LSI TEST SUPPORT SYSTEM —"PreTestStation" VIRTUAL TESTER—

SANO Naoki [*1] ***KUBO Norio*** [*1] ***OGASAWARA Atsushi*** [*1] ***ANZAI Sadaki*** [*1]

The "PreTestStation" virtual tester is an LSI test support system, which integrats Yokogawa's own technologies of EDA software "VirtualICE" and IC testing. The PreTestStation, which consists of an accurate and high-speed ATE model written in Verilog-HDL and a tester OS integrated with testing and debugging capabilities, runs on a Verilog simulator at a workstation and offers a virtual test environment common.

The PreTestStation facilitates an early verification of test programs before the completion of the first silicon without a real tester by simulating test programs with a target DUT model and an ATE model. This results in the reduction of testing and debugging time after the device is completed. This paper outlines the PreTestStation and its application results.

INTRODUCTION

Recently LSIs are becoming increasingly large-scaled, advanced, and complicated. Accordingly LSI processes have become more microscopic and an increased number of system LSIs have been developed. This makes it an increasingly burdensome task to properly test LSIs. Thus reducing the time for debugging LSI test programs and the testing costs has become an issue to manage.

Previously there was no choice but to fully debug test programs on an LSI tester using the LSI devices after they were produced. With this method, if any problem is found during the debugging work, it is extremely difficult to determine whether it is a problem in the test program itself, the LSI devices, the LSI design, the performance board, or some other limiting factor resulting from the LSI tester's characteristics, thereby demanding an extra amount of time for isolating and solving the problem [(1)]. Consequently it took a considerable amount of time before proceeding to the intended primary task or the testing of the LSI device, thus prolonging the total time for debugging the LSI test program.

Therefore, to reduce the necessary time period for debugging test programs, it is desirable to verify the programs, and to locate and solve or forecast as many problems as possible without using the LSI tester before the LSI device is completed.

To this end, Yokogawa has developed and commercialized the PreTestStation virtual tester as one of our LSI test support systems, which runs on an HDL (Hardware Description Language) logic simulator on a workstation. The PreTestStation enables users to build a virtual test environment that is similar to the actual one, and to verify test programs prior to the completion of the target LSI device. Being a common test platform to be used by test engineers and IC designers, the PreTestStation facilitates the debugging of test programs while establishing a close cooperation between the engineers and designers.

The PreTestStation was developed by merging Yokogawa's testing technology with EDA (Electronic Design Automation) technologies, and the idea behind it is based on the concept of our "VirtualICE" hardware/software co-verification tool [(2)]—namely, the realization of early verification of hardware/software before the completion of hardware prototypes.

This paper describes existing debugging methods of test programs and their shortcomings, as well as outlines the PreTestStation's system configuration and its application results.

*1 ATE Business Headquarters

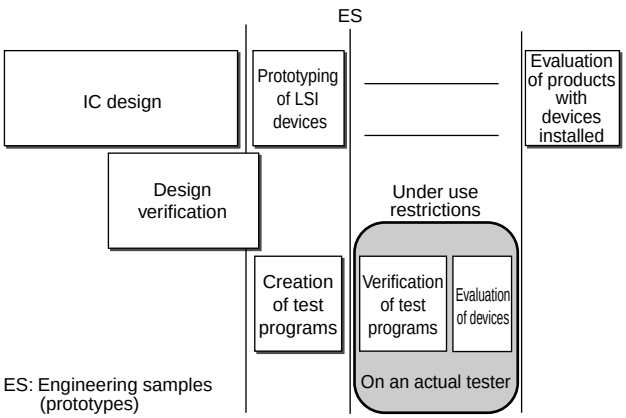

Figure 1 Debugging Flow with Existing Methods

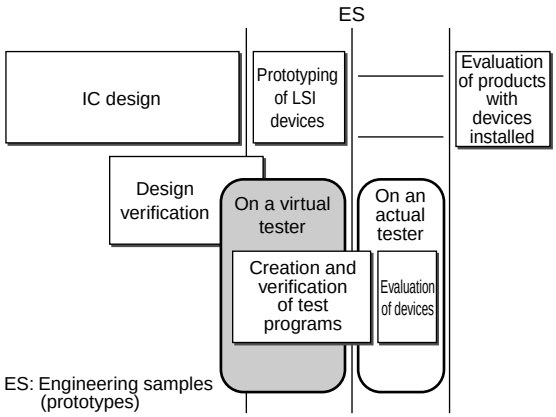

Figure 2 Debugging Flow Using the PreTestStation

EXISTING METHODS OF DEBUGGING TEST PROGRAMS AND THEIR SHORTCOMINGS

With existing methods, as shown in Figure 1, test engineers lead the debugging work in general, where they verify the test program first using completed actual LSI devices and then evaluate the devices, and finally the verified devices are entrusted to IC designers who evaluate products that incorporate the devices.

However, these processes have the following shortcomings which prevent any further reduction of the time period for debugging test programs:

- Test engineers are forced to verify test programs on an actual LSI tester that have not been sufficiently pre-debugged. Thereby, when a problem occurs, this makes it quite difficult to determine whether the problem exists in the test program itself, the LSI devices, the LSI design, the performance board, or some other limiting factor resulting from the tester's characteristics. Therefore, these existing methods require an excessive period of time.
- Although the tester is requisite to debug the test programs, it is not always available when necessary because of the restricted available time on an actual LSI tester, and this further hinders efficient debugging.
- Even though tracking down problems often requires help from the IC designers, there is a considerable gap between a test environment centering on LSI testers for test engineers and a design environment for IC designers. As a result, test information and design information cannot be easily associated with each other across the two different environments, and even more time is consumed to solve the problems.
- If a problem originating in the LSI design occurs in the course of the debugging work, the devices must be re-designed and re-prototyped, causing an increase in the total cost and considerably delaying the development schedule.

OUTLINE OF NEW SYSTEM

Positioning of PreTestStation

Figure 2 shows a debugging flow of test programs with the PreTestStation (hereafter referred to as PTS). With this latest system, test programs are first verified to the greatest possible extent using a virtual tester instead of the actual tester before the device is completed. When the device has been realized, the remaining part of the test programs which could not be checked on the virtual tester are verified using the actual tester and then the device itself is intensively re-evaluated.

System Configuration and Features

Figure 3 shows the PTS system configuration. The PTS is a virtual test environment built on an HDL logic simulator (Verilog/Bilingual simulator) that runs on a workstation, and consists of an ATE (Automated Test Equipment) model which acts as the actual tester's hardware, a tester OS (TS Virtualizer) which acts as the actual tester's software, and a PTS debugger which provides a debugging environment for the test programs.

A test program is comprised of test patterns and a main program, both of which have been installed to the ATE model and the tester OS. A PFB (performance board) model contains a DUT (Device Under Test) model—namely, DUT design information which is expressed in the form of gate-level logical netlists or RTL design data. By simulating the test program and the PFB model simultaneously, the test program can be verified effectively and accurately.

The ATE model simulates the functions and timing provided by the actual tester's hardware, which are perfectly reproduced in the Verilog-HDL. Yokogawa's HDL design technology and years of behavior modeling expertise enable the ATE model to run at high speed and maintain accuracy as well. Currently the PTS simulates the TS6000H$^+$, our field-proven VLSI tester, as the ATE model exclusively. The TS Virtualizer tester OS is the same as that used by the actual tester's software. The PTS debugger is comprised of the TSWB integrated tester debugging tool which provides the same screen and operations as that of the actual

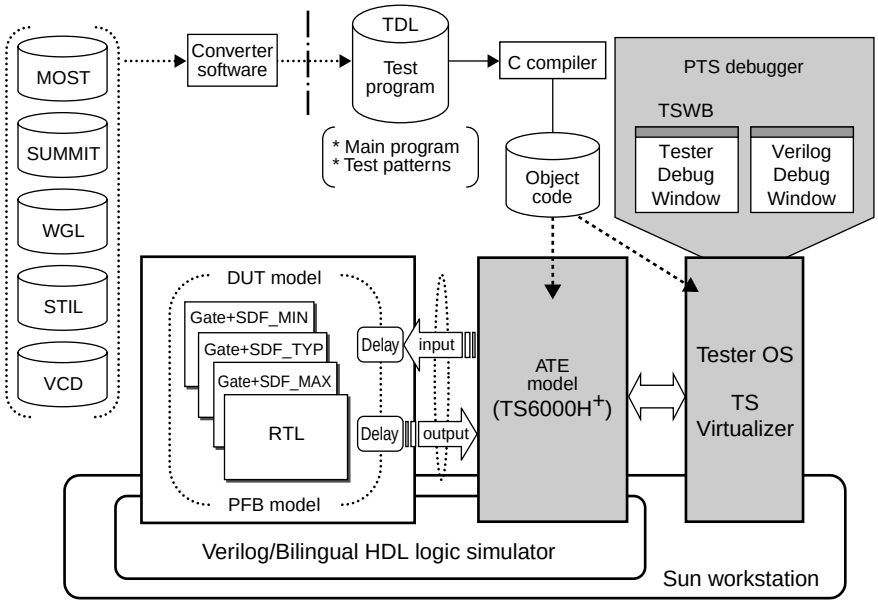

Figure 3 PreTestStation System Configuration

tester, and a waveform display tool that comes standard with the HDL logic simulator.

Meanwhile, the test program is identical with the one to be downloaded to the actual tester. Basically it is written in TDL (Test Description Language), Yokogawa's standard test language, but other languages such as MOST, SUMMIT, WGL, and STIL can be converted into TDL using our converter software for making the most of the earlier test programs. It is also possible to directly convert VCD (Value Change Dump) files, which are dumps produced by the HDL logic simulator and popular with IC designers, into test patterns.

Furthermore, the Virtualizer mode is available for concise verification of the main program portion of the test program without using the HDL logic simulator.

Debugging Procedure

The following shows the typical procedure for the preparatory arrangements and execution steps for running the PTS.

(1) Preparatory Arrangements

① Create a TDL test program and an executable object code.
- Compile the main program and test patterns.

② Create HDL files for a PFB model.
- Connect wiring of DUT model pins and ATE model pins with HDL.
- Set delays for each of the wiring connections between DUT and ATE model pins as necessary (the default is no-delay).

(2) Execution Steps

① Start the TSWB integrated tester debugging tool.

② Start the PTS virtual tester (automatically started by the HDL logic simulator).

③ Load the test program (the main program and test patterns).

④ Execute continuous simulation of test patterns using the test program.
The PTS maintains the DUT model's internal statuses after executing a test pattern, and executes the next pattern.

⑤ Stop the HDL logic simulator immediately if a pattern failure is detected.
- Display the failed address, pin, and the difference between the expected and actual outputs on the Fail Map screen.
- Perform necessary analyses and debugging at the failed address using the waveform display tool.

⑥ Check related timing margins using a Shmoo plotting tool.

Advantages of the New Debugging Method

This new simulation-driven method of debugging test programs through PTS offers the following advantages:

- As test programs can be pre-debugged satisfactorily, more problems can be solved or anticipated in advance.
- Test programs (the main program and test patterns) that have been corrected on the PTS can also be run on the actual tester.
- As PTS is a virtual test environment that runs on a workstation, it can be used anytime, anywhere.
- Test engineers and IC designers can share their test environments each other, which makes it possible for them to work together efficiently in investigating the cause of problems.
- Even if a problem originating in the LSI design occurs in the course of the debugging work on the PTS, IC designers can receive feedback from test engineers in no time and do not necessarily have to re-prototype the LSI devices.
- The actual tester can be used chiefly for the testing of LSI devices, promoting the effective use of both the virtual and actual testers.

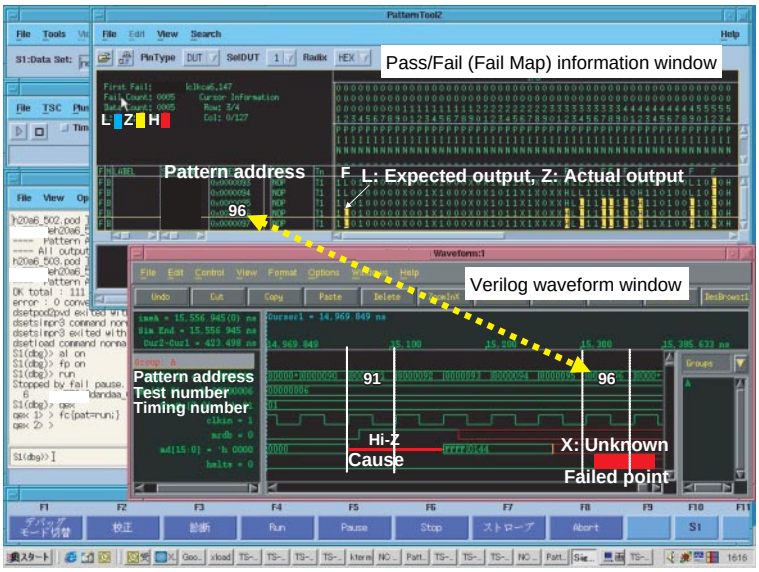

Figure 4 A PreTestStation Debugging Screen

SUMMARY OF APPLICATION RESULTS

The following summarizes some of the excellent results of the PreTestStation obtained from user applications.

(1) It is a very powerful tool for verifying test programs (both the main program and test patterns).

(2) Timing margins can be checked using a Shmoo tool.

(3) HDL logic simulator's waveform tool can display pattern addresses, test numbers, and input timings. Users can easily associate them with the Fail Map screen and analyze failures, a totally new debugging environment for test engineers. Figure 4 shows an example of the debugging screen.

(4) The processing time is fast enough for practical application. Therefore when a test pattern is comprised of several components, having more than one PreTestStation that processes them simultaneously can reduce the total processing time.

(5) By determining the relationship between power supply voltages (minimum, typical, and maximum) and gate delays in a DUT model's cell library (Gate_MIN, Gate_TYP, and Gate_MAX), test programs that accept margins in power supply voltage and frequency can also be verified.

(6) By utilizing the Pass/Fail results of test patterns that incorporate both the gate delays in a DUT model's cell library and the PFB model delays, various corrective procedures to enable the patterns to pass can be forecast, such as modifying input timings and strobe output timings, masking output patterns, and redesigning test patterns. This makes it possible to provide sufficient conditions that enable a test program for LSI prototypes to smoothly evolve into one for commercial products which takes their individual variations into account.

CONCLUSION

This paper has introduced the PreTestStation virtual tester which is part of our LSI test support systems and realizes earlier verification of test programs.

Being a common test platform to test engineers and IC designers, the PreTestStation provides a new debugging method of test programs. We are confident that it will contribute greatly to reducing the total period of time and cost for debugging LSI test programs.

We will continue to vigorously provide customers with products that meet a variety of market needs. In addition, we will also offer opportunities to exchange opinions freely among test engineers or between test engineers and IC designers. ◆

REFERENCES

(1) Kaga Hiroshi, "Chapter 2: Basic Knowledge of ATE and Test Process Necessary for Designers," Design Wave Magazine, March 2001, pp. 35-45 (in Japanese)

(2) Kobayashi Fumihiko, et al., "VirtualICE: Hardware/Software Co-verification Tool," Yokogawa Technical Report, Vol. 45, No. 2, 2001, pp. 119-122 (in Japanese)