

# Adaptive Test System to Improve PCB Testing in the Automotive Industry

Steffen Ostendorff, Joerg Sachsse, Heinz-Dietrich Wuttke and Jorge Meza Escobar  
Ilmenau University of Technology

## ABSTRACT

This paper presents an adaptive test approach to improve the structural testing of printed circuit boards (PCB) found in electronic automotive components. The approach makes use of FPGAs available on the PCBs, and its applicability is supported by the global trend taking place in the automotive industry of replacing ASICs with programmable devices such as FPGAs.

For structural testing of PCBs, Boundary Scan (BScan) is mostly used. However, BScan has the disadvantage of being a static test method due to the slow execution speed reducing the fault coverage concerning dynamic faults.

FPGAs support BScan as well, but they also offer a vast number of programmable resources. These resources can be configured for testing purposes.

Our approach is to speed-up the testing process during the PCB manufacturing by moving data intensive processing from the external software side (Test-PC) to the programmable hardware side on-board (FPGA), reducing the data transfer over the slow JTAG interface. The resources of the FPGA are used to implement a complete FPGA based test system capable of generating, applying, and analyzing test patterns at high speeds.

In this way, we reduce execution time and increase the fault coverage for high speed circuits such as high speed interfaces and memories.

**CITATION:** Ostendorff, S., Sachsse, J., Wuttke, H. and Meza Escobar, J., "Adaptive Test System to Improve PCB Testing in the Automotive Industry," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-1230.

## INTRODUCTION

The automotive industry has seen a significant growth in the application of electronics. The percentage of mechanical parts in the vehicle added value is constantly decreasing, while the value of a today's fully equipped car is mostly determined by its electric, electronic, and mechatronic systems. For this reason, these systems are a key area of development in the automotive industry.

In recent years, the application of wireless and information technology products such as rear seat entertainment, digital satellite radio and navigation systems have presented the potential for significant new growth opportunities and product differentiation in the automotive electronics industry. These new features often require fast processing power and large memory to handle video data streaming, image recognition, or other signal processing services.

The hardware and software complexity of automotive components is increasing in order to implement customer requirements. As a result, the added software functionality is causing an exponential growth rate in the automotive

embedded software space. Functions that cannot be implemented in software are often implemented in custom application specific integrated circuit (ASIC) solutions. [1]

The ASIC solution normally requires a significant lead-time and nonrecurring engineering (NRE) charges to bring the design to fruition. This is in conflict with industry trends. A solution to this problem, gaining acceptance in the automotive industry, is the use of Programmable Logic Devices (PLDs) and Field Programmable Gate Arrays (FPGAs). Especially, FPGAs offer increased device functionality and shorter time-to-market attributes [1, 2].

Further advantages of the FPGA solution compared to ASICs are the facts that less time is needed to be invested in planning the design process and architecting a verification methodology. Moreover, an FPGA solution does not require a significant investment in design automation tools beyond what is supplied by the FPGA vendors, only moderate investment is needed to train FPGA-designers (generally SW design techniques are used), and FPGA development and integration time is less than that of an ASIC.

In summary, FPGA usage is growing rapidly because it satisfies the automotive industry's demands for faster processing speed, higher logic density, shorter time-to-market cycle, and re-configurability [1]. Active-safety systems and infotainment systems are the most popular areas for FPGA applications. These applications need to process large amounts of streaming input data and provide responses in real time [3].

The influence on electronics on automobiles became possible only because of the low cost and high reliability of electronic components. In order to keep the production costs low and the reliability high, it is required to detect defects as soon as possible. To address this problem at the printed circuit board (PCB) level, different test techniques have been developed and put into practice [4, 5, 6, 7, 8, 9].

So far, for structural testing of ASIC-based PCBs Boundary Scan (BScan) was used most of the time. However, BScan has the disadvantage of being a static test method due to the slow execution speed. This causes the fault coverage to be low concerning dynamic faults that may occur especially on modern memories and high-speed interfaces or caused by interference with other components in the car.

FPGAs support BScan as well, but they also offer a vast number of programmable resources. These resources can be configured during manufacturing or in the field for testing purposes.

In the next section the motivation of our approach is presented followed by the concept, giving details about the goals, the embedded test system architecture, and the automated generation flow based on DUT (Device Under Test)-models. In section 'Example', a detailed report of the first experimental results using this automatic generation flow is given, including a description of the test setup and the achieved speedups for different DUT-tests such as e.g. an LCD display test. The paper closes up with conclusions and the plans for future work.

## MOTIVATION

Despite the continuous growth of test features and test algorithms the functionality of electronic systems and their components grew disproportionately high, facing the test engineer with dedicated problems as long test times or untested features, both causing possible higher costs either at the test procedures or at services due to a higher number of system failures.

Nowadays an increasing amount of electronic systems in the automotive industry include FPGAs that can easily be reconfigured during the test stage to implement some special test functionality. Our concept is driven by the need to improve the current process of board level testing by using only on board available resources. This allows the concept to be applicable to basically all currently available PCBs containing at least one FPGA. The goal is to improve the test process without the need to change or add any hardware components on the circuit board and to offer the full benefit

just by reconfiguration of the involved configurable hardware and software components.

## CONCEPT

Our approach is to accelerate the test execution by moving data intensive processing tasks from the external software side (on a Test-PC) to the reconfigurable hardware side on board, reducing the data transfer over the slow JTAG interface. In this approach on-board resources (FPGA) are used to implement parts of the test data generation and analysis directly in HW instead of transferring the test data and results step by step via the JTAG interface.

## Goal

The main goal is the implementation of the test functionality (test system) according to the PCB under test. This means the FPGA based test system adapts to the board to be tested, its structure and components. Another important goal is the application of the test system to a wide range of PCBs independent of FPGA-vendors and families. Last but not least our goal is to simplify the integration of the generated test system into commercial BScan test tools by using the common JTAG interface as the communication link to the test tools on the PC.

To make the test system highly flexible and to allow even complex and elaborate test algorithms to run directly on the FPGA, we aim to implement a special soft-processor within our test system.

In order to relieve the test engineer from dealing with implementation details an automated generation flow is realized producing all required hardware (HW) as well as software code (SW).

## Approach

In our approach FPGA resources are used to implement parts of the test functionality directly in hardware.

To manage the complexity of creating such a test system, it is based on a layer concept presented in [10, 11]. This concept divides the test process into 5 layers used to split the test process in independent and stack organized functions. [Figure 1](#) illustrates the layer stack. Besides, the test system requires two interfaces to communicate with the Test-PC and the board device to be tested (DUT), as well.

The interface 1 simply adapts the FPGA test system to the inputs and outputs of the device under test (DUT), while interface 2 is the usual interface to the Test-PC via JTAG. The interface to the Test-PC is done by establishing an adapted chain of BScan cells inside the FPGA's logic. In this way only the test data related to the DUT has to be transmitted from the PC to the FPGA.

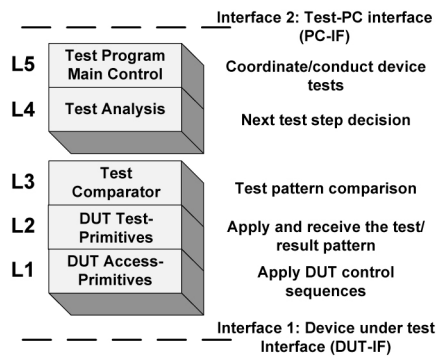


Figure 1. Layer stack [11]

The layers use the functions supplied by the lower layers and provide their functionality to the upper layers.

The DUT access-primitives layer (layer 1) contains basic functions to access the DUT. Using layer 1 eliminates the need to emulate the pin behavior for certain control cycles on a low level via BScan shifts.

The DUT Test-Primitives layer (layer 2) is dependent on the test algorithms used. This layer consists of a set of test primitives like shift-, count-, rotate- or other special operations. Implementing these primitives directly in hardware improves the test speed significantly.

The layer 3 collects and compares the response of the test primitives and gives the result (failed/ passed) to the test analysis layer (layer 4).

Layer 4 decides the next test step, depending on the results, provided by layer 3. Layer 5 coordinates the main procedure of the test algorithm.

The functional layers (layer 1 to 5) can be implemented in hardware or as software functions at the Test-PC outside the PCB (, if the test software supports programming features). Depending on the resources, available at the FPGA, a specialized soft-processor can also be implemented on the FPGA, running some or all of the functional layers in software directly at the test system.

The main advantages of the specialized soft-processor, compared to a standard processor, are:

- FPGA-type independent processor architecture
- Unique JTAG compatible communication interface independent of the FPGA-type.

Using the layer model makes it easier and more transparent to implement the test system. Depending on the available resources and needed performance the layers can be implemented hardwired in FPGA-logic or as software functions either running on the Test-PC or on the soft-processor.

Due to the test system flexibility, achieved with the layer structure, the soft-processor is e.g. capable of executing layer 1 and layer 2 functions as software routines while layers 3, 4 and 5 still run at the Test-PC.

The layers to be implemented in hardware are transformed to their hardware descriptions and the layers

realized in software are either transformed to the soft-processor code or to software code to be executed at the Test-PC.

## Device Model

To make this layer concept applicable and easy to use for test engineers, a device model supplied by a test tool vendor for every DUT (DUT-M) is needed. Such a DUT-M describes the device interface, its functionality, the test algorithms as well as parameters. As can be seen from Figure 2, it is divided into four main sections: the interface section with data-, address- and control-bus information, the timing section with the DUT timing information, the states section where the DUT control states are defined, and the test algorithm section where the procedures and the main program are located.

```

MODULE SRAM()
-----
INTERFACE(
  DATA (INOUT[16] io;)
  ADDRESS(IN[18] a;)
  CONTROL(IN CEn;IN WEn;IN OEn;)
)
-----
TIMING ()
-----
STATE idle()
BEGIN
  CEn := 0;
  WEn := 1;
  OEn := 1;
END
STATE write()
BEGIN
  cEn := 0;
  wEn := 0;
  oEn := 1;
END
...
-----
PROCEDURE l2_apply_pattern(IN[2] wr_rd, ...)
BEGIN
  ...
END
PROGRAM l5_sram_test()
VAR l4_err;
BEGIN
  ...
END

```

Figure 2. Example of a DUT model (skeleton)

Based on this concepts, the test system will be automatically generated, including all required hardware descriptions (e.g., VHDL), software code, and the test patterns.

## Architecture

An example of the test system architecture is depicted in Figure 3. It shows a PCB that can be connected via JTAG to external test equipment. On the PCB there are different DUTs, like DRAM, Flash or Display devices, and an FPGA directly connected to them.

In this example the embedded test system is implemented within the FPGA as processor based architecture. The