Computing and Informatics, Vol. 31, 2012, 411–426

# TEST STRATEGIES FOR EMBEDDED ADC CORES IN A SYSTEM-ON-CHIP, A CASE STUDY

Franc Novak, Peter Mrak, Anton Biasizzo

*Computer Systems Department*
*Jožef Stefan Institute*
*Jamova cesta 39*
*1000 Ljubljana, Slovenia*
*e-mail:* {franc.novak, peter.mrak, anton.biasizzo}@ijs.si

Communicated by Elena Gramatová

**Abstract.** Testing of a deeply embedded mixed-signal core in a System-on-Chip (SoC) is a challenging issue due to the communication bottleneck in accessing the core from external automatic test equipment. Consequently, in many cases the preferred approach is built-in self-test (BIST), where the major part of test activity is performed within the unit-under-test and only final results are communicated to the external tester. IEEE Standard 1500 provides efficient test infrastructure for testing digital cores; however, its applications in mixed-signal core test remain an open issue. In this paper we address the problem of implementing BIST of a mixed-signal core in a IEEE Std 1500 test wrapper and discuss advantages and drawbacks of different test strategies. While the case study is focused on histogram based test of ADC, test strategies of other types of mixed-signal cores related to trade-off between performance (i.e., test time) and required resources are likely to follow similar conclusions.

**Keywords:** System-on-chip, built-in self-test, test strategies, mixed-signal testing, histogram test

**Mathematics Subject Classification 2010:** 94, 94C12

## 1 INTRODUCTION

System-on-chip (SoC) design integrates large reusable blocks (i.e. cores) that have been verified in earlier applications in practice. Embedded cores provide a wide range of functions, like CPUs, DSPs, interfaces, controllers, memories, and others. The advantage of embedding reusable cores in the design of a new product is a shortened design cycle resulting in reduced time-to-market and reduced cost.

The cores put together in a SoC normally originate from different core providers. In order to protect their intellectual property core providers do not completely reveal design and implementation details which makes the problem of SoC testing rather challenging to the core user (i.e., SoC designer). On the other hand, correct operation of a core in the target SoC is of interest of both core user and core provider. In order to provide an independent openly defined design-for-testability method to facilitate testing of integrated circuits containing embedded cores, IEEE 1500 Standard for Embedded Core Test [16] has been proposed and is now widely accepted in practice.

The main entity of the test architecture defined by IEEE Std 1500 is a test wrapper placed around each core of a SoC. Test wrapper provides an interface between the embedded core and its environment. For testing a core, a test source generating test vectors and a test sink collecting the test responses must be provided. Test access mechanism (TAM) transports test vectors from the source to the core and test responses from the core to the test sink. It also allows testing of interconnects between SoC cores. Test access architecture is sketched in Figure 1.

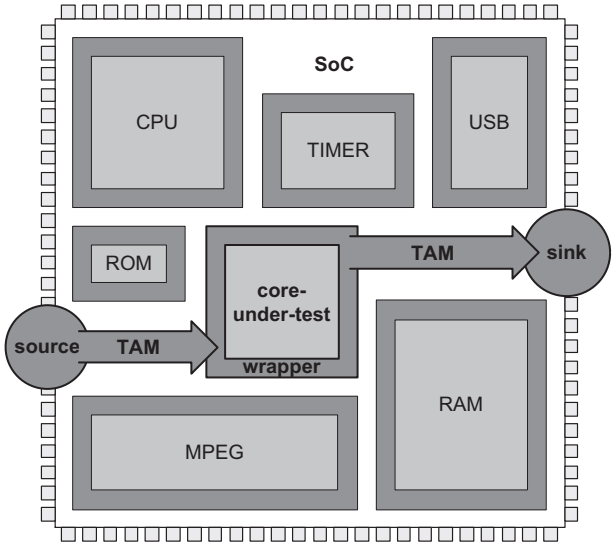



Fig. 1. Test access architecture

In a conventional test approach, TAM is connected to external automatic test equipment (ATE) which generates test vectors and collects test responses. However, for a SoC with complex embedded cores a huge amount of test data needs to be transferred between ATE and the core-under-test, which drastically increases test time. A viable alternative is to implement built-in self-test (BIST) in the target SoC. Embedded BIST logic takes the role of ATE by generating test vectors and analyzing test responses. In this way, the communication bottleneck between ATE and SoC is avoided, only the remaining low-speed operations required for the execution of the complete test of the core-under-test (such as initiation of test mode, start of BIST and evaluation of test results) are left to the ATE which can be managed by a low-cost tester. In a typical BIST implementation, test source and test sink are implemented within the test wrapper of the tested core. In an alternative solution, some other core in the SoC may take the role of test source and/or test sink.

While the problem of optimizing test infrastructure of digital cores has been investigated by many authors [19, 23, 24, 25], its counterpart in the area of mixed-signal cores remains an open issue. So far, no thorough analysis supported by empirical results from wrapper implementations in practice has been reported. Minimization of communication with external test instrumentation, hardware overhead and test time are common yet in many cases exclusive objectives. Since a general optimization problem is almost certainly intractable it is more reasonable to expect partial solutions suitable for specific test techniques. A design-for-test architecture that allows automated test development in compliance with IEEE Std 1500 supported by in-house development tools has recently been reported by Zivkovic et al. [29]. According to the authors, the proposed approach has been successfully used for test development and characterization of mixed-signal cores in several industrial products. The described environment presents an efficient platform for comparison and evaluation of different test strategies, yet their optimization issues have not been publicly elaborated.

In this paper, an attempt is made to share experiences regarding the trade-off between hardware overhead and test time in a BIST of a mixed-signal core implemented in a IEEE Std 1500 wrapper. A case study of three extreme BIST solutions for testing an embedded analog-to-digital (ADC) core using the popular histogram-based technique [11, 15, 17] is presented. The first BIST configuration employs test wrapper logic with minimum hardware overhead. The second solution aims at test wrapper optimized for minimum test time. In the third solution, processor core performs ADC core test and computes the required test parameters. Comparison of the employed resources and the corresponding test times may serve as an aid for selection of appropriate BIST strategy.

For the preliminaries, we briefly summarize the main points of histogram based test technique with reference to previous work.

## 2 HISTOGRAM BASED TEST TECHNIQUE – IN BRIEF

An ADC is a device that converts an input analog voltage to a digital number proportional to the magnitude of the input signal. The resolution of the converter is equal to the number of discrete values that the converter can produce over the range of analog values. ADC performance can be verified in terms of static performance parameters such as offset, gain, differential nonlinearity (DNL) and integral nonlinearity (INL). In histogram test, a known periodic input stimulus is applied and the histogram of code occurrences (i.e., counts of individual codes) is computed over an integer number of input wave periods. From the collected results static ADC parameters can be determined. The principle is sketched in Figure 2. Histogram test has been attracting research attention for more than two decades. Papers [6, 7, 8, 9, 4, 5, 12, 14, 10, 26, 27] primarily concentrate on how the code density can be interpreted to compute the differential and integral nonlinearities, gain error and offset error, and estimate the achieved accuracy under different measurement conditions. Papers [1, 2, 21, 28] focus on implementation of histogram based test of ADCs in a BIST arrangement. A triangle-wave input signal is the most convenient for BIST of static parameter or linearity, because it can be characterized only by two values (count of extreme codes and count of other codes). The count of extreme codes is normally greater than the count of non-extreme codes due to the overflow of the input signal.

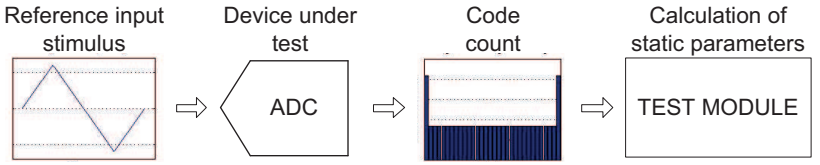



Fig. 2. The principle of histogram test with triangle-wave input signal

Coherent sampling must be provided: the ratio between the number of samples $N$ and the number of input signal periods $M$ must be equal to the ratio between the sampling frequency and the frequency of the input signal. In addition, the slope of the input signal is determined by the reference voltage $V_{\mathrm{ref}}$, the number of ADC bits $n$ and the number of input signal periods $M$.

The calculation of static parameters in the case of triangle-wave input stimulus is described in more details below in order to get a better understanding of measurements and computations performed by the BIST solutions presented in the paper.

### 2.1 Offset Error

The first output code transition should occur at an analog voltage that is 0.5 LSB (least significant bit) above zero. The offset error is the deviation of the first actual

code transition from 0.5 LSB. In histogram based test, the offset error is calculated by the following equation

$$\text{Offset} = \frac{H(2^n - 1) - H(0)}{2H_{\text{ideal}}}, \tag{1}$$

where for a $n$-bit ADC: $H(2^n - 1)$ is the number of occurrences of the MSB code and $H(0)$ of the LSB code, respectively. In all equations, $H_{\text{ideal}}$ represents the count of each code theoretically determined for the ideal ADC and the given stimulus waveform. (Thus, $H_{\text{ideal}}$ is an analytically obtained value.)

### 2.2 Gain Error

The gain error is the error in the slope of the transfer curve. For an ideal ADC, the gain is equal to 1 and the code count for any non-extreme code is equal to $H_{\text{ideal}}$. The ratio between the count of any non-extreme code and the ideal count $H_{\text{ideal}}$ determines the actual ADC gain at that code. If the code count for a given code is smaller than $H_{\text{ideal}}$, the actual ADC gain at that code is greater than the ideal gain, and vice versa. While the measured value varies from code to code, ADC gain estimation is determined by averaging over $m$ central codes of the range of the ADC

$$\text{Gain}^{-1} = \frac{\sum_{i=N_1}^{N_2} H(i)}{mH_{\text{ideal}}}. \tag{2}$$

### 2.3 Non-Linearity

Non-linearity can be presented by two different measures: differential non-linearity (DNL) and integral non-linearity (INL).

DNL measures the relative deviation of each code width from the ideal value (1 LSB). Given the histogram data, DNL is defined as the relative difference between the measured and the ideal code counts

$$\text{DNL}(i) = \frac{H(i) - H_{\text{ideal}}}{H_{\text{ideal}}}. \tag{3}$$

The INL determines the deviation of each output code center from the ideal straight center line. The INL for code $i$ is computed from the cumulative sum of DNLs of the previous codes

$$\text{INL}(i) = \sum_{j=1}^{i} \text{DNL}(j). \tag{4}$$

The above equations include the operations of addition, subtraction, and division. While addition and subtraction are fairly simple operations, hardware implementation of division is more demanding. However, if the denominator is the power

of 2, the division can be performed as shifting of the decimal point. In our case denominators are $H_{\text{ideal}}$, $2H_{\text{ideal}}$, and $mH_{\text{ideal}}$. In order to simplify the division, the $H_{\text{ideal}}$ and $m$ should be the power of 2. For subsequent explanation assume that $H_{\text{ideal}} = 2^P$, where $P$ is some suitable integer value.

## 3 BIST STRUCTURES

The concept of the implemented histogram based BIST in a IEEE Std 1500 wrapper is shown in Figure 3 a). We use triangle-wave input signal, which in an ideal case results in a constant code count for non-extreme codes and equal code count for extreme codes.

We first implemented two different BIST structures in a IEEE Std 1500 wrapper targeted either at low hardware overhead or at minimum test time. The first solution performs measurements and computations of static parameters in a sequence of individual steps. The ADC output data is processed on the fly in each step and the results (i.e., Offset, Gain, maximum value of DNL($i$), maximum value of INL($i$)) are transmitted out via test access mechanism. The second solution first collects the complete ADC test responses and stores them in a RAM. Next, the computation of static parameters is performed and the results are transmitted via TAM as in the previous case. We denote the first solution by sequential BIST and the second by RAM-based BIST.

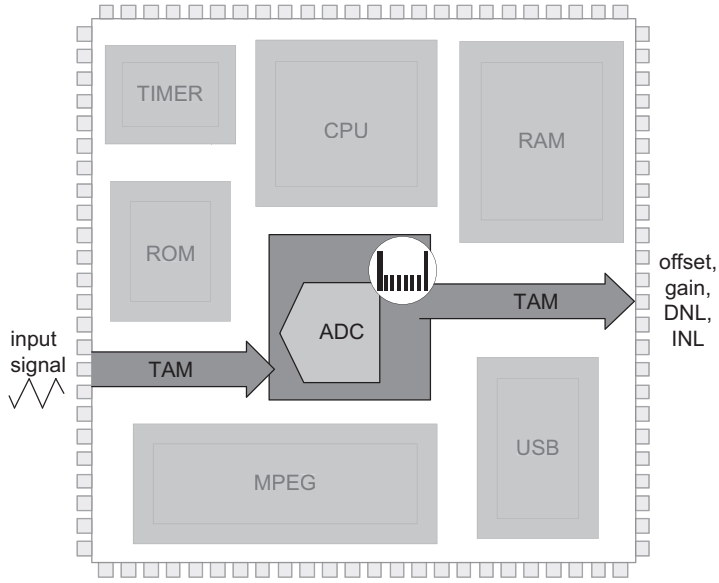
In addition, another solution (shown in Figure 3 b)) was conceived in which a processor core performs ADC core test and computes the required test parameters.

### 3.1 Sequential BIST Approach

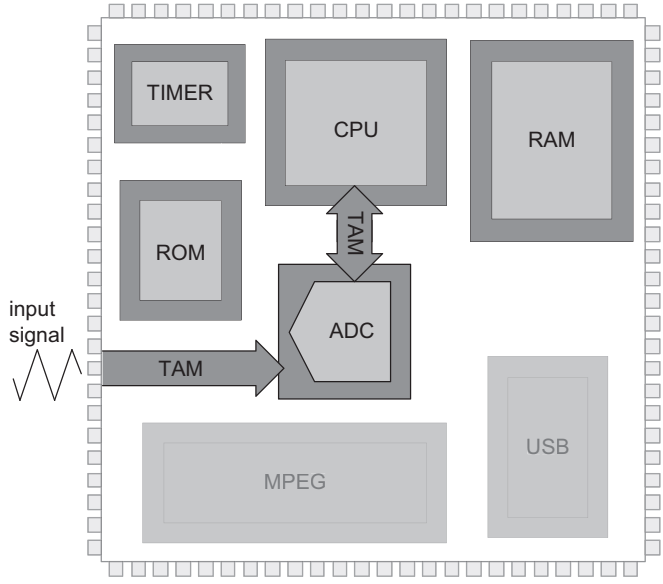
The sequential BIST structure depicted in Figure 4 operates in accordance with the concept proposed in [2]. It was, however, a bit modified in order to completely automate the evaluation of the DNL and INL parameters. The BIST structure is composed of three basic blocks:

- detector module, which monitors the codes generated by the ADC and signals when the code is equal to the preset value,
- exploitation module, which receives the signals from the detector module and performs the required operations (counting, complementing, etc.),
- control module, which coordinates the processing of both modules and connects the BIST structure to the IEEE Std 1500 test infrastructure. The control module is a state machine which generates the corresponding control signals of the detector module and exploitation module. It also coordinates the operation of BIST and the IEEE Std 1500 test wrapper logic.

The detector module contains a counter and a comparator. The counter is used for setting the required preset value which is stored in the DM register. The contents of the DM register is compared with the ADC code. If the ADC code equals the




a)




b)

Fig. 3. The concept of a) implemented BIST in IEEE Std 1500 test wrapper, b) processor-based BIST

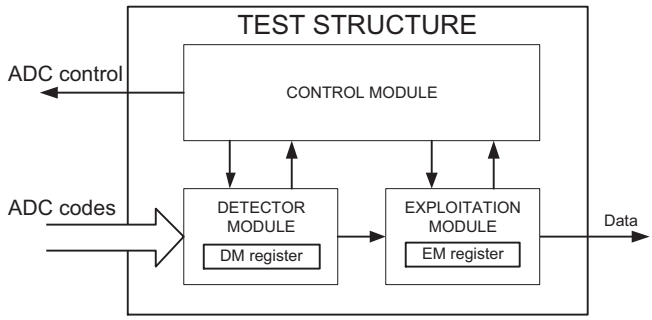



Fig. 4. Sequential BIST structure

contents of the DM register, exploitation module is triggered. The exploitation module is an up/down counter with clear. In addition it is capable to calculate the complement of the current value. The temporary result is stored in the EM register.

The calculations of offset, gain, DNL and INL are separate processes performed in strictly sequential manner. While the sequential BIST structure described above requires low hardware overhead, the test time may become excessive. This is due to the fact that sequential approach takes $N$ samples for the calculation of Offset, $mN$ samples for the calculation of Gain, $(2^n - 2)N$ samples for the calculation of DNL and $(2^n - 2)N$ samples for the calculation of INL. As an alternative, RAM-based BIST has been developed.

### 3.2 RAM-Based BIST Approach

The RAM-based BIST test structure shown in Figure 5 consists of three components, which play a similar role as their counterparts in sequential BIST. The three components are:

- RAM module, in which the ADC code count of a complete histogram is stored,
- computation module, which computes the ADC static characteristics from the values stored in RAM,
- control module, which (similar to the sequential BIST) coordinates the processing of both blocks and connects the BIST structure to the IEEE Std 1500 test infrastructure.

In contrast to sequential BIST approach, only $N$ samples are taken for the computation of static parameters of ADC. The samples are first stored in RAM, then the static parameters are computed in accordance with Equations (1)–(4). Some programming tricks are used in order to simplify the computation. For example, subtraction in Equation (3) can be eliminated if the counts are first initialized to $-H_{\text{ideal}}$ and then increased by the number of occurrences of the code. RAM is

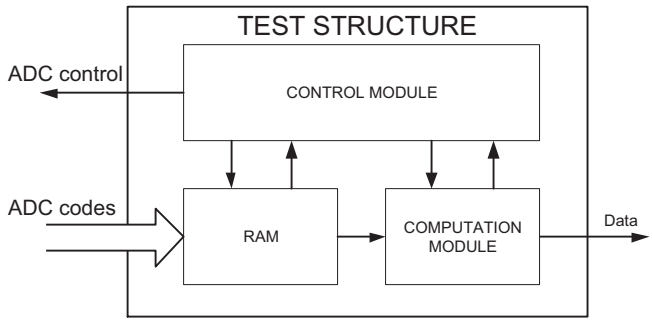



Fig. 5. RAM-based BIST structure

organized in such way that the ADC code actually presents the RAM address of its code count. In this way, data manipulation is simplified.

### 3.3 Processor-Based BIST

Processor-based BIST exploits available microprocessor and other cores on the SoC for the execution of the ADC test. The idea is to reuse already present resources in the SoC and their interconnections to perform functional test of the ADC core. The microprocessor with timer interactions controls the ADC core, gathers the responses and stores them in the SoC RAM. After the stimulation period the gathered responses are evaluated by the processor. In a sense the processor with timer performs the role of the control and computation module. While this approach does not introduce any notable hardware overhead in the digital part of the SoC, additional ROM storage is required for the BIST program storage. In order to perform ADC BIST at full speed, which is generally faster than normal SoC operation, the processor in collaboration with other cores must be able to respond in time to the incoming ADC codes.

## 4 INTEGRITY CHECK OF THE TEST STRUCTURE

Integrity of the test infrastructure must be assured before we can trust the results of the implemented histogram test. For this purpose, integrity check is performed by replacing the tested ADC by a counter as shown in Figure 6. The counter generates codes corresponding to the ideal ADC with ideal triangular input waveform and the resulting static parameters are checked (Offset = 0, Gain = 1, DNL = 0, INL = 0). The two empty boxes in Figure 6 represent either detector and exploitation module, or RAM and computation module since the same integrity check is used for sequential or RAM-based BIST.

In processor-based BIST, integrity check of the test infrastructure includes testing of processor, timer and RAM cores, test of interconnects and test of ROM with

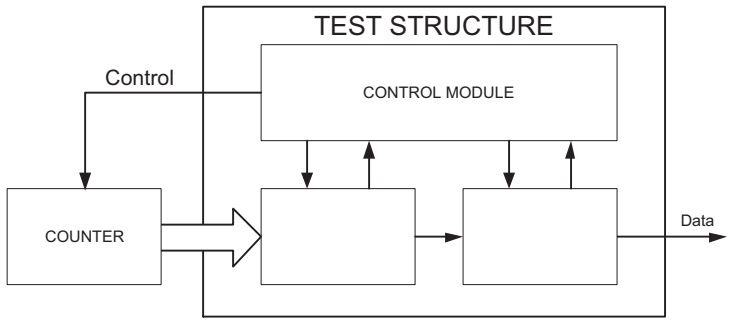



Fig. 6. Integrity check of a test structure

stored BIST program. Since most of the above activities are included in general SoC test, only the test of ROM (i.e., checksum of its contents) is required in addition.

## 5 IMPLEMENTATION ISSUES

IEEE Std 1500 test wrapper with the above BIST structures was implemented in a Spartan3 XC3S200 FPGA. ADC0808 (an 8-bit ADC) was initially chosen as the unit-under-test. Measured parameters of the employed ADC were in conformance with the specifications provided by the manufacturer which demonstrates the feasibility of the histogram based test technique by the implemented test structures. Let us now compare the three solutions from some different points that may serve as the basis for selection of suitable strategy for BIST implementation of the embedded core.

The measurement uncertainty $\Delta$ (expressed in terms of LSB) is defined by

$$\Delta = \frac{1}{H_{\text{ideal}}}. \tag{5}$$

In sequential BIST approach, the number of samples required for processing a single ADC code is

$$N \approx 2^n H_{\text{ideal}}, \tag{6}$$

where $n$ is the number of ADC bits. In our case, test signal was generated such that the number of occurrences of the two extreme codes was equal to the number of occurrences of the non-extreme codes, which gives

$$N = 2^n H_{\text{ideal}}. \tag{7}$$

The number of samples needed for the complete measurement is

$$N_{\text{all}} = (1 + m + 2^{n+1} - 4)N, \tag{8}$$

where $m$ is the number of codes on which the gain calculation is performed.

In RAM-based BIST and processor-based BIST approach all codes are processed simultaneously. The number of samples required for the complete measurement is

$$N_{\text{all}} = N = 2^n H_{\text{ideal}}. \tag{9}$$

Table 1 presents measurement uncertainty $\Delta$ and overall test time of the three approaches for different values of $H_{\text{ideal}}$ at 5 $\mu$s sample time.

| | measurement | Test time [s] for 8-bit ADC | | |
|---|---|---|---|---|
| | uncertainty | sequential | RAM-based | processor-based |
| $H_{\text{ideal}}$ | $\Delta$ [LSB] | BIST | BIST | BIST |
| 4 | 0.250 | 3.37 | 0.725 | 0.729 |
| 8 | 0.125 | 6.08 | 0.732 | 0.838 |
| 16 | 0.063 | 11.49 | 0.745 | 0.867 |
| 32 | 0.031 | 22.33 | 0.772 | 0.990 |
| 64 | 0.016 | 43.99 | 0.825 | 1.362 |

Table 1. Measurement uncertainty and overall test time of sequential, RAM-based and processor-based BIST

Notice that the test time in the case of sequential BIST is much higher than that of RAM-based BIST and processor-based BIST. While the test time of the 8-bit ADC at measurement uncertainty of 0.250 LSB may still be acceptable, the situation gets worse when testing ADC cores higher than 8-bit. Table 2 shows the test time of sequential, RAM-based and processor-based BIST for 8, 10, 12 and 16-bit ADC. The test times of sequential BIST of 12 and 16-bit ADCs are obviously not acceptable in practice.

| | Test time [s] ($H_{\text{ideal}} = 4$) | | |
|---|---|---|---|
| number of | sequential | RAM-based | processor-based |
| ADC bits | BIST | BIST | BIST |
| 8 | 3.37 | 0.725 | 0.729 |
| 10 | 44.18 | 0.907 | 0.922 |
| 12 | 697.69 | 1.161 | 1.278 |
| 16 | 178,543.32 | 3.116 | 3.222 |

Table 2. Test time of sequential, RAM-based and processor-based BIST for different ADCs

RAM-based BIST and processor-based BIST are much faster but there are other issues that should be considered. As regards hardware overhead, RAM-based BIST employs much more resources than sequential BIST. We synthesized both approaches with Cadence RTL Compiler and the resulting designs are shown below. In order to get a realistic view of hardware overhead of individual solutions, sequential and RAM-based BIST logic implementation without IEEE Std 1500 wrapper logic are summarized in Table 3.

Next, the resources of complete BIST implementations including wrapper control logic of the three solutions are summarized in Table 4.

| number of ADC bits | sequential BIST | | | RAM-based BIST | | |
|---|---|---|---|---|---|---|
| | gates | flip-flops | RAM cells | gates | flip-flops | RAM cells |
| 8 | 881 | 233 | – | 461 | 156 | 2 560 |
| 10 | 911 | 249 | – | 541 | 172 | 12 288 |
| 12 | 961 | 265 | – | 583 | 188 | 57 344 |
| 16 | 1 067 | 297 | – | 684 | 220 | 1 179 648 |

Table 3. Sequential and RAM-based BIST logic implementation (without IEEE Std 1500 wrapper logic)

| number of ADC bits | sequential BIST | | | RAM-based BIST | | | processor-based BIST | | |
|---|---|---|---|---|---|---|---|---|---|
| | gates | flip-flops | RAM cells | gates | flip-flops | RAM cells | gates | flip-flops | RAM cells |
| 8 | 1 012 | 293 | – | 594 | 216 | 2 560 | 63 | 59 | – |
| 10 | 1 063 | 319 | – | 612 | 242 | 12 288 | 71 | 72 | – |
| 12 | 1 131 | 345 | – | 680 | 268 | 57 344 | 97 | 95 | – |
| 16 | 1 208 | 371 | – | 889 | 320 | 1 179 648 | 205 | 127 | – |

Table 4. BIST implementations including IEEE Std 1500 wrapper control logic

Processor-based BIST has minimal hardware overhead since the majority of its tasks are performed by the cores already included in a SoC. However, such a solution is specific for the given processor core and is not easily portable to other SoCs. On the other hand, sequential BIST and RAM-based BIST are autonomous and can be ported together with ADC core to other designs.

Scalability of the above solutions manifest itself in different issues. Analysis of BIST solutions for $n$-bit ADC cores shows that by increasing $n$ the hardware overhead of sequential BIST increases linearly, exhibits exponential growth in the case of RAM-based BIST and practically does not change in processor-based BIST. Test time of sequential BIST grows exponentially while practically does not change in the case of RAM-based or processor-based BIST.

Providing high-quality stimulus signals is crucial to accomplish efficient and consistent BIST of a mixed-signal core. The signal can be internally generated in the test wrapper, or supplied via TAM either from an external signal generator or generated by another core (if available) in the SoC-under-test. In our case, including a triangle-wave generator [3] in the test wrapper would considerably increase the complexity hence externally supplied stimulus is a preferred approach. The impact of MOS switches connecting the external source to the ADC core was simulated by introducing an ABM module of a IEEE Std 1149.4 compliant test chip. For this purpose, our proprietary test chip [18] was employed. In order to assess the impact of MOS switches of analog boundary module on static parameter measurements, direct connection to the ADC input was optionally provided. The measured parameters were in good agreement with the values obtained by the conventional technique

using laboratory measurement equipment which gives confidence to the described approach.

## 6 CONCLUSIONS

The case study demonstrated that the three BIST solutions considerably differ in required resources, achieved test time, portability and scalability. Selection of the best test approach in practice depends on strategic decision taken in the early user requirement specification phase and is, of course, specific for each mixed-signal core of a SoC. Although the described implementation focused on a specific test technique (histogram based test) for a specific mixed-signal core (ADC) the three approaches can be generalized to other types of mixed-signal cores as follows:

- test wrapper performs core test and evaluates the results on the fly,
- test results (i.e., measurements) are stored in RAM and processed in batch,
- other cores of the SoC (if available) take the role of testing.

In purely digital SoC, the first two approaches are preferred since in most cases stimuli are generated by a LFSR (linear-feedback shift register) logic and test results are collected by a MISR (multiple input signature register) based on the same principles. This logic is compact and scalable and provides portability of design.

BIST of a SoC with mixed-signal cores faces different problems. Test methods possibly implemented in a test wrapper require specific logic which differs considerably from one case to another. There are no common parts, everything must be conceived from scratch. In fact, since a mixed-signal core is put in a SoC with well defined target application it is prudent to consider possible reuse of the application software (i.e., device drivers) in a processor-based BIST.

Processor-based BIST is traditionally employed in testing embedded memories. Its application in mixed-signal core test has not been widely investigated in literature although proprietary solutions in practice no doubt exist. In this regard, the comparison of empirical data presented in the paper allows an assessment of pros and cons of the two alternative approaches and may serve as a guideline in similar applications.

Generation of test stimuli is another issue in which BIST of mixed-signal cores differs from BIST of digital cores. While test pattern generation with LSFR is a common and widely practiced technique in digital word [13], analog stimulus generation [22] is a problem with no unique solution. Test signal generator included in the test wrapper results in increased complexity and hardware overhead. Besides, integrity check of a signal generator embedded in a test wrapper poses additional problems and increases test time. A simple alternative solution employing external signal generator is more likely to be applied in practice. In some cases, another core of a SoC might take the role of stimulus generator, like for example oscillation-based test structures reported in [20].

At this time, IEEE Std 1500 test wrapper is defined only for digital cores. Its possible future extension to mixed-signal cores will probably follow the way of IEEE Std 1149.1 extended to IEEE Std 1149.4. The impact of MOS switches connecting the input signals (either from external source or from another core in a SoC) to the mixed-signal core will remain an important issue in practice.

## REFERENCES

[1] Azaïs, F.—Bernard, S.—Bertrand, Y.—Renovell, M.: Towards an ADC BIST scheme using the histogram test technique. In Proceedings of the IEEE European Test Workshop Proceedings, pp. 53–58, 2000.

[2] Azaïs, F.—Bernard, S.—Bertrand, Y.—Renovell, M.: Implementation of linear histogram BIST for ADCs. In Proceedings of the Design Automation & Test in Europe Conference, pp. 590–595, 2001.

[3] Bernard, S.—Azaïs, F.—Bertrand, Y.—Renovell, M.: On-Chip Generation of Ramp and Triangle-Wave Stimuli for ADC BIST. Journal of Electronic Testing, Vol. 19, 2003, No. 4, pp. 469–479.

[4] Blair, J.: Histogram Measurement of ADC Nonlinearities Using Sine Waves. IEEE Transactions on Instrumentation and Measurement, Vol. 43, 1994, No. 3, pp. 373–383.

[5] Carbone, P.—Chiorboli, G.: ADC Sinewave Histogram Testing With Quasi-Coherent Sampling. IEEE Transactions on Instrumentation and Measurement, Vol. 50, 2001, No. 4, pp. 949–953.

[6] Corrêa Alegria, F.—Arpaia, P.—Daponte, P.—Cruz Serra, A.: An ADC Histogram Test Based on Small-Amplitude Waves. Measurement, Vol. 31, 2002, No. 4, pp. 271–279.

[7] Corrêa Alegria, F.—Cruz Serra, A.: Error in the Estimation of Transition Voltages With the Standard Histogram Test of ADCs. Measurement, Vol. 35, 2004, No. 4, pp. 389–397.

[8] Corrêa Alegria, F.—Cruz Serra, A.: Overdrive in the Standard Histogram Test of ADCs. Measurement, Vol. 35, 2004, No. 4, pp. 381–387.

[9] Corrêa Alegria, F.—Cruz Serra, A.: Standard Histogram Test Precision of ADC Gain and Offset Error Estimation. IEEE Transactions on Instrumentation and Measurement, Vol. 56, 2007, No. 5, pp. 1527–1531.

[10] Cruz Serra, A.—Corrêa Alegria, F.—Martins, R.—Fonseca da Silva, M.: Analog-to-Digital Converter Testing-New Proposals. Computer Standards & Interfaces, Vol. 26, 2004, No. 1, pp. 3–13.

[11] Dallet, D.—Machado da Silva, J.: Dynamic Characterisation of Analogue-to-Digital Converters. Springer, 2005.

[12] Doemberg, J.—Lee, H.-S.—Hodges, D. A.: Full-Speed Testing of A/D Converters, IEEE Journal of Solid State Circuits, Vol. 19, 1984, No. 6, pp. 820–827.

[13] Garbolino, Tomasz.—Papa, G.: Genetic Algorithm for Test Pattern Generator Design. Applied Intelligence, Vol. 32, 2010, No. 2, pp. 193–204.

[14] Hsin-Wen, T.—Bin-Da, L.—Soon-Jyh, C.: Histogram Based Testing Strategy for ADC. In Proceedings of the 15th Asian Test Symposium, pp. 51–54, 2006.

[15] Huertas, J. L.: Test and Design-for-Testability in Mixed-Signal Integrated Circuits. Kluwer Academic Publishers, 2004.

[16] IEEE Std 1500 – Standard for Embedded Core Test. Availaible on: `http://grouper.ieee.org/groups/1500/`, 2009.

[17] Mahoney, M.: DSP-Based Testing of Analog and Mixed-Signal Circuits. IEEE Computer Society Press, 1987.

[18] Kač, U.—Novak, F.—Azaïs, F.—Nouet, P.—Renovell, M.: Extending IEEE Std. 1149.4 Analog Boundary Modules to Enhance Mixed-Signal Test. IEEE Design & Test of Computers, Vol. 20, 2003, No. 2, pp. 32–39.

[19] Nadeaut-Dostie, B.—Adham, S. M. I.—Abbott, R.: Improved Core Isolation and Access for Hierarchical Embedded Test. IEEE Design & Test of Computers, Vol. 26, 2009, No. 1, pp. 18–25.

[20] Novak, F.—Santo Zarnik, M.—Kač, U.: Stimulus Generation of a Built-In-Self-Test Using Oscillation Based Test Structures. International Journal of Electronics, Vol. 89, 2002, No. 11, pp. 811–820.

[21] Renovell, M.—Azaïs, F.—Bernard, S.—Bertrand, Y.: Hardware Resource Minimization for a Histogram-Based ADC BIST. In Proceedings of the 18th IEEE VLSI Test Symposium Proceedings, pp. 247–252, 2000.

[22] Roberts, G. W.—Lu, A. K.: Analog Signal Generation for Built-in Self-Test of Mixed Signal Integrated Circuits. Kluwer Academic Publishers, 1995.

[23] Sandeep, K. G.—Marinissen, E. J.: A Test Time Reduction Algorithm for Test Architecture Design for Core-Based System Chips. Journal of Electronic Testing, Vol. 19, 2003, No. 4, pp. 425–435.

[24] Sehgal, A. E. J.—Sandeep, K. G.—Marinissen, E. J.—Chakrabarty, K.: Hierarchy-Aware and Area-Efficient Test Infrastructure Design for Core-Based System Chips. In Proceedings of the Design Automation & Test in Europe Conference, pp. 285–290, 2006.

[25] Sinanoglu, O.—Marinissen, E. J.: Analysis of the Test Data Volume Reduction Benefit of Modular SOC Testing. In Proceedings of the Design Automation & Test in Europe Conference, pp. 182–187, 2008.

[26] Vanden Bossche, M.—Schoukens, J.—Renneboog, J.: Dynamic Testing and Diagnostics of A/D Converters. IEEE Transactions on Circuits and Systems, Vol. 33, 1986, No. 8, pp. 775–785.

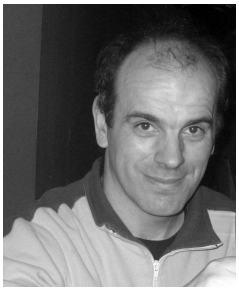
[27] Wagdy, M. F.—Awad, S. S.: Determining ADC Effective Number of Bits Via Histogram Testing. IEEE Transactions on Instrumentation and Measurement, Vol. 40, 1991, No. 4, pp. 770–772.

[28] Yong-Sheng, W.—Jin-Xiang, W.—Feng-Chang, L.—Yi-Zheng, Y.: Optimal Schemes for ADC BIST Based on Histogram. In Proceedings of the 14th Asian Test Symposium, pp. 52–57, 2005.

[29] Zivkovic, V.—Schat, J.—van der Heyden, F.—Seuren, G.: Core-Based Testing of Embedded Mixed-Signal Modules in a SoC. IEEE Design & Test of Computers, Vol. 26, 2009, No. 3, pp. 78–86.

**Franc Novak** gained B. Sc., M. Sc., and Ph. D. degrees in electrical engineering from the University in Ljubljana in 1975, 1977, and 1988, respectively. Since 1975 he has been with the Jožef Stefan Institute, where he is currently the head of the Computer Systems Department. He is also Professor at the Faculty of Electrical Engineering and Computer Science, University of Maribor. His research interests are in the areas of electronic testing and diagnosis, and fault-tolerant computing. His most recent assignment has been on designs for the testability of analogue circuits.

**Peter Mrak** gained his Ph. D. degree at Jožef Stefan International Postgraduate School, Ljubljana. After his Ph. D. work he was with Gorenje d.d. developing new generation refrigeration appliances. Currently he is the head of research and development department of Tesnila GK, Prevalje. His research interests include electronic test, system diagnosis, control of cooling systems and quality control.



**Anton Biasizzo** has been a researcher at the Jožef Stefan Institute since 1991. He received a Ph. D. degree from the University in Ljubljana in 1998. His research interests include efficient algorithms for sequential diagnosis, constraint logic programming, model-based diagnosis and automatic test-pattern generation. He has also designed compact AES encryption/decryption cores for dependable applications. Currently he is working on built-in self-test and on-line test and repair of FPGA based systems.