Implementation of a Linear Histogram BIST for ADCs

F. Azaïs, S. Bernard, Y. Bertrand, M. Renovell
LIRMM - University of Montpellier II
161, rue Ada - 34 392 Montpellier - France

Abstract

This paper validates a linear histogram BIST scheme for ADC testing. This scheme uses a time decomposition technique in order to minimize the required hardware circuitry. A practical implementation is described and the structure together with the operating mode of the different modules are detailed. Through this practical implementation, the performances and limitations of the proposed scheme are evaluated both in terms of additional circuitry and test time.

1. Introduction

Design for Test (DfT) and Built-In Self-Test (BIST) for analogue and mixed signal circuits have received the growing attention of industry and research community in order to alleviate increasing test difficulties. Testability is viewed now as a design specification and has to be considered in the early design stages. In addition to improved manufacturing testability, BIST offers a possibility of in-field verification and test. A number of papers concerned with the definition of BIST techniques for analog and mixed-signal ICs can be found in the literature [1-8]. Most of the proposed technique address devices that include both a ADC and a DAC [1-3], or rely on the use of DSP capabilities to compute the characteristic parameters of converters [4-6]. Concerning mixed-signal ICs including solely an ADC, only a limited number of BIST techniques have been proposed based on the reconfiguration of the circuit in test mode in order to create oscillation [7] or on the monitoring of the LSB in order to determine the converter linearity [8].

One of the most popular technique used for external testing of ADCs is the histogram test technique [9]. It is based on a statistical analysis of how many times each digital code word appears on the ADC output in order to determine the ADC characteristic parameters. The on-chip implementation of this technique is generally not considered as a viable solution because of the huge amount of required additional circuitry. The authors have recently proposed the concept of time decomposition in order to reduce the corresponding additional circuitry. An original BIST scheme for linear histogram testing has been defined and a preliminary high level evaluation have shown the potential of the technique [10,11]. The objective of this paper is to further validate the time decomposition concept and evaluate the performances of the BIST scheme through a practical implementation.

2. ADC BIST scheme

2.1. Histogram-based BIST

A very classical technique used in the industrial context to determine the ADC parameters is the histogram test technique. It involves the application of an analog signal on the converter input and the record of the number of time each code appears on the converter output. These recorded samples are then compared with theoretical samples and comparison results are processed in order to determine the ADC parameters, namely offset, gain, differential and integral non-linearity. The analog input signal can be any wave whose amplitude distribution is known. Figure 1 illustrates the histograms obtained using either a triangle-wave or a sine-wave signal for an ideal ADC.

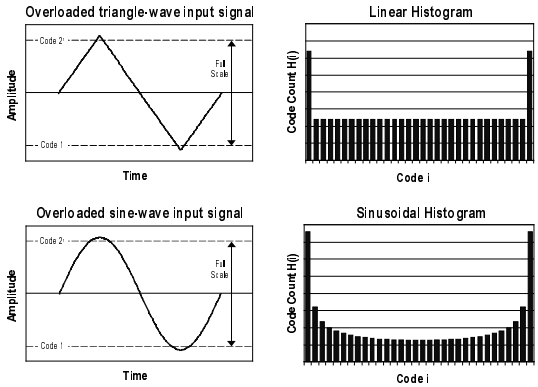

Figure 1. Histograms for an ideal ADC

From a general point of view, a complete BIST scheme for ADCs requires the definition of an analog input generator and a digital output response analyzer. On-chip generation for mixed-signal devices is a classical problem for which solutions can be found in the literature [12]. Consequently, we focus in this paper on the problem of defining the digital analyzer able to implement the histogram test technique.

The straightforward implementation of the histogram test technique requires a number of hardware resources, both in terms of memory and operative resources. Indeed, the determination of the ADC parameters is based on a comparison between the measured and ideal histograms together with subsequent calculations. It is therefore necessary to store both the measured and ideal histograms, implying two memories of 2^n words for an n-bit converter. Then, complex computations have to be performed on these data to extract the parameters, implying the use of a DSP or a microprocessor. Finally, the complete process is managed by a control unit. Figure 2 summarizes these different on-chip resources. It is clear that, unless memory and DSP capabilities are already available on-chip, such a direct implementation of the histogram test technique is not viable because of the huge amount of additional circuitry.

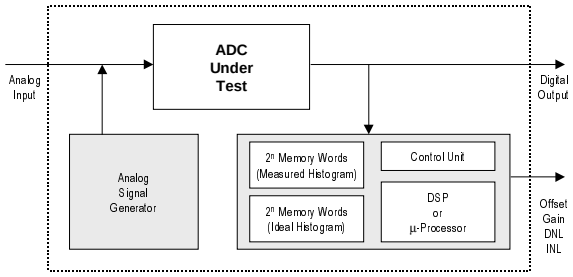

Figure 2. General BIST scheme

2.2. BIST resource minimization

This section summarizes the optimization concepts previously proposed by the authors in order to reduce the area overhead [13].

Classically, either a sine or a triangle-wave is used as input signal to build up the histogram. Any of these two techniques may be used to extract the ADC parameters. However, the linear histogram technique presents a very interesting feature concerning memory saving for the storage of the ideal histogram. Indeed, because the output code count is constant for every code in the converter (expect for the extreme codes if an overloaded input signal is used), it is not necessary to use 2^n memory words. In fact, the complete histogram can be represented with only two values, one corresponding to the ideal count of any non-extreme code (H_{ideal}), and one corresponding to the ideal count of the extreme codes ($H_{extreme}$). As a result, the memory for the storage of the ideal histogram reduces from 2^n memory words down to 2 memory words. A first minimization of the required hardware resources is then achieved by choosing the linear histogram rather than the sinusoidal histogram.

Another advantage of the linear histogram technique concerns the calculation of the ADC parameters. Indeed, the exploitation of a sinusoidal histogram is much more difficult, due to the non-uniform distribution caused by the input signal. The parameter computation actually involves a rather complex trigonometric procedure. When using a triangle-wave input signal, every code in the converter should exhibit an equal density. Because of this uniform distribution, it is possible to derive simple expressions for the ADC parameters. Details on these computations are given above.

Offset computation. The offset (in LSB) is proportional to the difference between the measured counts for the two extreme codes H(1) and $H(2^n)$:

$$\text{Offset} = \frac{H(2^n) - H(1)}{2 \cdot H_{ideal}} \qquad \text{(Eq.1)}$$

Gain computation. The gain (in LSB) is simply given by the ratio between the measured count for any non-extreme code H(i) and the ideal count H_{ideal}:

$$\text{Gain} = \frac{H_{ideal}}{H(i)} \qquad \text{(Eq.2)}$$

However, for real measurements, the count may varies from a code to another due to regularity defects in the sample distribution. It is consequently reasonable to average the measure on several codes. Considering m codes around the center code, we obtain the following expression for the ADC gain:

$$\text{Gain}^{-1} = \frac{\sum_{i=N1}^{N2} H(i)}{m \cdot H_{ideal}} \qquad \text{(Eq.3)}$$

DNL computation. The differential non-linearity (in LSB) of a given code i is defined as the relative difference between the measured and ideal counts:

$$\text{DNL}(i) = \frac{H(i) - H_{ideal}}{H_{ideal}} \qquad \text{(Eq.4)}$$

INL computation. The integral non-linearity of a given code i is then expressed as the cumulative sum of the DNL of all preceding codes:

$$\text{INL}(i) = \sum_{j=1}^{i} \text{DNL}(i) \qquad \text{(Eq.5)}$$

These expressions allow the determination of the ADC parameters using only elementary operations such as addition, subtraction, multiplication or division. As a result, the DSP required in the general case can be replaced by a much simpler operative unit. So, choosing the linear histogram rather than the sinusoidal one also permits to minimize hardware operative resources.

Finally, the last minimization concerns the memory required for the storage of the measured histogram. The fundamental idea is to modify the histogram test procedure in a code-after-code test procedure so that only a limited number of storage elements are required. In other words, we propose to concurrently store and process the histogram code-after-code. Using such a time decomposition technique, a unique memory word can be used for the storage of the measured histogram.

The time decomposition actually takes place at two levels:

- At high level, the concurrent calculation of the ADC parameters is replaced by a phase-after-phase

procedure in which each parameter (offset, gain, DNL and INL) is determined sequentially.

- At low level, each test phase is decomposed in several steps, each individual step requiring only the storage of 1 code count.

Using this approach, all the resources used in a given step of the test procedure are liberate for the following step and thus, can be reused. Note that this applies for both the operative and memory resources. So, not only the time decomposition permits to minimize the memory required for the storage of the measured histogram, but also the operative unit implementing the ADC parameter calculation.

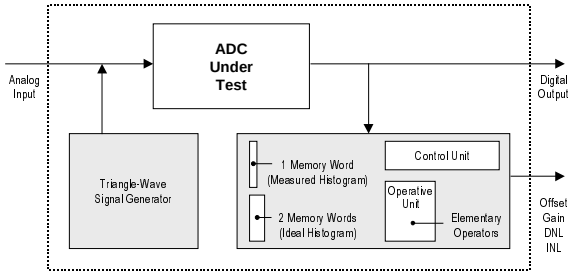

Figure 3. Linear histogram BIST structure

To sum up, it appears that combining the linear histogram with an optimized time decomposition method permits to drastically reduce the hardware resources. Indeed, the BIST structure now comprises only 2 memory words for the storage of the ideal histogram, 1 memory word for the sequential storage of the measured histogram, a simple operative unit with elementary operators to extract the ADC parameters and a control unit to manage the test procedure. Figure 3 illustrates this BIST architecture.

3. ADC BIST implementation

The previous section has introduced general concepts for minimizing the BIST circuitry and the high-level architecture of the digital analyzer. To further validate our approach, it is now necessary to consider the low-level implementation. Of course the idea is still to optimize as much as possible the resources during this hardware implementation phase.

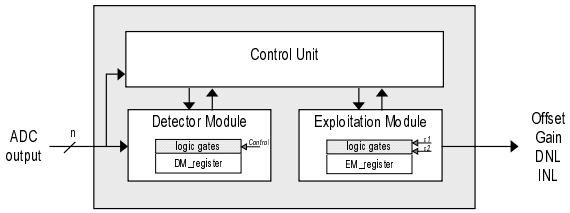

Figure 4. Block-level implementation

Figure 4 gives the block diagram of the optimized digital analyzer. This structure is composed of (i) the Detector Module (DM) which performs ADC output code detection, (ii) the Exploitation module (EM), which performs the calculations required for the test procedure and (iii) the Control Unit (CU), which manages the test process. Note that this structure differs from the general architecture presented in figure 3 because memory and operative capabilities are now combined together in the Detector and Exploitation Modules for optimization purpose.

3.1. Detector module

The Detector Module (DM) is designed to implement two different functions. First, it positions the reference code to be processed. Then, it compares this reference code with the running code delivered on the output of the ADC. So, this module contains a register (DM_register) together with a configuration logic. Depending on the Control signal, the register is configured in either a counter or a comparator. Code selection is achieved in the counter mode by incrementing the counter until the reference code is reached; then code comparison is performed in the comparator mode. The schematic description of such a DM is given in figure 5.

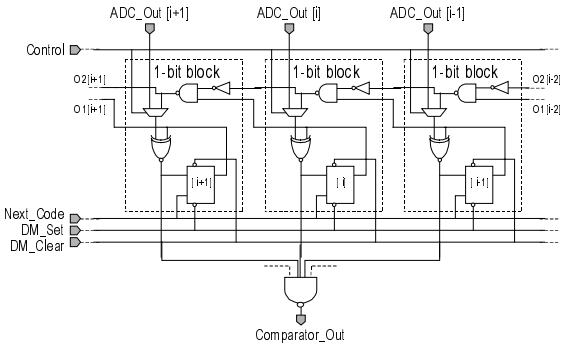

Figure 5. Detector module schematic

3.2. Exploitation module

The Exploitation Module (EM) is designed to calculate the ADC parameters. The schematic description of this module is given in figure 6. As for the DM, the module is composed of a register (EM_register) together with a configuration logic controlled by the two signals c1 and c2. Depending on the values applied on these control inputs, the module is configured in different operating modes that permit to perform the calculations required for the ADC parameters. Basically, the module can either operate as an up/down counter to realize addition and subtraction or provide the 2's complement of the value stored in the register to manage the sign of results. More details on the different operating modes and how they are used to compute the ADC parameters will be given below.

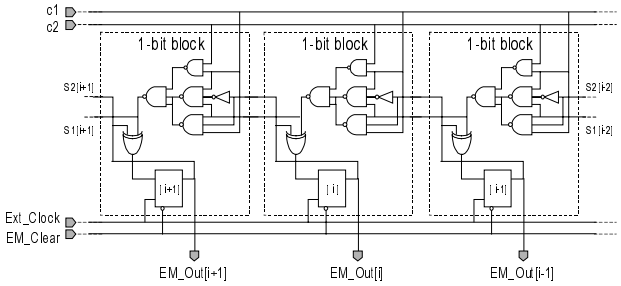

Figure 6. Exploitation module schematic

Offset evaluation. As defined in equation 1, the offset is determined through the counts of the two extreme codes. The EM has actually to perform the subtraction of these two counts. In practice, this operation is carried out by configuring the block into an up/down counter, which is programmed to count down when the output code is 00...0 and to count up when the code is 11...1 during one application of the input test pattern. So, at the end of the pattern, the EM_register directly holds the difference between the counts of the two extreme codes, and it is then necessary to divide the result by $2.H_{ideal}$ to obtain the offset value. Note that such a division is relatively heavy to implement in the general case, but can be easily carried out if H_{ideal} is a power of 2. Indeed, choosing $H_{ideal} = 2^P$, the division is simply equivalent to a (P+1)-bit shift in the register. Finally, the Control Unit verifies the sign of the result and the offset value is delivered on EM_Out as the content of the register or its 2's complement in case of a negative value.

The pseudo algorithm given below summarizes the offset computation procedure.

```
initialization of DM_register = 0, EM_register = 0
for number_of_samples = 1 to N_T
        if (ADC_output_LSB = 1) then DM_register = 11...1
                if (ADC_running_code = DM_register) then EM counts up
        else DM_register = 00...0
                if (ADC_running_code = DM_register) then EM counts down
Shift EM_register (P+1)-bits left
if (negative result) then EM_Out = 2's complement (EM_register)
else EM_Out = EM_register
```

Gain evaluation. As defined in equation 3, the gain is determined through the counts of the m central codes. In order to perform the accumulation of these code counts, the EM is configured as an up-counter and m input test patterns are applied. For the first input test pattern, code N1 is positioned as the reference code in the DM_register and the EM_register is incremented each time the running code occurs on the ADC output. This operation is repeated m times incrementing the reference code until code N2. So, by the end of the m input test patterns, the EM_register holds the cumulative sum of the counts for the m central codes. To obtain the $Gain^{-1}$ value, this cumulative sum has to be divided by $m.H_{ideal}$. As for the offset calculation, this division simply corresponds to a shift in the register if $m.H_{ideal}$ is a power of 2. Then, we impose $H_{ideal}=2^P$ and $m=2^Z$, and the division by 2^{P+Z} is equivalent to a (P+Z)-bit shift.

The pseudo algorithm given below summarizes the gain computation procedure.

```
initialization of DM_register = N1, EM_register = 0
while (DM-register < N2)
        for number_of_samples = 1 to N_T
                if (ADC_running_code = DM_register) then EM counts up
        DM_register = DM_register + 1
Shift EM_register (P+Z)-bits left
EM_Out = EM_register
```

DNL & INL evaluation. The DNL has to be determined for each one of the 2^n converter codes as defined in equation 4. Therefore, 2^n input test patterns are applied, each one dedicated to a given code i. The calculation procedure simply consists in positioning the reference code i in the DM_register and counting up in the EM_register how many times this code appears on the ADC output. Assuming that H_{ideal} is a power of 2 ($H_{ideal}=2^P$), the result of the division is then available in the EM_register considering a P-bit shift. If this value exceeds 1, we have a positive DNL value and the P less significant bits directly correspond to the DNL decimals. Otherwise, we have a negative DNL value and the 2's complement of the register is performed before outputting the result on EM_Out.

The pseudo algorithm given below summarizes the DNL computation procedure.

```
initialization of DM_register = 0
while (DM-register < 2^n)
        initialization of EM_register = 0
        for number_of_samples = 1 to N_T
                if (ADC_running_code = DM_register) then EM counts up
        DM_register = DM_register + 1
         Shift EM_register P-bits left
        if (negative result) then EM_Out = 2's complement (EM_register)
        else EM_Out = EM_register
```

The INL of a given code is calculated as the cumulative sum of the DNL for all the preceding codes, as defined in equation 6. So the calculation procedure is basically the same than for DNL, but without initializing the EM_register between successive input test patterns.

3.3. Control unit

The Control Unit (CU) is the module that manages the test process. So basically, this module generates the different signals to clock the successive codes and control the DM and EM configurations. This unit can be simply defined in VHDL and then synthesized by an automatic tool such as Synopsis.

4. Performances and discussion

4.1. Area overhead

To illustrate our solution, we propose to derive the BIST structure for testing a 6-bit ADC working at 100MHz frequency rate. In the AMS 0.8µ library, this ADC presents an area of 3.3mm².

First, we build the Detector Module. For a 6-bit converter, we cascade 6 one-bit blocks (see figure 5) and we obtain a module of 0.037mm² area.

Then, we build the Exploitation Module. As for the DM, we cascade a number of one-bit blocks. However, the length of the EM_register does not depends on the number of bits of the converter, but is determined by the calculation procedure. Indeed, we have seen that the offset determination requires a (P+1)-bit shift in the register, the gain determination requires a (P+Z)-bit shift and the non-linearity determination requires a P-bit shift. As a result, the length of the register has to be at least P+Z+1 bits, where P is determined by the ideal count ($H_{ideal}=2^P$) and Z by the number of codes on which the gain measurement is performed ($m=2^Z$).

The choice of the ideal count value H_{ideal} actually depends on the desired accuracy on the measurements. For instance, we can use the desired accuracy on the DNL measurement to determine this value. Starting from the DNL expression, the measurement accuracy is defined as:

$$\delta DNL(i) = \frac{\delta H(i)}{H_{ideal}} \qquad \text{(Eq.6)}$$

where δH(i) is the error for code i.
In practice, H(i) is necessarily an integer, thus the maximum error is equal to 1. Consequently, the measurement accuracy can be evaluated by:

$$\delta DNL = \frac{1}{H_{ideal}} \qquad \text{(Eq.7)}$$

Hence, imposing an accuracy better than 0.05 LSB on the DNL measurement corresponds to choose H_{ideal} higher than 20. Taking into account that the ideal count must be a power of 2, we obtain P=5, which corresponds to $H_{ideal}=2^5=32$ and the accuracy on DNL is now 0.03 LSB.

Then, we have to choose the number of central codes on which the gain measurement is performed. Of course, the higher this number, the better the accuracy, but the higher the area overhead. So, we empirically choose to take ¼ of the total number of codes, i.e. $m=2^4=16$ codes among 64, which corresponds to Z=4.

Finally, the Exploitation Module is built as a cascade of P+Z+1=10 one-bit blocks (see figure 6), which gives a module of 0.076mm² area.

The complete BIST structure comprises the Detector Module and the Exploitation Module together with the Control Unit that has been synthesized using the Synopsis automatic synthesis tool. The layout of the structure implemented in 0.8µm AMS technology is illustrated in figure 7. This BIST structure represents an area of 0.223mm², which corresponds to an area overhead of 6.7% when compared to the original ADC.

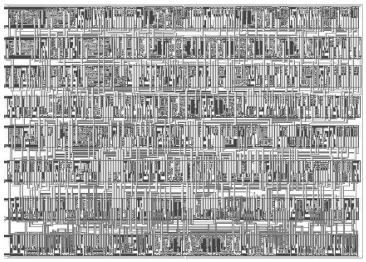

Figure 7. Optimized BIST structure layout

4.2. Test time

An important point to discuss is the impact of our technique on the test time. Indeed, our technique is based on a sequential decomposition of the global test procedure, implying that a high number of input test patterns are required to complete the test. The time decomposition permits to drastically reduce the additional circuitry, but it is clear that this reduction is obtained to the prejudice of the testing time.

Let us evaluate the test time for a n-bit converter. This time of course depends on the number of required test patterns, but also on the length of the test pattern.

So first, we estimate the number of required test patterns as 1 pattern for offset evaluation, m pattern for gain evaluation, 2^n patterns for DNL evaluation and 2^n patterns for INL evaluation. Considering the gain measurement is chosen to be performed on ¼ of the total number of codes and neglecting the single input pattern required for the offset determination, we obtain a rough estimation of the number of required input test patterns:

$$N_{pattern} \approx 2.25 * 2^n \qquad \text{(Eq.8)}$$

Then we estimate the length of a test pattern, which corresponds to the time needed to collect N_T samples at a sampling frequency rate F_s.

$$T_{pattern} = \frac{N_T}{F_s} \qquad \text{(Eq.9)}$$

From the expression given in section 4.1, and neglecting the contribution of overload ($A_{in}/A_{FS}=1$), the number of samples is given by $N_T \cong H_{ideal}.2^n$, and the length of a test pattern can be expressed as:

$$T_{pattern} \approx \frac{32}{F_s} * 2^n \qquad \text{(Eq.10)}$$

So finally, an estimate of the test time can be defined as:

$$TEST\ TIME = N_{pattern} * T_{pattern} \approx \frac{72}{F_s} * 2^{2n} \qquad \text{(Eq.11)}$$

It clearly appears on this expression that testing time considerably increases with the number of bits of the converter. For illustration, table 1 reports test time results for converters from 6 to 14bits, taking into account a sampling frequency rate varying from 1MHz to 100MHz.

Number of bits	TEST TIME				
n	F_s=1MHz	F_s=10MHz	F_s=20MHz	F_s=50MHz	F_s=100MHz
6	300ms	30ms	15ms	6ms	3ms
8	5s	500ms	250ms	100ms	50ms
10	1mn 15s	7.5s	3.8s	1.5s	750ms
12	20mn	2mn	1mn	24s	12s
14	5h 22mn	32mn	16mn	6mn	3mn

Table 1: ADC test time

The first comment on these results is that our technique produces large test times for high-resolution converters: testing a 14-bit converter requires several minutes even with a sampling frequency as high as 100MHz. Hence, the technique should be limited for testing 6 to 12-bit converters. An other comment is that the test time may be still considered as too important for low-frequency converters. However possible optimizations may be considered to reduce testing time. The first evident one consists in computing DNL and INL in the same test phase. Indeed, this permits to divide the test time by almost a factor 2 and only costs an additional register in the BIST circuitry. A second optimization of the technique consists in replacing the code-after-code process of the histogram by a p-code after p-code process for instance. Again a factor p can be gained on the test time, but of course implying an increase in the BIST area. In fact, there is a trade-off to define between the area overhead and the test time.

Note the problem of test time limitation is consistent with the problem of the test signal generation. The resolution of the input signal is generally chosen to be 2-bit more than the ADC one. For example, a resolution better than 75 µV is needed for a 14-bit ADC with a full scale of 5V. It is a not a trivial task to design a linear signal generator with such a high resolution. So, the linear histogram testing should be preferred for medium-resolution converters (6 to 12 bits). In this context, our approach permits to integrate the test technique with a reasonable area overhead and an acceptable test time.

5. Conclusions

This paper validates the time decomposition concept proposed to minimize the additional circuitry required to implement a linear histogram BIST technique. A practical implementation is described and the performances and limitations of the proposed scheme are evaluated.

The practical implementation clearly demonstrates the viability of the scheme since the area of the additional circuitry is found to be around 7% of that of the original Analog-to-Digital Converter. This drastic reduction of the BIST circuitry is a consequence of the application of the time decomposition concept but it is clear that this reduction is obtained to the prejudice of the testing time.

The testing time has been evaluated to around 1s for a 10 bits converter with a sampling frequency of 100Mz. This result proves that our linear histogram BIST is perfectly viable for a large range of A-to-D Converters. For larger converters with larger test time, solutions are indicated that can reduce drastically the test time as for example parallel computation of INL and DNL, p-after-p code exploitation…

With a test time of about 1 second, the BIST scheme can internally determine the main ADC parameters: offset error, gain error and non-linearity. The result of the test can simply be directly shifted out or shifted through a boundary scan chain. It is also important to note that the internal histogram exploitation is purely digital making the BIST scheme insensitive to parameters variations.

6. References

[1] M.J. Ohletz, "Hybrid Built In Self Test (HBIST) for Mixed Analog/Digital Integrated Circuits", *Proc. European Test Conference*, pp.307-316, 1991.

[2] N. Nagi, A. Chatterjee, J. Abraham, "A Signature Analyzer for Analog and Mixed-Signal Circuits", *Proc. ICCD*, pp. 84-7, 1994.

[3] S Sunter, N. Nagi, "A Simplified Polynomial–Fitting Algorithm for DAC and ADC BIST", *Proc. International Test Conference*, pp. 389–395, 1997.

[4] E. Teraoca, T. Kengaku, I. Yasui, K. Ishikawa, T. Matsuo, "A Built–In Self–Test for ADC and DAC in a Single–Chip Speech CODEC", *Proc. International Test Conference*, pp. 91–796, 1993.

[5] M.F. Toner and G.W. Roberts, "A BIST Scheme for a SNR, Gain Tracking and Frequency Response Test of a Sigma–Delta ADC", *IEEE Trans. Circuits & Systems II*, Vol. 42, pp. –15, 1995.

[6] M.F. Toner and G.W. Roberts, "A Frequency Response, Harmonic Distortion, and Intermodulation Distortion Test for BIST of a Sigma–Delta ADC", *IEEE Trans. Circuits & Systems II*, Vol. 43, No. 8, pp. 608–613, 1996.

[7] K. Arabi, B. Kaminska, "Efficient and Accurate Testing of Analog–to–Digital Converters Using Oscillation–Test Method", *Proc. European Design & Test Conference*, pp. 48-52, 1997.

[8] R. de Vries, T. Zwemstra, E. Bruls, P. Regtien, "Built-In Self–Test Methodology for A/D Converters", *Proc. European Design & Test Conference* , pp. 353-358, 1997.

[9] M. Mahoney, "DSP–based Testing of Analog and Mixed–Signal Integrated Circuits", *IEEE Computer Society Press*, ISBN 0-8186-0785-8, 1987.

[10] M. Renovell, F.Azaïs. S. Bernard, Y. Bertrand, "Procédé et dispositif de test intégré pour un CAN", *CNRS Fr. Patent 911304*, filed September 9, 1999.

[11] M. Renovell, F.Azaïs. S. Bernard, Y. Bertrand, "Hardware Resource Minimization for a Histogram-based ADC BIST", *Proc. VLSI Test Symposium*, May 2000.

[12] G.W. Roberts, A.K. Lu, "Analog Signal Generation for Built-n Self-Test of Mixed-Signal Integrated Circuits", *Kluwer Academic Publishers*, ISBN 0-7923-9564-6, 1995