TESEDA Agilent Technologies

Application Note AP-0005

Transporting STIL between the Teseda V500 and the Agilent 93000 SOC Tester

Original date issued: **February 16, 2004**

Abstract

This Application Note addresses STIL transportability between the Teseda V500 Integrated Circuit Validation System and the Agilent 93000 SOC Test System. The note describes what to consider when planning DFT-Focused testing using the V500, and also how to back-annotate test changes to preserve STIL compatibility between the V500 and the Agilent 93000.

Web site references

http://www.teseda.com

Home page for Teseda Corporation.

http://www.agilent.com/see/dft

Agilent's site for Design-for-Testability (DFT) Solutions.

http://grouper.ieee.org/groups/1450/

Home page for the IEEE 1450—Standard Test Interface Language (STIL).

Overview

This application note describes the transportability issues to consider when setting up test programs that are intended to be compatible with both the Teseda V500 and the Agilent 93000. These tests rely on STIL as the medium for specifying test procedures and expected outputs. Figure 1 illustrates a general process for managing STIL test programs between the two systems.

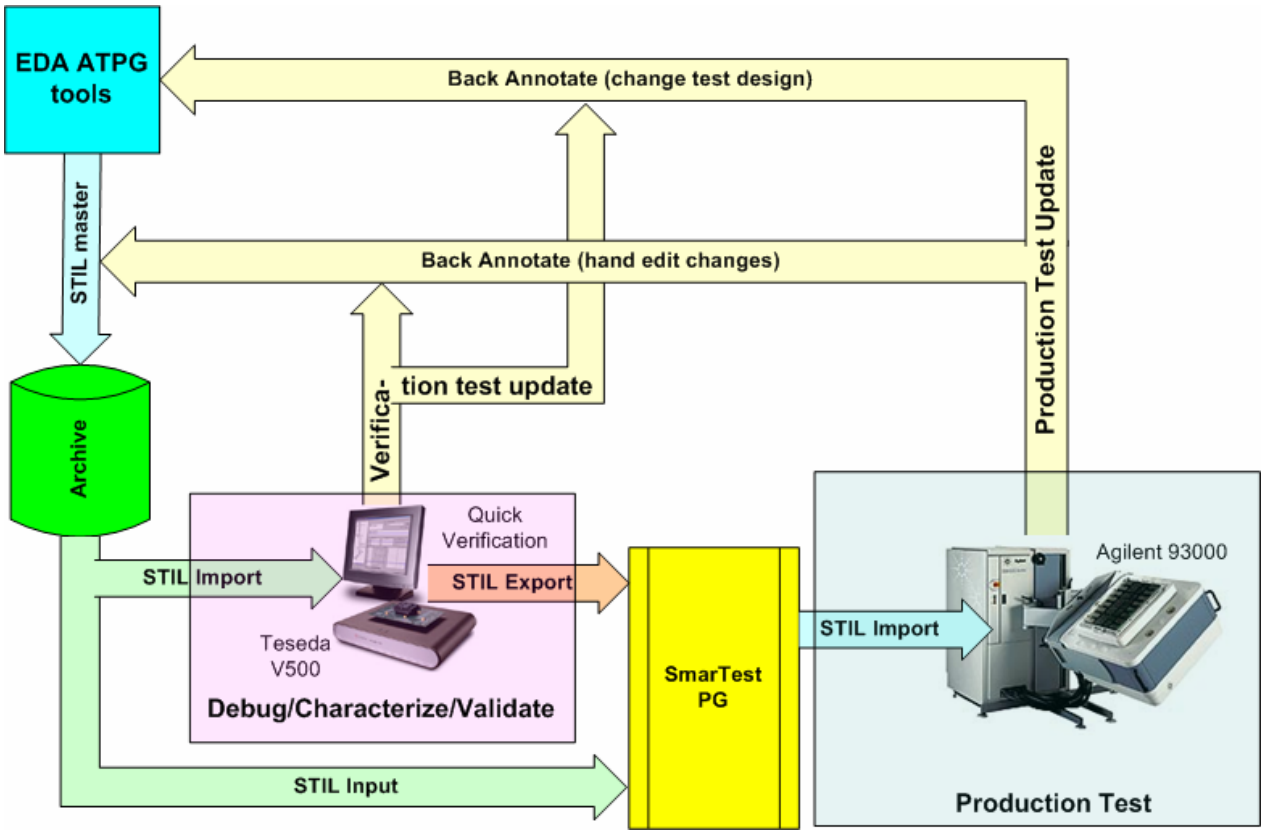

Figure 1—Managing transportable STIL between the V500 and the Agilent 93000

Transportability considerations

To ensure that STIL data imported into the V500 can subsequently be transported to the Agilent 93000 some key differences between the two systems must be taken into account. Most differences have to do with timing. By studying the STIL waveform tables in the timing block, you can identify most potential problems. It may be necessary to alter the timing to ensure that there are no violations of system specifications as the tests are transported from one system to the other.

The primary system differences to keep in mind are:

- **Signal timing and accuracy.** The V500 waveform capability (on its data pins) does not have the same number of edges nor the accuracy of the 93000. The waveform tables in the STIL timing block will reveal any waveform incompatibilities between the V500 or the Agilent 93000.
- **Clock drivers.** The V500 has a limited number of precision clock drivers, while the 93000 has full per-pin precision capability.
- **Cycle times.** The Agilent 93000 is capable of vector cycle times below 20ns, whereas the V500 is not. Therefore, the STIL period specifications should not require a cycle time faster than 20ns.
- **Number of pins.** The Agilent 93000 allows configurations of more than1000 pins whereas the V500 is limited to 348. The STIL `Signals` lists all of the primary signals. If the number of signals exceeds the resources of either system, test modifications should be made.

Addressing all these issues, by itself, does not ensure transportability of tests, however. A certain amount of planning is necessary to ensure that tests developed on one of the systems will transport seamlessly to the other. Some of these planning issues are addressed in the next section.

To minimize manual editing/debugging due to the feature set differences between the V500 and the Agilent 93000, a good strategy is to develop a test on the V500 (with the smaller feature set) and then transport to test program the Agilent 93000. This is generally a satisfactory solution, since the V500 is capable of handling the majority of DFT test requirements.

STIL for transportability—permanent and non-permanent changes

The V500 handles a full range of DFT-Focused™ tests including stuck-at scan (DC scan), delay fault scan (AC scan), I_{DDQ}, and BIST (both memory and logic). The V500's repertoire also includes test compaction schemes such as Embedded Deterministic Test (Mentor Graphics TestKompress), XDBIST and DBIST (Synopsys SoCBIST), and OpMISR (Cadence Encounter).

Each of these test types has its own STIL requirements, with corresponding ways of handling STIL to ensure its transportability between the Teseda V500 and the Agilent 93000.

When planning DFT testing on the V500, there are two categories of test changes to consider:

- **Non-permanent changes**. These are applied during test debugging and validation. These changes adjust the original test details so that parts of a test can be evaluated. Such changes, if made permanent, could create incompatible STIL between the V500 and the Agilent 93000. However, transportability is preserved since non-permanent changes are not back annotated to the reference STIL.
- **Permanent STIL changes**. These adjustments, compatible with both the V500 and Agilent 93000, are back annotated to the reference STIL. Although such changes alter the original ATPG output, they do not compromise transportability.

Scan tests

In scan-related tests, few edits are actually possible because of the nature of the test generation process. An ATPG tool with circuit knowledge generates a set of test vectors, customized to a specific device's circuitry. Any alterations to the test typically involves either a complete or incremental re-generation of part or all of the test. Therefore, adding vectors or pins, or arbitrarily modifying existing vectors won't produce useful results. However, certain edits could have value, such as masking by applying **X** values for parts of a vector.

More common changes, though, are those altering the timing relationships between signals, and the length of the vector period. Often these kind of changes are done during debug as a shmoo plot in conjunction with altering power, voltage, or other parameters. The V500 makes is easy to manage all these kinds of changes during test debug and validation, particularly shmooing and timing adjustments, before finalizing tests for production.

I_{DDQ} tests

The ATPG-generated test is based on I_{DDQ} fault models. In I_{DDQ} testing, it is not advisable to alter a vector. Altering vectors have the potential to affect the fault coverage of I_{DDQ} testing, just as such changes risk reducing coverage for scan testing or any other tests that have been fault graded.

However, non-test edits—such as those aimed at shutting off parts of the circuit not being tested or shmooing ranges of test parameters—are often done since the ATPG-generated patterns may be inadequate. Such permanent edits usually involve the addition of control signals to the test and the specification of additional bit sequences. To ensure transportability, any changes in control signals should be made to the execution macros or procedures rather than by adding executable vectors. All major EDA systems utilize macros and/or procedures to execute tests rather than vector sequences.

BIST tests

BIST tests are rarely generated by an ATPG tool or program generator. They are typically written from scratch by an individual with knowledge of the structure of the BIST engine. The tests are often built on top of a "template" such as a IEEE 1149.1 template. These templates, written in STIL, contain macros and procedures to handle the common controls used to set up the BIST execution. Addition of the specific commands to the templates is often all that is necessary. These edits are preferably performed on the STIL template itself, however; and not in the tester.

Other common BIST edits are the alteration of bits in the control streams to make some of the internal signals available for monitoring, (via external instrumentation, for example). Additionally, analysts often change the timing and add shmoo for debug purposes. See Additional BIST considerations, below.

Data compaction schemes

When using data compaction software—such tools as Mentor Graphics TestKompress, Cadence Encounter, and others—any editing of the data stream in for DFT-Focused tests is difficult. To do so would require a complete understanding of the internal vector generation process and the test data compaction scheme in use. Therefore, timing alterations in STIL are usually all that can be expected in these cases.

Temporary edits may be necessary that make it possible to capture the uncompacted raw data from the scan chains. The collected information is then processed by the EDA system during the debug process. However, since such edits are not permanent, transportability is not an issue.

Managing transportable STIL

Any STIL that you can successfully import into theV500, validate and debug, then export out will be compatible with the Agilent 93000. The key is to ensure that any required STIL modifications are properly back annotated and archived. This section address the key steps in managing STIL programs to be compatible with both the V500 and the Agilent 93000. See Figure 1 on for a diagram of this process.

For more information about STIL importing and exporting procedures in either the Teseda V500 or the Agilent SmarTest PG and 93000, see the corresponding product documentation for these systems.

Importing STIL into the V500

You can import a STIL file directly into the Teseda WorkBench (TWB) software of the V500. To prepare the STIL for running a test, you must also import or create a pin map as part of a project, then compile the project data. Since neither the pin mapping nor the compiling change the original STIL, you can export the STIL intact. See Exporting STIL from the V500 below if you want to make changes to the timing information or the test vectors before exporting the STIL

from the V500 to another application, such as the SmarTest PG environment of the Agilent 93000.

Validating and debugging DFT tests in the V500

At this stage, you can make any number of non-permanent changes to your tests while validating and debugging the source STIL imported into the V500. Keep track of all the permanent STIL changes you want to make; these are the STIL changes to be back-annotated.

Exporting STIL from the V500

To export STIL from the V500, you can choose either the PROJECT > EXPORT STIL or the PROJECT > EXPORT VECTORS AS STIL menu options in the Teseda WorkBench (TWB) software.

Using the PROJECT > EXPORT STIL menu command, the TWB writes a STIL file that reflects any changes made to the waveform timing and event data of an imported file. If you have edited a signal which is part of a group definition, the signal group is replaced with definitions for each signal because the original signal grouping characteristics may no longer apply.

Any editing changes you have made in the TWB vector table may be saved by using the PROJECT > EXPORT VECTORS AS STIL menu command. This export preserves any vector table edits and writes the signal and timing pattern data as "flattened" vectors. This exported "flat" STIL can be read into a 93000 and used to quickly verify compatibility of changes.

The vector table is the "result" of compiling the STIL source, so edits made to the vector table do not modify the STIL source. Therefore an export of the vector table does not look entirely like the imported STIL. The exported STIL does not retain the procedure structure or design hierarchy of the originally imported STIL, but it is STIL just the same. The major difference is that the PATTERN block is translated into a series of vector statements, and the MACRO and PROCEDURE calls are eliminated.

Importing STIL into the Agilent 93000

STIL is imported into the 93000 through the Agilent SmarTest PG environment. There are some setup requirements to be addressed so that the ATPG-generated STIL can be accepted seamlessly. For specific information, see the SmarTest PG documentation.

STIL may be processed through the V500 and exported to SmarTest PG, or may be input directly from the ATPG output as input to SmarTest PG. STIL exported from the V500 can be used as a means for "quick verification" of the transportability of changes. This check takes place before back annotation.

Back annotating STIL to a master reference file

Once you have debugged and validating your DFT tests in the V500 environment, it is critical to ensure that any STIL adjustments are back annotated to the master reference STIL and the file is archived.

Back annotation can take two paths: either modifying the master reference STIL by hand and saving the edits, or redesigning the tests through changes made in the EDA ATPG tools. By and large, the edits will encompass two categories:

- Changes of the timing relationships between signals. These changes are described in more detail, above, in Transportability considerations.
- Alterations of the data in some of the vectors. Typically, these are the outcome of the STIL debug process, such as masking with **X** placeholders to exclude looping vectors or other portions of a test.

Rather than edit the STIL itself, any test changes that can be incorporated through the EDA ATPG tools should be. The original EDA-produced file needs to be the reference or master test archive for both the V500 and Agilent 93000 systems; all permanent changes should be reflected in the original STIL code.

Back-annotating STIL with this approach provides a method for saving the test information in a standard language that guarantees transportability between systems. To be successful at this, a level of programming discipline and control will be necessary to insure that all changes are properly done and that some system of version control is available.

NOTE

It is a good idea to use a version control system, like VCS, to guarantee the master reference file can be checked out and checked in by multiple users without the risk of having them write over each other's work.

Verifying STIL compatibility

Regardless of the which back annotating method you choose for making changes to the original STIL, all permanent changes can be run and tested on the V500.

Once the final version of a STIL test is ready, the archived version of the STIL should be imported into the V500, compiled and run. The same archived file may sent through Agilent SmarTest PG to ensure that nothing has been accidentally lost during the back-annotation process. If any failures show up in the Agilent 93000 where no errors are expected, then the problems can be traced back to the V500 and uncovered.

Additional BIST considerations

There are two kinds of Built In Self Test (BIST) engines, logic and memory, both generated without the benefit of ATPG tools. However, many BIST tools are available explicitly for the purpose of creating BIST tests, even if the test language is not in STIL but in another output such as WIGL. The recommended practice is to create STIL templates for the execution control of the BIST engine. The most common control mechanism for both memory and logic BIST is the IEEE 1149.1 Test Access Port (TAP).

For running BIST on the V500, all that would be necessary is an 1149.1 template utilizing macros to handle shifting in instructions to the `IR (SHIFT_IR)` and shifting data into and out of the data registers (`SHIFT_DR`) along with a run BIST procedure that outputs a specific number of clock cycles (`RUNBIST`).

The `Pattern Block` contains calls to the various procedures to load up instructions in the `IR` and load/examine/capture data in the data registers. Each macro/procedure call has a label attached to identify it so that changes done using the V500 can be easily back-annotated where needed (the V500 editor identifies patterns by their labels).

It is critical to note that the V500 Teseda WorkBench software expects tests in a very specific STIL format. If this format is not adhered to, information useful in the debug process is left out.

For additional information about how the V500 TWB software handles STIL, see the *V500 Teseda WorkBench and Hardware User's Guide*, particularly Appendix C, which describes how to handle post shift control and other unique characteristics of different ATPG outputs.