

# CADENCE PSD14.0

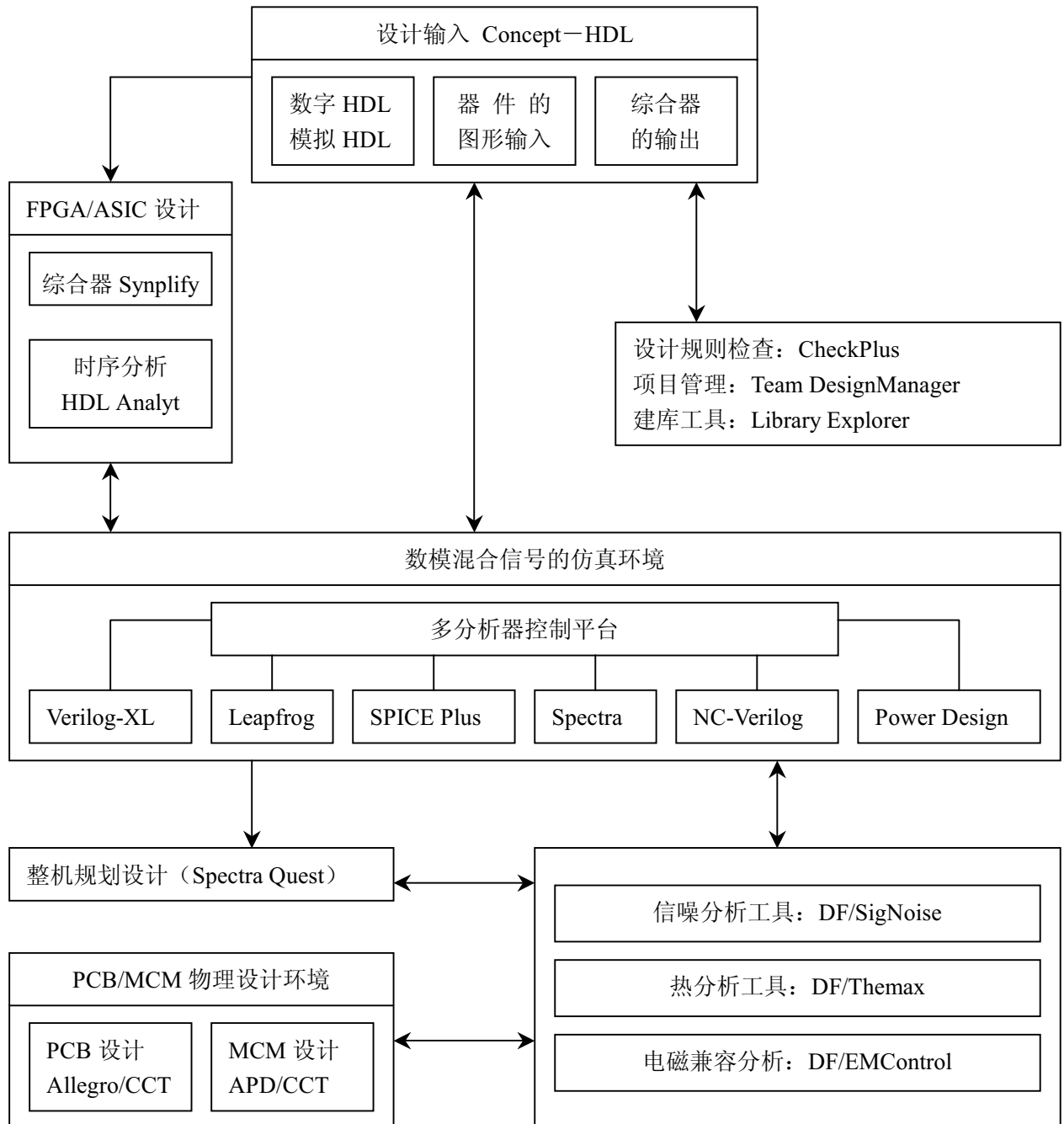
## 入 门 教 程

(V1.0)

## 序

Cadence 公司是全球最大的电子设计自动化公司，它的 EDA 软件在全球计算机、通信、航空航天及民用消费类电子产品的设计、研发部门中获得越来越多的青睐。

Cadence 在电子线路 PCB 板级系统开发应用中提供的设计工具流程如下：



在从前到后的设计流程中，又可以细分为 FPGA 可编程逻辑器件的设计，数据混合电路仿真原理图到 PCB 板布局/布线设计及高速 PCB 板规划和分析四大部分。

由于时间、篇幅和个人能力的限制，在这里我们只能将一些设计流程中最基

本的方法，技巧提供给大家。对于书中一些叙述不详细的地方，请读者们谅解和指正。关于布线器 Spectra 的使用方法在以后会补充进来。

希望大家看完之后，多提意见和本文的错误之处。

**该软件可到：<ftp://166.111.67.188/> 下载**

## 第一章 系统简介

### § 1.1 系统组成

#### 1.1.1 库

所有的 EDA 软件都离不开库的支持。库的丰富程度在一定程度上决定着 EDA 工具的实用性。Cadence 板级设计系统的库按调用它的软件工具可分为四类：

- \*符号库——ConceptHDL 调用
- \*信噪库——SingNoise 调用
- \*逻辑功能库——VerilogXL 调用
- \*物理库——Allegro 调用

#### 1.1.2 原理图输入

在 Cadence 板级设计中的原理图输入工具为 Concept-HDL，它可以灵活高效的原理图送入计算机，并生成后继工具能够处理的数据。

#### 1.1.3 设计转换和修改管理

它实际是原理图与 PCB 之间的一个纽带，负责将原理图中所表述的器件连接关系及元件封装说明翻译成 PCB 所能接受的格式，并将 PCB 信息反馈到原理图中，以保证原理图与 PCB 的同步。这种由前到后和由后到前的设计数据的转换与传递都是由这一环境完成的。

#### 1.1.4 物理设计与加工数据的生成

这一环境主要完成 PCB 版图的设计（包括布局、布线）和生成后继制造与加工 PCB 板所需的各种数据文件。

#### 1.1.5 高速 PCB 规划设计环境

在该环境中可以对 PCB 版图进行电磁兼容性（EMC）和信号完整性分析，并将分析果传递到 Concept 和 Allegro，从而不断修改和完善 PCB 版图。这一工具在信号频率较高的 PCB 版图设计中尤为有用。

### § 1.2 设计流程

Cadence 板级系统设计的基本思想可用下图表示，其中：

\*Project Manager：管理项目设计所使用的工具及工具所产生的数据

\*Design Sync: 设计同步, 保持原理图与 PCB 的一致性

\*Allegro: 版图设计工具

\*Floor Planner: 高速 PCB 规划设计

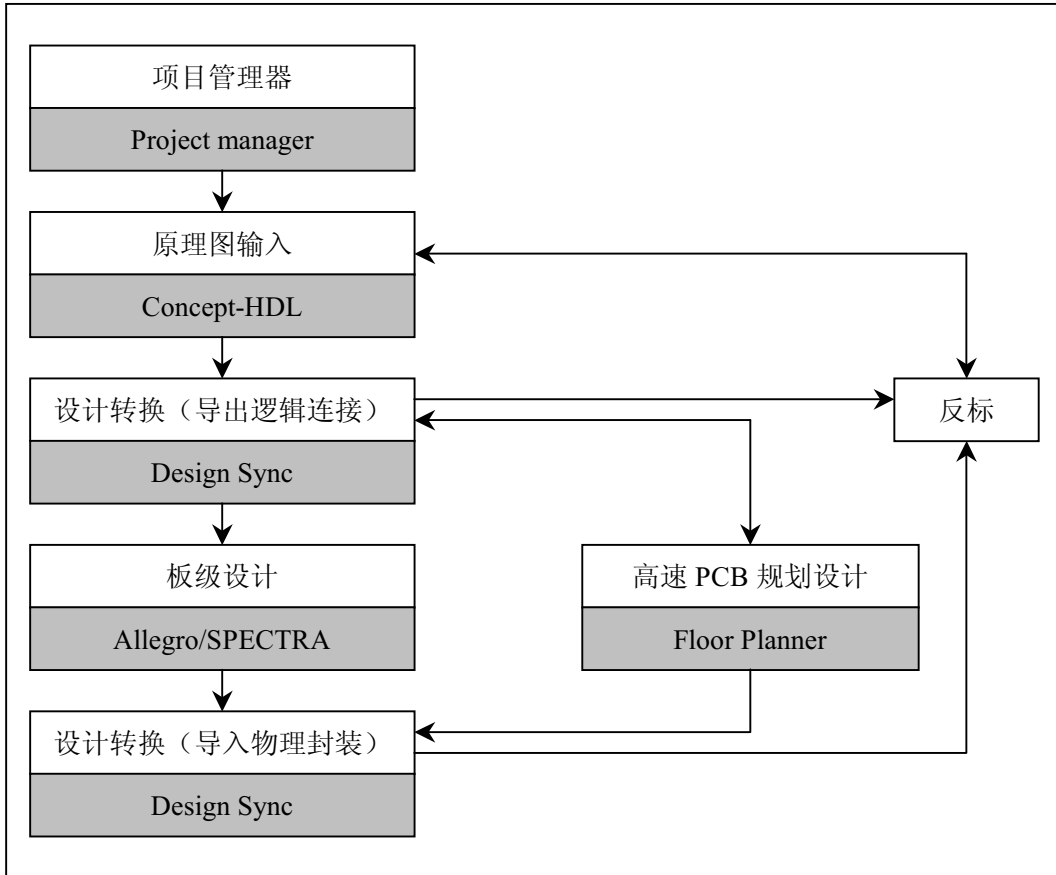


图 1.1 板级设计流程

## 第二章 项目管理器

在 UNIX 环境下进入,键入 projmgr 即可启动项目管理器, 如下图所示:



图 2.1 项目管理器

项目管理器是对用户的设计进行统一管理以及环境设置的工具, 在这个环境中可以调用各种工具进行后继的设计。在这一章中首先介绍如何创建一个工程项目, 然后介绍项目的目录结构, 最后介绍 Cadence 的库结构。

### § 2.1 创建新的设计项目

#### 2.1.1 创建过程

1. 首先创建一个自己的工作目录, 用来存放所有的设计文件, 例: f:\myworking。
2. 启动项目管理器 (projmgr)。
3. 选择 New
4. 在 Project name 中输入项目名 myproj, 在 Location 中输入 f:\myworking, 点击 Next。如图 2.2 所示

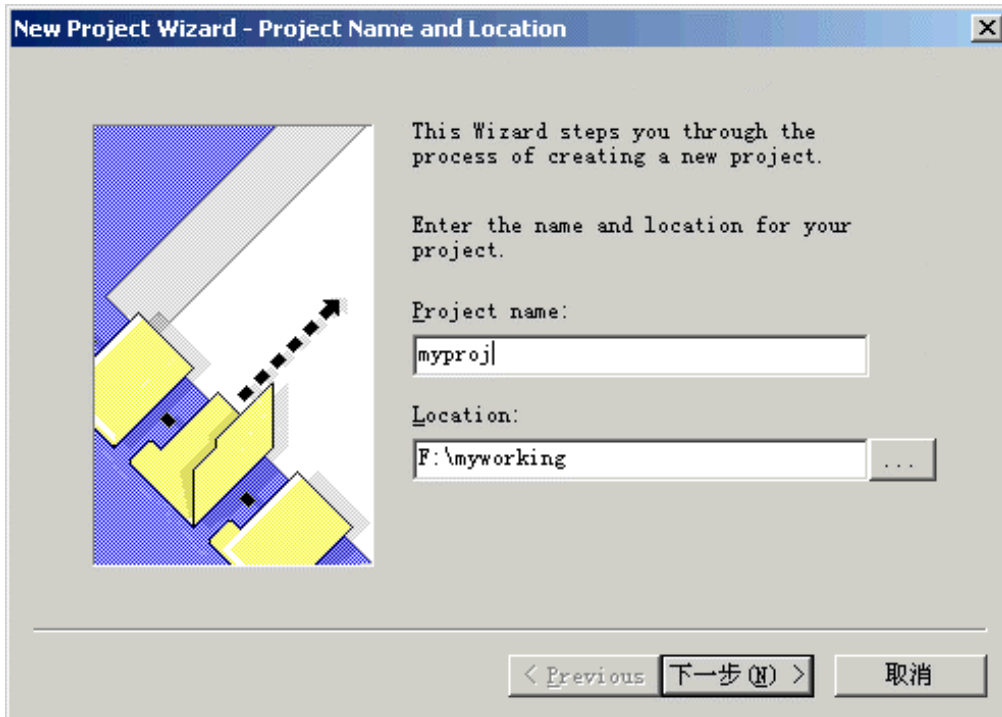


图 2.2

5. 确认 myproj\_lib 和 standard 库均出现在 Project Libraries 中，如图 2.3 所示：

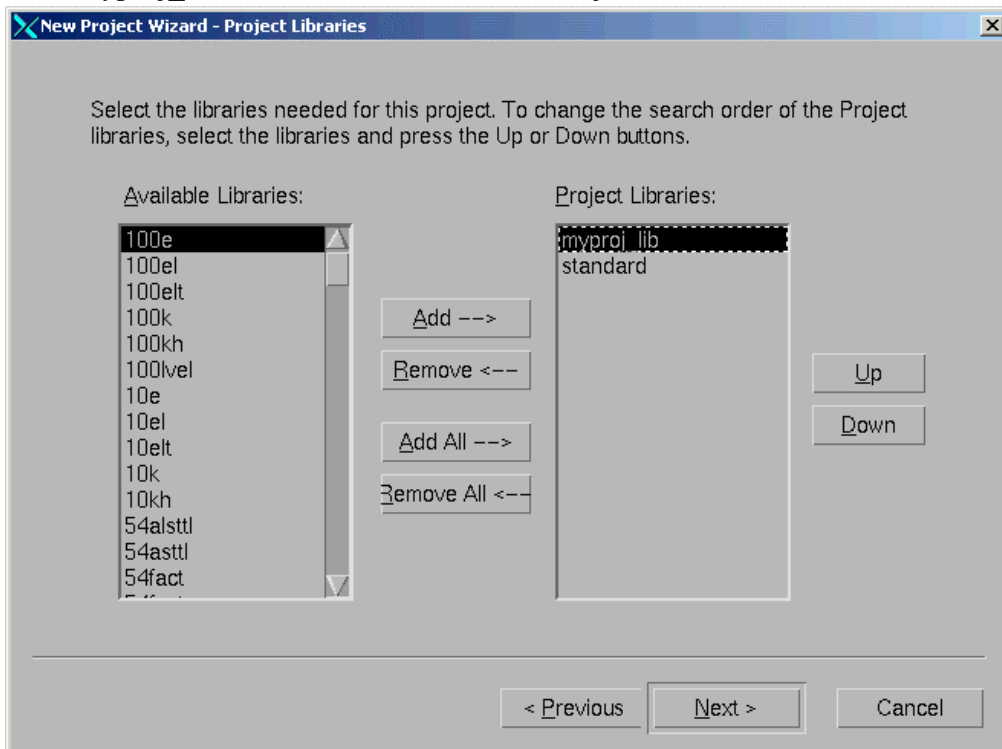


图 2.3

如果缺少任意一个，则可能输入的项目名或指定的路径不是新的，请点击 previous 回到上一级修改。若无错误则点击 Next。

6. 输入一个设计名 mydesign（一个工程项目可能有许多设计，每个设计又能包含许多原理图），如图 2.4 所示：

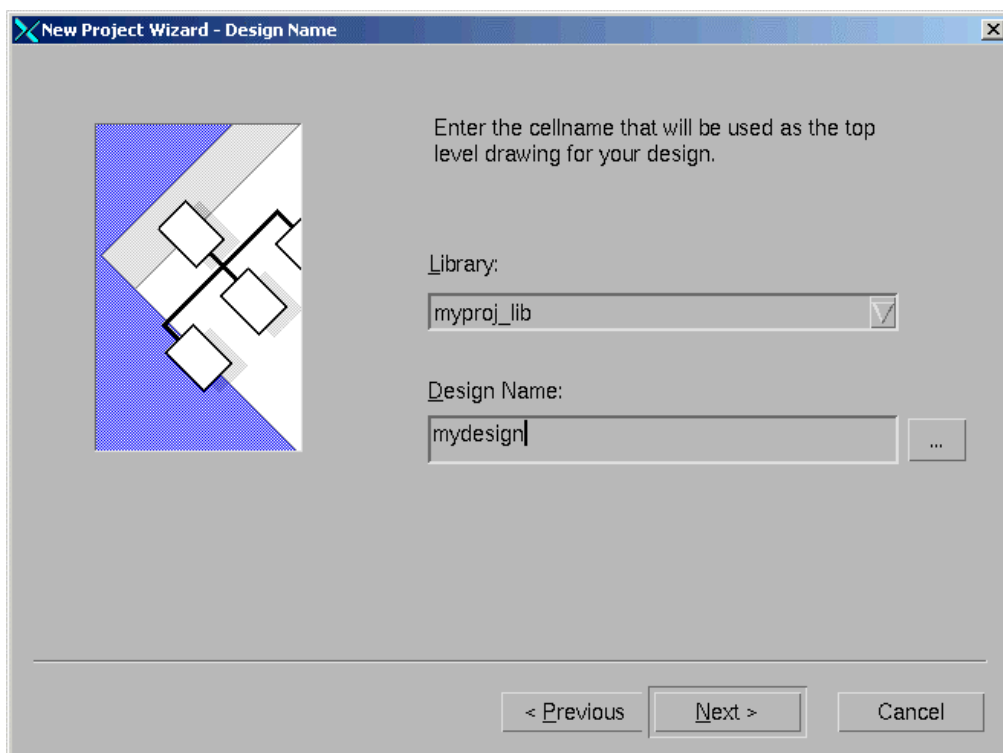
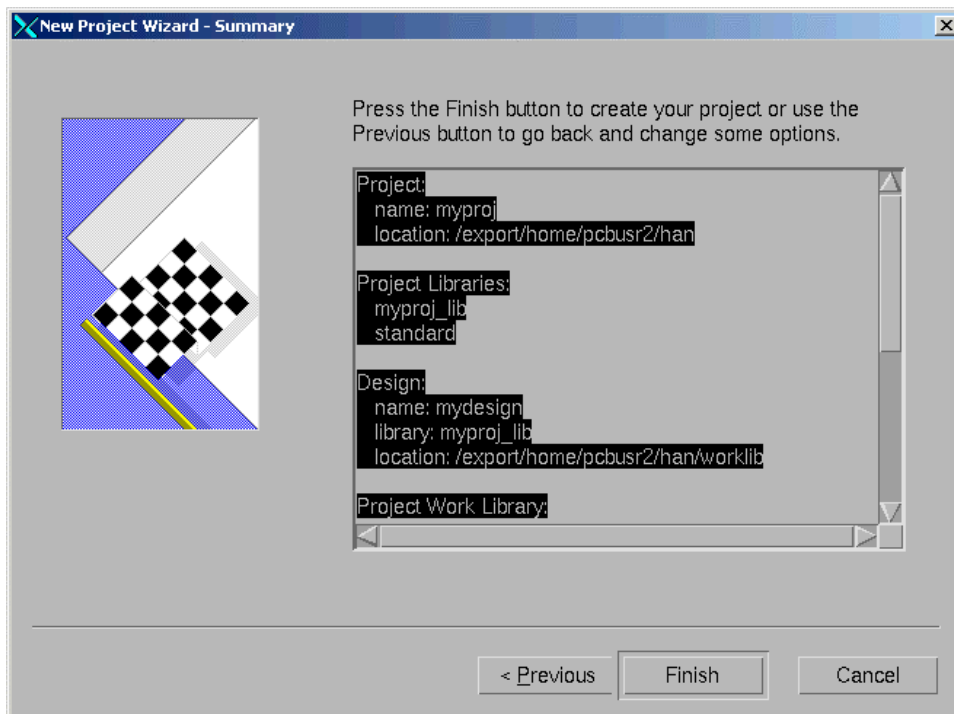


图 2.4

7. Next (确保所有的项都应被选中)



8. Finish

图 2.5

## § 2.2 项目的目录结构



打开 f:\myworking 目录，可以看到如下的结构：

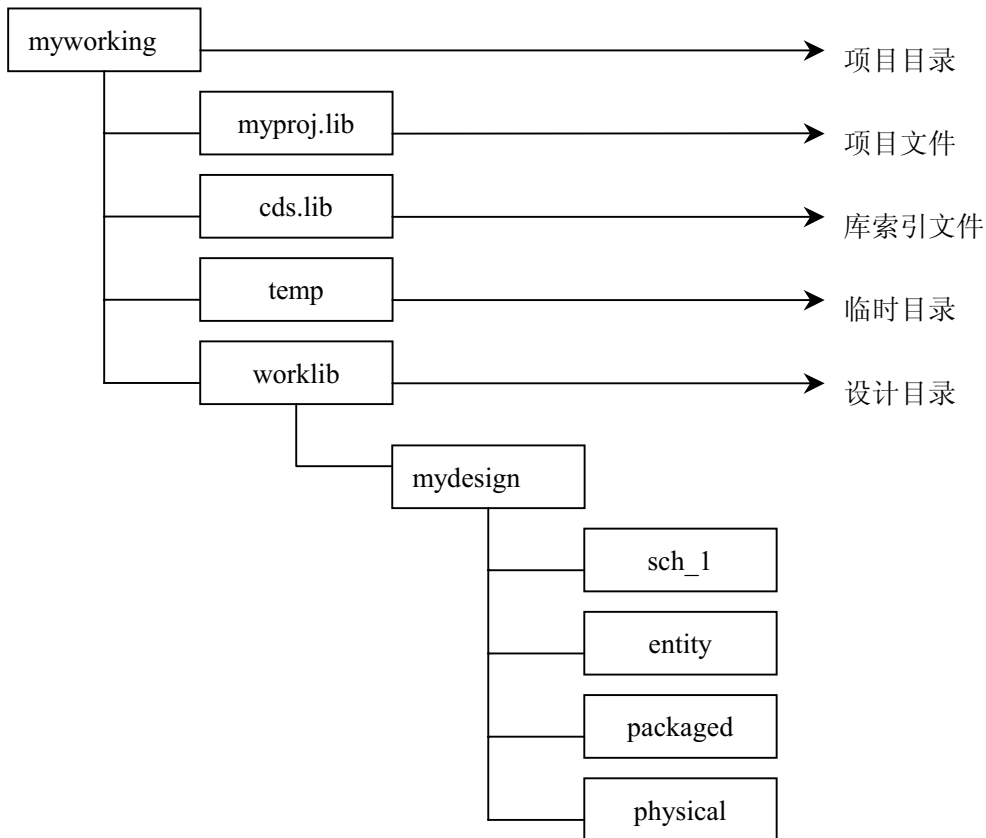


图 2.6 项目的目录结构

其中 `myproj.cpm` 是项目文件，为文本格式，该文件存放了项目的环境设置信息，如：原理图名称，项目使用的元件库及 `part_table` 文件等。项目文件可以在三个目录下存在，这三个目录分别为用户目录、特定目录、系统目录，其中用户目录的优先级最高。

`cds.lib` 为库索引文件（文本格式，可以由用户修改），指定了 Concept-HDL 所用的元件库的路径，打开的方法是点击 `setup -> Global -> cds.lib -> edit`，如图

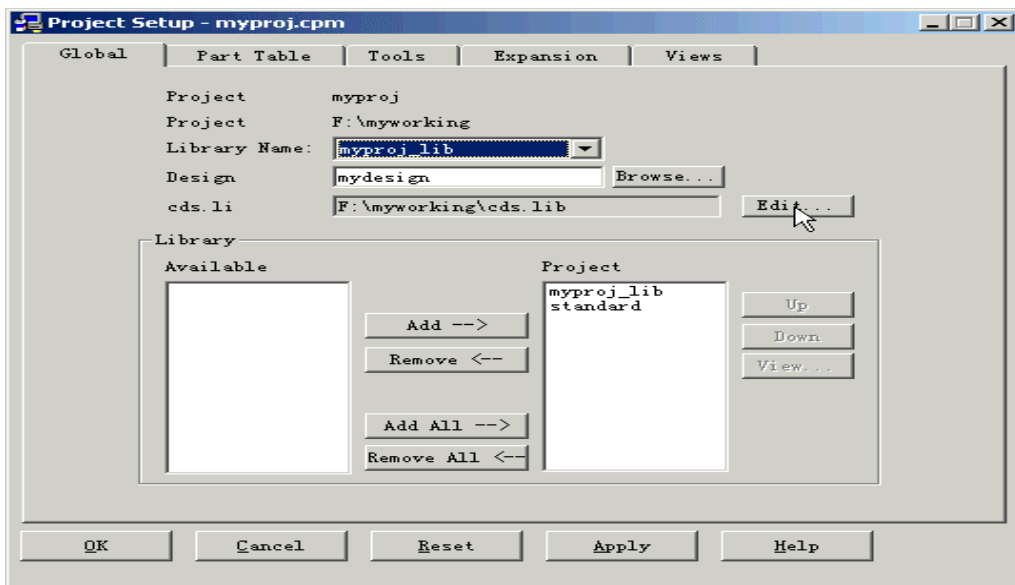


图 2.7

2.7 所示。

文件内容如下：

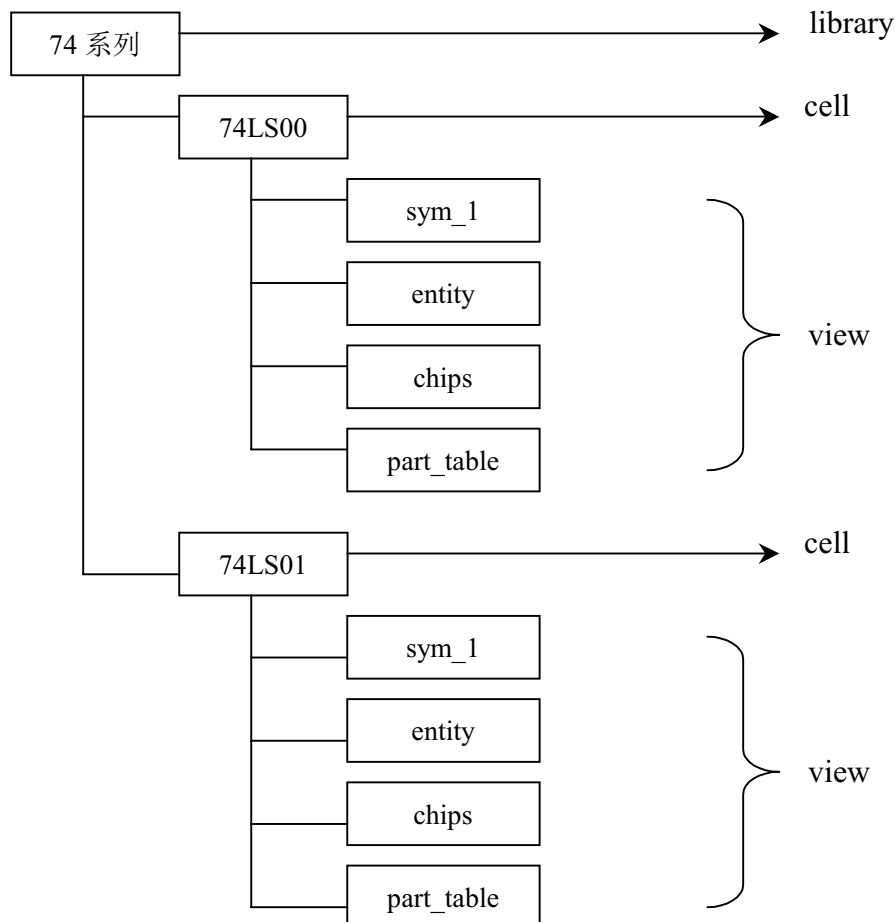
```
DEFINE myproj_lib worklib
INCLUDE E:\Cadence\share\cdssetup\cds.lib
```

该文件的每一行都由 DEFINE 或 INCLUDE 关键词定义了一个元件库的索引指针。INCLUDE 命令定义了对另一个索引文件的引用,类似于C语言主函数中的 INCLUDE 语句。如果用户自建元件库并且想在项目中使用,就需要用 DEFINE 命令来指定,格式为: DEFINE 元件库名 路径

例: 用户建了一个名为 amp 的库, 在 (d:\mylib 目录下, 则可在 cds.lib 中加入如下的语句: DEFINE amp d:\mylib\amp

## § 2.3 库的结构

Cadence 的元件库结构分为三级: 分别为 library、cell、view, 这三级实际上是三级目录。Library 包含若干个 cell, 每个 cell 又包含若干个 view。Library 这一级对应于元器件的类型, 例如可以为 74 系列、语音功放、分立元件等; cell 这一级对应于具体器件, 如果库起名为 74 系列, 则库中的 cell 可起名为 74LS00、74LS01……分别对应于具体器件; view 是对具体器件从各个方面进行描述, (系统默认的名字有 sym\_1, entity, chips, part\_table 等。sym\_1 存放的文件用来描述元件符号; entity 存放的文件用来描述文件端口的高层语言描述; chips 目录下的文件是对元件的物理封装的说明和层次描述; part\_table 目录下的文件存放元件的附加属性, 用于构造企业特定部件。层次图如下:



在图 2.6 中，可以发现项目结构中还有一个名为 `worklib` 的目录，该目录是创建项目时系统自动生成的，用来存放原理图文件，PCB 文件等，该目录实际构成了该设计的设计库，起结构图同一般的元件库一样，只是 `cell` 和 `view` 的名称有所改变，其中 `sch_1` 用来存放原理图；`entity` 用来存放原理图端口的高层语言描述，`packaged` 存放原理图的封装编译结果；`physical` 存放 PCB 设计结果，`sym_1` 存放当把该原理图作为层次图中的一个子块时，模块化后的功能符号。

## 第三章 原理图设计

在这一章里将介绍原理图的输入工具——Concept—HDL 的具体使用方法。第一节介绍 Concept 的常用工具；第二节结合一个实例说明工具的具体用法；第三节介绍层次原理图的设计。

### § 3.1 Concept-HDL

#### 3.1.1 Concept—HDL 的启动

按第二章中介绍的方法创建一个项目后，在项目管理器中点击 Design Entry，则可进入 Concept—HDL，如下界面：

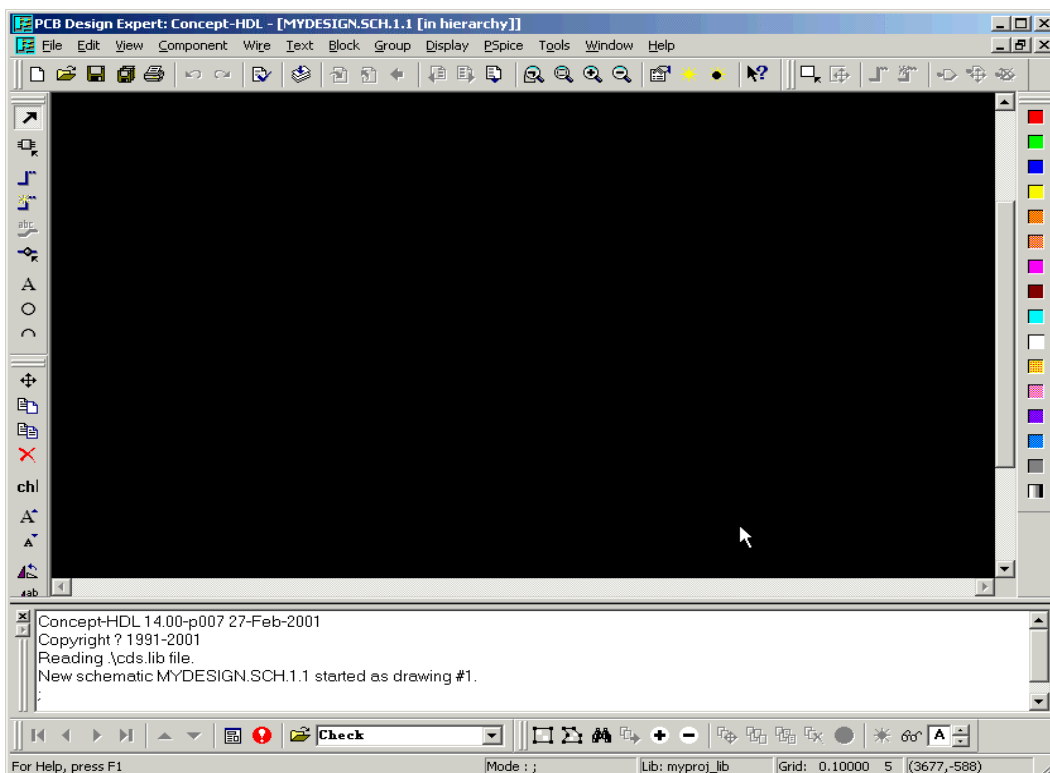


图 3.1

在标题栏中显示的是将要编辑的原理图文件信息，即：mydesign.sch.1.1。它的格式为<Cell>.<View>.<Version>.<Page>。这里的 Version 并不是原理图经过多次修改后每次的版本号，而是将该原理图作为一个单一的逻辑功能元件时，出现在原理图中的元件符号版本。

#### 3.1.2 添加元件

Cadence 中的元器件不仅包括一般的物理意义上的元器件实体，还可以包括逻辑上的一些符号，如总线连接符号、总线拆分符号等，甚至图纸也是作为元器件来添加的。

一个元器件可以有不同的版本（即视图），系统默认为 1，添加元件时，点击右键选 Version 来选择不同的视图符号。一个元器件也可以有多个封装类型，添加时，选择物理方式添加，可以根据 chips 文件和 part\_table 文件中定义的封装类型来添加元件。

添加元件时点击 component->add，则出现选择器件对话框，如图 3.2

点击 `physical` 可以进行物理方式添加；如果元器件的库不在 `library` 列表中，可以点击 `search stack` 进行库的添加。除了选择菜单命令外，还可以直接从 `ADD` 工具栏中点击添加器件按钮



图 3.3

也可以直接在命令窗口中输入 `add` 命令



图 3.4



图 3.2

### 3.1.3 画线

Concept-HDL 的画线有两种方式，一种是手动画线（Draw）在连线的同时，点击右键可输入信号名（网络标号），另一种是自动画线（Route），分别点击线的起点和终点，线会自动连接。这两种都在 `wire` 菜单中。在 `Draw` 方式下双击左键或点击右键->`next` 画下一条线，右键->`done` 则结束本次画线。

### 3.1.4 添加信号名

点击 `Wire->Signal Name...` 出现添加信号名对话框，依次键入要添加的全部信号的名称，在信号名所在的信号线上点击左键（使十字光标与信号线重合）则完成一次添加，添加后信号名会自动从对话框中消失。

如果用户不添加信号名，存盘时系统会自动给没有信号名的网络添加信号名，但系统添加的信号名冗长繁琐，建议用户对每个网络都添加信号名以利于后来的 PCB 设计及仿真。

### 3.1.5 总线信号名的命名规范

对于总线，命名格式为：信号名<最高位..最低位>。例如 8 位数据总线可命名位 `D<7..0>`。给一条信号线添加总线信号名后，线会自动加粗以区别于一般信号线。

一个信号名后加后缀 `N` 表示该信号是一个逻辑低有效的信号，在信号名后加上 `"G"` 或在信号名前加上 `"I"` 或 `"!"` 都表示该信号是一个全局有效的信号。

对于数字元件电源和地管脚已在 `Chip.prf` 文件中进行了说明。为了取代 `Chips.prt` 文件中定义的 `POWER_PINS` 属性，可以在原理图中给元件添加 `POWER_GROUP` 属性。如图 3.5



图 3.5

### 3.1.6 添加属性

Cadence 中的属性有四种类型：

- \*System properties-是由系统所指定的，分配给 wires(nets)或 pin。
- \*Schematic properties-是由用户指定的，分配给 Component、wires(nets)、或 pin。
- \*Symbol properties-是在元件库中指定的。
- \*Occurrence properties-由用户指定，可以对同一属性指定不同的值。

以下是几种常用的属性的名称：

PATH: 器件在原理图中的标号，放置元件时系统自动添加，如 I12

SIG\_NAME: 信号名，添加信号名时系统自动生成，也可以作为属性添加

LOCATION: 元件所属的物理器件的名称，如 U2

BODY\_NAME: 器件的物理名称，如 74ls00

点击 Text->Property 出现添加属性对话框，填入属性的名称和值，点击 OK，鼠标指针变成十字，点击想要添加属性的器件、管脚或网络，完成一次添加。

如果要添加多个器件，选择 Text->Attribute，再点击想要添加属性的器件、管脚或网络，出现添加属性列表，如图 3.6

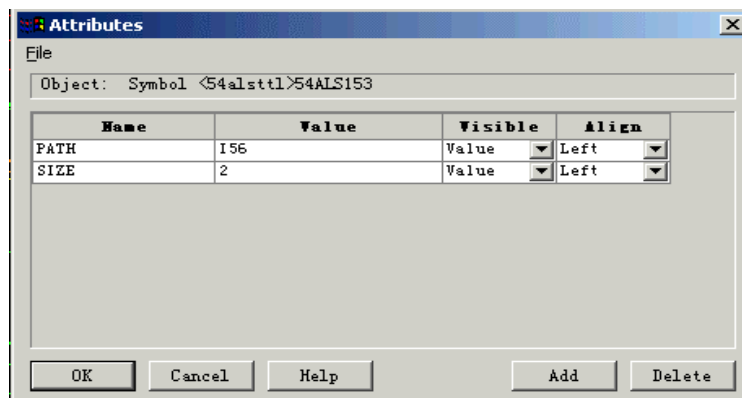


图 3.6

填入想要添加的属性名称和值，然后在 Visible 中选择显示类型，是名称和值都显示 (Both)，还是都不显示 (None) 或只显示一种 (Name 或 Value)，在 Align 中选择对齐方式，点击 OK，完成添加。

### 3.1.7 常用的操作

在 View 菜单中有各种命令可以对图纸进行放大，缩小，局部放大，这些命令都有其对应的热键，读者可在下拉菜单中找到。

### 3.1.8 器件操作

在 Component 菜单中的命令可以对器件和属性进行编辑操作，这些命令也

可以直接从 Edit 工具栏中选取，如图 3.7。

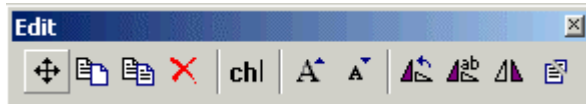


图 3.7

### 3.1.9 组操作

组是原理图中某些元素的集合。这些元素可以是器件、网络、属性、注释，但不能是管脚。定义了组以后，用户可以方便地对组中的所有元素进行操作。



图 3.8

有三种定义组的方法：

- \*矩形框
- \*多边形框
- \*表达式

这三种方法都可以在 Group 菜单中找到。创建一个组以前，用户应用一个字母先给组命名。否则下次定义的组会覆盖本次的结果。名称选择可以点击 Group->Set Current Group，也可从 Group 工具栏，图 3.8 中选择。

定义了组以后，用户可以对组进行 Move, Delete, Copy, Replace, Display 等各种操作，但再次之前必须设定当前要进行操作的组。方法和定义组时的操作一样。

### 3.1.10 错误检查

Concept-HDL 具有强大的差错功能，在默认情况下，系统会在每次存盘时对原理图进行错误检查。检查类型包括电连接检查、图符检查、名字检查、其他检查，用户可以在 Tools->Option->Check 标签下进行设定。用户也可点击 Tools->Check 进行实时检查。另外 Cadence 还提供了更强大的查错工具——CheckPlus，它在 Project Manager->Tools 下，具体的设定和使用，用户可以查阅帮助文件（安装目录\doc\checkplus\checkplus.pdf）。

### 3.1.11 点画命令

点画命令是 Cadence 独特的一种快捷方式，可以直接用鼠标在屏幕上写特定的笔划，不同的笔划表示不同的命令。例：Z 表示放大，W 表示整体显示。用户可以根据自己的习惯来定义笔划。方法是在命令窗口敲入 stroke，出现一个设定对话框，如图 3.9，如何定义可以参看帮助。

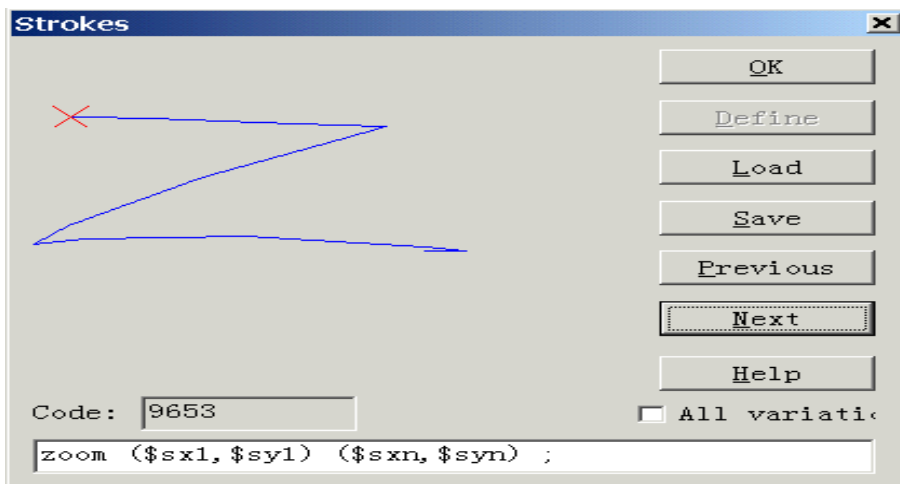


图 3.9

### 3.1.12 原理图展开

在层次和多页原理图设计中，原理图的展开可以使用户深入到设计的每一层每一页，以便对设计进行全局的跟踪、查找。原理图展开后，用户的设计将成为一个整体保存在数据库中。展开的方法是 Tool->Expand Design

原理图展开后，用户可以进行器件、网络的查找。方法是 Tool->Global Find，如图 3.10，在 Name 中输入想要查找的网络或器件名，点击 Find，点击列表中的器件或网络，该器件或网络就会 Highlight 显示。其中 Hierarchical Name 和 Library Location 是查看方式选项。

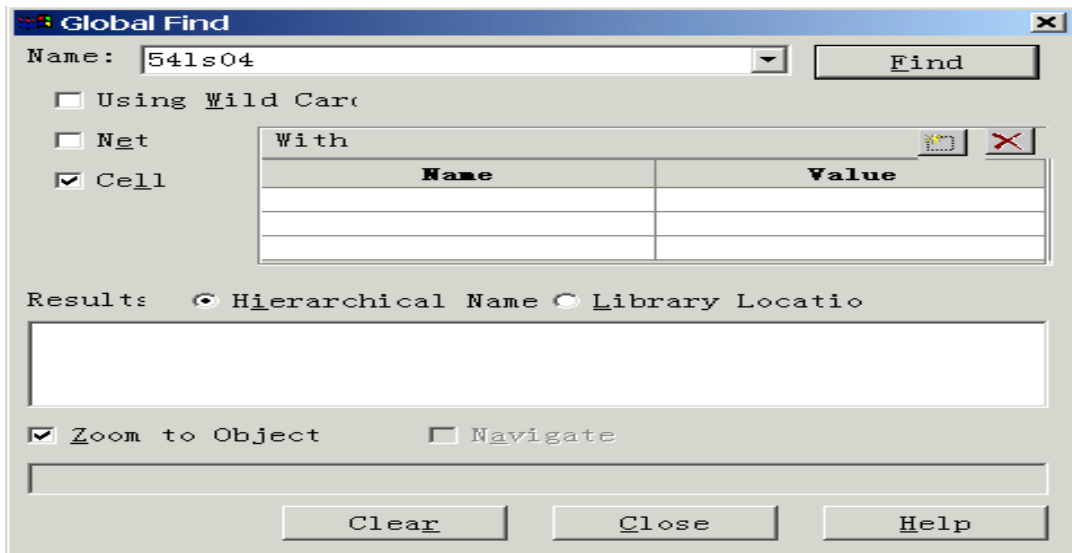


图 3.10

跟踪网络的方法是 Tool->Global Navigate，出现对话框后，点击任意 wire，则与该 wire 相连网络的会被加入列表，点击列表中的任意网络均会被 Highlight 显示。

### 3.1.13 Standard 库

这里介绍在 Standard 中一些常用符号的用法。

\*Bus Tap: 用来指明某根单线在总线中的位置。例：有一条总线 D<3..0>，要将 4 条单线从中分出。选 Wire 菜单->Bus Tap，分别在总线上引出 4 根线后，如图 3.11，每条线上均有一个“？”，“？”是用来添加 BN

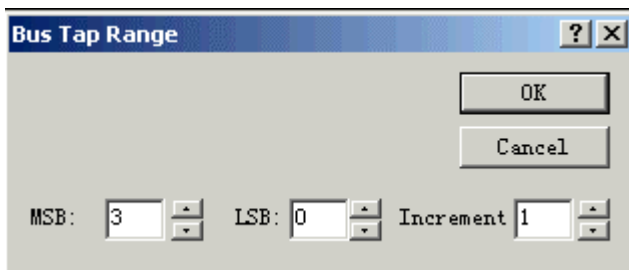


图 3.12

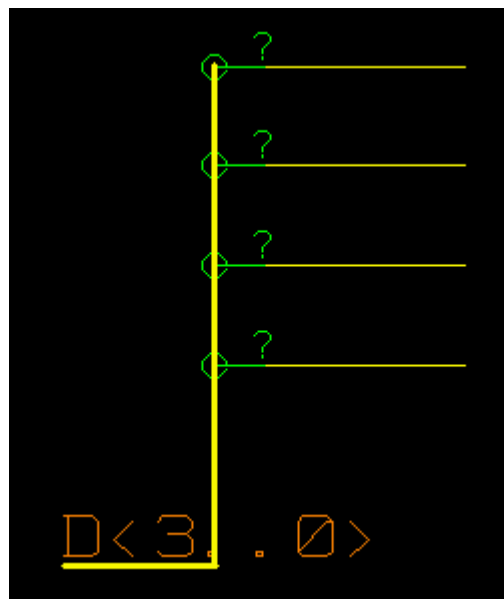


图 3.11

（位号）属性的值。添加方法是 Wire->Bus Tap Value。出现如图 3.12 的对话框。在 MSB（最高位）中填入 3，在 LSB（最低位）中填入 0，Increment（步进值）



中填入 1，用鼠标划一条线，穿过 4 条线，从高位到低位，如图 3.12，松开鼠标后，“？”会自动变成位号。如图 3.13

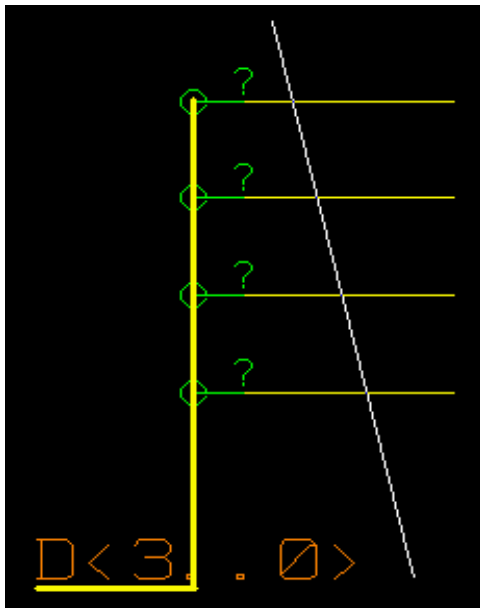


图 3.12

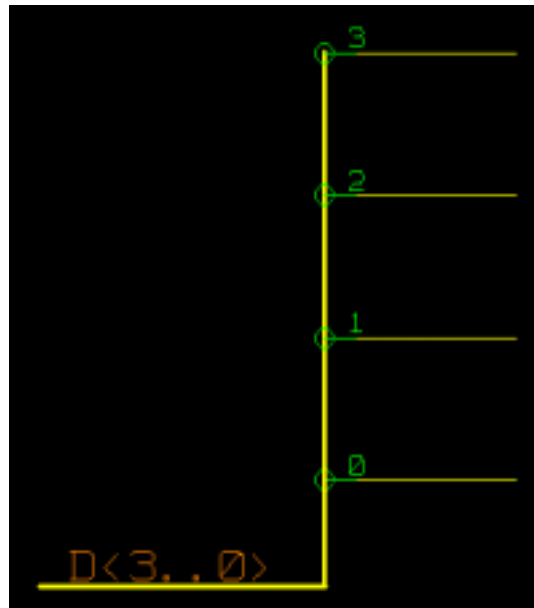


图 3.13

\* **Merge:** 用来合并或分割总线。输入总线的宽度和输出总线的宽度必须一致。具体的用法和规则可以查看 `Concept-HDL` 的帮助文件 (`Help->On Line Book->Printable version of Concept-HDL`)。

\* **Slash:** 用来指明总线的宽度。

\* **Alias:** 用来给同一个网络起不同的名字。

### 3.1.14 常用快捷键

系统默认的一些快捷键有：

F1: 帮助

F2: Windows Fit 显示

F3: Display Both(属性的 Value 和 Name 都显示)

F4: Display Attachment (显示连接关系)

F5: Refresh (刷新屏幕)

F6: Show Property (显示属性)

F11: Windows in (放大)

F12: Windows out (缩小)

Ctrl+方向键: 移动屏幕 (也可用鼠标右键)

在 `Tools->Customize->Key` 中可以自己定义快捷键。

## § 3.2 一个实例

本节通过一个 7 段显示码转换成 BCD 码的例子来使读者进一步熟悉 `Concept-HDL` 的基本用法。

### 3.2.1 添加库

1. 在 Add 工具栏中选择添加按钮。

2. 在添加器件对话框中，点击 `Search->Stack->Edit`，添加 `54aLsttl` 库和 `element` 库。

### 3.2.2 添加图纸

1. 在添加器件对话框中，将 Library 选为 Standard。
2. 从列表中选择 B SIZE PAGE 并关闭元件浏览器，B SIZE PAGE 出现在光标处。
3. 点击左键放置图纸
4. 按 F2 键，Windows Fit 显示。

### 3.2.3 添加文本注释

1. 选择 Text->Note
2. 在注释窗中输入 SEGTOBCD
3. 将其放在图框的 TITLE 项中，其它的项也可照此输入。
4. 选择 View->ZOOM Fit。

### 3.2.4 添加元件

1. 选择 Component->Add，并将 Library 设为 54alsttl。
2. 选择 Ls153，点击左键放置在原理图中。
3. 点击右键 Done。
4. Edit->Copy，左键点击 Ls153，另外复制一个 Ls153，放置在上一个的右边。
5. 依次放置 4 个 ls04 和一个 ls02。如图 3.14

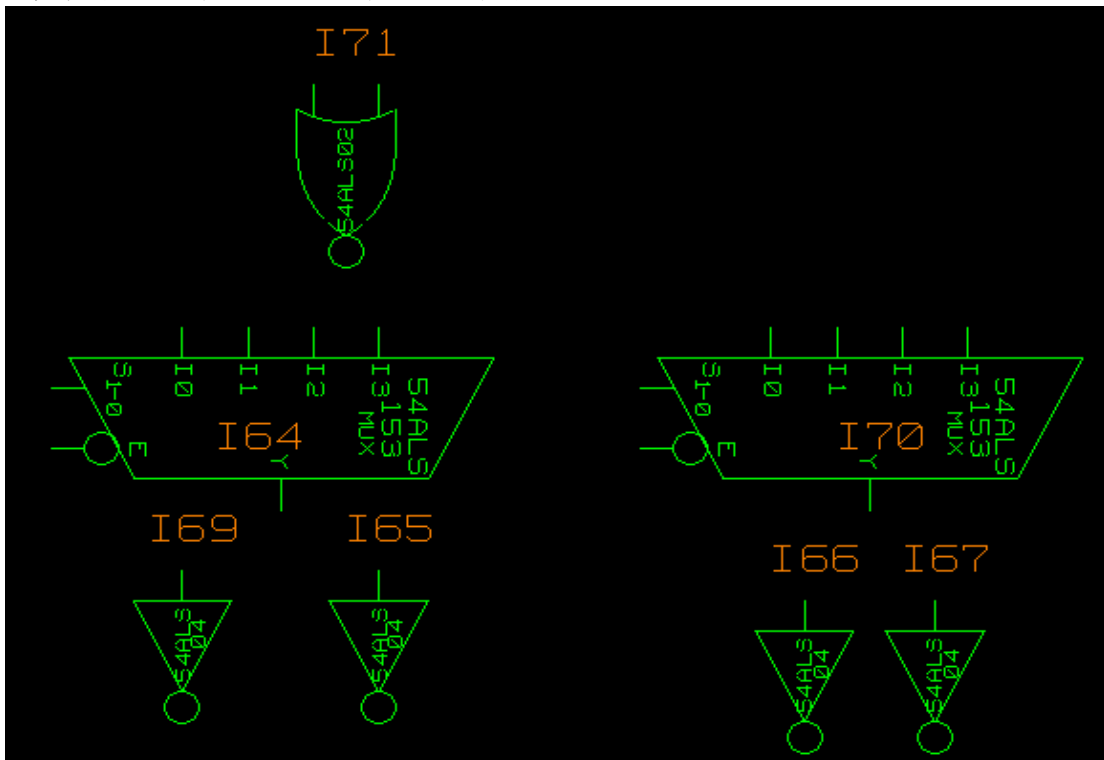


图 3.14

6. Edit->Move 将元件的位置调整好。

### 3.2.5 画线

1. 先画一条总线，再用 wire->signal name 添加信号名 SEG<4..0>，则总线自动加粗。
2. 再用 wire->Bus Tap 从总线中分出 5 条信号线，用 Wire->Bus Tap Value 分别给 5 条信号线加上位号 (BN)，也可用 Text->change 分别将 “?” 改成位号。如图 3.15 所示：
3. Ls153 的 I0—I3, Y 以及管脚在定义时是作为总线类型定义的，S1—0 管脚则是一个可以由用户指定宽度的管脚。（双击 Ls153 可以看到其管脚的定义方式，选择 File->Edit Hierarchy->Ascend 回到上一级）。因此 I3—I0 和 Y 管脚

的信号名必须为总线型，分别定义为 1D?<2..1>、2D?<2..1> (? 从 3-0) 和 Y?<2..1> (? 从 2-1)。用 Text->Attribute 给两个 Ls153 加入 SIZE 属性，属性的值为 2，如图 3.16 所示，S1-0 的信号名为 A?<1..0> (? 从 1-2)。

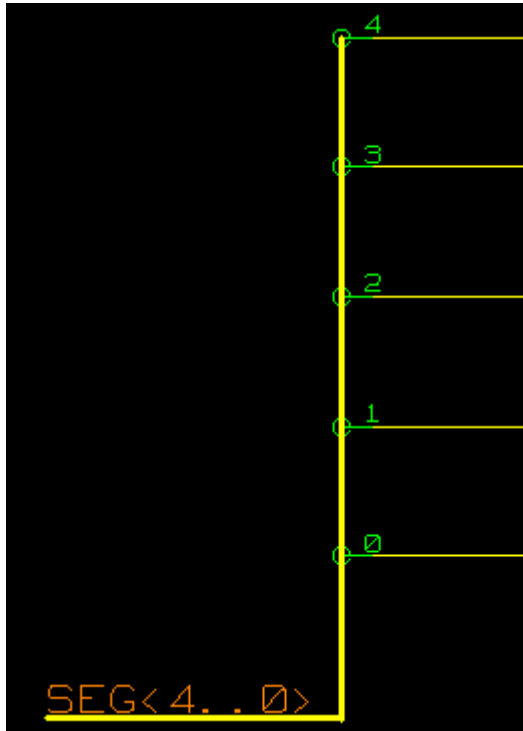


图 3.15

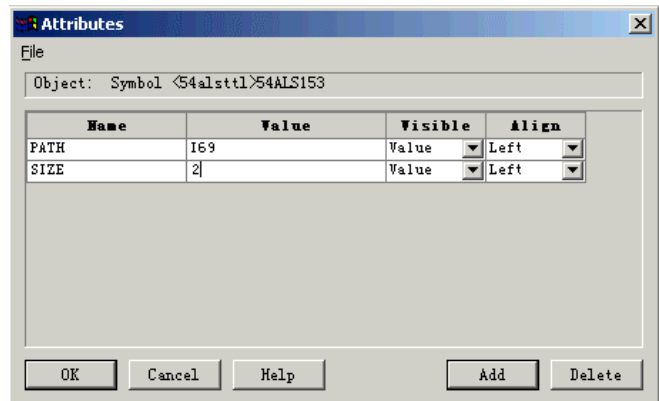


图 3.16

4. 分别从上面定义的总线中用 Bus Tap 分出各自的信号线，用 Bus Tap Value 给 Bus Tap 标上 BN 属性的值，如图 3.16
5. 将剩余的线连在总线 SEG<4..0>上。如图 3.17，其中地和 Vcc 从 Element 库中选取。

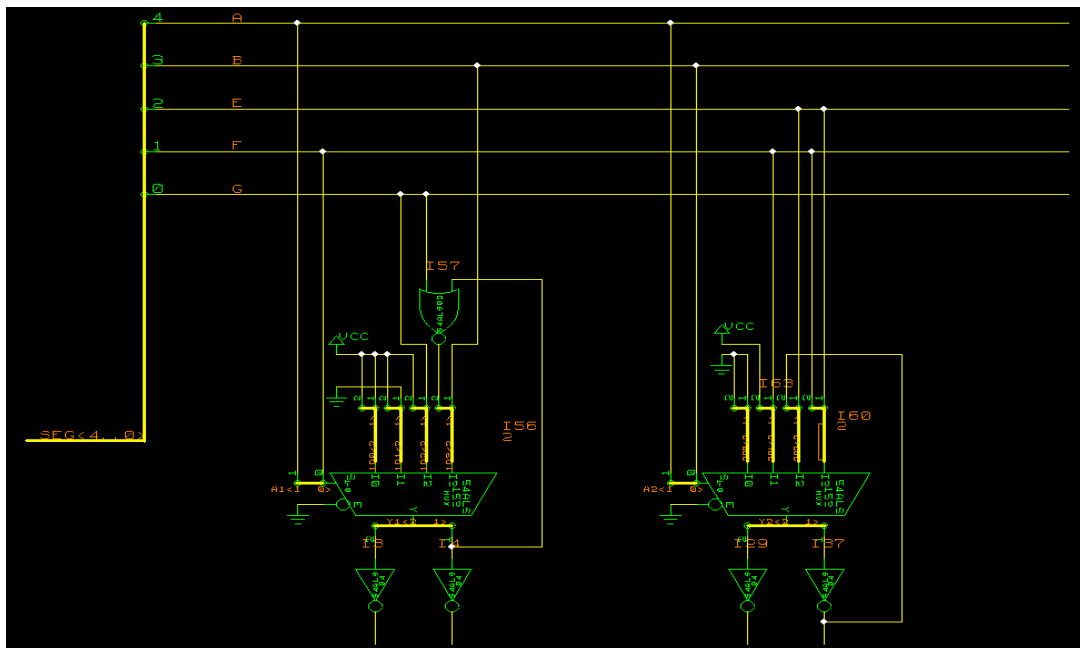


图 3.17

### 3.2.6 查错

1. File->Save, 提示有三个错误, View Error 后出现如图 3.18 的对话框
2. 点击每个错误, 则系统自动定位到错误处, 并用方框标记出来。
3. 这里错误的原因是有 3 个 Is04 的输出管脚没有和任何网络相连而且又没有信号名标注。

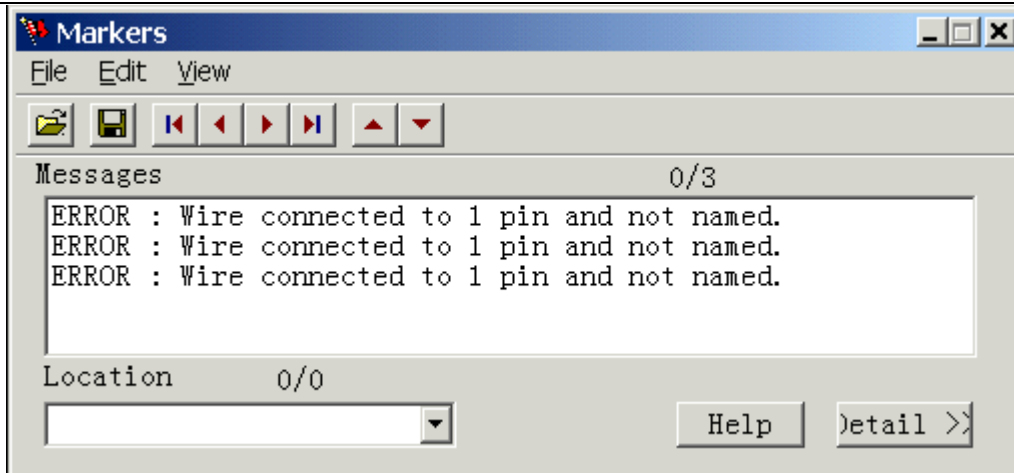


图 3.18

4. 给 4 个 LS04 的输出管脚分别加上信号名 OUT3—OUT0，如图 3.19，再次存盘，则不再出现错误提示。

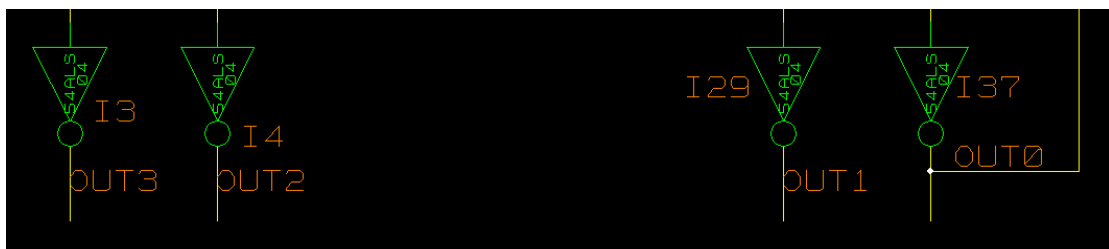


图 3.19

### 3.2.7 设计展开

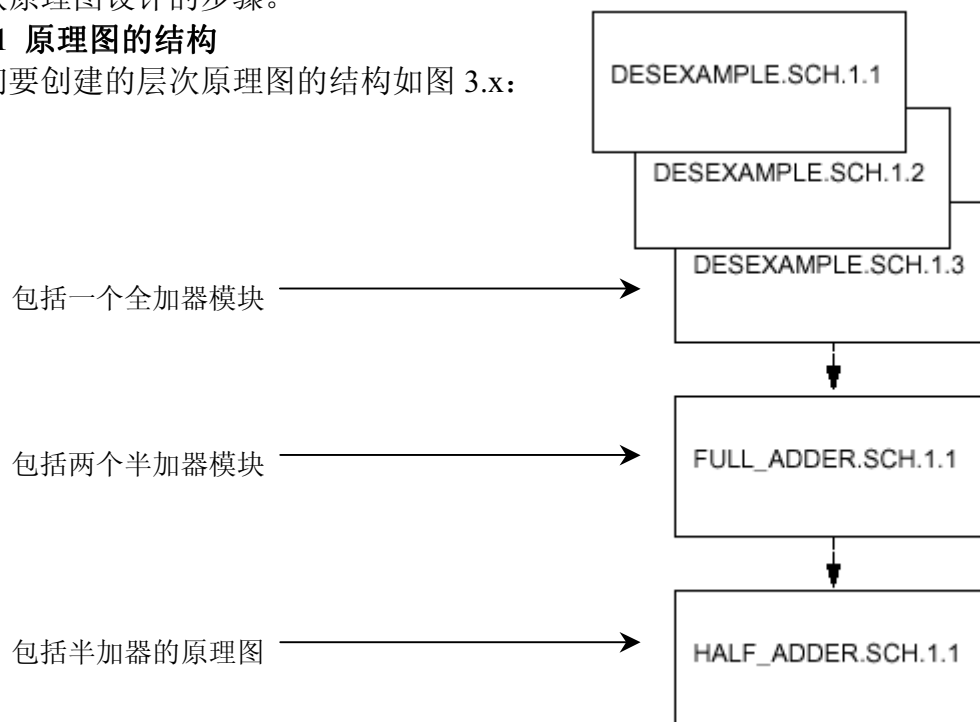
1. Tools->Expand Design，将设计展开
2. Tools->Global find 可以进行器件和网络的查找。

## § 3.3 层次原理图的设计

在这一节里，用一个自顶向下 (Top-Down) 的例子介绍一下 Concept-HDL 层次原理图设计的步骤。

### 3.3.1 原理图的结构

我们要创建的层次原理图的结构如图 3.x:



首先进行 DESEXAMPLE.SCH1.3 的设计，在 DESEXAMPLE.SCH1.3 中添加一个名为 FULL\_ADDER 的模块，然后为这个模块创建一个原理图叫 FULL\_ADD.SCH.1.1，在这个原理图中又有两个名为 HALF\_ADDER 的模块，最后，我们为 HALF\_ADDER 创建原理图。

### 3.3.2 创建 FULL\_ADDER 模块

1. 确认当前的文件是 DESEXAMPLE.SCH1.1。
2. 加入新的一页，选择 File->Edit Page/Symbol->Go To，出现图 3.20 对话框。
3. 在 Page/Symbol 中输入 3，点击 Ok
4. Concept 自动生成一个空白的工作区。
5. 加入一个模块，选择 Block->Add
6. 画一个矩形的模块，默认的名称为 BLOCK1
7. 右键选择 Done。

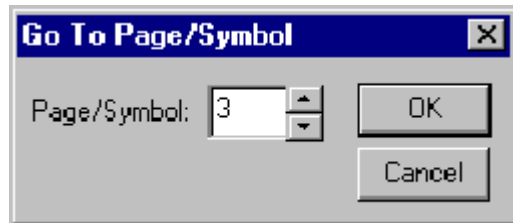


图 3.20

8. 给模块命名，选择 Block->Rename，出现如图 3.x 对话框：
9. 在 Block Name 中输入 FULL\_Block，点击鼠标将 Block1 换成 FULL\_ADDER。如图 3.21。

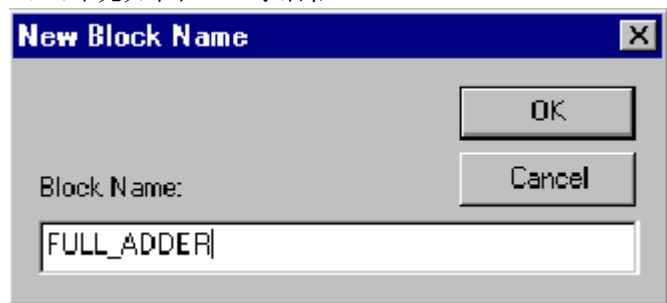


图 3.21

10. 选择 Block->Add Pin，选择 Pin 的类型为 Input。输入下列管脚名：input1、input2、carry\_in，依次点击鼠标把 Pin Name 放在模块的左边，如图 3.22。
11. 同样的方法放置输出管脚。

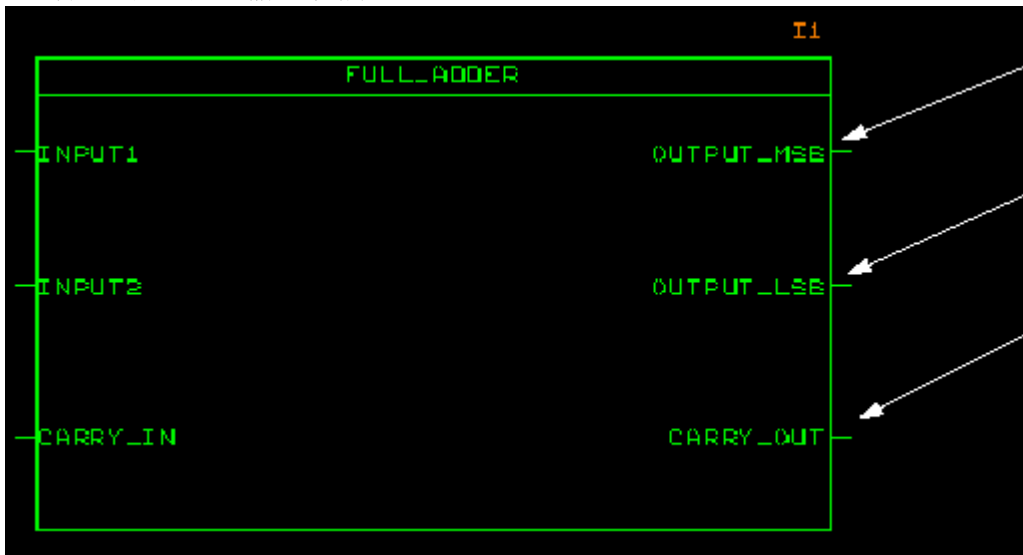


图 3.22

12. 分别给各个管脚分配信号名：IN1、IN2、CIN、COUT、OUT\_LSB、OUT\_MSB。加入信号名后的原理图如图 3.23。

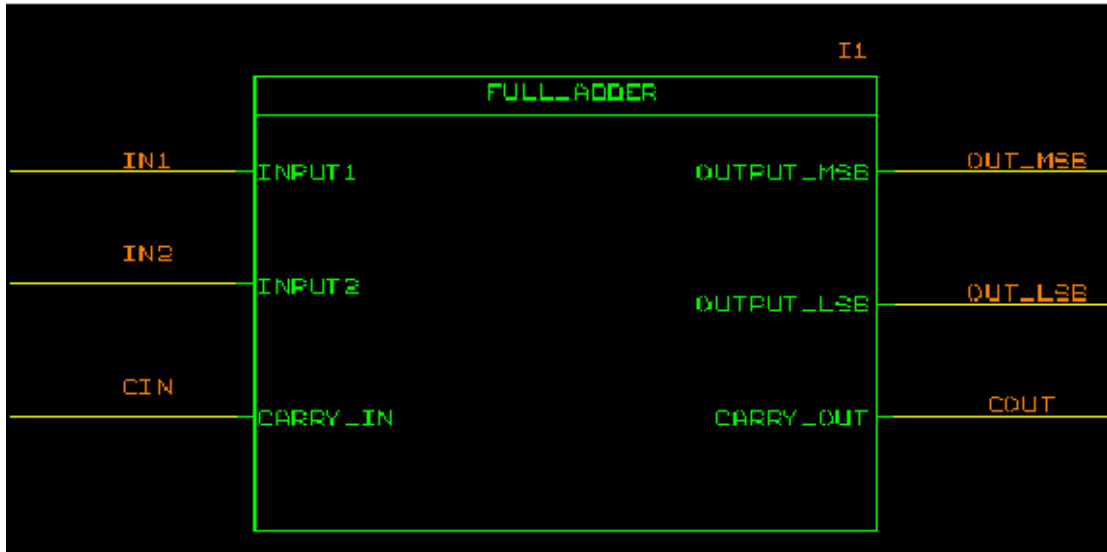


图 3.23

13. 入端口，从 Standard 库中选择 INPORT 添加在 IN、IN、CIN 信号名所在的线上。同样的方法添加 OUTPORT。如图 3.24

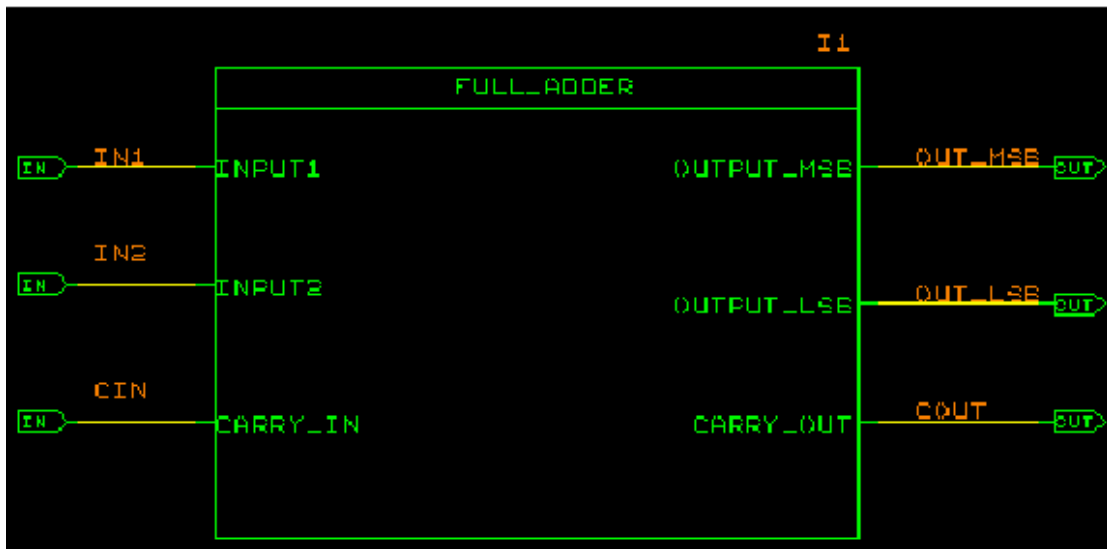


图 3.24

14. 选择 File->Save

### 3.3.3 创建 FULL\_ADDER 的原理图

1. 选择 File->Open。
2. 输入 FULL\_ADDER 作为 Cell。
3. 选择 Schematic 作为 View，如图 3.25。
4. 点击 Open，则当前页即为 FULL\_ADDER 的原理图页。
5. 用上面的方法加入两个名为 HALF\_ADDER 的模块，如图 3.26

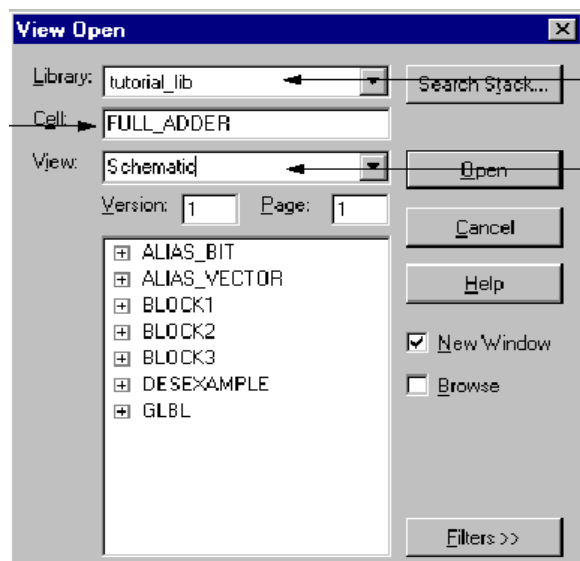


图 3.25

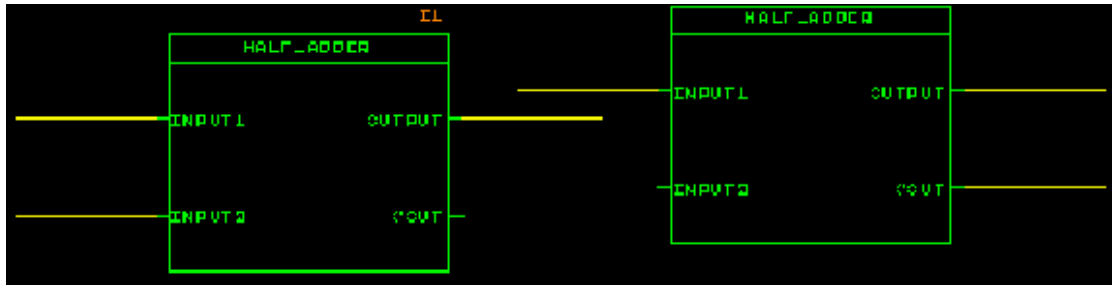


图 3.26

6. 将 I1 的 COUNT 和 I2 的 INPUT2 相连。
7. 给剩余的 6 根线加上信号名和端口，如图 3.27。

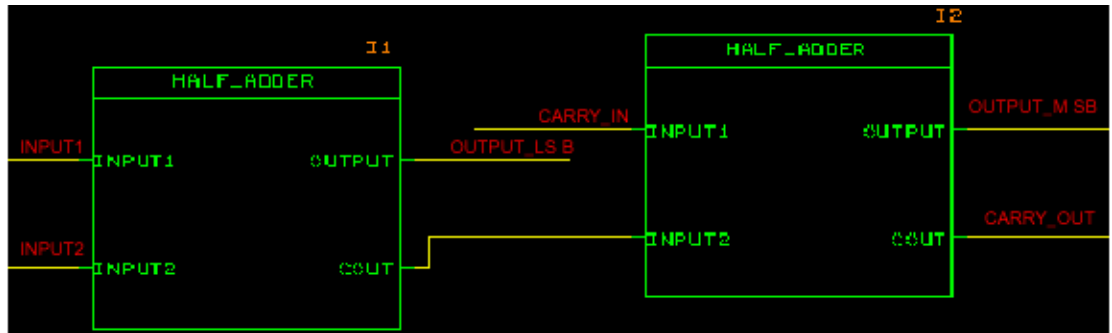


图 3.27

8. File->Save
9. 选择 File->Edit Hierarchy ->Ascend，Concept - HDL 显示 DESEXAMPLE.SCH1.3 中包含了一个 FULL\_ADDER，双击 FULL\_ADDER，则 FULL\_ADDER 展开成 FULL\_ADDER.SCH.1.1

### 3.3.4 创建 HALF\_ADDER 的原理图

1. 选择 File->Open，出现如图 3.28 的对话框。
2. 选择 HALF\_ADDER 作为 Cell

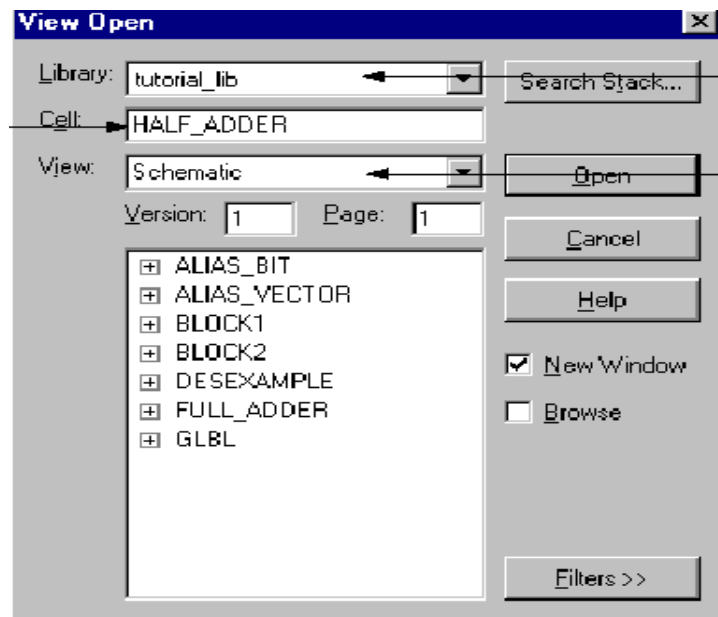


图 3.28

3. 选择 Schematic 作为 View，点击 Open。
4. 用创建一般原理图的方法画出 HALF\_ADDER 的原理图。如图 3.29

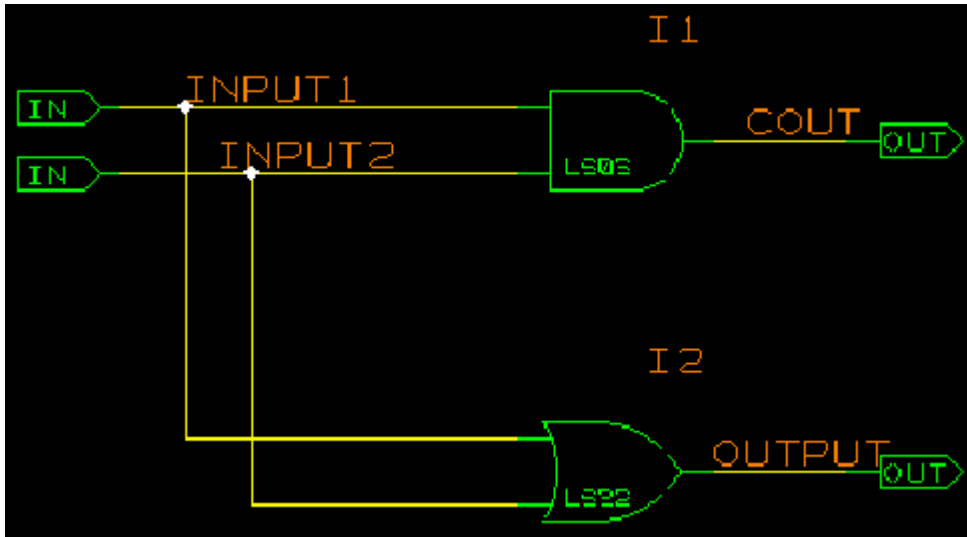


图 3.29

5. 选择 File->Save

### 3.3.5 浏览设计

1. 选择 File->Open, DESEXAMPLE.SCH1.3 作为 Cell, Schematic 作为 View, 点击 Open, 则可看到 FULL\_ADDER 模块, 双击 FULL\_ADDER 模块, 则可看到两个 HALF\_ADDER 模块, 双击 HALF\_ADDER 模块则可看到最低层的原理图。

2. 选择 File->Edit Hierarchy->Ascend, 则回到上一级。

注: Concept-HDL 中还有许多的命令和具体的操作以及各种描述文件的结构和作用由于篇幅、时间和个人能力的关系没能在此说明。读者可以查阅 Concept-HDL 的 Help 文件 (Project Manager->Help->On line books->Printable Concept HDL) 获得更多的信息。

## 第四章 从原理图到 PCB

这一章主要介绍从原理图到 PCB 的过渡方法以及其中的一个关键的工具—Package—XL 的使用方法。

### § 4.1 Package—XL

#### 4.1.1 Package 的作用

Package—XL 是一个进行器件封装的工具, 它主要有三个作用:

1. 将原理图中的逻辑器件映射成物理器件 (即封装)
2. 生成材料清单
3. 进行 ERC 检验

#### 4.1.2 器件封装

所谓封装就是为几个在物理上属于同一器件的逻辑单元指定同一个位号。在 Concept 中是通过指定 Location 属性来实现的, 具有相同 Location 属性的器件会被封装在一起。

用户可以手动封装, 也可由 Concept 自动封装。手动封装就是用户给逻辑器件添加 Location 属性 (前面没有 "\$"), 并给其赋值。自动封装就是由 Concept 给器件指定 \$Location 属性及赋值 (注意前面有个 "\$")。如果用户不加说明, 则 Concept 会按照生成的物理器件最少的原则进行封装。Location 是用户级属性,



系统在自动封装时不会修改。

在自动封装前，用户可以对自动封装的结果进行限制，例如：希望某些虽属同一类型的器件但不要被封装在一起。实现这种功能只需给器件指定 GROUP 属性，具有不同 GROUP 属性值的器件决不会被封装在一起。用户还可以指定封装后逻辑器件在实际器件中的位置，这一功能可以用 Section 命令实现（Component->Section 点击器件，则器件的管脚号会被显示出来）。

#### 4.1.3 封装类型的确定

一个元件可能有多种封装，在添加元件时选择物理方式添加（Physical Mode），并选择所需要的封装，如图 4.1 所示。如果添加时没有指定，则系统按照创建器件库时的默认封装添加。如果用户想改变封装类型，可以给器件进入 PART\_TYPE 属性，属性的值必须是建库时定义好的封装类型之一。

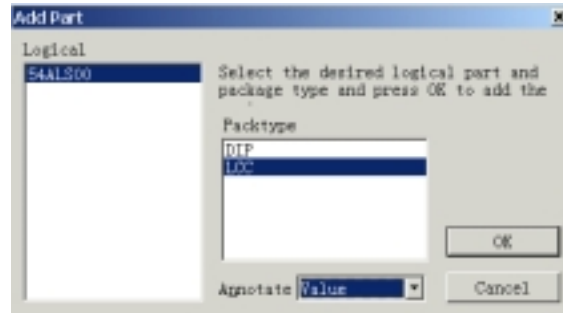


图 4.1

#### 4.1.4 自动封装的过程

Package 在对原理图进行自动封装时，自动搜索几个文件，这些文件是 chip.prt、proj.cpm、cds.lib、verilog.v。系统按照元件的物理封装说明及其所在的库以及连接关系来封装器件。

选择 File->Export Physical，出现图 4.2 的对话框。Package Option 有三个，Preserve 是不改变以前的封装结果；Optimize 是优化封装；Repackage 是不保存以前的封装结果，重新进行封装。

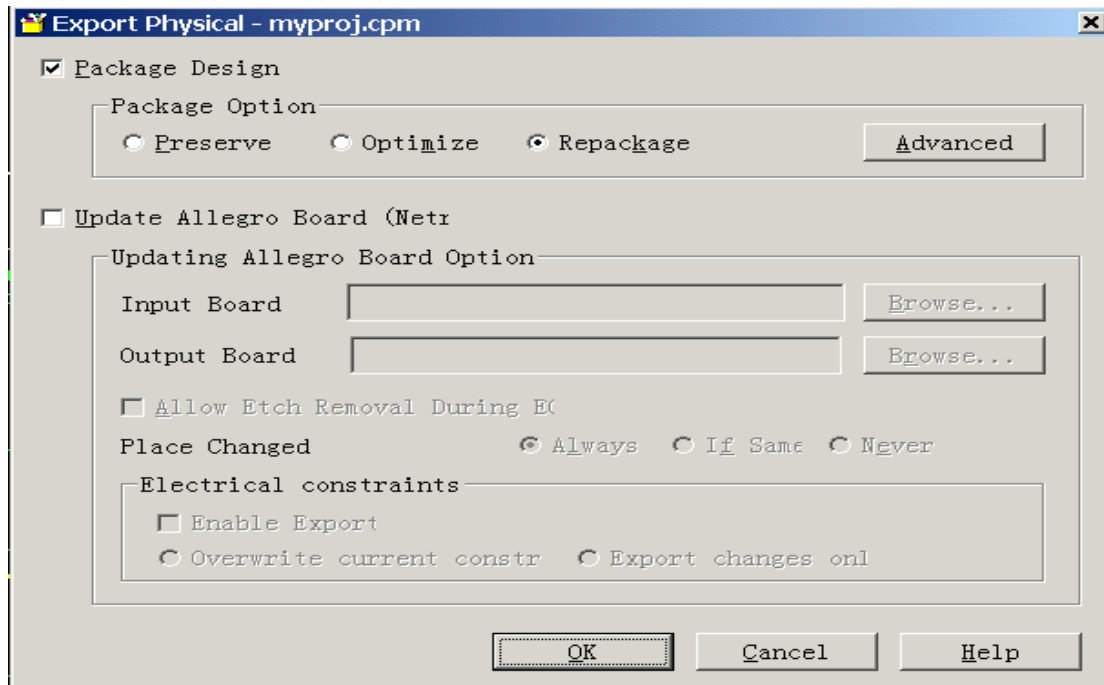


图 4.2

#### 4.1.5 反标

封装的结果并不会自动显示在原理图中，用户要通过反标（Tools->Backannotate）命令来实现这一步。反标后，器件的管脚号和其所在的物理器件名称会显示在原理图中。

## § 4.2 一个封装的实例

我们以第二章第二节的例子来说明封装的过程。

### 4.2.1 自动封装

1. 不添加任何属性，先选择 File->Export Physical，选择“否”掠过查看文件。
2. 选择 Tools-> Backannotate，不难发现 4 个 LS04 被封装在一起，名为 U3，且元件的管脚标号被显示了出来，如图 4.3

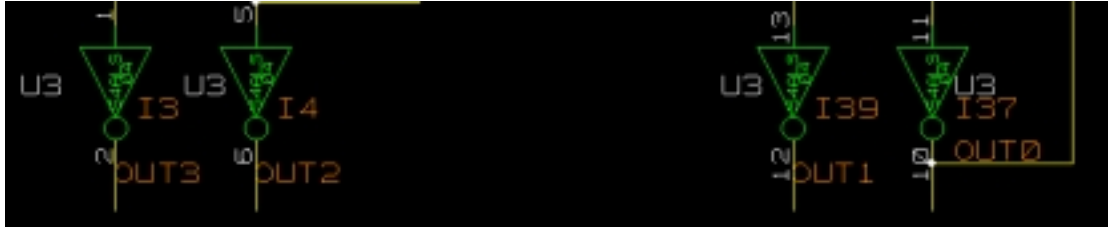


图 4.3

### 4.2.2 添加 Location 属性

1. 仅给 I3 添加 LOCATION=U1 的属性，存盘。
2. 按上面的方法重新封装并反标
3. 结果如图 4.4，4 个 LS04 还是被封装在一起（系统按照最少器件原则封装），但物理器件名变为 U1（系统没有改变用户对 I3 的设置，而其余的 LS04 按照最少器件原则均与 I3 有同样的物理器件名）。

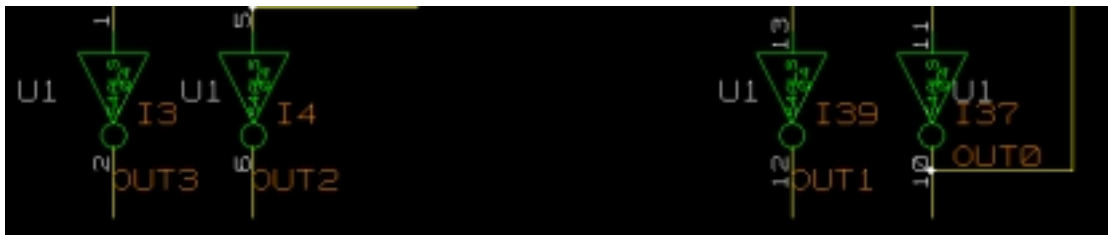


图 4.4

### 4.2.3 添加 GROUP 属性

1. 给 4 个 LS04 添加 GROUP 属性，其中 I3 的值为 a，I4 与 I39 的值为 b，I37 的值为 c，存盘。
2. 按上面的方法重新封装并反标
3. 结果如图 4.5，I3 (U1) 与 I37 (U3) 被单独封装，而 I4 与 I39 (U2) 被封装在一起。

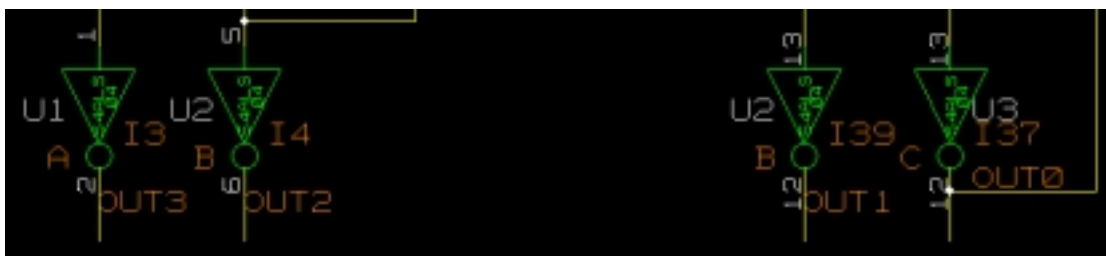


图 4.5

### 4.2.4 网表的生成

1. 选择 Tools->Package Utilities->Netlist Report 调出 Netlist Reports 窗口。
2. 点击 Run。
3. 查看结果

## 4. 退出报告

### § 4.2 设计转换

用 Package—XL 完成器件的封装以后，会生成若干文件，这些文件可分为 5 类，分别是反标文件、报告文件、网表文件、状态文件、错误文件。其中网表文件是向 PCB 转换过程中系统要调用的文件。网表文件包括三个文件：pstxprt.dat、pstxnet.dat、pstxchip.dat，它们包含了器件的封装信息、器件的连接关系及器件的电器特性。

下面举例说明转换的过程

#### 4.2.1 准备工作

1. 点击 Project Manager 中的 Layout，启动 Allegro。
2. 选择 File->Save as myboard.brd 保存一个空白的模板到 mydesign->physical 目录。（以后可以利用这以模板生成不同的 PCB 图）
3. 关闭 Allegro

#### 4.2.2 装载网表

1. 在 Project Manager 中选择 Design Sync->Export Physical（Design Differences 是用来比较原理图与 PCB 的不同的）
2. 选中 Update Allegro Board
3. 在 Input Board File Field 中输入 myboard.brd
4. 在 Output Board File Field 中输入 bcd.brd
5. 将 Place change components 设为 If Same。
6. 点击 OK。

至此，Concept 的原理图已经导入到 Allegro 的模板中。

## 第五章 ALLEGRO 做 PCB 设计

### § 5.1 ALLEGRO 用户界面。

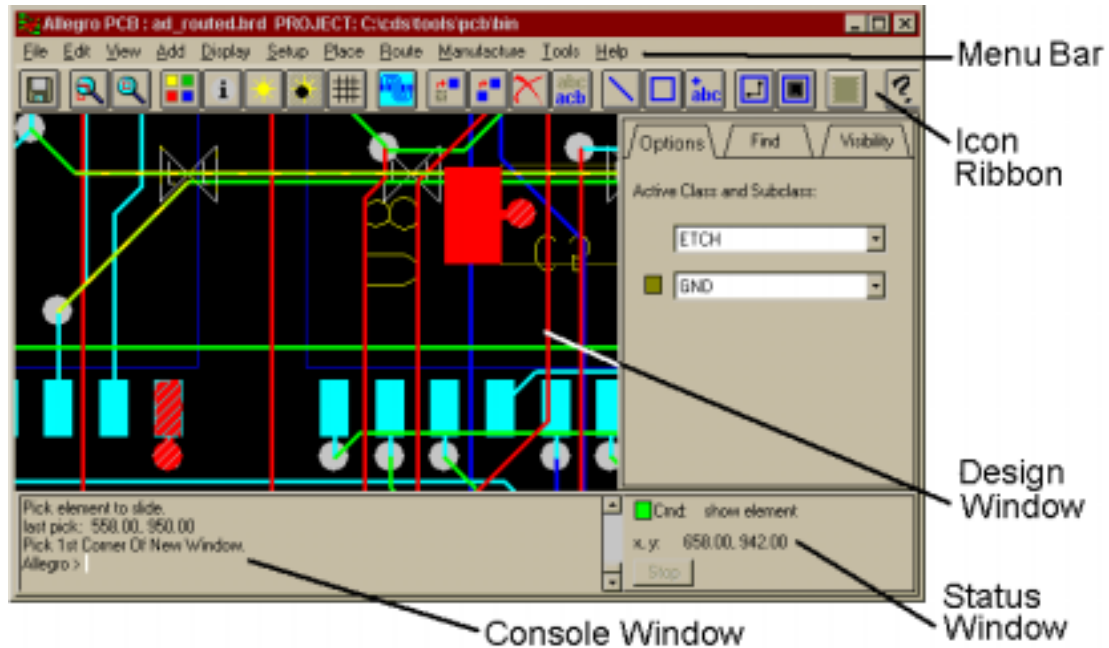


图 5.1

主要包括了菜单栏、图标栏、设计窗口、命令窗口、状态窗口和控制面板。

其中，控制面板包括三个可选按钮即：Option、Find 及 Visibility，还有一个全局视窗，位于下方。为了提供更大的设计空间，Allegro 允许你将控制面板移到屏幕的任何地方。

选择 View->Customization,如下： .

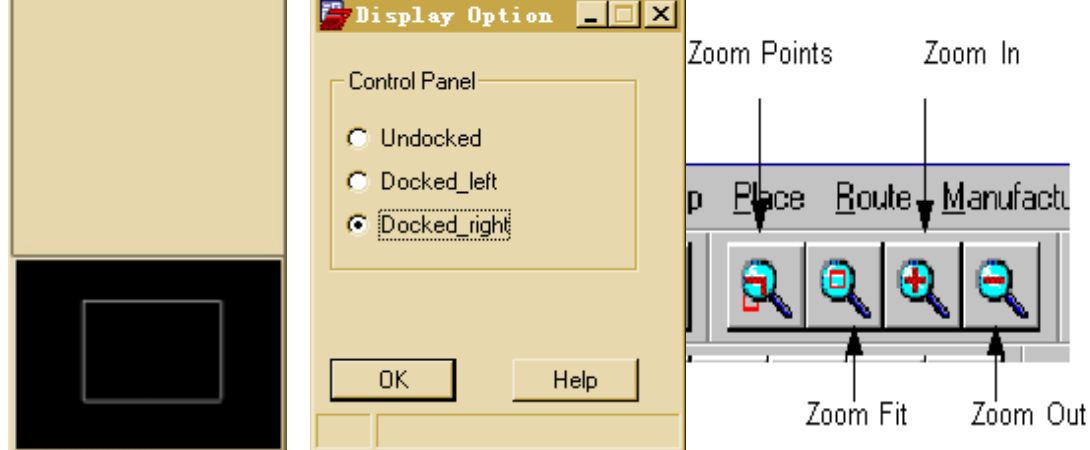


图 5.2

你可以通过 ZOOM 菜单放大或缩小设计的局部或整体，也可以用 PAN 命令来拖拉设计（Shift+右键）。

在 ALLEGRO 数据库中，所有元素都有一个类属性(CLASS)或子类属性(SUBCLASS)。例如，在 TOP 层的一条连接线，它的类属性就是 Etch，子类属

性就是 top。Etch 类在电路板每一层都有子类属性。相似的类构成组(GROUP)例如，Etch、Pin、Via、DRC 等都属于 Stack-up 组。

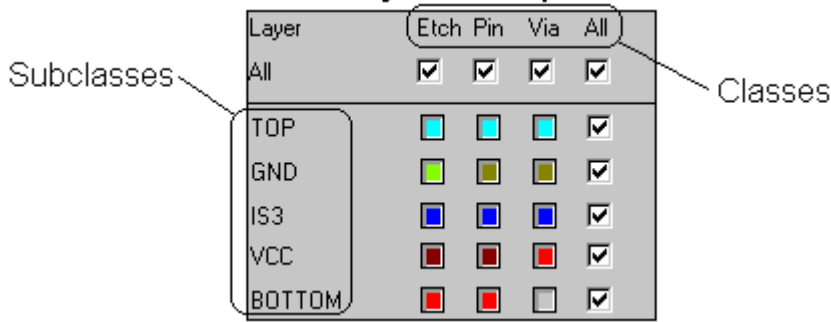


图 5.3

通过控制面板的 Visibility 标签，可以改变 Etch、Pin、Via 和 DRC 的各个子

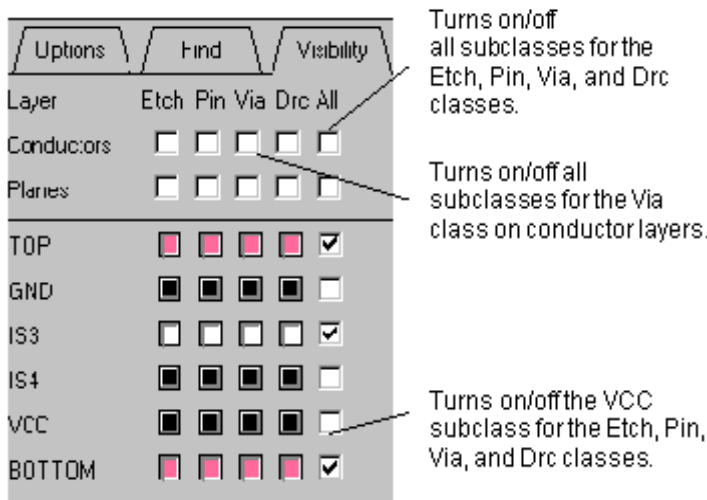



图 5.4

你还可以选择 Display->Color/Visibility 或单击图标  来确定所有组中各个类和各子类的颜色。

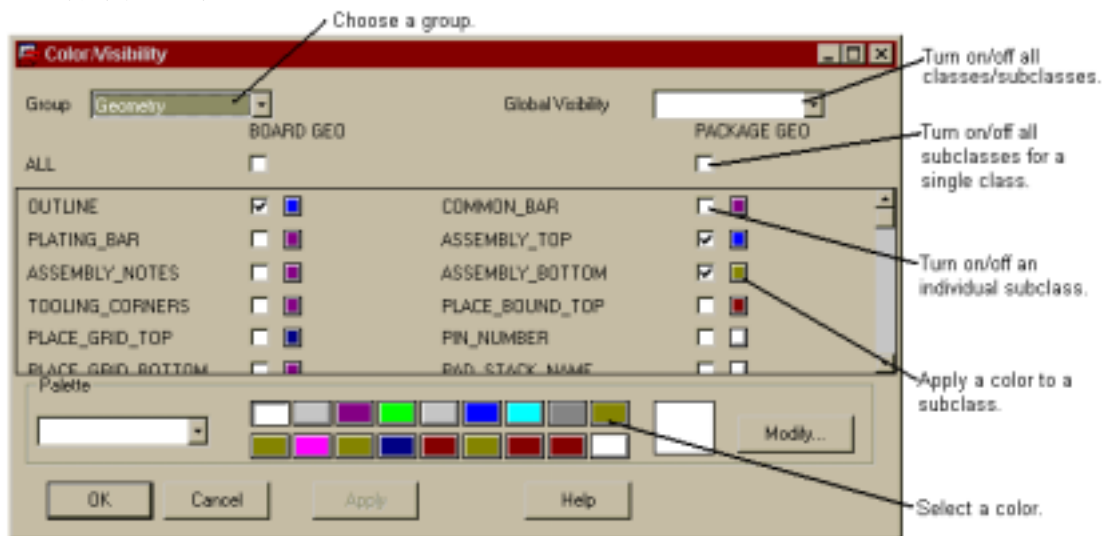




图 5.5

如果要查看某个元件的信息，可以通过 Display->Element,或单击图标。

如果要打开、新建或保存文件，可以选择图标。

如果需要帮助，可以选择图标。

## § 5.2 ALLEGRO 设计流程

### 5.2.1 准备工作

1. 开发所需元件库
2. 创建所需焊盘;
3. 创建所需的符号(封装、机械、规格或多边形填充)。

### 5.2.2 导入网表文件

1. concept—HDL, 或第三方逻辑;
2. 导入第三方设计文件。

### 5.2.3 布局准备

1. 创建有个新的图纸;
2. 定义图纸的大小; .
3. 定义pcb的物理层;
4. 制定各种属性;
5. 定义各种规则。

### 5.2.4 PCB设计

1. 定义布局;
2. 交换管脚和门;
3. 添加电源和地平面;
4. 布线。

### 5.2.5 同步及设计检查

1. 运行testprep程序;
2. 运行glossing程序;
3. 重新命名参考标识(ref designator);
4. 运行DRC检验;
5. 创建丝印层;
6. 反标CONCEPT-HDL逻辑图。

### 5.2.6 设计输出

1. 生成各种报告;
2. 生成组装文件;
3. 生成打印文件;
4. 生成光绘文件;
5. 生成第三方设计数据。

下面，就简要介绍以上六步的具体实现方法。

## § 5.3 设计流程的具体实现

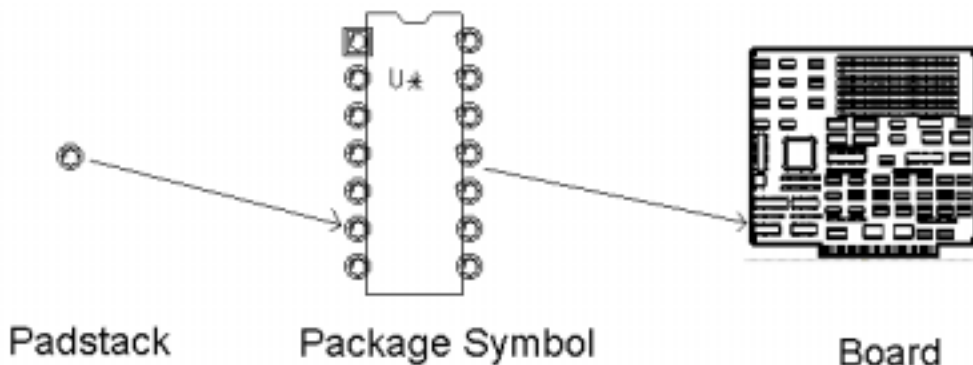
### 5.3.1 开发所需元件库

假设在你的设计中，需要DIP14这种封装，但是库中没有，就可以按以下方

法来创建。

## 1. 设计（60mil，38mil）的圆焊盘。

DIP14封装及其焊盘如下图所示：



通过焊盘编辑器可以创建所需的焊盘（Padstack）。

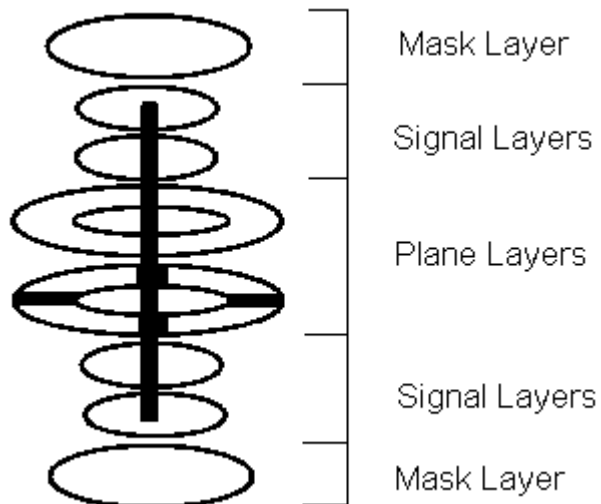
启动方法：NT：程序—>Cadence PSD14.0->allegro utilities->padstack editor

Unix: pad\_designer

创建如右图所示的焊盘。具体方法可以参考相应的帮助文件。

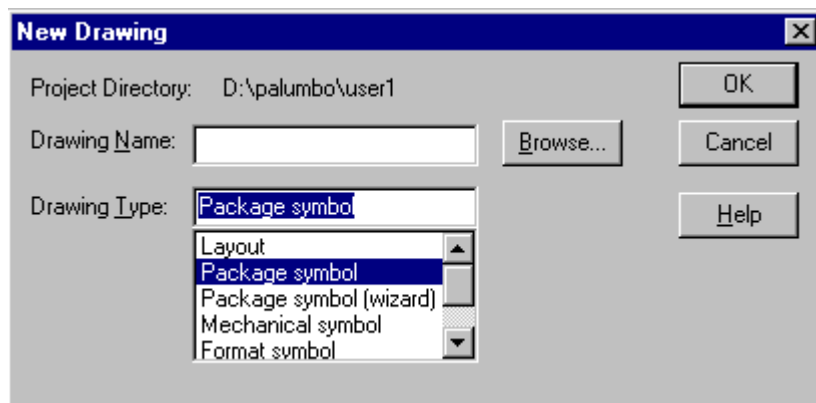
**Thermal Relief Pad:** 热漂移焊盘。用于于铜皮相连（如电源层或地层）的管脚。

**Anti Pad:** 阻焊焊盘。用于通过铜皮区域，但不与之连接的管脚。  
注意：保存焊盘时，要保存到设计的physical目录下，否则就要修改env文件设置焊盘所在的目录。出于项目的完整性考虑，还是放到physical目录下合适。



## 2. 建立封装符号

利用Allegro可以创建封装符号。从File—new，选择Physical Symbol。



先添加管脚；再添加封装边界（Package Geometry/Assembly top）；然后添加丝印边界和参考标识；最后保存结果。

注意：一定要保存两种格式，即\*.dra 和\*.psm。



or

Layout->Pins

### Options Window

Options			Find			Visibility		
<input checked="" type="radio"/> Connect			<input type="radio"/> Mechanical					
Padstack:			via					
Copy Mode:			Rectangular					
	Qty	Spacing	Order					
x:	1	100.00	Right					
y:	1	100.00	Down					
Rotation:			0.000					
Pin It:			1			Inc: 1		

### Add Pins

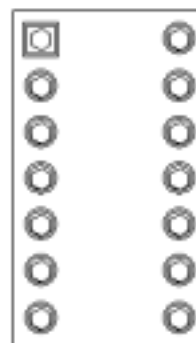


### Options Window

Options			Find			Visibility		
Active Class and Subclass:								
			PACKAGE GEOMETRY					
			ASSEMBLY_TOP					
Line Lock:			Line			45		
Line Width:			0.00					
Line Font:			Solid					

Add->Line

### Add Package Geometry

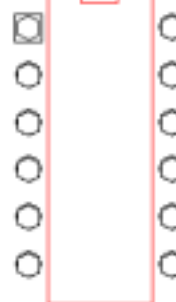


### Set Options Window

Options			Find			Visibility		
Active Class and Subclass:								
			PACKAGE GEOMETRY					
			SILKSCREEN_TOP					
Line Lock:			Line			45		
Line Width:			10					
Line Font:			Solid					

Add->Line

### Add Silkscreen Definition



### Set Options Window

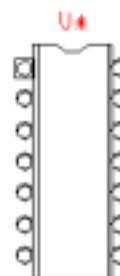
Options			Find			Visibility		
Active Class and Subclass:								
			REF DES					
			ASSEMBLY_TOP					
<input type="checkbox"/> Mirror								
Marker Size:			50.00					
Rotate:			0.00					
Text Block:			3					
Text Just:			Left					



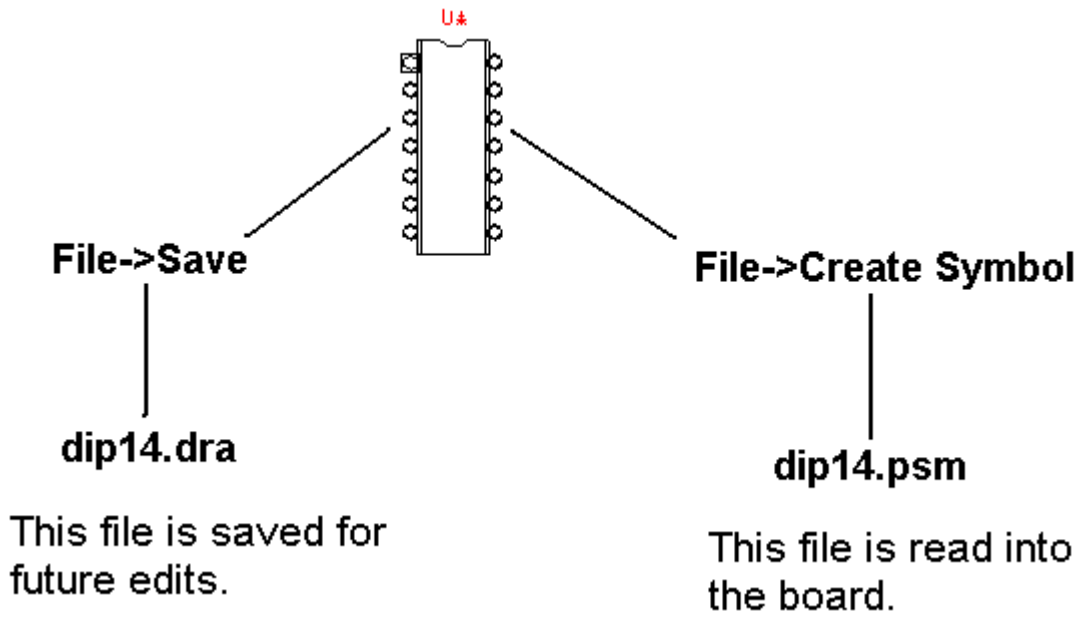
or

Layout->Labels->RefDes

### Add RefDes

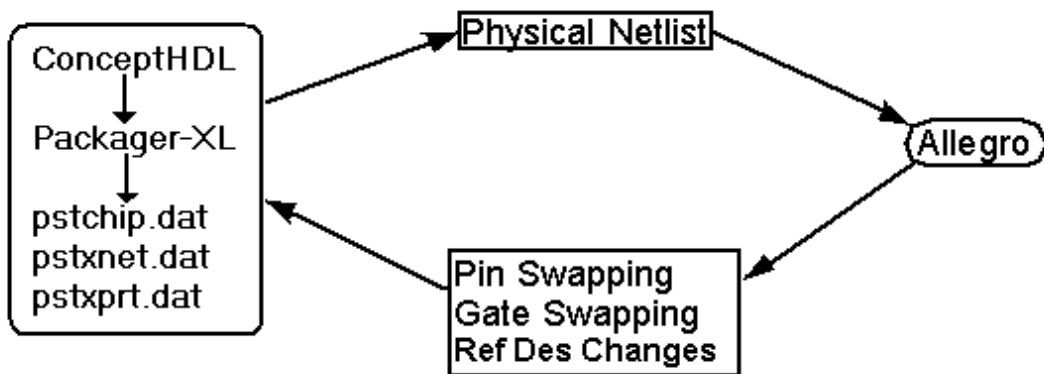




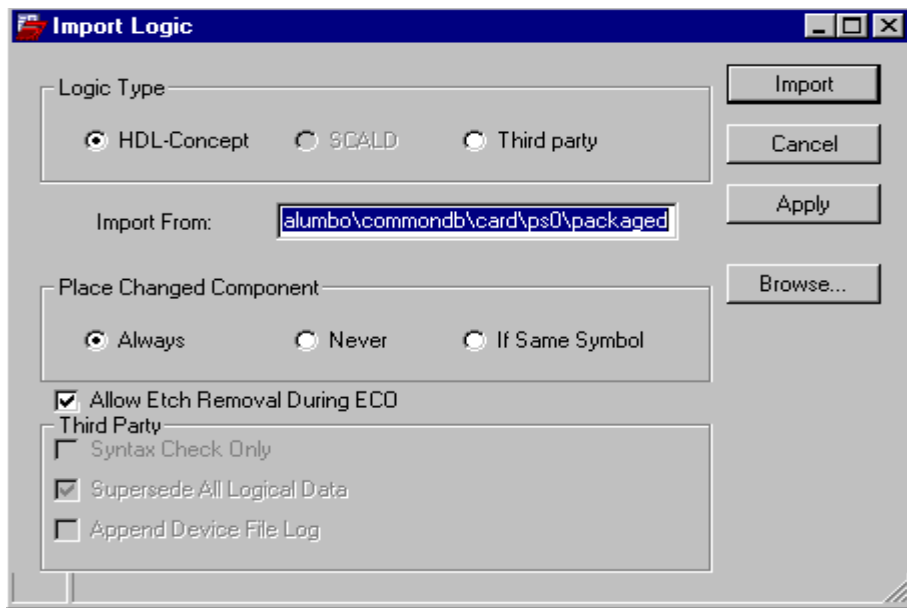


### 5.3.2 导入网表文件

所需文件由Package-XL生成，由Design Sync中的Export Physical导入。

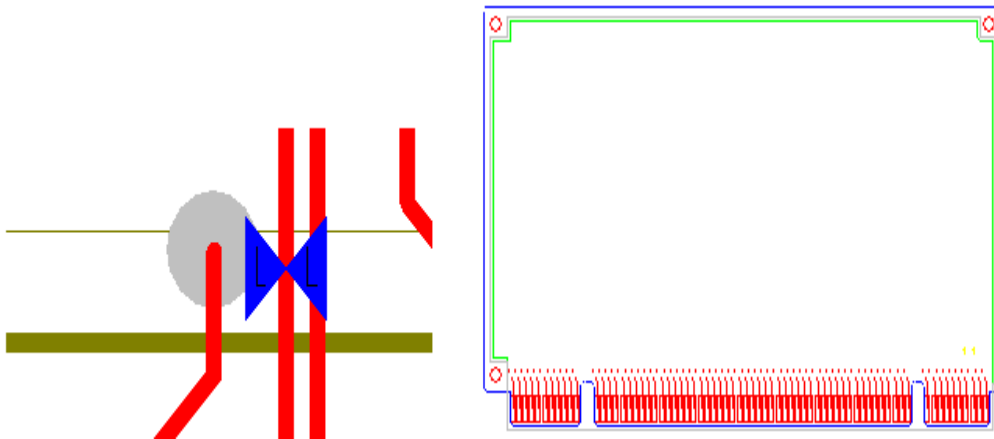


或者从File—import—logical来导入。

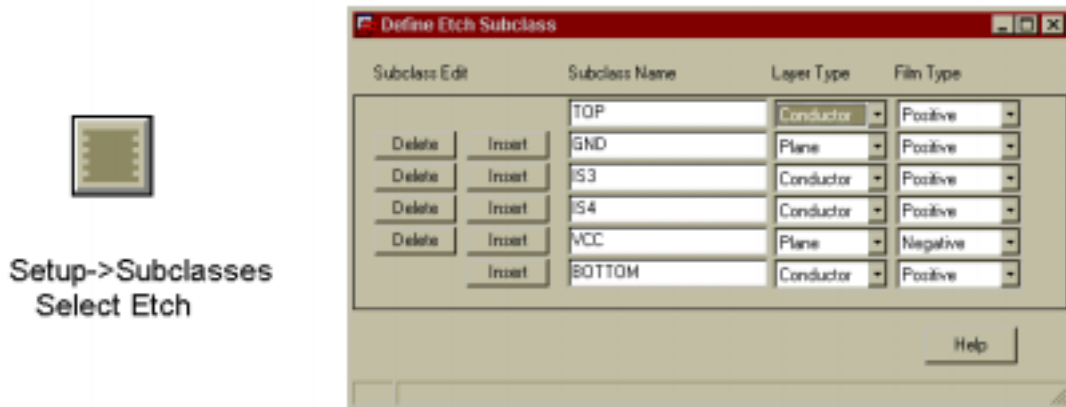


### 5.3.3 设置设计规则

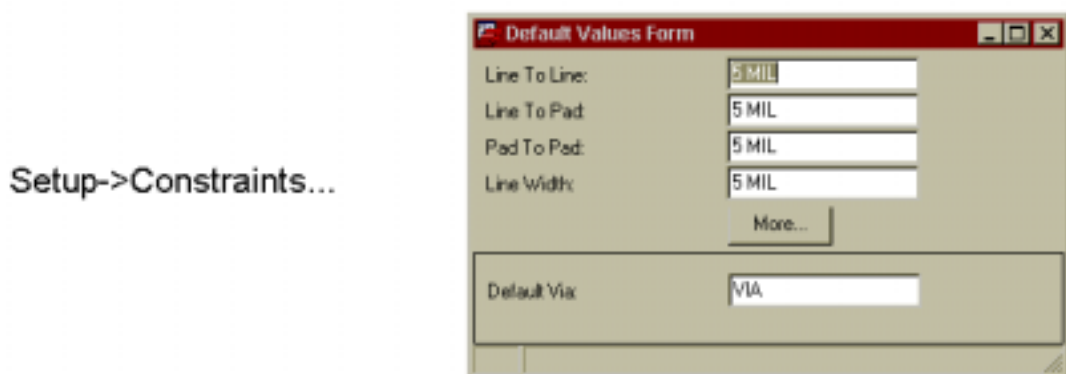
Allegro 允许你为设计设置详细的规则，用于在线的 DRC 检验或驱动自动布线器 Spectra。如果某处的网路违反了规则，会出现 DRC 符号，如下：



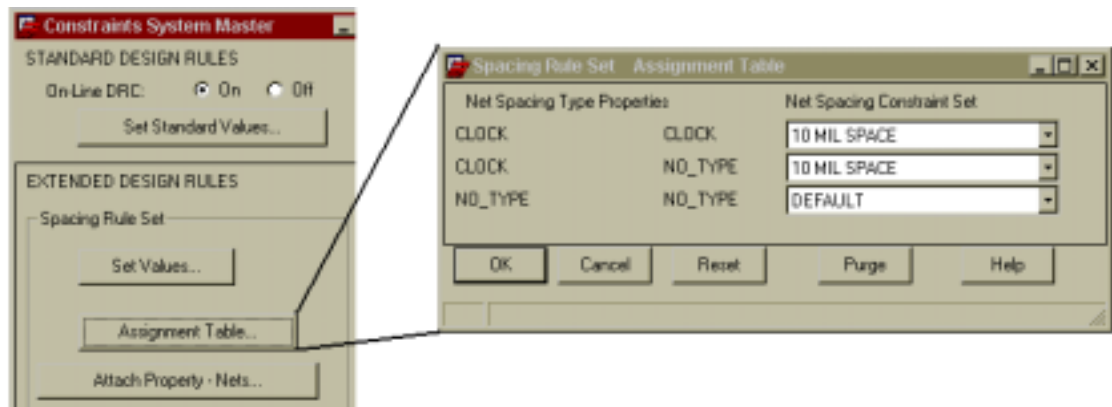
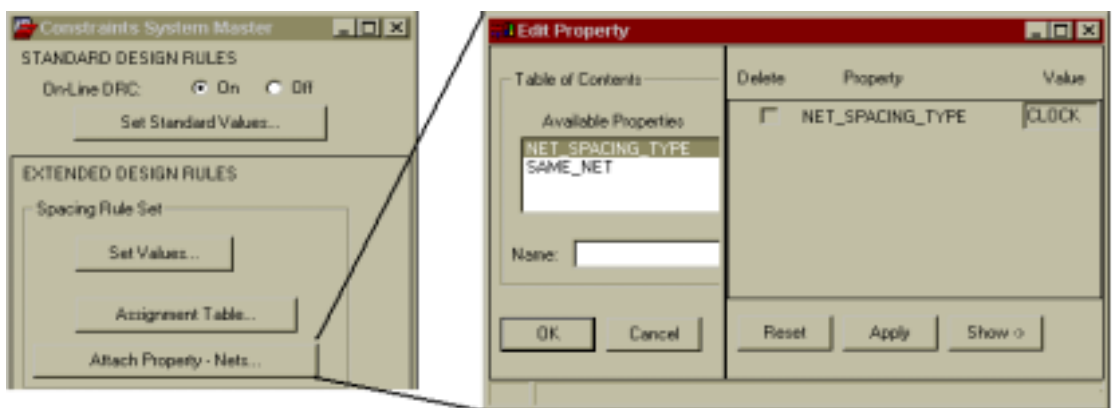
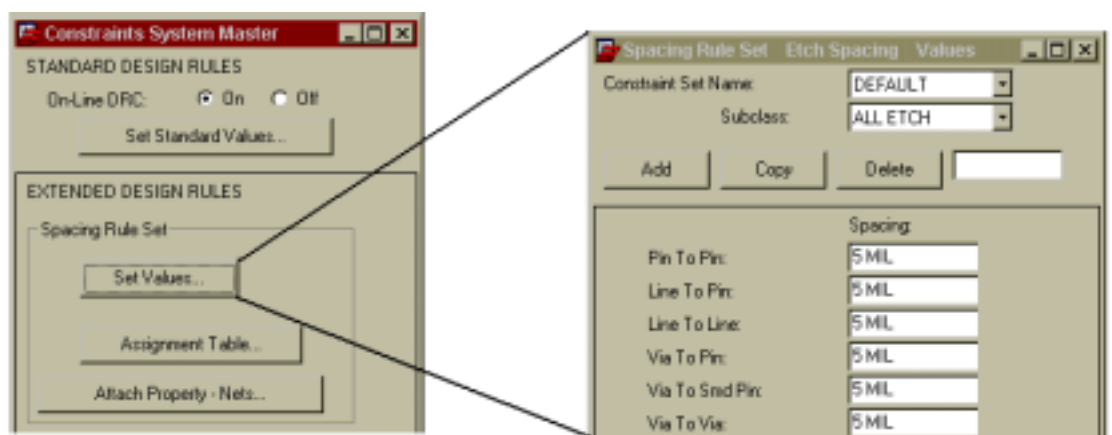
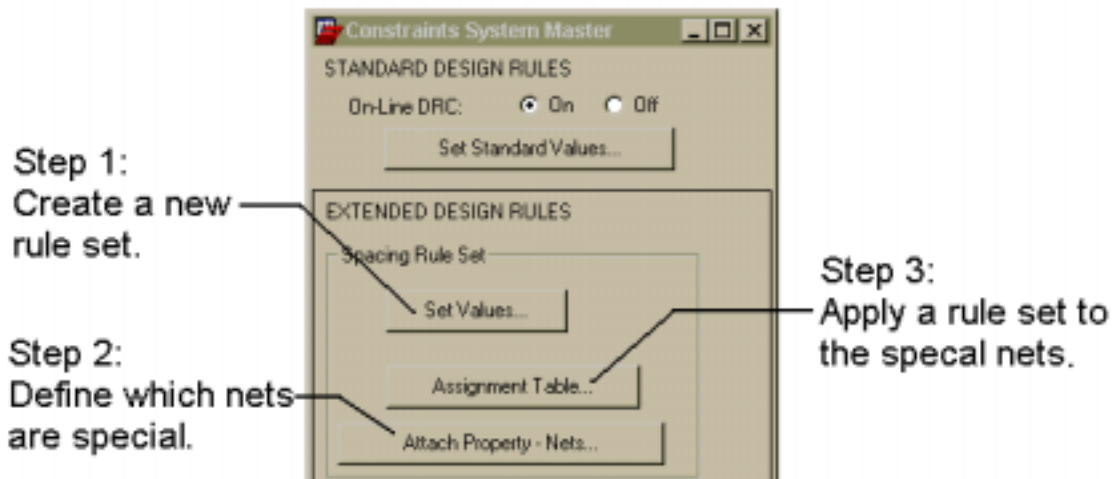
首先载入已经创建好的 PCB 的物理外形，创建方法同封装符号。然后，设置 PCB 的各层的材料、厚度等。



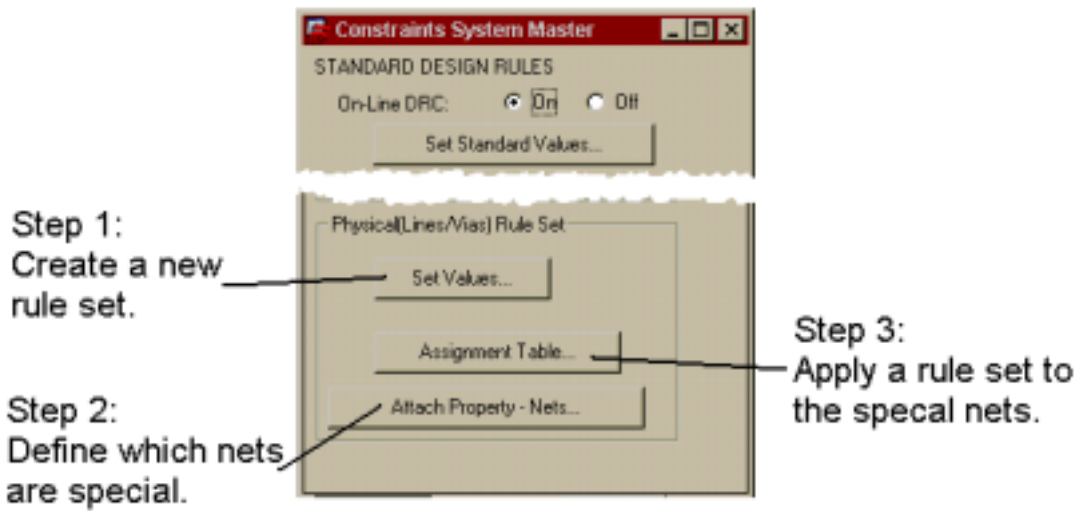
对于简单的设计，只要设置标准规则就可以了。



如果你的设计中某些网络或区域需要特殊的距离规则，就可以选择扩展的距离规则。例如，板上的线间距离要求为 5mil，而时钟信号要求与其他信号线距离至少为 10mil。扩展的距离规则设置方法和步骤如下：



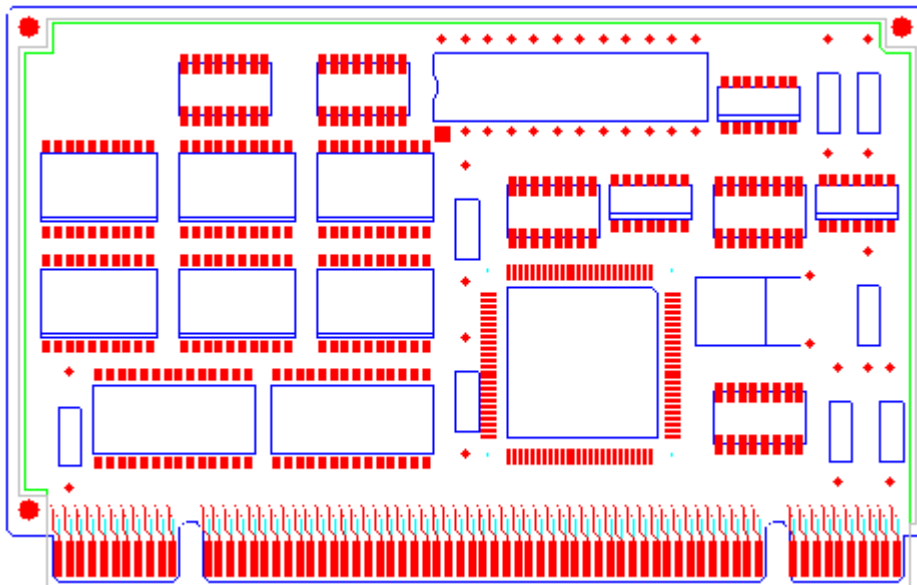
如果你的设计中某些网络或区域需要特殊的物理规则，就可以选择扩展的物理规则。例如，板上的线宽要求为 8mil，而时钟信号要求至少为 10mil。扩展的物理规则设置方法与扩展的距离规则设置相似。



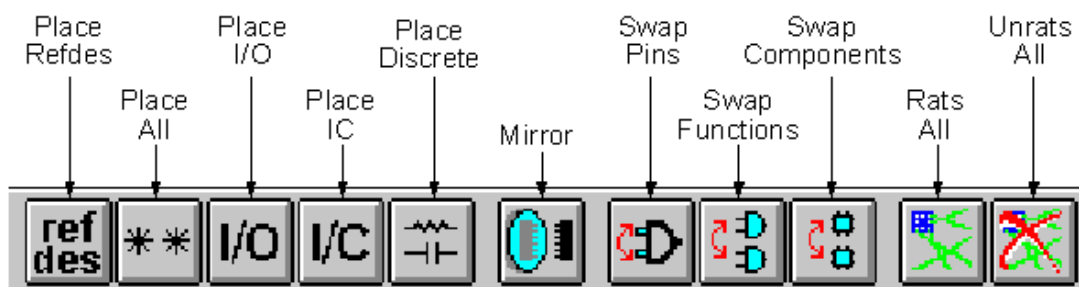
### 5.3.4 PCB设计

#### 1. 元件的布局

在准备好所需要的封装，导入了网表，设置好规则之后，就可以进行布局了。



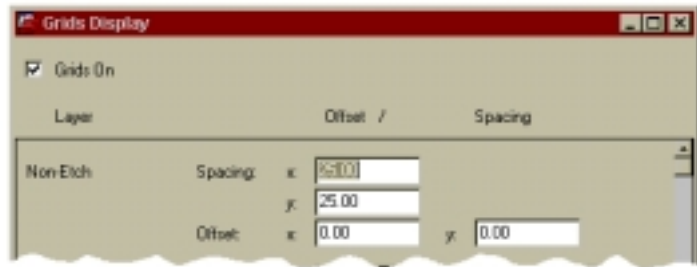
交互式布局所需的工具都可以在图标栏中找到。



在布局过程中，所有的元件都会锁定到最近的栅格。因此，布局之前要设置栅格

大小。一般设为 50mil，对于密度较大，表贴元件较多的板面通常设为 25mil。

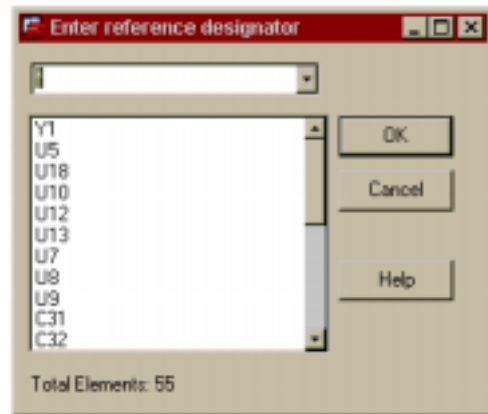
Setup->Grids...



摆放元件时，可以按参考标识从未布局的元件中选择

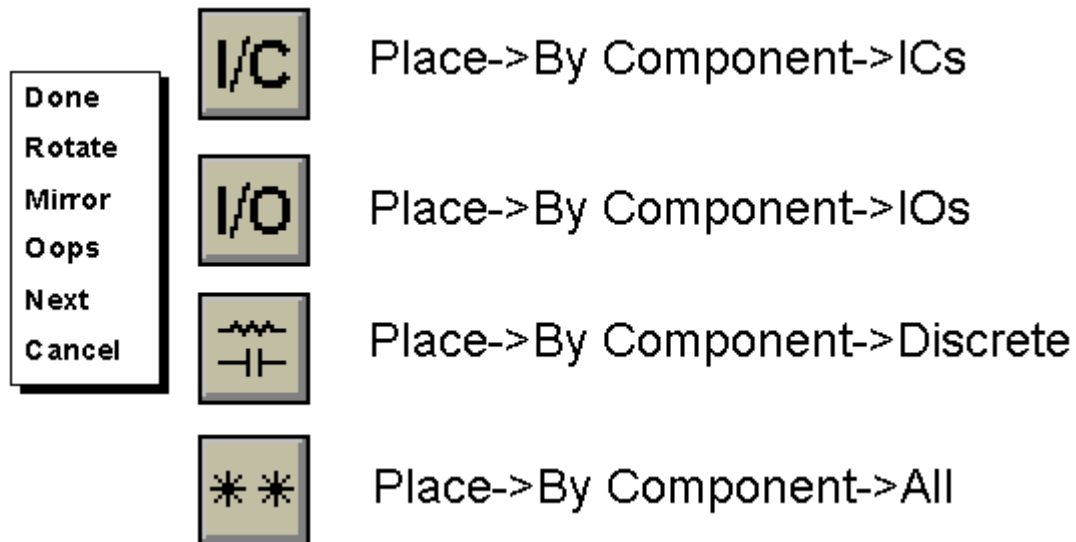


or



### Place->By RefDes

选择一个元件之后，元件会跟着鼠标移动，按右键，出现弹出菜单，可以结束操作、旋转、放到反面、返回上次操作、选择下一个元件或取消操作。



摆放元件时，可以按器件种类从未布局的元件中选择。器件种类包括：集成电路、I/O 端口、离散元件和所有元件。

预布线 (ratsnest) 可以让你在布局时对线长有个估计。这些预布线可以按你的意

愿显示或隐藏。



Display->Show Rats->All

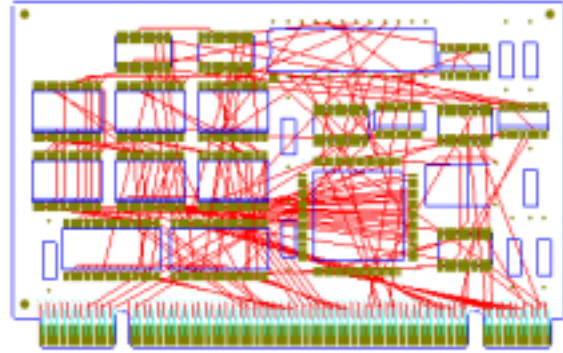
Display->Show Rats->Component

Display->Show Rats->Net

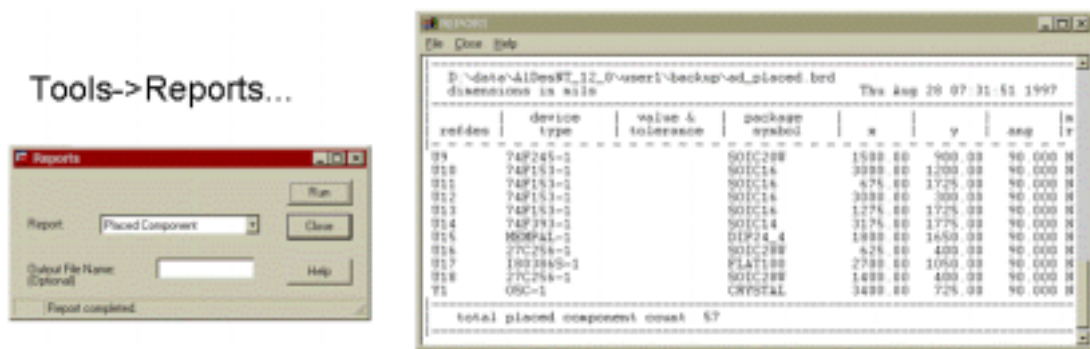
Display->Blank Rats->All

Display->Blank Rats->Component

Display->Blank Rats->Net



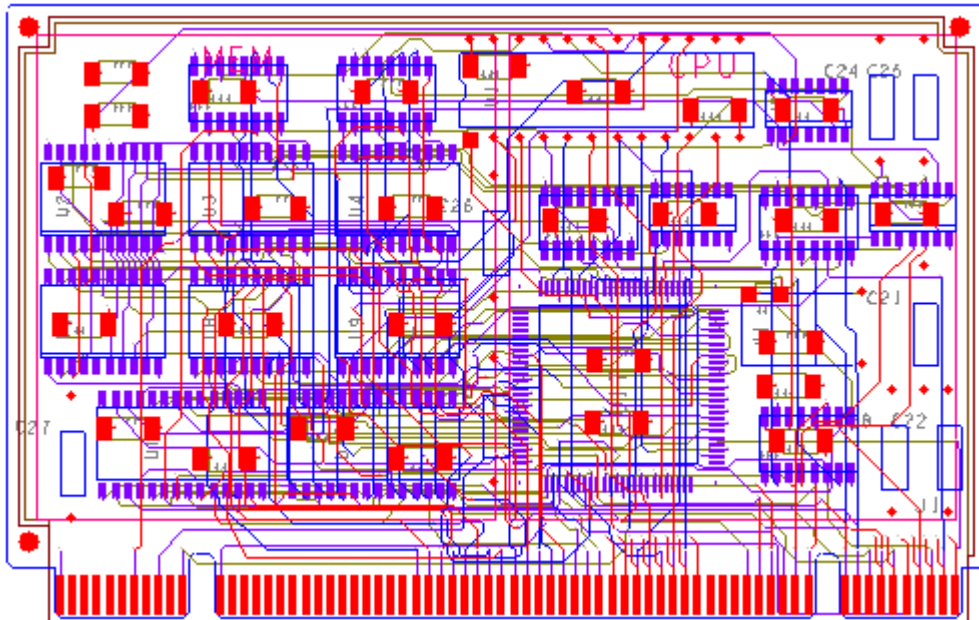
在布局完成之后，可以生成关于已布元件或未布元件的报告。特别是未布元件的报告，可以帮你检查是否存在未布的元件。



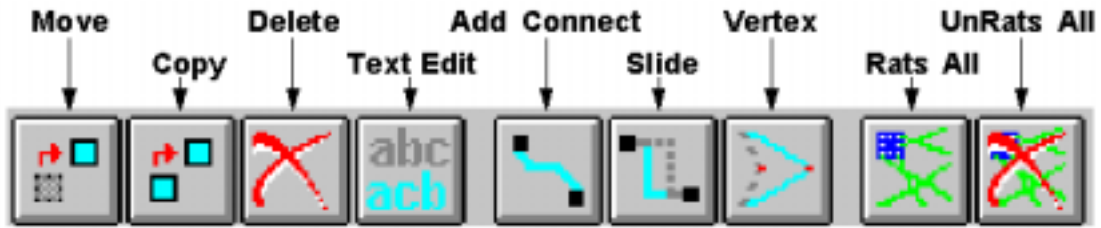
在布局过程中，你也可以采用自动布局和扩展布局功能。具体方法可以参考帮助文档。

## 2. 布线

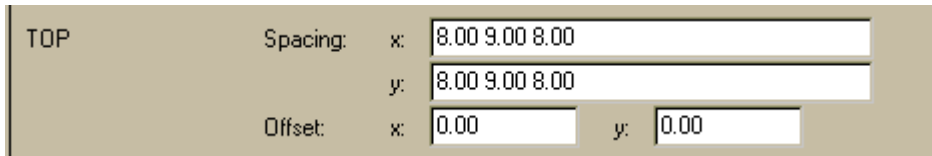
现在，可以对电路板进行布线了。布线后，得到



布线所需的工具可以从菜单栏或图标栏中选择



在设置布线栅格时，你可以设置一组变化的值，这样可以优化布线，得到最大的布线通道数。



如表中设置

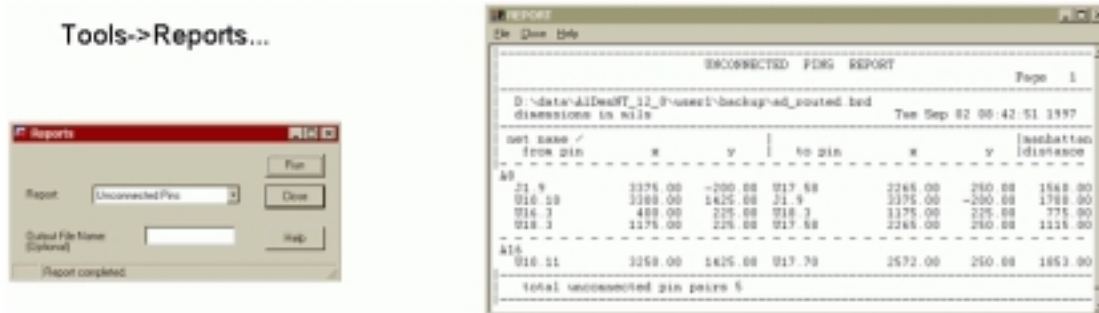
8, 9, 8, 则布线栅格变化规律为 0, 8, 17, 25。

布线过程可以自动进行也可以交互式进行。Allegro 中可以调用 Spectra 自动布线器来布线。



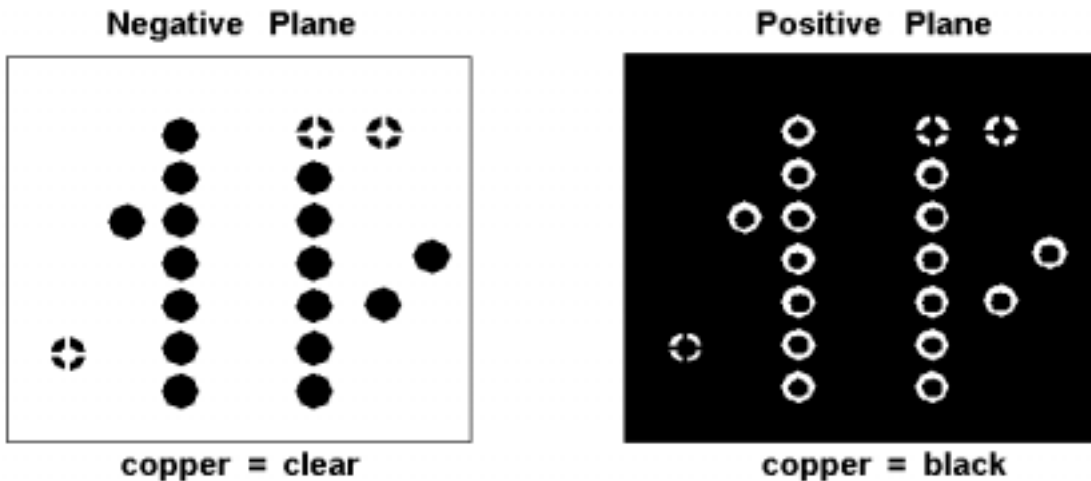
关于 Spectra 的使用方法，在下一章会详细讲解的。

最后，你可以生成一份关于未连接线的报告，用于检查。



### 覆铜处理

Allegro 的覆铜处理技术可以分为两类，即：阴板覆铜和阳极覆铜处理。



## 两者各有优、缺点。

### 阴板覆铜：

优点：生成光绘数据量小，能自适应动态的布局、布线修改。

缺点：所有的焊盘必须具有 flash 名，以便能将 Gerber 文件载到 Allegro 后，查看结果。

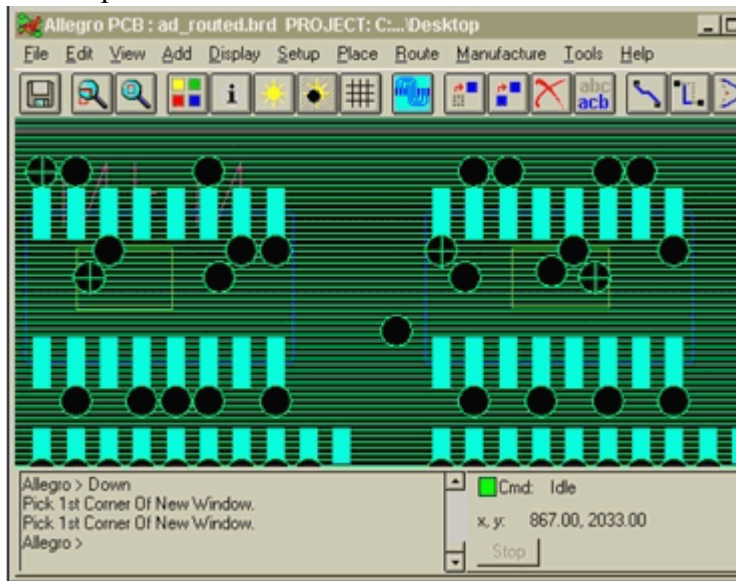
### 阳板覆铜：

优点：直观，所见即所得，焊盘无须 flash 名。

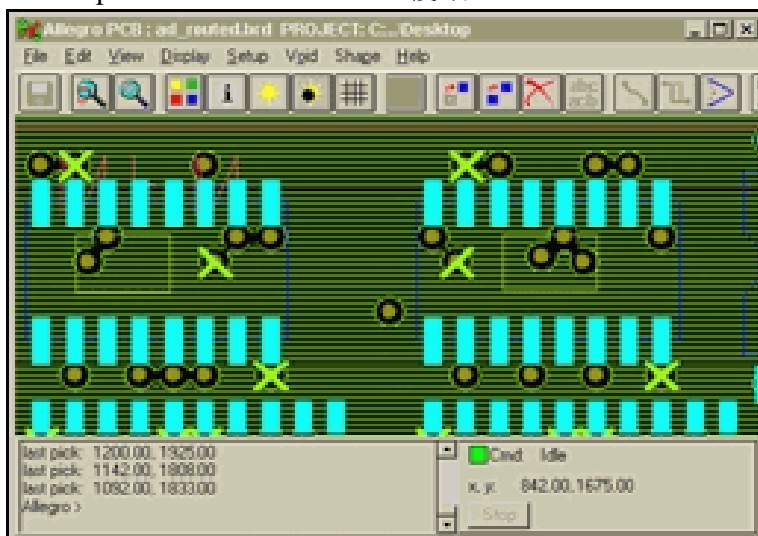
缺点：光绘文件数据量大，不能自适应动态的布局、布线修改。

生成阴板覆铜的步骤：

- 1) Add->Shapes->Solid Fill: 画出要覆铜的边界；
- 2) Edit->Change Net(Pick): 为覆铜区选择一个网络；
- 3) Shape->Fill: 覆铜。

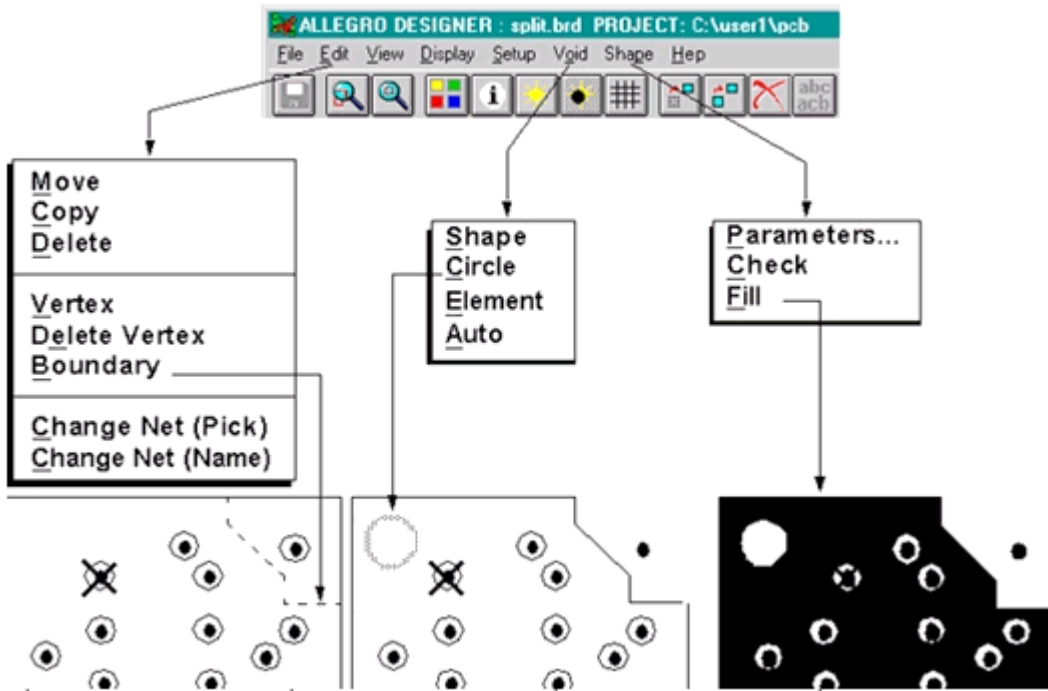


1. Add->Shapes->Solid Fill: 画出要覆铜的边界；
2. Edit->Change Net(Pick) : 为覆铜区选择一个网络；
3. Shape->Parameters : 选择填充方式；
4. Void->Auto : 选择避让方式；
5. Shape->Fill : 覆铜。





在你添加了一块覆铜区域之后，还可以对它编辑，如修改边界、在阳板上做特殊避让等等。编辑之前要取消填充，编辑之后一定要恢复填充。

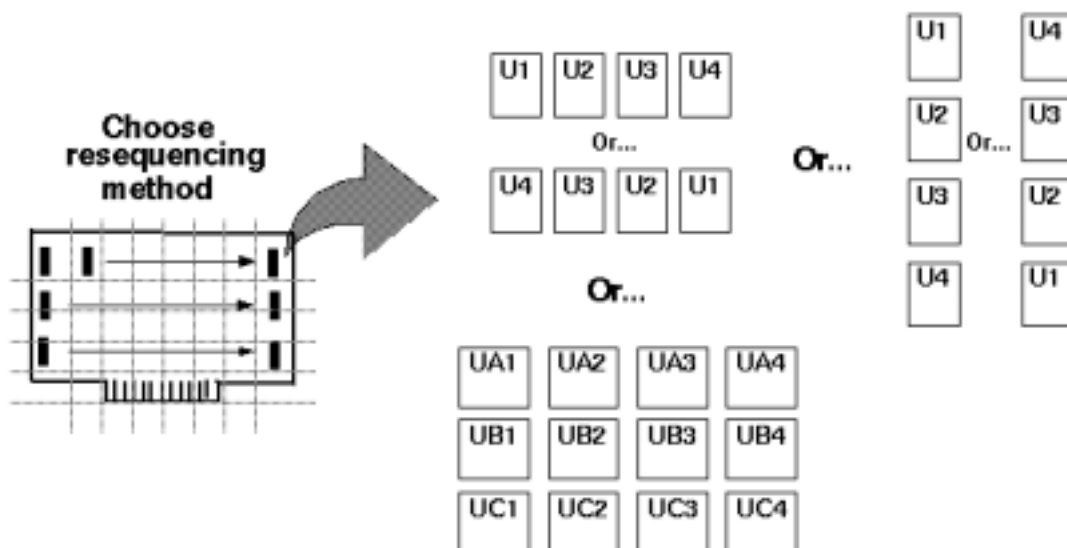


### 5.3.5 制造输出和文档生成

这是 PCB 流程的最后一个环节，主要讲述对 PCB 板图的后处理，以便加工制造。这道工序可以分以下步骤进行：

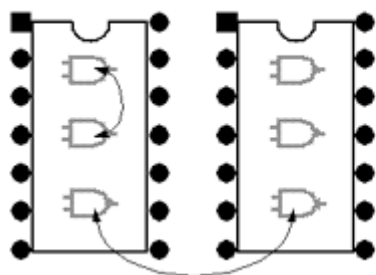
- 重新定义元件位号；
- 反标；
- 绘图；
- 生成光绘文件；
- 生成数控钻孔文件；
- 生成文档。

1. PCB 设计完成之后，也许要对元件位号重新编号，以便辨认装配和调试。这项工作可以手工完成，也可以自动生成。利用 Logic-Auto Rename 命令，选择位号的编写风格和需要编号的元件，点击 OK 执行。重新编号之后，一定要反标会原理图 Congcep-HDL。

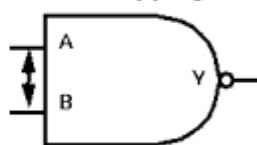


2. 反标。File->Export Logic

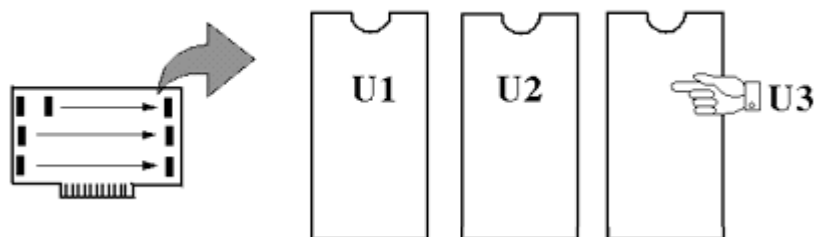
**Gate Swapping**



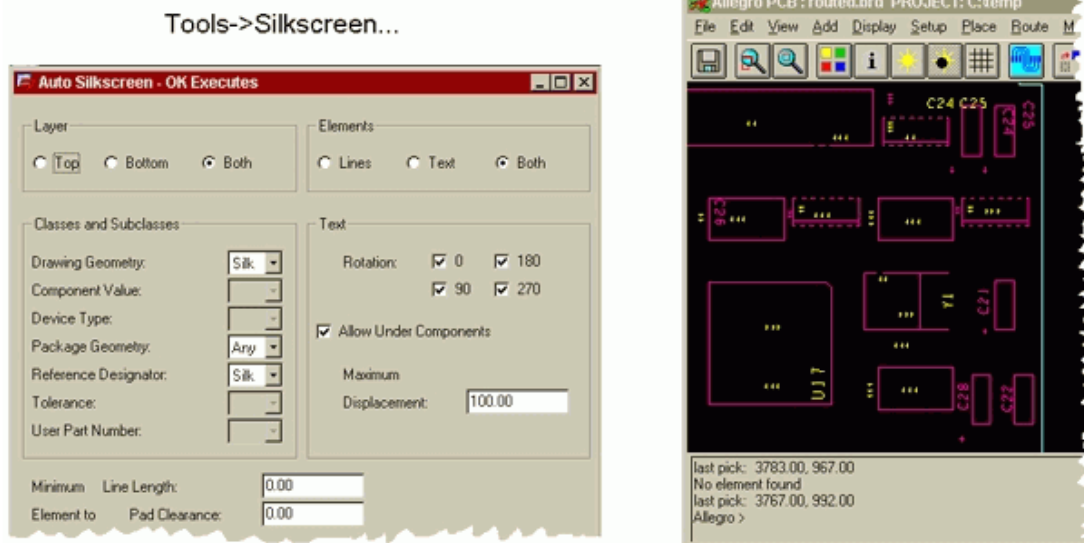
**Pin Swapping**



**Reference Designator Rename**



3. 生成丝印面。



4. 绘图。File->plot 可以将 PCB 输出到打印机或绘图仪。

5. 生成光绘文件。

需要的输入文件

-Film Control Records : 胶片定义包含在设计文件中(\*.brd);

-Parameter file(art-param.txt): 参数设置文件;

-Aperture file(art-aper.txt) : D 码表。

输出文件

-Log File(photoplot.log): 日志文件;

-<file>.art : 胶片 GERBER 文件。



生成参数设置文件。

Manufacturing->Aartwork->General Parameter

Device Type: Gerber 4x00

Max Aperture Per Wheel: 99

File->save

生成光圈文件(art-aper.txt):

Manufacturing->Aartwork->Apertures

点击 Auto, 选择 Without Rotation 或 With Rotation。

点击 OK。

生成胶片定义:

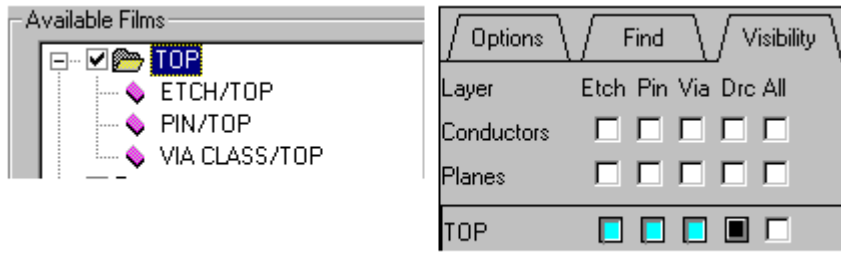
Manufacturing->Aartwork->Films;

对每张胶片都可以编辑, 选择需要的图形元素。

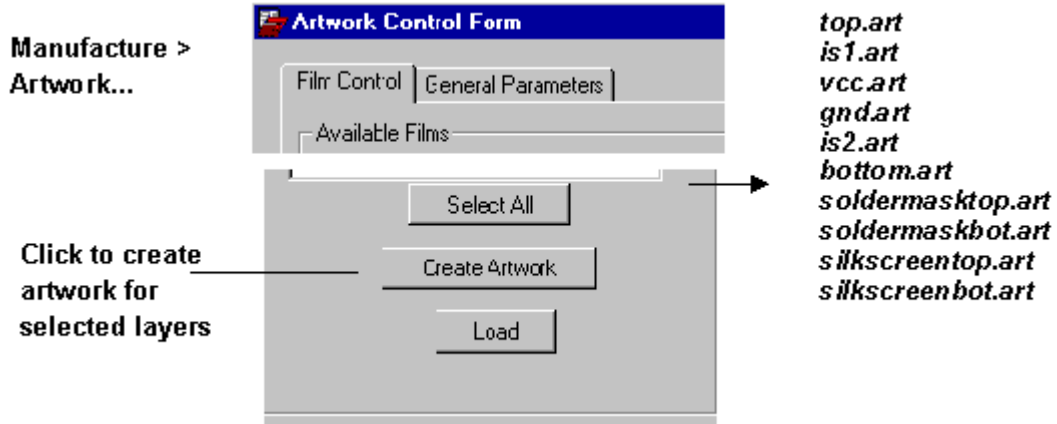
在编辑表格中, Plot Mode 选项对信号层选 Positive, 对电源和地层选 Negative。

将 Undefined Line Width 设置为 D 码表中的一个合适的线宽如 8mil, 12mil 等。

选中 Draw Missing Pad Apertures。

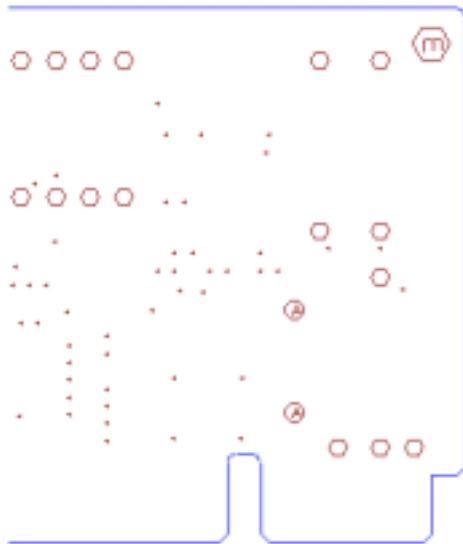


生成光绘文件。



### 6. 生成钻孔文件 建立钻孔表单

Manufacture—NC-Drill Legend。

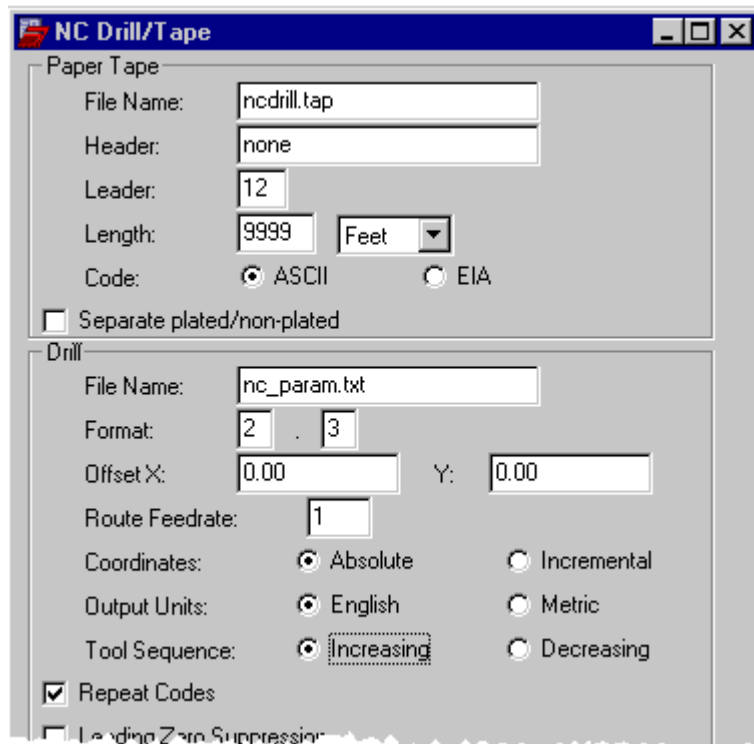


DRILL CHART

FIGURE	SIZE	PLATED	QTY
.	16.0	PLATED	481
o	36.0	PLATED	1
o	36.0	PLATED	39
o	38.0	PLATED	2
m	109.0	NOT PLATED	3

生成钻孔参数文件(nc\_param.txt)

Manufacture—NC-Drill Parameters。



#### 7. 生成报告文件。

Allegro 的文档工具可以生成各种报告文件，如：元件清单、布通率、DRC、无连接引脚等等。你可以选择 Tools—Report 命令来生成需要的报告。

到此为止，一个完整的 PCB 设计就全部结束了。



