Reuse of Structural Volume Test Methods for In-System Testing of Automotive ASICs

Cook, Alejandro; Ull, Dominik; Elm, Melanie; Wunderlich, Hans-Joachim; Randoll, H.; Döhren, S.

Proceedings of the 21st IEEE Asian Test Symposium (ATS'12) Niigata, Japan, 19-22 November 2012

doi: http://dx.doi.org/10.1109/ATS.2012.32

Abstract: The automotive industry has to deal with an increasing amount of electronics in today's vehicles. This paper describes the advantages of structural tests during in-field system test, reusing existing test data and on-chip structures. Demonstration is the embedded test of an ASIC within an automotive control unit, utilizing manufacturing scan-tests.

Preprint

General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by IEEE.[1]

[1] **IEEE COPYRIGHT NOTICE**
©2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Reuse of Structural Volume Test Methods for In-System Testing of Automotive ASICs

Alejandro Cook, Dominik Ull, Melanie Elm,
Hans-Joachim Wunderlich
University of Stuttgart
Institute of Computer Architecture and Computer Engineering
D-70569 Stuttgart, Germany
email: {cookao, ulldk, elmme, wu}@iti.uni-stuttgart.de
phone: 0711 / 685 88 389

Helmut Randoll, Stefan Döhren
Robert Bosch GmbH
DGS-EC/EPQ & DGS-EC/EHM
Wernerstrasse 1, D-70469 Stuttgart, Germany
email: {helmut.randoll, stefan.doehren}@de.bosch.com

***Abstract*—The automotive industry has to deal with an increasing amount of electronics in today's vehicles. This paper describes the advantages of structural tests during in-field system test, reusing existing test data and on-chip structures. Demonstration is the embedded test of an ASIC within an automotive control unit, utilizing manufacturing scan-tests.**

***Index Terms*—system test, scan-test, in-field, automotive, electronic control unit**

I. Introduction

The importance of electronics in the automotive industry has been steadily growing in the past few years and is still expected to grow in the future as more and more features — which were previously handled mechanically — are now realized in software [1]. It is estimated that, for the most part, innovations in the industry will be driven by advances in electronics, and specially, by the widespread use of electronic control units (ECU). For example, a high-end automotive model contains up to 100 interacting ECUs [2] spatially distributed across several sub-networks. This increasing number of electronic components poses several technical challenges to original equipment manufacturers (OEM) as they need to verify their integrity in the field under demanding environmental conditions. This task becomes critical in newer manufacturing technologies which are employed, for example, for driver assistance or infotainment, and are increasingly compute-intensive.

Traditional system tests in the automotive domain rely on a wide array of functional tests in order to assess the state of the system and reason about the observed erroneous behavior [3], [4]. Unfortunately, due to its low fault coverage, functional tests are not well suited to deal with complex semiconductor failures. Structural tests have long been recognized as a fundamental step in the chip manufacturing process. The execution of such tests in the field is a promising solution to the limitations of traditional functional testing. Every manufactured chip contains some non-functional resources devoted to improving structural test effectiveness and reducing test cost. These pieces of design-for-test (DfT) infrastructure may support a wide range of test techniques, from test data compression and compaction to be driven by external automated test equipment (ATE) [5], to fully autonomous Built-In Self-Test (BIST) for memories [6] and random logic [7]. These resources, however, are traditionally used only once after chip fabrication.

In this paper, the tradeoffs and benefits in the reuse of manufacturing test methods are examined with regard to their applicability for field test. The execution of such tests is accomplished by means of a programmable component in the ECU, i.e the microcontroller, that has to access the DfT infrastructure of an ASIC in the same board. For this purpose, a functional interface of the microcontroller has to transfer the related test data and control information to the target device-under-test (DUT). To the best of the authors' knowledge, this paper shows for the first time the successful realization of such a test strategy in an industrial ECU.

The paper continues with a brief overview of the failure analysis process in the automotive domain. Then, a selection of related work is presented, followed by a description of a generic embedded scan-test methodology. In Section V, a real case scenario is presented consisting of an automotive ECU for engine control. Section VI presents test coverage results while section VII provides some closing remarks.

II. Failure Analysis in the Automotive Domain

Fig. 1 shows the supply chain of a typical automotive application. The arrows in 1.a indicate how the test and diagnostic activities take place during traditional failure analysis when an error is detected in the field: after a battery of functional tests is executed, the failure information is gathered at the workshop by the OEM and one or more faulty ECUs are replaced. In order to optimize the production process and reduce subsequent service costs, the replaced ECUs are analyzed by the Tier 1 suppliers, who, after running another set of tests, relay the failure analysis to the Tier 2 suppliers. This process may continue for several other iterations between different companies before a faulty chip can be identified. At this point, the failure information has to be transferred back to the suppliers and the OEM in order to perform any appropriate

corrective action. This may be to discontinue the use of a certain batch of chips in the assembly of an ECU, or to replace potentially faulty ECUs in the workshop before the failure manifests itself. The traditional automotive semiconductor

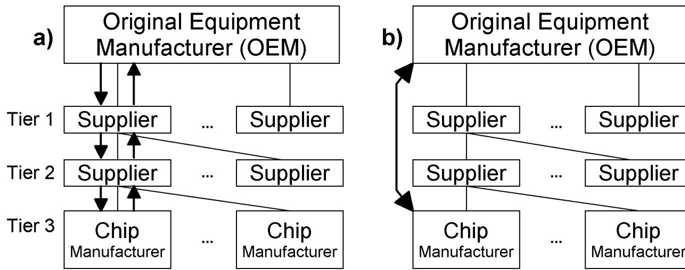

Fig. 1. Failure Analysis In The Automotive Domain

failure analysis flow has two main disadvantages: on the one hand, functional tests may have poor diagnostic capabilities at the ECU level, that is, in a complex system of several interacting ECUs the faulty ECU may not always be correctly identified and, consequently, the detailed semiconductor failure analysis of its electronic components does not reveal any defect in the chip. This also leads to increased service costs, as fault-free ECUs may be replaced in the workshop during repair.

On the other hand, when a faulty system is correctly identified, failure analysis has to be carried out by the OEM together with a potentially large number of companies that comprise the whole automotive supply chain. This may take a long period of time before the faulty chip is identified and the chip manufacturer can proceed with failure analysis at the semiconductor level. If the observed erroneous behavior is caused, for example, by a parametric defect in the chip manufacturing process, then every produced vehicle is likely to be affected by the same issue, which again translates into increased service and repair costs.

In order to overcome this limitation, in-field structural tests offer attractive capabilities: they typically achieve very high fault coverage, can isolate testing components and therefore directly identify a faulty chip. No fault-free system component is ever replaced. Moreover, as Fig. 1.b shows, the direct interaction between the OEM and the chip manufacturer can greatly accelerate the time required for semiconductor failure analysis.

The execution of structural tests in the field can be achieved by reusing the test procedures traditionally conducted during manufacturing test. However, such a strategy requires access to the relevant DfT infrastructure on-chip. Although BIST methods can greatly simplify this task since they need only an activation command to initiate the application of a given test, the related costs to make a design fully self-testable may be too high for some cost-sensitive applications such as the automotive domain [8]. Similarly, the adaption of ATE-based tests for autonomous structural test is not straightforward, as these tests require external equipment, which may not be available, as well as physical access to certain pins, which may not be accessible after system assembly.

III. Related Work

The access to on-chip DfT resources has received a lot of attention both by industry and academia in recent years due to the increasing amount of test infrastructure in modern integrated cicuits. The IEEE 1149.1 standard [9] was originally devised for board and interconnect test and is nowadays used for system-level test and debug. However, complex integrated circuits place additional requirements on the test procedure well beyond the capabilities of the IEEE 1149.1 standard. Such designs make use of several intellectual property (IP) blocks to achieve the desired funcionality on-chip. The IEEE 1500 standard [10] offers such an interface for core-based test and specifies the interaction between the logic cores in order to ease on-chip test integration. More recently, the IEEE P1687 standard proposal [11] describes the access to any on-chip instrumentation for test and debug. If such a wrapper is available on-chip, the underlying test infrastructure can be leveraged in the field if the chip is equipped with the appropriate control infrastructure [12], or the core wrappers can be accessed by an external component [11]. In either case, the test access mechanisms (TAM) to reach a given core, to apply test patterns or to retrieve the test responses are part of the chip design activities [13].

The use of a dedicated processing element for test purposes within a chip has been studied in the past [12], but requires substantial changes to the functional description of the design. In [12] the system bus is used to transport test control data from external memory to a micro-tester or to a TAM controller, respectively. These test data receivers forward the information to a set of logic cores with IEEE 1500-compliant wrappers. The micro-tester and TAM controller are both master and slave in the system interconnection bus. The slave interface is used by an on-chip general purpose processor to configure and schedule the test of every core, while the master interface is used during the test to fetch the necessary test input data, that is, a test program for the micro-tester or the raw test data in case of the TAM controller.

External interfaces can also be employed to access on-chip DfT infrastructure. They can be extended to support additional test procedures or to access one of the available test wrappers. While previous work has successfully employed such an interface in order to gain access to DfT infrastructure after chip manufacture, a piece of external equipment, which is not part of the target system under test, is always required.

IV. Embedded Scan-Test

The autonomous in-system execution of structural tests addresses one component at a time. This is achieved by leveraging the functionality of the enclosing system in order to mimic the manufacturing test environment and provide a suitable test harness, which can be used to isolate the device-under-test, apply a suitable test procedure and store

the obtained test responses.

A. Test access

Before any structural tests are applied, the state of the system must be switched from a fully operational mode to a hybrid mode where some components remain fully functional while others are driven into test mode and do not conform to the functional specification of the system. In this mode, the capabilities provided by the target DUT and (most likely) by the components comprising the test harness are no longer available.
In order to gain access to the DfT resources of the DUT, a fault-free unit or *test controller* may employ the DUT's available test interface. This goal can be achieved without a major cost penalty if an available functional interface in the test controller is exploited for the application of the test. For example, if the test controller is a programmable unit, i.e. a microcontroller, the test interface can be driven programmatically by means of its Input/Output subsystem. Fig. 2 compares the test setup for manufacturing test with that of in-system test. In Fig. 2.a the DfT infrastructure of the target DUT is accessed directly from the ATE, while in Fig. 2.b this is achieved with a test interface to the test controller in addition to any other functional interface in the system. The cost of the additional test interface on the board may have a great impact on the overall hardware cost of the system. In order to decrease the cost of the proposed in-field solution, the test interface and one of the functional interfaces of the DUT can be merged to a single interface between the programmable unit and the DUT. This is shown in Fig. 2.c. Such hardware modification reduces the overall system cost but requires some design changes in the DUT.
The fault-free operation of the test controller can be verified in the absence of other programmable devices by means of some other non-intrusive test techniques, like Software-Based Self-Test [14].

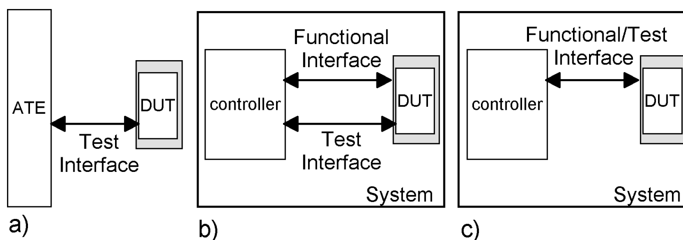

Fig. 2. (a) Manufacturing Scan-Test, (b) Embedded Scan-Test, (c) Embedded Scan-Test With Interface Reuse

B. Test application process

In the following, the general procedure for the in-system application of structural tests is explained.

- *Input test data transfer*
  The test data inputs are produced by the test controller. Depending on the nature of the target test, the test input data can range from a simple activation command (logic and memory BIST) to a fully-specified test pattern set. The test data input can be stored locally in the test controller, in a memory in the same board, so that the test controller has direct access to it, or can eventually be transmitted by means of a system bus like CAN or FlexRay.

- *DUT access*
  After the test controller has driven the DUT into test mode, a stream of data has to be downloaded into the chip in order to select a given structural test. For autonomous tests (BIST) an activation command is the only piece of information necessary for the test application. If scan tests are activated, however, a special operation mode is entered that enables to shift the scan data in and the test responses out. The same DUT may feature several scan configurations in order to support different ATE's with varying capabilities.

- *Test application (scan test only)*
  The scan patterns are applied and the test responses are evaluated. The test controller needs to shift one test pattern in, apply one capture clock cycle and shift the test responses out. If delay faults are targeted, two or more at-speed clock cycles must be applied before collecting the test outputs. The evaluation of the test may proceed during scan-out or, if test response compaction in time is used, after a given number of patterns.

- *Response test data transfer*
  Similar to the test input data, the outcome of the test needs to be stored for later retrieval and failure analysis. The storage can take place either in a local memory on the same board or externally in another memory inside the system. After this step, the DUT may be restarted and hence driven back to functional mode.

C. Design requirements for structural in-system test

ECUs can be safety-critical and are cost-sensitive systems, whose design is driven by reliability and hardware cost. Reliability issues within an in-field structural test application arise from the unspecified behavior the system may exhibit when one of its components is in test mode. For this reason, the design of an automotive structural test application must ensure that the execution of any test does not disrupt the deterministic behavior of the functional applications in the system. The amount of design effort to guarantee this goal ranges from moderate in workshop test to extremely high when the functional and test applications run concurrently. This imposes stringent requirements during the design of any online structural test solution [15].
The additional hardware cost required for the execution of an automotive structural test application is another important

factor to be considered. Because of high volume production, any chip or board area overhead, no matter how small, has great impact on the economics of the produced ECU. With these reliability and cost constraints in mind, an exemplary structural test solution for an automotive ECU is presented in the next section.

V. Case Study of an industrial ECU

The feasibility of an automotive structural test solution is demonstrated on an industrial ECU designed by Robert Bosch GmbH. The main objective of this case study is to analyze the tradeoffs and benefits of an in-field structural test solution under realistic cost and reliability constraints.
In order to assure passenger safety and system reliability, the unintended activation of the new structural test capabilities during normal operation has been carefully avoided. For this purpose, the sequence of commands required for test mode activation is always stored externally in a low-cost tester (laptop). Test access procedures and access keys are not allowed to be present in any part of the vehicle. Therefore, the unwitting activation of a component's test mode is completely prevented.
Hardware cost is kept to a minimum by extensive reuse of the system's functional capabilities and available interfaces to apply the test to the DUT. In the following sections the architecture of the test solution and its test procedure are explained in some detail.

A. ECU architecture

The ECU architecture is shown in Fig. 3, consisting of one main controller (32 bit, 150 MHz) and several ASICs (A-H). The main controller has an integrated CAN-Bus interface for communication with any other node in the CAN network. This network allows to upload temporary firmware into the controller's SRAM and to manage its execution, thus providing a bootstrapping mechanism for placing the dedicated test software.
The ASICs of the ECU share a multiplexed serial peripheral interface (SPI) bus with the main controller as bus master. They are addressed in multiple steps: the first step is done in hardware with chip-select signals (CS) to enable or disable up to 4 SPI slaves at once. Within such a group of ASICs, each slave features a fixed, unique address encoded in 2 bits. A slave shall only answer a SPI data frame from the master when it is enabled via chip-select and targeted by the address bits, which are the first bits of any request frame. In this way the slave can only know it is addressed after the first two bits of the frame were transferred.
An IEEE 1149.1 compliant test access port (TAP) is present on the chip for the management of test application processes. For in-system application of structural tests, access to this test interface is mandatory. In order to meet the cost constraints of the ECU design, an inexpensive interface has been developed as described in Section IV. The JTAG test interface is combined with the functional SPI interface, thus named

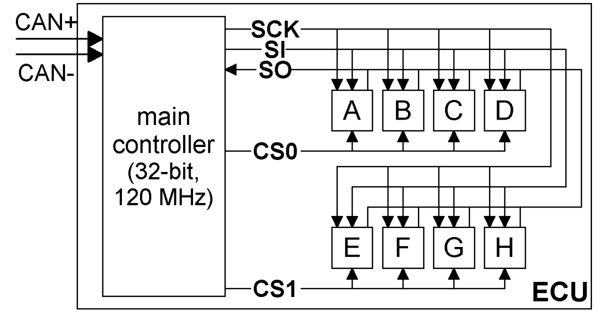

Fig. 3. Typical Automotive ECU Architecture

JTAG@SPI (patent [16]). The functional SPI port can be used as the first communication channel to activate the JTAG interface of ASICs. On the hardware side this requires an additional multiplexer to switch the incoming bus signals onto the TAP or the internal serial interface (Fig. 4). For the control of this multiplexer a new internal test mode is introduced. A specific sequence of SPI commands activates the internal test mode thus switching over to the JTAG protocol (a). Similarly, a JTAG instruction for returning to the functional mode exists as well (b). The intuitive signal mapping of SPI and JTAG protocols is shown in Table I.
A software-based driver on the main controller guarantees correct test data transfer to one ASIC while other SPI bus slaves still reside in functional mode. Therefore, the test communication is adapted to comply with the hardware-based (chip-selects) and software-based (address bits) multiplexing of the SPI protocol, so that test communication with the DUT is never misinterpreted as SPI data, and no functional bus node is ever inadvertently addressed. For this purpose, a software driver monitors the SPI bus. As soon as it observes the transmission of a SPI data frame, which is not addressed at the DUT, the chip-select signal is shortly driven to its inactive state. During this SPI frame reset action the test clock is halted to ensure correct test communication.

JTAG signals	SPI signals
Test Clock (TCK)	Serial Clock (SCK)
Test Mode Select (TMS)	Chip Select (CS)
Test Data Input (TDI)	Serial Input (SI)
Test Data Output (TDO)	Serial Output (SO)

TABLE I. Interface Reuse Signal Mapping(JTAG@SPI)

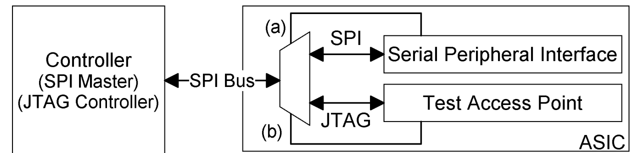

Fig. 4. JTAG@SPI: Multiplexing Test (JTAG) And Functional (SPI) Protocol

B. Manufacturing Scan-Test

The ASIC manufacturing test ensures high quality requirements in mass production. Numerous digital and analog,

functional and structural test applications are conducted at different operating conditions, assisted by complex automatic test equipment (ATE). The ATE is directly connected to the DUT, supplying it with the appropriate power sources and driving its JTAG test interface. This happens either before packaging with microprobes on the die pads, or on the package pins when the die is already encapsulated. Supported test methods include scan-tests for stuck-at and transition delay faults as well as programmable BIST methods for internal memory blocks. The manufacturing scan-tests are applied in a test-per-scan fashion as shown in Fig. 5. The ATE sets the test input signals {CS0, SCK, SI, RST} and evaluates the test outputs {SO, PWMEn, IRQ} on every clock pulse on the ECK pin, which is the test clock.

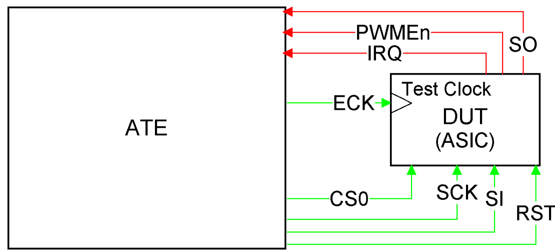

Fig. 5. ASIC Manufacturing Scan-Test Setup

C. Implementation Setup

The employed test setup is shown in Fig. 6. For clarity further ASICs are not shown. The setup makes use of the KWP 2000 protocol over CAN [17], [18] to exchange data between the tester and the ECUs. A laptop is employed as the client device, although any other low cost tester may be used for this purpose. The laptop serves as storage device for test mode activation and access mechanisms (software) as well as the scan data applied later on. The data is transferred to the ECU by means of the ASC@CAN interface, transmitting up to 255 Bytes per asynchronous data frame over the CAN-Bus. This communication channel is used to upload the main controller's test software, to execute it, and to transmit test data and test responses between the controller and laptop.

D. Test Application Process

- *Input test data transfer*: At first, the ECU's main controller has to be loaded with the test firmware and scan test data, which is provided by the tester via the CAN-Bus interface. Due to its cost-driven design, the ECU does not hold enough volatile memory to buffer all input test data at once. Therefore, the test application is partitioned into sessions which are executed by several request messages. The provided test data from ASIC manufacturing contains approximately 150,000 test vectors and is able to detect 97% of all stuck-at faults. The test vectors contain the fixed values for the input signals as well as the expected results for the output signals. The latter, however, may

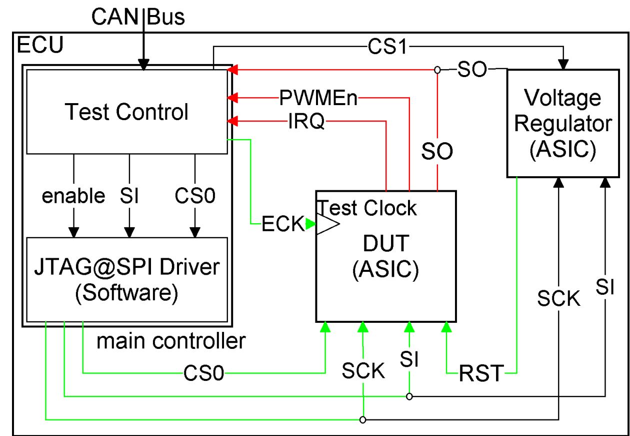

Fig. 6. Embedded Scan Test Implementation

contain *don't care* values (Xs). During manufacturing test, each vector contains an additional timing directive. Since this in-field implementation cannot cope with the prescribed clock speeds, this timing information is ignored.

A binary encoded vector requires 1 bit for each of the 4 test inputs and 2 bits for every of 3 outputs, which sum up to 10 bit in total. In contrary to the inputs, the outputs allocate 2 bits in order to encode the additional *don't care* state. To enable byte-wise decoding, the total vector size is rounded up to 16 bits. A perl script encodes an ASCII data set provided by a commercial ATPG tool into a different format suitable for easy binary parsing. This information is then dumped into a template script file. The resulting script can be executed on the laptop to transfer the data over the CAN network.

- *DUT access*: the ASIC is brought into test mode by executing appropriate SPI commands received from the main controller. For scan tests, the test interface activation by itself is not sufficient. Further JTAG instructions are used to prepare the DUT by setting other internal modes and clock selection registers.

- *Test application*: Multiple request messages, each containing 125 sequential test vectors, are transmitted over the CAN interface to the main controller, which in turn sets the test inputs accordingly before pulsing the test clock signal and comparing the test outputs with the expected values. The only test input not directly controllable by the main controller is the DUT's reset signal RST. This obstacle can be overcome by utilizing the voltage regulator ASIC, which can be forced to set the RST line to its inactive state. When a test vector with inactive RST state appears, a toggle in the RST line can be triggered with a special SPI command sent to the regulator ASIC. During the transfer and execution of this command, the test clock is halted. After a short time span of typically several microseconds, the RST line

falls back to its active state. During this time period the current test vector is applied by pulsing the test clock. The test application process resumes after the RST line returns to its inactive state.

- *Response test data transfer*: The main controller keeps track of the test result by monitoring the test outputs. If a test output does not match its expected value, the vector index and the states of the output signals are sent to the laptop in a response message. If no error occurs, the request message is simply acknowledged. Therefore, the presented embedded scan-test solution achieves the same amount of diagnostic information as the ATE-driven manufacturing test.

VI. Results

In the presented case study a scan test pattern set for an ASIC has been applied in the field. The pattern test set, although generated for manufacturing test, is applied without removing the target ASIC from the assembled system. The test application process has been controlled with a portable computer connected to a functional bus (CAN). The architecture of the presented test application is fully in-line with the traditional functional test procedures applied in the automotive domain.

Table II shows the test application performance. As the same test data set is used, the fault coverage is equal for both, manufacturing test and in-field test. Due to implementation and system constraints the clock frequency for in-field test is rather slow and cannot reach that of its manufacturing test counterpart, which is driven at 16 to 20 Mbit/s. The main reason for this is the indirect control of the RST signal, which gets locked in its inactive state for several microseconds. Another reason is the software overhead in the main controller, introduced by the JTAG@SPI driver. These implementation-specific timing issues hinder the application of delay tests from the manufacturing test specification. In order to prevent this issue, additional constraints need to be considered during the design process of the ECU. If this is the case, the proposed method could be applied for any fault model. The complete test time of the presented in-field test is also impaired by the interface between the external portable computer and the ECU's main controller, which is limited to 4 Mbit/s.

The implemented method also provides valuable information for fault diagnosis at gate level, as the failing scan cells are identified and can be used for further analysis to narrow down the fault location.

application	test time	# of test patterns	fault coverage
manufacturing	0.03 s	150,000	97% stuck-at
embedded	132 s	150,000	97% stuck-at

TABLE II. Comparison of test applications

VII. Conclusion

Current automotive systems face the challenging task of testing electronic components in the field. The presented test solution shows that structural test techniques originally deviced for manufacturing test can be applied in the workshop. In the presented solution an existing test pattern set with 97% fault coverage is applied with the help of an inexpensive portable computer. The test outcome can be used to correctly identify a faulty component in the system and to speed up semiconductor failure analysis in the automotive industry, significantly.

References

[1] S. Furst, "Challenges in the design of automotive software," in *Proc. Design, Automation Test in Europe Conference and Exhibition (DATE'10)*, mar. 2010, pp. 256–258.

[2] S. Kim, E. Lee, M. Choi, H. Jeong, and S. Seo, "Design optimization of vehicle control networks," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 7, pp. 3002 –3016, sept. 2011.

[3] C. Kihoon, L. Jianhui, K. Pattipati, S. Namburu, L. Q., and S. Chigusa, "Data reduction techniques for intelligent fault diagnosis in automotive systems," in *Proc. IEEE Autotestcon (AUTOTESTCON'06)*, sept. 2006, pp. 66–72.

[4] Y. Zhang, G. Gantt, M. Rychlinski, R. Edwards, J. Correia, and C. Wolf, "Connected vehicle diagnostics and prognostics, concept, and initial practice," *IEEE Trans. on Reliability*, vol. 58, no. 2, pp. 286–294, jun. 2009.

[5] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 776–792, may 2004.

[6] M. Bushnell and V. Agrawal, *Essentials Of Electronic Testing for Digital, Memory, And Mixed-Signal VLSI Circuits*. Springer science+business media, 2000.

[7] H.-J. Wunderlich, "BIST for systems-on-a-chip," *Integration, the VLSI Journal*, vol. 26, no. 1-2, pp. 55–78, 1998.

[8] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, "Logic BIST for large industrial designs: Real issues and case studies," in *Proc. International Test Conference (ITC'99)*, 1999, pp. 358–367.

[9] "IEEE standard test access port and boundary-scan architecture," *IEEE Standard 1149.1-1990*, pp. 0–1, 1990.

[10] "IEEE standard testability method for embedded core-based integrated circuits," *IEEE Standard 1500-2005*, pp. 1–117, 2005.

[11] J. Rearick, B. Eklow, K. Posse, A. Crouch, and B. Bennetts, "IJTAG (internal JTAG): A step toward a DFT standard," in *Proc. IEEE International Test Conference (ITC'05.)*, nov. 2005, pp. 8–815.

[12] K.-J. Lee, C.-Y. Chu, and Y.-T. Hong, "An embedded processor based SoC test platform," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'05)*, may 2005, pp. 2983–2986 Vol. 3.

[13] E. Larsson, K. Arvidsson, H. Fujiwara, and Z. Peng, "Efficient test solutions for core-based designs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 758–775, may 2004.

[14] M. Psarakis, D. Gizopoulos, E. Sanchez, and M. Reorda, "Microprocessor software-based self-testing," *IEEE Trans. on Design Test of Computers*, vol. 27, no. 3, pp. 4–19, may-jun. 2010.

[15] A. Dutta, M. Shah, G. Swathi, and R. Parekhji, "Design techniques and tradeoffs in implementing non-destructive field test using logic BIST self-test," in *Proc. IEEE International On-Line Testing Symposium (IOLTS'09)*, jun. 2009, pp. 237–242.

[16] P. Poinstingl, H. Randoll, C. Knaupp, T. Braun, T. Wieja, S. Wirth, S. Doehren, and R. Kraemer, "Offenlegungsschrift DE 10 2010 002 460 A1 2011.09.01: Verfahren zum Testen eines integrierten Schaltkreises."

[17] *ISO 14230-3: Road vehicles – Diagnostic systems – Keyword Protocol 2000 – Part 3: Application layer*, Std.

[18] *ISO 15765-3: Road vehicles – Diagnostics on Controller Area Networks (CAN) – Part 3: Implementation of unified diagnostic services (UDS on CAN)*, Std.