TSSI
Solstice-TDS™
Quick Start Guide

Test Systems Strategies, Inc. (TSSI)

Version 2019.0
Rev 00

Copyrights and Trademarks

Restricted Rights Legend

Test Systems Strategies, Inc.
920 SW Sixth Avenue, 12th Floor
Portland, Oregon 97204 U.S.A.
www.tessi.com

Trademarks

Contents

1. Preparing to Run Solstice-TDS

1.1 Installing Solstice-TDS

Solstice-TDS only runs on a Linux 64-bit platform. Like most programs running on Linux, it is best to install the executable at a single location on the network where multiple users can access them. Likewise, a license server with the necessary Solstice licenses needs to be set up (Note: that Solstice license is based on Flexera FlexLM. License installation is covered in a different documentation.)

Download Solstice-TDS

If you haven't done so, download the software package from TSSI's website or as instructed by TSSI staff.

Install Solstice-TDS

Description	Example Linux Commands
1. Create a central location for Solstice	`mkdir /tools/tssi/solstice-2019.0`

2. Move the downloaded Core module to the central location.
 	
NOTE: all subsequent modules are to be untar'ed at the same central location
	`mv sol2019.0.r4379_lin_package.tgz /tools/tssi/solstice-2019.0`
	
	`cd /tools/tssi/solstice-2019.0`

3. Untar the package	`tar xvfz sol2019.0.r4379_lin_package.tgz`

4. Create symbolic link (this is highly recommended for easy upgrades to future versions). This is to be done only once when installing the Core module	`cd /tools/tssi`
	
	`ln –s solstice-2019.0 solstice`

5. Set the mandatory environment variable, TDSDIR	`export TDSDIR=/tools/tssi/solstice/tds`
	
	`NOTE: the extra 'tds' at the end of the path`

System Requirements

1. Processor: Intel 64-bit x86 with EM64T support; or AMD 64-bit x86 architecture.

2. RAM: 4G minimum.

3. Operating System: Redhat 4.8 or later; SuSE 11.1 or later. Linux Kernel supporting "long mode".

4. Applications: openMotif 2.2.x installation is required.

NOTE: Other Linux versions running on different architectures (AMD64, IA64, 64-bit SPARC) are not guaranteed to be 100% compatible.

Environment Variables

Each user needs to set their path variables to point to the location of Solstice.

If Solstice has been properly installed on your network and you have access to executable, all you need to do to get it to operate correctly is to set a few operating system (OS) environment variables. For this explanation, it's assumed you are using a .bashrc file (GNU Bourne shell). Other Linux shells have a slightly different style of setting environment variables, but set-up is essentially the same.

Example .bashrc snippet:

```
export TDSDIR=/tools/tssi/solstice/tds
export PATH=$PATH:$TDSDIR
```

To test if your setup is correct, you can type the following command:

```
which wavemakerplus
```

If this command cannot find "wavemakerplus," then you will need to further troubleshoot the problem or contact hotline@tessi.com.

Online User Manuals

Once the Solstice-TDS GUI, WaveMaker+, is launched:

- Select "Help->WaveMaker+ Manual…" for documentation on the WaveMaker+ operations and features.

- Select "Help->Solstice Manual…" for documentation on Solstice-TDS operations and features.

2. Introduction to WaveMaker+

2.1 Invocation of WaveMaker+

Almost always, you will start a session with Solstice by invoking its user interface tool called WaveMaker+. It is invoked with the following command (do not type the "%" prompt):

```
% wavemakerplus
```

2.2 Overview of the WaveMaker+ User Interface

The initial user interface will look similar to the one in Figure 1. As you can see, there are three primary areas of the WaveMaker+.

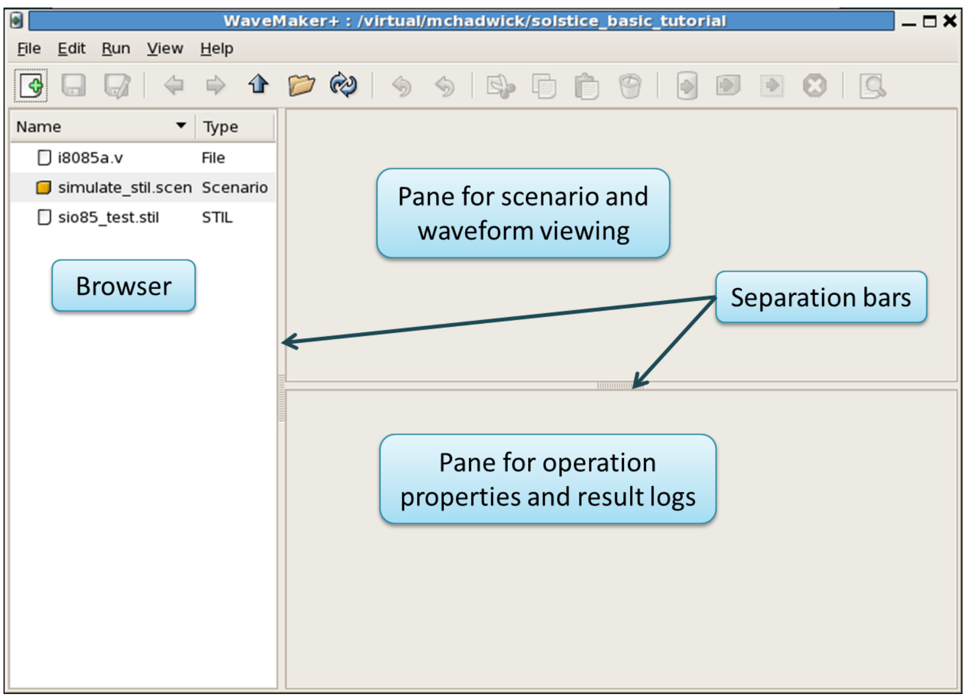


Figure 1: Initial screen upon invoking WaveMaker+

The ***browser*** area along the left side lists out the objects in the current directory. The objects are analyzed and "typed." These types are of significance to Solstice. The different type possibilities are described in more detail later in this document.

The ***pane*** on the top right will display either ***scenarios*** or ***waveforms***. A scenario is a graphical representation of process that will execute in WaveMaker+. It consists of a number of objects that are connected together to form the flow diagram of the process. The objects primarily consist of blocks representing *operations*, files, and directories. WaveMaker+ comes with a set of predefined operations.

The waveforms are graphical representation of the test patterns.

The ***pane*** on the bottom right is for listing out the properties of the ***operations*** (as described in the previous paragraph), and for displaying log files as scenarios and simulations are executed.

2.3 Working with Scenarios

A scenario is a graphical representation of a defined process to be executed in Solstice. To start a session with Solstice you need to either create a scenario, use a pre-existing scenario, or use a script that was created from a scenario. Therefore, it stands to reason that become proficient with scenarios is important for using Solstice.

Here are the eight basic skills you need to know when working with scenarios. In the subsequent sections we will go through each one of them.

- Opening
- Running
- Creating
- Saving
- Placing file
- Placing operations
- Connecting objects
- Editing operation properties

Opening a Scenario

WaveMaker+ identifies existing scenarios by type in the browser (see Figure 2). To open an existing scenario, simply double-click it with the left-hand mouse button. (Note that a scenario will show up as a directory within a Linux shell.)

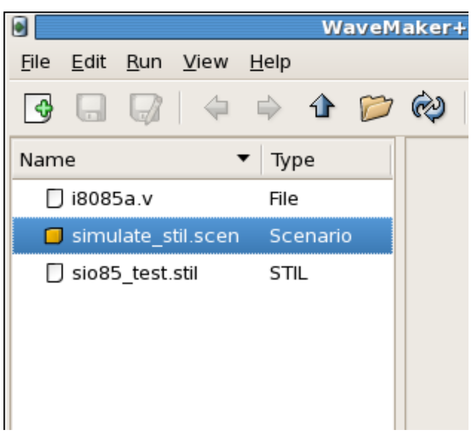


Figure 2: Scenarios are indicated by type within WaveMaker+

Running a Scenario

Once you open a scenario, you can run it. Some of the objects in a scenario are *operations*. Operations usually read from and write to files and waveform databases. Figure 3 shows four examples of operations—“stil in”, “ascii in”, wdb conditioner” and “stil out.” When we run the scenario, these operations will execute. You can use run commands by using either the run menu or the four associated icons. When you use “Run Scenario,” all the operations in the scenario will execute—in the case of Figure 3, there are four operations. If you use “Run Selected,” only the highlighted operations are executed—in the case of Figure 3, the “wdb conditioner” operation is selected. Multiple operations can be selected by using the <cntl> <left-hand mouse> buttons.

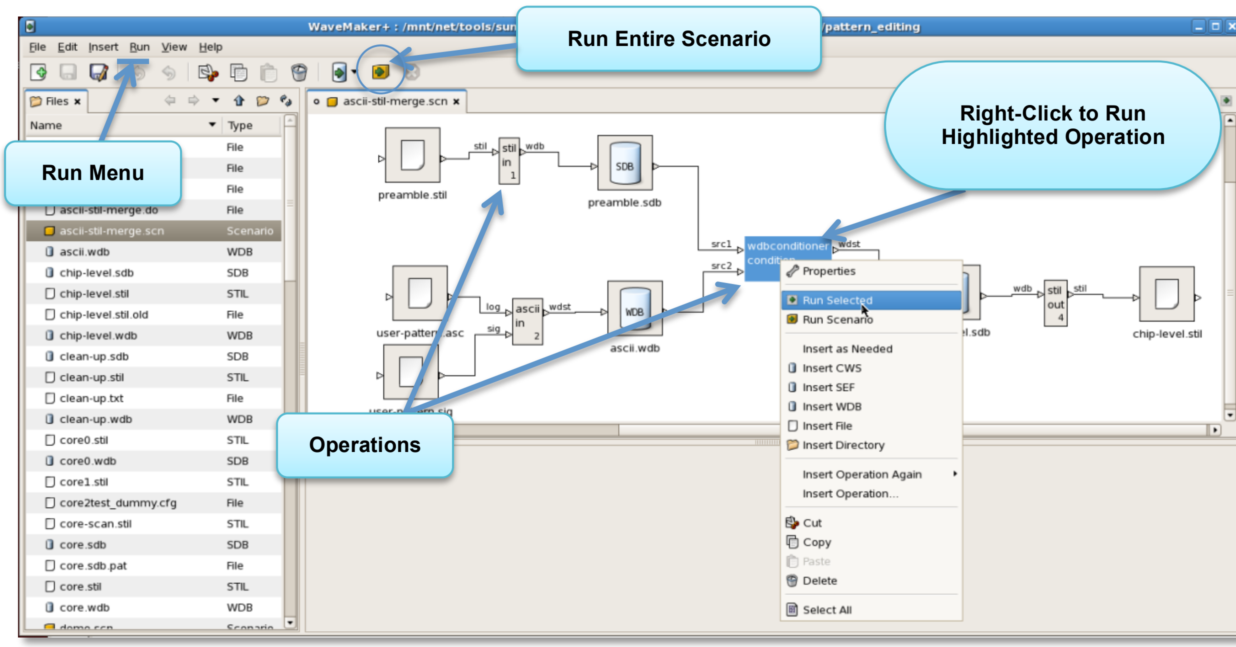


Figure 3: Running a scenario in WaveMaker+

Creating a Scenario

To create a new blank scenario select the ***File -> New*** menu item. A white canvas will appear in the scenario pane. There is also an icon for performing this.

Saving a Scenario

You can use the ***File -> Save*** menu item or the associated icon to save your scenario. Once it has been saved, it will be listed in the browser with type “Scenario.” The scenario will show up as a directory of files in a Linux shell.

Placing Files, Directories, and Similar into a Scenario

A scenario is a collection of interconnected objects. These objects are either operations or objects that will be operated upon. Most of these objects will be files. There are three ways to place objects into a scenario:

- Drag a file from the browser onto the scenario canvas. Note that there are icons which allow you to navigate to other directories.

- Use the ***Insert*** menu to access the files.
- Place the cursor over an operation that already exists in the scenario, bring up the localized menu with the right-hand mouse, and select ***insert as needed***. If there are any unconnected ports on the operation, objects will be automatically connected to those ports. The objects will be given default names.

A file can be renamed once it has been placed on the canvas by selecting the name and editing.

Directories, waveform databases, and similar objects are manipulated in the same fashion.

NOTE: If an object is represented in a scenario, that does not necessarily mean that the objects exists. This is particularly true for objects connected to the outputs of operations, where they may require you to run the scenario first to generate them.

Placing Operations into a Scenario

To place an operation into a scenario select the ***Insert -> Operation...*** menu item. Upon selection, a dialog window similar to the one in Figure 4 will appear. The "Operation" pane consists of a series of submenus that must be expanded to show the operations. This expansion is accomplished by selecting the small triangles along the left column.

The "History" pane lists out frequently used operations for easier access.

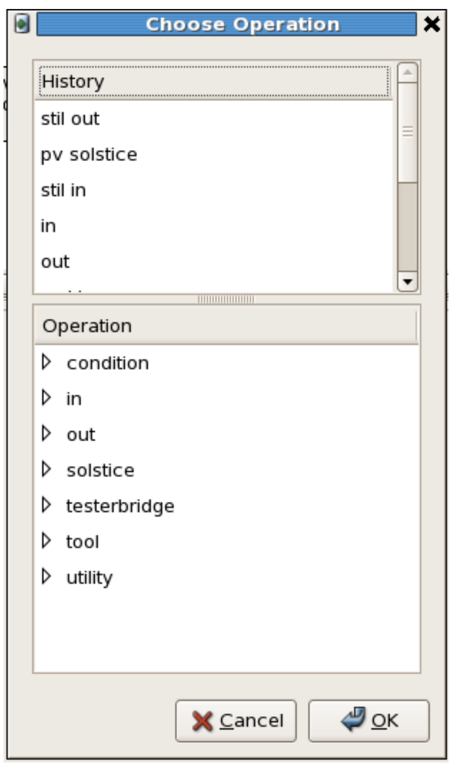


Figure 4: Insert operation dialog. The small triangles expand/collapse the submenus

Figure 5 show that once an operation has been selected, an outline of the object will follow the movement of the cursor. Move the cursor to the desired location on the scenario canvas and place the operation with a mouse click. Don't worry about the exact placement since all objects can be dragged to different positions at any time.

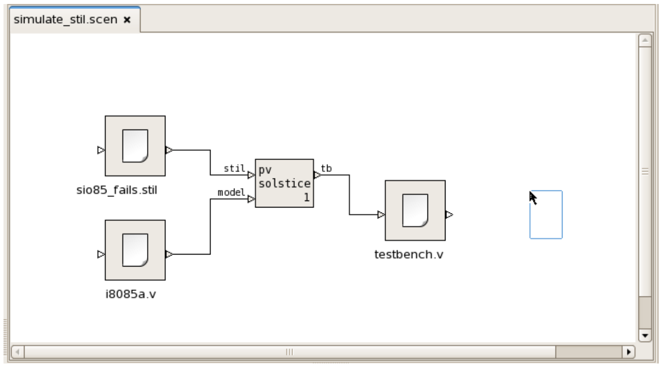


Figure 5: An outline of the operation follows the cursor until it is placed

Connecting Objects in a Scenario

Operations read from and write to files, directories, and databases. To establish this flow, the objects' ports must be connected. To connect ports click on a port of one of the objects, "rubber-band" the connection to the port of the other object, and click. The connection will show a line between them, as illustrated in Figure 6.

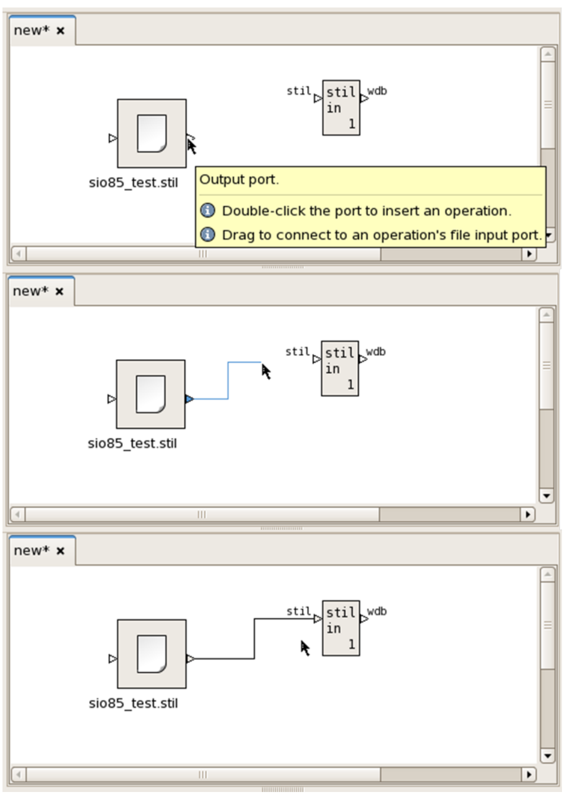


Figure 6: The three actions for connecting objects

Editing an Operation's properties

You can view and edit an operation's properties by double-clicking it's icon in the scenario. The properties dialog will be displayed in the lower pane as shown by Figure 7. Figure 7 indicates that there are three different types of fields. The one marked by a red square indicates an external connection of the operation to another object. In this example there are three of these connections. The blue square indicates a field in which the value is typed in. The green square indicates that there is a pull-down menu of choices. Most of the fields come with a default value. Notice that there is no "Save" or "OK" button, meaning that the values selected in the dialog are immediately applied to the operation.

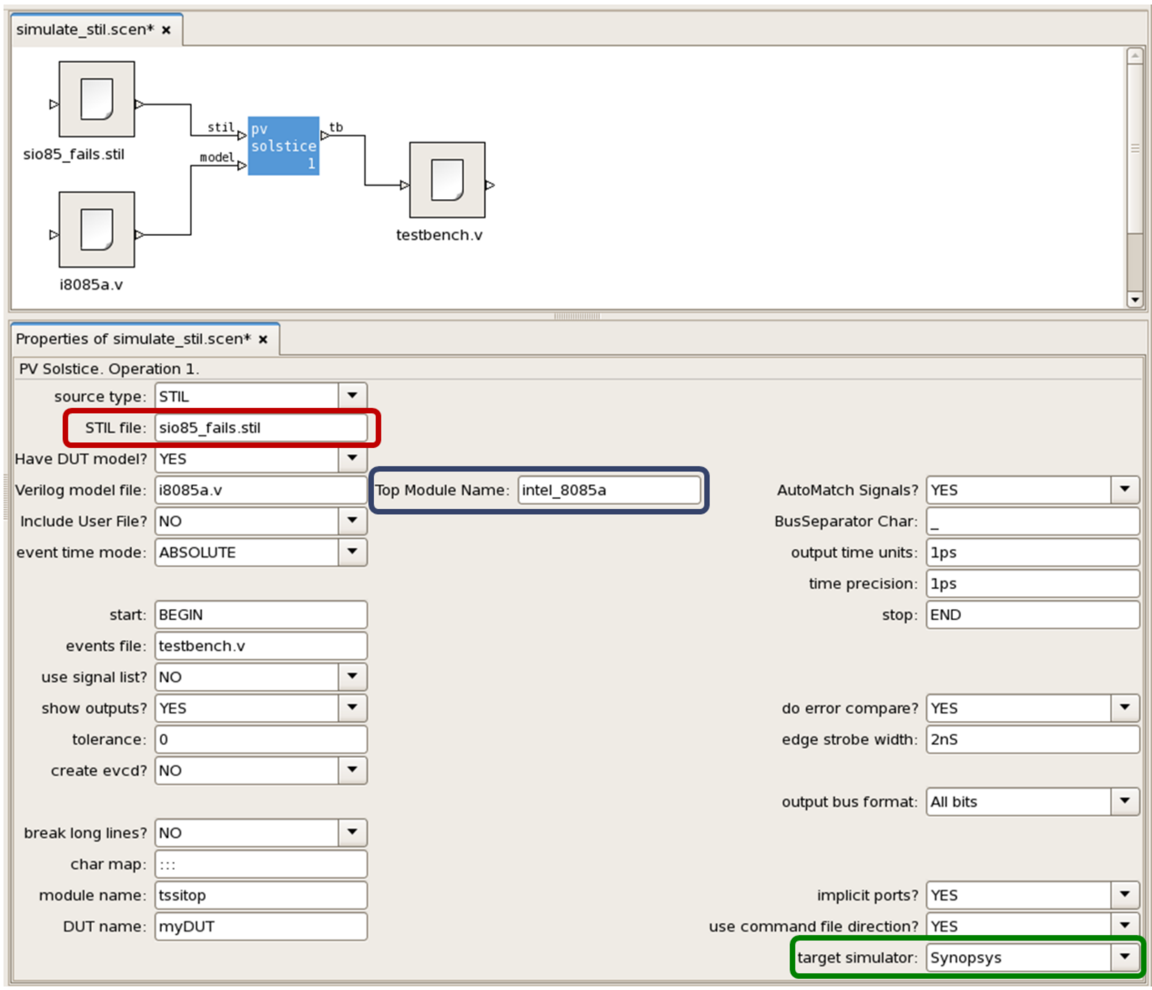


Figure 7: Properties dialog indicating the three different types of fields

2.4 *Waveform Editor*

The primary tool for viewing your test data in Solstice is the waveform editor. Figure 8 shows an example waveform editor session.

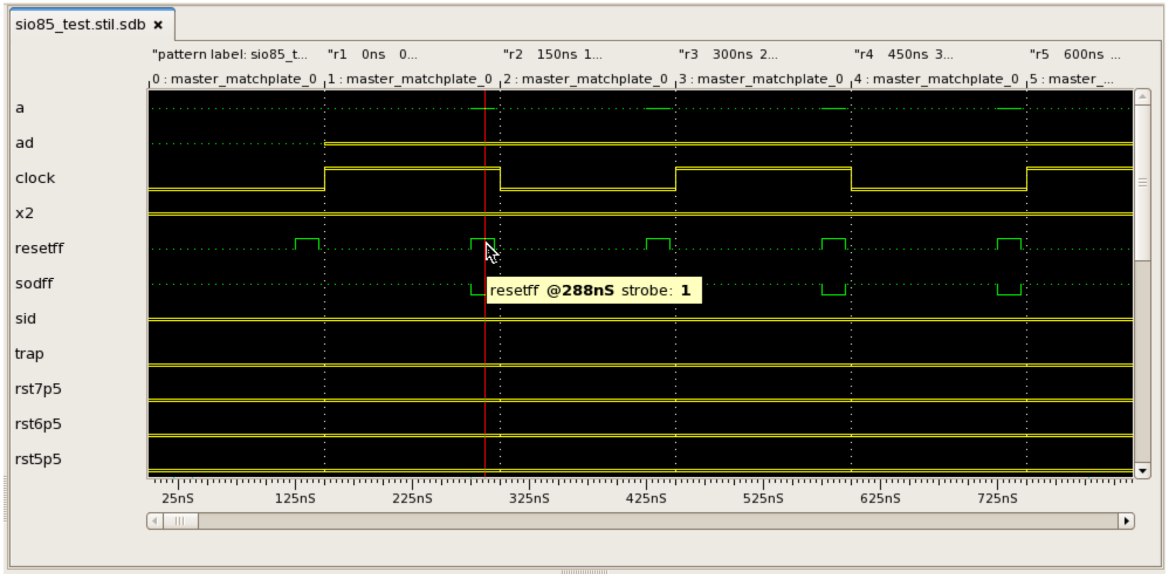


Figure 8: Example of a waveform editor session

STIL Database (SDB) and WGL Database (WDB)

Test pattern data used within Solstice is normally held in a “STIL database” (SDB). It is the next-generation of WGL Database (WDB). The difference is the additional STIL data structures stored in SDB. SDBs are backward compatible to WDBs. Throughout the tool suites, the terms WDB and SDB will be used interchangeably. The waveform editor supports both SDB and WDB.

Input test pattern files in formats such as STIL, WGL, and specific ATE formats will be converted to an SDB before viewing in the waveform editor.

Converting input file formats to SDB requires an operation such as the “STIL In” (if the input format is STIL).

The primary tool for viewing an SDB is the waveform editor. The reciprocal is also true—most of the test data viewed by waveform editor is in SDB format.

Features of the Waveform Editor

Figure 9 shows the primary features of the waveform editor. Each signal in the test data is listed along the Y-axis, and the accumulative time of the tests along the X-axis. Most test data formats (STIL, WGL, ATE specific, etc.) are split into three main sections: signals, timing, and bit patterns. One way to think of the waveform data is that it is a combination of these three sections. The result is a set of waveforms that represent the entire test.

The signals are color coded—green are outputs (in respect to the DUT), yellow are inputs, light gray are undetermined (not shown in Figure 9). Bidirectional signals will be both green and yellow, representing when they are acting as input or output, respectively.

The waveforms are delineated by test cycles. Above each cycle is the name of the timeplate that is being applied and the pattern number. If comments existed in the text file, they are also listed. Output signals will indicate when strobe output is valid.

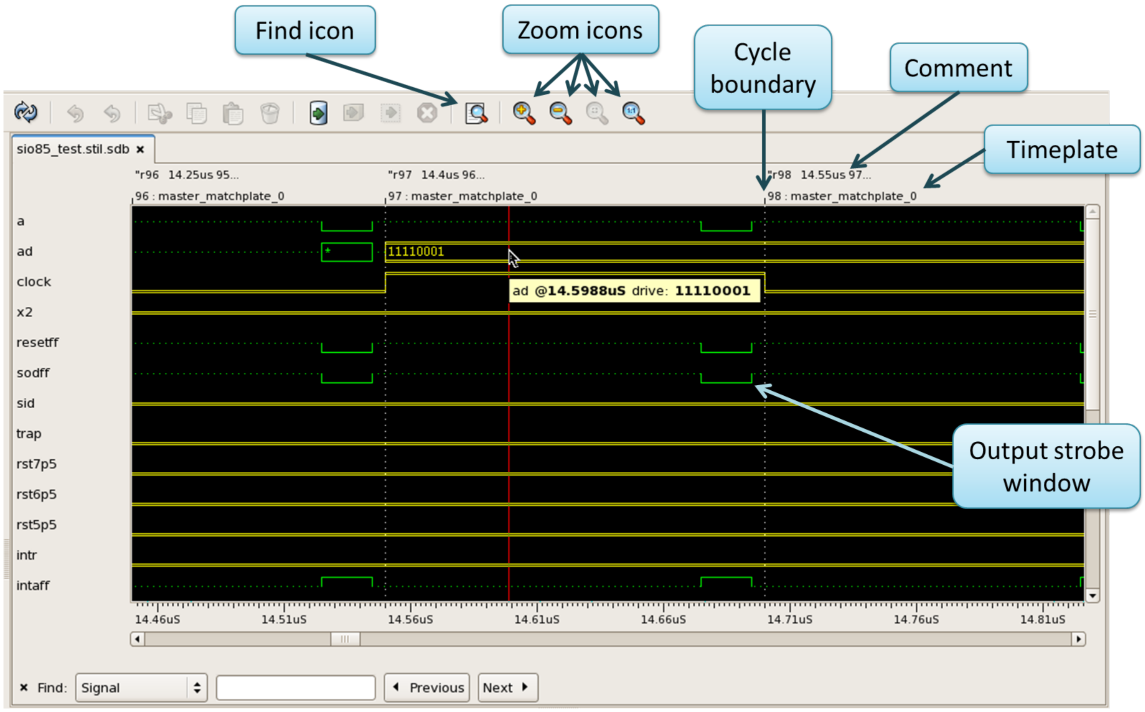


Figure 9: Key Features of the waveform editor

Exact information can be read by hovering the cursor over the area of interest.

There are several icons associated with the waveform editor. A bubble help will appear if the cursor is briefly hovered over the icon.

One of the icons turns on the “find” dialog which will appear along the bottom of the waveform editor, as shown in Figure 9. This is a powerful utility that will allow you to easily navigate large test pattern files. It lets you search according to any one of the following criteria:

- Comment
- Cycle
- Cycle Boundary
- Difference (between two SDB’s)
- Edge
- Error
- Match Section
- Signal
- Time
- TimePlate

Find Comment—finds the next comment embedded within the patterns, regardless of what the comment is.

Find Cycle—finds a particular cycle by its number.

Find Cycle Boundary—goes to the next cycle boundary

Find Difference (between two SDB's)—finds the next highlighted difference between two SDB's that are being compared (see Figure 10). This requires a two-step process. First open the reference SDB with the waveform editor. Next choose the SDB to compare by selecting the ***View -> Compare Database...*** menu item. Navigate to and select the SDB to compare, or just drag the second SDB from the Browser into the waveform viewing pane. As seen in Figure 10, there will be two waveforms displayed for each signal. Differences between the SDB's will be indicated in red. The "Find Difference" feature will go to the next difference.

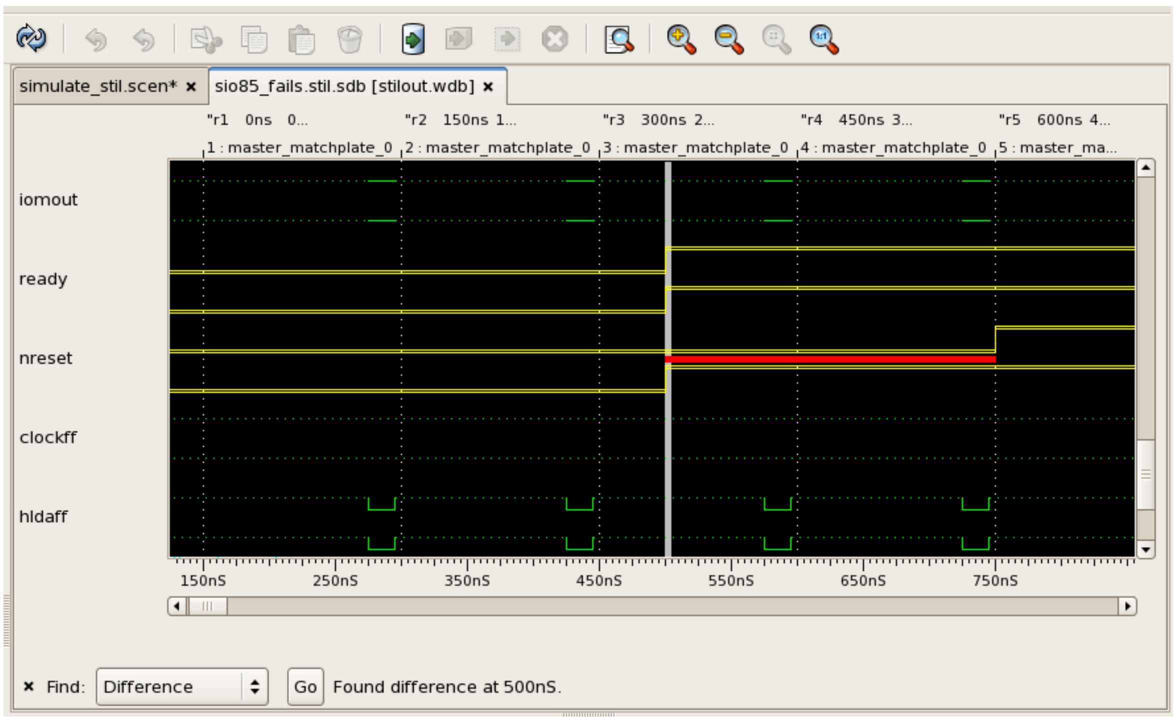


Figure 10: Difference between two compared databases highlighted after using the "Find Difference" feature

Find Edge—select a signal name along the left-hand side of the waveform viewer and select the "Find Edge" feature to go to the next edge.

Find Error—if a simulation (i.e. pattern verification) is executed and some of the patterns fail, mismatches will be projected, with red highlighting onto the waveforms. The "Find Error" feature will find the next failing bit.

Match Selection—highlight a section of a waveform and find the next occurrence of it.

Find Signal—will move along the x-axis to find a particular signal by name.

Find Time—will move to the time value entered. Note: the value must have an “s” suffix.” For example: 32.5us

Find Timeplate—will find the next occurrence of the timeplate selected from the pull-down.

3. Quick Start Test Case

3.1 Pattern Conversion

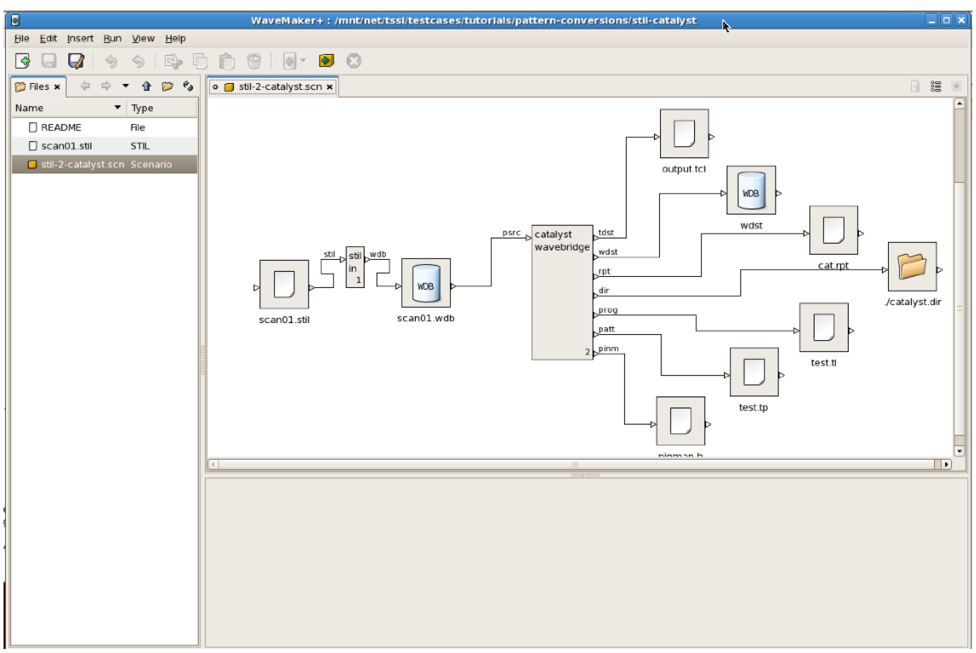
3.1.1 STIL to Catalyst

Converting a STIL pattern file to be converted to Teradyne Catalyst files.

Running the Quick Start Test Case inside WaveMaker+

Description	Example Linux Commands
1. Un-tar the included test case	`tar xvfz stil-catalyst.tgz`
2. Go into the right directory	`cd stil-catalyst`
3. Invoke Solstice GUI	`$TDSDIR/wavemakerplus`
4. Open the flow ("Scenario")	`Double click on "stil-2-catalyst.scn"`

5. Run the Scenario to start the conversion flow and the STIL file will be converted to Catalyst files and stored in the "catalyst.dir" folder.	`Select from the menu "Run->Scenario"`
	
	`When process finished, double click on the "catalyst.dir" folder to see the result files. Double click on the files to view content.`



Running the Quick Start Test Case from the Command Line

At a Linux system prompt, do: $TDSDIR/wavemakerplus -b stil-2-catalyst.scn

3.1.2 STIL to V93000

Converting a STIL pattern file to be converted to Advantest V93000 files.

Running the Quick Start Test Case inside WaveMaker+

Description	Example Linux Commands
1. Un-tar the included test case	`tar xvfz stil-v93000.tgz`
2. Go into the right directory	`cd stil-v93000`
3. Invoke Solstice GUI	`$TDSDIR/wavemakerplus`
4. Open the flow ("Scenario")	`Double click on "stil-2-v93000.scn"`

5. Run the Scenario to start the conversion flow	`Select from the menu "Run->Scenario"`
	
	`When process finished, double click on the "v93000.dir" folder to see the result files. Double click on the files to view content.`
	
	`Also for the V93000, a test.run script will be generated to invoke v93k compiler to generate binary .binl patterns`

Running the Quick Start Test Case from the Command Line

At a Linux system prompt, do: $TDSDIR/wavemakerplus -b stil-2-v93000.scn

3.1.3 Merging a STIL Pattern with a Space-Delimited ASCII Pattern

Running the Quick Start Test Case inside WaveMaker+

Description	Example Linux Commands
1. Un-tar the included test case	`tar xvfz stil-ascii.tgz`
2. Go into the right directory	`cd stil-ascii`
3. Invoke Solstice GUI	`$TDSDIR/wavemakerplus`
4. Open the flow ("Scenario")	`Double click on "stil-ascii.scn"`
5. Run the Scenario to start the conversion flow	`Select from the menu "Run->Scenario"`

Running the Quick Start Test Case from the Command Line

At a Linux system prompt, do: $TDSDIR/wavemakerplus -b stil-ascii.scn

3.2 Pattern Validation

A Quick Start test case for pattern validation, stil-c3360-pv.tgz, contains a STIL pattern file (sio85.stil) to be converted to a Chroma 3360 format, and a DUT netlist that the STIL pattern was generated from (i8085a.v) to be simulated with the STIL pattern for validation purposes.

NOTE: This test case requires access to one of the three leading logic simulators: Cadence IUS 8.0 or later, Mentor Questa 6.6 or later, or Synopsys VCS v2009.12 or later.

Running the Quick Start Test Case inside WaveMaker+

Description	Example Linux Commands
1. Un-tar the included test case	`tar xvfz stil-c3360-pv.tgz`
2. Go into the right directory	`cd stil-c3360-pv`
3. Invoke Solstice GUI	`$TDSDIR/wavemakerplus`
4. Open the flow ("Scenario")	`Double click on "stil-c3360-pv.scen"`

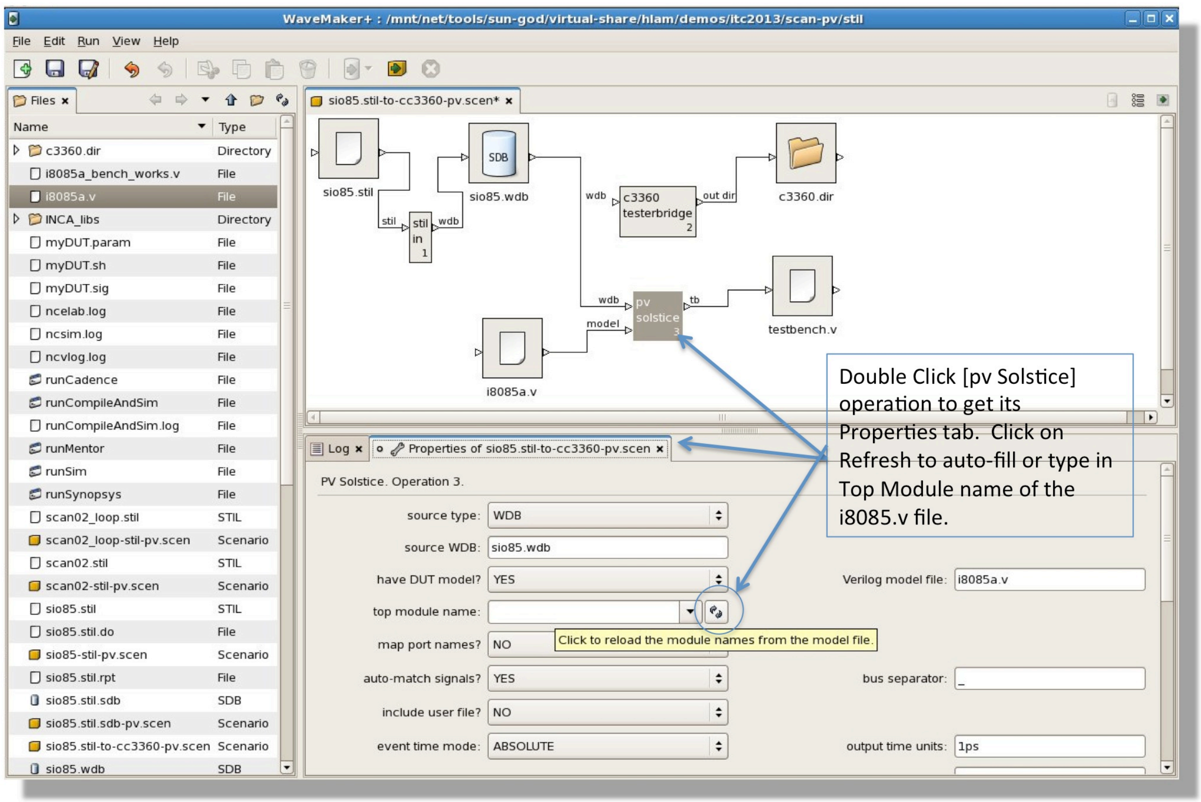
5. Run selected operations by control-click on [stil in] and [c3360 testerbridge] boxes to convert the STIL file into c3360 files and stored in the "c3360.dir" folder.	`Highlight [stil-in] and [c3360 testerbridge] boxes and right click to select "Run Selected"`
	
	`When process finished, double click on the "c3360.dir" folder to see the result files.  Double click on the files to view content.`

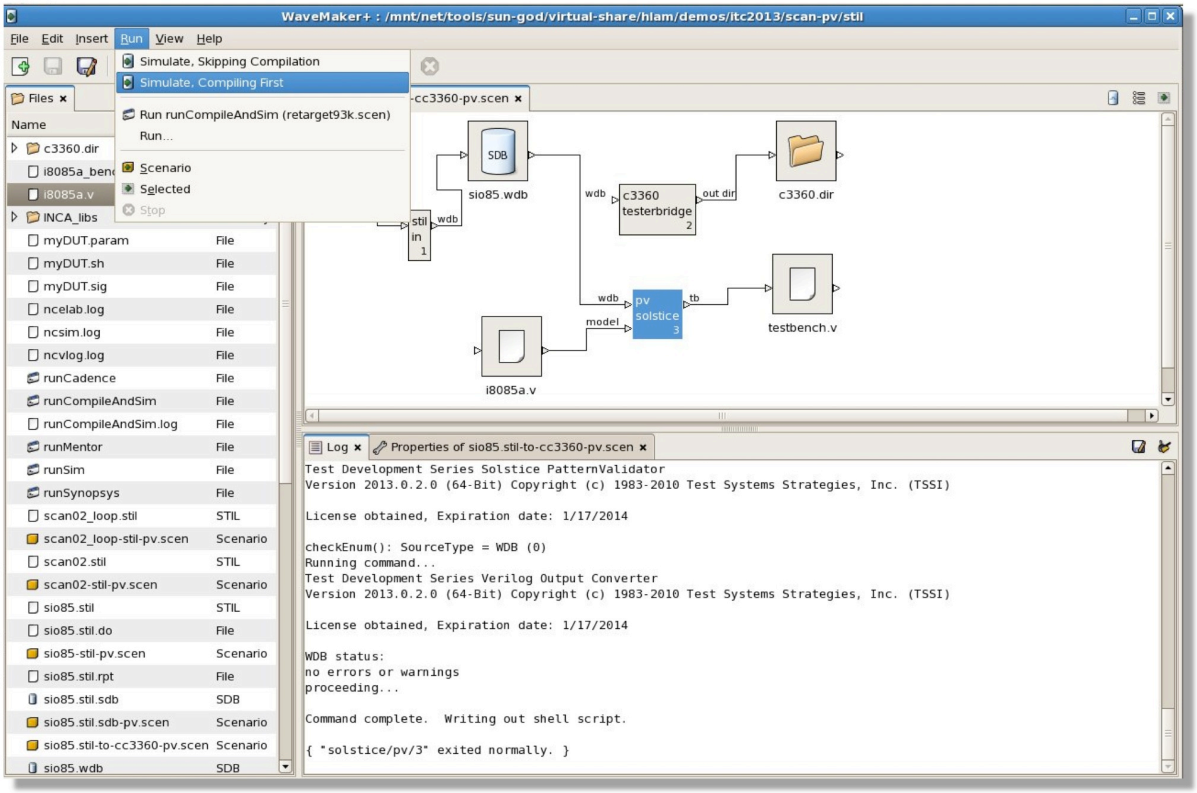
6. Validation. The concept here is that any modification needed for the c3360 output files is to be done at the SDB database stage, then re-convert using the c3360 testerbridge. The reason is that when the real silicon is being tested on the physical tester, errors can be compared with the original waveform and traced to the root cause.	`Connect sio85.sdb database to the input port of [pv solstice]`
	
	`The i8085a.v Verilog file is the DUT's netlist that should be available from the design team where the STIL scan pattern was generated from.`
	
	`The output "testbench.v" will be generated by "pv solstice"`

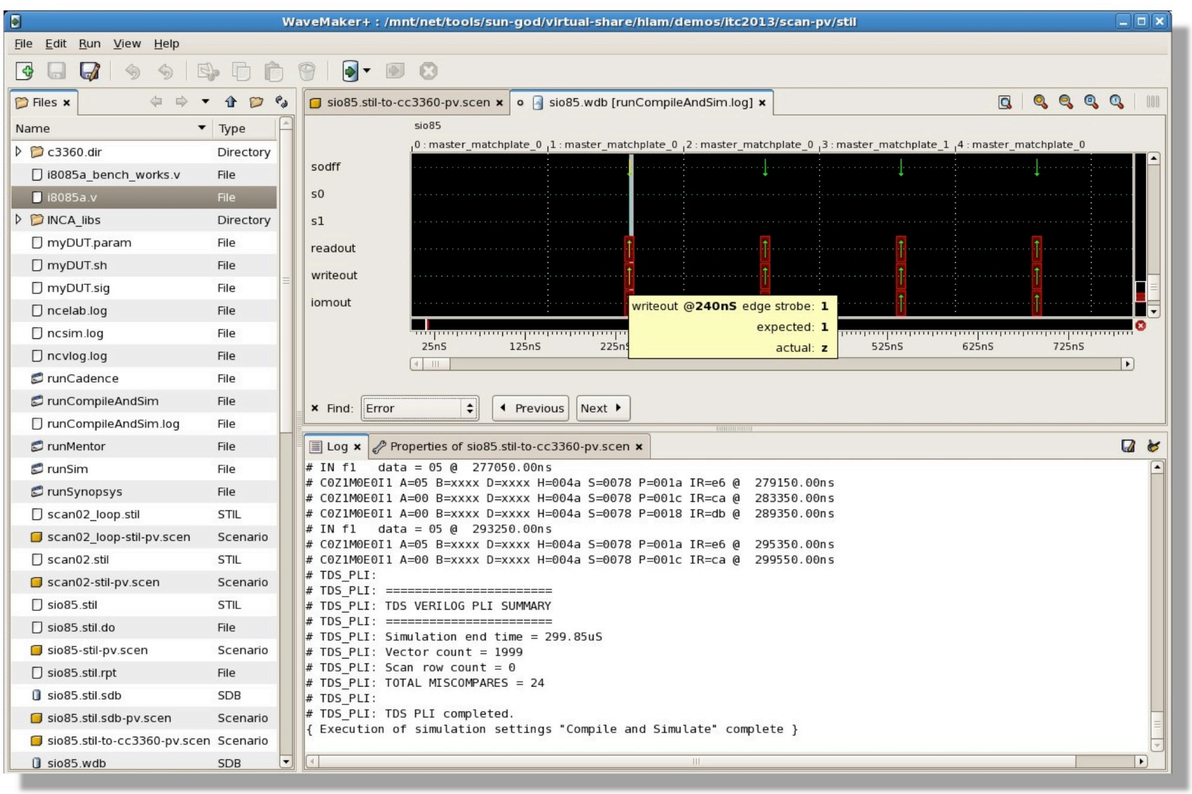
7. Configure Solstice PV (see picture)	`Double Click on "pv solstice" to open its Properties tab.`
	
	`a. Look for the "top module name:" field and click on the "reload symbol" to the right of the field to automatically search and fill in the top module name, or one can always type in the name if known.`
	`b. Scroll to the bottom of the tab to find the "target simulator:" field and select the one you have access to.`

8. Compile Solstice PV	`Right click to select the "pv solstice" box and select Run->Selected.`

	Note: don't select "Run->Scenario" OR all operation boxes will be executed. In this case, you don't need to repeat previous operations and only want the "pv solstice" operation to be executed.
9. Perform Compile Verilog and Simulate testbench	Select "**Run->Simulate,Compiling First**" and you'll see simulation and resulting waveform being displayed on the fly.




WaveMaker+ : /mnt/net/tools/sun-god/virtual-share/hlam/demos/itc2013/scan-pv/stil
File Edit Insert Run View Help
Simulate, Skipping Compilation
Simulate, Compiling First
Run runCompileAndSim (retarget93k.scen)
Run...
Scenario
Selected
Stop
Files
Name
c3360.dir
i8085a.v
INCA_libs
myDUT.param File
myDUT.sh File
myDUT.sig File
ncelab.log File
ncsim.log File
ncvlog.log File
runCadence File
runCompileAndSim File
runCompileAndSim.log File
runMentor File
runSim File
runSynopsys File
scan02_loop.stil STIL
scan02_loop-stil-pv.scen Scenario
scan02.stil STIL
scan02-stil-pv.scen Scenario
sio85.stil STIL
sio85.stil.do File
sio85-stil-pv.scen Scenario
sio85.stil.rpt File
sio85.stil.sdb SDB
sio85.stil.sdb-pv.scen Scenario
sio85.stil-to-cc3360-pv.scen Scenario
sio85.wdb SDB
-cc3360-pv.scen
SDB
sio85.wdb
wdb
c3360 testerbridge
out dir
c3360.dir
stil in
wdb
pv solstice
tb
model
testbench.v
i8085a.v
Log
Properties of sio85.stil-to-cc3360-pv.scen
Test Development Series Solstice PatternValidator
Version 2013.0.2.0 (64-Bit) Copyright (c) 1983-2010 Test Systems Strategies, Inc. (TSSI)
License obtained, Expiration date: 1/17/2014
checkEnum(): SourceType = WDB (0)
Running command...
Test Development Series Verilog Output Converter
Version 2013.0.2.0 (64-Bit) Copyright (c) 1983-2010 Test Systems Strategies, Inc. (TSSI)
License obtained, Expiration date: 1/17/2014
WDB status:
no errors or warnings
proceeding...
Command complete. Writing out shell script.
{ "solstice/pv/3" exited normally. }


WaveMaker+ : /mnt/net/tools/sun-god/virtual-share/hlam/demos/itc2013/scan-pv/stil
File Edit Run View Help
Files
Name Type
c3360.dir Directory
i8085a_bench_works.v File
i8085a.v File
INCA_libs Directory
myDUT.param File
myDUT.sh File
myDUT.sig File
ncelab.log File
ncsim.log File
ncvlog.log File
runCadence File
runCompileAndSim File
runCompileAndSim.log File
runMentor File
runSim File
runSynopsys File
scan02_loop.stil STIL
scan02_loop-stil-pv.scen Scenario
scan02.stil STIL
scan02-stil-pv.scen Scenario
sio85.stil STIL
sio85.stil.do File
sio85-stil-pv.scen Scenario
sio85.stil.rpt File
sio85.stil.sdb SDB
sio85.stil.sdb-pv.scen Scenario
sio85.stil-to-cc3360-pv.scen Scenario
sio85.wdb SDB
sio85.stil-to-cc3360-pv.scen
sio85.wdb [runCompileAndSim.log]
sio85
0 : master_matchplate_0 1 : master_matchplate_0 2 : master_matchplate_0 3 : master_matchplate_1 4 : master_matchplate_0
sodff
s0
s1
readout
writeout
iomout
writeout @240nS edge strobe: 1
expected: 1
actual: z
25nS 125nS 225nS 525nS 625nS 725nS
Find: Error
Previous
Next
Log
Properties of sio85.stil-to-cc3360-pv.scen
IN f1 data = 05 @ 277050.00ns
C0Z1M0E0I1 A=05 B=xxxx D=xxxx H=004a S=0078 P=001a IR=e6 @ 279150.00ns
C0Z1M0E0I1 A=00 B=xxxx D=xxxx H=004a S=0078 P=001c IR=ca @ 283350.00ns
C0Z1M0E0I1 A=00 B=xxxx D=xxxx H=004a S=0078 P=0018 IR=db @ 289350.00ns
IN f1 data = 05 @ 293250.00ns
C0Z1M0E0I1 A=05 B=xxxx D=xxxx H=004a S=0078 P=001a IR=e6 @ 295350.00ns
C0Z1M0E0I1 A=00 B=xxxx D=xxxx H=004a S=0078 P=001c IR=ca @ 299550.00ns
TDS_PLI:
TDS_PLI: ======================
TDS_PLI: TDS VERILOG PLI SUMMARY
TDS_PLI: ======================
TDS_PLI: Simulation end time = 299.85uS
TDS_PLI: Vector count = 1999
TDS_PLI: Scan row count = 0
TDS_PLI: TOTAL MISCOMPARES = 24
TDS_PLI:
TDS_PLI: TDS PLI completed.
{ Execution of simulation settings "Compile and Simulate" complete }

4. Contact TSSI

When contacting TSSI, please have your customer number ready. This number is needed before you can be given support. (The customer number is in the heading of your license file. Evaluating customers will see either "eval" or a temporary customer number.)

If you have questions about your software license, or want your license file e-mailed to you, contact the TSSI Order Fulfillment department via either of the following methods:

Email: oe@tessi.com

Telephone: +1 503-764-2308

For platform or CPU ID changes, please contact the TSSI Sales Representative from your area. For a list of contacts worldwide, please visit our website at www.tessi.com.

For software installation questions, contact your TSSI representative, or contact Customer Support via either of the following methods:

Internet: www.tessi.com and select "support"

Email: hotline@tessi.com

Telephone: +1 503-764-2308

TSSI World Headquarters is located at the following address:

920 SW Sixth Avenue, 12th Floor
Portland, Oregon 97204
USA