

# DVE 用户指导手册

本文档只供交流和学习使用，欢迎您指出本文档中的不足和提

# 供您宝贵的建议！！

1 入门.....	3
基本要求：.....	3
启用调试.....	3
编译时间选项.....	3
必要文件.....	3
调用 DVE.....	3
64 位技术支持平台.....	3
后台处理程序：.....	3
交互式界面.....	3
脚本命令（Scripts）：.....	3
DVE 的日志文件.....	3
在命令行运行仿真.....	3
DVE 与 VCS, SystemVerilog, 和 NTB (OV 和 SV).....	3
VCS MX 和 VHDL.....	3
检查点还原的方法.....	3
C, C++, 和 SystemC 代码.....	3
在 GUI 界面运行仿真.....	3
后台处理程序（Post-processing）.....	3
创建并且运行一个交互式的会话.....	3
运行仿真.....	3
工具栏的应用.....	3
模拟器（Simulator）菜单命令的应用.....	3
命令行的应用.....	3
保存会话或布局图.....	3
保存一个布局图会话.....	3
载入已保存的会话（session）.....	3
关闭数据库.....	3
退出 DVE.....	3
2 图形化界面应用.....	3
DVE 窗口属性概述.....	3
DVE 窗格（panes）.....	3
管理 DVE 窗口.....	3
管理目标窗格.....	3
停放与移除窗口和窗格.....	3
拖拽和删除一个已停放的窗口.....	3
菜单栏和工具栏的应用.....	3
菜单栏参考.....	3
文件菜单.....	3
编辑菜单.....	3
视图查看菜单.....	3
仿真菜单.....	3
信号菜单.....	3
作用域菜单.....	3
追踪菜单.....	3
窗口.....	3
信号.....	3
模拟器.....	3
时间操作.....	3
扩大或缩小.....	3
扩大或缩小和变动记录.....	3
上下文记录命令菜单的应用.....	3
设置显示的首选项（Preferences）.....	3
自定义显示.....	3
3 使用层次数据和信号群组窗格.....	3
层次窗格.....	3
给信号赋值.....	3

4 使用源窗口.....	3
管理交互仿真中的断点.....	3
控制源窗口中的行断点.....	3
在对话框中管理断点.....	3
行断点.....	3
时间断点.....	3
信号断点.....	3
断言断点.....	3
任务/函数断点.....	3
编辑断点.....	3
5 运用波形窗口.....	3
建立目标窗口.....	3
设置波形窗口首选项.....	3
重命名信号组.....	3
过滤信号.....	3
添加信号分离器.....	3
设置复制信号的显示.....	3
创建一个用户定义基数.....	3
管理用户定义基数.....	3
导入和导出一个用户定义基数.....	3
运用波形窗.....	3
制定波形显示.....	3
指针和标记.....	3
使用指针.....	3
运用标记.....	3
移动, 隐藏和删除标记.....	3
扩展和收缩波形信号.....	3
在图像窗中搜索.....	3
比较信号, 域和组.....	3
建立总线并设置表达式.....	3
使用总线/表达式对话框.....	3
总线/表达式列表框.....	3
显示/隐藏细目表格.....	3
创建一个总线.....	3
修改总线成分.....	3
管理表达式列表.....	3
移位信号.....	3
8 断言与覆盖率.....	3
SV 断言的编译.....	3
断言结果的观察.....	3
显示标准的设置.....	3
断言的调试.....	3
观察波形窗口中断言的失效.....	3
观察断言失效时间增量.....	3
源代码的定位.....	3
9 动态驱动和载入的跟踪.....	3
跟踪功能.....	3
支持的功能.....	3
不支持的功能.....	3
Vera 的功能.....	3
跟踪驱动程序.....	3
跟踪载入.....	3
Vera 的功能.....	3



# 1 入门

本章描述了怎样开始使用 DVE

它涵盖以下内容:

- 基本要求
- 启用调试
- 调用DVE
- 在命令行运行仿真
- 在GUI界面运行仿真
- 保存会话或布局图
- 载入已保存的会话
- 关闭数据库
- 调用 DVE

## 基本要求:

您必须使用相同版本的 VCS 和 DVE, 以确保任意仿真中任意问题都能调试。

查询该 DVE 版本的三种方法:

- 在命令行选项中键入 **dve -v**。
- 在 DVE 的命令行输入 **gui\_get\_version** 指令。
- 使用 **Help>About** 帮助菜单选项。

## 启用调试

本节介绍如何调用调试选项来进行仿真。

注:

如果你在一个不能写入数据的文件夹中运行 DVE, DVE 将不能够记录日志文件。这种情况下 DVE 将警告您不能够写入 DVE 文件。

## 编译时间选项

### **-debug**

调用 UCLI, 并为 DVE 的基本调试建立必要的文件

### **-debug\_all**

调用 UCLI 和 DVE, 并为进一步的 DVE 调试建立必要的文档。它加入了单步调试、内存释放、和基于断言的调试。

### **-debug\_pp**

创建一个 VPD 文件 (当与 VCS 系统任务 `$vcdpluson` 一起使用的时候) 并且用 `-debug_pp` 命令来对 design 进行后台处理 (post-processing), 这样能够节省编译的时间因为它间接的消除了 `-debug` 和 `-debug_all` 的编译的预处理程序。这样它能够最低限度的进行调试以便进行 VPD 和基于断言的调试。这个效果在 DVE 的后台处理模式下是显而易见的。

## 必要文件

DVE 启动它的调试功能需要如下的输入文件:

- VPD 文件 - VPD 文件是平台独立的, 在仿真中你可以在这个根目录下抽取选定的信号。DVE 在这些文件里可以得到层次化的、改变值的、和一些断言的信息。基本的调试只用一个 VPD 文件就能在后台(post-processing)进行。然而, VPD 文件不能保证设计的所有层次都包涵在内, 因为用户可以有选择地选择将设计的一些子模块存放在 VPD 文件中。VPD 文件是平台独立的。VPD 文件是有版本要求的。如果您使用相同版本的 VCS 和 DVE 来读写 VPD 文件时是没有问题的。然而, 如果你使用 DVE 从一个比它版本老的 VCS 中读取 VPD 文件时, 它就不能保证所有的 DVE 的功能都能使用或执行正确。

注: 为了保障 DVE 调试的功能性, 用一个编译时间选项 (**-debug,-debug\_pp,** 和**-debug\_all**) .不推荐使用 VCD 文件并且也不允许调试。

- **MDB 库文件。**DVE 用 MDB 库来保证信息的连通性。DVE 需要 MDB 信息为了原理图和所有的驱动和负载有关的调试。MDB 库文件将被存放在运行模拟的文件夹内。MDB 文件是依附于平台的 (因此为了应用 DVE 必须和 MDB 库在同一平台上运行)。更严格地说, 为了确保操作的正确性, VPD 文件和 MDB 库应该都由相同的仿真产生。
- **OVA 库。**DVE 用这个库是为了进行更高级的断言调试。当 VCS 的编译选项都用的正确并且一个设计中包含了 OVA/SVA/PSL 的断言方法时才能生成这个库。这个库也是依附于平台的。
- **覆盖率数据库 (coverage databases)** .在 DVE 中你需要指定 3 种类型的 coverage databases 当中的一种来显示覆盖信息。如果其他类型的覆盖也存在, DVE 也会将他们打开。

你可以选择下面 2 种类型的数据库中的任意一种:

- 一个代码覆盖率的文件夹 (被 VCS 或 VCS MX 在 Verilog 顶层设计中默认为 **simv.cm** 或者时被 VCS MX 在 VHDL 顶层设计中默认为 **scsim.cm**) 。
- 一个 Open Vera 或 SystemVerilog 断言库文件夹 (被 VCS 默认为 **simv.vdb**)

## 调用 DVE

这节描述了怎样调用 DVE。

### **dve -help**

显示 DVE 基本的指令

**dve -help all**

显示所有的 DVE 指令

**dve -v|-V**

显示版本信息

## 64 位技术支持平台

**-full64**

当你已经指定平台的时候，在运行时间输入就能够启动 64 位的 DVE 功能。为了达到 64 位的支持，输入下面指令：

**dve -full64**

可以输入下面指令中的一个来指定平台：

- linux RH3.0 64-bit:

**Sentenv VCS\_TARGET\_ARCH amd64**

- SuSE9 64-bit:

**Sentenv VCS\_TARGET\_ARCH suse64**

- Solaris 64-bit;

**Setenv VCS\_TARGET\_ARCH sparc64**

## 后台处理程序:

### **dve**

不带参数, 打开并且清空了一切 DVE 的顶层窗口。在这里 DVE 能够在后台或交互式的模式下进行使用。

### **dve -vpd filename**

打开 DVE 并且读入在命令行给出的 VPD 文件, 并为设计打开一个顶层视图。

### **dve -vpd filename -session filename**

打开 DVE 并且读入在命令行给出的 VPD 文件, 它们打开一个以前保存过的 TCL 文件的会话。

## 交互式界面

### **dve -nogui**

启动了 UCLI 模式进行调试。DVE 的 GUI 界面是不显示的。

### **sim -ucli**

运行 VCS/VCS MX 来帮助 UCLI 调试, DVE GUI 不能被显示出来。

### **simv -gui**

在 simv 仿真的 0 时刻打开 DVE。

### **vcs -gui -R**

与上面相同但是在编译时调用。

### **dve -toolexe name -toolargs simulator args**

运行一个与模拟器相连的 DVE, 这个模拟器的名叫 *name* 并且运行 DVE 的时候具有在模拟器中指定的 *args* 属性。

### **simv -gui -tbug**

调用集成平台进行调试。

## 脚本命令 (Scripts) :

### **dve -cmd "cmd"**

启动 DVE 并且执行括在双引号中的 TCL 指令。当需要执行多条指令的时候是允许用分隔号的。

### **dve -script name**

启动 DVE 并且读入由 name 所指定的 TCL 脚本文件。

### **dve -session name**

启动 DVE 并且读入一个会话文件 (session)。如果 **-session** 和 **-script** 选项一起使用的话, 先执行会话命令后执行脚本命令。

## DVE 的日志文件

调用 DVE 的时候会在文件夹里生成下面 2 个日志文件 (log files)。这些日志在有问题的情况下反馈给 Synopsys 公司是非常有用的。

- **dve\_gui.log** –包含所有通过控制台日志的输入和输出。
- **dve\_history.log** –包含了发生在调试会话周期中的所有指令。这个文件对记录脚本再运行是非常有帮助的。

# 在命令行运行仿真

## DVE 与 VCS, SystemVerilog, 和 NTB (OV 和 SV)

运行 DVE，你必须在编译时调用它。你可以用 **-debug**, **-debug\_all** 或者 **-debug\_pp** 属性来调用 DVE，或者将 DVE 设置成默认的命令界面。

一起运行 DVE 和 VCS，需要输入带有 DVE 使能的 VCS 命令行选项。

```
vcs (-debug | -debug_all | -debug_pp) [-sverilog] [-ntb] [VCS_options]  
design.v [testbench_files]  
simv -gui [runtime_options]
```

## VCS MX 和 VHDL

### 纯粹的 VHDL

运行一个带有 DVE 的 VHDL 仿真，需要输入一个带有 DVE 使能选项的 VCS MX 指令：

```
vhdlan design.vhd  
scs cfg_tb (-debug | -debug_all)  
scsim -gui [runtime_options]
```

## 一个拥有 Verilog 为顶层文件的混合仿真

运行一个 Verilog 为顶层文件的 Verilog/VHDL 的混合仿真时，输入下面带有 DVE 使能选项的指令（options enabling DVE）：

```
vcs -mhdl (-debug | -debug_all) [options] design.v  
simv -gui [runtime_options]
```

## 一个拥有 VHDL 为顶层文件的混合仿真

运行一个 VHDL 为顶层文件的 Verilog/VHDL 混合仿真，输入下面带有 DVE 使能选项的命令：

```
vlogan Verilog_files [options]  
vhdlan vhdl_filename -vlib Verilog  
scs cfg_tb (-debug | -debug_all) -verilogcomp "options"  
scsim -gui -verilogrun "-q" [options]
```

## 检查点还原的方法

当保存和还原一个仿真的时候，用相同的技术或流程来还原你曾经保存的检查点（checkpoint），例如：

- 应用 UCLI 指令保存和还原
- 在 DVE 中保存和还原
- 应用 SCL 指令保存和还原
- 应用 CLI 指令保存和还原

千万不要将保存和还原的指令相互之间弄混，例如：

- 用 UCLI 指令保存却用 SCL 指令还原
- 在 DVE 中保存却用 CULI 指令还原
- 保存用 UCLI 指令却用 DVE 还原
- 用 CLI 指令保存却用 UCLI 指令还原

并且如果你正在提交一个用 VHPI 或 PLI 与 VCS MX 通信的外部申请，并且已经有文件为了这个申请被打开了，在你还原后你必须在你下次保存或者打开之前将这些文件关掉。

## C, C++,和 System C 代码

以下步骤主要概括了用 UCLI 去调试 VCS 或是 VCS MX 上的那些含有 C,C++,和 SystemC 的源代码的仿真。

像通常那样编译包含 C, C++,和 System C 模块的 VCS 或 VCS MX，确保所有的你要调试的 C 文件都被编译到。

例如，一个有 Verilog 在 C 或 C++模块顶层的设计：

```
gcc -g [options] -c my_pli_code.c  
vcs +vc -debug_all -P my_pli_code.tab my_pli_code.o
```

或者一个有 Verilog 在 SystemC 模块顶层的设计：

```
syscan -cflags -g  
syscan -cpp g++ -cflags "-g" my_module.cpp:my_module  
vcs -cpp g++ -sysc -debug_all top.v
```

注意你必须用 `-debug` 或 `-debug_all` 来调用调试功能。

在 DVE 中打开仿真

```
Simv -gui
```

3、选择 **Simulator > C/C++ Debugger** 来启动 C 调试器。

## 在 GUI 界面运行仿真

你可以在 GUI 界面打开 DVE 并且运行仿真。

### 后台处理程序 (Post-processing)

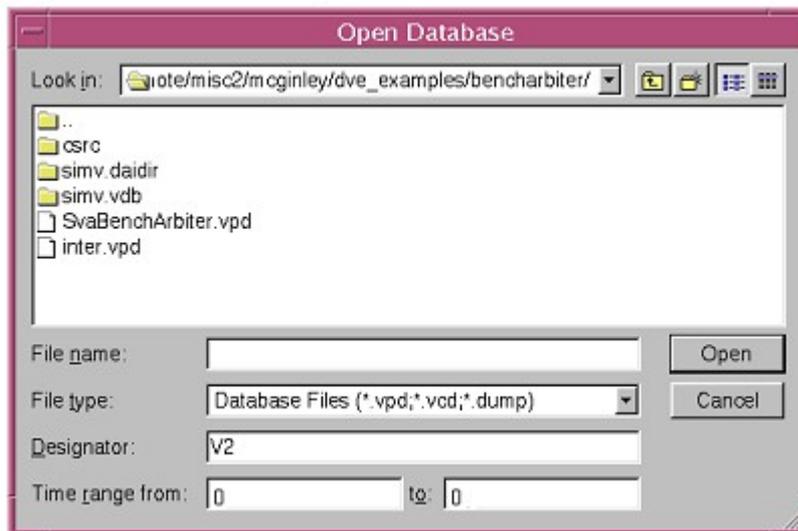
对于后台处理程序来说你会载入并且显示任何数目的 VPD 文件。在 DVE 中打开一个数据库文件:

做下面两个操作中的任意一个:

- 在 **Menu** 按钮下, 选择 **File>Open Database**.
- 在工具栏点击  图标。

打开数据库文件的对话框就出现了

Figure 1-1 Open Database Dialog Box.



2、在打开数据对话框里, 翻阅到并且选择你要载入的 VPD 文件。

3、为你的设计输入或选择一个名字。

4、输入时间范围来载入。默认的是从仿真开始到结束。

5、点击 **Open**

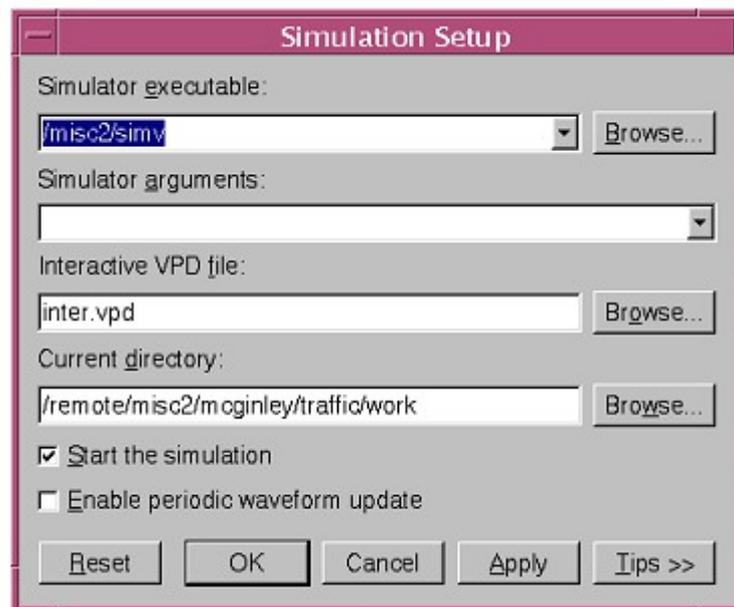
DVE 就载入了所选的 VPD 文件。

## 创建并且运行一个交互式的会话

另外为了后台处理载入的 VPD 文件，你也可以应用一个编译过的 Verilog,VHDL,或是二者的混合设计，来实时的创建并且交互式的运行一个仿真。在命令行打开 DVE

**%dve**

选择 **Simulator>Setup**,然后从模拟器的创建对话框选择开始仿真。



1. 翻到一个模拟器的可执行界面（simulator executable）。
2. 输入模拟器的属性。
3. 设置这个 VPD 文件的名字或选择一个已存在的文件，该文件将会在交互式的会话中被写入。
4. 点击 **OK** 创建仿真。

## 运行仿真

这节描述了应用 DVE 来运行并且控制仿真。

涵盖了下面的几方面内容：

- 工具栏的应用
- 仿真其菜单命令的应用
- 命令行选项的应用

### 工具栏的应用

当开始一个仿真的时候，DVE 激活了工具命令行来运行和控制仿真。在模拟器中点击下面这些图标来控制仿真。

图标	功能描述
 Start/Continue	运行仿真程序直到一个中断点（breakpoint）被触发为止，例如仿真结束，或该期间在对话框或者是工具栏指定了持续时间。
 Run for Specified Time	运行指定时间的仿真后停止。
 Stop	当运行仿真的时候该图标被激活，点击它来停止仿真
 Next	对于 VHDL、Verilog、和 TB 代码而言，进行下一步任务或函数的操作。（For VHDL, Verilog, and TB code, next steps over

	tasks and functions.)
 Step In	将仿真逐行代码扫描式的进行，不考虑代码的语言。这与 UCLI 的 Step 指令是一样的
 Step In Active Thread	在当前激活的线程里逐步的跳转到下一个可执行的命令行。
 Step In Any Test-bench Thread	对于本地测试平台(NativeTB)OpenVera 和 SV TB 而言，在平台中下一个可执行的命令行处停止
 Step Out	扫描到下一个可执行命令行跳出当前的函数或任务
 Restart	停止当前运行的仿真并且用当前仿真的设置来从启该仿真。这个操作保留所有打开的窗口和 GUI 界面设置。如果没有运行仿真该操作就启动仿真

## 模拟器（Simulator）菜单命令的应用

在开始仿真后，可以用菜单命令来运行和控制仿真程序。选择下面的命令来控制仿真：

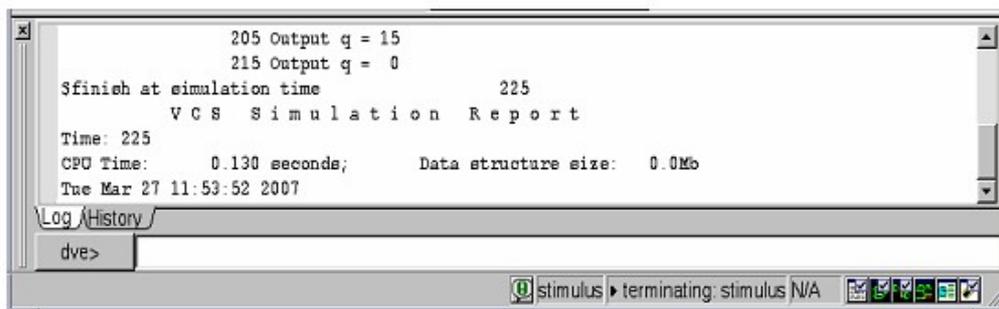
指令	功能描述
<b>Start/Continue</b>	运行仿真程序直到一个中断点（breakpoint）被触发，或仿真结束，或到了该期间在对话框或者是工具栏指定的持续时间。。
<b>Step</b>	逐行代码扫描式的进行仿真，不考虑代码的语言 这与 UCLI 的 Step 指令功能相同

<b>Next</b>	对于 VHDL、Verilog、和 TB 代码而言，进行下一步任务或函数的操作。（For VHDL, Verilog, and TB code, next steps over tasks and functions.）
<b>Step In Active Thread</b>	在当前激活的线程里逐步的跳转到下一个可执行的命令行。
<b>Step In Testbench</b>	对于本地测试平台(NativeTB)OpenVera 和 SV TB 而言，在平台中下一个可执行的命令行处停止
<b>Step out</b>	扫描到下一个可执行命令行时跳出当前的函数或任务
<b>Restart</b>	停止当前运行的仿真并且用当前仿真的设置来重启该仿真。这个操作保留所有打开的窗口和 GUI 界面设置。如果没有运行仿真该操作就启动仿真
<b>Stop</b>	停止一个正在运行的仿真 (与 UCLI 的 <b>stop</b> 指令一样)

## 命令行的应用

用 DVE 顶层窗口底部的命令行进入 DVE，并且统一命令行界面（UCLI）运行并且控制仿真的执行。图 1-2 展示了在哪输入指令并且有结果显示在命令行上面的日志文件（Log tab）里。

*Figure 1-2 Command Line with the Log tab*



查看 DVE 的指令可以输入：

### help -gui

应用 UCLI 时为了得到完整的信息，注意统一的命令语言的用户向导。例如一个快速视图查看的 UCLI 指令和它的应用，在 DVE 命令行快速的输入：

### help -ucli [argument]

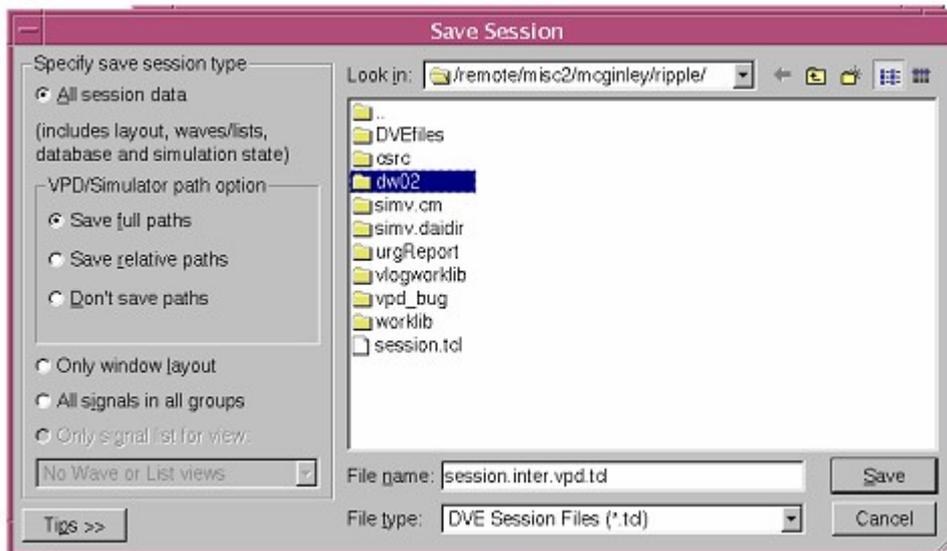
当输入没有指定属性的时候，UCLI 的指令列表和一个简短的描述就被显示出来了。输入一个命令名作为属性的时候，描述和指令的规则就被显示出来了。

UCLI 指令和详细的描述被显示出来。

## 保存会话或布局图

用保存会话对话框来存储（preserve）会话数据显示图和 VPD 路径选项。

### 1、选择 **Select File>Save Session**



2、在保存会话对话框中为该会话输入一个文件名

3、在选项上进行选择来指定会话的类型后保存：

- 所有的会话数据包括布局图、波形列表、数据库和仿真状态。
- 窗口布局图保存了窗口协议（arrangement）、外观（views）、和窗格为了今后的复用。这个选项不会保存任何的数据内容。

- 所有组里面的全部信号保存了所有信号组的信号列表。
  - 对于观看一个波形或列表来说，只有为了观看的信号列表保存了信号信号列表。在打开任何必要的模拟器或 VPD 文件后，这个信号列表能够用当前的信号列表重新载入一个窗口。
- 4、为 VPD 选择一个路径选项：
- 对于多个打开的 VPD 文件来说默认是保存绝对路径。
  - 对于一个打开的 VPD 文件来说保存的是相对路径（相对指向于会话文件存储的文件夹）
  - 不要保存打开的 VPD 文件。如果混合设计被打开了，该选项就被屏蔽掉了。
- 5、选择保存内容：
- 仅保存窗口布局图
  - 所有组中的全部信号
  - 观看一个波形或列表，仅保存用来观看的信号列表
- 6、点击保存。

## 保存一个布局图会话

保存一个窗口布局图，并且保持窗口的位置，窗口的大小，和一些其他的所出现的设置而不保存窗口的内容，进行下面的操作即可：

- 1、选择 **File > Save Layout Session**
- 2、在保存会话对话框为该布局图会话输入一个文件名。
- 3、点击保存(**Save**)。

## 载入已保存的会话（session）

载入一个保存过的会话：

- 1、载入一个 VPD 文件。
- 2、选择 **File>Load Session**.
- 3、在载入会话对话框，导入该会话并且在保存过的会话 TCL 文件列表中将其选

中。

#### 4、点击载入 (Load)

## 关闭数据库

关闭当前打开的数据库:

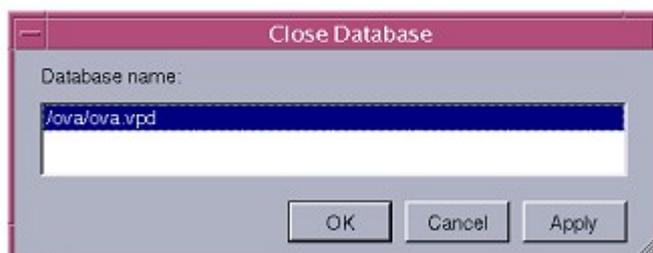
做下面 2 步中的任意 1 步

- 选择 **File>Close Database**

- 在工具栏点击关闭数据库图标 。

出现了关闭数据库对话框。见图 1-3。

*Figure 1-3 Close Database Confirmation Dialog Box*



确保选对数据库，然后点击 **OK**。

## 退出 DVE

选择 **File>Exit**，退出 DVE。

## 2 图形化界面应用

本章描述了应用图形化界面得基本方法和窗口管理。包含以下章节：

- DVE 窗口属性概述
- DVE 窗格 (panes)
- 管理 DVE 窗口
- 菜单栏和工具栏应用
- 设置显示的首选项 (Preferences)

### DVE 窗口属性概述

DVE 拥有一个完全自由的窗口模式。这种模式是基于顶层窗口的想法建立的。一个顶层窗口包含一个框架、菜单栏、工具栏、状态栏、目标窗口。你可以打开任意数目的窗口。在开始的默认状态下是开启一个窗口。

DVE 顶层窗口是一个显示设计和调试数据的框架。默认的 DVE 窗口属性只显示包含层次化的浏览器在左边，在底部显示控制台窗口，并且剩下的所有空间都是源窗口。可以用首选项文件来改变默认属性，例如一个会话文件或一个开始脚本文件。图 2-1 展示了默认的顶层窗口。

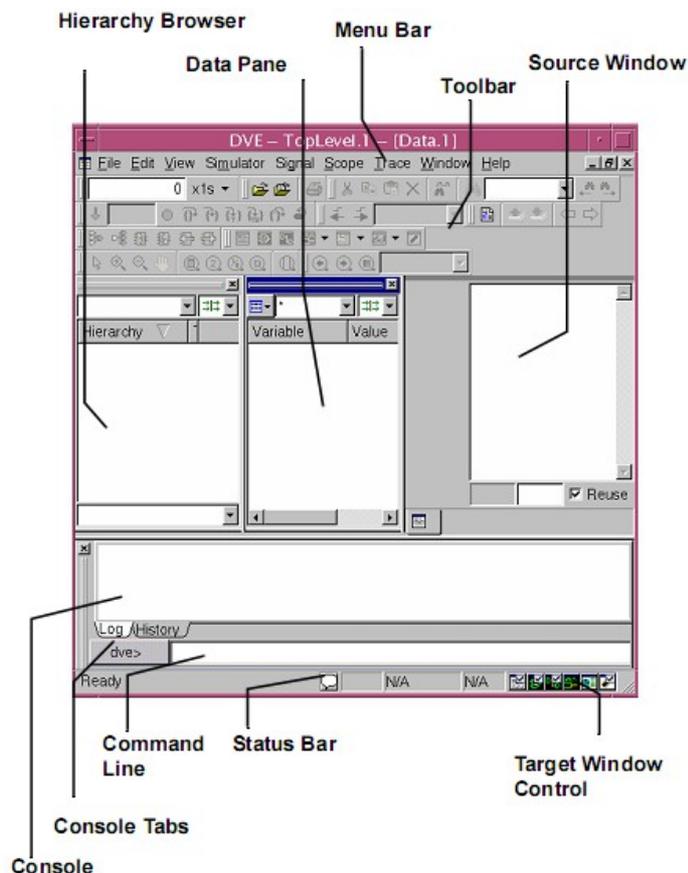
可以在根目录下创建一个 `synopsys_dve_usersetup.tcl` 文件来保存 DVE 的快捷键。

例如：

```
gui_set_hotkey -menu "Signal->Compare..." -hot_key "c"
```

`synopsys_dve_prefs.tcl` 文件存储了用户的常用属性。

该文件由工具自动生成。



## DVE 窗格 (panes)

一个顶层窗口能够包含任意数目的窗格。窗格就是提供一个指定调试目的窗口。例如层次目录 (Hierarchy)，数据，波形，列表，记录和原理图等都是窗格。

## 管理 DVE 窗口

DVE 顶层窗口能够包含任何数目的 DVE 窗口和窗格。通过设置默认值可以选择让数据显示在一个或多个 DVE 窗口和窗格中，可以用状态栏窗口来控制或创建和取消你工作的窗口。

## 管理目标窗格

目标说明（policy）指定了窗格将在哪里创建。在每一个顶层框架在其右下方由一个目标图标（图 2-2）。这些图标代表了窗格的类型。

Check marks indicate targeted windows are attached to the current window.



No check in targeted Wave window icon

目标图标有下面两种状态：

- 上面带钩得为目标图标，它意味着需要创建一个新的窗格时候在当前的框架下创建。
- 不带钩的为非目标图标，它意味着需要创建一个新的窗格时候同时要创建一个新的顶层窗口来包含这个窗格。

在一个新的顶层窗口下打开一个窗格：

- 1、在状态栏点击图标来移动复选标记。

 <b>Source</b>	在一个新顶层窗口中之指向一个新的源窗格
 <b>Schematic</b>	在一个新顶层窗口中指向一个新的原理图窗格
 <b>Path Schematic</b>	在一个新顶层窗口中指向一个新的路径原理图窗格

 Wave	在一个新顶层窗口中指向一个新的波形窗格
 List	在一个新顶层窗口中指向一个列表窗格
 Memory	在一个新顶层窗口中指向一个新内存窗格

在工具栏点击一个相应类型的窗口图标来打开一个该类型的窗口。这个窗口不会被附加到当前窗口而是会在一个新顶层窗口中打开。

## 停放与移除窗口和窗格

可以用窗口菜单来插入与撤销窗口和窗格

- 选择 **Windows > Dock in New Row** 然后选择行位置在该处停放当前激活的窗口。
- 选择 **Windows > Dock in New Column** 然后选择列位置在该处停放当前激活的窗口。
- 选择 **Undock** 移除当前激活得窗口或窗格。

点击窗格角上的 X 图标来删除一个窗口，这对于删除所有的窗口来说是相同的。深蓝色的停放标志标示着该停放的窗口处于激活状态。这个操作对于所有可停放的窗口来说是相同的。有一些操作是必须执行的，例如：让窗口被激活你必须点击一下。

## 拖拽和删除一个已停放的窗口

左键点击停放标识然后拖拽并将其放到一个新停放位置或是一个未被停放的窗

口。

右键点击停放图标出现了一个弹出菜单:

<b>Undock</b>	删除但前激活的窗口
<b>Dock</b>	<b>Left</b> -将所选窗口停放到顶层窗口的左边 <b>Right</b> -将所选窗口停放到顶层窗口的右边 <b>Top</b> -将所选窗口停放到顶层窗口的上面。不推荐使用。 <b>Bottom</b> -将所选窗口停放到顶层窗口的下面。

## 菜单栏和工具栏的应用

菜单栏和工具栏允许你执行一些标准的仿真分析任务，例如打开和关闭一个数据库，移动波形表来显示不同的仿真时间，或是视图查看 HDL 源代码。

本节涵盖了以下题目：

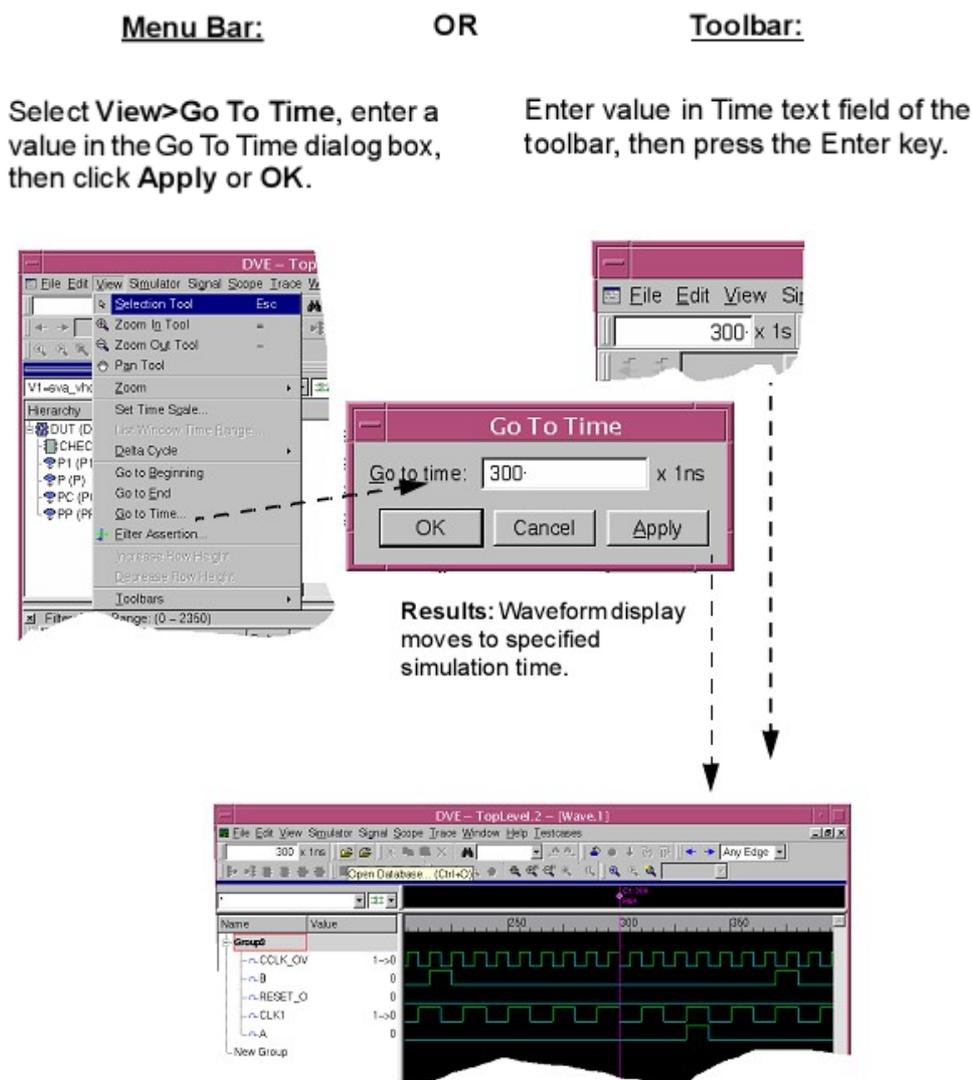
菜单栏参考

工具栏参考

菜单栏的大多数条目与工具栏中的图标或文本域是一一对应的。例如，你可以通过下面 2 个操作中的任意一个来设置显示在波形表中的仿真时间：

- 选择 **View>Go To Time** 然后在 **Go To Time** 对话框输入一个值，然后点击 **Apply** 或 **OK**。
- 在工具栏的时间文本域输入一个值后按回车键

例如图 2-3



注:

- 菜单栏和工具栏功能的详细描述见菜单栏和工具栏参考。
- 设置仿真时间和波形窗口的应用详见 Chapter4“波形窗口的应用”。

## 菜单栏参考

本节提供了下面的顶层窗口菜单的概述:

- 文件菜单

- 编辑菜单
- 视图菜单
- 仿真器菜单
- 信号菜单
- 作用域 (Scope) 菜单
- 标示 (trace) 菜单
- 窗口菜单
- 帮助菜单

## 文件菜单

下面的条目构成了文件菜单：

<b>Open Database</b>	显示打开数据库对话框，它可以让你选择和打开一个仿真数据库来进行后台处理
<b>Close Database</b>	显示关闭数据库对话框，它可以让你关闭一个仿真数据库 (VPD) 文件。
<b>Reload Database</b>	载入一个以前打开的数据库
<b>Open File</b>	显示打开源文件的对话框，它可以在源窗口让你选择并且显示一个源文件。
<b>Close File</b>	关闭显示在激活的源窗口或窗口中的文件
<b>Save Values</b>	通过下面的选项来保存值： <b>制表表单</b> <b>基于事件的表单</b> <b>内存内容</b>
<b>Execute Tcl Script</b>	显示执行 TCL 脚本文件对话框
<b>Load Session</b>	显示载入会话对话框它可以让你载入一个已存储的会话。
<b>Save Session</b>	显示存储会话对话框，它可以让你存储当前的会话
<b>Print</b>	将一个激活的波形、列表、或原理图窗口的内容打印到打印机或文件中
<b>Save Wave/List Session</b>	保存波形和列出会话数据
<b>Recent Databases</b>	显示一个最近打开的数据库列表然后从中选择
<b>Recent Tcl Scripts</b>	显示一个最近运行的脚本列表然后从中选择
<b>Recent Sessions</b>	显示一个最近打开的会话列表然后从中选择
<b>Close Window</b>	在顶层窗口中关闭当前激活的窗格
<b>Exit</b>	退出 DVE

## 编辑菜单

下面的条目构成了编辑菜单:

<p><b>Cut/Copy/Paste/ Paste From/ Delete</b></p>	<p>复制工作在任何文本中。如果复制功能将一个文本作为选定目标，复制将会复制该目标。在其他方面将会复制选中的文本。被复制的文本能被粘贴到任何支持文本的部件，例如一个编辑器或 DVE 的命令行。目标复制工作在小部件中，例如 DVE 窗格，它支持了 DVE 目标整理 DDVE 目标例如任何的 DVE 窗格</p> <p>注：剪切和删除仅工作在 DVE 目标上和一些窗口中，并且在一些窗口中是受限的，例如波形、列表和内存窗口。</p> <p>从一个生成的剪切板目标复制过来的目标文件</p>
<p><b>Expand/ Collapse</b></p>	<p>这些菜单条目总体的工作在激活的 DVE 列表部件和 DVE 源窗口中。列表部件是层次化的显示列表的窗格，也就是说在这里面条目可以有一些子条目。如果一个条目有一个子条目并且是不可访问的，那个子条目被叫做压缩的。总条目会有一个 '+' 标记指示至少有一个子条目通过左键点击 '+' 能被看见。如果总条目有 '-' 标记，它意味着该条目已经被展开来展示它的子条目。左键点击 '-' 将会指示子条目合并成总条目。</p>
<p><b>Expand By Levels &gt;</b></p>	<p>提供了以下子菜单：</p> <p><b>ALL</b> - 在当前选中的条目中展开所有层次的全部条目。注：这如果在一个大型设计中执行会花费很长的时间。</p> <p><b>2</b> - 从当前选中的条目中展开 2 层的子条目</p> <p><b>3</b> - 从当前选中的条目中展开 3 层的子条目</p> <p><b>4</b> - 从当前选中的条目中展开 4 层的子条目</p> <p><b>5</b> - 从当前选中的条目中展开 5 层的子条目</p>
<p><b>Expand All</b></p>	<p>展开选中条目中的所有总条目下的子条目</p>
<p><b>Collapse Parent</b></p>	<p>折叠当前选中的条目中的总条目。如果没有条目被选中，将不会执行任何操作。</p>
<p><b>Collapse All</b></p>	<p>将选中的条目中的所有子条目折叠成总目录不顾任何后果</p>
<p><b>Synchronize Selection</b></p>	<p>不是全局选择。你可以在不同时候在不同窗格中选择不同条目。同步选择允许你同时将所有窗格一次选中。例如如果你有一个信号在波形窗口中被选择但是他的总作用域没有在层次的窗口中，点击同步选择将会导致该信号的主作用域添加到层次窗口中也会被添加到数据窗格中和其他任何该信号存在的窗格中。该功能在在原理图窗口中是非常实用的</p>

<b>Select by Levels</b>	<p>提供了以下子菜单:</p> <p><b>ALL</b> - 在当前选中条目中选中所有层里的全部条目。注: 这如果在一个大型设计中执行会花费很长的时间。</p> <p><b>2</b> - 从当前选中条目中选择 2 层子条目</p> <p><b>3</b> - 从当前选中条目中选择 3 层子条目</p> <p><b>4</b> - 从当前选中条目中选择 4 层子条目</p> <p><b>5</b> - 从当前选中条目中选择 5 层子条目</p>
<b>Select All</b>	在窗口或窗格中选择所有目标
<b>Find</b>	显示查找对话框
<b>Find Next</b>	在查找对话框或查找菜单行有任何文件存在的时候激活该项。如果点击它, 将会找下一个文本目标在激活的窗格中
<b>Find Previous</b>	与查找下一个相同但是它查找的是以前查找的文本
<b>Goto Address</b>	显示了一个你可以输入地址的对话框。如果内存窗口被打开且有安装有原件的该选项就被激活
<b>Search for Signals/ Instances</b>	显示查找信号对话框。用该功能来查找任何在打开且当前数据库存在的目标。如果该目标没有被载入对话框将试图载入它。
<b>Create Marker</b>	在波形窗口中创建一个指示器。这项仅是在一个波形窗格存在于一个顶层窗口, 对于该窗口编辑菜单是被激活的。如果一个波形窗格存在, 点击这个菜单选项将你带到一个创建指示器的模式下。一个白色的建议标识被创建了并且它在波形窗格中紧随鼠标。该标识被放在了鼠标左键下一次点击的位置。该标识将被放在鼠标左键下次点击的地方
<b>Markers . . .</b>	显示标记对话框 (见 5-17)
<b>Goto Marker</b>	提供了一个动态的子菜单。如果出现该子菜单包含了所有在当前激活的波形窗格中可访问的标识。点击任何的标识该标识被展示在波形窗口的中间以供视图查看。如果没有标识出现在当前窗口, 该子菜单将会是空的
<b>Delete Marker</b>	提供了一个动态的子窗口。如果出现该窗口其包含了所有在当前激活的波形窗格中的可访问的标识。点击任何早子菜单中的标识来删除它。如果没有标识存在该窗口将为空的。
<b>Move Marker</b>	提供了一个动态的子窗口。如果出现该窗口其包含了所有在当前激活的波形窗格中的可访问的标识。点击任何标识展示了选择标识, 然后点击在波形窗格中所期望的位置来移动该标识。如果没有标识存在该窗口将为空的。
<b>Set Reference Marker</b>	设置一个当前选定的标识作为参考标识在与其它标识链接时显示标识值, 这对于下一个菜单条是非常有用的。
<b>Show Marker Values</b>	对于在一个已选标识的信号来说纯粹的显示, 附近的或相关的值。详见 (5-17)
<b>Preferences</b>	显示常用对话框 (详见 (2-38))

## 视图查看菜单

下面条目构成了视图查看菜单：

<b>Selection Tool</b>	用来视图查看原理图。改变鼠标的行为例如左键单击和拖拽所选目标。左键单击在一个单独的目标上选定它。 <b>Ctrl</b> 加左键将目标添加到所选项中。按住左键并且拖拽所有在框中的目标将被选中。
<b>Zoom In Tool</b>	用来视图查看原理图。改变鼠标行为左键单击扩大。按住左键拖拽创建了一个原理图的新视角。
<b>Zoom Out Tool</b>	用来视图查看原理图。改变鼠标的行为左键单击缩小
<b>Pan Tool</b>	用来视图查看原理图。变鼠标的行为按住左键且拖拽移动所有原理图中的目标的位置。例如按住左键向下拖拽，将在视线里的所有目标下移。选择这个菜单条目同时将光标转换成了手型。点击工具菜单选择将光标转换成常规模式
<b>Zoom&gt;</b>	<p>为所有的扩大或缩小操作创建一个子菜单。这些菜单条目只被原理图和波形窗格所接受。</p> <p><b>Zoom Full</b>-让所选的区域填满窗格</p> <p><b>Zoom In</b>-使当前视图查看的目标变成 2 倍大小因此会看见更少的目标</p> <p><b>Zoom Out</b>-使当前视图查看的目标变成 1/2 大小因此会看见更多的目标</p> <p><b>Zoom Fit Selection</b>-使当前视图查看的窗口处于所选目标原理图设置的中心并且所有选中的目标在当前窗口中是可访问的</p> <p><b>Zoom Fit Highlight</b>-和 <b>Zoom Fit Selection</b> 的含义一样，只是突出对象而已，如果对象已经突出，这个操作也是可行的</p> <p><b>Zoom To Selection</b>-</p> <p><b>Pan To Selection</b> 移动视图以便使选定的对象为中心，并可在当前面板中浏览，但并不能改变缩放</p> <p><b>Pan To Highlight</b>-和 <b>Pan To Selection</b> 的含义一样，只是它突出了原理图对象而已</p> <p><b>Zoom to Cursors</b> 如果有两个波形窗口光标存在，将两个光标之间的区域两边位置调换</p> <p><b>Zoom to Time Range</b> 在显示的对话框中输入一个时间范围，然后放大这一时间范围</p> <p><b>Back in Zoom and Pan History</b> 对当前窗口通过保存缩放来进行迭代，当你在视图中更改了缩放，DVE 会存储先前的视图，以便你可以找回</p> <p><b>Forward in Zoom and Pan History</b> 如果你清除了</p>

	<p><b>zoom/pan</b> 的历史记录, 该选项提供一个非常简单的方法使你进行下一个视图, 点击该选项就可以获得当前运行窗口 <b>Named Zoom and Pan Settings</b> 在显示的对话框中, 你可以选择任何的视图, 然后命名保存</p>
<b>Set Time Scale</b>	在该对话框中你可以改变仿真中的时间精度
<b>List Window Time Range</b>	<p>显示一个对话框, 你可以在列表窗格中更改脉冲时间。此选项只有在列表窗格被打开时才是可用的</p>
<b>Delta Cycle &gt;</b>	<p>为增量周期创建一个子选项, 只有增值周期存在值变化时, 该选项才是可行的。</p> <p>展开时间 - 从 C1 时刻到增值周期这段时间内展开。</p> <p>压缩时间 - 压缩从增值周期扩展出的时间。</p> <p>全部压缩 - 无论 C1 在何处, 压缩所有超出周期的时间</p>
<b>Go to Beginning</b>	迅速的开始仿真 (通常为 0 时刻)。也可以通过移动 C1 的光标和改变波形图中的视图来改变仿真时间。
<b>Go to End</b>	仿真时间的结束
<b>Go to Time</b>	在模拟器中会显示出一个对话框, 允许你改变时间 C1, 在 C1 时刻, 波形图中的视图会发生变化
<b>Link C1 to Sim Time</b>	根据现行的仿真时间来设定调试时间, 此项只能在调试会话窗实现。
<b>Move C1 to Sim Time</b>	同步调试时间和现行仿真时间,
<b>Increase Row Height</b>	增加波形窗口中所有追踪信号的行高
<b>Decrease Row Height</b>	降低波形窗口中所有追踪信号的行高
<b>Set Default Row Height</b>	将波形图中所有高度恢复到默认值
<b>Toolbars &gt;</b>	<p>通过检验菜单的子选项, 你可以打开和关闭以下工具条:</p> <p><b>Edit</b></p> <p><b>File</b></p> <p><b>Scope</b></p> <p><b>Trace</b></p> <p><b>Window</b></p> <p><b>Signal</b></p> <p><b>Simulator</b></p> <p><b>Time Operations</b></p> <p><b>Zoom</b></p> <p><b>Zoom and Pan History</b></p>

## 仿真菜单

下面的条目构成了仿真菜单

<b>Setup . . .</b>	显示新建仿真的对话框，你可以修改默认的仿真时间。注意，但你并不可以控指仿真编译的设定，详见 1-11 页“ <b>Running a Simulation from the GUI</b> ”。
<b>Rebuild and Start</b>	创建一个 csrc/文件就可重建一个 Verilog 仿真，然后开始仿真。此选项在 VCS MX 和 VHDL 不起作用
<b>Start/Continue</b>	点击此键可以停止当前的仿真，或是仿真完成，或是 Set Continue Time 对话框和工具菜单中的时间已到，都可以使仿真停止。
<b>Stop</b>	停止当前运行的仿真程序（和 UCLI 的停止命令一样）。
<b>Step</b>	逐行的执行代码，不论是什么语言的代码，和 UCLI 的 Step 命令是一样的。
<b>Next</b>	对于 VHDL, Verilog, 和 TB 代码，执行下一个任务和函数。
<b>Step In Active Thread</b>	在当前运行的线程中，在将要执行的代码行前停止
<b>Step In Testbench</b>	对于 Native TestBench(NTB) OpenVera SystemVerilog 测试平台，在将要执行的测试代码前停止。
<b>Step Out</b>	跳过任务和代码，执行下一个将要执行的代码。
<b>Restart</b>	停止当前运行的仿真并新建仿真来重新开始。
<b>Show stack</b>	显示 c 语言，SystemVerilog 和 NativeTestBench 测试平台 栈
<b>Move up Stack</b>	向上移动当前堆栈
<b>Move down Stack</b>	向下移动当前堆栈
<b>Breakpoints</b>	显示断点对话框，允许你视图查看，新建，编辑，启用和删除断点。详看 4-13 页“managing breakpoints in interactive simulation”
<b>Save State</b>	在一个文件浏览器对话框中，可以将当前状态的模拟器作为文件名来保存。
<b>Restore State</b>	在一个文件浏览器对话框中，你可以恢复以保存的仿真状态。
<b>Terminate</b>	终止一个运行的和已停止的仿真。
<b>Dump Full Hierarchy</b>	整合在层次中的所以信息。
<b>Add Dump . . .</b>	在转储对话框中将指定的信号赋值，在当前时刻开始改变。详见 3-13 页“Dumping Signal Values”。
<b>Dump</b>	在当前面板中打开所有被选定的
<b>Add Force</b>	在 Force Value 对话框中，你可以改变信号的值。详见 3-15 页“Forcing Signal Values”。

<b>Capture Delta Cycle Values</b>	在当前时刻切换开启/关闭来开启增值周期。请注意这大大增加了 VPD 文件的大小，你应该限制时间跨度来转存周期值。
<b>Set Continue Time ...</b>	显示一个对话框用于输入一个时间。如果没有点击断点，仿真就会在指定的时间期限内运行。如果设定为 10，当你按工具菜单中的 <b>Continue</b> 时，仿真就会运行 10 秒。你可以使用工具菜单做到这一点，以及一条快捷方式
<b>Periodic Waveform Update Interval ...</b>	显示一个对话框，在仿真运行时使用户可以选择一个支持定期更新的值来改变数据，这样在仿真时就可以看到动态的波形，时间间隔越小，效果越差。
<b>C/C++ Debugging</b>	可以调试 C/C++ 和 System C 代码。

## 信号菜单

<b>Display Signal Groups &gt;</b>	创建一个动态子菜单，可以对信号组的可见度进行操作。 <b>Select</b> – 显示一个对话框，让你可以打开/关闭任何信号组的可见性。 <b>All</b> – 打开信号组中所有的信号的可见性。
<b>Add to Waves</b>	为已选择信号增加以下选项： 新波形 最近波形 可用波形，为当前的波形创建一个子菜单。
<b>Add to Lists</b>	和 <b>Add to Waves</b> 一样，只是对于列表窗口。
<b>Add to Signal Groups</b>	<b>New</b> – 创建一个新的信号组，用 <b>Group&lt;n&gt;</b> 来命名，n 是比信号组现有的最高数还高的数。新的信号组创建在信号列表的最上端。 <b>GroupName</b> – 增加了选择的信号或是先前定义的信号。
<b>Show Memory</b>	如果已选择的信号是一个内存或是 MDA, 那么这个菜单选项就会在内存视图上显示。如果没有内存视图，DVE 就会根据目标创建一个。
<b>Set Insertion Bar</b>	在波形或是列表窗口中选定的信号之上设定一个插入工具菜单。在别的窗口中这个菜单项是看不到的。如果有两个或更多的信号被选中，DVE 会将插入工具菜单放在离信号列表最近的那个信号之上。
<b>Insert Divider</b>	在波形显示中插入一个空白分割行
<b>Sort Signals</b>	通过声明、升序、降序来显示信号
<b>Set Bus ...</b>	显示 <b>Bus/Expression</b> 对话框来控制总线 and 表示新建和删除。详见 5—26 页 <b>“Building Buses and Setting Expressions”</b>

<b>Set Expression</b>	显示表达式对话框来控制总线并表示新建和删除。详见 5—26 页 “Building Buses and Setting Expressions”。
<b>Set Search Constraints&gt;</b>	<p>创建一个动态的子菜单，来控制 Search Previous 和 Search Next 按钮。当约束相匹配时，C1 光标就会移动到那个时候的位置。如果没有信号被选择，那么查找是没有用的。所支持的约束有：</p> <p><b>Any Edge-</b> (默认) 查找下一个或上一页找到光标 C1 所在的位置。</p> <p><b>Rising</b> – 在下一个或上一个上升沿处，查找光标 C1 的位置。</p> <p><b>Falling</b> -在下一个或上一个下降沿处，查找光标 C1 的位置。</p> <p><b>Failure</b> – 如果信号是一个断言才是可用的，下一个或是先前的断言都是失败的。</p> <p><b>Match</b>-如果信号是一个断言才是可用的，下一个或是先前的断言都是错配的。</p> <p><b>Success</b> -如果信号是一个断言才是可用的，下一个或是先前的断言都是成功的。</p> <p><b>Signal Value</b> – 显示一个小窗口，你可以输入一个具体值作为约束。如果这个值是可以找到的，查找就会停止，并且光标会落在所需要的信号输入值处。值必须符合当前选定信号的基数。</p>
<b>Search Backward</b>	使用上面指定的约束，按时间向后查找约束
<b>Search Forward</b>	使用上面指定的约束，按时间向前查找约束
<b>Compare...</b>	显示信号比较对话框，允许你指定一个信号来比较。详见 5-24
<b>Show Compare Info</b>	只有在信号已经进行比较时才是可用的，显示最后的信号比较结果。
<b>Shift Time</b>	显示 <b>Shift Time</b> 对话框，指定一个具体的参数来移动一个信号。
<b>Set Radix &gt;</b>	<p>为所选的信号创建一个含有改变基数的合理选项的子菜单，为信号改变基数是全局的，不管信号显示在模拟器的什么位置。</p> <p>User Defined&gt;在对话框中你可以指定和编辑用户默认的类型。详见 5-10 页 “<b>Creating a User-Defines Radix</b>”。</p> <p><b>Enumerated Type</b></p> <p><b>ASCII</b></p> <p><b>Binary</b></p> <p><b>Octal</b></p> <p><b>Decimal</b></p> <p><b>Hexadecimal</b></p> <p><b>Unsigned</b></p> <p><b>Signed magnitude</b></p>

	<b>One`s Complement</b> <b>Two`s Complement</b> <b>Strength</b> <b>Default</b>
<b>Default Properties</b>	为所选的信号提供默认的工具
<b>Properties</b>	显示信号道具对话框, 你可以熟练的控制信号的外观

## 作用域菜单

<b>Show Source</b>	显示所选信号的资料。如果有成倍的对象倍选中, 显示第一个被选中的对象。如果已经在检验菜单中检验了, Source 窗口用的就是当前打开的资源窗口。如果没有资源窗口存在, DVE 就会根据目标创建一个新的资源窗口。
<b>Show Schematic</b>	像上述的一样, 只是对象在 <b>Dsign Schematic</b> 面板中显示。
<b>Show Path Schematic</b>	像上述的一样, 只是对象在 <b>Path Schematic</b> 面板中显示。
<b>Note:</b> 以下菜单选项会对当前的资源窗口产生影响	
<b>Move Up to Parent</b>	在当前所选行的范围内将资源窗口中的选项移动到最高一层。如果当前运行的行就在最高层或是没有选中的行, 这个选项就是不可用的。
<b>Move Down to Definition</b>	将选项移动到当前对象定义开始的地方。特别注意资源窗口, 只有对象被选定时才起作用, 如果整条被选中并且至少有一个在别的行被选中, 则这个选项是不起作用的。
<b>Back</b>	在现行窗口中移动先前资源信息的视图。DVE 保存一个历史记录, 所以很容易找到先前用过的资源视图。这对大的资源文件和减少上下行的切换是十分有用的。
<b>Forward</b>	像前所述, 只是向前查找资源视图, 如果你已经向后查找过了。
<b>Show &gt;</b>	<p>创建一个动态的子菜单, 对相关的找到合适的类型做个导航, 这些类型都是基于所选信号的类型。</p> <p><b>Current Scope</b> – 改变当前范围的第一行选项。</p> <p><b>Assertion</b> – 如果对象是一个断言是实例的话, 该选项就会改变已选断言的资源视图。</p> <p><b>Unit binding</b> – 对于 OVA 的断言实例, 在带有断言声明的模块的上方, 通过此选项可以改变该位置的资源视图。</p>

	<p>Entity – 如果此选项是在 VHDL 的体系框架下运行，那么此操作会改变整个体系的资源视图。</p> <p><b>Architecture</b> – 如果现行选项是在整个 VHDL 中运行，此选项会改变整个体系的资源视图。</p>
<b>Edit Source</b>	根据你对资源编辑的首选项设定来显示一个文本编译器（默认的是 vi 编译器）。DVE 会预载带有资源文件的编译器，这些文件都是在当前 DVE 资源面板上活动的，并且是和所选择在相同的位置。，如果在资源窗口中没有文件被打开或有不同类型的 DVE 面板在运行，这个条目就是不可用的。
<b>Edit Parent</b>	和上述一样，父类实例会预载到文本编译器中。
<b>Expand Path</b>	对当前路径进行扇入和扇出的扩展，来对所选对象增加一个额外的层次。
<b>Add Fanin/ Fanout</b>	只可以在 <b>Path Schematic</b> 中使用—显示扇入/扇出对话框，你可以为路径电路图指定参数。
<b>Annotate Values</b>	在 <b>Source/Date/Schematic</b> 窗口中，你可以对当前范围切换注释的开启和关闭。注释会让你看到上下文这显示出的值。例如：在一个资源窗口，注释会在资源文本的下方来显示电路图中的变量。
<b>Properties</b>	对 <b>Schematic Path Schematic</b> 弹出的对话框，为现行的原理图和路径原理图对象显示出所有可用道具。

## 追踪菜单

<b>Trace Assertion</b>	自动追踪断言并在当前或新的波形窗口中显示结果。只有一个断言在当前窗口被选中并且必要条件满足时，此选项才可用。如果一个指定的断言意图被选定，那么这个企图就将被跟踪。如果你选定的是未被指定的意图，DVE 会跟踪第一个意图。断言追踪会给出指定断言的调试详情，当断言失败时，你可以很容易的找到问题的所在。
<b>Assertion Attempts</b>	显示断言意图对话框。如果在当前面板选定的对象是一个断言，那么对话框就会为这个断言显示很多意图信息。如果所选对象不是断言，那么对话框是空的。
<b>Trace Drivers</b>	如果在当前窗口选定的一个变量或是信号的话，此选项会起作用。要想实现其功能，设计必须被其中一个调试选项编译过，并且必须存在一个 <b>mdb</b> 库。找到这个对象的驱动，在 <b>drivers/loads</b> 面板上显示，并还在当前窗口中可重用的资源面板上显示。如果没有窗口显示，DVE 会创建一样新的，如果驱动面板已经存在，那么新的追踪面板信息就加在其

	中。
<b>Trace Loads</b>	除了载入信号外，其他与上相同。
<b>Drivers/Loads</b>	创建一个动态的子选项，你可以对 <b>drivers/loads</b> 显示进行管理。
<b>Follow Signal</b>	在资源面板中，为所选信号创建一个带有信号实例的动态列表。选择一个实例便可进入
<b>Note: 如下选项只针对 Schematic 和 Path Schematic 面板</b>	
<b>Set Current Color</b>	设定 <b>Current Color&gt;- Creates</b> 子选项，你可以设定原理图和路径原理图中网表突出的颜色。
<b>Highlight &gt;</b>	<p>创建一个动态的子选项，使你能够在原理图和路径原理图面板中可以突出和调试被突出的对象。</p> <p><b>Selected-</b> 使用 “<b>Set Current Color</b>”,来改变当前原理图中被选中对象的颜色，或是路径原理图中被选中的颜色。</p> <p><b>Selected by color</b> – 如上所述，只是它提供了一个子选项，你可以在 “<b>Set Current Color</b>”中选择不同的颜色。</p> <p><b>Clear selected</b> – 清除当前窗口中所有被选对象的突出颜色。</p> <p><b>Clear</b> – 如上所述，提供一个已给出颜色的子选项，如果它和所选定的颜色是匹配的，那么它就会被移除。</p> <p><b>Clear All</b> – 清除所有突出的颜色，不管对象是否被选中，还是没有在原理图和路径原理图中运行。</p>
<b>Spot Signal Path</b>	跟随设计中的已选信号
<b>Stop Signal Spotting</b>	停止跟随信号命令。
<b>Trace X</b>	在资源信号中追踪已选信号的 <b>X</b> 变量，认为这个信号是源于 <b>X</b> 值。

## 窗口

下面的条目构成了窗口工具菜单：

图标	描述
 <b>Show Source</b>	打开一个新的源窗格并且显示选中目标的源路径

 <p><b>Show Schematic</b></p>	打开一个新原理图窗格、
 <p><b>Show Path Schematic</b></p>	打开一个新的路径原理图窗格。
 <p><b>Show Wave</b></p>	打开一个新的波形窗格或显示一个以前打开过的窗格
 <p><b>Show List</b></p>	打开一个新的列表窗格或显示一个以前打开过的窗格
 <p><b>Show Memory</b></p>	打开一个新的内存窗格

## 信号

下面的条目构成了信号工具菜单：

图标	描述
 <p><b>Search</b></p>	<p>逆向或正向的箭头为在列表中选中的限定条件进行及时的查找。</p> <p><b>Any Edge</b> –（默认）查找停止并且 C1 位置的光标停在下一个或已经找到的边缘。</p> <p><b>Rising</b> –查找停止并且 C1 位置的光标在下一个或已经找到的上升沿。</p> <p><b>Falling</b> –查找停止并且 C1 位置的光标在下一个或已经找到的上升沿。</p> <p><b>Failure</b> –仅是可访问的如果信号是一个断言；停止在下一个或先前的断言故障处</p> <p><b>Success</b> –仅是可访问的如果信号是一个断言；停止在下一个或先前的断言触发处</p> <p><b>Signal Value...</b>-显示一个小的对话框允许你输入一个指定的值作为约束条件。如果该值被找到，查找停止并且 C1 位置会采该信号的输入值。该值必须符合为该信号当前选定的基数。</p>

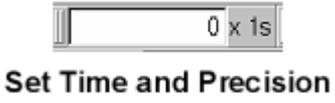
## 模拟器

下面的条目构成了模拟器的工具菜单：

图标	描述
 Start/Continue	运行仿真程序直到一个中断点（breakpoint）被触发，或仿真结束，或到了该期间在对话框或者是工具栏指定的持续时间。。
 Run for Specified Time	运行指定时间的仿真后停止。
 Stop	当运行仿真的时候该图标被激活，点击它停止仿真
 Next	对于 VHDL、Verilog、和 TB 代码在任务或函数层上进行下一步。
 Step In	每一行代码的向前进行仿真程序，与代码的语言无关。这与 UCLI 的 <b>Step</b> 命令是功能一致的
 Step In Active Thread	在当前激活的线程中一步一步地执行到下一个可执行行
 Step In Any Testbench Thread	对于 NTB OpenVera 和 SystemVerilog 测试平台来说在测试平台的下一个可执行行处停止
 Step Out	执行到下一个可执行行时退出当前的函数或任务
 Restart	停止当前运行的仿真并且用当前的仿真设置重启它。该操作保留所有打开的窗口和 GUI 设置。如果仿真没有运行该操作启动仿真

## 时间操作

下面的条目构成了时间操作工具菜单:

图标	描述
	<p>显示 C1 光标当前的时间。通过在这个框里输入一个新时间来设置当前的时间。为显示仿真数据提供显示时间单位。选择 <b>View &gt; Set Time Scale</b> 来设置时间单位和精度。</p>

## 扩大或缩小

下面条目构成了扩大或缩小工具菜单:

图标	描述
	<p>用来视图查看原理图。改变鼠标的动作左键单击且拖拽选择目标。左键单击一个单独的目标选定它。<b>Ctrl</b> 加鼠标左键添加目标到已选项中。按住左键后拖拽创建了一个框所有在框中的目标将被选中。</p>
	<p><b>Zoom In</b>-让所选的区域填满窗格 <b>Zoom Out</b>-让当前视图查看的目标适合所选区域</p>
	<p>用来视图查看原理图。变鼠标的行为按住左键且拖拽移动所有原理图中的目标的位置。例如按住左键向下拖拽, 将在视线里的所有目标下移。选择这个菜单条目同时将光标转换成了手型。点击工具菜单选择将光标转换成常规模式</p>
	<p><b>Zoom Full</b>-将所有可访问目标转换成当前视图查看。 <b>Zoom In 2x</b>-使当前视图查看的目标变成 2 倍大小因此会看见更少的目标。 <b>Zoom Out 1/2</b>-使当前视图查看的目标变成 1/2 大小因此会看见更多的目标。 <b>Zoom Fit Selection</b>-使当前视图查看的窗口处于所选目标原理图设置的中心并且所有选中的目标在当前窗口中是可访问的</p>

## 扩大或缩小和变动记录

下面的条目构成了扩大或缩小和变动记录工具栏：

图标	描述
 <p>Go to Zoom and Pan Settings</p>	<p><b>Back in Zoom and Pan History</b>-回指针通过保存扩大或缩小和变动记录重复视图查看当前窗格。当你在一个窗口进行扩大或缩小或变动的时候 DVE 存储先前的视图查看页面因此你可以检索它。</p> <p><b>Forward in Zoom and Pan History</b>-如果你已回到先前视图查看记录，向前指针会提供一个简单方法来下一个视图查看页面。点击这个条目会最终回到当前视图查看页面</p> <p><b>Named Zoom and Pan Settings</b>-跳转一个保存的已命名的视图查看界面选择下拉按钮。</p>

## 上下文记录命令菜单的应用

在任何窗口，右键点击就会显示出一个上下文记录菜单，然后选择一个命令。该层次化得导入上下文记录菜单就如下所示：

命令	描述
<b>Copy</b>	复制选定文本
<b>Add to Waves</b>	将选定得一个或多个信号在波形窗口中显示
<b>Add to Lists</b>	将选定得一个或多个信号在列表窗口中显示
<b>Show Source</b>	将源代码显示在源窗口中选定的作用域
<b>Expand By Levels</b>	可以通过一步操作在不同级别展开
<b>Expand All</b>	立刻展开所有层。当交互得进行工作时从仿真得到层次属性可能需要一些延迟
<b>Collapse</b>	折叠选中的区域
<b>Collapse All</b>	折叠所有展开得区域
<b>Select Scope By Levels</b>	允许通过层选择范围。你可以同时选择多个层次。
<b>Select All</b>	选择所有可见得层次

# 设置显示的首选项（Preferences）

你能够设置首选项来自定义 DVE 窗口和窗格。

## 自定义显示

在顶层窗口中，选择 **Edit > Preferences**。

应用程序首选项选择对话框显示了全局设置分类。

如下的选择设置：

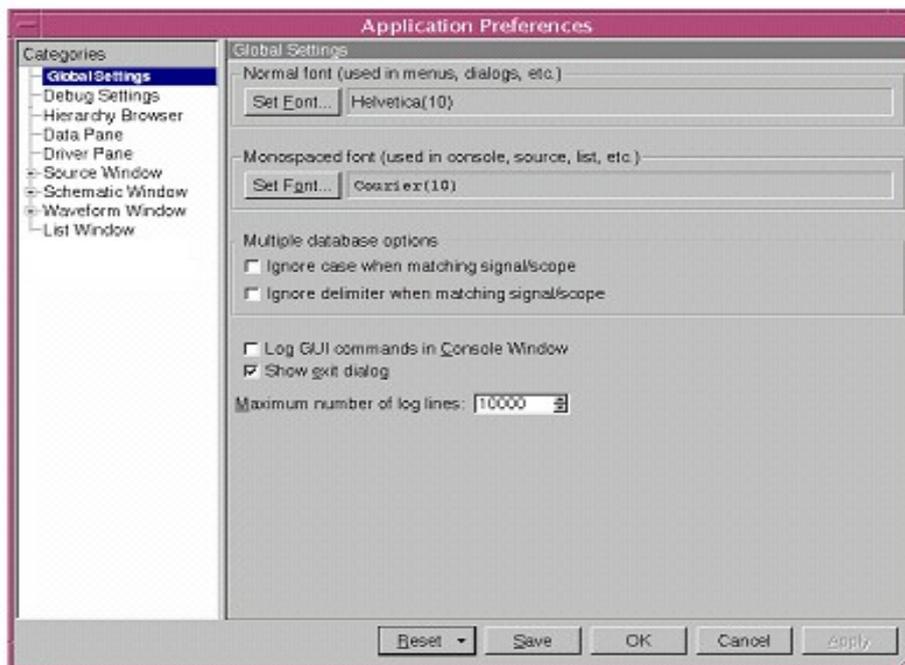
- 全局设置

选择设置显示在 DVE 窗口中得字体和字体大小

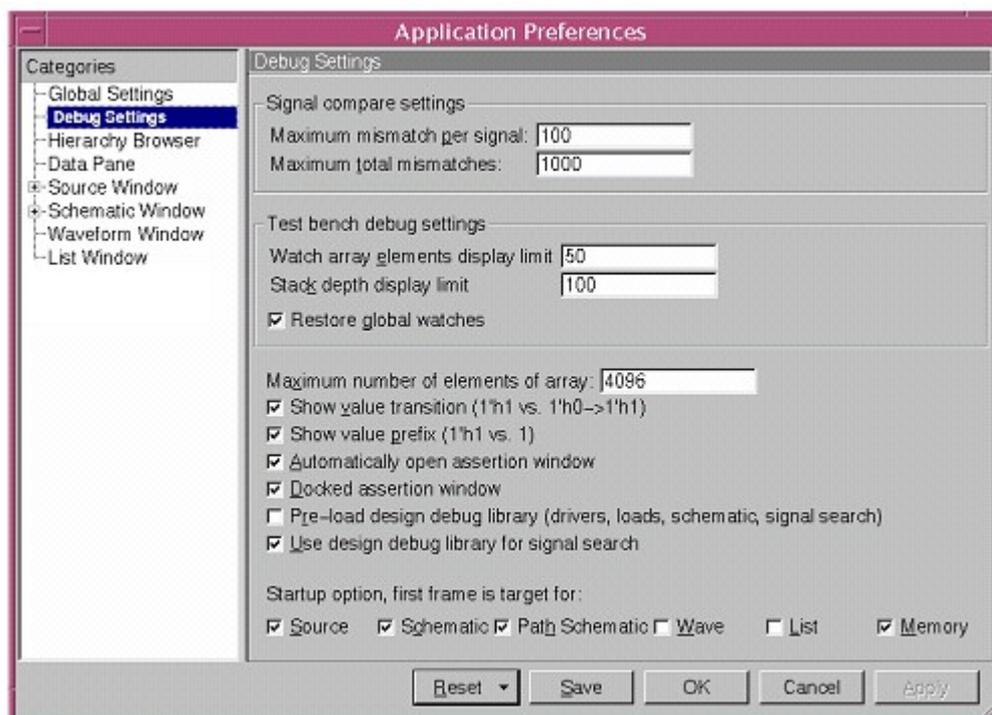
指定混合数据库选项通过选择是否忽略条件或在分配信号时候选择一个定界符或范围。

默认的日志只是显示 UCLI 命令。为了使 GUI 命令也在日志中显示在控制台窗口检查框选择 GUI 日志命令。

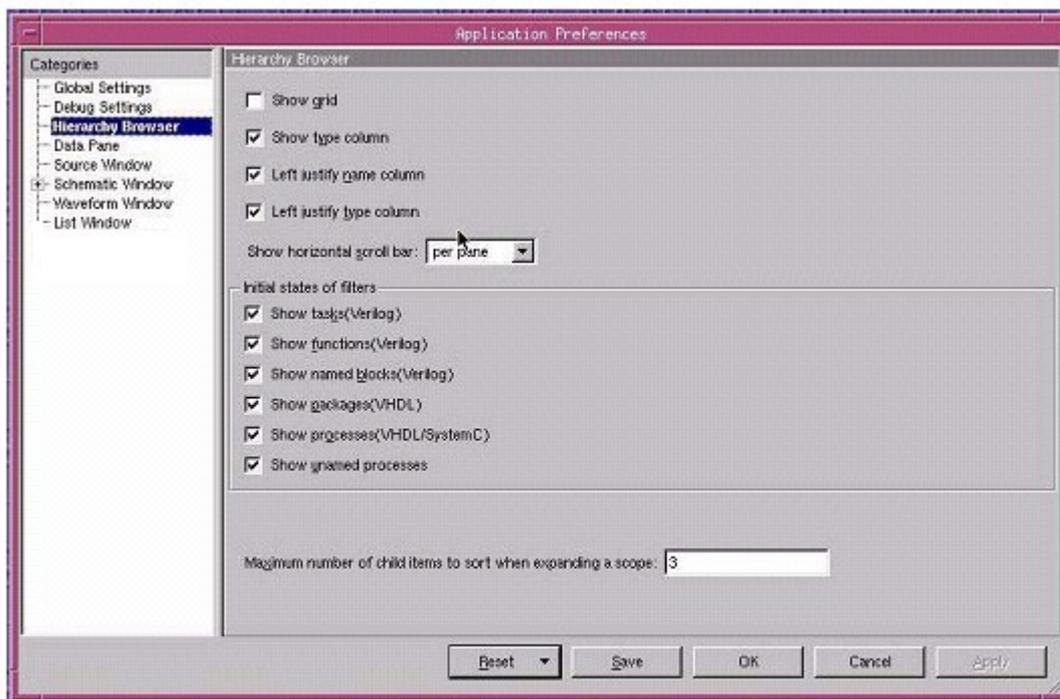
选择是否显示退出对话框在关闭 DVE 时候。指定了日志文件最大得显示行数。



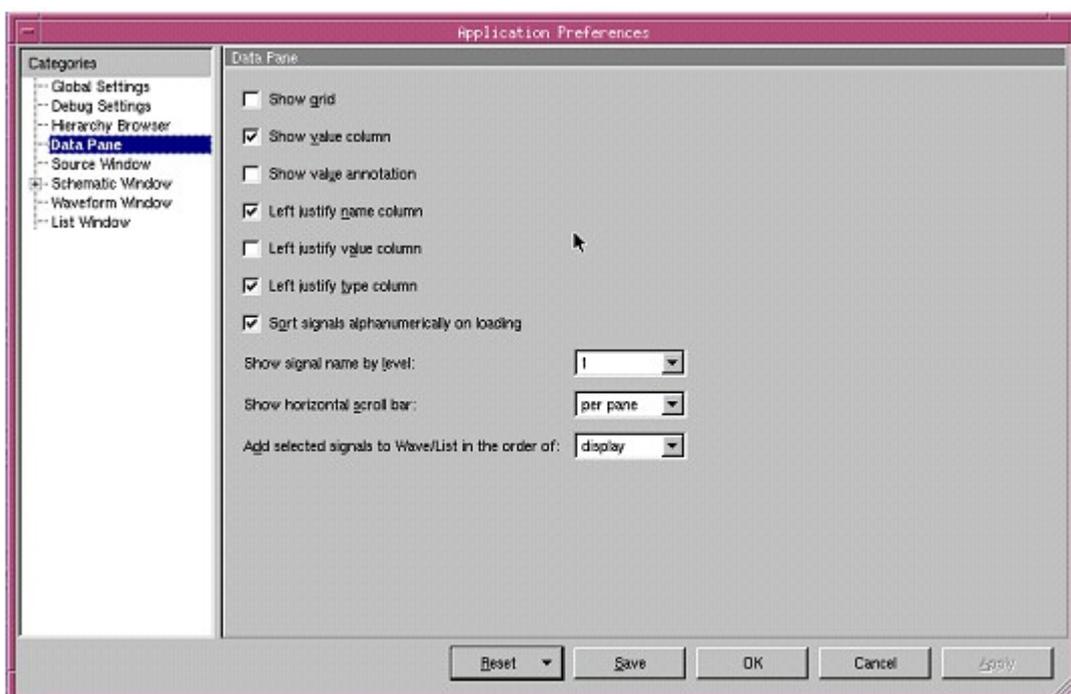
- 调试设置-选择信号对照参数，值过度，和断言窗口插入默认项，和第一个目标结构设置选项。



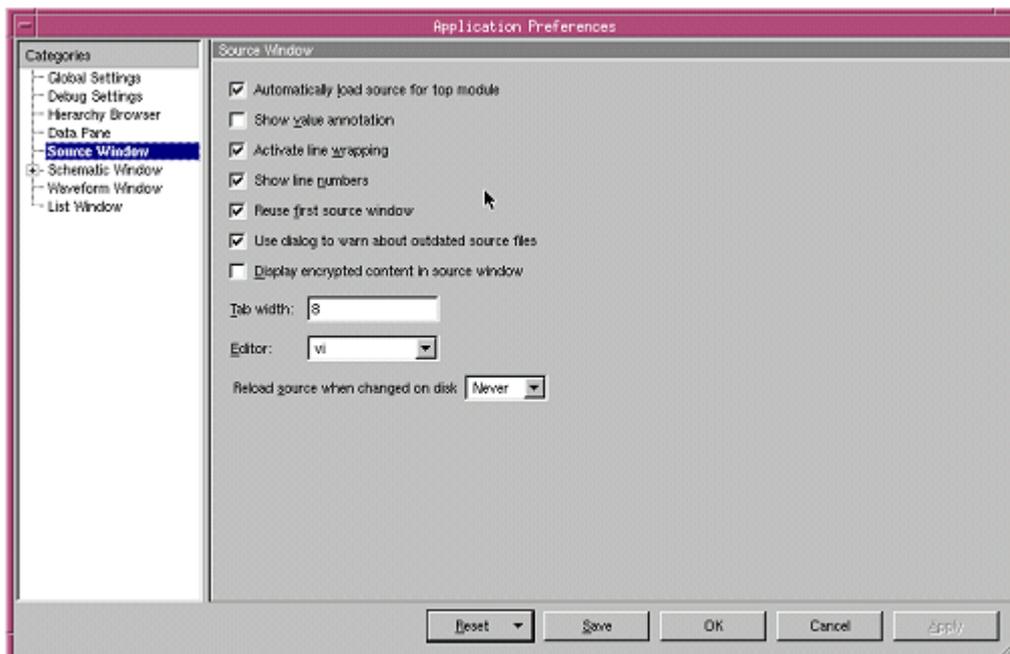
## 等级浏览器-设置外观和初始化过滤器状态



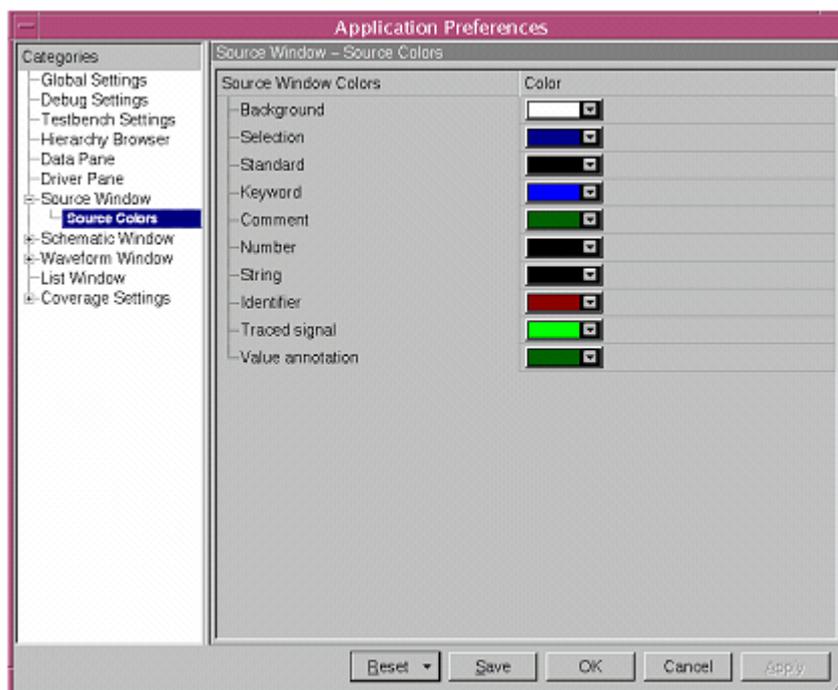
## 数据窗格-设置外观参数、信号排序、信号显示层次、和滚动条条件



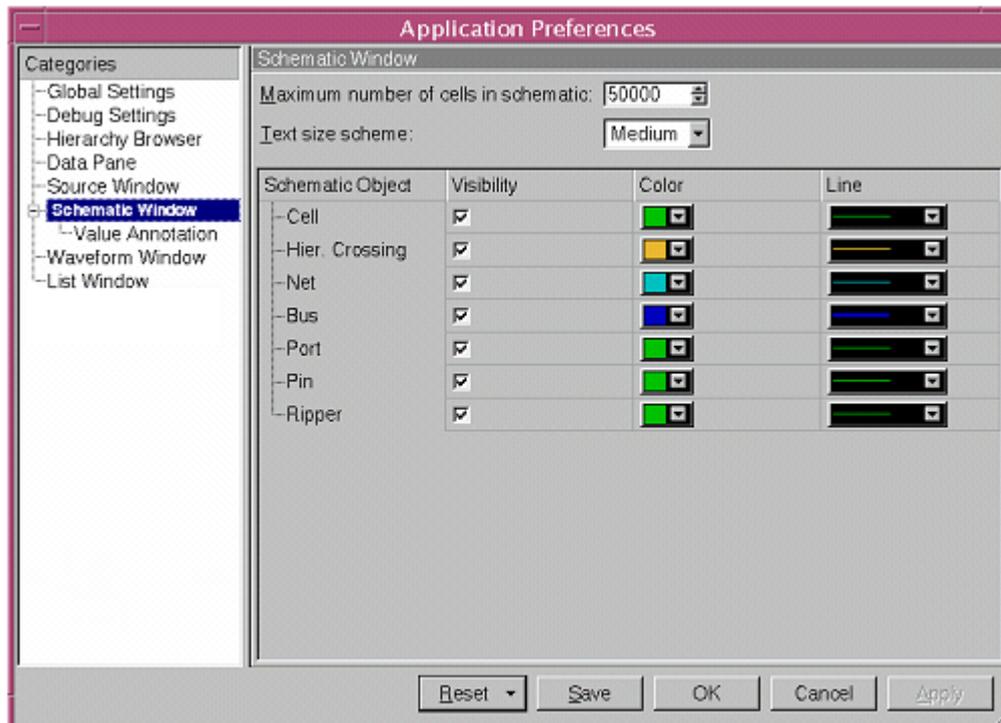
源窗口-指定了数据和注释载入选项、行总结、行数显示、标签宽度、默认编辑器、和自动载入改变后得源代码



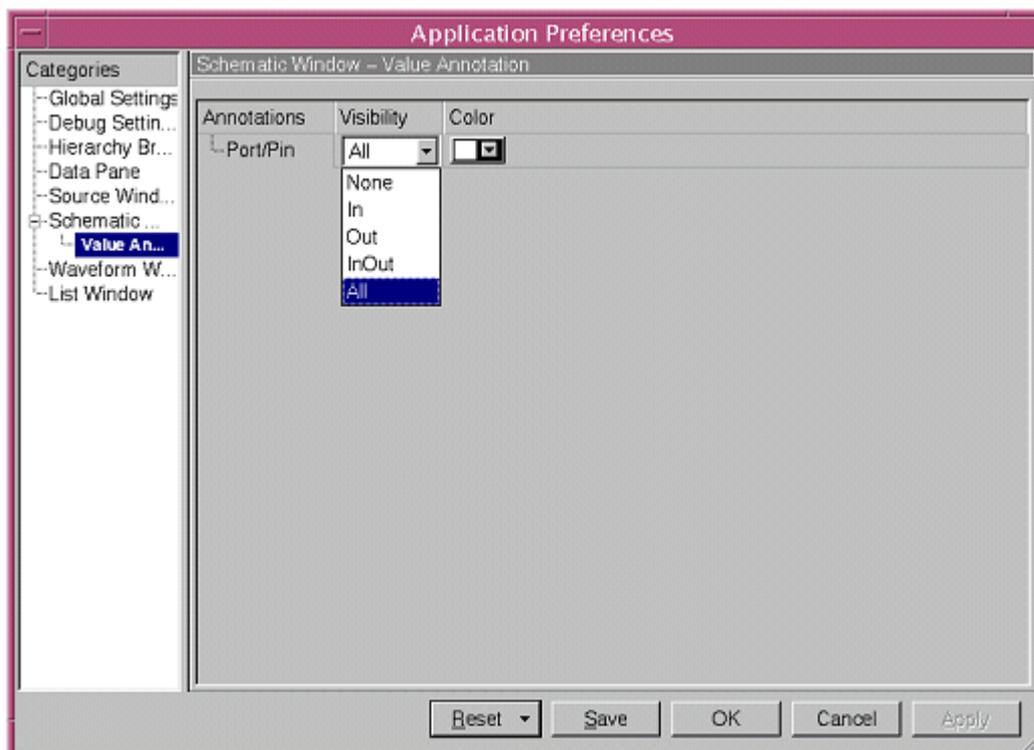
源颜色分类-指定了显示颜色的源窗口部件通过点击下拉条并且选中一个颜色



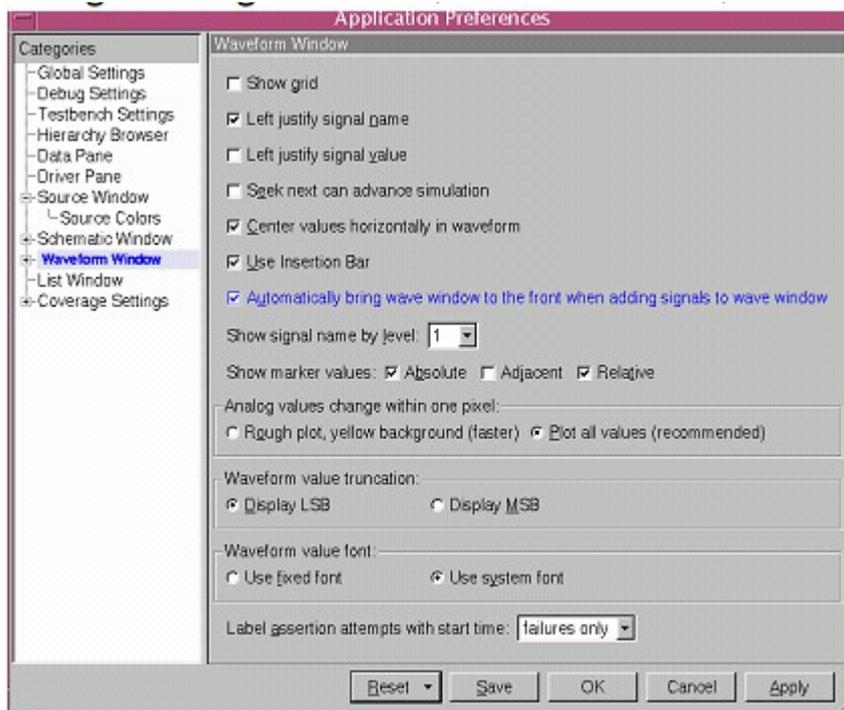
## 原理图窗口-设置目标原理图的行颜色在原理图和路径原理图窗口



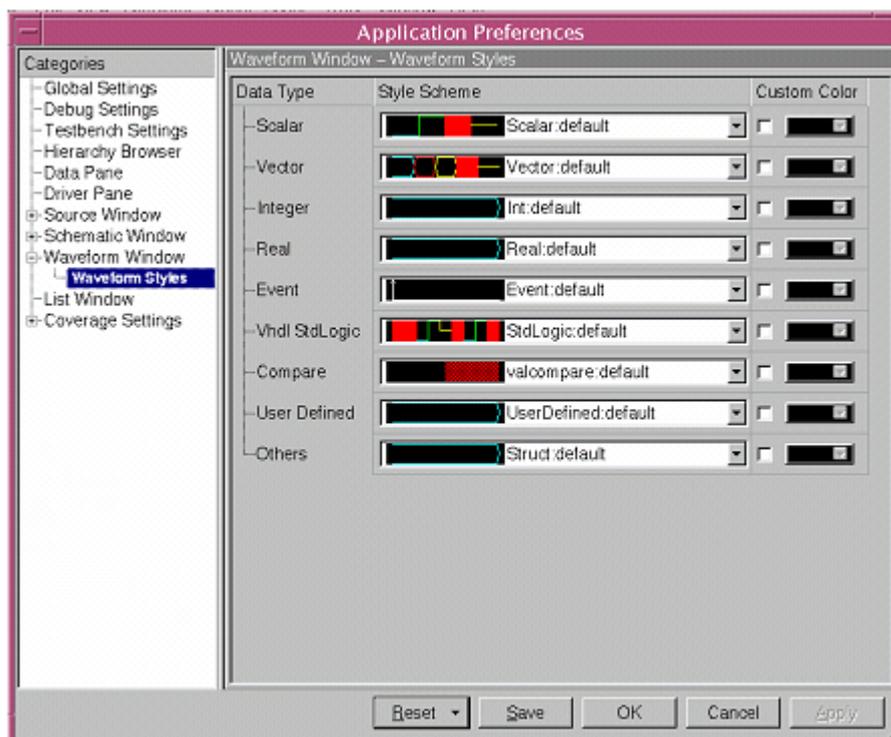
## 原理图值注释分类-设置了端口/管脚的可见度和颜色



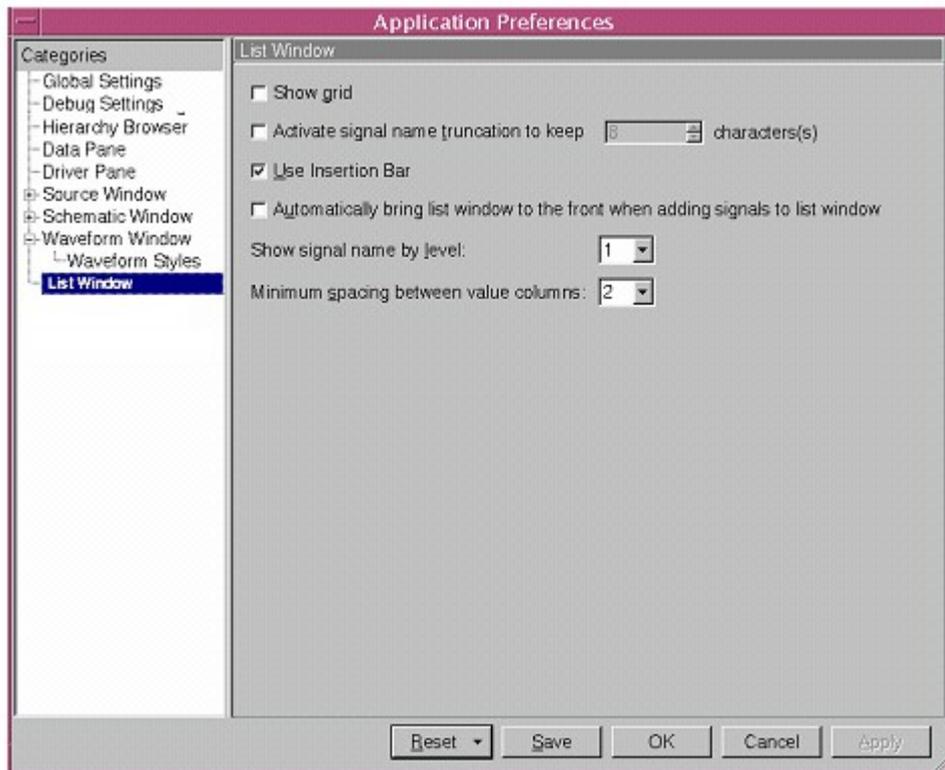
## 波形窗口-设置外观参数和信号层、标记值和波形值的显示设置



## 波形风格分类-为数据类型选择波形显示风格



列表窗口-指定了网格显示, 信号名截断, 显示信号层, 和列空间设置



3、点击 **OK** 保存你的选择并且关闭对话框, **Save** 保存你的设置并且保持对话框打开, 或 **Reset** 回到默认的设置。

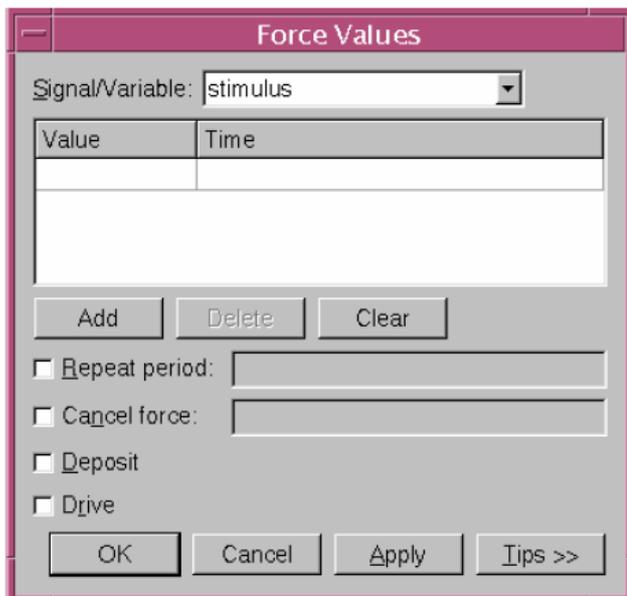
## 3 使用层次数据和信号群组窗格

### 层次窗格

## 给信号赋值

在 Force Values 对话框中，按照赋值标准对信号或者变量进行赋值。

Figure 3-7 Force Values dialog box.



- 1、选择 **Simulator > Add Force** 显示添加赋值对话框。
- 2、在信号/变量区域，指定将要被赋值的信号或是变量。
- 3、在表格中，指定值的时序，值与时间是成对的。
  - 时间精度在时间表达式中是可选的，默认为工具中的时间精度。
  - @符号是可选的，表示后面的时间表达式和仿真的开始是相对的。如果它被省略，时间表达式被认为是同当前仿真时间相对的。
- 4、单击 **Add** 或 **Delete** 来添加或删除表格中的行。
- 5、使用 **Repeat period** 指定一个循环的延迟，之后赋值的顺序必须被重新启动。
- 6、在 **Cancel force** 区域为赋值命令设定一个注销时间。
- 7、单击 **OK** 应用你的赋值规范然后关闭对话框，**Apply** 能应用你的规范并让你的对话框保持打开状态，**Cancel** 是关闭对话框而不应用你的规范。

## 4 使用源窗口

### 管理交互仿真中的断点

DVE 在交互仿真中逐步运行或整体运行时允许设置引起工具停止的断点:

- 仿真中每次执行一个指定的行时行断点都会到达（看显示行属性的区域，在对话框中管理断点会设置更多的信息）。你也可以指定一个实例让该工具只停在指定实例的行上。
- 时间断点停止在仿真中一个指定的绝对或相对时间点上。
- 信号断点当遇到一个指定信号上升，下降或改变时才触发。
- 断言断点停止在一个指定的断言事件上。
- 任务/函数断点停止在指定的任务或函数上。

### 控制源窗口中的行断点

在源窗口属性区域你可以用鼠标控制行断点

- 为了设置一个断点，在源窗口属性区挨着一个可执行行的地方左键单击。一个实心的红圆显示出来意味着一个行断点被设置好了。  
注意：一个行断点只能被设置在一个可执行行的位置。如果一个行不是可执行的，右键挨着它单击时没有断点会被设置。
- 右键单击源窗口的属性区，然后从上下文菜单中选择 Set Breakpoint。一个实心的红圆显示出意味着一个行断点设置好了。
- 为了取消一个行断点，左键单击这个实心的红圆。这个圆不再是实心的，表示这个断点被取消。
- 右键单击一个有效或无效的断点，然后选择 **Disable Breakpoint** 或者 **Enable Breakpoint**, **Delete Breakpoint** 或者 **Delect All Breakpoint** .你也可以在

属性区域删除任何地方的断点。

下面的表格描述了使用属性区来控制断点：

断点图标	描述	行为
 有效断点	显示一个行断点被设置在这个行上它是有效的。行逐步运行，仿真器将会停在这个行上。	单击红圆使行断点无效
 无效断点	显示一个行断点被设置在这个行上它是无效的。行逐步运行，仿真器将不会停在这个行上。	单击红圆删除它

## 在对话框中管理断点

你能在断点的对话框中管理所有类型的断点。

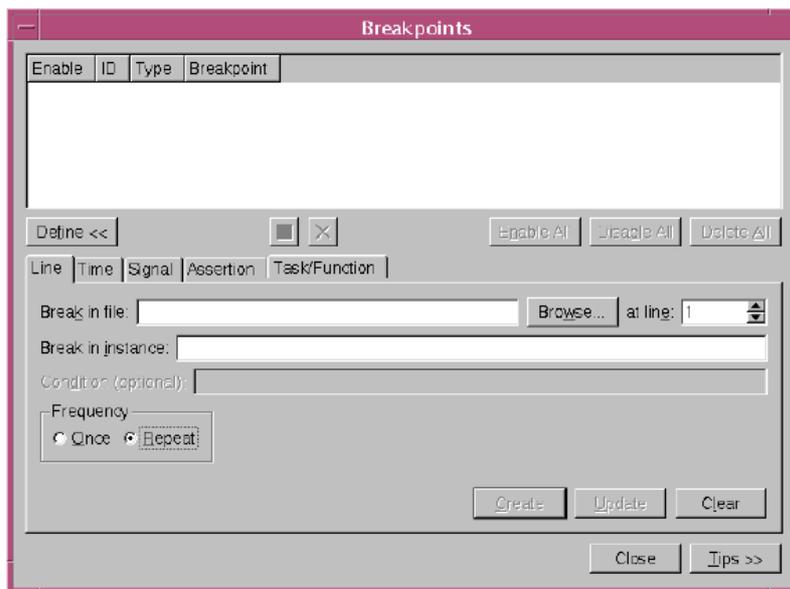
## 行断点

你能在交互仿真中使用断点的对话框来创建断点。

创建一个行断点需执行下列操作：

- 1、 右键单击源窗口行属性区或者下拉仿真器菜单，然后选择 **Set Breakpoint** 来显示断点对话框（图示 4-8）。

Figure 4-8 创建断点



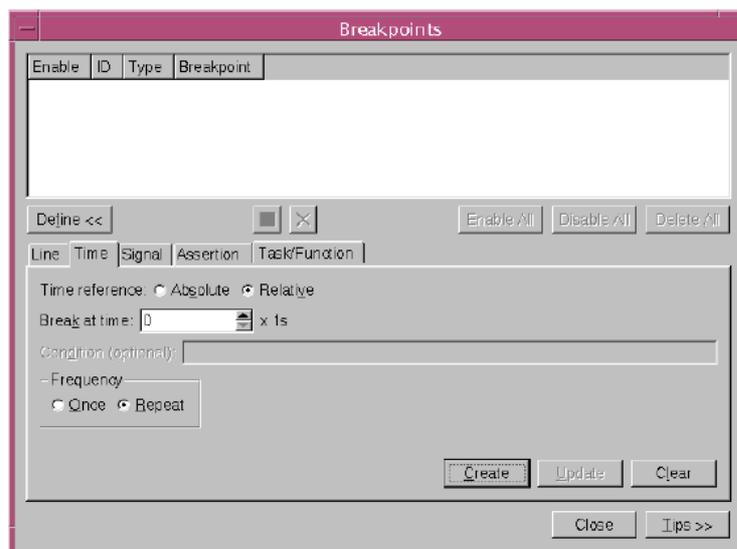
- 2、行制表被选择后，输入文件名字，或是浏览你想要创建的断点的文件。
- 3、为断点输入行号。
- 4、输入断点将会激活的实例。
- 5、如果你想要断点只激活一次，在频率选项中选择 **Once**，否则选择 **Repeat**。
- 6、你可以有选择的为 VHDL 对象输入一个条件使断点激活。  
注意该条件不被 Verilog 对象支持。
- 7、单击 **Create**。  
断点被创建并出现在断点列表框中。

## 时间断点

创建一个时间断点需执行下列操作：

- 1、右键单击源窗口的行属性区或下拉仿真器的菜单，然后选择 **Breakpoints** 来显示断点对话框（图示 4-9）。
- 2、单击 **Define** 显示断点创建制表。
- 3、选择 **Time** tab(图示 4-8)。
- 4、在时间制表里，选择绝对或相对时间，然后输入在什么时间设置断点（图示 4-9）。

Figure 4-9 断点对话框时间制表



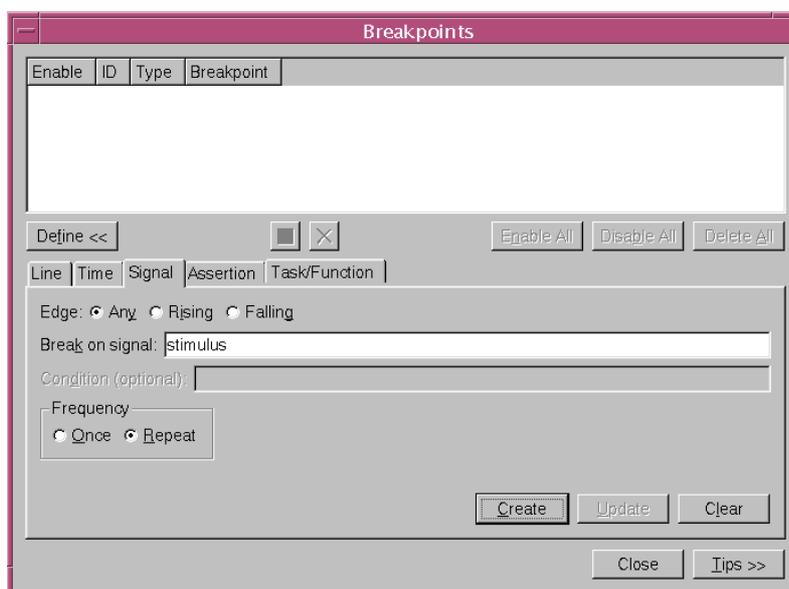
- 5、你可以有选择的为 VHDL 对象输入一个条件使断点激活。  
注意该条件不被 Verilog 对象支持。
- 6、单击 **Creat**.  
断点已被创建并出现在断点列表框里。

## 信号断点

创建一个信号断点需执行下列操作：

- 1□ 右键单击源窗口的行属性区或者下拉仿真器的菜单，然后选择 **Breakpoint** 显示断点对话框（图示 4-10）。
- 2□ 单击 **Define** 显示断点创建制表。
- 3□ 选择 **Signal** 制表。
- 4□ 在信号制表里，在 Break on signal 文本中输入想要的信号。（图示 4-10）

Figure 4-10 断点对话框信号制表



- 5□ 选择 Any, Rising 或者 Falling Edge 定义断点事件。
- 6□ 指定 Frequency。
- 7、你可以有选择的为 VHDL 对象输入一个条件使断点激活。注意该条件不被 Verilog 对象支持。
- 8、单击 **Creat**.  
断点被创建并出现在断点列表框中。

## 断言断点

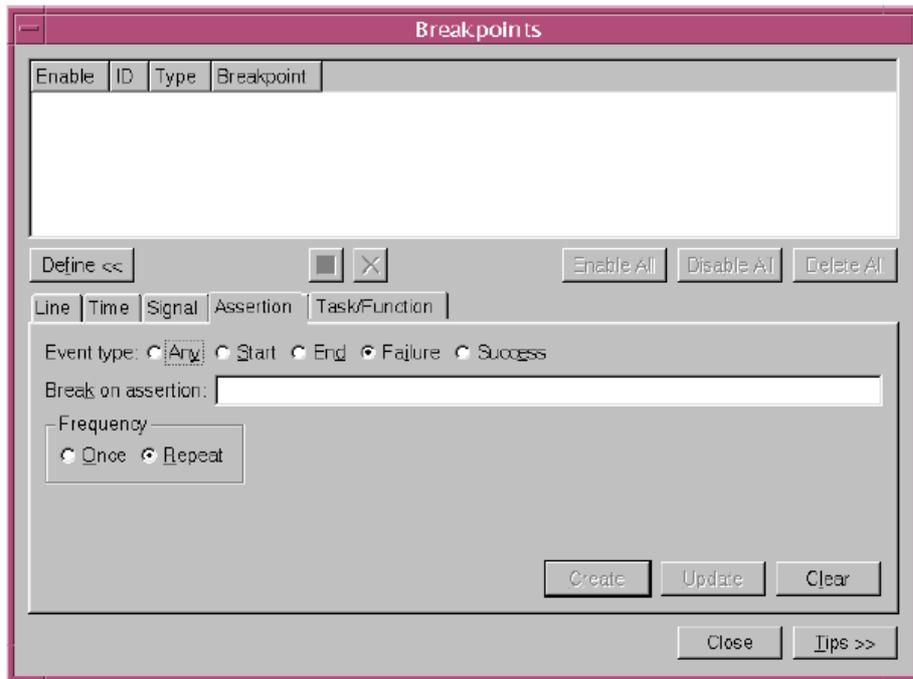
创建一个断言断点需执行下列操作：

- 1、右键单击源窗口的行属性区或者下拉仿真器的菜单，然后选择 **Breakpoints**

显示断点对话框。(图示 4-11)

- 2、单击 **Define** 显示断点创建制表。
- 3、选择 **Assertion** 制表。
- 4、在 **Break on assertion** 文本区域中给断言输入全路径。(图示 4-10)

Figure 4-11 断点对话框断言制表



5、从 **Any**, **Start**, **End**, **Failure** 或 **Success** 中选择一个要触发断点的事件类型。

6、单击 **Creat**。

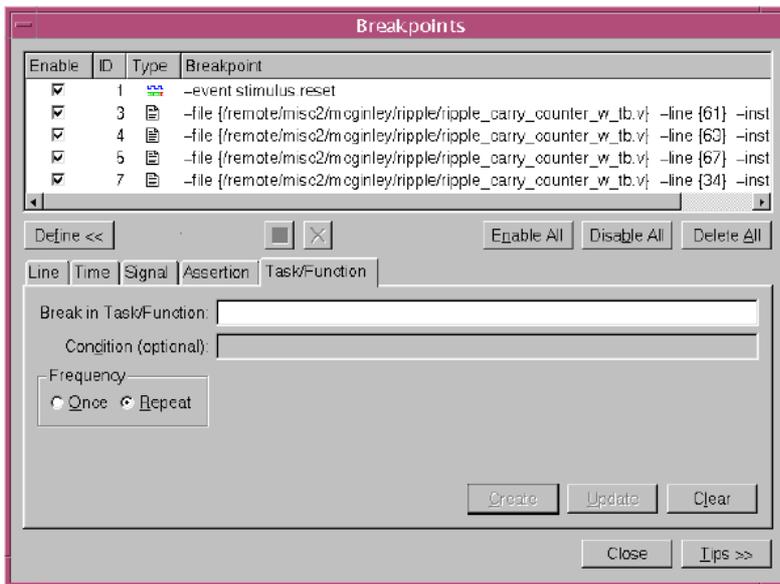
断点被创建并出现在断点列表框中。

## 任务/函数断点

创建一个任务/函数断点需执行下列操作：

- 1、右键单击源窗口行属性区或是下拉仿真器的菜单，然后选择 **Breadpoints** 显示断点对话框(图示 4-11)。
- 2、单击 **Define** 显示断点创建制表。
- 3、选择 **Task/Function** 制表...
- 4、在 **Break in Task/Function** 区域中给任务或函数输入全路径 (图示 4-12)。

Figure 4-12 断点对话框断言制表



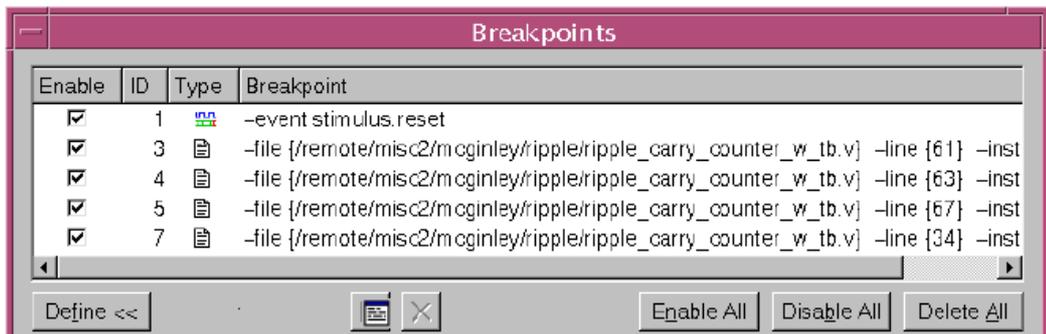
- 5、你可以有选择的为 VHDL 对象输入一个条件使断点激活。注意该条件不被 Verilog 对象支持。
- 6、选择 **Once** 或 **Repeat** 的频率。
- 7、单击 **Create**。  
断点创建好了并出现在断点列表框中。

## 编辑断点

编辑断点需执行下列操作：

- 1、在源窗口行属性区或从仿真器菜单中，右键单击并选择 **SetBreakpoints** 显示断点对话框（图示 4-13）。

Figure 4-13 断点对话框



## 2、操作定义的断点需执行下列操作：

- 为了使断点有效或无效，选择或取消 Enable 框或选择列表中的断点单击 **EnableAll** 或 **DisableAll**。

- 为了删除一个断点，选中列表中的断点并单击 或单击 DeleteAll 删除所有的断点。

- 查看断点中的源代码需单击 。

# 5 运用波形窗口

## 建立目标窗口

默认设置是在一个新的顶层窗口中打开波形。你可以接受默认模式或者使用在状态栏里的目标窗口控制，在当前的顶层窗口中将波形显示出来。

目标窗口管理按以下切换的方式完成新顶层窗口的创建：

- 无目标（默认）- 当波形图标没有箭头右上角的小对号标志时，如图 5-1，就会创建包含波形窗的新顶层窗口。

图像 5-1 目标窗口控制



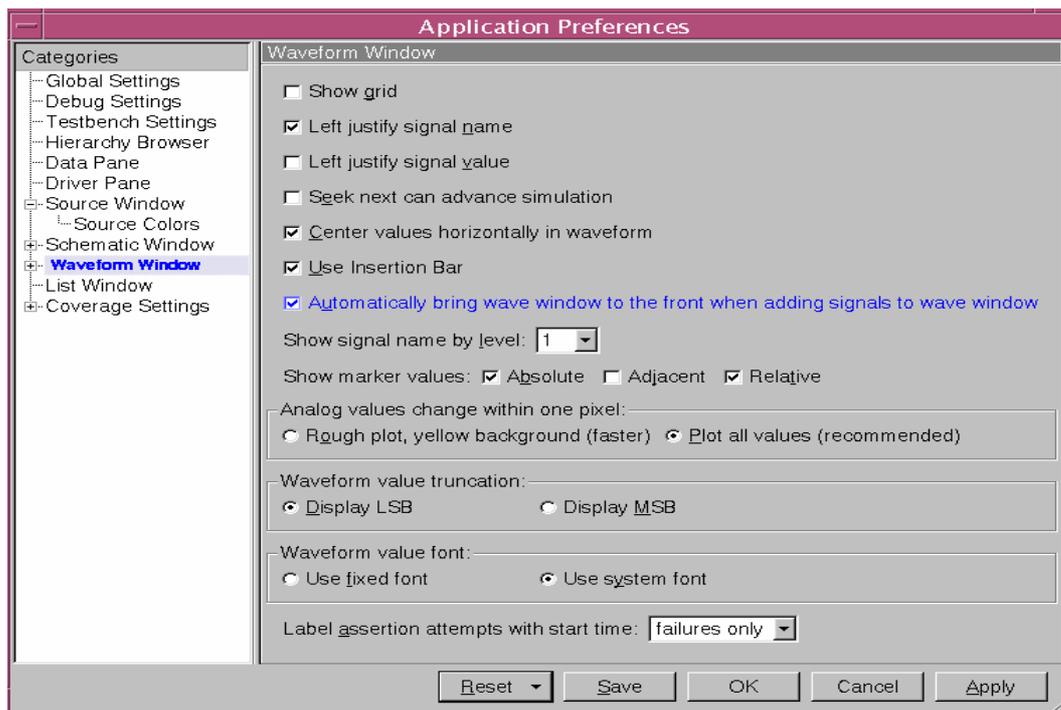
- 有目标 - 当图标有箭头时, 如图 , 就会在当前顶层窗口框中创建一个新波形窗。  
如果在当前顶层窗口中打开一个波形。你可以在菜单选择中 **windows>undock** 来改变顶层窗口的显示。

## 设置波形窗口首选项

使用应用首选项对话框**制定设置**信号, 标记和波形的显示方式。

- 1、选择 **edit>preferences**。  
显示应用首选项对话框。
- 2、在类别窗中选择 **waveform window**。

图像 5-3 波形应用首选项对话框



按以下设置首选项:

显示坐标—按级别显示信号名并为信号窗对齐首选项。这些设置与数据窗首选项很相似。

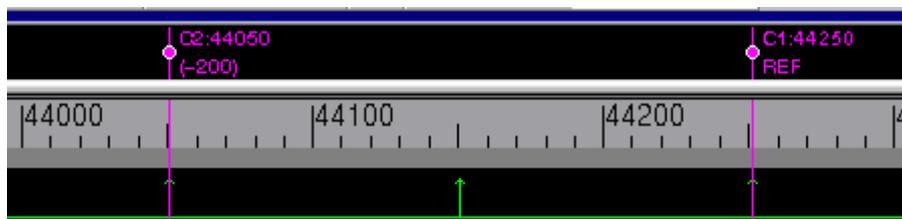
寻找下一个可以提高仿真—只对交互式仿真有效。为了完成要求的寻找, 带有寻找下一个值或沿的控制管理可以功能性地提高仿真。

显示标记值—标记可以显示 1、2 或 3 个值。这些首选项控制值 (可以是任何组合或所有) 的显示。

绝对的一标记被设定的时间。

相邻的一在标记与他最近标记之间每一边上的时间。

相对的一参照标记的时间（通常为 C1），如下。



~~模拟值改变不超过一个像素—选择要标出的值的数目。~~

~~Rough plot 可以加速性能，plot all values 会比较慢。~~

~~Rough plot—平缓地标出值。~~

~~Plot all values—显示所有，这个选择会比较慢。~~

~~Label assertion attempts with start time—选择用断言开始时间注释断言的试图波形（向上箭头）。在一个单元里能够看到开始和结束的时间，这么做是对于在一个位置看时和结束的时间很有用，但如果是一个短时期断言的情况下就会给波形显示添乱。~~

~~failures—注释失败。~~

~~always—显示所有断言的开始和结束时间。~~

~~never—没有注释断言。~~

~~点击 apply 来查看你的更改并保持对话框打开，或点击~~

~~OK 来应用你的改变并关闭对话框。~~

点击 **reset** 重新应用默认模式。

使用信号窗

信号窗显示以组的形式显示信号：

数量信号的值以 2 进制格式显示

矢量信号的值以 16 进制格式显示

整型、实数和时间以浮点格式显示

信号窗的组成部分：

在波形窗口中允许信号的过滤显示

名字栏显示信号名。

数值栏显示在仿真时被 C1 光标选中的信号值。

以图 5-4 信号窗的为例

—

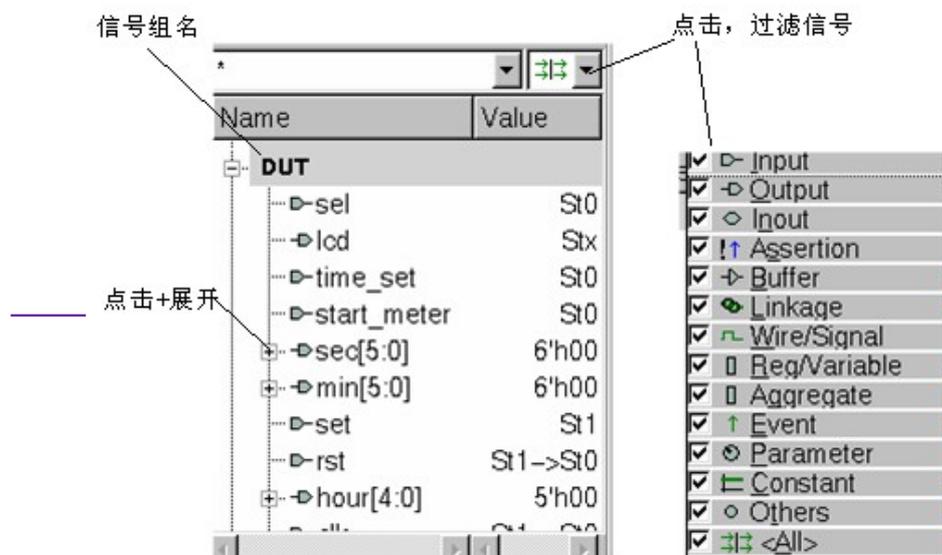


图 5-4 信号窗

## 展开 verilog 矢量，整型，时间和实数

点击在信号名左边的加号图标  可以将矢量信号扩展到单个字节。在你扩展显示之后，每一位都被添加到信号窗内，并且每一位的波形也被添加到波形窗口当中。

DVE 以 32 位代表整型，所以你可以在信号窗中显示它每一位的波形。同样，DVE 以 64 位代表时间数据，所以你可以以 64 个单独波形显示时间。

不能扩展实数这种数据类型。

~~断言也可以被扩展。在扩展断言之上，他的子系将包括那些组成断言的断言时钟，信号和事件（队列和 SVA 性能）。~~

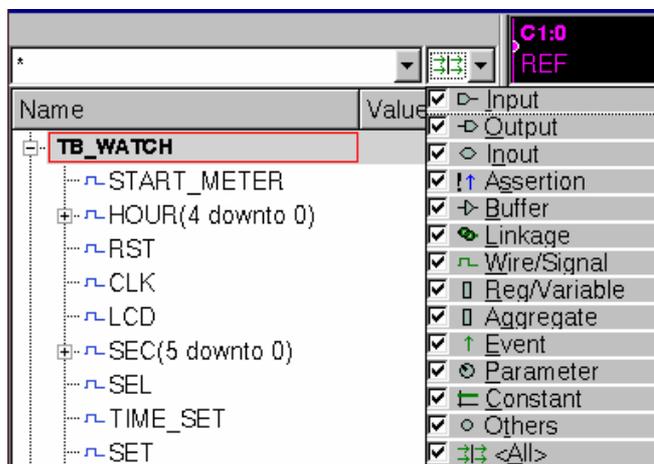
## 重命名信号组

重命名信号组。在层次窗里的信号上左击选择信号组，然后再次左击它的名字，就可以输入新的信号名。

## 过滤信号

用信号窗右上的菜单，将在信号窗中的信号过滤显示。撤销选择[这个相关](#)信号会将它隐藏，选择检验栏可以再次把它显示出来。

图 5-5 过滤信号显示菜单



## 添加信号分离器

你可以通过选择 **signal>insert divider** 在信号之间加入分离器，在波形窗口中将信号分离。只要你喜欢，你可以在信号之间添加很多个分离器。

分配器可以插入到在波形窗口内打开的所有实例的信号组里。分离器被保存在对话 TCL 文件里，当对话打开时，分离器被恢复。

## 设置复制信号的显示

当显示复制信号时，你可以在不影响任何复制品显示的方式的前提下设置 **被复制信号信号实例** 的显示方式。

选择一个信号，然后将 **signal>default properties** 关掉。

选择 **signal>properties** 并改变信号的方案，颜色。注意：你也菜单里右击选择 **properties**。

对被选信号的改变不会影响复制品的显示。

注意：如果你没有关闭默认功能，所做的改变将变成默认模式并且复制信号的显示也会改变。

如果同一个实例中的一个信号组在两个波形窗口中，改变一个信号，另一个窗口中的这个信号也会改变。

使用用户定义基数

本章将描述如何创建，编辑，**传入和传出导入和导出**用户定义基数。

## 创建一个用户定义基数

你在波形窗口中可以定义一个从值到字符串的 **习惯记忆的映射通俗易懂的表达方式**。

- 1、选择 **signal—set radix—user defined—edit** 显示编辑用户定义基数对话框。
- 2、点击 **new**，输入一个基数名字，然后按 **Return**，便可以创建一个用户定义基数。
- 3、点击 **add row** 为用户定义基数激活一行，为每一行的 **输入项目** 选择文本和背景颜色。



- 4、点击 **OK** 或者 **apply** 来存储用户定义基数。

- 5、给一个信号应用用户定义基数，在波形窗中选择信号，选择 **signal>set**

**radix**，然后从列表中选择用户定义基数。

## 管理用户定义基数

编辑或删除一个用户定义基数：

- 1、选择 **signal—set radix—user defined—edit** 显示编辑用户定义基数对话框。
- 2、删除一个用户基数，从用户定义基数列表中选择基数，然后点击 **Delete**。  
编辑一个用户基数，从用户定义基数列表中选择基数，在值或显示表格中点击一个单元，然后加入你的更改。
- 3、点击 **OK** 或 **apply** 储存更改。

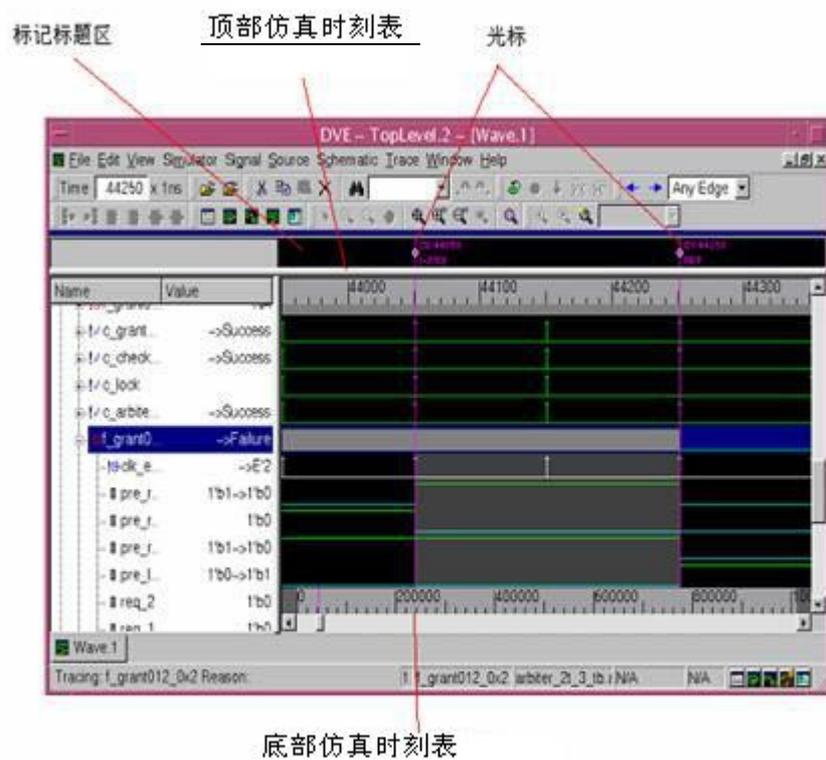
### 输入输出导入和导出一个用户定义基数

- 1、选择 **signal>set radix>user defined>edit** 显示编辑用户定义基数对话框。
- 2、**输入导入**基数点击 **inport**，然后浏览并选择期望的基数。  
**输出导出**基数点击 **export**，从用户定义基数目录中选择基数，然后输入一个基数名。
- 3、点击 **OK** 或 **apply**。

## 运用波形窗

波形窗显示**过度信号值值的改变**和断言成败。

图 5-6 波形窗口

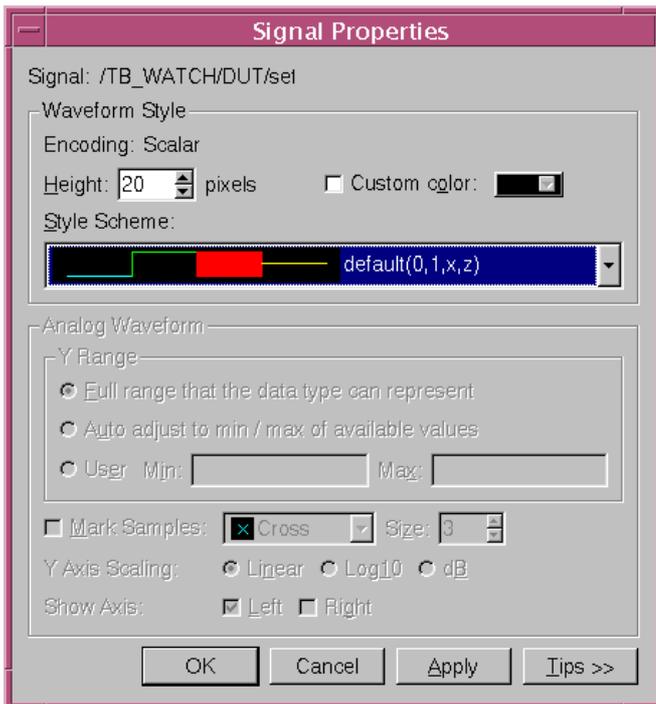


[光标指针](#)和标记将在 4-18 中的“[光标指针与标记](#)”中介绍。

波形窗口有较高和较低的时间精度顶部和底部的仿真时刻表。在波形窗口中，较高时间精度波形窗口顶部的仿真时刻表显示目前仿真的时间域，较低时间精度波形窗口底部的仿真时刻表显示通过整个仿真的时间域。

## 制定波形显示

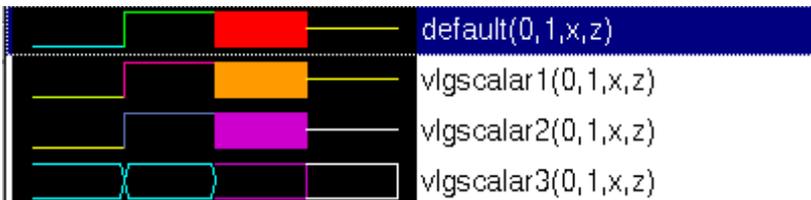
- 1、在波形窗口中选择一个信号，然后选择 **signal>properties** 或者从菜单中右击选择 **properties** 显示出信号性能属性对话框。



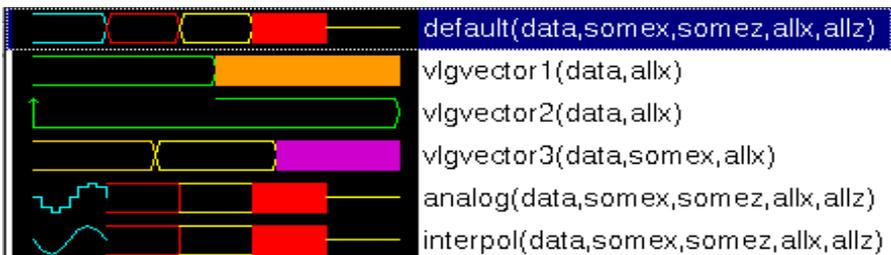
2、在波形种类区域中，给波形制定高度和颜色。

3、设置示意图风格种类图标。

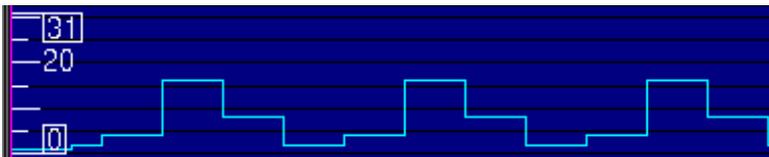
- 对于计数波形，点击箭头并按下图所示从计数图标中进行选择。



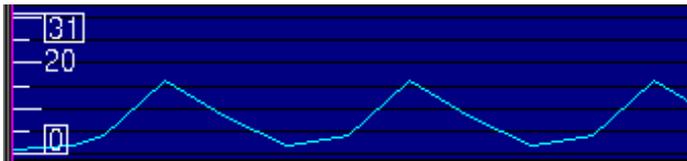
- 对于模拟波形，点击箭头并按下图所示从向量图标，模拟图标或内插图中标中选择。



**analog** 选项的显示方式是像楼梯一样的模拟波形，它会一直保持原值，直到下一个报告值改变。



- **interpol** 选择显示了内插在每个报告值变动之间的一个模拟波形。



- 4、如果你要显示模拟选项，就需要选择下面来设置 Y 的范围值：
  - 以数据类型代表所有范围显示
  - 自动校正变量值的最大最小值
  - 显示用户定义最大最小范围
- 5、是否选择标记样本、标记符和尺寸。
- 6、从 **linear**, **log<sub>10</sub>** 或者分贝 (**dB**) 中选择总 Y 轴的缩放比例。
- 7、设置坐标显示方式。
- 8、点击 **OK** 来应用设置并关闭对话框。点击 **apply** 来应用改变并保持对话框打开，或者点击 **cancel** 来关闭对话框并忽略改变。

## 指针和标记

在图形显示中，你可以插入指针和标记。

### 使用指针

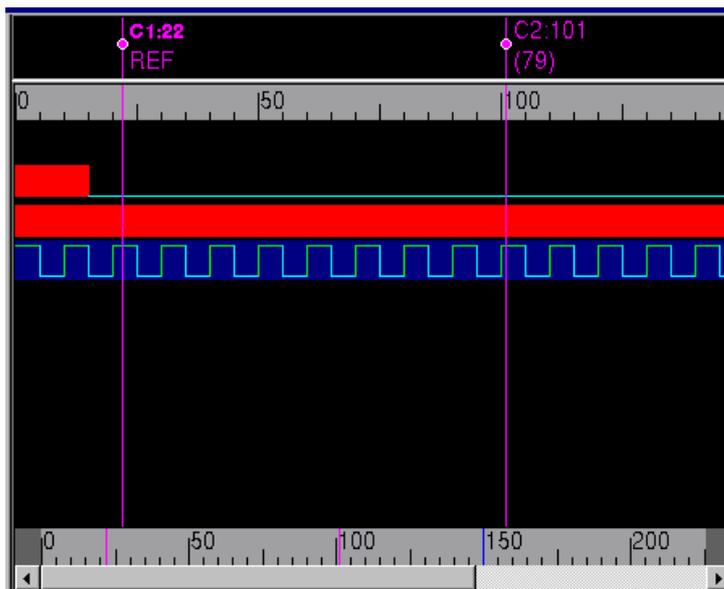
插入指针：

- 在图形显示中点击鼠标左键放置指针 C1。指针 C1 的默认位置是在时间 0 点。在图形显示中左击其他地方，指针 C1 便会移动到新的位置。
- 在图形显示中点击鼠标中间钮放置指针 C2。像指针 C1 一样，在图形显示中中击其他地方，指针 C2 便会移动到新的位置。

移动指针。在指针区域内移动指针，按住左键并拖动指针到你想要的位置。在波形或指针区域中你既可以点击左键也可以点击中键来分别移动 C1 或 C2。

在两个指针之间的间隔常常在标题区显示出来。

图 5-7 指针的图像显示



按图 5-7 显示的那样，在指针 C1 和指针 C2 之间的仿真时间和增量间距都在标记标题区显示出来。

## 运用标记

在插入与移动的方法上标记与指针不同。但像指针一样，标记的间距也在指针 C1 和 C2 之间显示出来。

你可以用标记对话框，查看在 4-22 页的“移动，隐藏和删除标记”或者按照以下方法来插入标记：

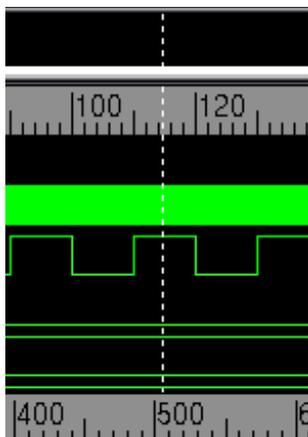
1、在图像显示中右击。这样会出现一个菜单，如图像 5-8 所示。

图像 5-8 CSM 的图像显示



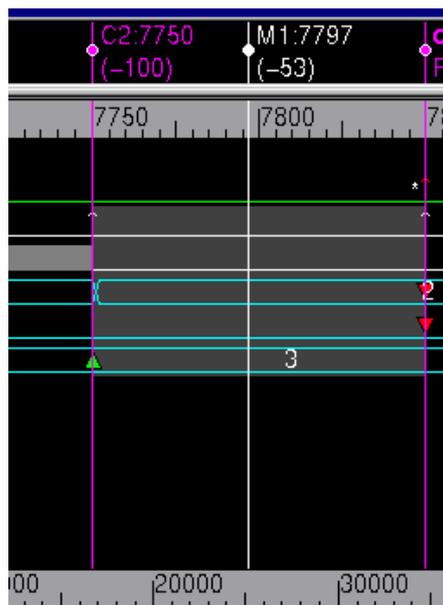
2、在菜单中选择 **create marker**。在图像显示中，在鼠标指针所在地方插入一条虚线。

图像 5-9 布置标记的虚线



3、当你在波形或标记的标题区移动鼠标时，虚线会追踪鼠标指针。**波形或指针的位置**。注意，标记注释显示标记位置和标记与指针 C1 之间的距离。

图像 5-10 新标记

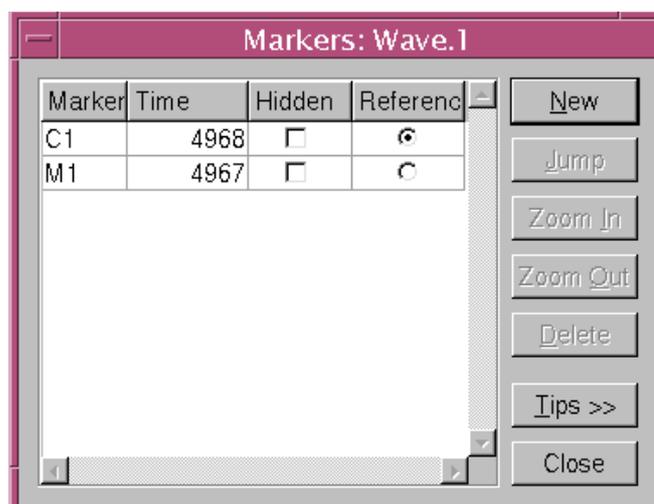


当你插入标记时，DVE 将他们命名为 M1,M2,M3 等等。你可以利用标记对话框重命名他们。

## 移动，隐藏和删除标记

标记对话框允许你新建，移动，隐藏和删除标记，建立标准标记，滚动图像显示直到释放标记。在波形窗中右击并从菜单中选择标记来打开这个对话框。

图像 5-11 标记对话框



C1 标记是默认的标准标记。在标准栏中选择标记来设置另一个标记作为标准标记。

点击 tips 钮来展开关于对话框帮助的对话框。

## 拖动缩放

在图像显示中有两种方法来放大和缩小:

在顶层菜单选择 **view>zoom** 菜单命令后中使用以下菜单

命令: **zoom in**, **zoom out**, **zoom full**, **zoom to time range** 和 **zoom to cursors**, 或者使用他们相应工具栏里的图标。在 CSM 中也有为图像显示设置的缩放命令。

在图像显示中拖动缩放。拖动缩放按以下描述。

拖动缩放, 移动鼠标光标到时间精度或波形显示区中的任意一点, 按住左键并拖到时间精度的某个地区。那个被选中的地区变成淡蓝色。当你松开鼠标时, 只有被选中的时间精度地区的图像被改变了。

## 扩展和收缩波形信号

你可以扩展和收缩波形信号的高度。选择 **view**, 然后选择 **increase row height** 或者 **decrease row height**。你也可以右击并选择 **properties**, 用信号属性对话框来增加或减小行的高度。

## 在图像窗中搜索

在图形窗中搜索时，如果在当前窗口中的任意一个信号被选中，将只会在此波形中搜索被选信号，如果没有信号被选，则将搜索所有信号。

可以使用工具栏里的前后搜索箭头  让 C1 从当前位置移动到下一位置。设置查找准则，点击工具栏里的向下箭头并选择下面的一项：

- any edge
- rising
- falling
- match
- mismatch
- value

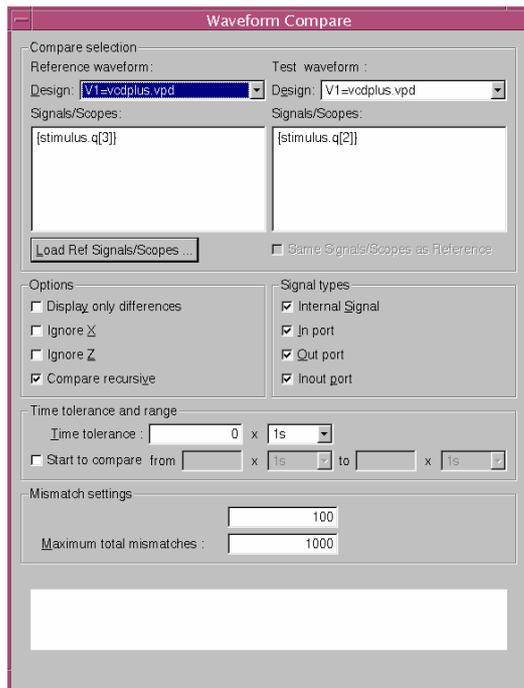
## 比较信号，域和组

你可以比较相同位宽的单个信号、域（为了比较变化的子集）、总线或在同一个或两个设计中的信号组。

查看比较：

- 1、从波形窗口的信号窗中选择一个或两个信号、信号组、域、总线。
- 2、显示比较信号对话框。右击信号窗，然后从菜单中选择 **compare**。

图像 5-12 波形比较对话框



3、 点击 **load reference signals/scopes** 然后选择带有标准信号和域的文本文件。

注意：如果你是从根目录中将两个设计比较，那么标准区域和测试区域可能是空的。

4、 在测试波形区域，选择测试设计和测试区域。如果你是从根目录中将两个设计比较，那么标准区域和文本区域可能是空的。

5、 选择 **only display differences** 会只显示那些在波形窗口中不匹配的结果。

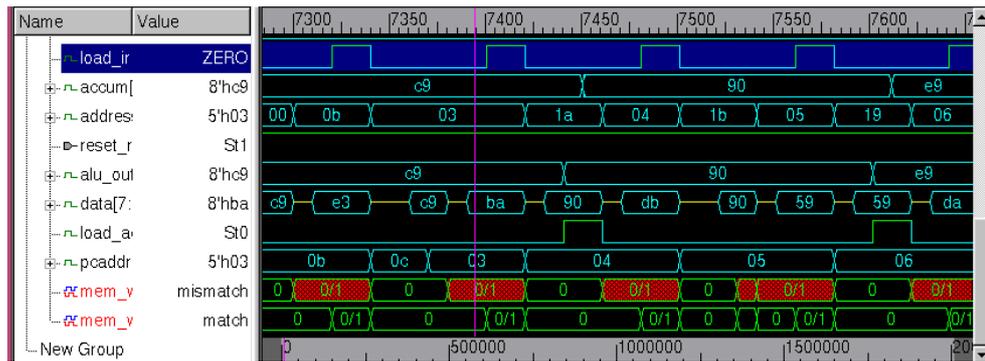
6、 在选项区中，你可以选择 **ignore x** 或 **ignore z** 中的一个或两个，举个例子，如果你选择 **ignore x**，当标准信号的值为 x 时，无论测试信号的值是什么都被视为匹配的。

7、 选择 **in port, out port, inout port** 和 **signals** 中的一个或全部来将信号进行比较。

8、 输入一个时间容限，用来过滤出那些时间范围比容限范围小的不匹配值。

9、 为每个信号的最大失配和失配最大数输入失配设置。

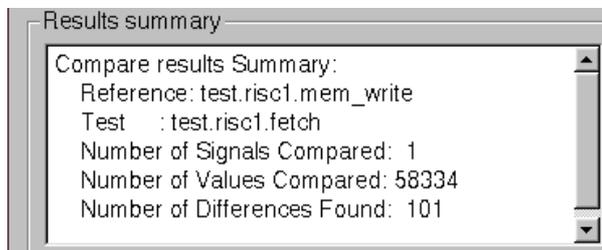
10、 点击 **apply** 开始比较并保持对话框打开，或点击 **ok** 开始比较并关闭对话框（你可以在任何时候在信号窗目录中打开它）。



11、再次查看比较信息。在波形窗中选择一个结果，右击，然后选择 **show compare info**。

结果概要在波形比较对话框中显示。

图像 5-14 波形比较概要



12、你可以改变选项然后再比较。

## 建立总线并设置表达式

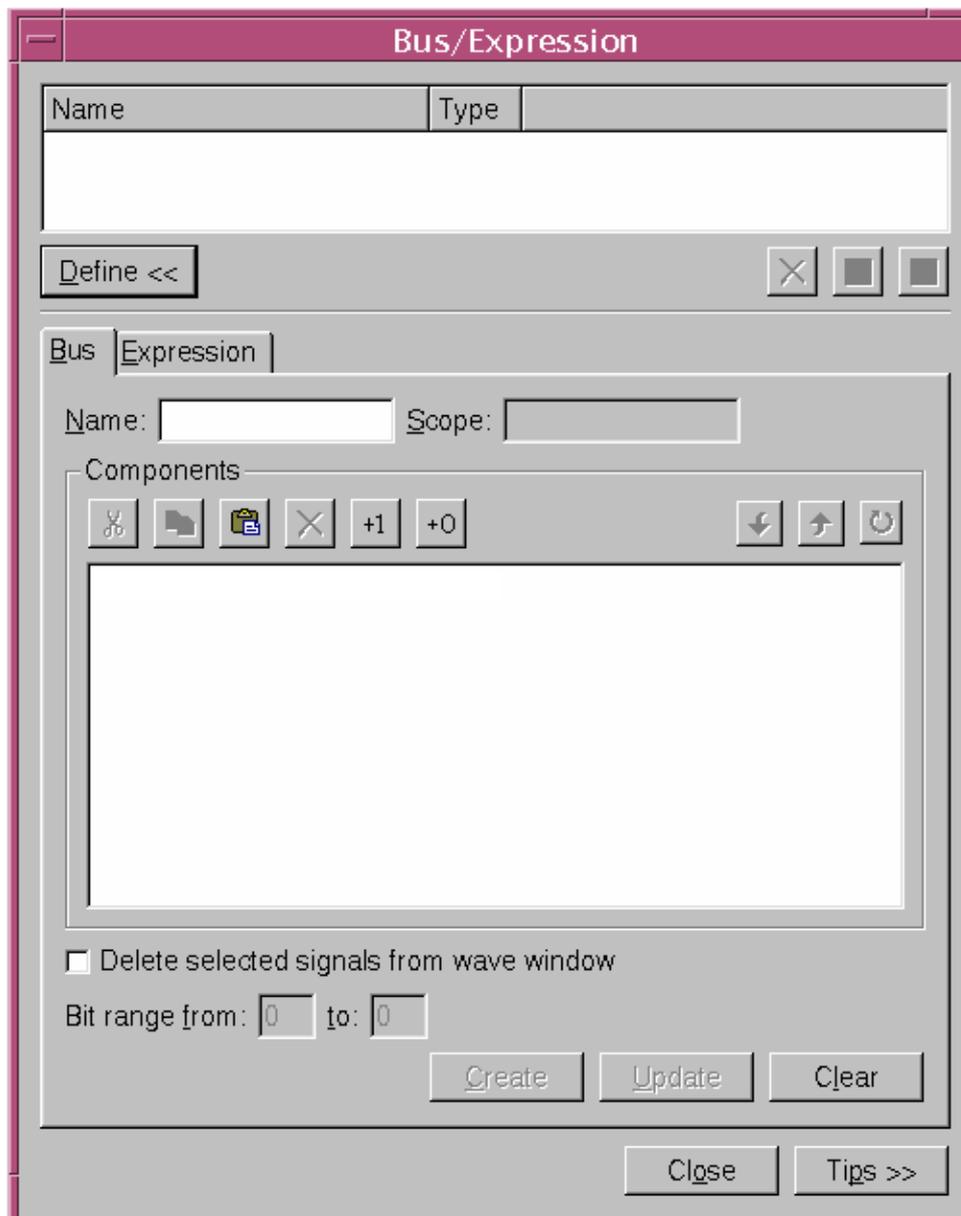
通过使用打开的设计中的信号，用 DVE 总线生成器函数来新建并编辑总线或表达式。你也可以在用户指定的范围下建立总线和表达式。

选择 **signal>set bus** 或右击在菜单中选择 **set bus** 来显示总线/表达式对话框。你可以通过选择在波形窗口中信号窗里的信号，或从层次浏览器拖动、拷贝到总线生成器对话框的方法来包含部分信号。你可以在总线中添加和删除信号或者改变他们的顺序。你可以将成分拖动到列表窗口来查看其值。在你新建一个总线后，你可以像在此设计中的任何信号一样地使用它。默认形式下，他存在于最高级的信号组中，对于它的成分来说，这很普遍。

## 使用总线/表达式对话框

图像 5-15 显示了总线/表达式对话框

图像 5-15 总线生成器对话框



## 总线/表达式列表框

列表框显示的是总线和表达式。

## 显示/隐藏细目表格

点击 **define** 按钮来显示或隐藏总线或表达式的列表。

总线标记

- **name** 是信号总线的名字。总线可以用语言（verilog,VHDL）命名为任何合

法名字。当在编辑模式中打开时，这个区域是不活动的。

- **scope** 是用户定义域的名字。当在编辑模式中打开时，这个区域是不活动的。
- **bit range** 是 0 到 N。N 是总线组成部分的数量。矢量和结构被按位展开。举个例子，如 “**top.risc.pc[3..0]**” 被添加到列表中，那么将被添加 4 项。
- 工具栏允许你建立和编辑总线，下面是对总线生成器工具栏中命令描述。

工具栏项目	功能
	剪切从成分列表中选择的部分。
	拷贝从成分列表中选择的部分。
	粘贴一个或多个成分的拷贝。
	删除所选部分。
	给总线增加一个 1`b1 的常量信号。
	给总线增加一个 1`b0 的常量信号。
	向下移动从成分列表中选择的部分。
	向上移动从成分列表中选择的部分。
	撤销命令。

**Components** 列出目前所选总线的单元。当创建一个新的总线时，你可以在波形窗口的信号窗中选择成分，右击，然后选择 **create bus** 创建一个来包含所选成分的总线。当创建一个新总线或编辑一个已存在的总线时，你可以从波形窗口和层次浏览器中拖动信号到成分列表中。你也可以用工具栏里的命令来修改成分和顺序。

## 创建一个总线

按以下步骤创建总线：

- 1、在波形窗口的信号窗中选择信号和总线用来包含新总线。  
或者不选择信号，稍后你可以选择成分。
- 2、右击信号窗或从菜单中选择 **signal** 然后选择 **set bus**。
- 3、输入一个新总线名字。

- 4、 在用户指定的范围下创建总线，输入一个域名。
- 5、 选择性的为总线添加常量 1 或常量 0 信号。
- 6、 给成分列表添加信号和总线：
  - 从波形窗口的信号窗或层次浏览器中拖动成分。
  - 在信号窗中选择成分，在总线/表达式工具中点击  后再点击 .
- 7、 点击 OK 存储总线并在波形窗中将它显示出来。

## 修改总线成分

你可以在一个新总线中用总线生成器工具编辑一个已存在的总线或修改成分和他们的顺序。

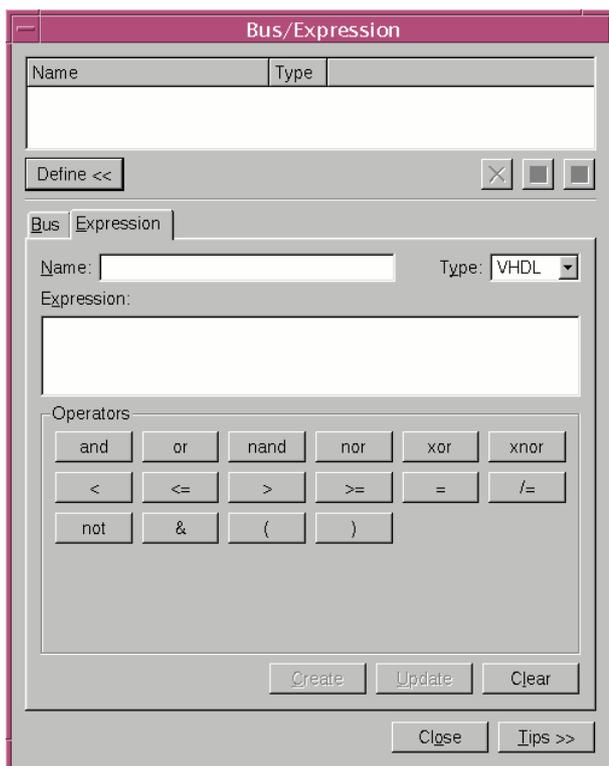
- 1、 如果你想要修改一个已存在的总线，先在波形窗口的信号窗中选择总线，再右击信号窗或在从菜单中选择 **signal**，然后选择 **set bus**。
- 2、 给成分列表添加信号和总线：
  - 从波形窗口的信号窗或层次浏览器中拖动成分。
  - 在信号窗中选择成分，在总线生成器工具中选择 ，然后点击 .
- 3、 删除成分，在成分列表中选择成分，然后点击总线生成器中的 .
- 4、 在列表中向上或向下移动成分。在成分列表选择一个或多个成分然后点击总线工具栏中的  或 .
- 5、 将成分彼此的顺序调换。在成分列表选择一个或多个成分，然后在总线生成器工具栏中点击 .
- 6、 点击 **OK** 存储总线，并将它在波形窗口中显示出来。

## 管理表达式列表

使用表达式列表创建和修改表达式。注意，不支持复杂的 SV 数据类型。

- 1、 在信号窗中右击，从菜单中选择 **signal**，然后选择 **set expressions**。  
总线/表达式对话框显示为图像 5-16。

图像 5-16 表达式标签

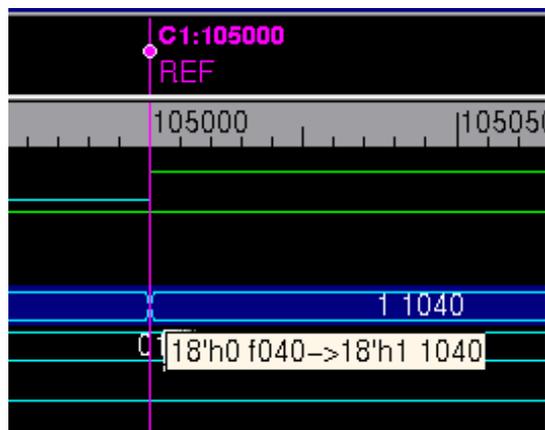


- 2、创建表达式。输入一个表达式的名字，修改表达式，从列表选择一个表达式并选择一种类型。
- 3、点击操作将他们插入到表达式中。
- 4、点击 **create** 或 **update** 存储表达式。

## 查看总线值

通过在波形上布置光标，在波形中查看总线值的工具提示。图像 5-17 显示了一个工具提示，此提示显示的是在转变的期间的变化值。

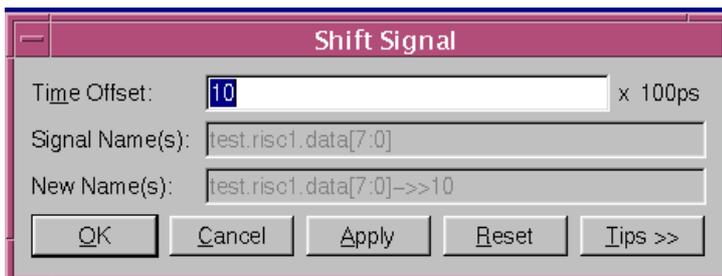
图像 5-17 查看在转变时的总线值



## 移位信号

用创建一个基于时间位移信号的方法来移位信号。

- 1、在信号窗中选择一个信号。
- 2、选择 **signal>shift time** 会显示出移位信号对话框。
- 3、输入在波形窗中一个正的时间偏移量来向右移位信号或一个负数来向左移。

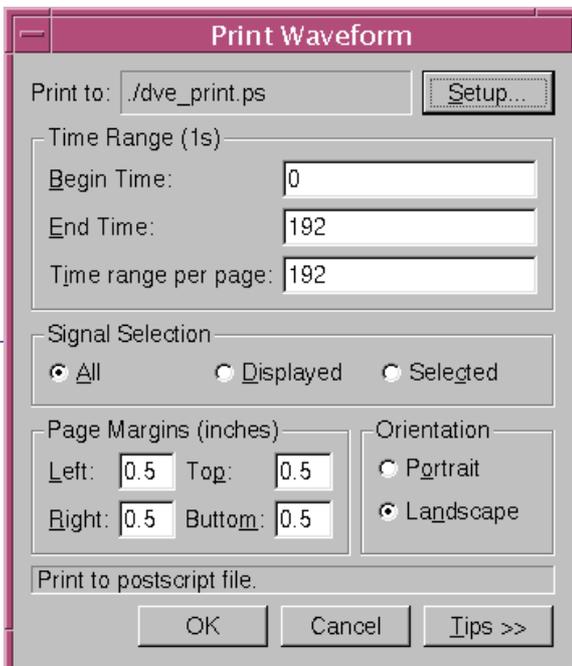


信号将会以原名后跟随时间偏移量的形式显示出来。上面的例子将会显示为 **test1.risc.daata(7:0)->>10**。

## 打印波形

你可以从一个动态波形窗口选择时间范围和信号后将波形打印到文件或打印机。

- 4、从一个活动的窗口中选择 **file>print** 或点击打印工具按钮  便可以显示打印波形对话框。



- 5、点击设定按钮来设置打印选项：

- 1、打印机或打印到文件

- 2、 彩色打印或黑白打印
- 3、 打印校准和纸张尺寸
- 4、 打印范围和数目选项
- 6、 选择总体打印的时间范围和单张的时间范围。
- 7、 选择是否打印 **all**、**displayed** 或者 **selected** 信号。
- 8、 选择页面空白。
- 9、 选择 **landscape** 或 **portrait** 校准。
- 10、 点击 **OK** 打印。

## 6 运用列表窗口

列表窗口以表格格式显示仿真结果。对于 **verilog**，列表窗口支持网型和寄存器变量。对于 **VHDL**，它显示信号和进程变量。

本章包括以下几点：

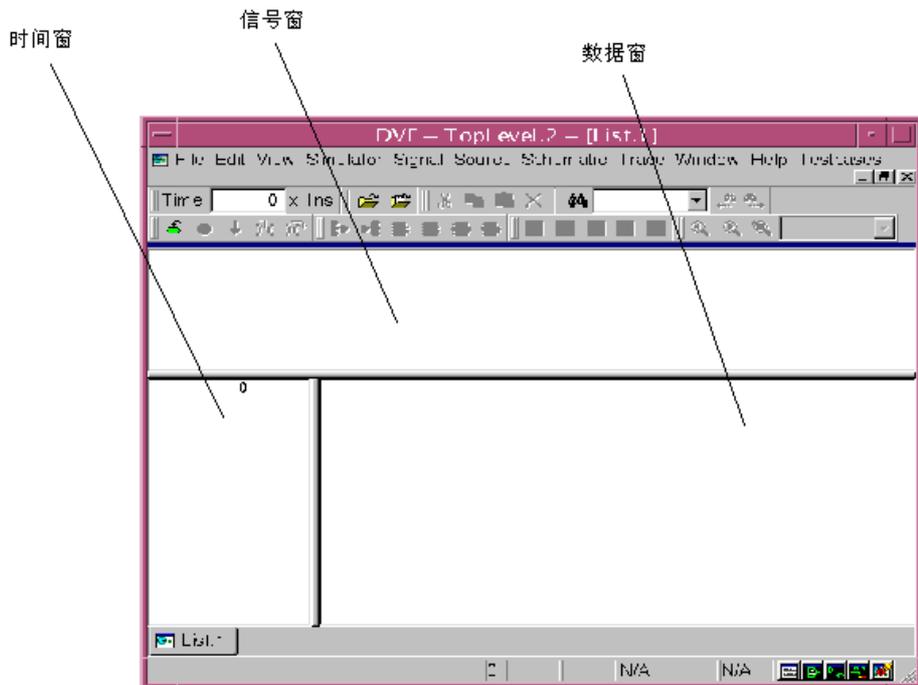
- 8、 列表窗口
- 9、 数据显示
- 10、 驾驭仿真数据
- 11、 制定显示
- 12、 储存列表格式

## 列表窗口

列表窗口包括三个章节:

- 13、 在仿真数据之上, 信号窗以信号名字作为题头。
- 14、 数据窗以列表格式显示仿真结果。
- 15、 仿真时间窗。

图像 6-1 列表窗口



## 数据显示

列表窗口显示结果的方式与波形窗口相似, 但它以列表格式显示结果。你可以在波形窗口中同样的方法在列表窗口中显示数据。

从另一个 **DVE** 窗口中拖动信号或断言。

使用 **file>open** 打开数据库。

选择 **file>load session**, 然后选择一个保存过的对话。

## 将信号拖动到列表窗口中

将其他 **DVE** 窗口中的数据在列表窗口中保留下来。

- 选择工具或 **view>list window**, 打开列表窗口。
  - 从层次浏览器、资源窗、波形窗口或断言窗口中将想要的域或断言拖动到列表窗口。
- 数据以默认格式显示。

## 打开数据库

- 1、 选择工具中的  或 **view>list window**, 打开列表窗口。
- 2、 选择 **edit>open database** 并选择 **.vcd**、**.vpd** 或 **.dump** 的数据库文件来打开存档的对话。  
数据以默认模式被显示。

## 载入对话

按照一下方式打开先前存储的对话:

- 1、 选择工具中的  或 **view>list window**, 打开列表窗口。
- 2、 选择 **edit>open database** 并选择 **.vcd**、**.vpd** 或 **.dump** 的数据库文件来打开存档的对话。  
数据以默认格式显示。

## 驾驭仿真数据

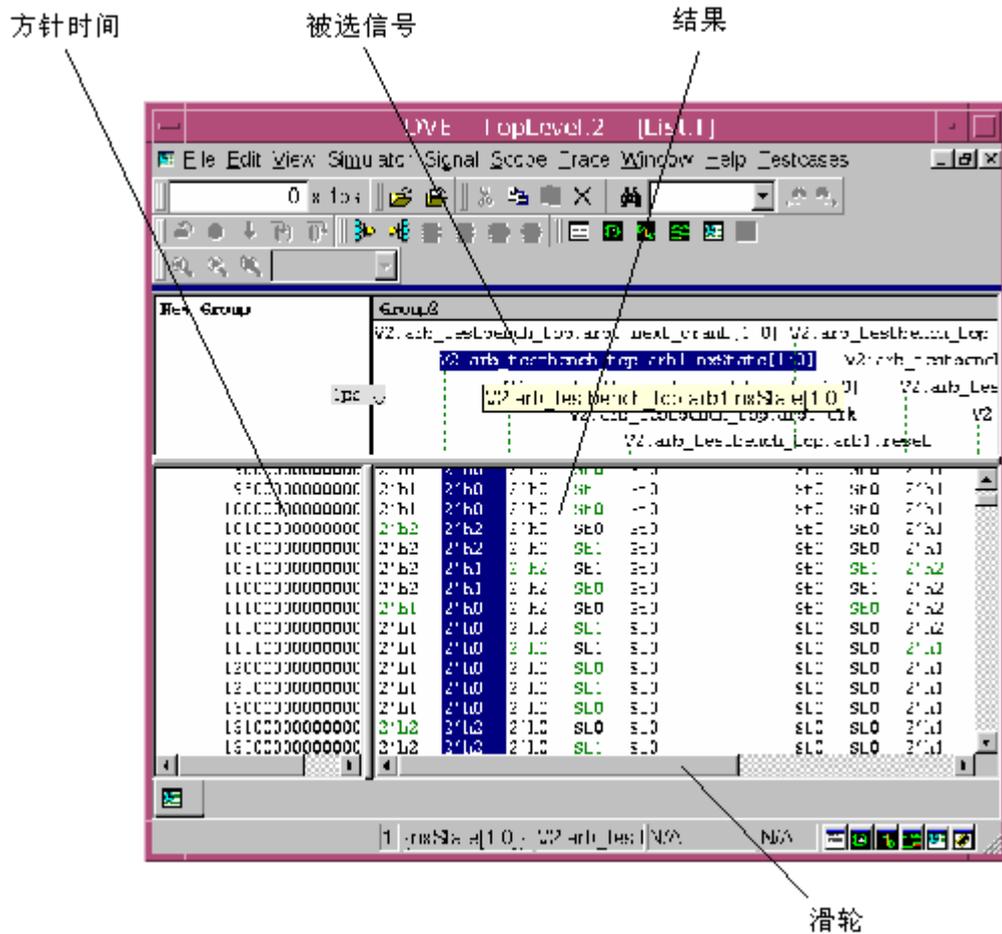
本章包括以下几点:

- \* 在列表窗口中查看数据
- \* 使用标记

## 在列表窗口中查看数据

- 16、 用底部滚动条的左右移动查看信号和他们的值。
- 17、 用左边滚动条的上下移动查看仿真时间。
- 18、 如图像 6-2 所示, 在信号窗内选择一个信号可以突出这个信号。

图像 6-2 列表窗口



## 使用标记

在列表窗口中设置些标记可以加速运送:

- 1、选择 **view>set markers** 显示标记列表对话框。
- 2、在列表表格里，点击 **new** 创建一个新的标记。
- 3、为每个新标记选择时间单元，在每个标记处输入时间，如果不想在数据窗中显示此标记可以点击 **hidden**，然后点击 **return**。
- 4、重复 2、3 步骤创建更多标记。
- 5、在标记列表对话框的列表中选择 一个标记，然后按 **jump** 将数据显示移动到被选标记。或者选择 **view>goto marker**，然后从列表中选择 一个标记。
- 6、移除标记，在标记列表对话框中选择 **delete**。或者选择 **view>delete marker**，然后从列表中选择 这个标记。

## 制定显示

本章包括以下几点:

- \* 设置信号性能

## 设置优先级

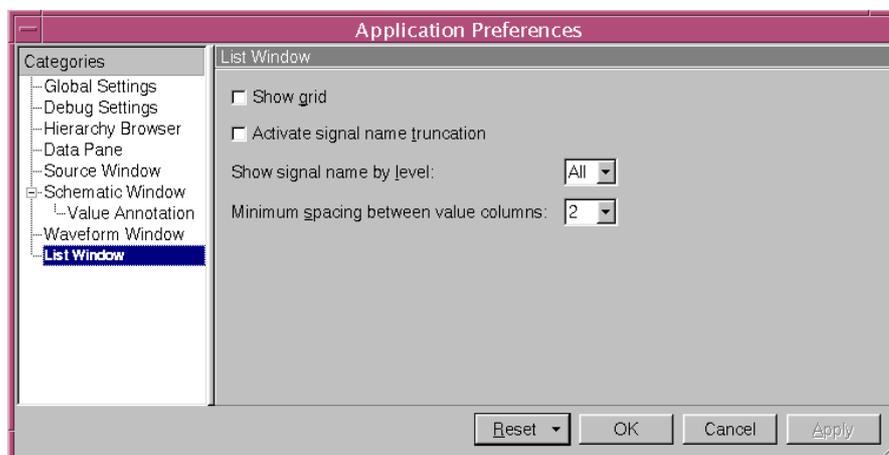
你可以制定列表显示为:

- 19、 开/关方格线
- 20、 缩减信号名
- 21、 分级别显示信号
- 22、 在行之间制定空间

制定显示:

- 1、 选择 **edit>preferences**。
- 应用优先级对话框显示。
- 2、 在类别窗中选择 **list settings**。

图像 6-3 应用优先级对话框



- 3、 选择或取消显示个, 全名, 与/或左对齐校验框。
- 4、 点击 **apply** 查看你的更改并保持对话框打开或者点击 **OK** 来应用你的改变并关闭对话框。
- 5、 点击 **reset** 再次应用默认模式。

## 设置信号性能

制定信号显示, 为个别信号设置信号性能。

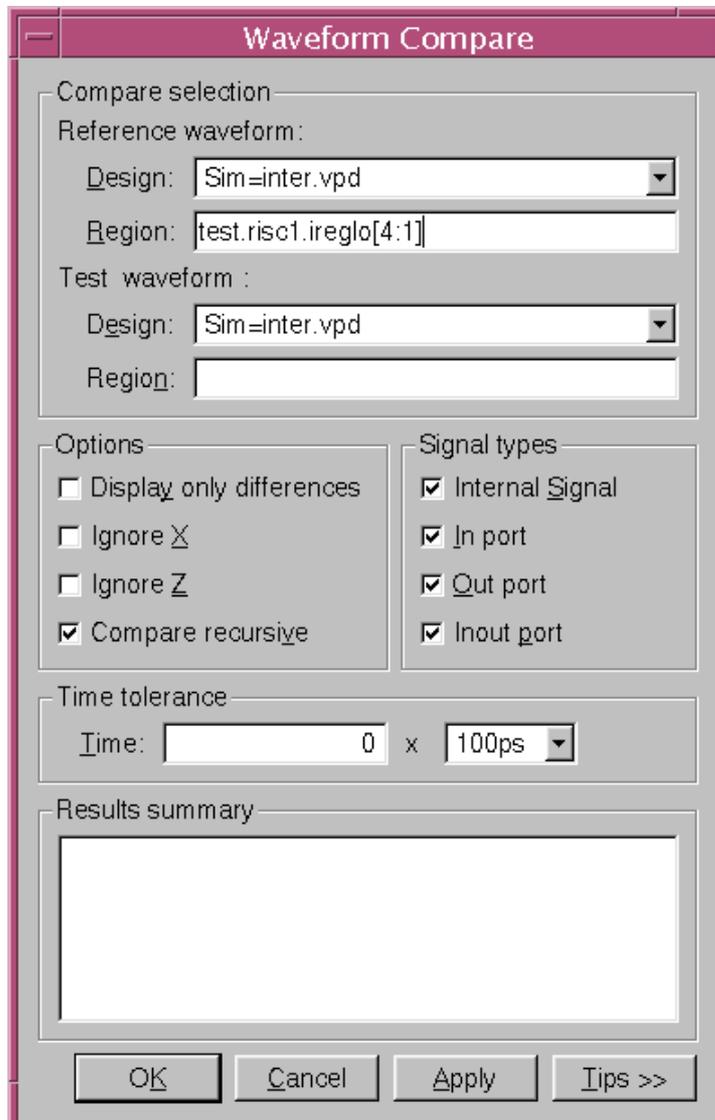
- 1、 在信号窗内选择一个信号。
- 2、 选择 **signal>signal properties** 可以显示信号功能对话框。
- 3、 输入被选信号值的列宽字符数。
- 4、 在数据窗中选择是否当一个信号值改变时引发一行新值。
- 5、 点击 **apply** 设定改变并保持对话框打开以便选择和设定更多的信号列宽。或者点击 **OK** 应用改变并关闭对话框。

## 信号对比

在列表窗口中对比信号与在波形窗口中的对比方式很相似。观察比较为：

- 1、 在任何 **DVE** 窗口的信号窗中选择一个或两个信号、信号组、域、或者总线。
- 2、 显示对比信号对话框，在信号窗内右击，然后从菜单中选择 **compare**。

图像 6-4 波形比较对话框



- 3、 在标准波形区域中，如果在第一步你没有选择标准设计和信号，而只选择了标准设计，那么输入标准区（信号，域或总线）用于比较。

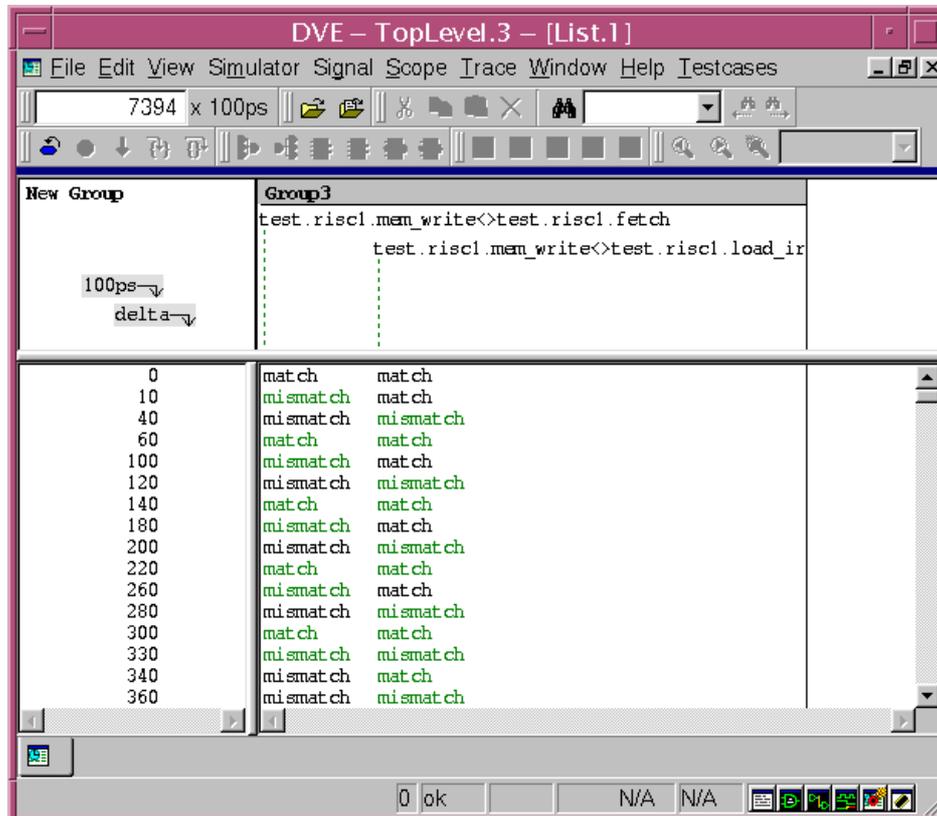
注意：

如果你从根目录将两个设计比较，那么标准区和测试区可能是空的。

- 4、 在测试波形区域内，选择测试设计和测试区。如果你从根目录将两个设计比较，那么标准区和测试区可能是空的。
- 5、 选择 **only display differences** 将只显示那些波形窗口中不匹配的结果。
- 6、 在选项章节中，你可以选择 **ignore x** 或 **ignore z** 中的一个或两个，举个例子，如果你选择 **ignore x**，当标准信号的值为 x 时，无论测试信号的值是什么都是匹配的。

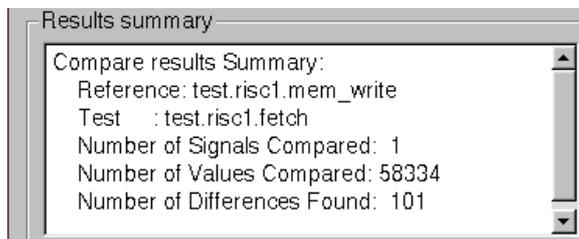
- 7、通过选择所有的 **in-port**, **out port**, **inout port** 和 **signals** 来选择信号进行比较。
- 8、点击 **apply** 开始比较并保持对话框打开。  
或者点击 **OK** 开始比较并关闭对话框（你可以在任何时候打开信号窗目录）。  
结果在当前波形窗口显示。

图像 6-5 在列表窗口中比较的信号组



- 9、回顾比较信息，在波形窗口中选择一个结果，右击，然后选择 **show compare info**。  
结果概要在波形对比对话框中显示。

图像 6-6 波形的对比摘要报告



- 10、你可以改变选项，然后比较。

## 保存列表格式

当你已经在列表窗口制定了显示后，你可以存储列表格式以便以后用到。按照以下方式储存列表格式：

- 1、选择 **file>save list format** 显示存储列表格式对话框。
- 2、用 **a.tcl** 来扩展你的格式文件名。

### 3、选择 **save**。

## 8 断言与覆盖率

这部分描述的是断言结果的观察与应用。下面的题目包括：

- SV断言的编译
- 断言结果的观察
- 断言的调试

### SV 断言的编译

使用DVE进行SV断言的编译时，在VCS命令行使用**-assert dve**标示进行调试。

**提示：**如果你使用Solaris操作系统（SUN开发的一种网络操作系统）版本是5.8以前的版本，那么连接步骤可能会花费很长时间。在Solaris操作系统下使用DVE调试设计编译时，用以下任意方法可以避免连接延时

- 确保你的**Solars** C编译器版本为5.8或者更高。检查你的编译器版本，在命令行键入以下命令：

**ld -V**

系统会提示你的连接器版本，例如：

**Ld:Software Generation Utilities – Solaris Link Editors:  
5.8 – 1.283**

- 编译时使用gcc C编译器，例如：

**Vcs -assert dve -PP -sverilog a.v -ld gcc**

## 断言结果的观察

断言面板可以显示SVA，OVA断言和覆盖性能结果

- DVE通过一些实例来显示断言结果，其中包括断言事件开始结束的时间，增量，断言失效中错误的字符串，失效的数量，执行成功的数量，未完成以及尝试的数量。图8-1就是一个断言面板，有效断言为绿色，无效的为红色。

图8-1. 断言结果

Name	Instance	Start	End	Delta	Reason	Failures	Incompletes	Successes	Attempts
HOUR_24_CHECKER	/TB_WATCH/CHECKERS:0	0	0			0	0	10000	10000
MIN_60_CHECKER	/TB_WATCH/CHECKERS:0	0	0			0	0	10000	10000
RESET_CHECKER	/TB_WATCH/CHECKERS:8295000	8295000	0		(((hour == 5'b0) && (min == ...	15	0	19985	20000
SEC_60_CHECKER	/TB_WATCH/CHECKERS:9400000	9410000	10000		(sec == 6'b0)	137	0	9863	10000

注释：DVE中，在VCS-assert命令行使用-PP标示跟踪SVA，如果没有对SVA进行跟踪，断言值的改变将会被存入VPD文件，在波形窗口中才能被显示出来，断言的尝试却不能被跟踪。

- 覆盖性能结果，如图8—2，显示项包括开始结束的时间，增量，以及竞争，搭配不当，未完成和尝试的数量。

图8—2 覆盖性

Name	Instance	Start	End	Delta	Matches	Mismatches	Incompletes	Attempts
HOUR_24_COVER	/TB_WATCH/CHECKERS:0	0	0		10000	0	0	10000
MIN_60_COVER	/TB_WATCH/CHECKERS:0	0	0		10000	0	0	10000
RESET_COVER	/TB_WATCH/CHECKERS:0	0	0		19985	0	0	19985
SEC_60_COVER	/TB_WATCH/CHECKERS:0	0	0		9863	0	0	9863

## 显示标准的设置

你可以使用断言面板的导航栏显示你所需要的显示项，如图8—3

图8—3 断言和覆盖性能

The image shows two screenshots from the DVE tool. The top screenshot is the 'Assertions' window, displaying a table of assertion results. The bottom screenshot is the 'Cover properties' window, displaying a table of coverage results.

Name	Instance	Start	End	Delta	Reason	Failures	Incompletes	Successes	Attempts
HOUR_24_CHECKER	/TB_WATCH/CHECKERS:0	0	0	0		0	0	10000	10000
MIN_60_CHECKER	/TB_WATCH/CHECKERS:0	0	0	0		0	0	10000	10000
RESET_CHECKER	/TB_WATCH/CHECKERS:8295000	8295000	8295000	0	{{(hour == 5'b0) && (min == ...	15	0	19985	20000
SEC_60_CHECKER	/TB_WATCH/CHECKERS:9400000	9410000	10000	10000	(sec == 6'b0)	137	0	9863	10000

Name	Instance	Start	End	Delta	Matches	Mismatches	Incompletes	Attempts
HOUR_24_COVER	/TB_WATCH/CHECKERS:0	0	0	0	10000	0	0	10000
MIN_60_COVER	/TB_WATCH/CHECKERS:0	0	0	0	10000	0	0	10000
RESET_COVER	/TB_WATCH/CHECKERS:0	0	0	0	19985	0	0	19985
SEC_60_COVER	/TB_WATCH/CHECKERS:0	0	0	0	9863	0	0	9863

以下是起始时间显示标准的设定:

- 1、单击显示控制栏中的箭头, 选择**all, staring, ending, starting and ending**.  
按已选的标准显示。
- 2、选**at time**或者**from**,然后选**begin,current**,或者**end to**设定窗口。
- 3 按以下调节设定:
  - 基于断言选择**failures,incompletes,successes**或者**all**。
  - 基于覆盖性能选择**uncoverd,coverd**或者**all**。默认的是**uncoverd**。

## 断言的调试

当你打开一个包含断言的设计时, 即使所有的断言都通过, DVE还是会显示断言面板。默认项是显示失效的断言。首先显示前十个失效和成功的断言, 点 + 来寻找你需要的断言。图8—4展示一个尝试列表。

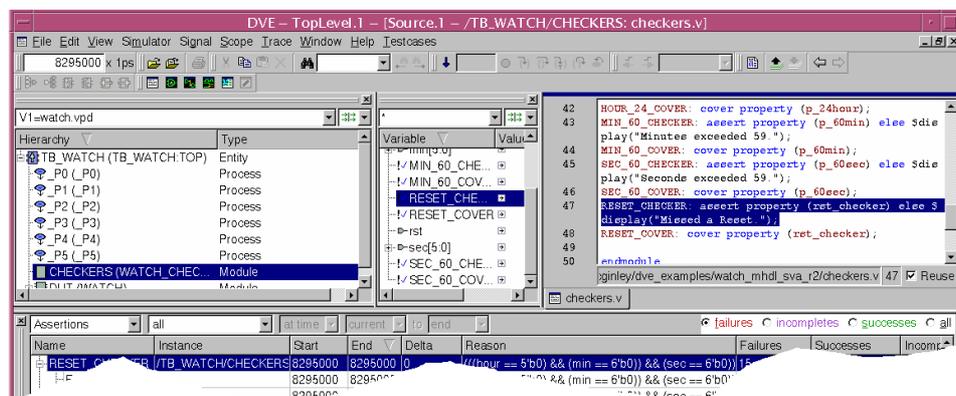
图8—4 断言尝试列表

Name	Instance	Start	End	Delta	Reason	Failures	Successes	Incompletes
RESET_CHECKER	/TB_WATCH/CHECKERS	8295000	8295000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)	15	19985	0
Failure1		8295000	8295000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure2		8305000	8305000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure3		8315000	8315000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure4		8320000	8320000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure5		8330000	8330000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure6		8335000	8335000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure7		8345000	8345000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure8		8350000	8350000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure9		8360000	8360000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Failure10		8365000	8365000	0	((hour == 5'b0) && (min == 6'b0)) && (sec == 6'b0)			
Success1		0	0	0				
Success2		5000	5000	0				
Success3		10000	10000	0				
Success4		15000	15000	0				
Success5		20000	20000	0				

断言面板与DVE其他面板窗口是互相连接的。在层次与变量面板，资源窗口和波形

窗口中，在断言面板中双击一个断言实例就能显示这个断言。如图8—5。

图8—5



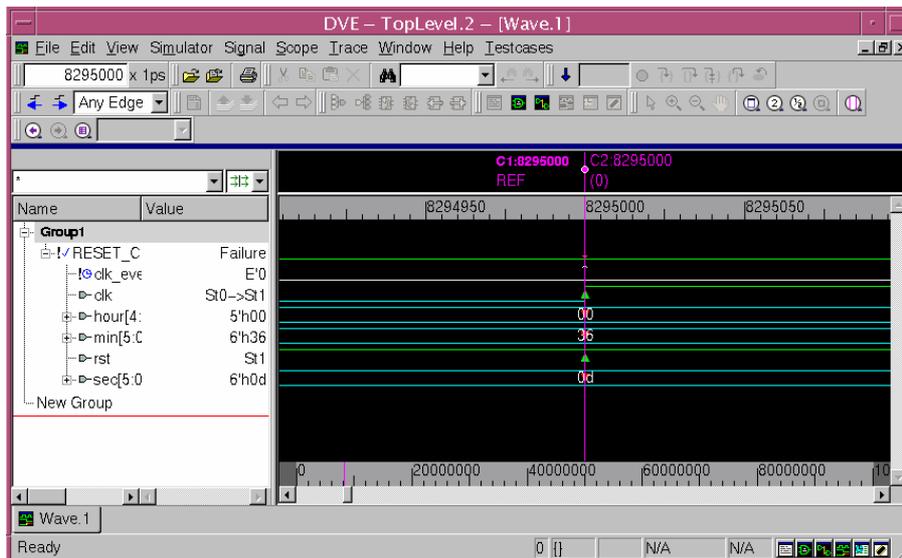
出现以下情况：

- 层次面板显示相关联单元或者包括断言的HDL。
- 变量面板显示的是相当于单元或者范围的HDL变量。这不是一个具体的断言（例如，它可能包含更多断言中用到的信号）
- 最多可以显示3个资源窗口：一个为HDL资源窗口，一个为约束资源窗口，一个为OVA定义资源窗口。
- 波形窗口显示的所选断言是以失效断言为中心的。光标会变灰来标记所选断言开始与结束的时间，一个绿色的圈会在特定的时间标记出信号变量，这样有助于断言的底层表达。底层表达失效时会用一个红色的圈标记。一个底层表达的失效可能会使整个断言失效。

## 观察波形窗口中断言的失效

图8—6所示，从开始到结束的时间里断言没有增量。

图8—6波形窗口中的断言



在图8—6中

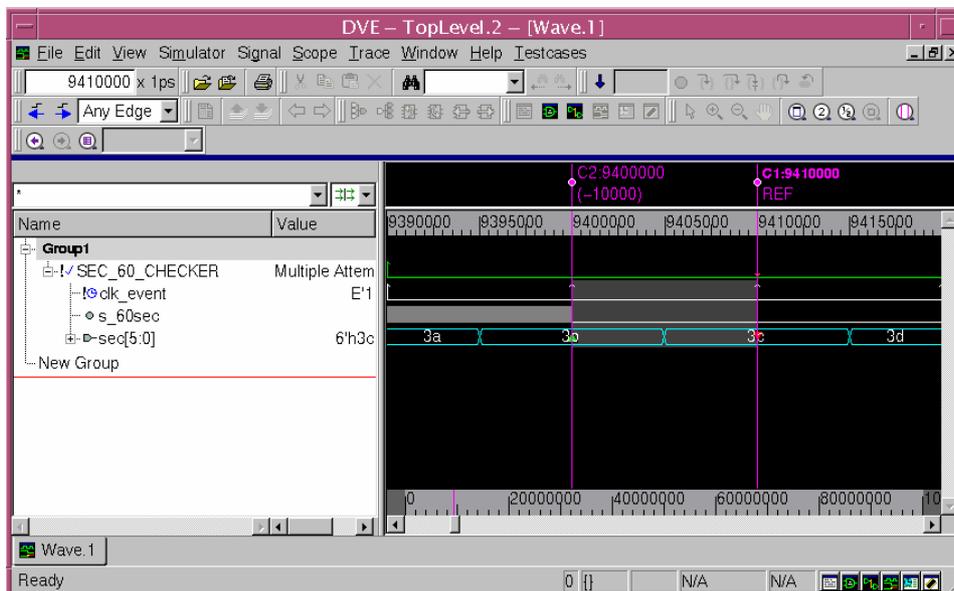
- C1与C2的光标会自动位于断言开始和结束的位置。在这个断言中，并不包含时间组成部分（时序），所以C1与C2的时间是相同的。
- 在信号Group1中，断言RESET\_CHECKER位于树形列表的最上部。这是一个信号的断言结果。波形由红色和绿色的箭头组成，绿色在断言成功的位置标注，红色标注的是断言失效的位置，红色的箭头标示的是第一个失效的位置。
- RESET\_CHECKER扩充的组件
  - 第一个组成是clk\_event，每个时钟事件都会显示断言开始的时间，时钟也会按时序产生。
  - clk,rst,hour,min,sec，这些信号将会被送到成功或失效的信号中。
  - 波形上绿色的点表示信号在那个时钟点OK。红色的点表示在那个时钟点信号断言失效。

## 观察断言失效时间增量

当一个断言失效有增量时，波形窗口会打开，显示所选断言失效。光标会在灰色

的区域中标记所选断言开始和结束的时间。一个绿色的圈在信号里表明这个信号或变量是有效的，红色的反向圈和C1光标表明失效。

图8—7 波形窗口中的断言失效时间增量



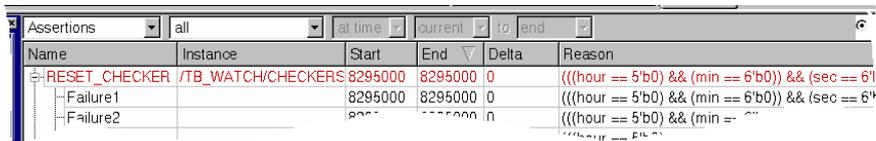
- 如果你把鼠标光标放在绿色或者红色光圈上，会在上方出现一个消息框，显示出信号作用的详细资料
  - 如果是绿色的箭头，提示会告诉你信号有效的原因
  - 如果是红色的箭头，提示会告诉你信号失效的原因
- 窗口中白色的箭头标注的是断言时钟事件
- 如果你把鼠标光标放在 **attempt failure** 上，会在上方出现一个消息框显示失效的详细资料。消息包含（对于当前结束的断言尝试）：
  - 开始时间
  - 结束时间
  - 增量
  - 实例
  - 因素

## 源代码的定位

显示断言有关的代码:

- 1、在断言窗口标签上双击assertion attempt (看图8—8)

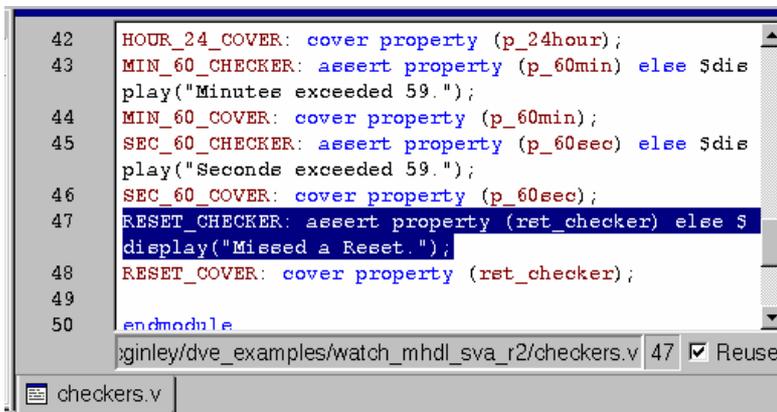
图8—8选择一个断言尝试



代码如以下所示:

- 资源面板显示被内嵌或绑定的断言处的HDL代码。
- 资源面板显示加亮的断言代码 (看图8—9)

图8—9断言源代码



- 2、在默认文本编辑器中编辑断言, 选Edit > Edit Source。

## 9 动态驱动和载入的跟踪

注释: 它具有limited-customer-availability(LCA)特性。你可以在没有许可的情况下使用LCA特性, 如果使用LCA特性你需要在VCS命令提示符下打入下列命令:

```
Vcs -lca Y-2006.06-sp1
```

这部分描述的是如何使用DVE(Discovery Visual Environment)跟踪信号的驱动与载入

包含下面几部分:

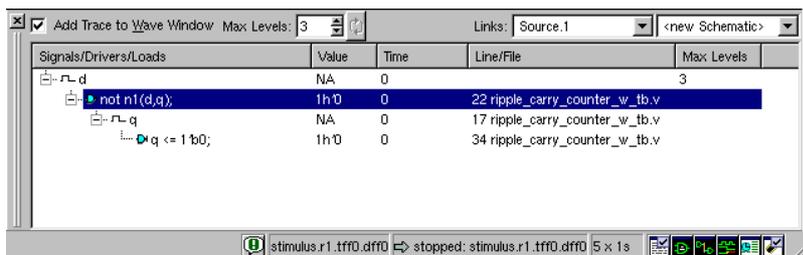
- 跟踪功能的概述
- 跟踪驱动
- 跟踪载入

### 跟踪功能

使用DVE跟踪驱动和载入并且观察影响信号值的连接

- 一个活动的信号驱动是在给定时间内给出变量的信号驱动
- **Displaying all drivers**会显示所有可能给出信号变量的驱动
- 信号的载入是输入端口, 输入输出双向端口和读信号值的声明

图9—1 驱动与载入面板



为跟踪驱动与载入, DVE提供以下功能:

- 多重驱动面板允许被多重顶层窗口包含

- 顶层之间驱动面板是相互独立的
- 点击X符号可以随时删除驱动面板
- 驱动面板可以随时停留在任何地方

驱动面板可以与在同一顶层的资源窗口和原理图窗口相连，资源窗口最多只能有一个被连接。链接：组合框，顶层面板可以显示当前相连的窗口。默认情况下第一个打开的资源面板与驱动面板相连而不与原理图相连（<new Schematic>中已指出）

资源窗口与原理图窗口的相连意味着这些窗口可以跟踪所选项。也就是说在驱动面板中选定一个目标，这个目标在相连的资源窗口与原理图窗口中也被选中。

检验添加波形检验栏可以把驱动面板中的全部信号添加到波形窗口中

未检验添加波形检验栏不会删除波形窗口中的任何项但能阻止驱动面板中的额外信号被添加到波形窗口中。

## 支持的功能

- 所有Verilog类型，结构，控制通路。
- Verilog门和UDPs（用户数据协议）
- VHDL 只对进程有效.所有进程内部的驱动都是活动的

## 不支持的功能

SystemVerilog 数据类型不被支持

## Vera 的功能

- 一个驱动器在被证实为不活动之前总是活动的。
- 控制器的控制通路，如果有部分被估计是错误时，驱动器都会变成不活动的
- 循环语句（例如 always,while,repeat）中的驱动程序不管控制路径如何评估都是活动的

### 驱动器面板菜单的使用

<b>Trace Drivers</b>	执行新的驱动跟踪所选信号
<b>Trace Loads</b>	执行新的载入跟踪所选信号
<b>Show Source</b>	显示源文件和所选信号线，如果有多条信号线被选只显示第一个信号线
<b>Show Schematic</b>	显示所选信号的设计原理图，信号线的颜色会被加亮（默认为白色）
<b>Show Path Schematic</b>	显示所选信号的路径原理图，信号线的颜色会被加亮（默认为白色）
<b>Select All</b>	选定当前所有可见项
<b>Delete</b>	撤销所有选定项.
<b>Delete All</b>	删除所有选定项
<b>Show Active Drivers</b>	切换当前所显示的驱动
<b>Show All Drivers</b>	切换所显示的驱动
<b>Clear Trace</b>	清除驱动面板中所选跟踪
<b>Clear All Traces</b>	清除所有驱动面板中的信息
<b>Add to Waves</b>	添加到波形窗口
<b>Add to Group</b>	添加到组
<b>Add Trace to Waves</b>	添加跟踪信息到波形图中
<b>Go to Time</b>	仿真接近规定的时限
<b>Synchronize Path Schematic</b>	切换按钮, On (checked) 意味着原理图窗口跟踪驱动面板中已选的信号，所选信号被加亮
<b>Go to Time</b>	仿真接近规定的时限

<b>Synchronize Schematic</b>	<b>Path</b>	切换按钮, On (checked) 意味着原理图窗口跟踪驱动面板中已选的信号，所选信号被加亮
------------------------------	-------------	---

## 跟踪驱动程序

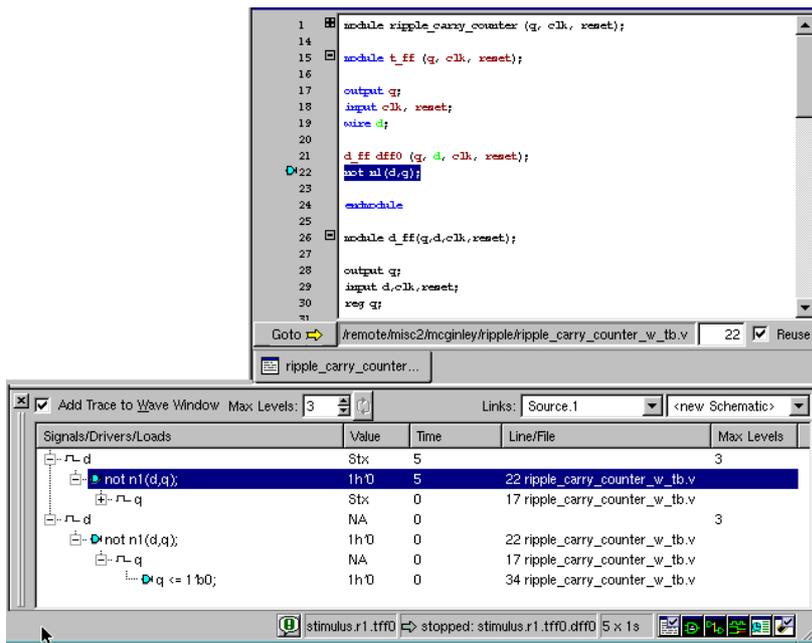
以下是跟踪驱动程序的方法：

- 在DVE窗口或面板中选定一个信号（如层次浏览器，数据面板，波形窗口，源窗口，列表窗口），单击驱动工具栏中的符号  。
- 在任意面板中选定一个信号，然后选**Trace >Trace Drivers**。
- 在波形窗口中，在波形上双击左键，例如从0到1，1到0的转变。
- 在波形窗口中选定一个信号，右键出现相关菜单，然后选**Trace Drivers**。
- 在源窗口中，双击一个信号。
- 在原理图窗口中选定一个信号，右键，然后在相关菜单中选**Trace Drivers**。

当一个驱动被跟踪时，如果当前顶层没有驱动面板，那么将会有个新的驱动面板被生成。如果驱动面板已存在，驱动信息会被添加到列表顶层。另外，源窗口中的第一个驱动会被加亮，间隙内会有蓝色的节点注解。如图9—2

注意：顶层框架中只允许执行一个驱动面板。

图9—2 波形窗口中选择一个transition显示驱动



## 跟踪载入

以下是跟踪载入的方法：

在DVE窗口或面板中选定一个信号（如层次浏览器，数据面板，波形窗口，源窗口，列表窗口或者波形窗口），单击载入工具栏的符号。

- 在任意面板中选定一个信号，然后选**Trace > Trace Loads**。
- 在波形窗口中选定一个信号，右键出现相关菜单，然后选**Trace Loads**。
- 在源窗口中选定一个信号，右键，然后在相关菜单中选**Trace Loads**。
- 在原理图窗口中选定一个信号，右键，然后在相关菜单中选**Trace Loads**。

当一个载入被跟踪时，如果当前首部没有驱动面板，将会有个新的驱动面板被生成。如果驱动面板已有，所有载入信息会被添加到列表顶层。

注意：顶层框架中只允许执行一个驱动面板。

## Vera 的功能

- 一个驱动器在被证实是不活动的之前是活动的。
- 通往控制器的控制通路，如果有部分估计是错误时，驱动器都会变成不活动的