培训 07--EDA 使用技巧

预备工作

完成培训 01--开发环境搭建中的进阶配置，能够在终端执行 bash、make、xvlog、xelab、xsim、code、git 等命令

Vivado 工程脚本化构建

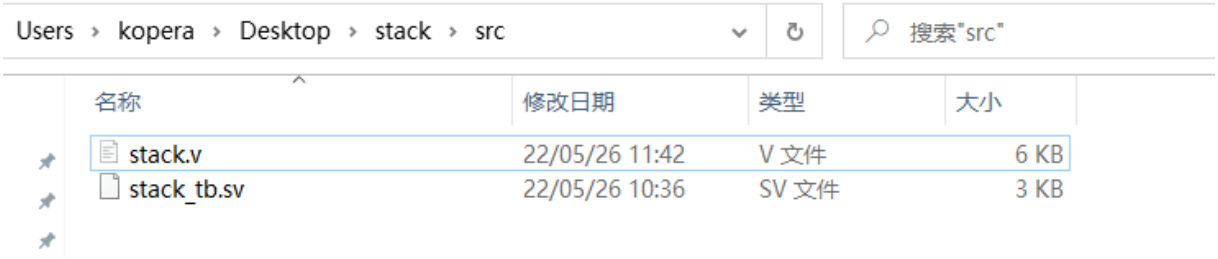
很多时候，我们需要对自己编写的 Verilog 小模块进行评估（功能仿真、综合原理图）和确认，通过 GUI 界面进行工程构建的方式时比较费时的。下面介绍两种脚本化的工程构建方式

使用 Batch Mode 进行功能仿真

(这里以一个小模块 stack(LIFO)为例，任意一个小工程都可以)

建立一个工作目录，例如 `stack`（如果你手头没有这样的小工程，可以去本文附录获取代码）

将所有源代码放入工作目录的 `src` 文件夹，



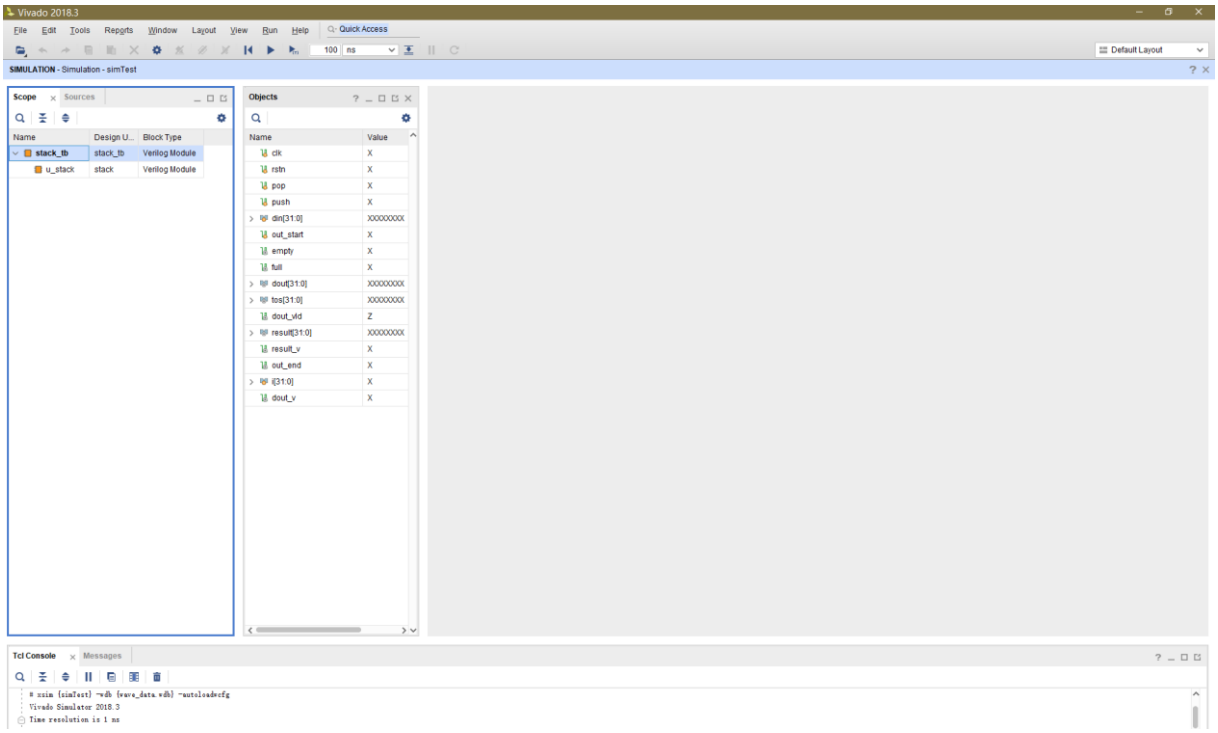
在 **Bash 命令行**中依次输入下列命令（在工作目录执行），

```
xvlog --sv --work mylib src/*.v src/*.sv
xelab --debug typical -s simTest mylib.stack_tb
xsim simTest -gui -wdb wave_data.wdb
```

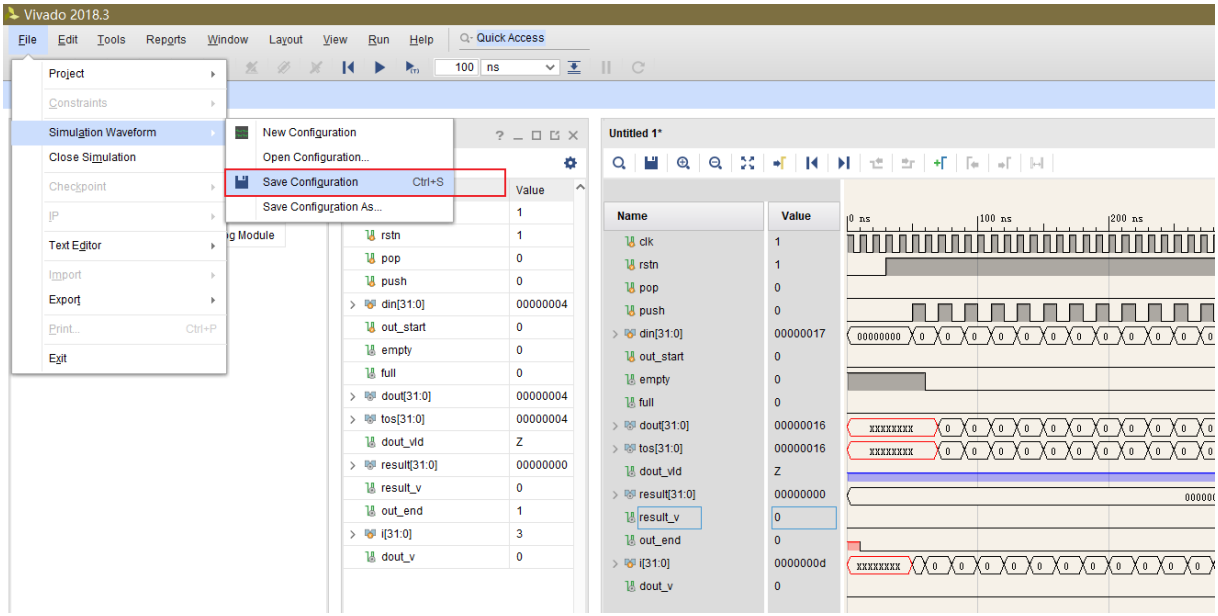
（注意：在 linux 下可以直接执行，在 win 下的 cmd 中需首先执行 `bash` 命令）

（其中 `stack_tb` 需要替换为实际仿真工程的的最顶层，一般是 tb 文件）

执行上述命令，**编译通过时**将自动调出仿真界面，



在该界面可以添加波形进行仿真，



下次启动 xsim 时，执行，

```
xsim simTest -gui -wdb wave_data.wdb -view ../wave_config.wcfg
```

即可直接调出上次的仿真配置

使用 Project Mode 进行仿真和综合

使用 Vivado 的 Batch Mode 进行仿真时，无法使用 Relaunch Simulation 功能。如果对源代码进行了修改，则只能重新运行一遍命令序列才能查看新的仿真。

为此，基于 Vivado 内建的 Tcl 指令，可使用自动化的 Project Mode 进行功能仿真。通过为仿真和综合指定不同的顶层，让观察波形、查看原理图更加方便。

编写 Tcl 脚本，命名为 run_sim.tcl ，放在工作目录下

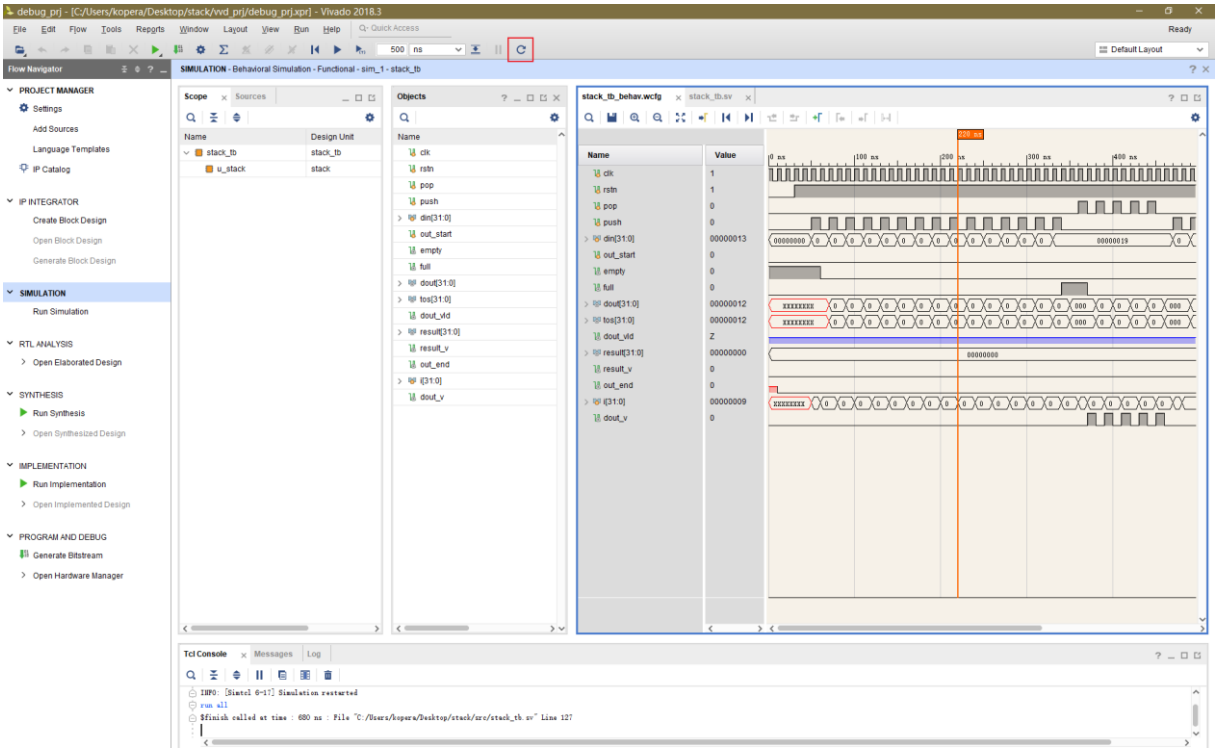
```
set sim_top [lindex $argv 0]
set synth_top [lindex $argv 1]

create_project debug_prj

add_files -fileset sources_1 -norecurse
[glob ../src/*.v ../src/*.sv]

set_property top $synth_top [current_fileset]
update_compile_order -fileset [current_fileset]
set_property elab_link_dcps false [current_fileset]
set_property elab_load_timing_constraints false [current_fileset]

set_property top $sim_top [get_filesets sim_1]
update_compile_order -fileset sim_1

set_property -name {xsim.simulate.runtime} -value {30ns} -objects
[get_filesets sim_1]
set_property -name {xsim.elaborate.load_glbl} -value {false} -
objects [get_filesets sim_1]

launch_simulation -mode behavioral
save_wave_config ../${sim_top}_behav.wcfg
start_gui
```

在 **Bash 命令行**中依次输入下列命令

```
mkdir vvd_prj
cd ./vvd_prj
vivado -mode tcl -source ../run_sim.tcl -tclargs stack_tb stack
```

（最后的两个参数 **stack_tb** 和 **stack**，需要替换为实际工程的的仿真最顶层和设计最顶层）

编译无误时自动调出仿真界面，且 Relaunch Simulation 功能可用



点击 Open Elaborate Design 可以查看综合出的电路原理图（默认芯片 xc7vx485tffg1157-1），

这里给出一组示例脚本的组合（make、shell、tcl 配合），实现输入一个名称，开始一个工作流（期间只需编写 RTL，其余工作均通过脚本执行）

模板

创建一个模板目录 Template，并创建下列文件

Makefile

```
#指定综合的顶层模块
SYNTH_TOP = _MODULE_NAME

#指定仿真的顶层模块
SIM_TOP = $(SYNTH_TOP)_tb

#指定新建工程的序号
INDEX = $(N)

Test_dir = batch_sim
Prj_dir = vvd_prj

OS_Type = $(shell uname)


ifeq ($(OS_Type),MSYS_NT-10.0)
    Nut_path = C:/Users/kopera/Desktop/Nutstore/Coding/Verilog
else
    Nut_path = /home/kopera/Nutstore/Coding/Verilog
endif

run:
    sh ./Touch_linux.sh

echo:
    echo $(Nut_path)

#新建文件
create:
    mkdir src
    touch src/$(SIM_TOP).sv
    touch src/$(SYNTH_TOP).v

link:
    cd $(Nut_path) && mkdir $(SYNTH_TOP) && mkdir $(SYNTH_TOP)/src
    ln src/* $(Nut_path)/$(SYNTH_TOP)/src/

#编译
build:
    @if [ -e $(Test_dir) ]; then rm -rf $(Test_dir); fi
    mkdir $(Test_dir)
    cd $(Test_dir) && xvlog --sv --work
```

run_sim.tcl

```
set sim_top [lindex $argv 0]
set synth_top [lindex $argv 1]

create_project debug_prj

add_files -fileset sources_1 -norecurse
[glob ../src/*.v ../src/*.sv]

set_property top $synth_top [current_fileset]
update_compile_order -fileset [current_fileset]
set_property elab_link_dcps false [current_fileset]
set_property elab_load_timing_constraints false [current_fileset]

set_property top $sim_top [get_filesets sim_1]
update_compile_order -fileset sim_1

set_property -name {xsim.simulate.runtime} -value {30ns} -objects
[get_filesets sim_1]
set_property -name {xsim.elaborate.load_glbl} -value {false} -
objects [get_filesets sim_1]

launch_simulation -mode behavioral
save_wave_config ../${sim_top}_behav.wcfg
start_gui
```

load_sim.tcl

```
    set sim_top [lindex $argv 0]

    start_gui
    launch_simulation -mode behavioral
    close_wave_config -force [current_wave_config]
    open_wave_config ../${sim_top}_behav.wcfg
```

上述三个脚本是基础，在 Win 下和 Linux 下使用不同的脚本调用即可开启一个工作流。

在 Linux 下

编写一个 shell 脚本调用 Makefile

Touch_linux.sh

```
    make run
```

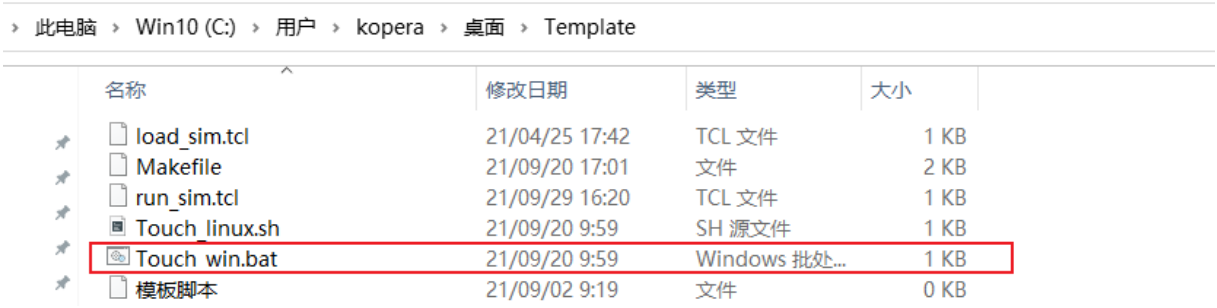
即可在上级目录创建一个工作目录并开始一个工作流

在 Win 下

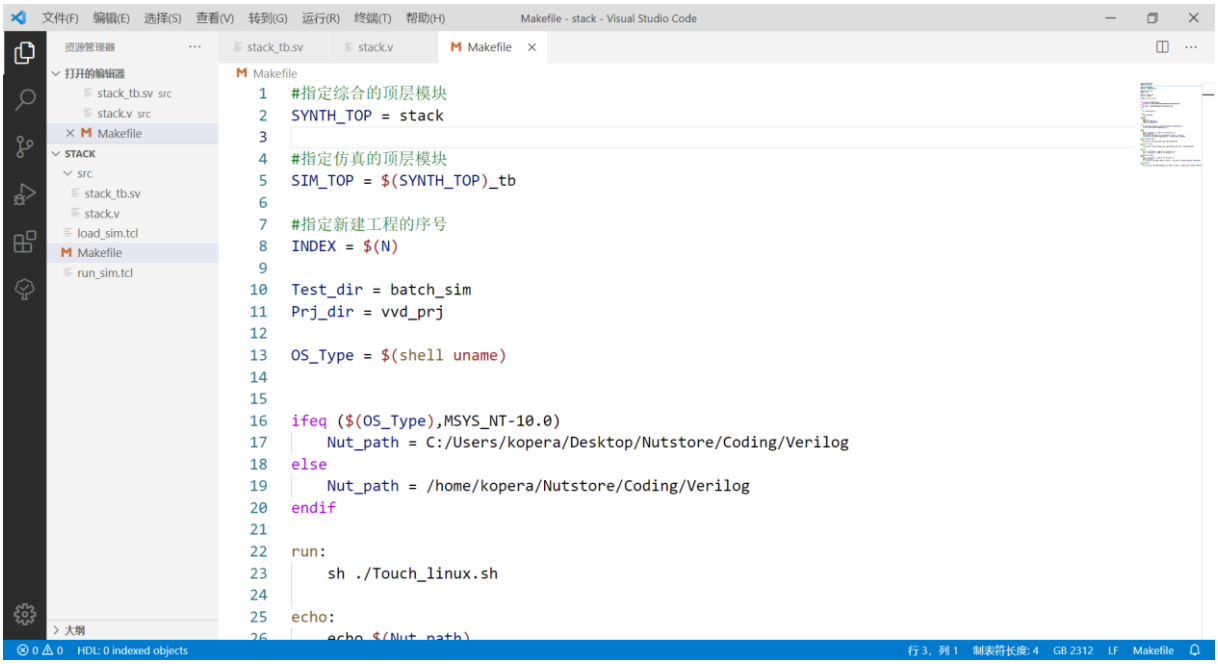
编写一个 cmd 脚本调用 Makefile

Touch_win.bat

```
    @echo off
    echo Please input the name...
    set /p name=
    md ..\%name%
    cp .gitignore load_sim.tcl Makefile run_sim.tcl ..\%name%

    cd ..\%name%
    sed -i "s/_MODULE_NAME/%name%/" Makefile

    make create
    code .

    pause
```



在 Win 下双击执行 Touch_win.bat



（Makefile 中的顶层名会被自动替换）

功能示例

1、编译代码

```
make build
```

2、调试仿真

```
make debug
```

3、启动上次的工程

```
make launch
```

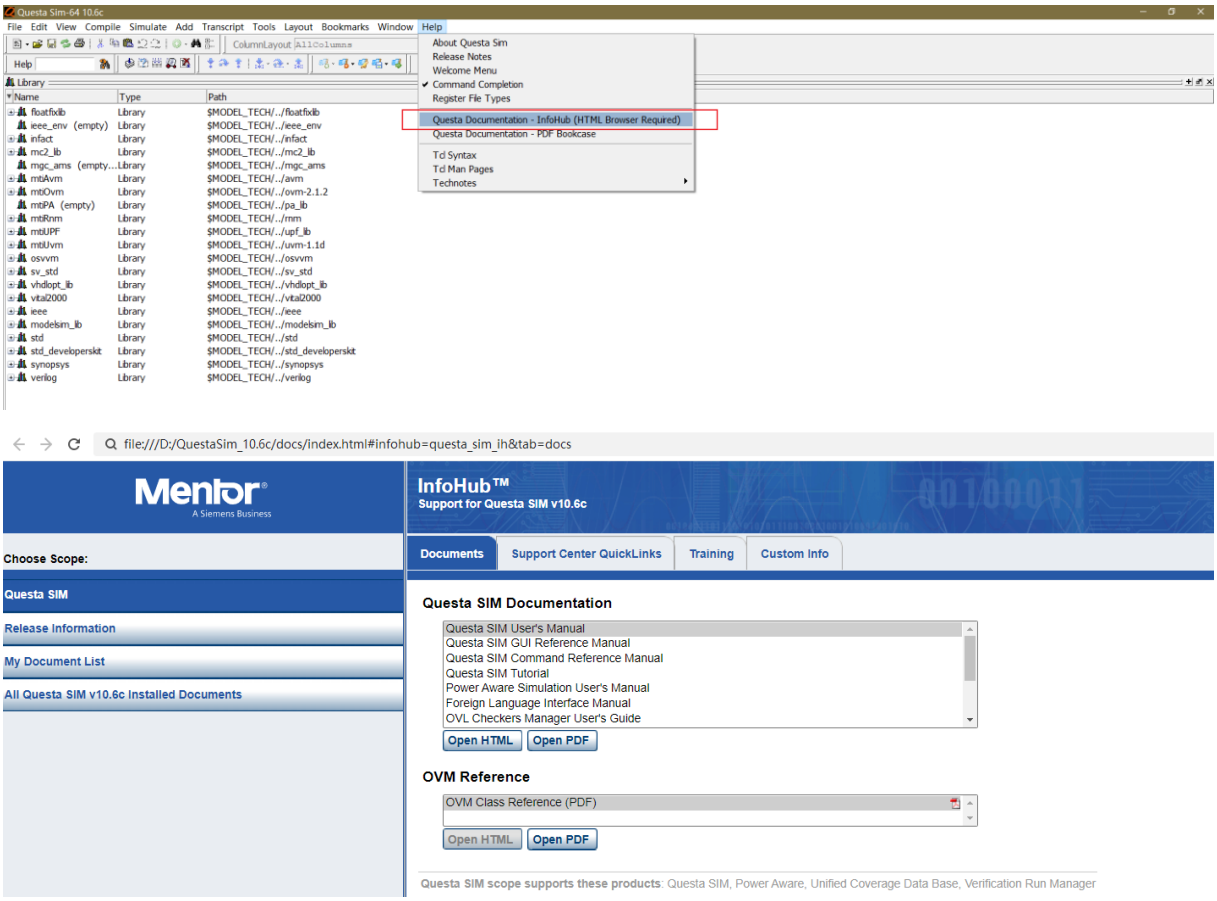
4、清空工程文件（保留源码）

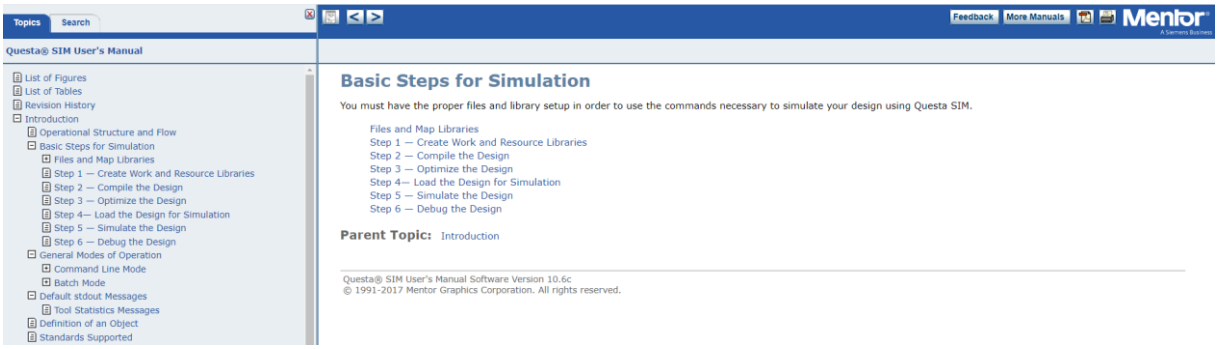
```
make clean
```

QuestaSim 的使用技巧

参考手册

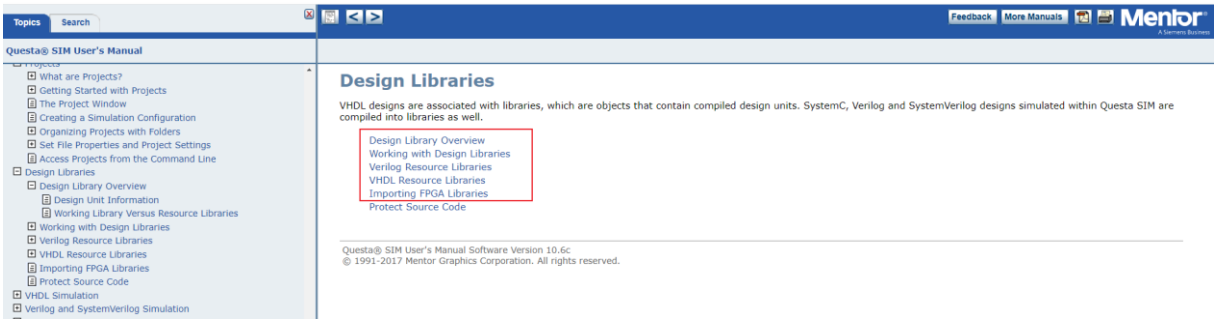
在这里查阅文档





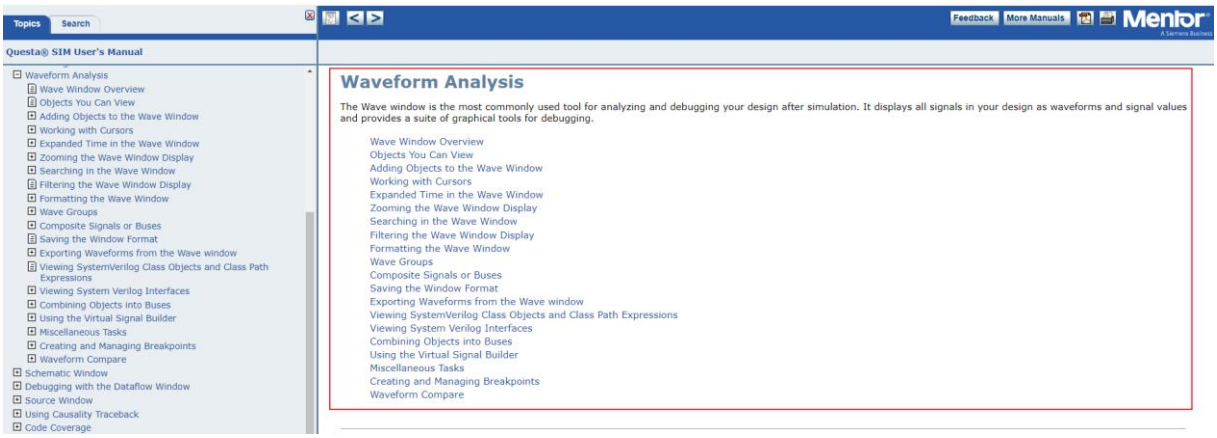
Tutorial ：Basic Simulation

2、User’ s Manual ：Design Libraries



Tutorial ：Working With Multiple Libraries

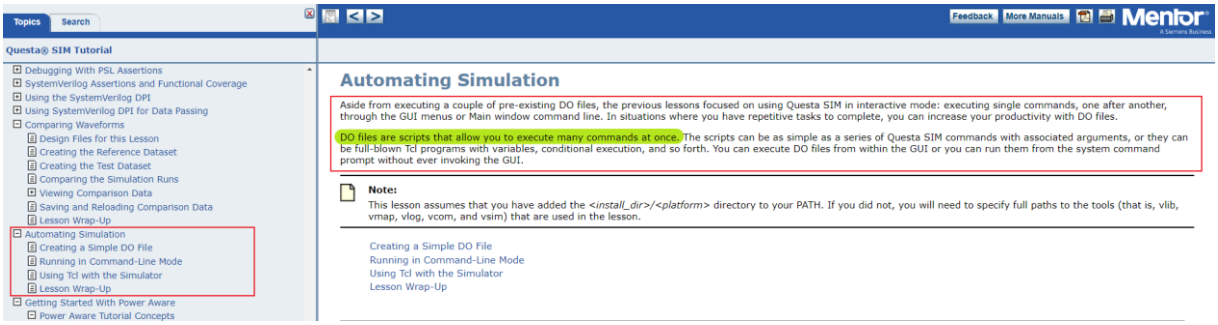
3、User’ s Manual ：Waveform Analysis



Tutorial ：Analyzing Waveforms

4、User’ s Manual ：Tcl and DO Files

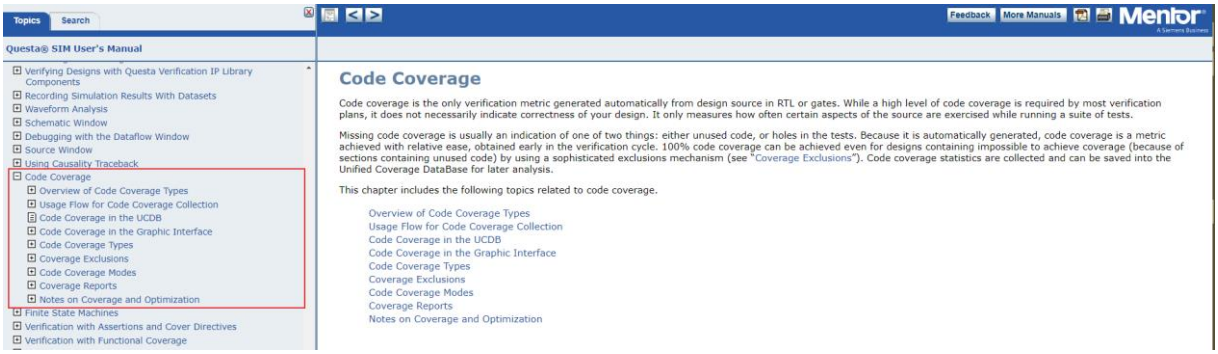
Tutorial ：Automating Simulation



进阶的章节

User’ s Manual ：Code Coverage

覆盖率，验证相关，实际项目中有用到

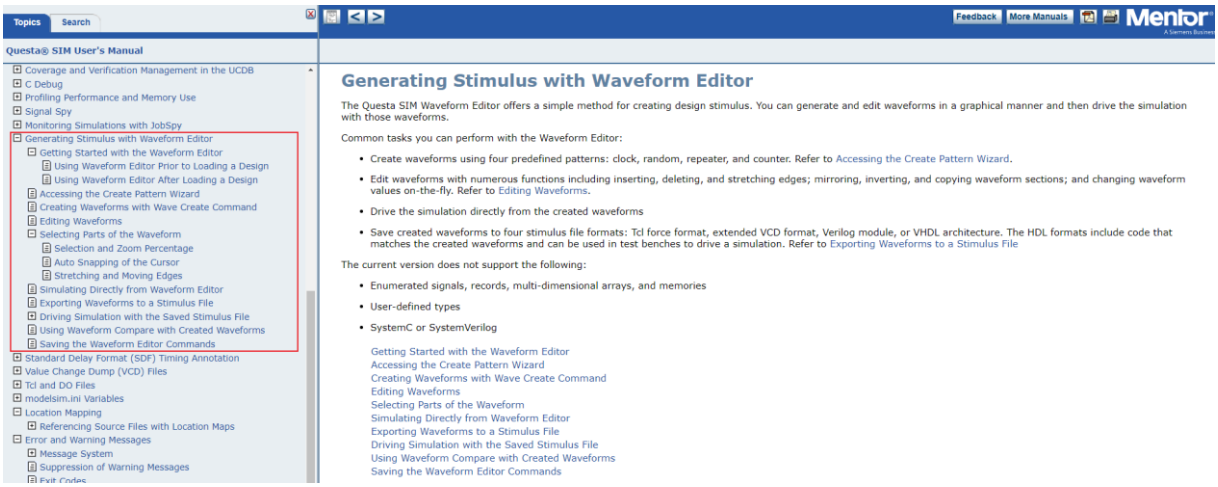


User’ s Manual ：Advanced Simulation Techniques - X Propagation in Simulation

X 态的传播，是一个曾被面试提问的主题

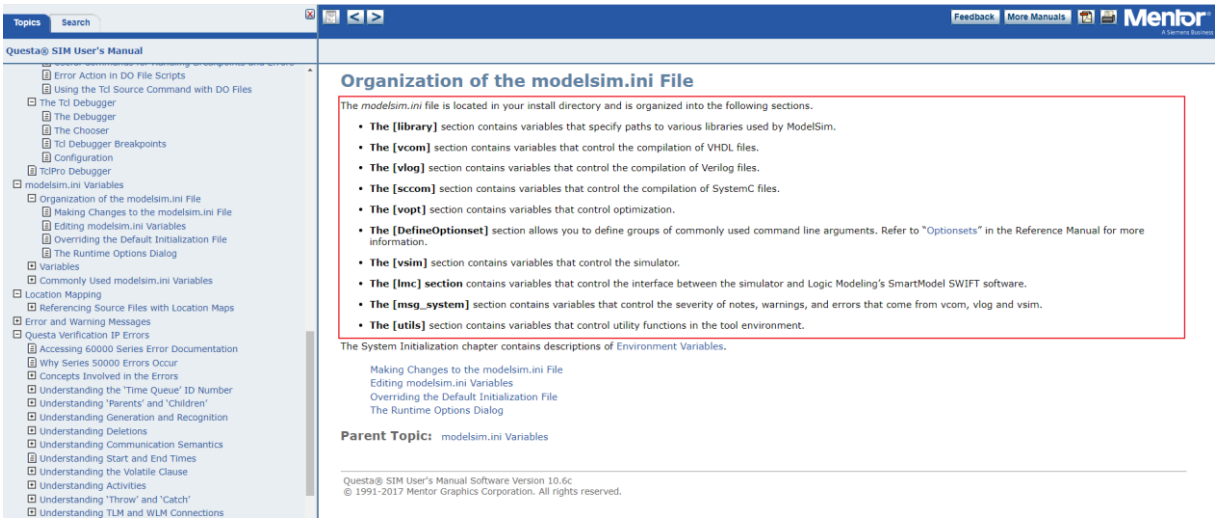
User’ s Manual ：Generating Stimulus with Waveform Editor

不写 tb，直接创建仿真激励，用于某些简单测试任务



User’ s Manual ：modelsim.ini Variables

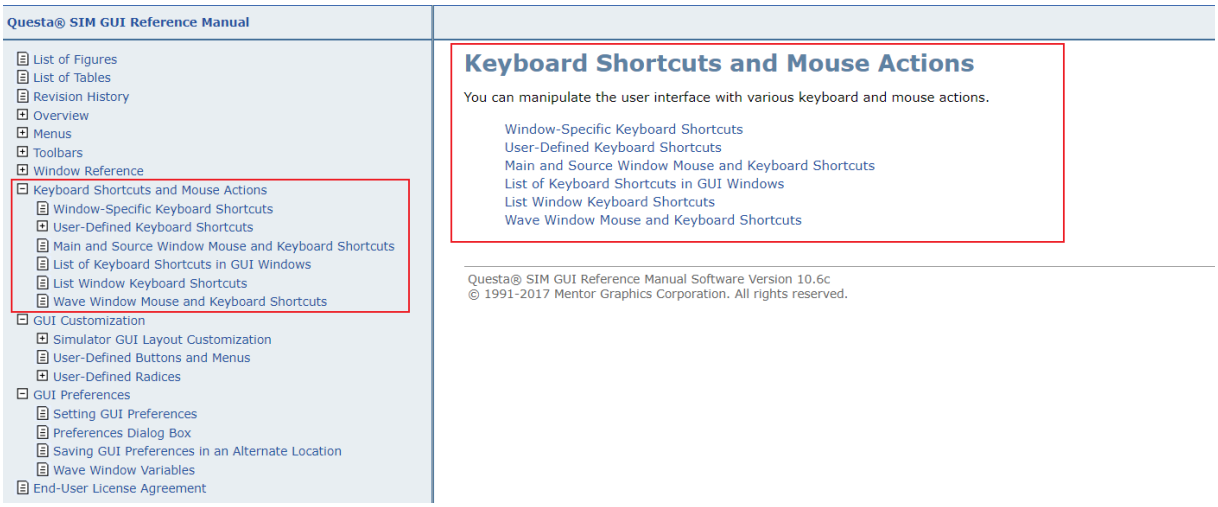
了解相关配置项



键鼠快捷操作

图形界面手册中给出了键鼠快捷操作，

GUI Reference Manual：Keyboard Shortcuts and Mouse Actions



其中关于波形界面的操作使用频率最高，需要注意掌握

Wave window mouse and keyboard shortcuts

The following mouse actions and keystrokes can be used in the Wave window.

Mouse action	Result
< control - left-button - click on a scroll arrow >	scrolls window to very top or bottom(vertical scroll) or far left or right (horizontal scroll)
< middle mouse-button - click in scroll bar trough> (UNIX) only	scrolls window to position of click

Keystroke	Action
i I or +	zoom in
o O or -	zoom out
f or F	zoom full; mouse pointer must be over the the cursor or waveform panes
l or L	zoom last
r or R	zoom range
<arrow up>	scroll waveform display up by selecting the item above the currently selected item
<arrow down>	scroll waveform display down by selecting the item below the currently selected item
<arrow left>	scroll waveform display left
<arrow right>	scroll waveform display right
<page up>	scroll waveform display up by a page
<page down>	scroll waveform display down by a page
<tab>	search forward (right) to the next transition on the selected signal - finds the next edge
<shift-tab>	search backward (left) to the previous transition on the selected signal - finds the previous edge
<control-f> Windows <control-s> UNIX	open the find dialog box; searches within the specified field in the pathname pane for text strings

常用命令

Quick Guide：Key Command Arguments

Key Command Arguments

Use <command> -help for a full list.

QVERILOG

The qverilog command compiles, optimizes, and simulates Verilog and SystemVerilog designs in a single step.
1. automatic work library creation
2. support for all standard vlog arguments
3. support for C/C++ files via the SystemVerilog DPI
4. implicit "run -all; quit" unless using -i, -gui, -do (see -R below)
5. vopt performance invoked (see the vopt section of this guide)

Key arguments to qverilog

<filename>	Verilog source code file to compile, one is required
[-R <sim_options>]	vsim command options applied to simulation

SCCOM

-link	Links source code, required
[CPP option]	C++ compiler option
[-g]	Compile with debugging info
-vv	Echo subprocess invocations on stdout
[-scv]	Includes SystemC verification library
<filename(s)>	SystemC files to be compiled

VCOM

[-2008 | -2002 | -93 |-87]	Choose VHDL 2008, 2002,1993, or 1987
[-check_synthesis]	Turn on synthesis checker
[-debugVA]	Print VITAL opt status
[-explicit]	Resolve ambiguous overloads
[-help]	Display vcom syntax help
[-f <filename>]	Pass in arguments from file
[-norangecheck]	Disable run time range checks
[-nodebug]	Hide internal variables & structure
[-novitalcheck]	Disable VITAL95 checking
[-nowarn <#>]	Disable individual warning msg
[-quiet]	Disable loading messages
[-refresh]	Regenerate library image
[-version]	Returns vcom version
[-work <libname>]	Specify work library
<filename(s)>	VHDL file(s) to be compiled

VLOG

[-vlog95compat]	Disable Verilog 2001 keywords
[-compat]	Disable event order optimizations
[-f <filename>]	Pass in arguments from file
[-hazards]	Enable run-time hazard checking
[-help]	Display vlog syntax help
[-nodebug]	Hide internal variables & structure
[-quiet]	Disable loading messages
[-R <simargs>]	Invoke VSIM after compile
[-refresh]	Regenerate lib to current version
[-sv]	Enables SystemVerilog keywords
[-version]	Returns vlog version
[-v <library_file>]	Specify Verilog source library
[-work <libname>]	Specify work library
<filename(s)>	Verilog file(s) to be compiled

VOPT

Design optimization options
1. Optimized designs simulate faster, while non-optimized designs provide object visibility for debugging.
2. Use +acc with vopt or vsim -voptargs with +acc for selective design object visibility during debugging.
3. Read "Optimizing Designs with vopt" in the User's Manual for additional information.

Key arguments to vopt

-o <name>	Optimized design name
<design>	Top-level design unit
[+acc=[<spec>]+[<module>]]	Enable design object visibility
+cover=bcefsx	Specifies coverage type(s)
-nocover	Disable coverage on all source files
-g	Assigns a value to generics and parameters with no value
-G	Forces value assignment for generics and parameters

Key arguments to vsim

[-vopt]	Run vopt if not automatically invoked
[-voptargs="<args>"]	Arguments passed to vopt, use +acc args for design visibility

VSIM

[-assertdebug]	Keep data for debugging assertion failures
[-assertfile <filename>]	Alternative file for recording assert messages
[-assume]	Simulate PSL and Verilog assume directives same as assert directives
[-c]	Run in cmd line mode
[-coverage]	Invoke Code Coverage
[-do "cmd" | <file>]	Run cmd or file at startup
[-elab]	Create elaboration file
[-f <filename>]	Pass in args from file
[-g|G<name=value>]	Set VHDL Generic values
[-hazards]	Enable hazard checking
[-help]	Display vsim syntax help
[-l <logfile>]	Save transcript to log file
[-load_elab]	Simulate an elaboration file
[-noassume]	Do not simulate PSL and Verilog assume directives
[-nopsl]	Disable PSL assertions
[-nosva]	Disable System Verilog concurrent assertions
[+notimingchecks]	Disable timing checks
[-quiet]	Disable loading messages
[-restore <filename>]	Restore a simulation
[-sdf[min|typ|max] <region>=<sdffile>]	Apply SDF timing data e.g., sdfmin /top=MySDF.txt
[-sdfnowarn]	Disable SDF warnings
[-sv_seed <seed>]	Specify a seed for the Random Number Generator of the root thread
[-t [<mult>]<unit>]	Time resolution
[-vcdstim [<instance>=]<filename>]	Stimulate the top-level design or instances from an Extended VCD file
[-version]	Returns vsim version
[-vopt]	Run vopt automatically
[-voptargs="<args>"]	Arguments to pass to vopt
[-view <filename>]	Log file for VSIM to view
[-wlf <filename>]	Log file to create
[<libname>.<design_unit>	Configuration, Module, Entity/Arch, or optimized design to simulate
[-wlfcachesize]	Specify WLF reader cache size (per WLF file.)
[-wlfslim <size>]	Specify the number of Megabytes to be saved in event log file
[-wlftlim <duration>]	Specify the duration of time to be saved in event log file

Code Coverage

Key Arguments to vcom/vlog

+cover=bcefsx	Specifies coverage type(s)

Key Arguments to vopt

+cover=bcefsx	Specifies coverage type(s)
-nocover	Disable coverage on all source files

Key Arguments to vsim

-coverage	Enables statistics collection

SVA & PSL

Key arguments to vcom and vlog

[-pslfile <file>]	External PSL file
[-nopsl]	Ignore embedded PSL assertions

Key arguments to vsim

[-nopsl]	Ignore embedded PSL assertions
[-nosva]	Ignore SystemVerilog concurrent assertions

Key modelsim.ini variables

AssertionFail*	Control assertion failure behavior
AssertionFormat*	Define messages for VHDL assertion types
AssertionPass*	Control assertion pass behavior
BreakOnAssertion	Stop the simulator after assertion message
Cover*	Control cover directive behavior
IgnoreSVA*	Control SVA message logging
Sv_Seed	Seed random number generator

Wave Window

add wave <item>	Wave specific signals/nets
add wave *	Wave signals/nets in scope
add wave -r /*	Wave all signals/nets in design
add wave abus(31:15)	Wave a slice of a bus
view wave	Display wave window
view wave -new	Display additional wave window
write wave	Print wave window to file
<left mouse button>	Select signal / Place cursor
<middle mouse button>	Zoom options
<right mouse button>	Context Menu
<ctrl-f>	Find next item
<tab> (go right)	Search forward for next edge
<shift-tab> (go left)	Search backward for next edge
i or + | o or -	Zoom in | Zoom out
f | l	Zoom full | Zoom Last

Key modelsim.ini variables

WLF*	Waveform management variables
WLFCacheSize	Change default or disable WLF file cache

8005 SW Boeckman Road
Wilsonville, OR 97070

Mentor® A Siemens Business

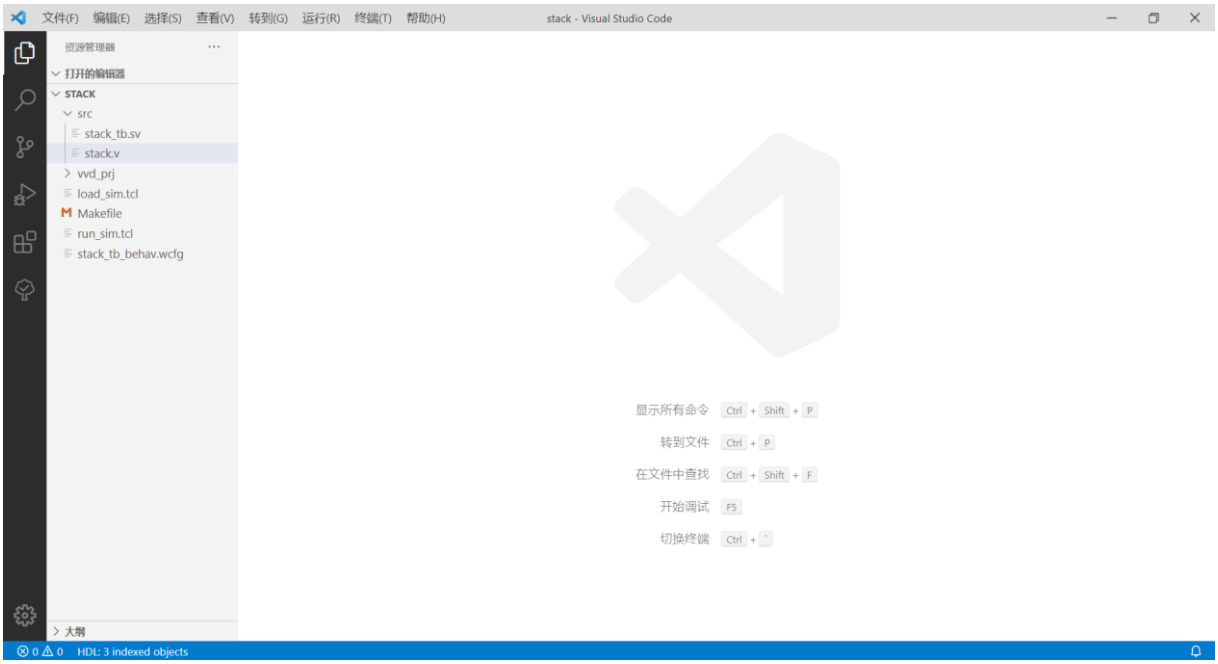
自行查阅 Command Reference Manual 并学习下面的常用指令

- .main clear
- add wave
- alias
- cd
- delete
- do
- force
- quit
- report
- restart
- run
- vlib
- vlog
- vopt
- vsim
- write format

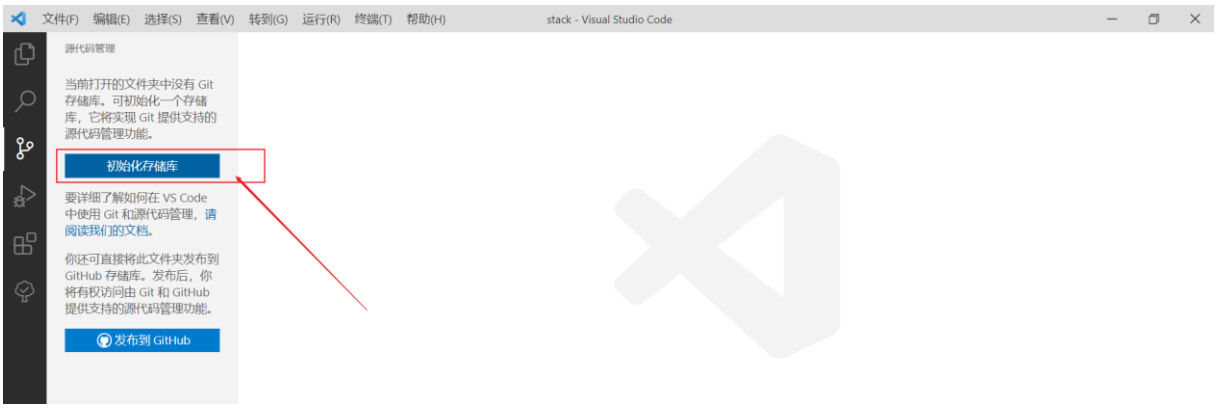
使用 Git 进行版本控制

如果你对 Git 的命令行不熟悉或者还不知道 Git 可以做哪些事情时，我建议从 VsCode 集成的 Git 支持（图形界面）开始用起（在用的过程中，你会逐渐明确需求）

仍然以之前的工作目录为例，



建立本地仓库



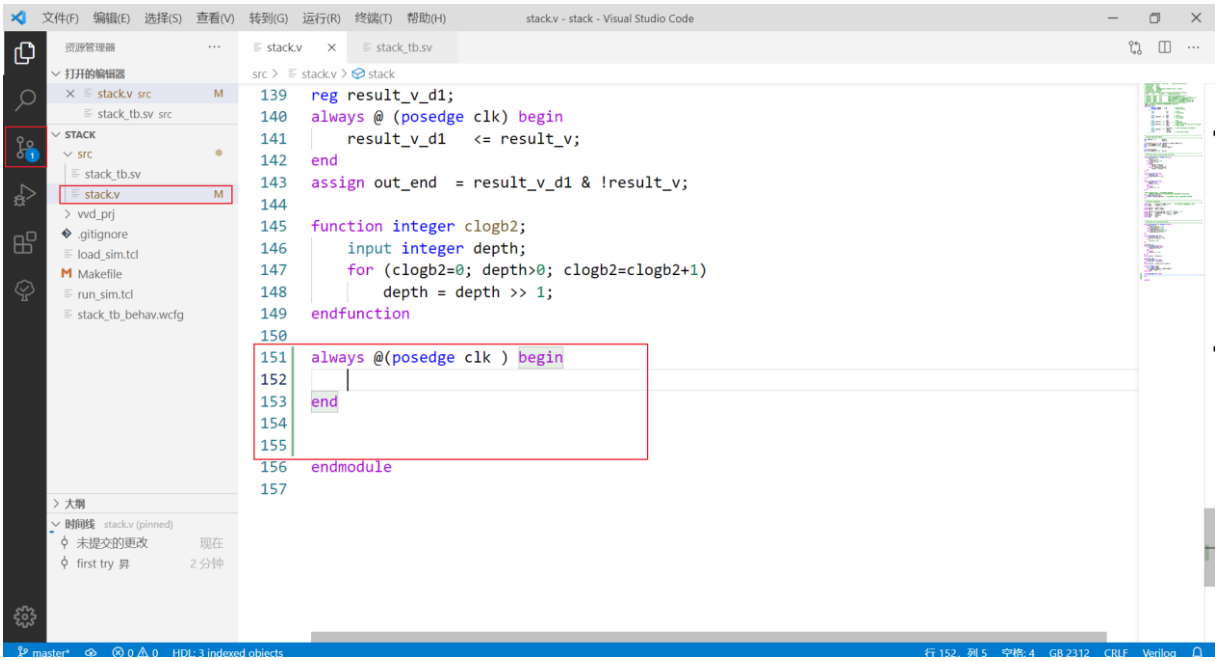
排除不关心的文件

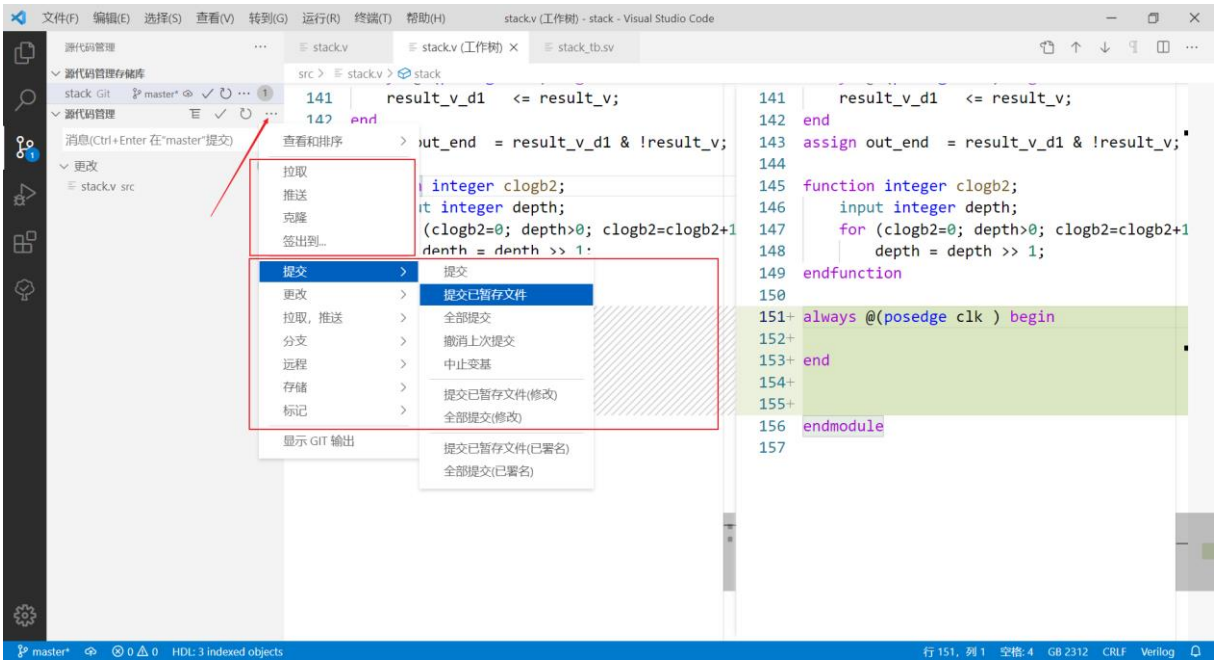
创建.gitignore 文件

```
#########
#Exclude all
#########
*
!.gitignore
#########
#Source files:
#########
#Do NOT ignore VHDL, Verilog, block diagrams or EDIF files.
!src/*
!*.wcfg
```

代码修改追踪

修改代码后，所有未提交的更改都会指示





关于 Git 的各概念及使用请自行学习

几处不重要的设置

Vivado

编辑器中文注释乱码

Windows 下，Vivado 内置编辑器可识别 GB-2312 字符集的汉字

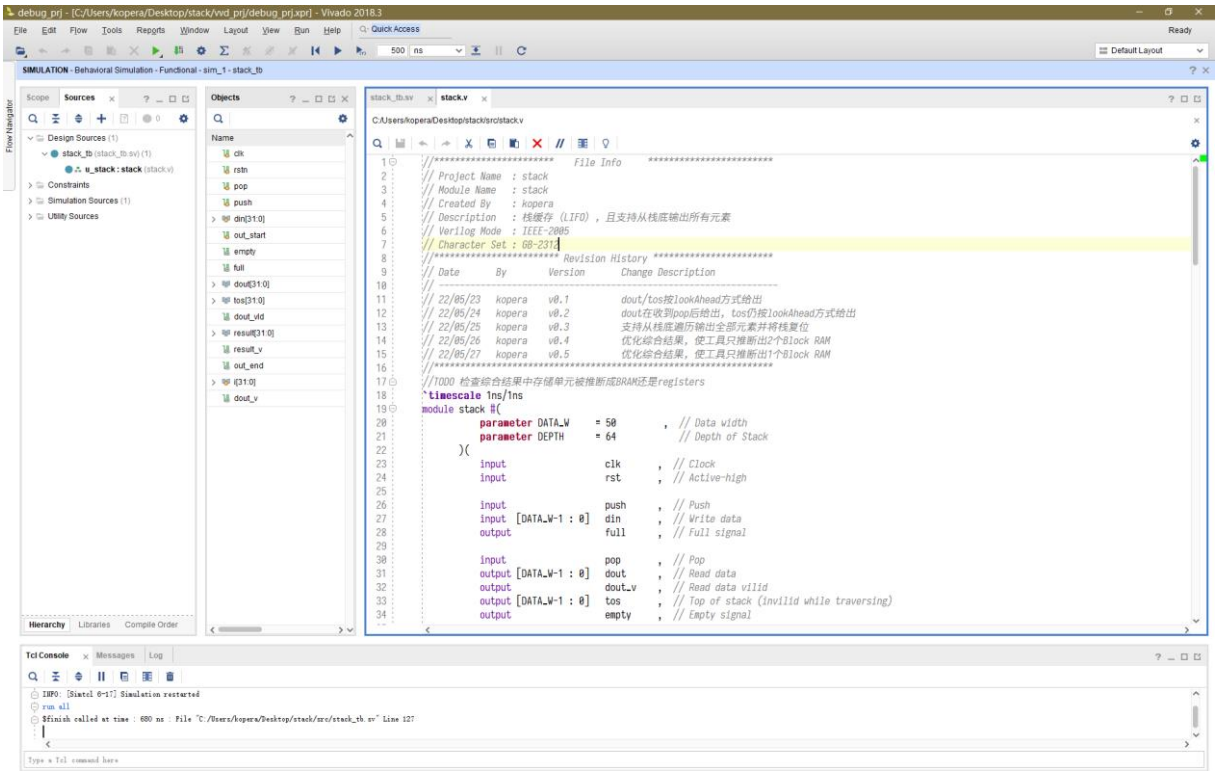
Linux 下，Vivado 内置编辑器可识别 UTF-8 字符集的汉字

编辑器更换等宽字体

大部分等宽字体都不支持中文（系统中自带的 Monospace 支持，但不够美观），等距更纱黑体 SC 支持中文字符集，可以将其设置为 vivado 自带编辑器的显示字体

拷贝字体到 Vivado 的 jre 环境中（Matlab 中设置字体同理）

Vivado 中显示效果如下



内置仿真器波形配置方案

Vivado 的常规设置项保存在下面的文件里

```
$(USER_PATH)\AppData\Roaming\Xilinx\Vivado\2018.3\vivado.xml
```

波形设置项保存在下面的文件里

```
$(USER_PATH)\AppData\Roaming\Xilinx\Vivado\2018.3\waveform\wv.ini
```

通过修改 `wv.ini` 替换配色方案

默认的方案（暗色调）

```
ARE_GRID_LINES_ON=true
IS_WAVEFORM_SHADOW_ON=true
IS_NUM_DISPLAYED_ELEMENTS_LIMITED=true
MAX_NUM_DISPLAYED_ELEMENTS=64
DEFAULT_RADIX=HEX
ARE_TRIGGER_MARKS_ON=true
ARE_NAME_TOOLTIPS_ON=true
RULER_UNITS=AUTO
WAVEFORM_AREA_BACKGROUND_COLOR=246 241 232
WAVEFORM_AREA_FOREGROUND_COLOR=0 0 0
NAME/VALUE_AREA_BACKGROUND_COLOR=220 220 220
WAVEFORM_AREA_TEXT_COLOR=0 0 0
NAME/VALUE_AREA_TEXT_COLOR=0 0 0
SELECTED_WAVEFORM_AREA_ITEM_COLOR=51 153 102
CURSOR_COLOR=255 102 0
MARKER_COLOR=0 204 255
TRIGGER_POINT_COLOR=255 0 0
DEFAULT_WAVEFORM_COLOR=0 0 0
WAVEFORM_TEXT_COLOR=0 0 0
WAVEFORM_VALUE_UNINITIALIZED_COLOR=255 165 0
WAVEFORM_VALUE_UNKNOWN_COLOR=255 0 0
WAVEFORM_VALUE_HI-Z_COLOR=0 0 255
WAVEFORM_VALUE_WEAK_COLOR=0 255 0
WAVEFORM_VALUE_DON'T_CARE_COLOR=190 190 190
TIME_SCALE_COLOR=0 0 0
GRID_LINES_COLOR=204 204 204
FLOATING_RULER_COLOR=225 207 0
DIVIDER_BACKGROUND_COLOR=192 192 192
CAPTURE_WINDOW_EVEN_COLOR=0 0 0
CAPTURE_WINDOW_ODD_COLOR=255 255 255
ASSOCIATION/DEPENDENCY_LINE_COLOR=0 255 255
NAME_COLUMN_WIDTH=175
VALUE_COLUMN_WIDTH=79
```

彻底关闭 WebTalk

Win 下修改如下文件

```
$(PATH)\Vivado\2018.3\bin\wbtcv.bat
```

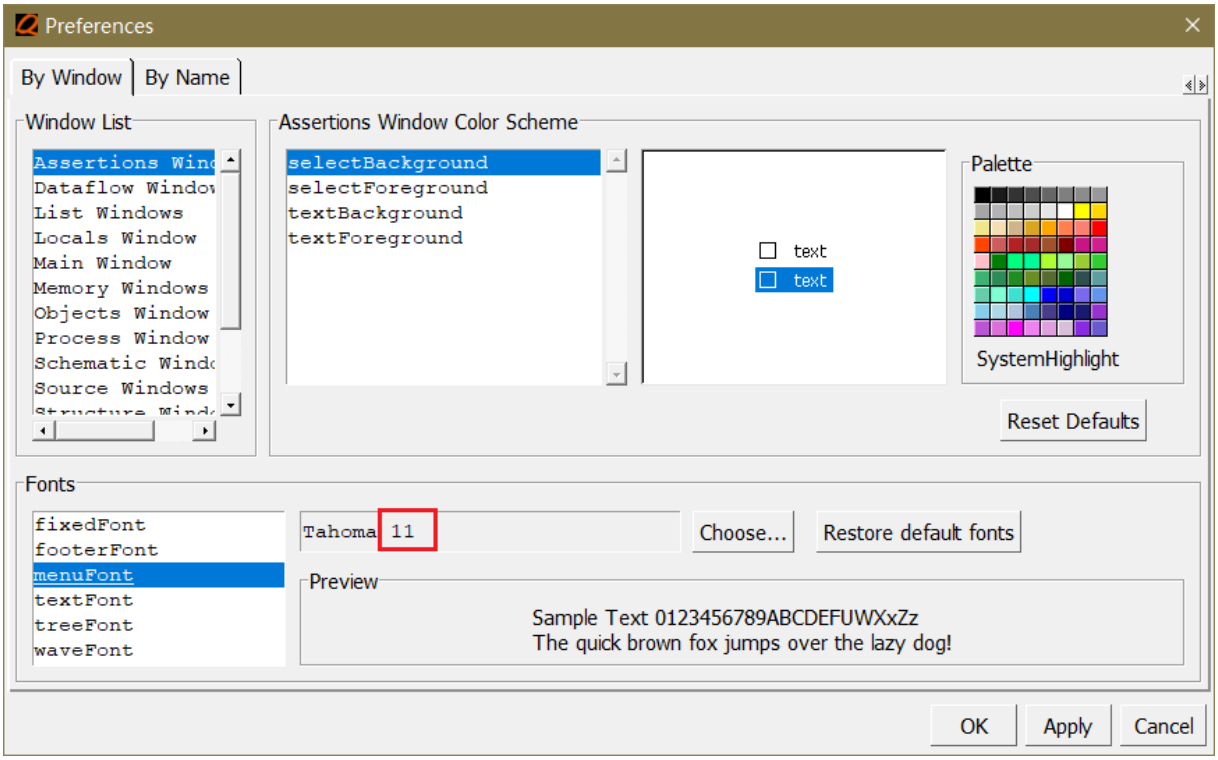
wbtcv.bat

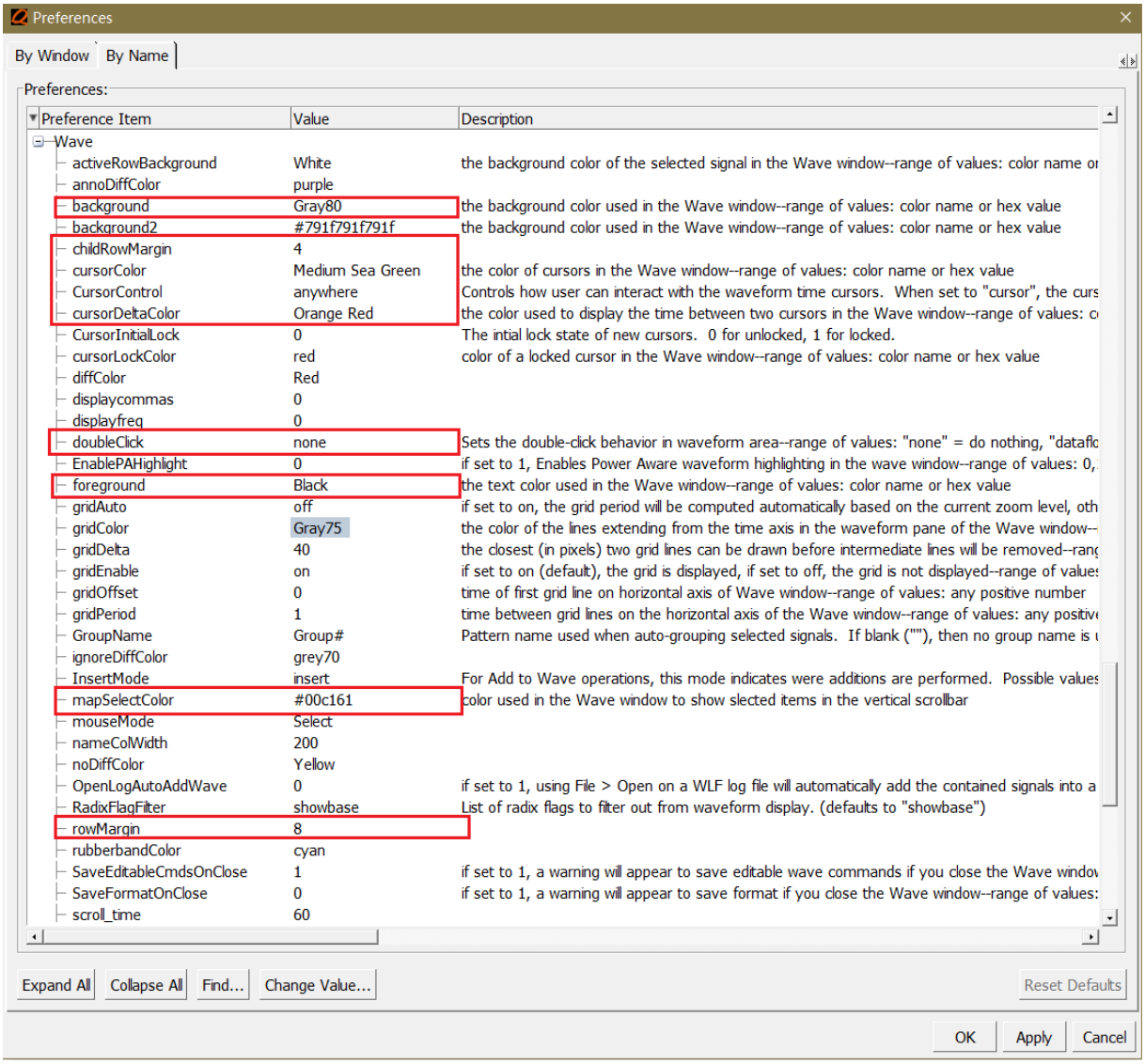
```
@echo off
setlocal
rem #
rem # COPYRIGHT NOTICE
rem # Copyright 1986-2014 Xilinx, Inc. All Rights Reserved.
rem #

rem ##
rem # Setup default environmental variables
rem ##
rem # RDI_BINROOT - Directory *this* script exists in
rem #  E.x.
rem #    /usr/Test/Install/bin/example
rem #    RDI_BINROOT=/usr/Test/Install/bin
rem #
rem # RDI_APPROOT - One directory above RDI_BINROOT
rem #  E.x.
rem #    /usr/Test/Install/bin/example
rem #    RDI_APPROOT=/usr/Test/Install
rem #
rem # RDI_BASEROOT - One directory above RDI_APPROOT
rem #  E.x.
rem #    /usr/Test/Install/bin/example
rem #    RDI_BINROOT=/usr/Test
rem ##
set RDI_PROG=%~n0
call "%~dp0setupEnv.bat"

rem # Set XILINX_VIVADO
set RDI_SETUP_ENV_FUNCTION=DIRNAME
call "%RDI_BINROOT%/setupEnv.bat" "%RDI_BINROOT%" XILINX_VIVADO

rem ##
rem # Launch the loader and specify the executable to launch
rem ##
rem #
rem # Loader arguments:
rem #   -exec   -- Name of executable to launch
rem ##
rem call "%RDI_BINROOT%/loader.bat" -exec %RDI_PROG% %*

```

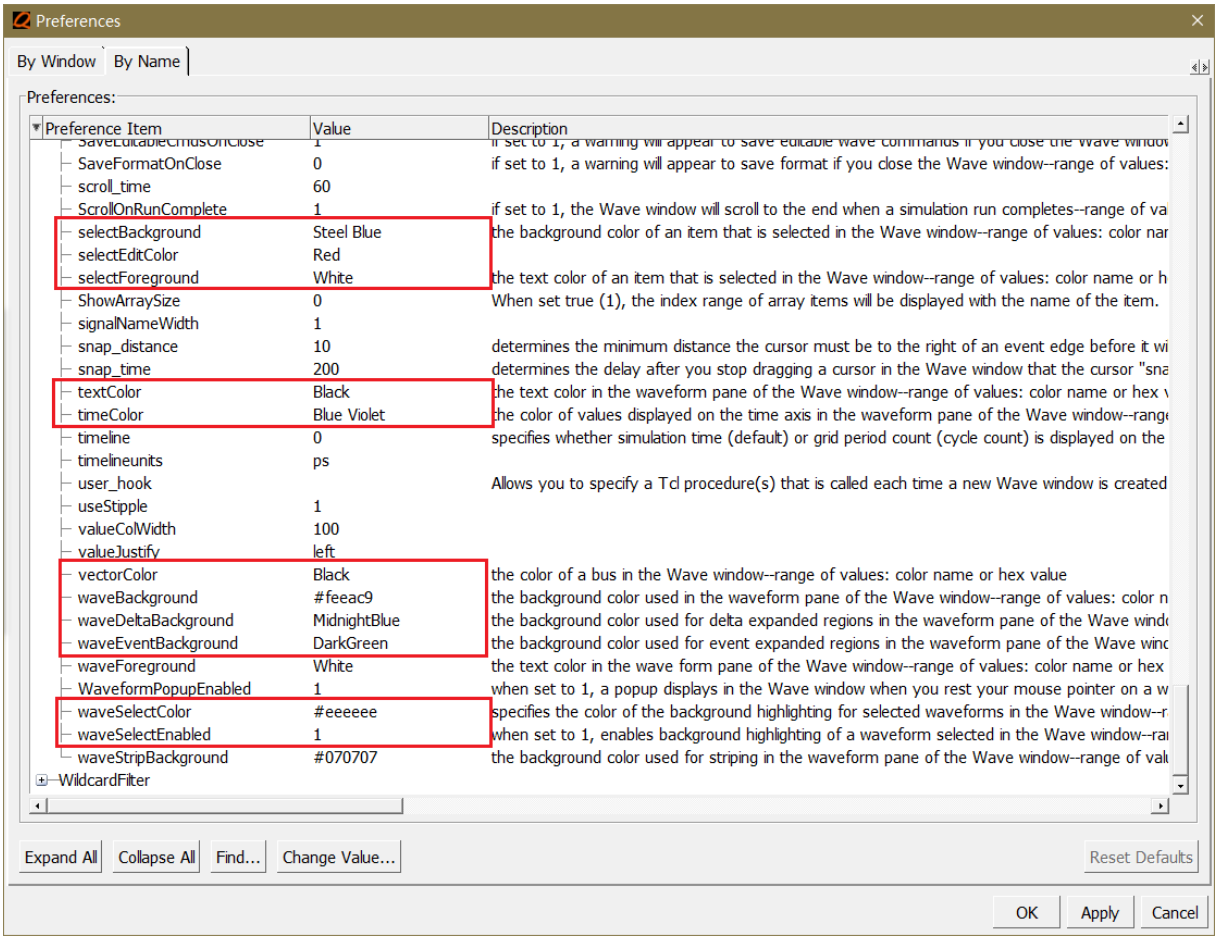
QuestaSim

调整配色方案

默认的配色方案是暗色调，这里给出一种亮色调配色

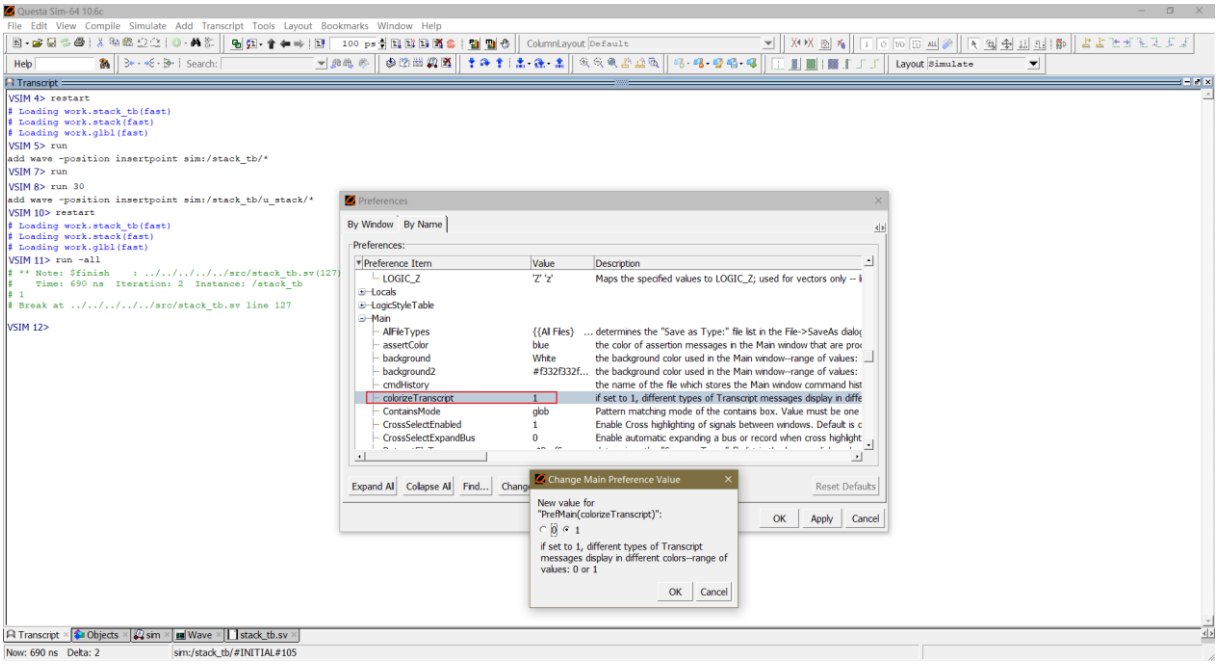

Preferences
By Window
By Name
Window List
Assertions Wind
Dataflow Windo
List Windows
Locals Window
Main Window
Memory Windows
Objects Window
Process Window
Schematic Windo
Source Windows
Assertions Window Color Scheme
selectBackground
selectForeground
textBackground
textForeground
text
text
Palette
SystemHighlight
Reset Defaults
Fonts
fixedFont
footerFont
menuFont
textFont
treeFont
waveFont
Tahoma 11
Choose...
Restore default fonts
Preview
Sample Text 0123456789ABCDEFUWXxZz
The quick brown fox jumps over the lazy dog!
OK
Apply
Cancel


Preferences
By Window
By Name
Preferences:
Preference Item
Value
Description
Wave
activeRowBackground
White
the background color of the selected signal in the Wave window--range of values: color name or
annoDiffColor
purple
background
Gray80
the background color used in the Wave window--range of values: color name or hex value
background2
#791f791f791f
the background color used in the Wave window--range of values: color name or hex value
childRowMargin
4
cursorColor
Medium Sea Green
the color of cursors in the Wave window--range of values: color name or hex value
CursorControl
anywhere
Controls how user can interact with the waveform time cursors. When set to "cursor", the curs
cursorDeltaColor
Orange Red
the color used to display the time between two cursors in the Wave window--range of values: c
CursorInitialLock
0
The intial lock state of new cursors. 0 for unlocked, 1 for locked.
cursorLockColor
red
color of a locked cursor in the Wave window--range of values: color name or hex value
diffColor
Red
displaycommas
0
displayfreq
0
doubleClick
none
Sets the double-click behavior in waveform area--range of values: "none" = do nothing, "dataflo
EnablePAHighlight
0
if set to 1, Enables Power Aware waveform highlighting in the wave window--range of values: 0,
foreground
Black
the text color used in the Wave window--range of values: color name or hex value
gridAuto
off
if set to on, the grid period will be computed automatically based on the current zoom level, oth
gridColor
Gray75
the color of the lines extending from the time axis in the waveform pane of the Wave window--
gridDelta
40
the closest (in pixels) two grid lines can be drawn before intermediate lines will be removed--rang
gridEnable
on
if set to on (default), the grid is displayed, if set to off, the grid is not displayed--range of values
gridOffset
0
time of first grid line on horizontal axis of Wave window--range of values: any positive number
gridPeriod
1
time between grid lines on the horizontal axis of the Wave window--range of values: any positive
GroupName
Group#
Pattern name used when auto-grouping selected signals. If blank (""), then no group name is
ignoreDiffColor
grey70
InsertMode
insert
For Add to Wave operations, this mode indicates were additions are performed. Possible values
mapSelectColor
#00c161
color used in the Wave window to show slected items in the vertical scrollbar
mouseMode
Select
nameColWidth
200
noDiffColor
Yellow
OpenLogAutoAddWave
0
if set to 1, using File > Open on a WLF log file will automatically add the contained signals into a
RadixFlagFilter
showbase
List of radix flags to filter out from waveform display. (defaults to "showbase")
rowMargin
8
rubberbandColor
cyan
SaveEditableCmdsOnClose
1
if set to 1, a warning will appear to save editable wave commands if you close the Wave windo
SaveFormatOnClose
0
if set to 1, a warning will appear to save format if you close the Wave window--range of values:
scroll_time
60
Expand All
Collapse All
Find...
Change Value...
Reset Defaults
OK
Apply
Cancel



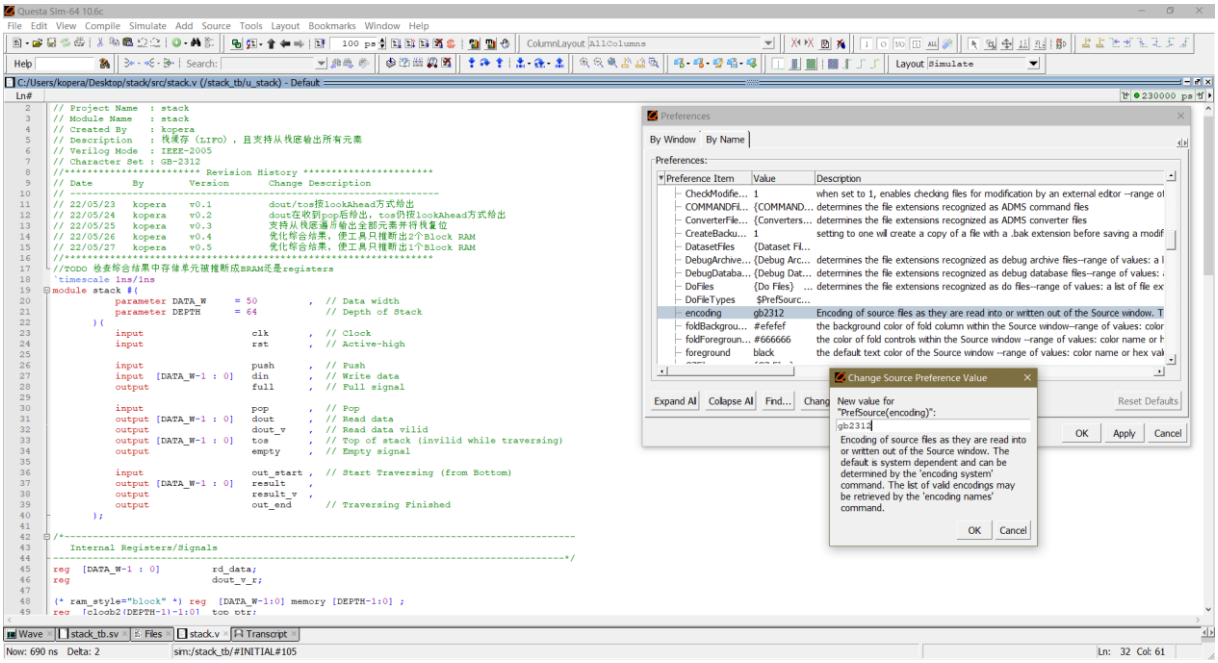
控制台输出高亮

设置控制台输出高亮，增强可读性



编辑器字符集设置

支持 UTF-8 和 GB2312，在这里设置



附录

工程源码和脚本模本可以在这里下载