World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:5, No:11, 2011

Concurrent Testing of ADC for Embedded System

Y.B.Gandole

***Abstract*—**Compaction testing methods allow at-speed detecting of errors while possessing low cost of implementation. Owing to this distinctive feature, compaction methods have been widely used for built-in testing, as well as external testing. In the latter case, the bandwidth requirements to the automated test equipment employed are relaxed which reduces the overall cost of testing. Concurrent compaction testing methods use operational signals to detect misbehavior of the device under test and do not require input test stimuli. These methods have been employed for digital systems only. In the present work, we extend the use of compaction methods for concurrent testing of analog-to-digital converters. We estimate tolerance bounds for the result of compaction and evaluate the aliasing rate.

***Keywords*—**Analog-to Digital Converter, Embedded system, Concurrent Testing

I. Introduction

MODERN very large-scale integrated mixed-signal devices often incorporate a microcontroller, field-programmable gate array and programmable analog [1]. These devices, referred to as systems-on-chip can represent a complete solution for many tasks. A task typically involves the acquisition of analog signals with further conversion to discrete form, digital processing, and the final conversion of the results back to analog form. Conversion of analog signals to digital form is performed by an analog-to-digital converter (ADC), whereas the opposite process is done by a digital-to-analog converter (DAC). An SOC may contain few ADCs and DACs.

The complexity of an ADC is normally much higher than that of a DAC. This makes an ADC more vulnerable to failures. In addition to sudden failures, it can be subject to gradual failures due to the presence of analog components. Even small deviations of the parameters of these components may drastically influence the overall operation of the device. The more accurate is an ADC, the greater could be the effect.It is therefore equally important to detect any of these types of a failure in the operation of an ADC.

Advances in digital testing have allowed the creation of a low-cost at-speed concurrent testing technique based on compaction [2]. The compaction technique has also been adopted for mixed-signal circuits [3]. However, the method examined in [3] is non-concurrent (off-line). Consequently, it requires the system under test to be switched off the normal operation and analog test stimuli to be applied to its inputs. Unlike to the purely digital signals, generation of accurate analog stimuli is notas easy of a task. Apparently, this problem would disappear with the development of a concurrent testing technique for an ADC. In the present work, we consider such a technique. The known off-line mixed-signal system compaction methods are based on the estimation of the sum (signature) of the digitized analog signals appearing at the test point during the test session. The signature is then compared against the fault-free circuit signature subject to tolerance bounds. These bounds have been estimated experimentally. We show how to calculate them theoretically, thereby making the process more accurate and defining the theoretically achievable limit. The aliasing rate will also be evaluated.

Failures in a system manifest themselves as errors. The problem of efficient error control has been studied in the theory of error-control codes. We will attempt to extend some principles of error-control coding to testing ofADCs.

Y.B. Gandole is with Dept. of Electronics Adarsha Science J.B.Arts and Birla Commerce Mahavidyalaya Dhamangaon Rly, India. e-mail: ygandole@indiatimes.com

II. Literature Survey

Signature analysis has been a digital testing technique that satisfies the requirements of small hardware overhead and low aliasing rate [4]. In this method, inputs of a device under test (DUT) are fed by test stimuli, while the output responses are compacted into a signature. The DUT is considered to befault-free, if the computed signature matches the one for the fault-free device. An example of the 3-bit digital signature analyzer (SA) is shown in figure 1. The operation of this SA is described by the polynomial $G(x) = x + \alpha$, where $\alpha$is the primitive element in the field $GF(2^3)$ (a root of the binary polynomial $g(x) = x^3 + x + 1$).

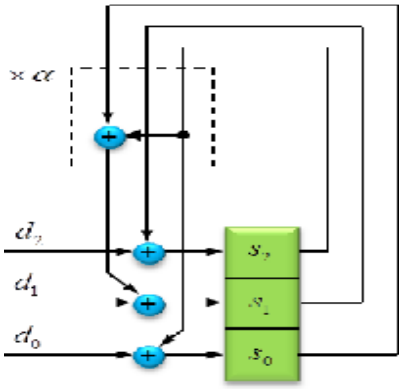

Fig. 1 A 3-bit digital signature analyzer

Feeding inputs of the DUT with the exhaustive sequence of digital test stimuli increases fault coverage, but compaction of the output responses causes some errors to escape detection due to aliasing. With the size of the signature being equal to 16, the aliasing rate is sufficiently low. This attractive feature has made signature analysis quite popularin the area of digital testing.

A similar technique has been used inthe analogue systems testing field. It is based on the estimation of the$mod\ 2^n$sum of the output words, where n is the resolution of the ADC employed [5].The compaction is done digitally; therefore, a high precision ADC is always present in an analogue SA (see figure 2for $n = 3$).In order to increase fault coverage and diminish the probability of error escape, the number of

analogue test stimuli must be large enough.In [6], the authors introduce a concurrent testing technique for combinational circuits based on compaction. An advanced concurrent testing technique is considered in [7], the authors apply it for ROM testing. Further improvement is done in [8], where authors offer a scheme exploiting "X" values in the output response. However, all these solutions have been applicable to digital circuits only. The concurrent testing technique for ADCs based on compaction has not been reported in the literature and is a subject of this work.In [16] authors apply signature analysis principle for compaction of output responses of an ADC. The permissible tolerance bounds for a fault-free ADC are determined and the aliasing rate is estimated.

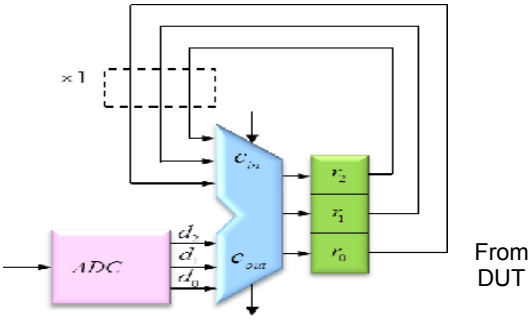

Fig. 2 A 3-bit analogue signature analyzer

III. Testing Method

In our case, the DUT is an ADC itself and the test stimuli are directly applied to its input. Strictly speaking, the exhaustive sequence of test stimuli, as applied to this particular case, should have included an infinite number of analogue signals that would cover the full scale range (FSR) of the ADC. In practice, this number is limited to the characteristic points of the FSR.If the ADC is a part of a measurement system that is intended to convert an active value (such as voltage), the test sequence can be generated by a precise waveform generator [9]-[11]. In the case of a system converting passive parameters (such as resistance), the input stimuli can be produced by high precision resistors (and/or capacitors) [12].

The proposed method is illustrated in figure 3. The Test Generator (TG) produces digital words that feed the Digital-to-Analog Converter (DAC). The output of the DAC is connected to one of the inputs of an Analog Comparator (AC). The operational analog signal feeds the ADC under test as well as the other input of the AC. When the operational analog signal matches the output signal of the DAC, a *HIT* signal is generated by the AC. The *HIT* signal initiates the operation of the Modulo Adder (referred to as a compactor or signature analyzer). At the same time, the *HIT* signal forces TG to produce the next digital pattern and the process repeats. When all patterns of the TG are exhausted, the final residue (signature) is compared with the reference signature against the tolerance bounds.

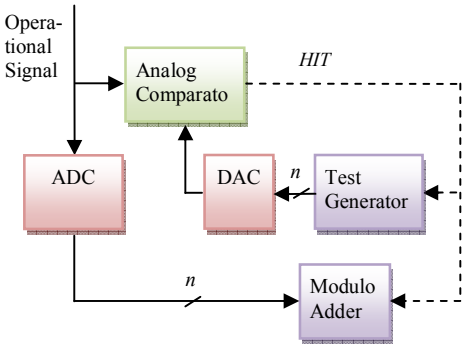

Fig. 3 A concurrent testing method for ADC

There could bea few issues associated with the scheme in figure3.The first issue is related to unavoidable quantization error present in the ADC. In a fault-free ADC, like the one represented in figure4 (here FSR=8V), transitions between steps of its transfer characteristic fluctuate within the permissible tolerance bounds, $k - 1 < T(k) < k$, where $T(k)$ is the k-th transition voltage of the ADC [13], [14]. Dotted lines surrounding the ideal transitions show the bounds. If we could apply a precise voltage, e.g. 4V, to the input of the ADC, the output code would be 100_2. However, the voltage that normally comes from the on-chip waveform generator is not accurate. Furthermore, the continued growth of the resolution of modern ADCs causes the quantization bin to shrink. Hence, the real test voltage (stimuli) becomes an interval value. Note that the rightmost permissible value for the transition $T(k)$ and the leftmost permissible value for the transition $T(k + 1)$ (e.g. points a and d in Figure 2, $k = 4$) lie in the infinitesimal neighbourhood of the centre of the quantization bin$k$. If an interval voltage that covers the centre of the bin is applied to the ADC, it will produce an interval code, even though no faults are present. For the 4V input signal, the permissible output codes will be 011, 100, and 101. This uncertainty complicates the use of algebraic compaction (shown in figure 1) for ADCs testing. Similarly, we can observe that the voltage, U_{in} belonging to the interval $|U_{in} - 3.5| < 0.5$, and being applied to the ADC, will produce only two permissible codes: 011 and 100. It must also be noted that refining the accuracy of the input voltage beyond 0.5 LSB does not reduce the output uncertainty and is, therefore, not required. This relaxes the accuracy requirements to the DAC. The DAC shown in figure 4 has an n-bit resolution, which matches the resolution of the ADC under test.

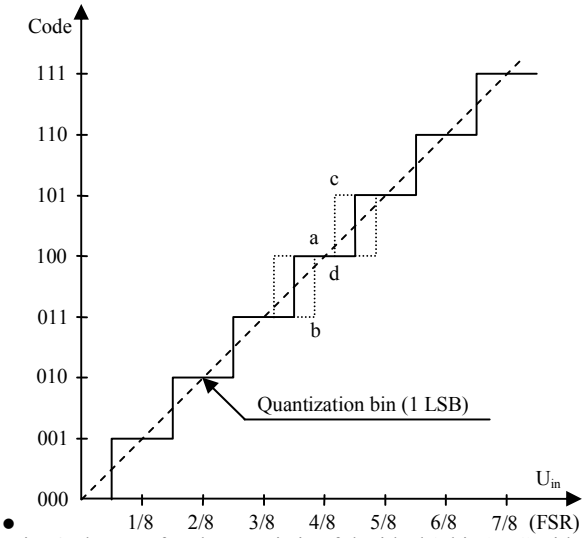

Fig. 4 The transfer characteristic of the ideal 3-bit ADC with FSR=8V

Failure in the analog or digital part of an ADC may change positions of the transition points beyond the permissible bounds, influencing the widths of the quantization bins. When two consecutive transition edges move toward each other, the width becomes zero. If this happens for a few adjacent quantization bins, there will be a sudden (exceeding unity) change in the output code. Based on this reasoning, we will consider the following (functional) ADC fault model. If the output code of an ADC fed by a test voltage exceeds the expected tolerance bounds (defined above), the ADC will be assumed faulty.The second issue appears with the selection of the type of response compactor. Similar to the principle used in a multiple-input signature analyzer, we will compress all *n* bits of the output codes of the ADC simultaneously. Figure 5 represents a structure of an arithmetic signature analyzer that does such a compression. In other words, it divides by $m = 2^{2n} + 1$ (*r* denotes a residue). However, if this analyzer is fed by interval digital values, the uncertainty of the final signature will increase even more, since these responses will be automatically multiplied by increasing powers of 2^n, where *n* is the resolution of the ADC being tested. Note that in the corresponding traditional signature analyzer (that multiplies by x^m) this problem does not occur at all and no uncertainty is introduced in the result of compaction. Therefore, for (arithmetic) compaction purposes we will use another class of arithmetic codes, namely modulo sum codes. This type of compactor (decoder) does not increase the uncertainty of the result of a compaction. The expanded structure of the 3-bit compactor that utilizes the above codes is shown in figure2.

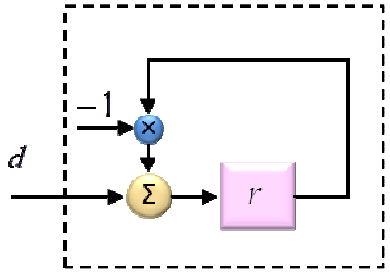

Fig. 5 An arithmetic compactor for a 3-bit ADC

In figure 3,the *HIT* signal is generated at the time when the input signal of the ADC is equal to the output signal of the DAC, x^0. Therefore, we can assume that the ADC under test is just fed by test stimuli x^0.Let *m* be the number of distinct values of the signal x^0that would be sufficient to detect all faults of a given class that may occur in the ADC. And let x_i^0be the value of x^oat time t_i. Then, the actual output code corresponding to the input value x_i^0will be $y_i = [x_i^0+0.5q] + \delta_i = y_i^0+\delta_i$, $i = 1,\ldots,m$. Here *q* is the width of the quantization bin; *[a]* represents the integer part of *a*; $y_i^0$is the ideal output code; and $\delta_i$is the actual error.

We will denote the permissible upper and lower tolerance bounds as $\hat{\delta}_i$and$\check{\delta}_i$,$i = 1,\ldots,m$. For a fault-free ADC: $\check{\delta}_i \le \delta_i \le \hat{\delta}_i$, $i = 1,\ldots,m$. Likewise, we can test the equivalent conditions,$\check{y}_i \le y_i \le \hat{y}_i$, where$\check{y}_i = y_i^0 + \check{\delta}_i$,$\hat{y}_i = y_i^0 + \hat{\delta}_i$. After adding up all the codes, the final sum will be:

$$\sum_{i=1}^{m} y_i = \sum_{i=1}^{m} y_i^0 + \sum_{i=1}^{m} \delta_i$$

Using the notations

$$Y = \sum_{i=1}^{m} y_i,\ Y^0 = \sum_{i=1}^{m} y_i^0,\ \Delta = \sum_{i=1}^{m} \delta_i$$

we will obtain $Y - Y^0 = \Delta$. Here Y^0 can be calculated based on the ideal transfer characteristic of the ADC. It has the same value regardless of the actual (faulty or fault-free) state of the ADC.

Since we consider symmetrical ADCs, then $\check{\delta}_i = \check{\delta}$, $\hat{\delta}_i = \hat{\delta}$ for every $i = 1,\ldots,m$. Taking this into account and introducing the following bounds:

$$\check{\Delta} = (\sum_{i=1}^{m} \check{\delta}_i) \bmod L, \quad \hat{\Delta} = (\sum_{i=1}^{m} \hat{\delta}_i) \bmod L$$

it can be shown that the ADC will be faulty, if the following condition is satisfied:

$$m\hat{\delta} < (Y - Y^0) \bmod L < L - m\left|\check{\delta}\right| \qquad (1)$$

Otherwise, we will assume that the ADC is fault-free. Here the residue $R = (Y - Y^0) \bmod L$ is the actual signature. Therefore, the fault free circuit signature must belong to one of the intervals: $[0, \check{\Delta}]$, or $[\hat{\Delta}, L-1]$.

Computation of the residue *R* is performed in the adder that is preliminarily loaded with the "seed" value, namely the two's complement of Y^0 , i.e. $\overline{Y}^0 = -Y^0 \bmod L$. Equation (1) will then have the form:

$$m\hat{\delta} < (Y + \overline{Y}^0) \bmod L < L - m\left|\check{\delta}\right| \qquad (2)$$

Example 1Let us consider the 8-bit ADC, whose offset has changed from 0 to +2 (FSR/256) due to a failure. The ADC is fed by the five test stimuli, x_1^0= 201/256 FSR, x_2^0= 202/256 FSR, x_3^0= 203/256 FSR, x_4^0= 204/256 FSR, x_5^0= 205/256 FSR. The actual readings of the ADC are accordingly: y_1^0= 203, y_2^0= 203, y_3^0= 205, y_4^0= 207, y_5^0= 206. Here m=5, n=8, $\hat{\delta} = \left|\check{\delta}\right| = 1$, Y =1024; Y^o=1015; $\overline{Y}^0$=9, $\check{\Delta}$ =5, $\hat{\Delta}$ =251. And condition (2), 5 <(1024 + 9) mod 256 <251, is satisfied. Therefore, the failure is detected. If the offset were 0, then *Y* would have been 1014. And condition (2) would not hold: 5 <255 > 251.

IV. Aliasing

Aliasing occurs when the signature of a faulty circuit matches the signature of a fault-free circuit. The aliasing rate

for an ADC can be estimated as the ratio of the number of all undetectable errors in the output response of the ADC, to the number of all possible errors in that response.

Let us first estimate the aliasing rate for the ideal ADC. The output response stream consists of $m \times n$ bits that are to be compacted into n bits. The number of faulty streams that will produce the fault-free circuit signature and, therefore, will not be detected is $(2^{mn}/ 2^n) - 1 = 2^{(m-1)n} - 1$. Since there are a total of $2^{mn} - 1$ erroneous streams, the aliasing rate will be $P_{ADC/idl} = (2^{(m-1)n} - 1) / (2^{mn} - 1)$. For many practical cases, $P_{ADC/idl} \approx 2^{-n}$. If the ADC is replaced by a purely digital circuit, this estimate remains true. Therefore, the aliasing rate for the digital system being tested by modulo sum method is $P_{dgt} = P_{ADC/idl} \approx 2^{-n}$.

For a real ADC, the number of faulty streams that will producethe "correct" signatures becomes

$$P_{ADC} = (2^{mn} / 2^n) - (|\bar{\delta}| + \hat{\delta} + 1)^m \cdot$$

And because there are a total of $P_{ADC} = 2^{mn} - (|\bar{\delta}| + \hat{\delta} + 1)^m$ erroneous streams,the aliasing rate now becomes

$$P_{ADC} = \frac{2^{(m-1)n} - (|\bar{\delta}| + \hat{\delta} + 1)^m}{2^{mn} - (|\bar{\delta}| + \hat{\delta} + 1)^m} \quad (3)$$

Under certain conditions, we can obtain $P_{ADC} \approx 2^n$.

Example 2 For the ADC considered in Example 1, equation (3) yields $P_{ADC} \approx 0.0039$.

For an arbitrary choice of m and n, equations (2) and (3) will have the following forms:

$$m < (Y + \bar{Y}^0) \bmod L < -m \bmod L \quad (4)$$

$$P_{ADC} = \frac{2^{(m-1)n} - 3^m}{2^{mn} - 3^m} \quad (5)$$

If the input of an ADC is fed by the stimuli matching the ideal transitions of the transfer characteristic, then expressions (2) and (3) are simplified to:

$$0 < (Y + \bar{Y}^0) \bmod L < -m \bmod L \quad (6)$$

$$P_{ADC} = \frac{2^{mn-m-n} - 1}{2^{mn-m} - 1} \quad (7)$$

And if $mn >> (m + n)$, then $P_{ADC} \approx 2^{-n}$.

By comparing (5) and (7) we can observe that for practical values of m and n, the aliasing rates for these two cases are almost the same and equal to 2^{-n}. The aliasing rate decreases with the growth of the resolution of an ADC. As an alternative, the size of the modulo adder can be increased, if the resolution cannot be raised further. It can also be noted that under these conditions the aliasing rate does not change with the further increase of m. This estimate is only accurate if errors in the output stream are equally likely.

V. Conclusion

We considered a compaction method that can be used for concurrent testing of analog-to-digital converters. The method involves feeding the ADC with operationalinput signals and evaluating the result of compaction of output responses, referred to as a signature. If the signature does not fall into the predefined interval, the ADC is considered to be faulty. The tolerance bounds for the signature of the fault-free ADC are evaluated. It is shown that these bounds depend on the input stimuli. The aliasing rate is estimated. Two sets of the input stimuli are examined. It is demonstrated that under an independent error model, the aliasing rate for these two sets is equivalent, and it does not noticeably change with the increase of the number of input stimuli. However, it does change when the resolution of the ADC being tested (or the length of the signature) is altered. Namely, the aliasing rate is reduced with the growth of the resolution.

In the case of a direct-conversion ADC with an intermediate conversion of the measured electrical value into time, implementation of the method is fairly simple. The binary counter used in such an ADC is utilized as a signature compactor. In the testing mode, it is reset only after a series of conversions for the entire sequence of test stimuli.

In order to further increase the sensitivity of a signature to special types of errors in the ADC, we can select the compaction modulo in the form of $L_p = 2^n - 1$. This compactor will then detect all single errors [15].

Practical implementation of the method is facilitated in the systems measuring frequency dependant parameters (such as impedance). In these systems the test stimuli can be obtained by deviation of the frequency of the current that feeds the impedance being measured. This will significantly lower test hardware overhead, althoughthe correlation rate between failures may increase.

[1] Actel Corporation, *SmartFusion Intelligent Mixed-Signal FPGAs: Innovative Intelligent Integration,* 2010; www. actel. com/ FPGA/ Smart Fusion

[2] I.Voyiatzis, A. Paschalis, D.Gizopoulos,N.Kranitis, and C.Halatsis, "A Concurrent BIST Architecture Based on a Self-Testing RAM", *IEEE Transactions on Reliability,* vol.54, No.1, 2005, pp. 69-78.

[3] C. Stroude, J. Morton, T. Islam and H. Alassaly, "A mixed-signal built-in self-test approach for analog circuits", *Southwest Symposium on Mixed-Signal Design*, Las Vegas, USA, 2003, pp. 196-201.

[4] R. Frohwerk, "Signature Analysis: A New Digital Field Service Method",*Hewlett Packard J.,* vol. 28, No 9, May 1977, pp. 2-8.

[5] G. Starr, J. Qin, B. Dutton, C. Stroud, F. Dai and V. Nelson "Automated generation of built-in self-test and measurement circuitry for mixed-signal circuits and systems," *Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems,* 2009, pp. 11-19.

[6] R. Sharma and K. Saluja, "Theory, analysis and implementation of an on-line BIST technique*," VLSI Design,* vol. 1, no 1, 1993, pp. 9-22.

[7] I. Voyiatzis, A. Paschalis, D. Gizopoulos, C. Halatsis, F. Makri, and M. Hatzimihail, "An input vector monitoring concurrent BIST architecture based on a precomputed test set*," IEEE Trans. Computers,* vol. 57, no. 8, 2008, pp. 1012-1022.

[8] I. Voyiatzis, D. Gizopoulos, and A. Paschalis, "An input vector monitoring concurrent BIST scheme exploiting "X" values," *15th IEEE Int. On-Line Testing Symposium,* 2009, pp. 206-207.

[9] Y.-S. Wang, J.-X.Wang, F.-C.Lai and Y.-Z. Ye, *"A low-cost BIST scheme for ADC testing",* Shanghai, China, *VI Int. Conf. on ASIC,* vol. 2, 2005, pp. 665-668.

[10] D. Lee, K. Yoo, K. Kim, G. Han and S. Kang, "Code-width testing-based compact ADC BIST circuit", *IEEE Trans. on Circuits and Systems-II: Express briefs,* vol. 51, No 11, 2004, pp. 603-606.

[11] G. Wu, J. Rao, A. Ren and M. Ling, "Implementation of a BIST scheme for ADC test", *V Int. Conf. on ASIC,* Beijing, China, vol. 2, 2003, pp. 1128-1131.

[12] A. Gookin, "A fast reading high resolution voltmeter that calibrates itself automatically", *Hewlett Packard J.,* vol. 28, No 6, 1977, pp. 2-10.

[13] V. Kneller, "Measurement, control and other processes: to the problem of knowledge systematization",Cavtat-Dubrovnik, Croatia, *XVII IMEKO World Congress, 2003,* pp. 1119-1124.

[14] G. D'Antona and A. Ferrero, *Digital Signal Processing for Measurement Systems,* Springer Science + Business Media Inc., Berlin, 2006.

[15] J. Wakerly, *Error Detecting Codes, Self-Checking Circuits and Applications,* Elsevier North-Holland, New York, 1978.

[16] V. Geurkov, L. Kirischian, and V. Kirischian "Signature Testing of Analog-to-Digital Converters". *Proceedings of the 19th IMEKO World Congress: Fundamental and Applied Metrology,* Lisbon,