A Low-Cost BIST Based on Histogram Testing for Analog to Digital Converters

Kicheol KIM†, Youbean KIM†, Incheol KIM†, Hyeonuk SON†, Nonmembers, and Sungho KANG†a), Member

SUMMARY In this letter a histogram-based BIST (Built-In Self-Test) approach for deriving the main characteristic parameters of an ADC (Analog to Digital Converter) such as offset, gain and non-linearities is proposed. The BIST uses a ramp signal as an input signal and two counters as a response analyzer to calculate the derived static parameters. Experimental results show that the proposed method reduces the hardware overhead and testing time while detecting any static faults in an ADC.

key words: ADC testing, BIST, histogram testing

1. Introduction

An ADC is one of the most frequently used analog blocks in an SoC (System on a Chip). As an SoC usually includes ADCs, testing an ADC has become an important issue. The inaccessibility of an ADC in an SoC turns the testing into a more challenging task. The required test cost of an ADC drastically increases because testing an ADC usually requires high quality ATE (Automatic Test Equipment). For these reasons, BIST techniques for the ADC testing which can alleviate test cost and provide accessibility have been proposed [1]–[3]. One of the most popular techniques used for the ADC testing is the histogram BIST [1], [2]. It is based on a statistical analysis of how many times each digital code appears on the ADC output in order to determine the ADC characteristic parameters. This technique is widely used for its precision. However, it needs a huge amount of additional circuitry and testing time to achieve statistically satisfactory results. The time decomposition scheme which reuses hardware resources by calculating ADC parameters sequentially to minimize the extra on-chip hardware is proposed in [1]. In terms of the testing time, however, testing each static parameter sequentially increases the testing time four times. In [2], schemes of BIST reducing the testing time are presented but hardware overhead increases about four times. Whereas previous works just reduce either the testing time or hardware overhead, the proposed BIST method reduces both of them. This letter is organized as follows. Section 2 describes the proposed ADC BIST scheme and Sect. 3 presents the BIST implementation. Experimental results are shown in Sect. 4 to demonstrate the idea. Finally, Sect. 5 provides the conclusion.

Fig. 1 Transfer curves of an ADC and histograms of typical functional errors.
where \( N \) is the total count for all codes, \( n \) is the resolution of an ADC, and \( H(i) \) is the code count for the \( i \)th code. The offset can be simply expressed using the relation: \( H(0)+H(2^n-1)=2H_{\text{ideal}} \) and \( N_f/2^n = H_{\text{ideal}} \) where \( H_{\text{ideal}} \) is the ideal code count for each code:

\[
Offset = \frac{H(0) - H_{\text{ideal}}}{H_{\text{ideal}}} \text{[LSB]}
\]

where \( N_f \) is the total count for all codes, \( n \) is the resolution of an ADC, and \( H(i) \) is the code count for the \( i \)th code. The offset can be simply expressed using the relationship:

\[
Offset = \left( \frac{H(0) - H_{\text{ideal}}}{H_{\text{ideal}}} \right) \text{[LSB]}
\]

ADC gain is the difference between the ideal and actual span of analog input values corresponding to digital output codes. The gain error results in a difference between the ideal transfer curve slope and the ideal slope and a change in average code count. It can be expressed as follows:

\[
Gain = \frac{(N - 1) \times H_{\text{ideal}}}{\sum_{i=1}^{N-1} H(i)}
\]

where \( N \) is the total number of codes \( 2^n \). Assuming that the ADC is faultless, the gain value would be 1. In this case, the following expression would be 0.

\[
\sum_{i=1}^{N-1} (H(i) - H_{\text{ideal}}) = 0
\]

DNL is the difference between each analog increment step and the ideal LSB. It can be calculated by using the difference between the measured count of each code and the ideal code count. INL is the deviation of an actual transfer function from an ideal straight line drawn through the end points. So, the INL error can be defined as the integration of the DNL. DNL and INL can be expressed as follows:

\[
DNL = \frac{H(i) - H_{\text{ideal}}}{H_{\text{ideal}}} \text{[LSB]}
\]

\[
INL = \frac{\sum (H(i) - H_{\text{ideal}})}{H_{\text{ideal}}} \text{[LSB]}
\]

Therefore, all static parameters of an ADC are related to \( H(i)-H_{\text{ideal}} \) or \( \sum (H(i)-H_{\text{ideal}}) \). If there is no fault in an ADC, the values of \( H(i)-H_{\text{ideal}} \) and \( \sum (H(i)-H_{\text{ideal}}) \) would be within the limits determined by specifications of the ADC.

In order to calculate \( H(i)-H_{\text{ideal}} \) or \( \sum (H(i)-H_{\text{ideal}}) \), the BIST architecture which mainly consists of two counters is proposed.

3. BIST Implementation

The overall BIST architecture for the ADC BIST scheme is depicted in Fig. 2.

The analog multiplexer is an ADC’s input signal selector for selecting either an ADC normal input signal in normal mode or the test stimulus in test mode. The ramp signal generator is used for the test stimulus generation. The voltage range of test stimulus should covers and fits the entire full scale range of the ADC under test. The resolution of the generator should be higher than one of the ADC. The analog comparator generates the EOT (End of Test) signal when the input ramp signal reached to \( V_{\text{MAX}} \). The transition detector consists of two D flip-flops and an exclusive OR gate. It monitors the LSB of the ADC’s output codes to detect a transition in the ADC output. The response analyzer consists of two counters and two digital comparators. The up counter and the down counter measure the \( H(i) \) and \( \sum (H(i)-H_{\text{ideal}}, \) respectively. The test precision can be improved by increasing the counter size. The more samples per code result in the more accurate test.

During test mode, the test stimulus of the ramp signal is applied to the ADC through the analog MUX. The test stimulus will be converted into a binary code at the sample frequency of the ADC. When the testing begins, the up counter keeps a record of the code counts. When a transition is detected, the content of the up counter (\( H(i) \)) is compared with the reference limits (roughly \( H_{\text{ideal}} \)) determined by the offset and DNL specifications of the ADC. Once the content is checked, a pass/fail decision is made for offset and DNL errors, and the counter is reset for next code counts measurement.

The gain and INL are determined by successively adding the \( H(i)-H_{\text{ideal}} \) from the first code, up to the code. For convenience, instead of calculating \( \sum (H(i)-H_{\text{ideal}}) \), \( \sum (H_{\text{ideal}}-H(i)) \) is computed. To calculate \( \sum (H_{\text{ideal}}-H(i)) \), the down counter is used. When the testing begins or the output code is changed, only the MSB of the down counter is set to ‘1,’ where ‘10...00’ is equal to \( H_{\text{ideal}} \). \( H_{\text{ideal}} \) can be adjusted by changing the slope of the input ramp. During the test the down counter counts down. When a transition is occurred, the content of the down counter contains the value of \( \sum (H_{\text{ideal}}-H(i)) \). At that time, the contents are compared with the reference limits (roughly zero) given by the gain and INL specifications of the ADC.

4. Experimental Results

The ADC, the analog comparator and the ramp signal generator are designed in SPICE code, and the BIST is designed in Verilog HDL. Mixed-signal simulator Synopsys’ VCS-
Table 1 Hardware overhead and test length comparison for 6-bit ADC.

<table>
<thead>
<tr>
<th>Hardware overhead</th>
<th>Previous work [1]</th>
<th>Previous work [2]</th>
<th>Proposed BIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements overhead</td>
<td>(Comparators, RAM or Counter, Registers, Adder, Subtractior, FSM to control phase)</td>
<td>(Comparators, 4 Resistors, Detector logic, Adder, Subtractor, Multiplier, Divider)</td>
<td>(Comparators, 2 Counters, Transition detector)</td>
</tr>
<tr>
<td>Area @technology</td>
<td>0.223 mm² @0.8 µm AMS</td>
<td>0.0276 mm² @0.25 µm TSMC</td>
<td>0.0119 mm² @0.35 µm SAMSUNG</td>
</tr>
<tr>
<td>Number of gates</td>
<td>About 450</td>
<td>About 1600</td>
<td>About 170</td>
</tr>
<tr>
<td>Testing time</td>
<td>Long (need 4 test phases)</td>
<td>Short (need 1 test phase)</td>
<td>Short (need 1 test phase)</td>
</tr>
</tbody>
</table>

MX is used to simulate the BIST and ADC.

A 6-bit flash ADC is used as a CUT (Circuit under Test). An n-bit flash ADC consists of a resistive divider with $2^n$ resistors which provides the reference voltage and $2^n-1$ comparators. To create each type of errors, ADCs that have an open or a short fault or parameter variations in either the comparator or resistances are generated. In order to verify the proposed BIST circuitry, 72 different flash ADCs are tested. Only 8 flash ADCs are good circuits, the others have soft or hard faults inducing static errors. The decision boundaries for the test are selected according to the CUT’s specifications. The certain limits of offset, gain, DNL and INL errors are ±0.2LSB, ±0.4LSB, ±0.2LSB and ±0.4LSB, respectively. The counters’ size determines the magnitude of detectable error. For 8 bits of test accuracy $(1/2^8)$ the 8 bits up counter and 9 bits down counter are used. The simulation results are shown in Fig. 3. As a result of simulations, the proposed BIST can detect each type of errors (offset, gain, DNL and INL) in all faulty circuits.

Table 1 shows a comparison between previous works and the proposed BIST in terms of hardware overhead and test length. While the previous works use some ALU (Arithmetic Logic Unit) the proposed BIST consists of only two counters without ALU and memories. For the verification and detail comparisons, the BIST is implemented using SAMSUNG 0.35 µm CMOS technology. For a 6-bit ADC, the digital response analyzer of the proposed BIST costs an area of 0.0119 mm² and about 170 gate count. According to the Table 1, hardware overhead reduction is practically achieved. In addition, it only needs only 1 test phase because all static parameters are computed at the same time. The results show that both the hardware overhead and test length are reduced while achieving an efficient detection of static fault in the proposed BIST compared with previous works.

5. Conclusion

An advanced low-cost ADC BIST, which is based on the histogram method, is proposed. Static parameters such as offset, gain and NL are analyzed, and good estimates of them are obtained. The ramp signal as a test stimulus and two counters as a response analyzer are used, so the BIST has reduced the amount of additional on-chip circuitry. Moreover, only one test period is needed to test an ADC, so the testing time also decreases. Experiments conducted on various faulty converters have validated the BIST.

Acknowledgments

This work was supported by the Ministry of Information & Communications, Korea, under the ITRC SP.

References