Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 1, 2004

Simulation based testing of complex electronic systems

Dan Simu[1]

Abstract – Complex electronic systems[3,21] are built of different electronic devices interacting between them. The devices are interacting using one or several common buses. The testing of each device is done as a functional testing but it doesn't prove the functioning of the hole system. Testing complex systems while simulating the environment with additional hardware could prove to be impossible. Simulating each device, assembling the system, generating the test program for the entire system and running the test while injecting errors and failures to some of the devices could be very useful for the assessment of the system reliability in time. This paper analyses the technical and the cost opportunities for the design of such a testing system.
Keywords: editing, Symposium, author

I. COMPLEX ELECTRONIC SYSTEMS

In our days the electronic devices could be found everywhere. The industry is building more and more complex systems. There are two sectors where complex systems are used on large scale: the aircraft industry and the automotive industry. Every year new devices are added to the previous architecture. An electronic complex system could be defined as a collection of electronic devices connected through one ore more buses performing together the tasks associated with the system. The system could have an hierarchical structure or could be only a collection of standalone devices performing different tasks at different moments. In the aircraft industry there are several subsystems (organized usually hierarchically) exchanging information on two or more buses (usually ARINC 429 or ARINC 629). In the automotive industry there are tens of devices using a CAN bus as a support for information exchange. Every device has its own functionality driven by an intelligent hardware running its own software. The usual testing procedure is to take each device and to verify it on a test bench. This type of verification is based on the ability of the test engineer to design the set of tests for a good coverage of the failures field of the specific unit. While the complexity of the systems is growing, the possibility of unforeseen situations becomes reality more often. For complex systems for economical reasons there are used Automatic Test Equipments with different Test Unit Adapters (interfaces between the standard equipment and the Unit Under Test) and different Test Program Software.

II. ATE BASED FUNCTIONAL TESTING[22,23,24,25]

The ATE based functional testing means that every stand alone device has a Test Requirement Definition designed by the manufacturer and based on this document, a suite of tests could be performed connecting the test equipment at the connector of the UUT. The device is seen as a black box and there is no access to the components inside. Usually the ATE is a collection of general purpose measurement and stimulus devices controlled by a computer. The computer is running the tests controlling all the measurement devices via a GPIB type bus or other common connections (serial, USB, parallel) . The functional testing represents a suite of tests organized to verify the complete functionality of the device. The testing design is based on the hardware architecture of the device. It verifies every block of the unit orderly, starting with the power block and continuing with the other blocs one by one. While this covers the entire hardware of the unit it assumes a reasonable functionality for the environment of the device. It means that the testing is more targeted to the hardware failures. Unfortunately for complex systems the verification of the software inside a device cannot be completely tested at the design stage. When the number of interacting devices is growing the number of possible situations is growing either. Using an ATE equipment to test several interconnected devices is almost impossible because of increasing costs.

III. COMPONENT MODELING AND SYSTEM SIMULATION[14]

Functional tests, because of their precise description are suitable for simulation. The UUT could be descript by a functional model. There is an open issue regarding the detail of modeling when simulating a device. The better and detailed model of the device means the more accurate and close to reality simulation. But a very detailed description could cost a lot of development time and also could

[1] Timteh Electronics Ltd.
Str. Plautius Andronescu 10, Timişoara, e-mail simu.dan@timteh.com

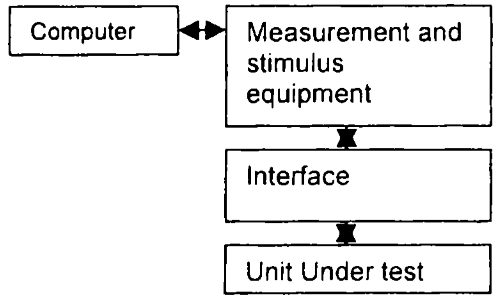
cost running time during testing. A functional description of the device based on the original testing requirements at least as the first testing iteration is convenient. The device could be descript using VHDL as a language. The new model will be easily tested based on the original functional test. Based on this modeling approach the entire system may be built as a library of models. The design of the VHDL models will have some supplementary features to enable the implementation of some new testing strategies and more important the error injection[13]. The model library is not enough to run the simulation. We have to add the model of the bus or busses used for information interaction and the model of the measurement and stimulus devices (the ATE models). The last component of the simulator will be the application running all the models. The application will have some important features. The models represents processes which are running simultaneously. The application like any VHDL simulator has to run all the models simultaneously based on a time sampling. None of the commercial software was built to run the VHDL models and to exchange data with other applications. But the ability to run test programs, to inject errors and create test reports is based on the possibility to exchange global data at specific moments with the simulator. The simulation environment will have an application running the models, an application running the test program and an application performing the error injection. All those applications will be able to exchange data and will be synchronized. A complete simulation will be very useful at the design stage of the system.

IV. OBJECTIVES OF THE SYSTEM LEVEL SIMULATION[5,2,3,7]

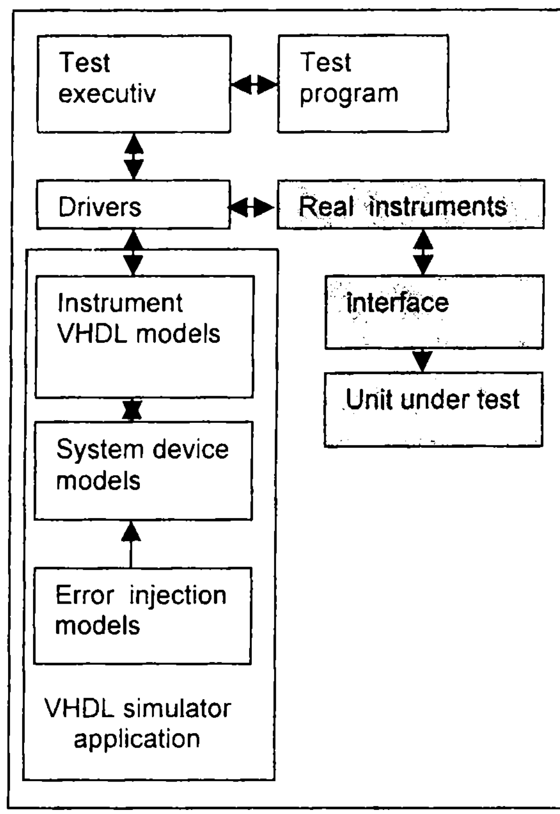
This paper presents the opportunity of the system simulation in connection with the system testing strategy. During the design stage, the possibility to simulate the system will improve the test requirement design of each component. It also will improve the verification of the software of each device and the implementation of a fault tolerant software. It also will allow the implementation for the hierarchical systems of a BIST (Built In Self Test) procedure[7] at the system level. During the service of the unit the testing is usually performed for maintenance reasons. In this situations the testing equipment is necessary to verify and diagnose if the unit is serviceable. For the complex systems the test design is never perfect. During service new scenarios are possible. While a standalone device could be available, the entire system is more difficult to have. In this cases a hybrid system partially simulated and partially hard implemented will prove very useful. For this reason the test program running application will be capable to control virtual instruments (real or simulated).

V. ENVIRONMENT ARCHITECTURE

The architecture of a testing environment for complex systems could have the following simple hardware structure:



and the following software structure:



VI. CONCLUSIONS

The simulation of the complex electronic systems together with the testing environment is improving the quality of test programs from the design stage. It also allows to implement fault tolerant systems and to better define the dependability of the system. The main inconvenient is the cost of model development. But avionic complex systems and automotive systems are using more standard units enabling general use library development. Also modeling at functional level will limit the complexity.

REFERENCES

[1] Norman E Fenton, Niclas Ohlsson"Quantitative Analysis of Faults and Failures in a Complex Software System", *IEEE Transactions on Software engineeringi*, Vol 26, No. 8, August 2000.

[2] S P Wilson, J A McDermid, C H Pygott, D J Tombs "Assesing Complex Computer based Systems using the goal structuring notation", *IEEE Transactions on Software engineeringi*, 1996

[3] Robert Tappe, Dietmar Erhardt, "Dynamic tests in complex systems", *ITC International Test Conference*, 2001

[4] Silvia Cataldo, Silvia Chiusano, Paolo Prinetto, Hans-joachim Wunderlich, "Optimal Hardware Pattern Generation for Functional BIST", 2001

[5] Richard C Scalzo, Michelle M. Hugue "A framework for Dependability Specification", *IEEE 1996*

[6] Ilham Benyahia "Using a Discrete Event Simulation to Test a Generic Architecture for Complex Real-Time Systems", *IEEE 1996*

[7] Eugene Zhang, Einat Yogev "Functional Verification with Completely Self-Checking Tests", *IEEE 1997*

[8] Henry Leung, Vinary Varadan, "System Modeling and Design using Genetic Programming", Proceedings of the First IEEE International Conference on Cognitive Informatics. 2002

[9] Mohamed-Amine Laalej, Veronique Delcroix, Sylvain Piechowiak, "A probabilistic-based algorithm to diagnose complex electronic systems", 15-th International Workshop on Principles of Diagnosis, Carcassone, France, June, 2004

[10] Jonah Z. Lavi, Michael Winokur, Amos Dagan, Ronie Rokach, "Multi level analysis of complex embedded computer systems", Israel Aircraft Industries, 1988

[11] Diane T Rover, Abdul Waheed, Matt W Mutka, Aleksandar M Bakic, "Software Tools for Complex Distributed Systems", IEEE 1998

[12] Michelangelo Interesse, "Test Manager: The Test Automation Component for the Maintenance of Large-Scale Systems", Proceedings of the international conference on software maintenance, IEEE, 2002

[13] YangYang Yu, Barry W. Johnson, Technical Report UVA-CSCS-FIT-001, A perspective on the state of research on fault injection techniques ,University of Virginia, 2002..

[14] Dave Rolince, Teradyne Inc. Boston, Applying virtual testing principles to digital test program development, Ma 02111, 2002.

[15] Changting Wang, Robert X. Gao, A virtual instrumentation system for integrated bearing condition monitoring, IEEE Transaction on Instrumentation and measurement, VOL. 49, NO. 2, APRIL 2000.

[16] Luigino Benetazzo, Matteo Bertocco, Franco Ferraris, Alessandro Ferrero, A web-based distributed virtual educational laboratory, IEEE Transaction on Instrumentation and measurement, VOL. 49, NO. 2, APRIL 2000.

[17] Alessandro Ferrero, Vincenzo Piuri, A simulation tool for virtual laboratory experiments in a www environment, IEEE Transaction on Instrumentation and measurement, VOL. 48, NO. 3, JUNE 1999

[18] Jahn Luke, Replacement strategy for aging avionics computers, IEEE AES Systems Magazine, March, 1999

[20] Edward Y. Chow, Integrating systems engineering software tools, IEEE AES Systems Magazine, November, 1998.

[21] Miron Abramovici, Digital systems and testable design, IEEE Press, 1995

[22] ARINC 625-1, Quality management process for test procedure generstion, ARINC Inc.

[23] ARINC 626-3, Standard ATLAS language for modular test, ARINC Inc.

[24] ARINC 627-2, Programmer's guide for SMART TM Systems using ARINC 626 ATLAS, ARINC Inc.

[25] IVI-3.1, Driver Architecture Specification, 2003 Edition, rev1.2, IVI Foundation, www.ivifoundation.org