E 课网 – 半导体集成电路在线教育平台

半导体集成电路教育内容和产品提供商

数字设计 EDA 工具

Verdi 与仿真器联合仿真 – Dump FSDB 波形文件

文档版本： Ver3.0

最后修改日期： 2022-04-17

修改人： William

实验简介：

Verdi Debug 系统适用于数字电路的设计与验证，它提供了一系列工具使工程师能够快速的了解和掌握设计的结构，并提供行为分析方法，提取整个区域里完整的、全局的、相关的图示，极大的提高 Debug 效率，从而有效的提高生产率。

Verdi 支持各种硬件描述语言如 VHDL、Verilog、SystemVerilog，同时集成支持验证描述语言 Vera、e 语言、SVTB、UVM 等，并且支持 Assertion 描述语言 SVA、psl 等。数字 IC 设计的整个验证过程都可以在 Verdi 中分析与调试。

Verdi 的主要功能模块有：

- nTrace：这是一套源码追踪工具，工程师可以查看设计层次结构和源码（支持 Verilog、VHDL、SystemVerilog、三者的混合代码以及电源设计语言如 CPF/UPF），跟踪 driver/load 信号，标注波形信号值等功能。
- nSchema：这是一套电路原理图萃取工具，可以根据相关的 RTL 或者 Gate-level 的代码智能的创建层次化的电路原理图，方便跨层次结构进行信号追踪。
- nWave：这是一套波形显示工具，提供了一个易于理解和非常直观化的视窗来帮助工程师分析信号。
- nState：有限状态机提取及分析
- nCompare：不同设计阶段的仿真结果的比较
- Assertion Debug：帮助工程师分析 Assertion 结果
- Testbench Debug：帮助验证工程师 Debug Testbench

Verdi 的工具不局限于此，更多了解请查阅 Verdi 的用户手册。

Verdi 为用户提供的强大功能可以极大的提高工程师的工作效率，因此掌握如何使用 Verdi 是每个工程师的必修课。

实验目的：

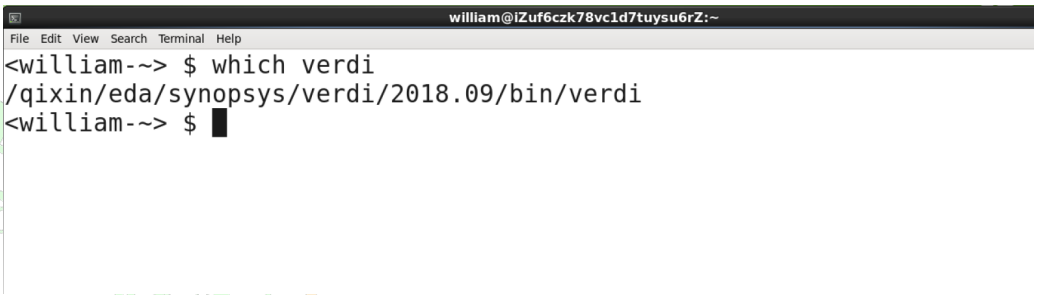
- 熟悉 Verdi 与逻辑仿真工具 VCS 的联合仿真的方法：Dump FSDB 波形文件

实验准备：

- 确认 Linux 系统环境中的 Verdi 可用：

  在 Terminal 中键入如下命令

  ```
  which verdi
  ```



  看看是否返回一条如上图的信息。

如果有，则说明系统的 Verdi 可用，并且可以看到其安装路径。

- 确认实验数据：

在 Terminal 中键入如下命令：

```
cd
cd verification/basic/labs/verdi_lab
```

检查改目录下的文件清单文件清单：

```
adder32.v
adder32_tb_random.v
Makefile
rtl.f
tb.f
```

实验步骤：

1. 了解波形文件

我们现在所接触的仿真波形文件主要有

- **Wlf 文件：** WLF 波形日志文件，是 QuestaSim 的专用文件。这个 wlf 文件只能是由 QuestaSim 来生成，也只能通过 QuestaSim 来显示。在用 QuestaSim 做仿真时，仿真结束都会生成一个*.wlf 的文件(默认是 vsim.wlf)。下次就可以通过通过 QuestaSim 直接打开这个保存下来的波形。vsim -view vsim.wlf -do run.do 其中 run.do 中的内容为要查看的波形信号。要强调的是不是一个通用的文件文件格式。

- **VCD 文件：** VCD 是一个通用的格式。 VCD 文件是 IEEE1364 标准(Verilog HDL 语言标准)中定义的一种 ASCII 文件。可以通过 Verilog HDL 的系统函数$dumpfile，$dumpvars 等来生成。我们可以通过$dumpvars 的参数来规定我们抽取仿真中某个特定模块和信号的 VCD 数据。它主要包含了头信息，变量的预定义和变量值的变化信息。正是因为它包含了信号的变化信息，就相当于记录了整个仿真的信息。可以用这个文件来再现仿真，也就能够显示波形。另外我们还可以通过这个文件来估计设计的功耗。因为 VCD 是 Verilog HDL 语言标准的一部分，因此所有的 verilog 的仿真器都要能实现这个功能。因此我们可以在 verilog 代码中通过系统函数来 dump VCD 文件。另外，我们可以通过 QuestaSim 命令来 dump VCD 文件，这样可以扩展到 VHDL 中。具体的命令： vcd file myfile.vcd vcd add /test/dut/* 这个就生成一个含 dut 下所有信号的 VCD 数据信息。我们在使用来进行仿真 vsim -vcdstim myfile.com test;add wave /*;run -all;

- **FSDB 文件：** fsdb 文件是 verdi 使用一种专用的数据格式，类似于 VCD，但是它是只提出了仿真过程中信号的有用信息，除去了 VCD 中信息冗余，就像对 VCD 数据进行了一次 huffman 编码。因此 fsdb 数据量小，而且会提高仿真速度。我们知道 VCD 文件使用 verilog 内置的系统函数来实现 的，fsdb 是通过 verilog 的 PLI 接口来实现的。$fsdbDumpfile,$fsdbDumpvars 等 另外，在 VCS 仿真器中还有一种 VCD＋的数据

格式 VPD，详细情况参照 VCS 的使用。注意：WIF：波形中间格式；WLF：波形日志文件。由于在 QuestaSim 下只能打开 WLF 文件 使用 QuestaSim 行命令 vcd2wlf 将 VCD 文件转化为 WLF 文件。

2. Verdi API

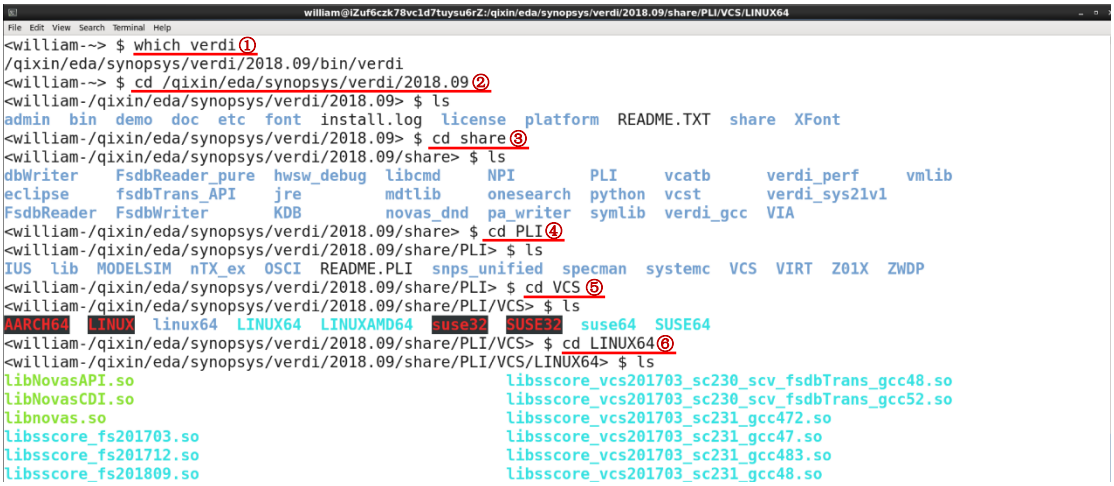
前面介绍过，如果要 dump FSDB 波形文件，需要调用由 Verdi 提供的 API 接口方法，常用的 API 有：

- **$fsdbDumpfile** – 指定 FSDB 文件名，限制 FSDB 文件大小
- **$fsdbDumpvars** – 转存信号指定实例和深度的变化
- **$fsdbDumpon/$fsdbDumpoff** – Dump 波形开/关
- **$fsdbDumpSVA** – dump assertion 开关

由于这些 API 是由 Verdi 提供的，并不是 VHDL 、Verilog 或者 SystemVerilog 中规定的，因此必须在仿真的时候添加由 Verdi 提供的库文件，该库文件根据仿真器以及操作系统的类型不同而不同，所以第一步就是要确认要使用的库文件。

3. 寻找 Verdi 提供的库文件

根据下面的步骤找出 verdi 提供的库文件



① 执行 **which verdi** 命令

可以看到系统返回了 verdi 命令的路径，这里的目的是找出 verdi 的安装路径：

/qixin/eda/synopsys/verdi/2018.09

② 执行 **cd /qixin/eda/synopsys/verdi/2018.09** 命令进入 verdi 的安装目录

③ 执行 **cd share** 命令，进入 share 目录

④ 在 share 目录下，可以看到有一个 PLI 目录，执行 **cd PLI** 命令进入该目录。在 PLI 目录下，可以看到里面包含了几个比较重要的目录：

- IUS – Cadence 的 NC-verilog 仿真器所需要的库文件
- MODELSIM – Mentor 的 modelsim 和 Questasim 仿真器所需要的库文件
- VCS – Synopsys 的 VCS 仿真器所需要的库文件

⑤ 我们研究 VCS，这里执行 **cd VCS** 命令进入 VCS 目录。这个目录中有三个目录，分别用于不同的操作系统：

- LINUX – 系统为 linux32 位
- LINUX64 – 系统为 linux64 位
- LINUXAMD64 – CPU 为 AMD 的 64 位 linux 系统

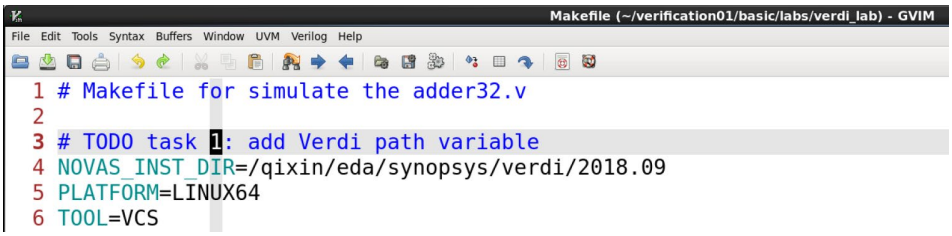
⑥ 我们服务器上的系统为 linux64 位，执行命令 **cd LINUX64** 进入 LINUX64 目录中。在该目录下可以看到一些.so 文件和 novas.tab、pli.a 等文件，这里我们需要 **novas.tab** 和 **pli.a** 文件。

4. Verdi 与 VCS 联合仿真

下面，我们用一个具体的例子来看看如何使用 Questasim 仿真器利用 Verdi 提供的 API Dump FSDB 波形文件，以及如何使用 Verdi 打开 FSDB 波形文件。

① 进入实验目录

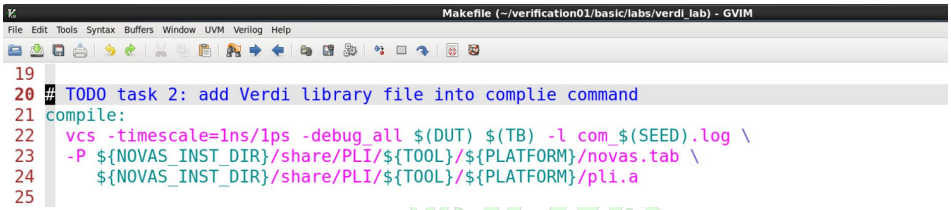
② 使用 Makefile 脚本来调用 Questasim 的仿真命令
在实验目录下执行命令 **gvim Makefile** 打开 Makefile 文件，找到
 # TODO task 1:
定义三个变量：



- NOVAS_INST_DIR – verdi 的安装路径
- PLATFORM – 操作系统类型
  - Linux32 位系统 PLATFORM=LINUX
  - **Linux64 位系统 PLATFORM=LINUX64**
  - AMD 系统 PLATFORM=LINUXAMD64
- TOOL – 选用的仿真器
  - **VCS 仿真器 TOOL=VCS**
  - Questasim/Modelsim TOOL=MODELSIM
  - NC-verilog TOOL=IUS

我们的服务器为 Linux64 位系统，本次使用 VCS 做为仿真器。

③ 在 Makefile 中添加 VCS 的编译和仿真命令
在 Makefile 文件中找到 **# TODO task 2:**
在 **vcs** 仿真命令（第 22 行）中加入 **-P** 选项（第 23，24 行），并且后面紧跟 verdi 的库文件的完整路径，目的是在执行 API 的时候可以找到相关的库。



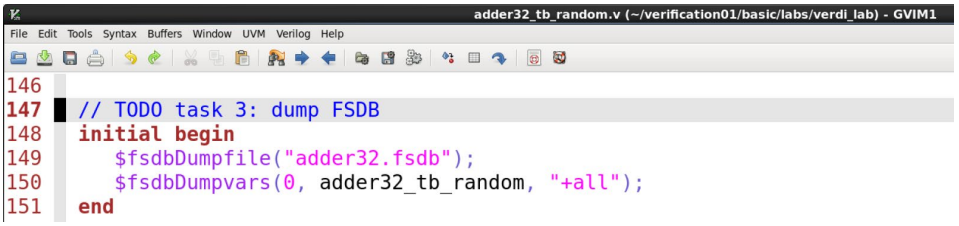
注意：在第 22，23 行的末尾使用了续行符“\”，请确保在“\”后面没有任何的字符，任何的空白字符也是不可以的（空格/Tab 等）

④ 在 Testbench 的最顶层，添加 API

使用命令 **gvim adder32_tb_random.v** 打开 **adder32_tb_random.v** 文件，该文件是 Testbench 的最顶层文件，在该文件中找到：

 # TODO task 3:

调用**$fsdbDumpfile** 和**$fsdbDumpvars API**。



- **$fsdbDumpfile(“adder32.fsdb”)**
  表示将 dump 到的 FSDB 文件波形文件命名为 **adder32.fsdb**。
- **$fsdbDumpvars(0, adder32_tb_random, “+all”)**
  这条语句的意思就是 dump 最顶层包括其所有子模块的所有变量的波形。有三个参数：
  - 参数 1 表示层次深度，0 表示所有层次的变量；
  - 参数 2 表示对哪一个模块进行 dump 波形，adder32 是实验的最顶层；
  - 参数 3 表示 dump 变量的类型，+all 表示所有类型的变量。

  其他参数的含义这里不做过多介绍，详情请查询 verdi 的使用说明。

⑤ 在 Makefile 中添加启动 verdi 的命令

在 Makefile 文件中找到 **# TODO task 4:**

按照下图添加使用 Verdi 打开波形的命令（46~49 行）

- **rtl.f** 文件是 RLT 的文件列表文件
- **tb.f** 文件是 TB 的文件列表文件
- 使用 cat 命令，将 RTL 的文件列表和 TB 的文件列表合并成 **all.f** 文件
- 在 verdi 命令中，使用**-f all.f** 指定要加载的文件列表，使用**-ssf** 指定要打开的 FSDB 波形文件，这里是 adder32.fsdb 文件

由于这里打开的波形文件 adder32.fsdb 是需要生成的，因此在打开波形文件之

前，需要先调用 VCS 进行仿真产生这个波形文件。

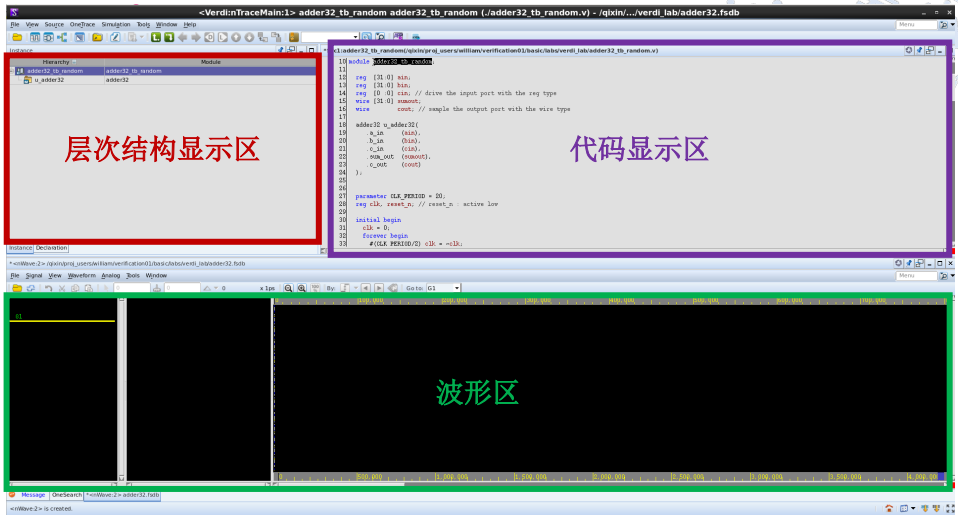
⑥ 运行仿真

在编辑完 Makefile 之后，即可保存文件。执行 **make** 命令运行仿真。待仿真执行完毕之后，可以在当前目录下看到有一个 **adder32.fsdb** 的波形文件产生。

```
<william-~/verification01/basic/labs/verdi_lab> $ ls
adder32.fsdb          adder32.v           csrc       novas_dump.log  sim_1650715861.log  simv.daidir  ucli.key
adder32_tb_random.v   com_1650715859.log  Makefile   rtl.f           simv                tb.f         vcdplus.vpd
<william-~/verification01/basic/labs/verdi_lab> $
```

⑦ 启动 verdi

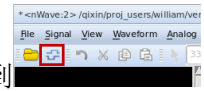
执行命令 **make verdi_fsdb**



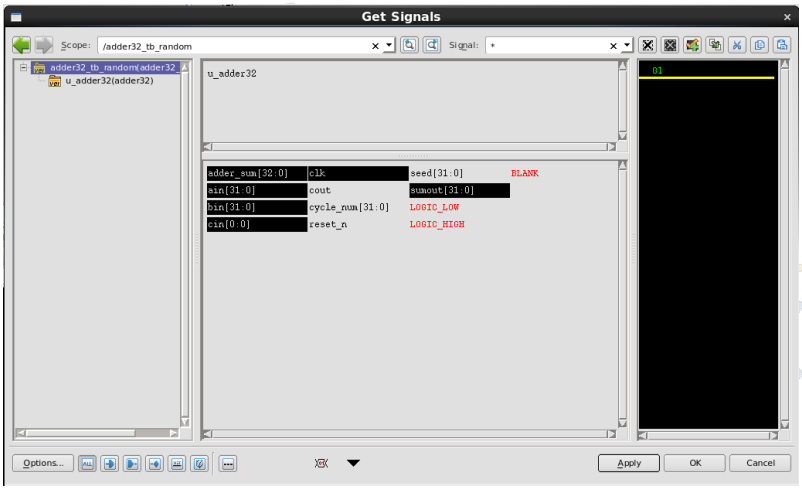
如上图，

- 红色方框区域为仿真模块的层次结构区域，可以点击查看整个仿真模块的每一个子模块。
- 紫色方框区域为代码显示区域，通过可显示每个模块的具体代码。是进行 Debug 的重要区域。
- 绿色方框区域为波形显示区域，可将想要观察的信号添加到此区域。是进行 Debug 的重要区域。

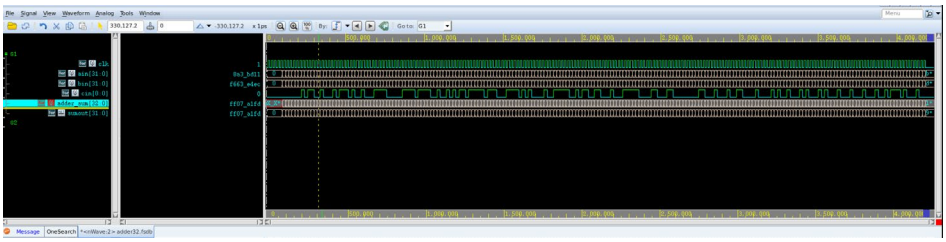
⑧ 查看波形

在绿色方框区域中，找到  图标并点击：

出现下图信号选择界面，可在左侧窗口选择要查看的模块，在中间窗口选择要查看的信号（点选信号，背景变色表示已经选择）。



选完后点击右下角 Apply/OK，即可将选择的信号添加到波形窗口中。



Tips：

1. 用鼠标先**点击一下波形窗口**，然后按下键盘上的“**g**”键，可以快速打开信号选择界面。
2. 在代码窗口，使用鼠标**选中要观察的信号**，然后使用“**Ctrl+w**”可快速将选中的信号加入波形窗口中。
3. 在代码窗口，使用鼠标**直接**将要观察的信号**拖拽**到波形窗口，可快速在波形窗口中添加波形。