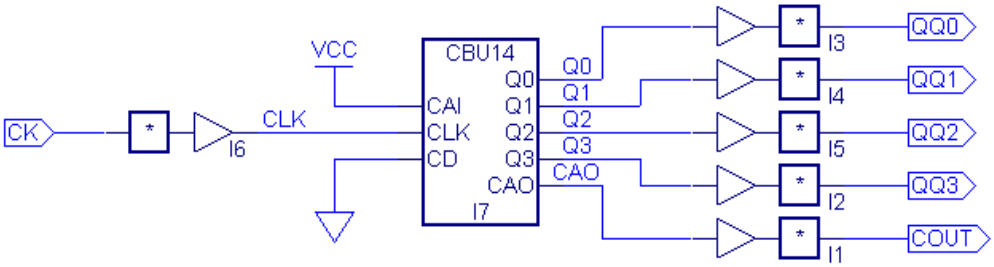
在 Constraints Editor 窗口中，按 Device=>Package View 菜单，窗口变成如下形式。



在右侧窗口中选中要锁定的信号名，按下鼠标左键，将该信号拖至窗口左边器件引脚图中对应的引脚上，放开左键，该信号就被锁定在对应的引脚上了。用这种方法锁定引脚方便、直观，在复杂设计中尤为如此。

附录一 ispLEVER System 上机实习题

实验习题一　按照所给电路图设计一个四位二进制加法计数器，并进行功能仿真



操作方法

1　建立一个名为 CNT14 的新设计项目，并打开原理图编辑器。

2　先按照上机操作教材第四节之 3 建立名为 CBU14 的逻辑元件符号。

3　调用逻辑元件 CBU14，完成原理图输入，并标注内部节点名称，然后存盘退出。

4　四位二进制加法计数器 CBU14 的 ABEL 描述语句为：

```
MODULE CBU14

CAI,CLK,CD  PIN;
CAO                PIN ISTYPE 'COM';
Q3..Q0             PIN ISTYPE 'REG';

count = [Q3..Q0];

EQUATIONS
count.CLK = CLK;
count.AR = CD;
count := (count.fb) & !CAI;
count := (count.fb + 1) & CAI;
CAO = Q3.Q & Q2.Q & Q1.Q & Q0.Q & CAI;
END
```

5　根据以上语句用文本编辑器建立 CBU14.ABL 文件，并用 Source 菜单中的 Import 命令调入设计环境。

6　用文本编辑器建立测试向量文件 CNT14.ABV。

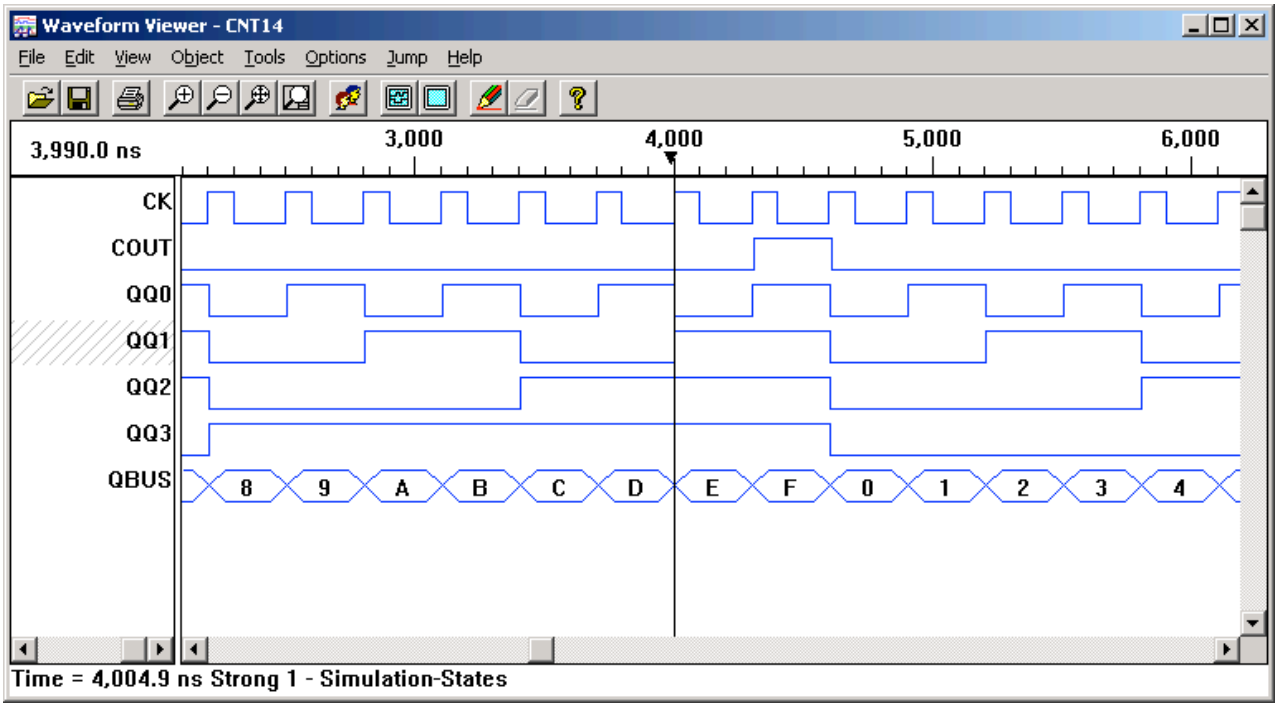
```
module CNT14;

"pins
CK                          pin;
QQ0,QQ1,QQ2,QQ3             pin ISTYPE 'REG';
COUT                        pin ISTYPE 'COM';

test_vectors  (CK -> [QQ0,QQ1])
      @repeat 35 { .c. -> [.x.,.x.]; }
```

end

7 调入测试向量文件 CNT14.ABV，运行时序仿真的编译过程，通过后显示出波形图。

8 通过 Edit 菜单中的 Show 或 Hide 命令显示出如下图所示的波形：

图中，QBUS 是由信号 QQ3，QQ2，QQ1，QQ0 所组成的总线信号。



附录二　ispLEVER 软件中文件名后缀及其含义

SYN	源 文 件	设 计 项 目 管 理 文 件
ABL	源 文 件	ABEL 硬 件 描 述 语 言 源 文 件
ABV	源 文 件	测 试 向 量 描 述 文 件
SCH	源 文 件	电 路 原 理 图 文 件
VHD	源 文 件	VHDL 硬 件 描 述 语 言 源 文 件
V	源 文 件	Verilog 硬 件 描 述 语 言 源 文 件
PPN	源 文 件	引 脚 锁 定 描 述 文 件 (用 电 路 图 锁 定 引 脚 时 为 中 间 文 件)
PAR	源 文 件	适 配 器 控 制 参 数 文 件
SYM	中 间 文 件	电 路 符 合 文 件
EQO	中 间 文 件	逻 辑 描 述 文 件 (由 ABL 编 译 所 得)
EQ1	中 间 文 件	简 化 逻 辑 文 件 (由 EQO 化 简 所 得)
EQ2	中 间 文 件	带 层 次 连 接 关 系 的 逻 辑 描 述 文 件
EQ3	中 间 文 件	经 优 化 的 逻 辑 描 述 文 件
EQ4	中 间 文 件	经 反 复 优 化 的 逻 辑 描 述 文 件
TMV	中 间 文 件	经 编 译 的 测 试 向 量 文 件
TT2	中 间 文 件	逻 辑 网 表 输 出 文 件, 适 配 器 输 入 文 件
FXP	中 间 文 件	逻 辑 布 局 结 果 文 件
LST	中 间 文 件	ABEL 源 文 件 的 列 表 文 件
LOG	中 间 文 件	运 行 流 程 记 录 文 件
SIM	中 间 文 件	仿 真 用 网 表 文 件
JHD	中 间 文 件	层 次 化 关 系 连 接 表 文 件
JED	结 果 文 件	熔 丝 图 文 件 (JEDEC 文 件)
REP	结 果 文 件	GAL 器 件 设 计 编 译 报 告 文 件
RPT	结 果 文 件	IspLSI 器 件 设 计 编 译 报 告 文 件
XRF	结 果 文 件	信 号 和 节 点 简 缩 名 称 文 件
ERR	结 果 文 件	错 误 报 告 文 件
MFR	结 果 文 件	频 率 分 析 报 告 文 件
TSU	结 果 文 件	寄 存 器 建 立 和 保 持 时 间 报 告 文 件
TPD	结 果 文 件	TPD 路 径 延 时 时 间 报 告 文 件
TCO	结 果 文 件	TCO 路 径 延 时 时 间 报 告 文 件