simulation

指定stdcell,io,memory等模块的仿真模型

```
set_simulation_library_source -f ./all_model.f
```

设置VCS安装目录

```
setenv VCS_HOME /tools/syn/vcs_mx-2022.6-SP1/
```

运行仿真命令指定仿真器是vcs

```
run_testbench_simulations \
  -parallel_simulations 10 \
  -simulator vcs \
  -compilation_options {-kdb +define+debussy} \
  -simulator_options {-kdb -debug_access+all -debug_region=cell+lib -lca} \
  -extra_verilog_files {../rtl/b.v ../rtl/c.v} \
  -extra_top_modules module_name_list  \
  [-use_design_view_per_simulation on | off ]
```

另外需要说明的是如果是前仿真，因为没有sdf的缘故出现时序违例的问题导致仿真不通过，可以进入tessent自动生成的仿真目录修改vcs.simulation_scrip在vcs命令添加nospecify或no_notifier和notimgcheck

```
run_testbench_simulations
[-design_name design_name]
[-design_id design_id]
[-pattern_id pattern_id]
[-report_list]
[-select pattern_name_glob_list]
[-exclude pattern_name_glob_list]
[-generate_scripts_only | -run_only]
[-parallel_simulations {1 | 2...MAX_INT | maxcpu}]
[-expected_miscompare_count int]
[-simulation_id string]
[-simulator questa | vcs | incisive]
[-simulation_timeout time | unlimited]
[-keep_simulation_data on | off | on_failure]
[-store_simulation_waveforms on | off]
[-simulation_run_commands simulator_commands]
[-waveform_configuration_commands simulator_command_list]
[-compilation_macro_definitions {macro[=value] ... }]
[-simulation_macro_definitions {macro[=value] ... }]
[-compilation_options option_list]
[-simulator_options option_list]
[-extra_verilog_files list_of_files]  #这可以单独加入tessent 不能处理的文件用于仿真，比如一些加密文件
[-extra_top_modules module_name_list] #可用于设置一些额外共同task或force
```

```
[-simulation_output_directory simulation_output_directory]
[-simdut_output_directory simdut_output_directory]
[-simdut_server] [-generate_simdut_files_only]
[-use_design_view_per_simulation on | off ] #设置是否用全文件，不用v_interface之类的；开启后RTL会一些编译，各自TB单独编译
[-no_wait]
```