**文档版本： Ver2.0**

**Lab3\_逻辑仿真工具环境\_VCS**

**请大家将自己的作业上传到发给讲师邮箱，word的命名采用“姓名\_第n次\_VCS”格式。**

**实验简介：**

通过这次实验，熟悉使用VCS工具编译仿真，理解逻辑仿真技术, 同时理解构成一个验证平台（以下称为Testbench）最基本的组件 – 激励生成、响应获取和结果比对。

**实验目的：**

1. 熟悉使用VCS工具的使用方法，包括GUI界面
2. 熟悉使用Makefile脚本使用VCS 工具进行编译和仿真
3. 对一个典型的基于Verilog的Testbench有一个大致的感受；

**实验准备：**

* 没有基础的同学，请预习文件中的脚本内容，如果其它仿真工具基础的同学，请参考脚本makefile其它后缀的文件。
* 使用VNC登录服务器，本次上机内容在以下路径，请自行拷贝到你的主目录下做上机题：

***/ASIC\_design/03-lab3\_logic\_simulation***

目录中，该目录中包含了本次实验所需要的相关代码和文档。

此处共有1个教程，4个文件夹，教程以full\_adder为例，大家根据教程做完示例后，可练习另外几个目录内的文件。

* 如果在上述准备工作中遇到任何困难，请及时与讲师联系解决。

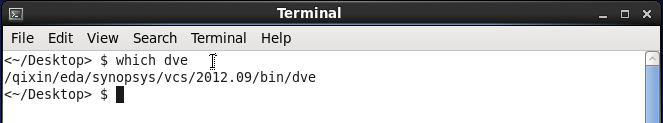
**实验步骤：**

1. **熟悉数字逻辑仿真工具: VCS**
   1. **VCS简介：**

VCS是Synopsys开发的EDA仿真工具，可提供高性能仿真引擎、约束条件解算器引擎、Native Testbench (NTB) 支持、广泛的 SystemVerilog 支持、验证规划、覆盖率分析和收敛以及完整的调试环境

* 1. **确认Linux系统环境中的VCS可用：**

在Terminal中输入命令：*which dve*

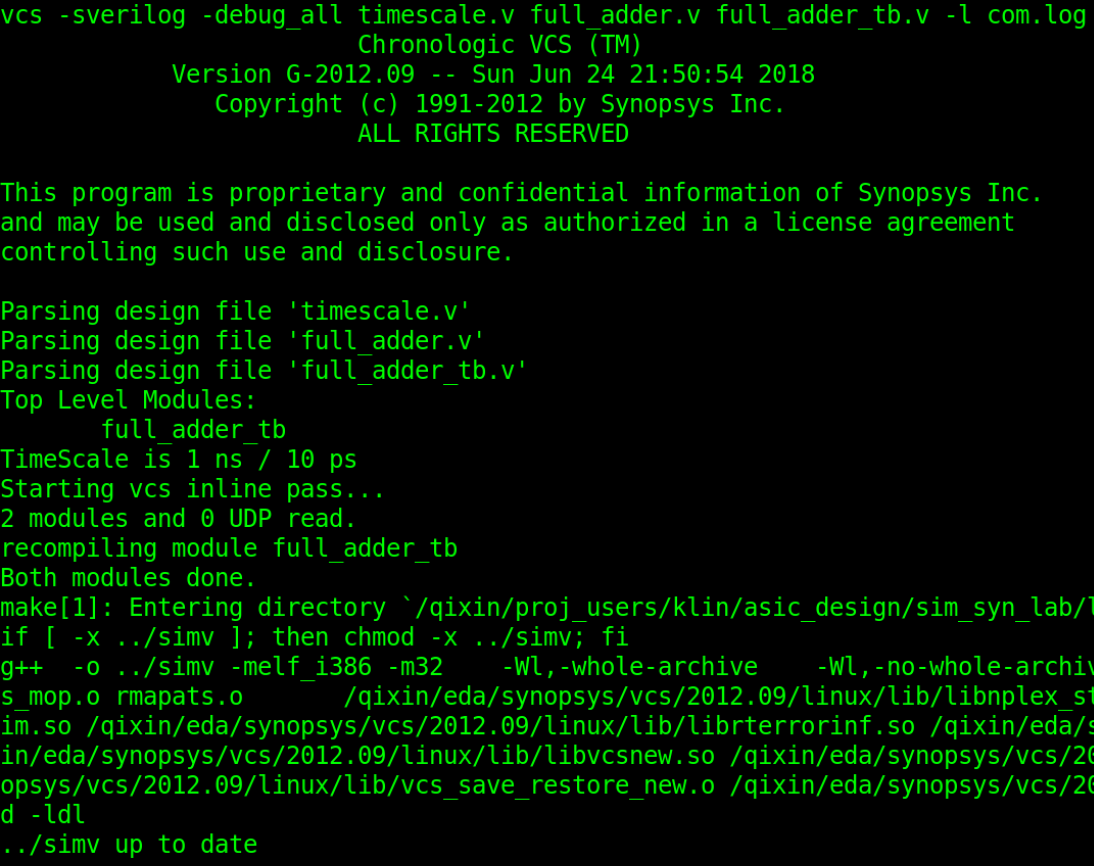


如果返回上图信息，则说明VCS可用，上图为安装路径。如果没有，说明VCS不可用，请检查系统配置是否正确，或者联系讲师或助教，及时获得帮助。

* 1. **编译生产simv文件：**

在Terminal中输入命令：

*vcs -sverilog -debug\_all timescale.v full\_adder.v full\_adder\_tb.v –l com.log*



执行完VCS的编译命令之后，在本目录下生成可执行过文件simv。

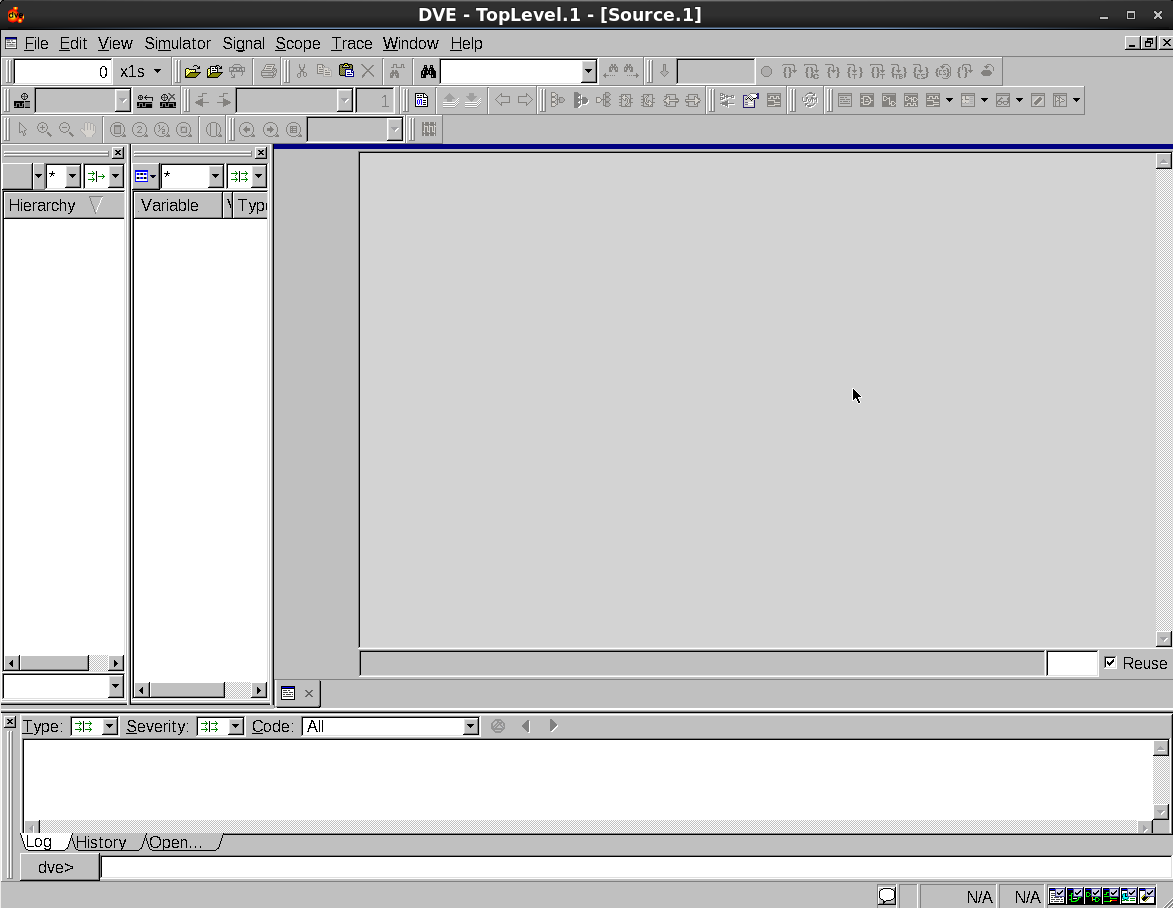
* 1. **打开vcs GUI界面：（dve命令是VCS仿真工具启动的图形化界面）**

在Terminal中输入命令：

*dve &* (&是为了让DVE在后台运行，Terminal还能输入命令)

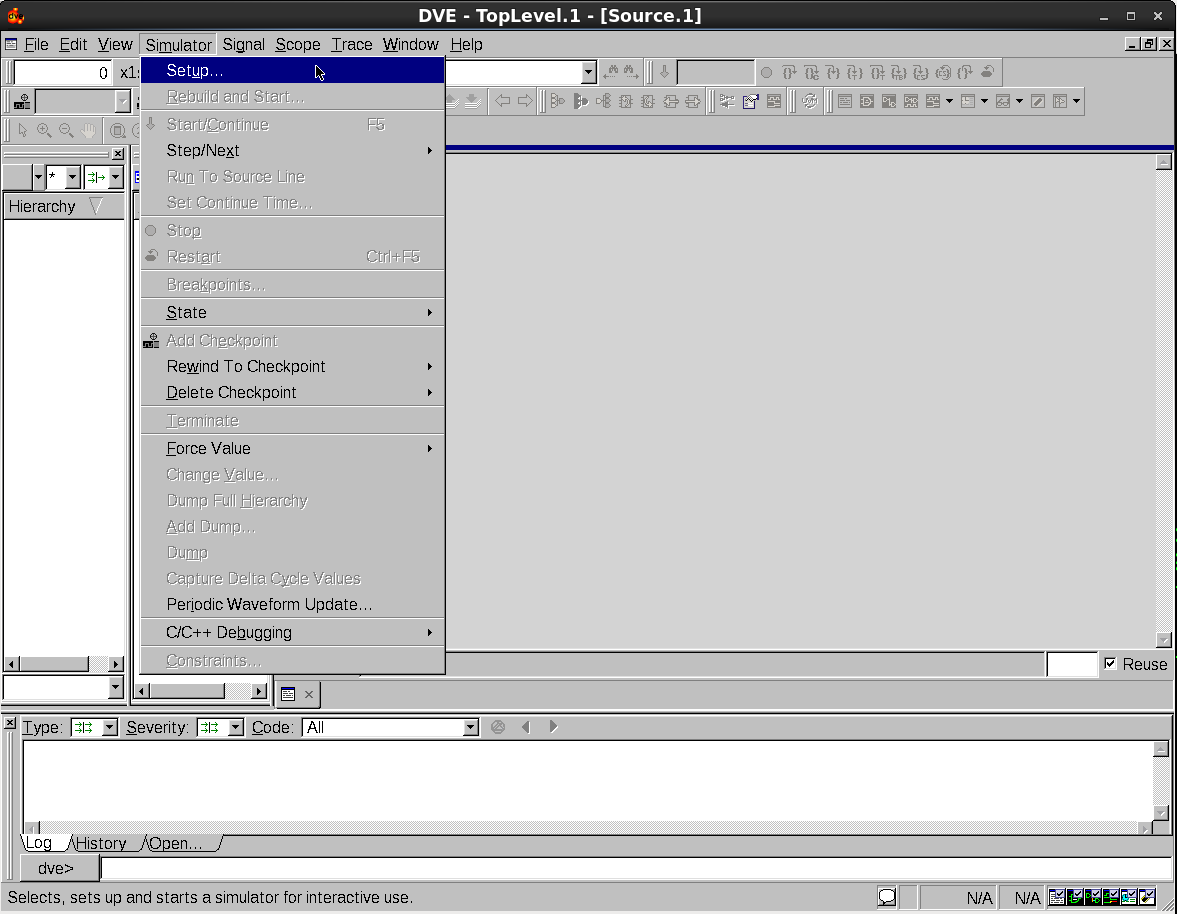


显示如下窗口，并点选红色方框内的所有窗口。



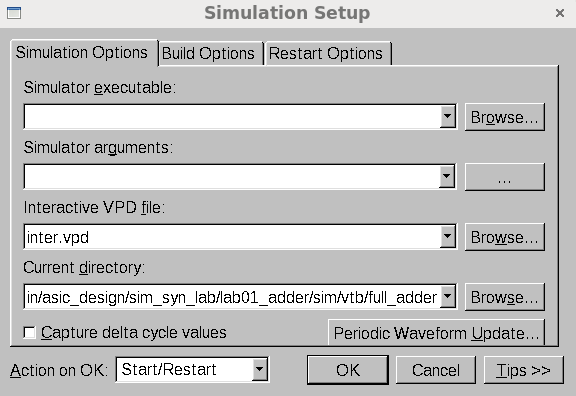
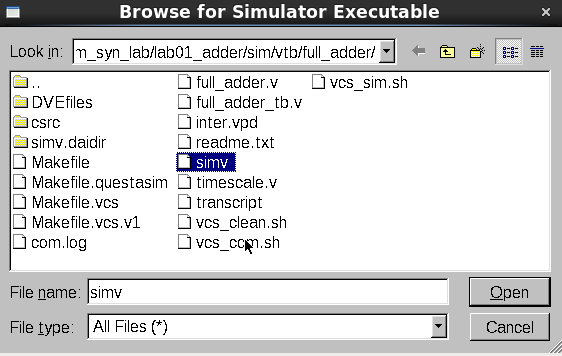
左键点选所有的窗口

* 1. **载入可执行文件simv与VPD文件，setup一个simulation**



左键点击Setup按钮

在弹出的对话框中，输入simv可执行文件以及vpd文件，点OK

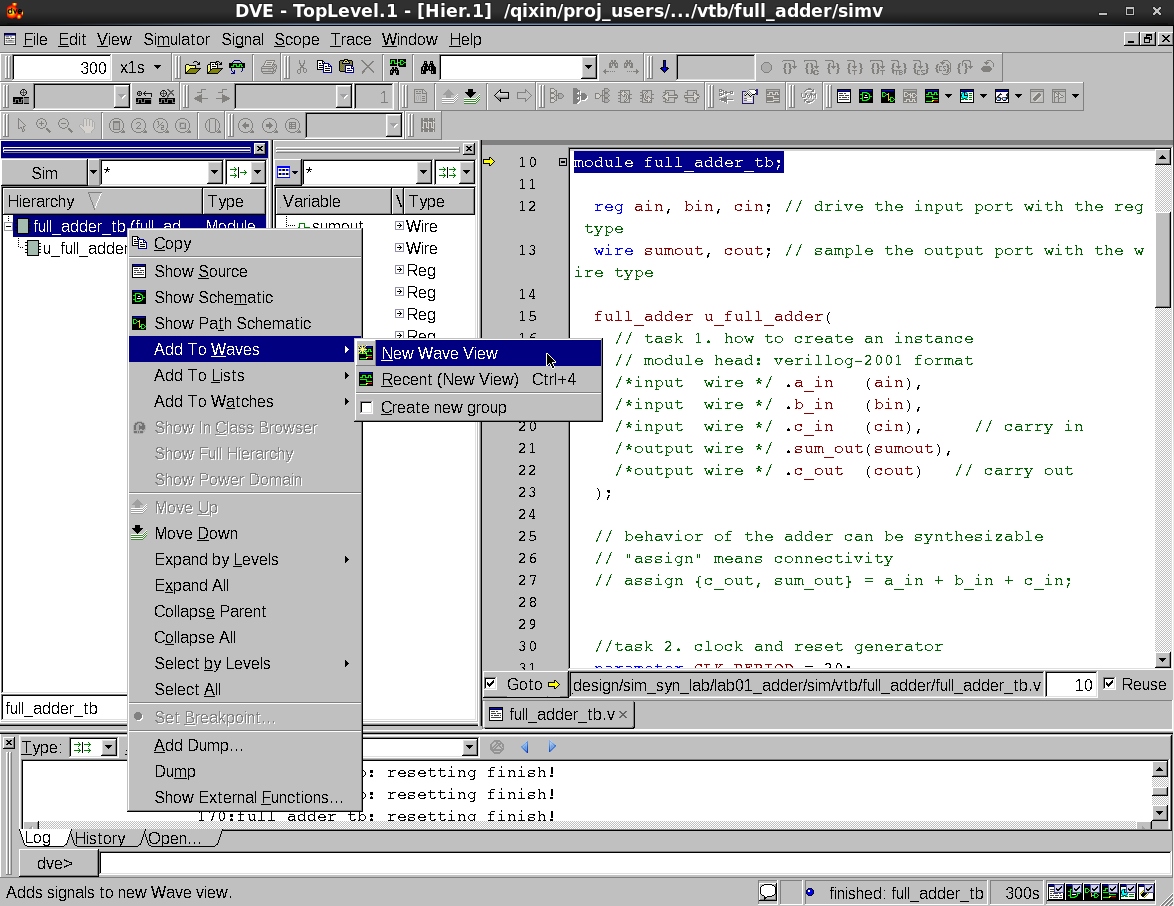
 

左键点击

simv可执行文件

左键点击Browse按钮

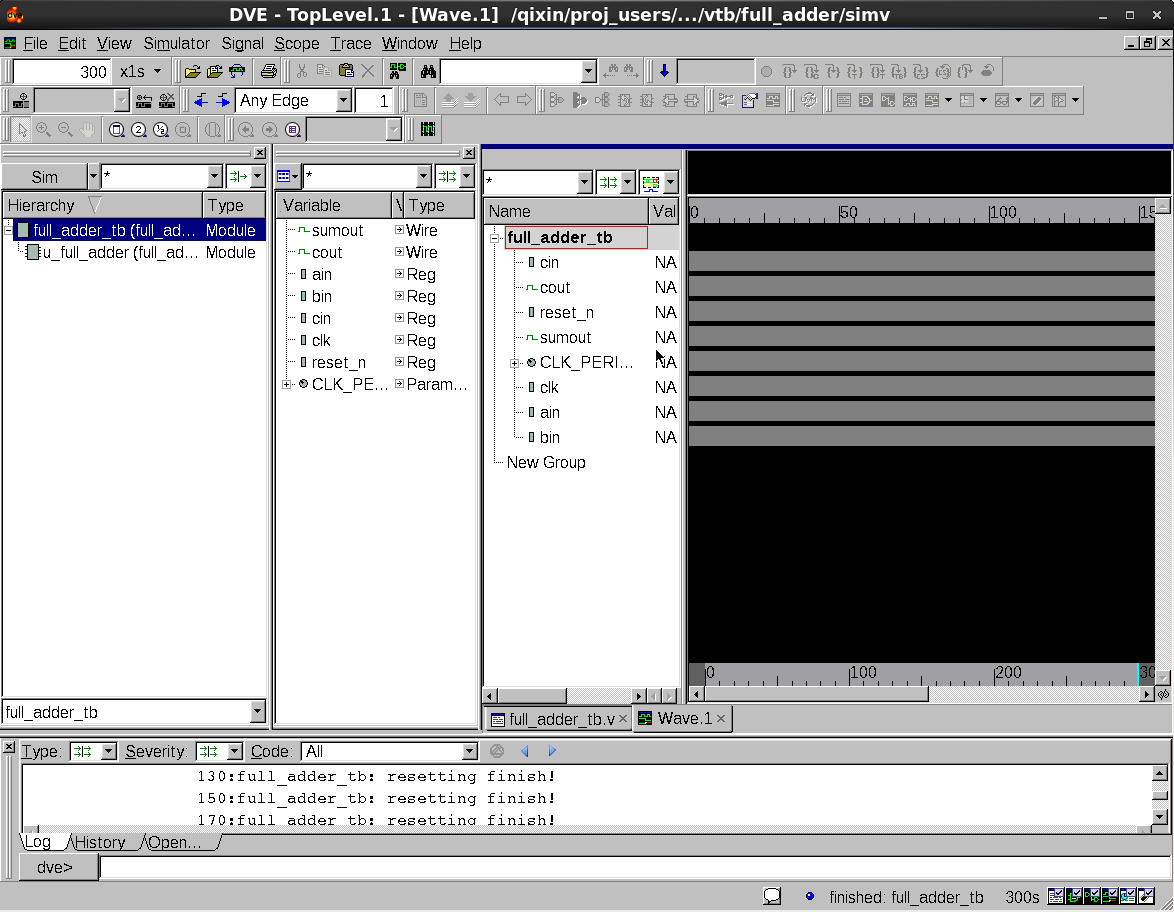
* 1. **选中需要查看波形的信号，将信号在波形中显示**



RTL源代码

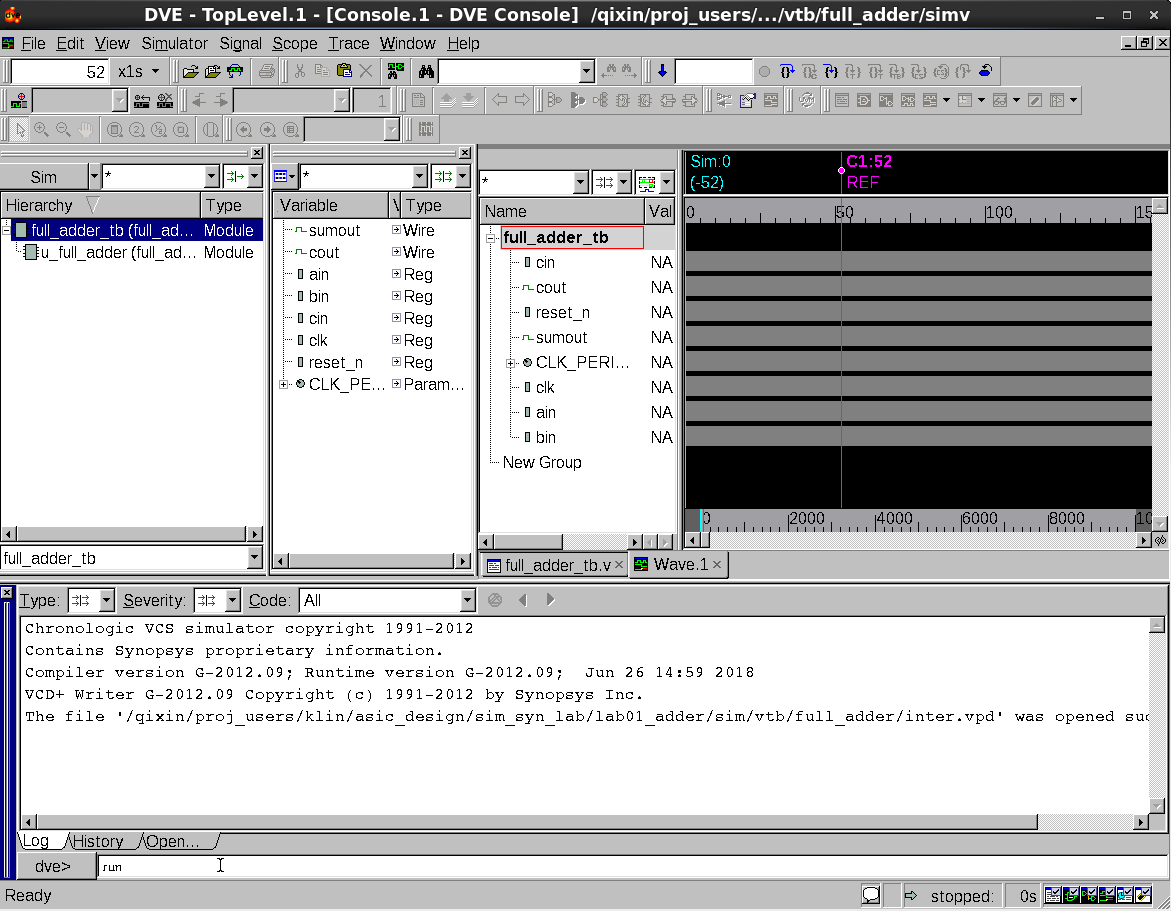
左键点击加减号“+/-”;

右键选择full\_adder\_tb，然后依次左键点击“Aadd To Waves”, “New Wave View”

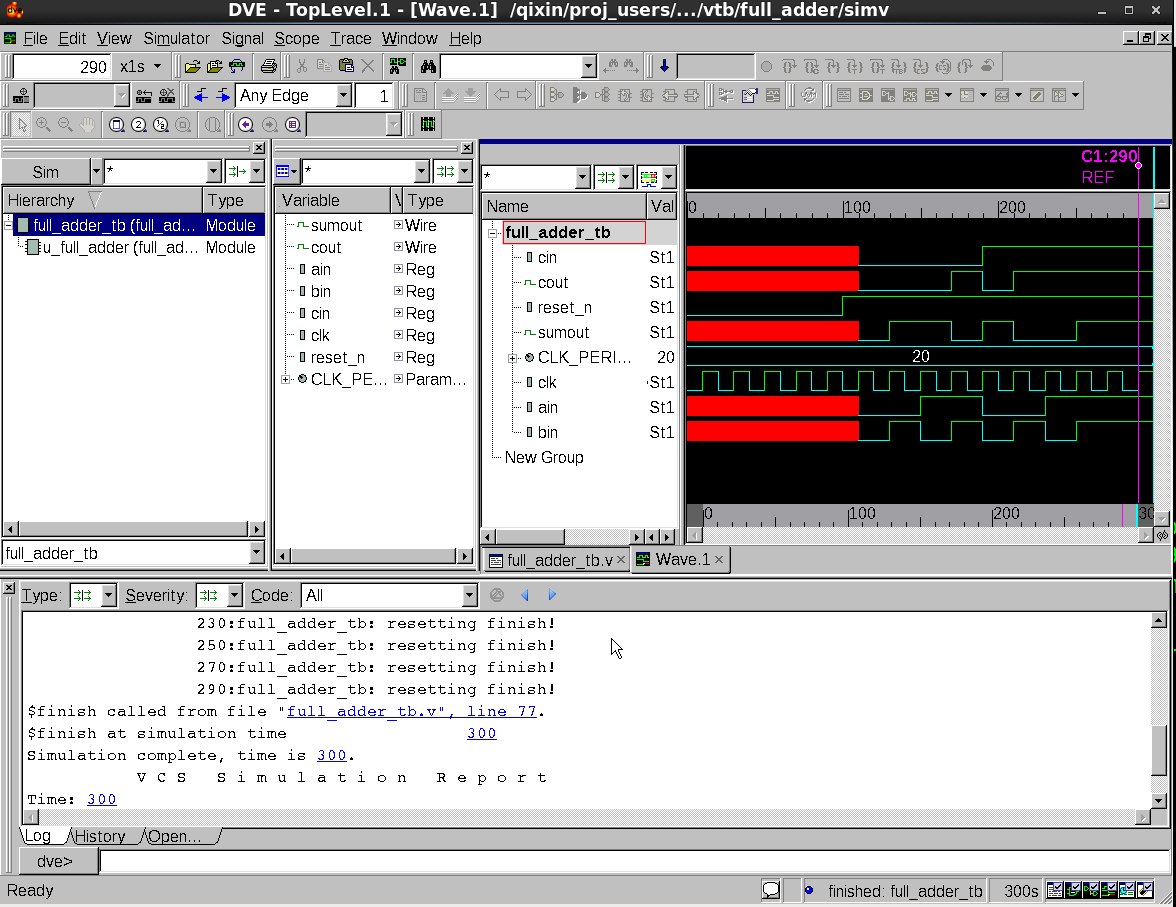


波形窗口

* 1. **在命令行里输入*run*命令可以跑完整个simulation**



在dve> 对话框中输入：run，然后回车



波形放大缩小

仿真波形

仿真日志

1. **使用命令行模型，调用VCS的命令进行编译和仿真**

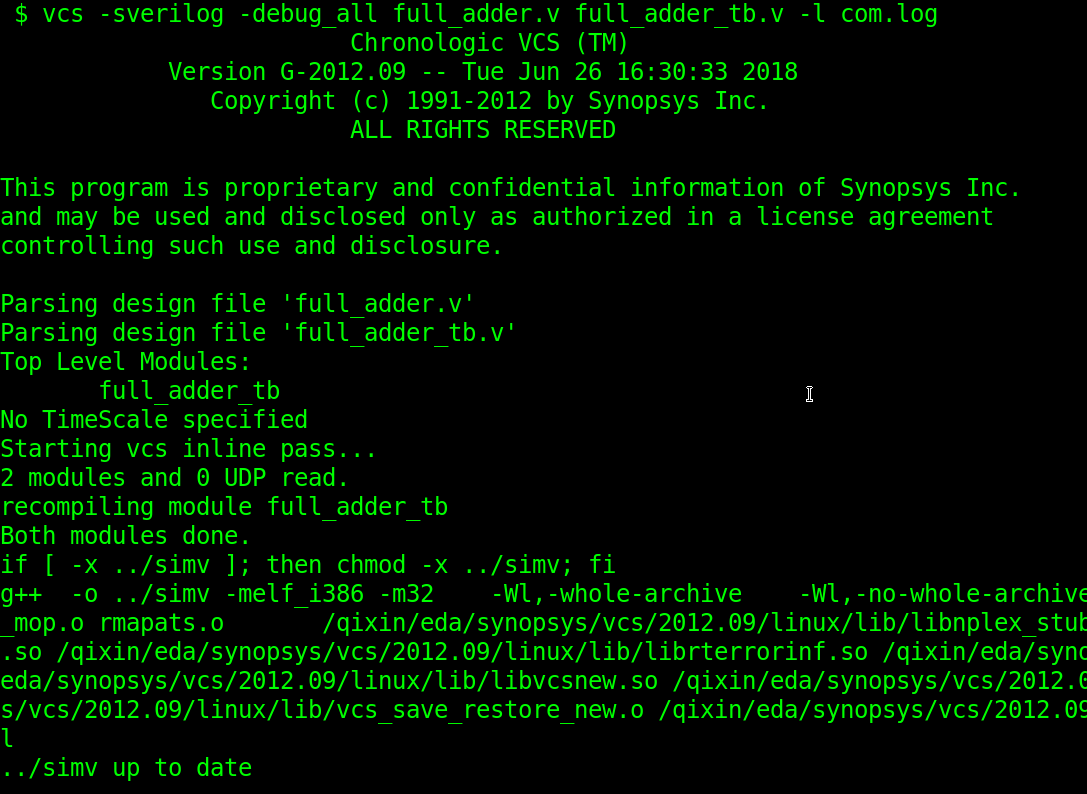
在实际工作中，一般很少用到GUI界面，除非是要用波形来**DEBUG**，通常情况都是通过命令行来进行编译和仿真源代码的，下面介绍命令行的步骤：

* 1. **删除刚才仿真过程的中间文件：**

*rm -rf simv\* csrc\* \*.tmp \*.vpd \*.key \*.log \*hdrs.v DVEfiles*

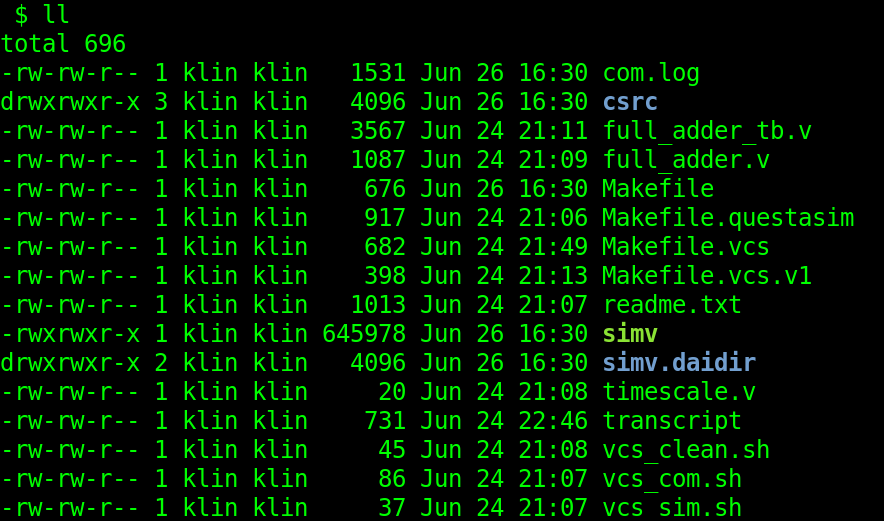
* 1. **编译RTL和Testbenhc源代码：**

*vcs -sverilog -debug\_all ./full\_adder.v ./full\_adder\_tb.v -l com.log*



仿真日志

临时文件

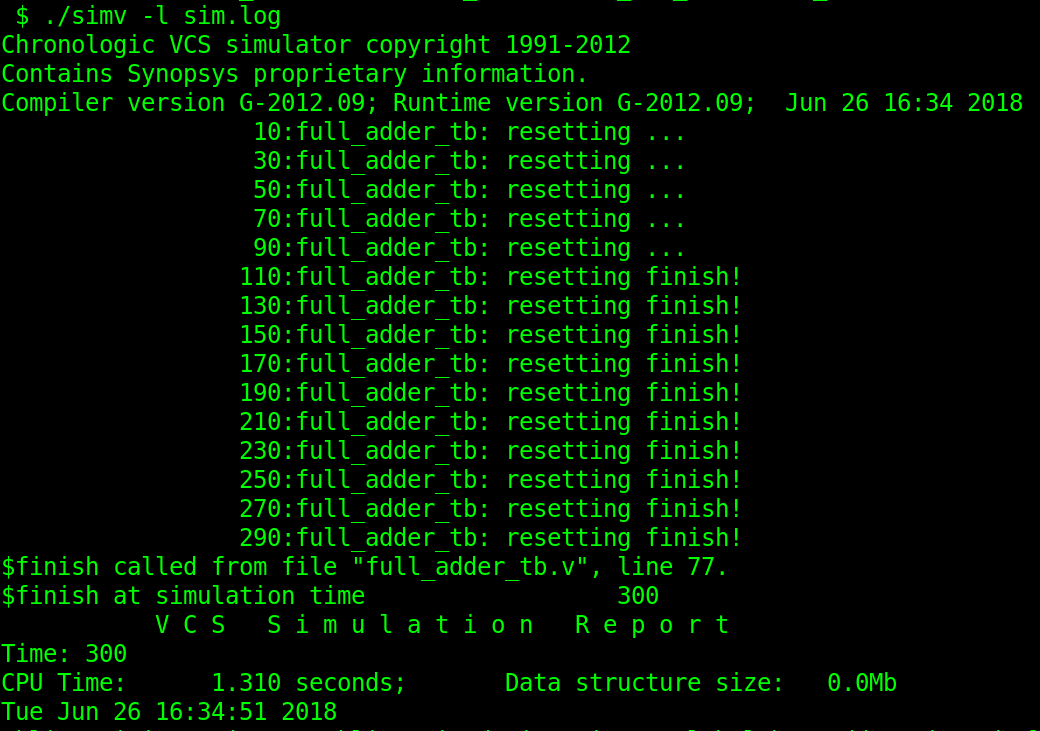


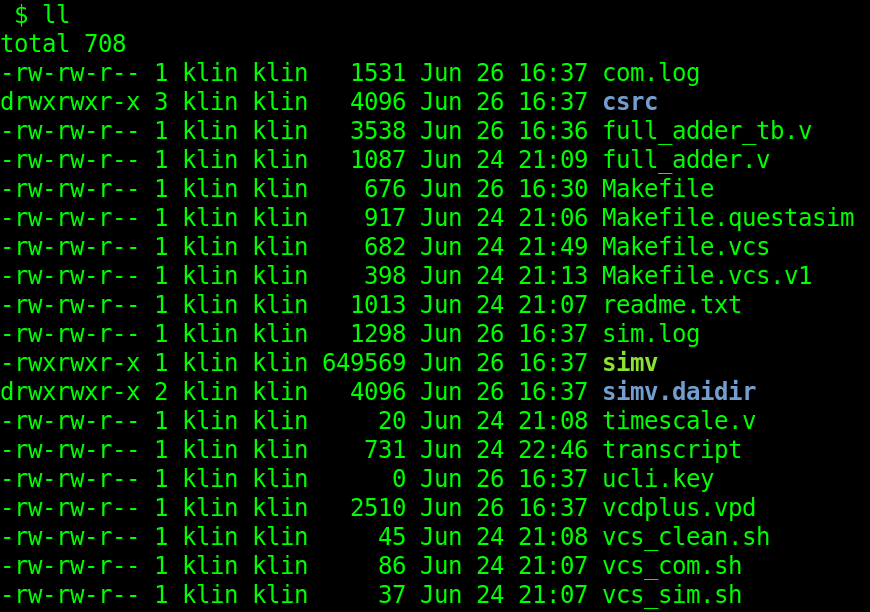
可执行文件和临时文件

* 1. **仿真，执行可执行文件simv:**

*./simv -l simv.log*

simv.log为仿真log文件，vcdplus.vpd为dump的波形文件

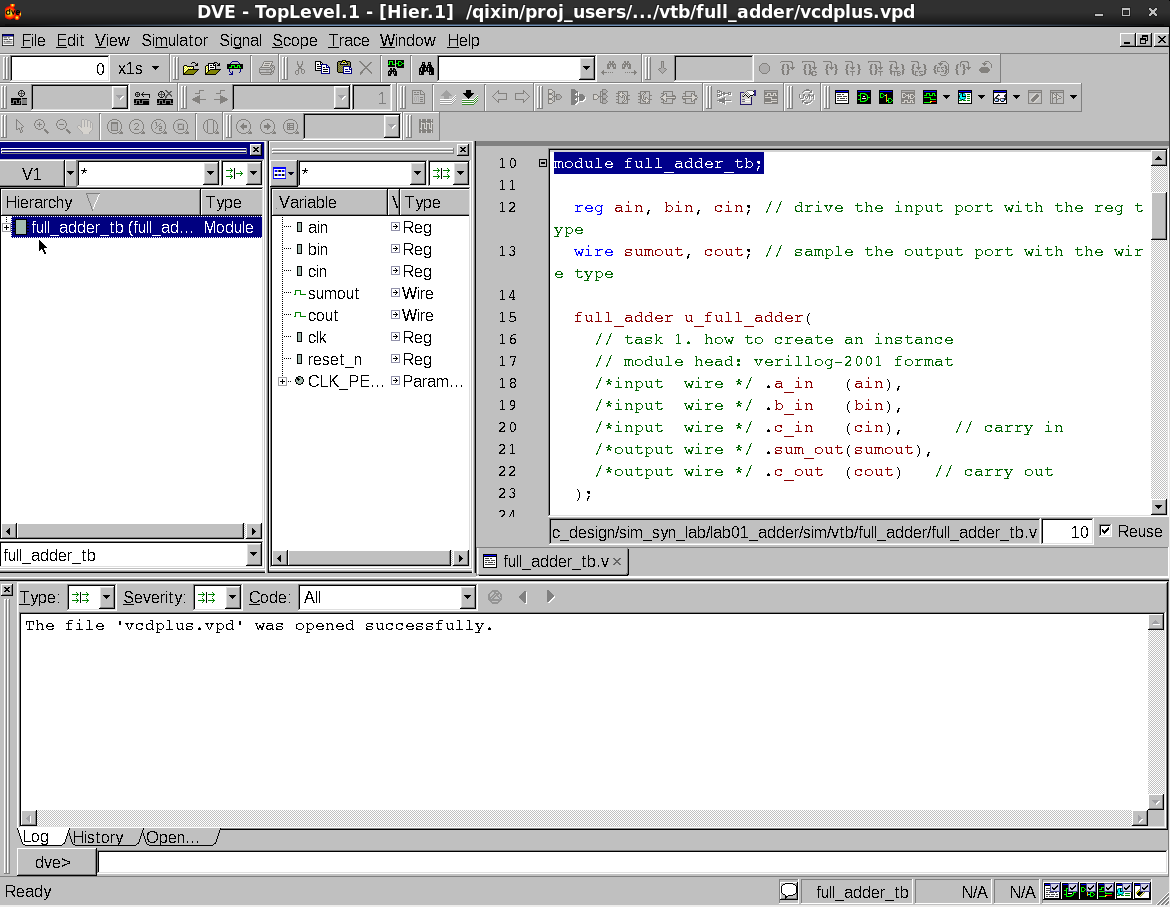




* 1. **打开GUI，进行DEBUG：**

*dve -vpd vcdplus.vpd &*

就可以按照上面介绍的打开波形图进行debug。



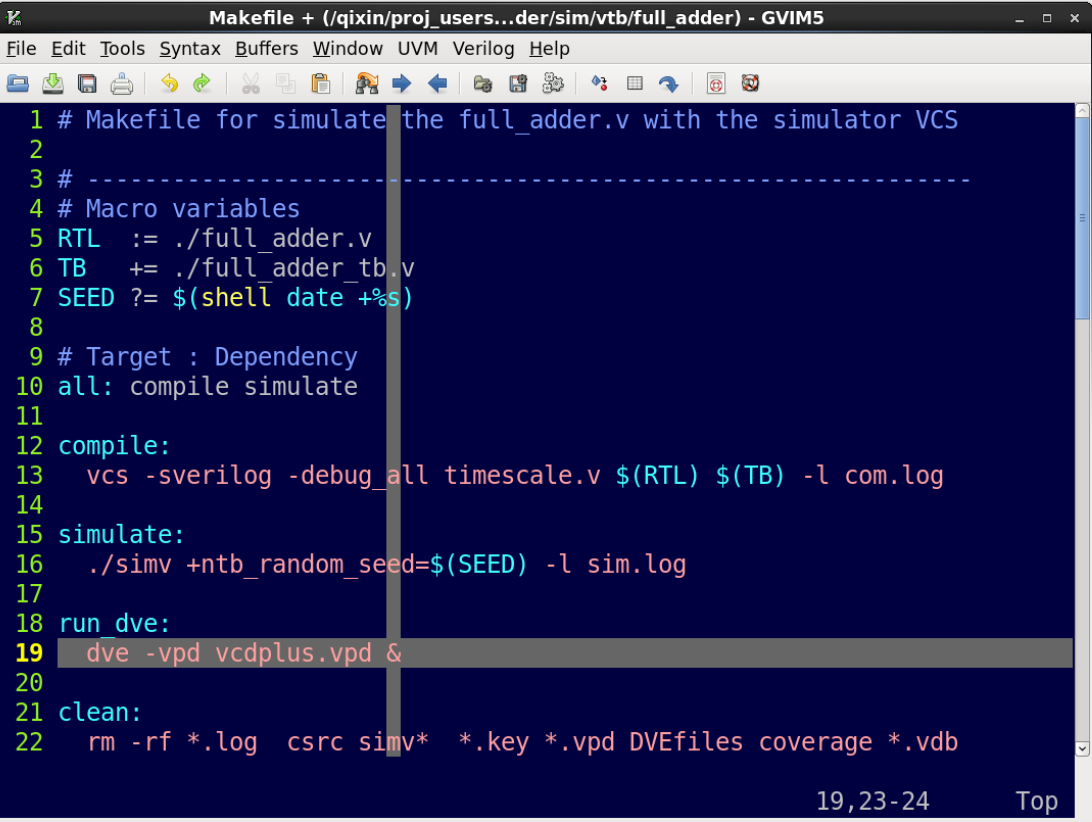
1. **使用Makefile脚本自动执行编译仿真命令**

虽然使用命令行可以方便执行仿真任务，但是实际工程中，文件量大，每次仿真都敲一遍所有命令，浪费时间，可以将这些语句写入自动化运行的脚本中，在需要的时候直接调用脚本执行命令即可自动完成编译和仿真的过程，这里介绍使用Makefile脚本来实现编译仿真

* 1. **用gvim来创建Makefile文件：**

*gvim Makefile*

* 1. **编辑该文件，写入如下命令：**



* 1. 如上图，当在terminal中输入make命令时，默认将执行目标all，而all依赖于compile和simulate的两个命令。首先执行compile命令，然后执行simulate命令。

如果单独执行compile和simulate命令，可以在terminal中输入：

*make compile; make simulate*

* 1. 当要打开GUI，进行DEBUG时，在terminal中输入：*make run\_dve*
  2. DEBUG结束，要删除中间文件，在terminal中输入：*make clean*

说明：在每一次修改源代码之后，重新运行编译和仿真时，都需要将当前的仿真目录删除干净，也就是把上一次编译和仿真生成的文件全部删掉；不然，旧的文件会影响下一次的仿真。

**实验问题：**

**请回答一下问题。**

1. 介绍一下什么是逻辑仿真工具，它的主要用途是什么？简单介绍你熟悉的逻辑仿真工具（5分）

|  |
| --- |
| 答： |

1. 逻辑仿真工具的使用时分几步？（5分）

|  |
| --- |
| 答： |

1. 如何利用逻辑仿真工具进行查看波形？（5分）

|  |
| --- |
| 答： |

1. 什么是Makefile，它有那些好处？在芯片设计过程中，如何利用Makefile的自动编译功能，简化逻辑仿真工具的使用？（5分）

|  |
| --- |
| 答： |

5.附上做这4个电路的仿真脚本和实验步骤截图，并各自说明这4个电路的功能。（每个电路20分，共80分）