System Tests for Reconfigurable Signal Processing Systems

Jesko G. Lamm, Adolfo Espinoza, Anna K. Berg
Bernafon AG, Morgenstrasse 131, 3018 Bern, Switzerland
Email: {jla, aes, abe}@bernafon.ch

***Abstract*—Reconfigurable signal processing systems enable the creation of multiple products based on one common platform. Designing test patterns for system tests of such products is challenging due to the reconfiguration possibilities in signal processing. Test pattern injection and test data extraction may be difficult due to non-idealties in analog parts of the signal processing. We propose a procedure for balancing black-box and white-box testing during the development of reconfigurable signal processing systems. A case report studies an exemplary application of the procedure in the domain of digital hearing instruments.**

I. INTRODUCTION

Signal processing is the key to modern multimedia applications, ranging from hearing instruments to portable internet terminals. Low power consumption and small size are frequent requirements for signal processing devices, no matter whether these operate in the audio, video or radio wave processing domain. Therefore, signal processing devices usually need highly integrated circuits. For example, so-called Systems-on-Chip have to provide complex signal processing functions on a single piece of silicon.

In signal processing, reconfigurable hardware can provide the compromise between flexibility and performance [1] in a domain where efficient systems must be designed for change. We define: a system is *reconfigurable* if its functionality can be modified after manufacturing its hardware. Typical reconfigurable devices contain programmable units like microprocessors, Fast Programmable Gate Arrays (FPGAs) or signal processing cores with parameters in programmable memory.

High integration demands lead to systems containing a mixture of reconfigurable and hard-wired blocks, as it has been documented for the case of Systems-on-Chip in [2]. Reconfigurable systems are thus more and more heterogeneous in the sense that they usually contain both dedicated software and hardware. In general, software and hardware parts cannot be described separately, because they are often on the same physical component and potentially even mapped [3] to the same entity of an abstract system description.

Testing such a heterogeneous system is thus neither a matter of software test methodologies nor a traditional hardware verification task. Probably this is the reason why we have not yet seen an established methodology for testing reconfigurable signal processing devices. We would like to make a proposal for filling this gap, by showing how we started the development of our best practices. We limit our scope to system tests of reconfigurable signal processing systems. Therefore we first have to describe what we mean by system tests. Then we point out the difficulties in system tests of reconfigurable signal processing systems and present how we propose to address those. Finally, we show a little case report about the proposed way of establishing system tests.

II. SYSTEM TESTS

A. Definition

A *system test* is a "test of a complete, integrated system to evaluate the system's compliance with its specified requirements" (glossary definition from [4]). We here look at system tests in the development of a new product and not at product manufacturing tests (e.g. [5]). For this paper, it does not matter whether the specifications to test against are expressed in plain text (e.g. [6]) or in a model [7], [8], [9].

System tests verify if the system works as a whole after integrating its parts. This is particularly important regarding the verification of *emergent behavior*, which is the behavior arising when the parts of a whole interact [10]. Verifying emergent behavior of a system has two aspects:

- The desired emergent behavior must be tested for compliance with its specification.
- Testing should ensure that the interaction of the system's parts does not produce undesired emergent behavior.

B. Black-box versus white-box testing

Black-box testing and white-box testing are typical test strategies in software testing (e.g. [11]). Here however, the black-box and white-box view are used in the domain of systems thinking, like in [12]. Consequently, we consider black-box testing and white-box testing as possible test strategies for system tests, based on [13]. We define that *black-box tests* only use what is outside the system boundary, where in contrast, *white-box tests* exploit the system's inner structure.

Concepts of the system's inner structure are validated in system tests. A white-box test depends on the system's inner structure and is therefore in general based on assumptions that are unverified before system testing. As a result, it would be circular reasoning to use a white-box approach in system testing. Section II-C illustrates this with an example.

From a theoretical point of view, white-box tests are thus disqualified from system testing. We therefore believe that an ideal system test should exclusively use black-box testing. However, choosing a black-box approach can lead to difficulties in test design. This will be discussed in the following.

C. Difficulties with reconfigurable signal processing systems

1) Test coverage: One of the reasons for building reconfigurable systems is the possibility to share an expensive hardware design across multiple products, using it as a *hardware platform* [3]. The full potential of a reconfigurable hardware platform is usually not exploited in the life cycle that the hardware is designed for. It occurs that a new invention can be realized by reconfiguring an existing hardware platform.

For system testing this means: it is in general not feasible or at least not affordable to forsee all needed test cases during the development of a hardware platform. As a consequence, the full coverage in testing cannot be guaranteed, because the parameter space of systems built upon a platform is not clear when developing the platform.

2) Test access: Test pattern generation schemes implicitly assume that test patterns can be injected without loss into the system under test. Signal processing systems are however often mixed-signal systems in which the input and output path are analog and therefore not lossless. For example, noise and nonlinearities have to be taken into account in test signal design (e.g. [14]). Note that the analog input and output path are usually within the system boundaries and thus have to be used in system testing according to the black-box paradigm from section II-B.

One can argue that the black-box paradigm is a rather theoretical construction that introduces unneccesssary formalism in test design. When there is no black-box paradigm, one can solve the problem of injecting test patterns and extracting the corresponding output. One solution is a *Test Access Mechanism* [15], which is a design-for-test concept: the system under test is designed with a built-in communication protocol that can stream data into signal processing components and access the resulting output vectors.

Tests based on a Test Access Mechanism can faciliate the injection and extraction of data; however at the price of missing defects in components not accessed by the Test Access Mechanism. Figure 1 shows an example: A fictitious system under test processes an input signal ("A") and ouputs the processed signal ("B"). The processing requires a signal processor, which accesses the input signal via an input driver and ouputs the processing result via an output driver. Obviously, the signal processor implements the key functionality of the system under test, which therefore offers access to the signal processor via a Test Access Mechanism. A typical use of the Test Access Mechanism would be the extraction of data ("C") from the signal processor's ouput, while a test pattern is injected into the signal processor by some mechanism not shown in figure 1. We assume now that the output driver in figure 1 has a defect that is only observable for certain processing results of the signal processor. Assume that there is a test with sufficient parameter space coverage to detect the defect. If this test was based on the Test Access Mechanism, then it would fail to discover the defect, because the defective output driver is bypassed in using the Test Access Mechanism.

The given example illustrates the earlier notion that a system test based on a white-box approach produces circular reasoning: The Test Access Mechanism exploits the systems inner structure by reading out system parameters at their source. This makes the implicit assumption that the system does actually treat the observed system parameters as specified. In the shown example, this implicit assumption is wrong, because system parameters are wrongly interpreted by the output driver due to a defect. As a consequence, the tests based on the Test Access Mechanism will pass, although the correct system test result would be "failed".

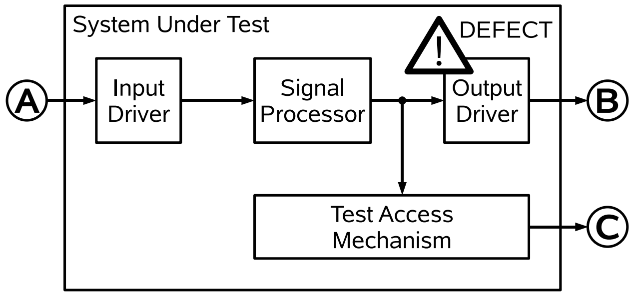

Fig. 1. Fictitious signal processing system.

III. Proposed solution

A. Method

1) Test strategy: We have pointed out that black-box tests of signal processing systems introduce problems regarding test access and test coverage, whereas a white-box approach is not ideally suited for system testing. This means that system testing has to balance the feasibility risk of black-box testing against the concept risk of white-box testing. As the ideal system test would be a black-box test with maximum test coverage, we define here that the *test maturity* of a system test is the degree to which it fulfills black-box and test coverage criteria.

We propose a pragmatic approach in which both black-box testing and white-box testing are regarded as possible means of system testing, but in which the goal should be maximized test maturity.

2) Test design: The starting point for test design can be a white-box procedure in which an early prototype of tests is established using a Test Access Mechanism. This eliminates test access problems in a first iteration, allowing the test designer to focus on the generation of test patterns that achieve the desired test coverage.

Once the problem of test pattern creation has been studied in the simplified context of a white-box test, the test access problems of black-box testing can be addressed; this means: the test developer tries to optimize test patterns for injection via a non-ideal signal input and for data extraction via a non-ideal signal output. This allows increasing test maturity during a step-by-step replacement of white-box activities by black-box testing. During the whole procedure, the original white-box test stays a fall-back solution for cases in which the difficulties of black-box test design turn out to be hard-to-solve problems.

This is the final (submitted) version, accepted for "TuZ09: 21. Workshop für Testmethoden und Zuverlässigkeit von Schaltungen und Systemen" (http://www.informatik.uni-bremen.de/tuz09/)

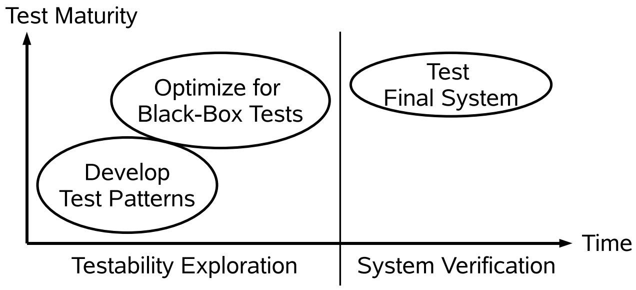

Fig. 2. Proposed Procedure.

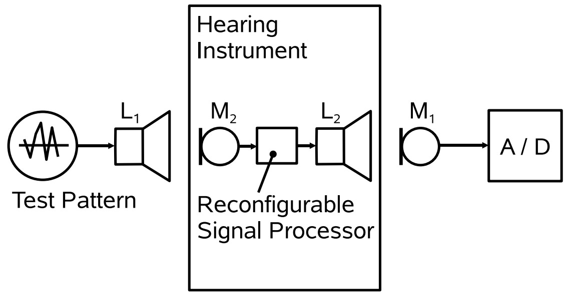

Fig. 3. Setup for the system test of a hearing instrument.

B. Procedure

Figure 2 illustrates the proposed procedure, in which test maturity continuously increases over development time:

- *Develop Test Patterns:* At an early stage of development, test patterns are created without accounting for the black-box paradigm and therefore at a low level of test maturity.
- *Optimize for black-box tests:* Try to increase test maturity by enabling black-box testing.
- *Test final system:* Once the desired test maturity has been reached or the project schedule requires the initiation of the final system verification, system tests can use test patterns and injection / extraction schemes in the most mature version available by that point of time.

We believe that the development process should allow test initiation at an early stage of development and in close interaction with specification and design activities. This is for example proposed as a process model in the "W-Model" [16], a variant of the well-known "V model" (e.g. [6], [16]).

It has been argued (p. 261 of [6]) that tests cannot be run during requirements development. However, we believe that early prototyping can enable test execution at very early stages of specification, e.g. as soon as a stable vision of the system under development exists. Of course, the basis for system testing is vague as long as system requirements are incomplete. However, in a domain in which platform-based design is the state of the practice, testability can be explored with existing candidate platforms for the implementation. Here, throwaway prototypes [6] of embedded software or of other sub-systems can replace missing parts in order to allow an early preview of the system built upon a given platform.

As pointed out earlier, test pattern injection and output data extraction are key problems in system testing of signal processing systems. Therefore, early *testability explorations* on candidate platforms can start building up the technology knowledge needed to create test patterns according to the following criteria:

- The test patterns are optimized for minimizing loss during their injection into the system.
- Even with the given loss in injecting the test pattern, the desired test coverage is obtained.
- The test patterns are designed for evoking a system response that is easy to measure or acquire during the output data extraction.

Figure 2 shows how testability explorations map on the time axis. These explorations are an iterative process, resulting in multiple loops of test design and test execution. To reduce the number of loops in such test development, one can use existing procedures (e.g. [2]) from the domain of system development, which can be easily transferred to test development. For example, the suitability of different test pattern creation schemes could be explored and quantified for each of the signal inputs of a given platform, already during platform development. Then, the possible coverage of the parameter space could be modeled as a function of the test pattern creation scheme and its parameters.

IV. Case Report

A. Overview

We studied the proposed procedure in the domain of digital hearing instruments during the development of a noise reduction function on a reconfigurable signal processing platform. A noise reduction function should reduce the level of disturbing environmental noise in an audio signal. In the use case we consider here, a hearing instrument picks up the audio signal it processes via a microphone. This means that there is only one signal input to consider regarding the injection of test patterns.

Figure 3 illustrates a typical setup for system tests on a hearing instrument: A Loudspeaker (L_1) allows the injection of test patterns into the hearing instrument's microphone (M_2). Test data extraction is done via a measurement microphone (M_1), which picks up the signal emitted by the hearing instrument's output transducer (L_2). An analog-to-digital converter (A/D) converts the signal to a digital data stream for further processing. In the given case both test patterns as well as extracted data are audio signals.

B. Testability Exploration

1) Testability constraints: Earlier, we proposed testability explorations during platform design, resulting in models characterizing different test pattern creation schemes regarding their suitability for generating input stimuli. In our case, the described models did not exist, but we had access to similar information:

- A previous project had worked out a first concept for generating test patterns for system tests related to noise reduction and had investigated the related difficulties in test pattern injection and test data extraction.

- Previous research [17] had shown solutions for the test pattern insertion problems as well as for test data extraction, however not with the given platform.

The detailed problems and solutions related to test pattern generation are only mentioned briefly here, because they have been discussed in [17]. The most important aspects were:

- The test patterns had to provide sufficiently constant energy over the frequency range of interest in order to achieve a suitable signal-to-noise ratio.
- The signals needed to have limited peak factors because of non-linearities in measurement equipment and in the device under test.

Whereas the literature had provided procedures for synthesizing such signals with a broad bandwidth and a stationary envelope, a procedure targeted at strongly bandlimited signals with a modulated envelope had to be developed for [17].

2) Test pattern development: In test pattern development, we used special development tools as early prototypes of the final software controlling the noise reduction system under test. As a starting point for our test, we observed the noise reduction via a Test Access Mechanism:

- We inserted very simple test patterns directly into the noise reduction block via an on-chip signal generator.
- We used a Test Access Mechanism to read out the noise reduction's reaction (i.e. to monitor its attenuation parameters).

The described setup allowed us to verify the time the noise reduction needed to reach steady-state. This validated our test timing at a very early stage of test development.

3) Optimization for black-box testing: As a next step, we still used the Test Access Mechanism for test data extraction, but now we stimulated the noise reduction block by injecting a first version of test patterns into the hearing instrument microphone via a loudspeaker (L_1) according to figure 3. This setup allowed us a first selection of the test patterns to be used and the elimination of certain candidates that were not suited for the given way of injection.

Now we could perform a black-box test, which no longer used the Test Access Mechanism. The test setup was the one described in [17]. This turned out to be the final setup for our system test. We used it to explore the possible coverage of the parameter space, which became sufficient after certain optimizations of the test patterns.

C. System verification

Once the noise reduction function had been integrated into the system under test, we verified the system as whole to find out if the interaction of the noise reduction with other parts had created the desired emergent behavior. We used the black-box test that had evolved according to IV-B. In this test, the system behaved as specified. However this was not the result of developing a defect-free system at once. System verification was successful at once, because the different stages of testability exploration had revealed design and implementation errors already in an early phase of test development. As a consequence, these errors had been eliminated by the time of the first system verification.

V. Discussion and Conclusion

The difficulties of System tests in the development of signal processing systems have been discussed. We have proposed a procedure with the goal of mature system testing for reconfigurable signal processing systems. In this procedure, system tests evolve in different steps, starting with the exploration of test patterns, and ideally the modeling of testability contraints.

We studied this procedure in an exemplary case, in which the results from prior work were our model of testability constraints. The early start of test exploration and the existing test procedures helped us in finding design and implementation errors at an early stage in the project. As a consequence, the first system verification passed at once.

It was high effort to establish a system test according to the black-box paradigm. Therefore we believe that a pragmatic system test approach will include black-box testing as well as white-box testing. Methods for balancing black-box and white-box testing are needed.

References

[1] R. Tessier and W. Burleson, "Reconfigurable computing for digital signal processing: A survey," *Journal of VLSI Signal Processing Systems*, vol. 28, no. 1-2, pp. 7–27, 2001.

[2] H. Blume, H. T. Feldkaemper, and T. G. Noll, "Model-based exploration of the design space for heterogeneous systems on chip," *Journal of VLSI Signal Processing Systems*, vol. 40, no. 1, pp. 19–34, 2005.

[3] K. Keutzer, S. Malik, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1523–1534, December 2000.

[4] A. Kossiakoff and W. N. Sweet, *Systems Engineering - Principles and Practice.* Hoboken, New Jersey: John Wiley & Sons, 2003.

[5] Y. Zorian, S. Dey, and M. J. Rodgers, "Test of future system-on-chips," in *ICCAD '00: Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*. Piscataway, NJ, USA: IEEE Press, 2000, pp. 392–398.

[6] K. E. Wiegers, *Software requirements*. Microsoft Press, 2003.

[7] T. Weilkiens, *Systems Engineering with SysML / UML*. Morgan Kaufmann OMG Press, 2008.

[8] D. Hatley, P. Hruschka, and I. Pirbhai, *Process for system architecture and requirements engineering*. Dorset House Publishing, 2000.

[9] P. Mishra and N. Dutt, "Functional validation of programmable architectures," in *DSD '04: Proceedings of the Digital System Design, EUROMICRO Systems*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 12–19.

[10] D. K. Hitchins, *Systems Engineering*. John Wiley & Sons, 2007.

[11] B. Beizer, *Black-Box Testing*. John Wiley & Sons, 1995.

[12] J. N. Martin, "Using the PICARD theory of systems to faciliate better systems thinking," *INCOSE INSIGHT*, vol. 11, no. 1, pp. 37–41, 2008.

[13] B. Broekman and E. Notenboom, *Testing Embedded Software*. London, U.K.: Addison Wesley, 2003.

[14] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. New York: IEEE Press, 2001.

[15] Y. Zorian, E. J. Marinissen, and S. Dey, "Testing embedded-core based system chips," in *ITC '98: Proceedings of the 1998 IEEE International Test Conference*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 130–143.

[16] A. Spillner, "The W-MODEL – strengthening the bond between development and test," in *STAReast 2002*, Orlando, Florida, USA, May 2002.

[17] J. G. Lamm, A. K. Berg, and C. G. Glück, "Synthetic signals for verifying noise reduction systems in digital hearing instruments," in *EUSIPCO 2008: Proceedings of the 16th European Signal Processing Conference*, Lausanne, Switzerland, August 2008.