Towards an ADC BIST Scheme using the Histogram Test Technique

F. Azaïs, S. Bernard, Y. Bertrand and M. Renovell
LIRMM - University of Montpellier
161, rue Ada, 34392 Montpellier, France

Abstract

This paper discusses the viability of a BIST implementation for the sinusoidal histogram technique classically used for ADC testing. An original approach based on (i) approximations to estimate the ADC parameters, (ii) decomposition of the global test in a code-after-code test procedure and (iii) piece-wise approximation to compute the ideal histogram is developed. These three features allow a significant reduction of the required operative resources as well as the required memory resources dedicated to the storage of both experimental and reference data.

1. Introduction

With the advances on analog-digital ICs, faster and more complex test equipment is required to meet ever more severe test specifications. An attractive alternative to simplify the test equipment is to move some or all the tester functions onto the chip itself. The use of Built-In Self-Test (BIST) for high volume production of mixed signal ICs is desirable to reduce the cost per chip during production testing by the manufacturers.

Many of the proposed BIST techniques for mixed signal ICs address devices that include both a ADC and a DAC [1-4], or use DSP capabilities to compute the characteristic parameters of converters [5-7]. It is clear that the viability of these techniques depends on the presence in the original circuit of both ADC and DAC, or the presence of a DSP.

Focusing now on mixed signal ICs including solely an ADC, only a limited number of BIST techniques have been proposed. An original approach is detailed in [8], which relies on a reconfiguration in test mode that creates oscillation in the circuit. Measurements on these oscillations then guarantee some tests. A BIST structure is proposed in [9] that permits to evaluate the converter linearity. Only the LSB is used for the determination of the linearity, the global functionality of the converter being tested with the comparison between the remaining bits and a counter clocked by the LSB.

A very classical ADC test technique used to determine the ADC characteristic parameters is the histogram method [10-11] It involves the application of an analog signal to the ADC input and the record of the number of time each code appears on the ADC outputs. These recorded samples are then used with theoretical samples in a complex computation to determine the ADC parameters, namely offset, gain, differential and integral non-linearity.

The histogram technique is widely used for the external testing. Howver, a histogram-based BIST technique with complete on-chip determination of the ADC parameters is generally not considered as a viable solution because of the huge amount of required additional circuitry. The authors have already proposed a BIST scheme for implementing the histogram technique in case of a triangle-wave input test signal [12,13]. The proposed scheme takes advantage of the intrinsic properties of this input signal to reduce the circuitry of the ADC output analyzer. This paper investigates the viability of a histogram-based BIST approach in case of a sine-wave input test signal. Note that only the possibilities of on-chip output data processing are explored in this research, existing solutions for on-chip test signal generation being detailed in [14].

2. Histogram-based BIST of ADCs

2.1. Histogram test technique

The histogram (or code density) method is one of the most popular techniques in the industrial context for ADC testing. Given an analog input signal, the histogram shows how many times each different digital code word appears on the ADC outputs. The analog input signal can be any wave whose amplitude distribution is known. Figure 1 illustrates the histogram obtained with an ideal ADC using a sine wave as input signal.

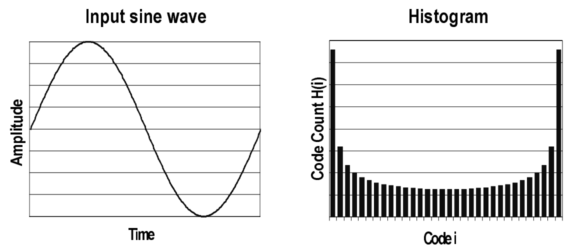

Fig.1: Sinusoidal histogram for an ideal ADC

Of course, ADC errors modify the output code count and so impact the histogram shape. As a result, comparing the measured histogram to the ideal one and performing some calculation permits to evaluate the following ADC parameters: offset, gain, DNL and INL. Details on computations performed to extract these parameters from the histogram data are given in the next section.

2.2. General BIST scheme

From a general point of view, a complete BIST scheme for ADCs requires the definition of an analog sine-wave generator and a digital output response analyzer. Solutions can be found in the literature for on-chip generation of a sine-wave signal [14]. Consequently, we focus here on the problem of defining the digital analyzer able to implement the histogram test technique.

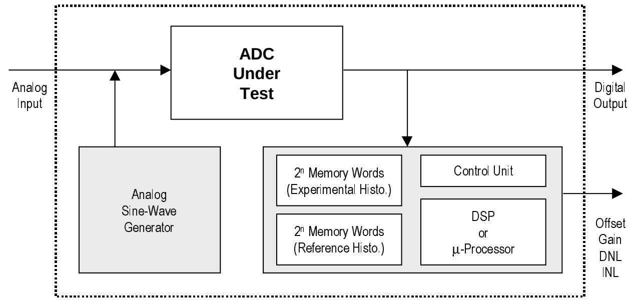

Fig.2: General BIST scheme

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 experimental and reference histograms together with subsequent calculations. It is therefore necessary to store both the experimental and reference 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.

3. BIST hardware resource minimization

3.1. Operative unit

In order to integrate the histogram test technique with a reasonable silicon area, the first point we consider concerns the calculations required to evaluate the ADC parameters. We want to investigate how it is possible to approximate the original complex expressions by simpler ones, which contain only elementary operations. We detail in the following the new expressions we derive for the ADC characteristic parameters.

Offset calculation - The offset is usually determined using the counts for the two extreme codes $H^{exp}(1)$ and $H^{exp}(2^n)$ of the experimental histogram. The original expression of the offset as defined in [11] in given by the following equation:

$$V_{offset} = \frac{A_{full\ scale}}{2} * \frac{\cos\frac{\pi H^{exp}(1)}{N_T} - \cos\frac{\pi H^{exp}(2^n)}{N_T}}{\cos\frac{\pi H^{exp}(1)}{N_T} + \cos\frac{\pi H^{exp}(2^n)}{N_T}}$$

where $A_{full\text{-}scale}$ is to the full scale amplitude of the ADC and N_T is the total number of samples.
Manipulating this equation, we can re-express the offset as:

$$V_{offset} = -\frac{A_{full\ scale}}{2} * \tan\frac{\pi\left(H^{exp}(1) + H^{exp}(2^n)\right)}{2N_T} * \tan\frac{\pi\left(H^{exp}(1) - H^{exp}(2^n)\right)}{2N_T}$$

A first observation concerns the sum of the counts for the two extreme codes $H^{exp}(1)+H^{exp}(2^n)$. We can consider that this sum is almost a constant value even in case of an offset error. Indeed, a positive offset error causes a decrease of $H^{exp}(1)$ and an increase of $H^{exp}(2^n)$, and conversely. These two variations nearly compensate each other when adding the counts for the two extreme codes. For illustration, we build the histogram for an 8-bit converter using 8191 samples with an input sine-wave of 262 LSB peak-to-peak amplitude. We observe that an offset error as high as 1 LSB causes only a 0.7% variation of the sum $H^{exp}(1)+H^{exp}(2^n)$. Consequently, our first approximation is to replace this experimental sum by a constant value derived from the reference histogram $H^{exp}(1)+H^{exp}(2^n) \approx 2H^{ref}(1)$,where $H^{ref}(1)$ is a known-value for a given number of bits n and number of samples N_T.

A second observation is that the difference between the counts for the two extreme codes $H^{exp}(1)-H^{exp}(2^n)$ is very small relative to the total number of samples N_T. Hence, using the common first-order approximation $\tan(\alpha) \approx \alpha$ for small values of α, we obtain :

$$V_{offset} \approx -\frac{A_{full\ scale}}{2} * \tan\frac{\pi H^{ref}(1)}{N_T} * \left(\frac{\pi\left(H^{exp}(1) - H^{exp}(2^n)\right)}{2N_T}\right)$$

Finally, an estimate of the offset can be expressed by:

$$V_{offset} \approx K * \left(H^{exp}(2^n) - H^{exp}(1)\right)$$

where $K = -\frac{\pi \cdot A_{full\ scale}}{4N_T} * \tan\frac{\pi H^{ref}(1)}{N_T}$ is a constant value that can be pre-determined for a given converter under test. We obtain a simple expression in which the offset is proportional to the difference between the counts of the two extreme codes.

Gain calculation - A common technique to derive the gain is to estimate the amplitude of the input sinusoid as seen through the "eyes" of the ADC under test; the gain is then the ratio between this amplitude and the actual value of the input amplitude $A_{input\ sine}$ [11]:

$$Gain = \frac{\frac{A_{full\ scale}}{2} - V_{offset}}{\frac{A_{input\ sine}}{2} * \cos\frac{\pi H^{exp}(2^n)}{N_T}}$$

where $A_{full\ scale}$ and $A_{input\ sine}$ correspond to peak-to-peak amplitudes.

This expression implies calculating a cosine function, which is not trivial to realize on-chip. In addition, it depends on the offset value, implying any error on the offset determination will affect the gain determination. We propose an alternative approach to estimate the gain. Our solution stems from the following observation. The histogram obtained with a sine wave as input presents a relatively flat section in the center part. This flat section actually corresponds to the ADC outputs for the almost linear part of the input sinusoid. Hence, we can make an analogy with the histogram obtained using a ramp as input signal. In this case, all ADC output codes have the same probability and all bins are equal; the ADC gain is therefore simply given by the ratio between the reference and experimental counts for any of these codes [13]:

$$\mathrm{Gain} = \frac{H^{ref}(i)}{H^{exp}(i)}$$

In case of a sine wave as input signal, this relation remains valid as long as the code under consideration is in the linear part of the sinusoid, i.e. i close to $2^n/2$. So it is possible to obtain an estimate of the ADC gain simply using the experimental count $H^{exp}(i)$ for a given code i in the linear section of the sinusoid and the corresponding reference value $H^{ref}(i)$. However for real measurements, the count may vary 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 N codes around the center code $2^n/2$, we obtain the following expression for the ADC gain:

$$\mathrm{Gain} \approx \frac{1}{N} \sum_{i=\frac{2^n-N}{2}+1}^{\frac{2^n+N}{2}} \frac{H^{ref}(i)}{H^{exp}(i)}$$

Of course, the greater the number of codes N considered, the more accurate the gain computation, provided that all N codes remain in the "flat" section of the histogram. For illustration, we build the histogram for an 8-bit converter using 8191 samples with an input sine-wave of 262 LSB peak-to-peak amplitude. We observe that all bins for codes between 92 and 163 present a constant value of 20 to within one sample. It is then possible to average the gain measurement on up to 70 codes.

DNL and INL calculation - The differential non-linearity of a given code i is defined as the relative difference between the experimental count $H^{exp}(i)$ and the corresponding reference one $H^{ref}(i)$ [11]:

$$\mathrm{DNL}(i) = \frac{H^{exp}(i) - H^{ref}(i)}{H^{ref}(i)} = \frac{H^{exp}(i)}{H^{ref}(i)} - 1$$

The integral non-linearity of a given code i is then expressed as the cumulative sum of the DNL of all preceding codes [11]:

$$\mathrm{INL}(i) = \sum_{j=1}^{i} \mathrm{DNL}(j)$$

Both these expressions involve only simple operations. There is consequently no need of modifying these expressions to integrate the calculation.

Operative resource minimization - New expressions have been proposed to determine the ADC characteristic parameters using only simple operators. Let us evaluate more precisely the hardware operative resources required to implement the different calculations. Analyzing the different expressions, it appears that:

- the offset calculation demands a subtracter (omitting the multiplication by a constant),
- the gain calculation demands an adder and a divider,
- the DNL and INL calculation demands an adder, a subtracter and a divider.

An evident minimization of the operative resources consists in replacing the concurrent calculation of all parameters by a phase-after-phase approach in which each parameter is determined sequentially. Indeed in this case, the subtracter required for the offset calculation can be reused in the DNL and INL calculation phase. In the same way, the adder and divider required for the gain calculation can be reused in the DNL and INL calculation phase. So finally, decomposing the global ADC test in a 3-phase procedure and using the simplified parameter expressions, the operative unit only consists of an adder, a subtracter and a divider.

1.2. Experimental histogram memory

Our objective is now to study how it is possible to minimize the memory required for storing the experimental histogram. The fundamental idea is based on the use of a time decomposition technique. Indeed, the previous section has established that it seems judicious to decompose the global test into different phases, each phase dedicated to the calculation of one ADC parameter. This time decomposition allows to reuse the hardware operative resources in the different test phases. The same approach can be adopted inside each test phase in order to minimize the hardware memory resources. The idea is to sequentially store only the code counts needed to compute the ADC parameter for a given test phase. More precisely, we suggest to decompose each test phase in several steps, each individual step requiring only the storage of one code count. In other words, we propose to store and concurrently process the histogram code after code.

Offset determination - The pseudo-algorithm given below details the offset calculation procedure. Two input patterns are necessary to collect the samples for the two extreme codes. During pattern application, register R1 (respectively R2) is incremented each time the output code is 1 (respectively 2^n). By the end of the two patterns, $H^{exp}(1)$ is stored in register R1 and $H^{exp}(2^n)$ in register R2. Consequently, subtracting these registers gives the difference between the counts for the two extreme codes. The offset value is then proportional to R1, with $K = -\frac{\pi \cdot A_{\text{full scale}}}{4N_T} * \tan\frac{\pi H^{ref}(1)}{N_T}$.

```
R1 = 0, R2 = 0
for ns = 1  to  NT
    if   code = 1     then     R1 = R1 + 1
    if   code = 2^n   then     R2 = R2 + 1
    ns = ns + 1
R1 = R2 - R1
Voffset = K . R1
```

Gain determination - The pseudo-algorithm given below details the gain calculation procedure. The gain is computed using the counts for the N center codes. However, computation is done sequentially one code after the other. For a given code i, a complete input pattern is used to collect the samples; N patterns are then necessary to carry out the calculation. So for a given code i, the register R1 is incremented each time the output code is i. By the end of the pattern, $H^{exp}(i)$ is stored in register R1. We assume that the corresponding reference value $H^{ref}(i)$ is available at this time in a dedicated register R^{ref}. We can then divide this register with R1 and store the result in R1.
Finally, for i varying from $\frac{2^n - N}{2}+1$ to $\frac{2^n + N}{2}$ we progressively compute in register R2 the cumulative sum of $H^{ref}(i)/H^{exp}(i)$ (coming from register R1).). The gain value is therefore available in register R2 by the end of the N input patterns.

```
R2 = 0
for i = (2n-N)/2+1  to  (2n+N)/2
    R1 = 0
    for ns = 1  to  NT
        if  code = i  then      R1 = R1 + 1
        ns = ns + 1
    R1 = Rref / R1
    R2 = R1 + R2
    i = i+1
R2 = R2 / N
Gain = R2
```

DNL and INL determination - The pseudo-algorithm given below details the non-linearity calculation procedure. Both DNL and INL have to be evaluated for each of the 2^n codes. Computation is performed code after code, using a complete input pattern for each code. So for a given code i, $H^{exp}(i)$ is accumulated in register R1 during the corresponding pattern. Dividing R1 by the reference value $H^{ref}(i)$ and subtracting 1 gives the DNL value. The INL value is then obtained by progressively accumulating all the DNL values in register R2.

```
R2 = 0
for i = 1  to  2^n
    R1 = 0
    for ns = 1  to  NT
        if  code = i  then      R1 = R1 + 1
        ns = ns + 1
    R1 = R1 / Rref
    R1 = R1 - 1
    DNL(i) = R1
    R2 = R1 + R2
    INL(i) = R2
    i = i+1
```

Memory resource minimization - Concerning memory resources, it clearly appears on the pseudo-algorithms presented above that besides a dedicated register (R^{ref}) for the storage of reference data, only 2 registers (R1 and R2) working with experimental data are sufficient to implement each ADC parameter calculation. Note that the same 2 registers can be used in the different phases because of the global test decomposition procedure. So it finally comes out that using the time decomposition concept, the experimental histogram memory unit only consists of 2 registers.
In addition, it is to remark that because of the sequential calculation procedure, the adder and the subtracter are never used at the same time. It is consequently possible to go further in the minimization process of the operative unit replacing these two operators by a single adder/subtracter.

1.3. Reference histogram memory

The last point to investigate now concerns the minimization of the reference histogram memory unit. Indeed, the direct storage of a histogram for a n-bit converter necessitates 2^n memory words, which is an unacceptable solution in terms of silicon area. A first evident minimization consists in storing only half the data due to the symmetrical property of the reference histogram. However, a memory of $2^n/2$ words still represents a too large silicon area. Hence, we propose an alternative approach that consists in computing on-chip the reference histogram, only one code bin being computed at a time. This approach is of course only valid with the code-after-code test procedure presented in the previous section.

The main difficulty stands in the computation of the reference histogram, which is given by the equation:

$$H^{ref}(i) = \frac{N_T}{\pi}\left\{\sin^{-1}\left[\left(\frac{2i-2^n}{2^n}\right)\cdot\frac{A_{\text{full scale}}}{A_{\text{input sine}}}\right] - \sin^{-1}\left[\left(\frac{2i-2^n-2}{2^n}\right)\cdot\frac{A_{\text{full scale}}}{A_{\text{input sine}}}\right]\right\}$$

The on-chip computation of such a complex function is obviously not compatible with the constraint of a small area overhead. We actually propose to approximate this function by a simpler one. A classical approach consists in determining a polynomial which best fits the set of data points given by the reference histogram. The on-chip computation of the polynomial then provides an estimate of the reference histogram. However, implementing this classical approach necessitates to add a multiplier in the BIST circuitry to perform the polynomial computation. Since our first concern is the minimization of the hardware BIST resources, we choose a slightly different approach. The idea is to realize the approximation with a function that can be computed with the operative resources already available on-chip, instead of using a polynomial. This is then a pragmatic approach. So, taking into account that the operative unit comprises an adder/subtracter and a divider, we propose to estimate the reference histogram with a curve described by:

$$H^{est}(i) = \frac{\alpha}{i+\beta}$$

where α and β are coefficients to be determined to ensure the "best fitting" between the reference (H^{ref}) and estimated (H^{est}) curves. Obviously, it is not possible using such a function to obtain a good fitting on the complete domain, i.e. for i varying from 1 to 2^n. But as already mentioned, only half the histogram can be considered due to the symmetrical property. Even in this case, the precision on the estimated histogram appears non acceptable. Consequently, we suggest to perform a piece-wise approximation defining several domains in half the histogram (see figure 3). For each domain, we can determine the α_k and β_k coefficients of the best fitting curve. The number of domains to consider directly depends on the desired precision for the estimated histogram.

We actually develop a software routine in Matlab that permits to define the different domains and the associated coefficients, for a given maximum error between the reference and estimated curves. As an example, we consider the histogram obtained for an 8-bit converter using 8191 samples and an input sine-wave of 262 LSB peak-to-peak amplitude. We specify a maximum error between the original and estimated curves of 0.05%, and the routine finds out that 6 domains are sufficient to fulfil this requirement.

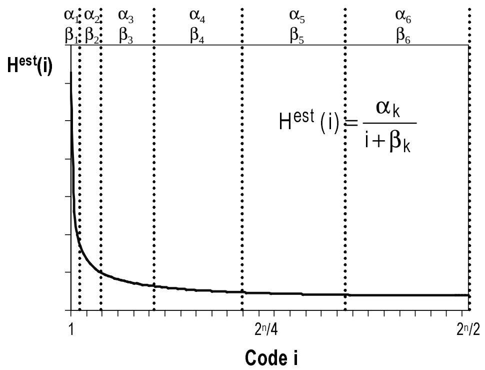

Fig.3: Piece-wise approximation of the reference histogram

Concerning hardware resources, we have seen that all the required operators are already available in the operative unit. It is therefore just necessary to consider memory resources. The implementation of our solution only necessitates 2 registers per domain for the storage of the α_k and β_k coefficients, plus an additional register R^{ref} that contains the result of the computation according to the different steps of the test procedure. For illustration, the reference histogram memory then contains 13 registers in case of an 8-bit converter.

4. Evaluation and discussion

To make the histogram method acceptable in the context of BIST, we have proposed different simplifications permitting to minimize as much as possible the additional circuitry. We finally obtain a structure composed of an adder/subtracter and a divider for the operative unit, 2 registers for the experimental histogram memory and 2k+1 registers for the reference histogram memory, k being the number of domains for the piece-wise approximation. This structure has to be compared to the original one, which comprises a DSP for the operative unit, 2^n memory words for the experimental histogram memory and 2^n memory words for the reference histogram memory. It is clear that we have a drastic reduction of the required hardware resources, then making viable a BIST solution.

In order to validate the structure, we now have to evaluate its performances. We want to compare our implementation of the histogram technique, which comprises a number of simplifications and approximations, to the original implementation with exact formulae. For this purpose, we develop an evaluation program in Matlab that permits to (i) simulate a converter (with or without errors), (ii) build the associated histogram and (iii) extract the ADC parameters using either our approach or the original one.

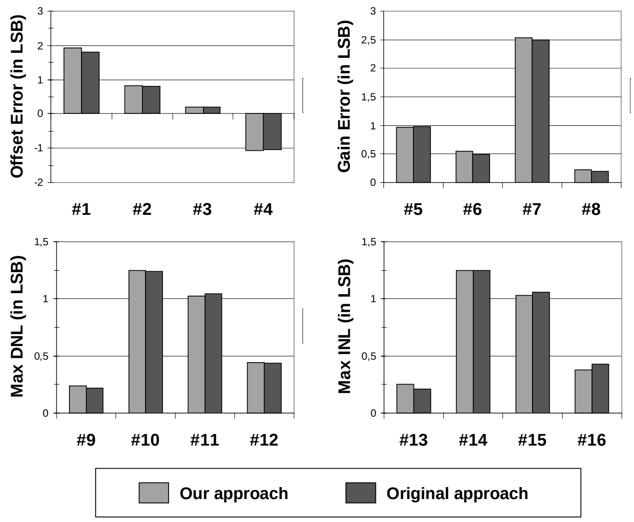

Fig.4: ADC parameter measurement results

As an example in this paper, we consider an 8-bit converter, an input sine wave of 262 LSB peak-to-peak amplitude and N_T=8191 samples to build up the histogram. We also specify a maximum error of 0.05% for the piece-wise approximation of the reference histogram and we choose to compute the gain using the 50 center codes. Then, we run the program on 16 different cases of converter error. Results are summarized in figure 4, which compares for each type of errors (offset, gain, DNL and INL) the measurements obtained using our approach or the original one. For all the simulated cases, we observe a very good agreement between measurements. In fact, it appears that except for the case #1, we obtain a difference between the measurements of less than 0.05 LSB. All these differences are reported in figure 5. Note that the case #1 corresponds to a relatively large offset error (around 2 LSB), which explains the imprecision of our measurement. Indeed, the approximation that replaces the sum of the code counts for the two extreme codes by a constant introduces an error all the more important as the offset is large.

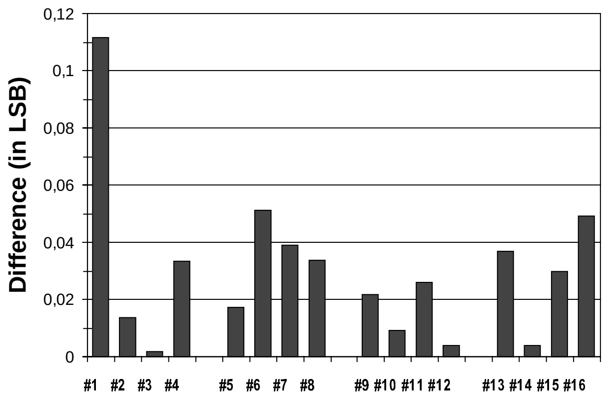

Fig.5: Error on ADC parameter measurements

5. Conclusions

The histogram test method for ADC is usually invoked in the context of external testing because of the large amount of hardware resources required for its implementation; two large RAMs and DSP capabilities. In order to reduce as much as possible this additional circuitry, this paper analyzes in detail the histogram technique and proposes several simplifications.

The first one concerns the calculation of the ADC parameters, which usually requires complex operations. It has been demonstrated that good estimates of offset, gain, DNL and INL can been obtained using operations as simple as addition, subtraction and division. As a result, only one adder/subtracter and one divider are necessary if the global test procedure is decomposed in several phases, each phase dedicated to the evaluation of a given ADC parameter. Indeed, the same operative resources can then be used in the successive different phases.

The second simplification concerns the storage of the experimental histogram. It has been shown that using the time decomposition concept, it is possible to sequentially store only the code counts needed to compute the ADC parameter for a given phase. A code-after-code procedure has been developed that employs only 2 registers for the storage of experimental data.

Finally the last simplification addresses the storage of the reference histogram. An original approach based on a piece-wise approximation has been proposed to reduce the amount of required memory. The idea is to perform the computation of the reference histogram using the operators already available on-chip, only one code being computed at a time. It has been shown that the implementation of this solution necessitates 2 registers per domain of approximation for the storage of the coefficients determining the best fitting curve, plus an additional register that contains the result of computation according to the different steps of the test procedure.

To summarize, this work shows that it is possible to drastically reduce the amount of additional on-chip circuitry required to implement the histogram test technique. Experiments conducted on various examples of faulty converters have validated the approach since comparable results are obtained using either our BIST structure or the original histogram technique. However, the reduction of the additional circuitry is obtained to the prejudice of the testing time. Indeed, the sequential decomposition of the test procedure implies that a high number of input test patterns are required to complete the test. Further investigations will concentrate on this issue.

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] K. Damm, W. Anheier, "HBIST Of Nonlinear Analog Building Blocks In Mixed-Signal Circuits", *Proc. International Mixed Signal Testing Workshop*, pp 257-262, June 1995.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] J. Doernberg, H.S. Lee, D.A. Hodges, "Full–Speed Testing of A/D Converters", *IEEE J. Solid–State Circuits*, Vol. SC–19, No. 6, pp. 820–827, Dec. 1984.

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

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

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

[14] 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.