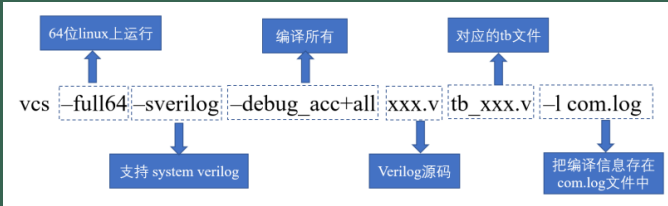


VCS的使用

举例

- 编译源代码
- 跑仿真 `/simv -l sim.log`
- 查看波形 `dve -vpd xxx.vpd &`



编译

- `-l filename.log` 把VCS编译和运行信息的保存在filename.log文件中
- `-full64` 以64位linux系统运行
- `-sverilog` 支持system verilog 语法
- `-debug_all`或`-debug_acc+all` 编译命令
- `+define+<macro_name>=<value>` 命令行进行宏定义
- `+v2k` 支持verilog2001特性
- `-o filename` 可自行改变仿真文件的名称
- `-v` 导入库文件的verilog模型
- `+vpdfile+ filename` 指定vpd波形文件名字默认为vcdplus.vpd
- `-y` 参考的库目录
- `+libext+<.v>` 使用VCS搜索库文件中的.v文件
- `+incdir+<directory>` 指定VCS搜索include指令只用的包含文件的目录
- `-f filename` 将所有目录下的文件的设计文件与参考库写入，一次编译仿真
- `-h` 列出常用的开关和简要功能

仿真

- `-gui` 以DVE图形化界面进行仿真
- `-ucli` 以ucli命令行模式进行仿真
- `-R` 编译后立即执行仿真

后处理

- `dve &` 后台打开DVE
- `dve -vpd filename` 一步打开dve文件并且读入波形文件

testbench 中后处理系统函

`$vcdpluson(level, num, ber, an, odule, _instance)` 记录波形形成的.vpd文件

```
initial begin
  $vcdpluson;
end
```

覆盖率统计

- `-cm cond+ tal+ lin+ fsm +path` 统计各种code的覆盖率开关
- `-cm _name` 覆盖率统计结果的文件名

门级网表仿真