An on-chip ADC BIST solution and the BIST enabled calibration scheme

Xiankun Jin[1], Tao Chen[2], Mayank Jain[1], Arun Kumar Barman[1], David Kramer[1], Doug Garrity[1], Randall Geiger[2], and Degang Chen[2]

[1]NXP Semiconductors
[2]Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa, USA 50011

***Abstract*—This paper presents a complete on-chip ADC BIST solution based on a segmented stimulus error identification algorithm known as USER-SMILE. By adapting the algorithm for efficient hardware realization, the solution is implemented towards a 1Msps 12-bit SAR ADC on a 28nm CMOS automotive microcontroller. While sufficient test accuracy is demonstrated, the solution is further extended to correct linearity errors of ADC. The entire BIST and calibration circuitry occupies 0.028mm^2 silicon area while enabling more than 10 times tester time reduction and >10dB THD/SFDR performance improvement over an existing structural capacitor-weight-identification calibration scheme. The added die cost is estimated to be 1/8 of the saved test cost from tester time reduction alone.**

I. Introduction

Analog and mixed signal (AMS) circuits are widely used in system(s)-on-chip (SoCs). Among them, the analog to digital converter (ADC) has consistently been one of the world's largest volume mixed-signal products [1]. At the same time, testing ADCs has become ever more challenging in the semiconductor industry. As manufacturing cost goes down, the test cost could become dominant for SOCs that include ADCs. Therefore, there is a need to develop a low-cost way to test the ADC within SoCs. Test cost reduction of SoCs is critical due to the SoC's high volume, large digital content and wide range of applications.

Testing ADCs is challenging due to the stringent requirements on the input stimulus and the long test time [2], [3]. To test an N-bit ADC, the rule of thumb is that the input stimulus has to be at least 2 bits more linear than the ADC under test and there are $2^N - 1$ code transitions that need to be tested. In a traditional histogram test, a ramp or sine wave with sufficient resolution and linearity/spectral purity to provide tens of hits per code (HPC) or higher is used as the input for the ADC static linearity test. This results in an extremely long test time with correspondingly high test and equipment cost.

Testing deeply embedded analog/mixed-signal blocks (such as ADCs) in an SoC is also becoming very challenging and expensive due to the lack of access to the internal nodes as well as the difficulty of maintaining adequate signal integrity while driving an accurate signal on and off chip [4]. A built-in self-test (BIST) capability is thus highly desirable since it doesn't require an external signal generator and the data can be processed using on-chip resources. Researchers have developed several methods to implement such a ramp or sine wave generator on-chip [5]–[7]. However, building such a highly linear signal generator on-chip is not trivial and is often not practical due to the extra cost and limited performance. The stimulus error identification and removal (SEIR) [8], [9] algorithm was proposed to relax the stimulus linearity requirement. In this method, two nonlinear signals with a constant voltage shift between them are used as the stimulus. The method can distinguish between the non-linearity of the ADC and the signal generator thus accurately identifying the ADC's integral nonlinearity (INL) and differential nonlinearity (DNL). Results shown in [8] that a 7-bit linear signal generator can be used with the SEIR technique to test a 16-bit ADC accurately. However, the SEIR is still based on the histogram method which requires a long test time.

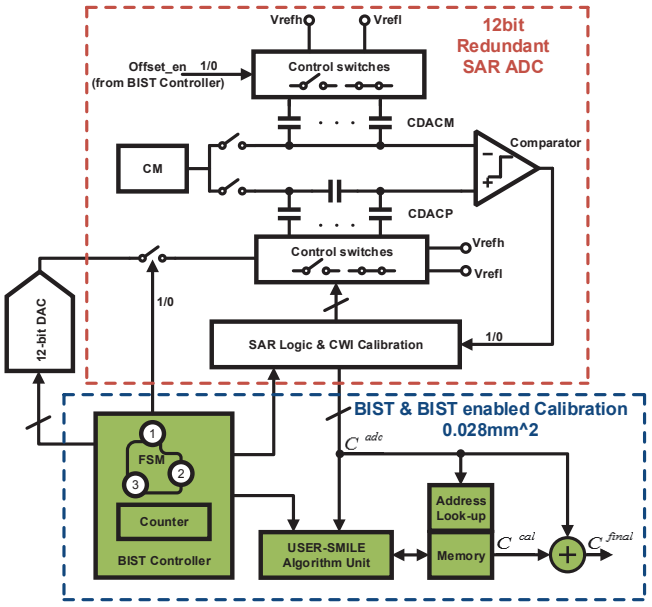

Fig. 1: ADC BIST Subsystem Block Diagram

Test time is also critically important in ADC testing. Many methods have been proposed to reduce the test time [10]–[20]. In [21], the ultrafast segmented model identification of linearity errors (uSMILE) algorithm was proposed. It takes a fundamentally different approach to modeling the ADC's INL

978-1-5386-3413-4/17/$31.00

by applying a "segmented non-parametric model". It achieves more than 100x test time reduction while maintaining similar or better test precision compared to a conventional histogram test. However, it still requires a highly linear input source.

In [22], adopted from SEIR and uSMILE, the ultrafast stimulus error removal and segmented model identification of linearity errors (USER-SMILE) algorithm was proposed to relax the input linearity requirement as well as reduce the test time. In this algorithm, two nonlinear input signals are generated with a constant voltage shift in between thereby easing the design of the signal generator. The segmented INL model reduces the number of samples needed for the INL estimation. With this combination, a low-cost ADC BIST with significantly reduced test time becomes practical.

In this paper, a completely on-chip ADC BIST circuit is developed based on USER-SMILE algorithm and demonstrated on a 28nm CMOS automotive microcontroller. The ADC test subsystem as shown in Fig. 1 includes a 12-bit digital-to-analog converter (DAC), a 12-bit, 1Ms/s single-ended successive-approximation-register (SAR) ADC with a built-in voltage shift generator, a BIST computation engine and dedicated memory cells. The silicon measurement results show a good correlation of test results between ADC BIST and traditional histogram test.

Another aspect of the paper is to leverage the test results to correct the 12-bit SAR ADC's linearity error. Although the switch capacitor based ADCs like charge redistribution SAR ADC appearing in this paper are popular due to their simple structure and power efficiency, the capacitor mismatches as well as other error sources limit their linearity performance as the resolution goes high. Therefore, a method of reliably calibrating the ADC is needed. Different calibration or trimming methods for these types of ADC have been proposed in [27]–[29]. Capacitor switching sequence, perturbation method, and split-ADC architecture are used in these studies to identify and calibrate the capacitor mismatches. With full code nonlinearity error of ADC known to us by the BIST solution, this paper proposes a simple digital calibration which demonstrates >10dB THD/SFDR improvement over an existing structural capacitor-weight-identification (CWI) calibration method.

The remainder of the paper is organized as follows. Section II reviews the fundamentals of USER-SMILE algorithm. Section III discusses the design choices of the USER-SMILE algorithm and adapts it for an efficient hardware implementation. In section IV, the details of the ADC test subsystem and its operation are described. Section V presents the silicon measurement and calibration results. Section VI concludes the paper.

II. Review of USER-SMILE algorithm

In the USER-SMILE algorithm, segmented INL model is used. For an N-bit 3-level segmentation, ADC is divided into N_{MSB}-bit most significant bits (MSB), N_{ISB}-bit intermediate significant bits (ISB) and N_{LSB}-bit least significant bits (LSB). And $N = N_{MSB}+N_{ISB}+N_{LSB}$. Each MSB code has a corresponding error terms e_M. There are $2^{N_{MSB}}$ different codes in MSB segments and there are $2^{N_{MSB}}$ different error terms correspondingly. Define e_M as a row vector:

$$e_M = \begin{bmatrix} e_M(0) \\ e_M(1) \\ \vdots \\ e_M(2^{N_{MSB}}-1) \end{bmatrix} \tag{1}$$

e_I and e_L are defined in a similar way for ISB and LSB. Therefore, for MSB code C_{MSB}, the value of its e_M can be written as a vector multiplication of a column matrix and e_M:

$$e_M(C_{MSB}) = \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} e_M. \tag{2}$$

There is only one 1 in the $C_{MSB}+1$ location and all the other locations are all 0s in the column matrix.

The INL for ADC output code C is modeled as:

$$\begin{aligned} INL(C) &= e_M(C_{MSB}) + e_I(C_{ISB}) + e_L(C_{LSB}) \\ &= \begin{bmatrix} 0 \cdots 1 \cdots 0 \end{bmatrix} e_M \\ &+ \begin{bmatrix} 0 \cdots 1 \cdots 0 \end{bmatrix} e_I \\ &+ \begin{bmatrix} 0 \cdots 1 \cdots 0 \end{bmatrix} e_L \\ &= \begin{bmatrix} 0 \cdots 1 \cdots 0 \cdots 1 \cdots 0 \cdots 1 \cdots 0 \end{bmatrix} \begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} \end{aligned} \tag{3}$$

where C_{MSB}, C_{ISB} and C_{LSB} are the code values for MSB, ISB and LSB bits respectively. In the column matrix, the first 1 appear at the $C_{MSB}+1$ location; the second 1 appear at the $C_{ISB}+1+2^{N_{MSB}}$ location and the last one appear at the $C_{LSB}+1+2^{N_{MSB}}+2^{N_{ISB}}$ location.

There are $K = 2^{N_{MSB}} + 2^{N_{ISB}} + 2^{N_{LSB}}$ unknowns to be solved in order to model the full-code INL. For a 16-bit ADC with 6-5-5 segmentation ($N_{MSB} = 6$, $N_{ISB} = 5$ and $N_{LSB} = 5$), only $K = 128$ unknowns need to be solved and the results can accurately reflect the actual INL.

In the USER-SMILE algorithm, two identical input signals with a constant offset a between them ($V_{in}^{(1)} = V_{in}^{(2)} + a$) are applied to the ADC in two separate samples. The ADC output codes after two conversions from the two input signals are $C^{(1)}$ and $C^{(2)}$. Then, the two input voltages can be expressed in terms of the INL:

$$V_{in}^{(1)} + n^{(1)} = V_{lsb} \cdot \left[C^{(1)} + INL(C^{(1)})\right] + q^{(1)} \tag{4}$$

$$V_{in}^{(2)} + n^{(2)} = V_{lsb} \cdot \left[C^{(2)} + INL(C^{(2)})\right] + q^{(2)} \tag{5}$$

where the noise $n^{(1)}$ and $n^{(2)}$ are the input-referred noise; q is the quantization noise; and V_{lsb} is 1 LSB expressed as a voltage.

By subtracting equation 5 from equation 4 and replacing the INL with the segmented model described in equation 3, equation 6 can be obtained. The term $q^{(2)}$, $q^{(1)}$, $n^{(1)}$ and $n^{(2)}$ can be considered as one noise term $noise$. $V_{in}^{(1)}$ and $V_{in}^{(2)}$ are removed without knowing any information of them. The only thing left is the difference of the two input signals a.

$$
\begin{aligned}
&C^{(1)} - C^{(2)} - a/V_{LSB} \\
=& -e_M(C^{(1)}_{MSB}) - e_I(C^{(1)}_{ISB}) - e_L(C^{(1)}_{LSB}) \\
&+ e_M(C^{(2)}_{MSB}) + e_I(C^{(2)}_{ISB}) + e_L(C^{(2)}_{LSB}) + noise
\end{aligned} \tag{6}
$$

Each set of output codes has one equation. With M sets of output codes, the over-determined system can be formed in equation 7.

$$
y_d = H \begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} + noise \tag{7}
$$

where y_d is a column matrix of M sets of $C^{(1)} - C^{(2)} - a/V_{LSB}$; H is an $M \times K$ matrix where there are three +1s and three -1s in each row and all the other locations are all 0s. Each row represents one set of equation 6. The over-determined system can be solved with least square method (equation 8) and the noise term will be effectively averaged out. After estimating all the unknowns, the full code INL can be constructed using equation 3.

$$
\begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} \approx \left(H^T H\right)^{-1} \left(H^T y_{dk}\right) \tag{8}
$$

As a summary, the USER-SMILE algorithm has the following characteristics and requirements:

1) Segmented INL model: The segmented non-parametric INL model is intended for high resolution ADCs whose architecture facilitates a segmented structure of the INL curve. It assumes that the lower-bit errors are contributed from the same sub-circuit of the ADC, which is true for most high resolution ADCs. Take SAR ADC as an example. Each bit corresponds each capacitor in the capacitor array. If the last 4 bits are the same for different conversions, the connection for last 4 capacitors are the same and the errors contributed from these capacitors will be the same. However, for flash ADC, different codes will correspond to different nodes in the resistor string. The lower bits error will not repeat. So, the USER-SMILE method is not intended for a flash ADC or a delta sigma ADC. For other types of Nyquist-rate ADCs such as SAR ADC, Cyclic ADC, Pipeline ADC, the USER-SMILE algorithm works well.

2) Stimulus requirement: As seen in equation 6, the linearity of test signal is no longer part of the final equations. Thus, the linearity requirement of test signal is completely dropped. This is one of the distinct advantages of USER-SMILE algorithm over SEIR algorithm which significantly eases the required tests of the signal source itself. However, the input signal needs to cover all the MSB, ISB and LSB segments to solve equation 8 providing sufficient number of codes are captured for each segment to ensure estimation accuracy of the error terms. The required resolution of signal source is established through extensive simulation.

3) Constant offset: The constant voltage offset is the most critical part of USER-SMILE algorithm. Any error in the voltage offset will cause estimation errors. The exact value of the voltage offset is also needed during the estimation steps. The offset a can be simply estimated by the average difference between the output codes $C^{(1)}$ and $C^{(2)}$. Although the constancy is not directly measurable in most cases, the voltage offset enabled by capacitor charge sharing scheme in this paper is believed to be the best possible solution for on-chip implementation.

III. Algorithm implementation

A. Design choices - hardware vs. software

If a digital processor is available on chip e.g. a microcontroller, the BIST algorithm can be executed in the form of software routines by the processor. The software approach is more cost effective than the hardware approach (i.e. hardening the algorithm on chip as a co-processor) since no dedicated computing hardware is needed. However, the hardware approach remains attractive because of a few critical advantages that it provides.

1) No dependency on on-chip processor: One obvious advantage the hardware approach offers is that it still works when an on-chip processor is not available. Even with the presence of on-chip processor, the hardware approach offers the opportunity to test the ADC in a non-intrusive and non-competing manner. This advantage is critical for in-field testing when the on-chip processor is busy handling multiple user applications.

2) Memory efficiency: In the software approach, the software implementation of the algorithm needs to be stored in on-chip memory (static RAM or non-volatile memory). This takes up system memory space which could cause issues especially when the system memory is tiny such as on some low-end microcontrollers. Contrary to the software approach, other than a small static RAM being used to hold ADC conversion results as well as intermediate data, no system memory is required for hardware approach.

3) Fast execution time: A dedicated computing resource realized by the hardware approach also speeds up the execution time of the algorithm significantly. Given same clock rate, it can be shown that the hardware approach is 38 times faster than the software approach based on ARM Cortex-M7 core. In the case of field-test, the hardware approach also means that the self-test can be completed much faster than the software approach. It makes a big difference in determining whether the self-test can be included as part of the power-on self-check sequence in an automotive application.

With all above being said, software approach still has its own advantages such as cost efficiency and flexibility. The decision regarding which approach to choose is driven by the availability of an on-chip processor, the technology node being used, the system memory size and the user application. In this paper, the hardware approach is chosen due to its independence from an on-chip processor and reduced implementation complexity. As demonstrated later, the algorithm

adaptation for hardware implementation and the advanced technology node being used help deliver a very efficient algorithm hardware.

B. Adaptation of USER-SMILE for hardware implementation

The fundamentals of the USER-SMILE algorithm are presented in above section. However, since H is an $M \times K$ matrix and equation 8 requires the matrix inversion of an $K \times K$ matrix, a prohibitively long computation time and large data storage are required which is not suitable for an on-chip implementation. In the above example, for a 16-bit ADC with 6-5-5 segmentation using 1 hit/code as the input, M will be 65536 and K is 128. To facilitate the ADC BIST on-chip computation and achieve fast and effective estimation, some modifications to the original equations must be made. In matrix H, three sub-matrices H_M, H_I and H_L can be defined, where H_M is column 1 to $2^{N_{MSB}}$ in H, H_I is column $2^{N_{MSB}}+1$ to $2^{N_{MSB}}+2^{N_{ISB}}$ in H and H_L is column $2^{N_{MSB}}+2^{N_{ISB}}+1$ to $2^{N_{MSB}}+2^{N_{ISB}}+2^{N_{LSB}}$ in H. Then, equation 7 can be expressed as:

$$y_{dk} = \begin{bmatrix} H_M & H_I & H_L \end{bmatrix} \cdot \begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} + noise \tag{9}$$

Multiply both sizes by H^T, which is $\begin{bmatrix} H_M & H_I & H_L \end{bmatrix}^T$.

$$H^T \cdot y_{dk} = H^T \cdot H \cdot \begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} + noise \tag{10}$$

In equation (10), $H^T \cdot y_{dk}$ is a column matrix after multiplication and $H^T \cdot H$ is a $K \times K$ matrix.

$$\begin{aligned} H^T \cdot H &= \begin{bmatrix} H_M^T \\ H_I^T \\ H_L^T \end{bmatrix} \begin{bmatrix} H_M & H_I & H_L \end{bmatrix} \\ &= \begin{bmatrix} H_M^T H_M & H_M^T H_I & H_M^T H_L \\ H_I^T H_M & H_I^T H_I & H_I^T H_L \\ H_L^T H_M & H_L^T H_I & H_L^T H_L \end{bmatrix} \end{aligned} \tag{11}$$

Note that in this equation, most elements are 0s in $H_M^T H_I$, $H_M^T H_L$, $H_I^T H_M$, $H_I^T H_L$, $H_L^T H_M$, $H_L^T H_I$. Then, these sub-matrices can be replaced with 0s and $H^T \cdot H$ can be estimated as:

$$H^T \cdot H \approx \begin{bmatrix} H_M^T H_M & \dots & 0 \\ \vdots & H_I^T H_I & \vdots \\ 0 & \dots & H_L^T H_L \end{bmatrix} \tag{12}$$

With the estimated $H^T \cdot H$, equation 10 can be expressed as:

$$\begin{bmatrix} H_M^T \\ H_I^T \\ H_L^T \end{bmatrix} \cdot y_{dk} \approx \begin{bmatrix} H_M^T H_M & \dots & 0 \\ \vdots & H_I^T H_I & \vdots \\ 0 & \dots & H_L^T H_L \end{bmatrix} \begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} + noise \tag{13}$$

Next, define $H_{ydM} = H_M^T y_{dk}$ and $H_{HM} = H_M^T H_M$. Then H_{ydI}, H_{ydL} and H_{HI}, H_{HL} are defined in a similar way. The matrix can then be divided into three smaller matrices as shown in equation 14 which are now independent of each other so that they can be solved sequentially.

$$\begin{aligned} H_{ydM} &\approx H_{HM} \cdot e_M \\ H_{ydI} &\approx H_{HI} \cdot e_I \\ H_{ydL} &\approx H_{HL} \cdot e_L \end{aligned} \tag{14}$$

Using the least-square method, e_M, e_I and e_L can be solved in equation 15.

$$\begin{aligned} e_M &\approx \left(H_{HM}\right)^{-1} H_{ydM} \\ e_I &\approx \left(H_{HI}\right)^{-1} H_{ydI} \\ e_L &\approx \left(H_{HL}\right)^{-1} H_{ydL} \end{aligned} \tag{15}$$

For every two samples, one set of ADC output codes will be obtained $C^{(1)}$ and $C^{(2)}$: one with offset and one without offset. Then, the matrix H_{HM} and H_{ydM} will be updated by the following equations:

$$\begin{aligned} H_{HM}\left(C_{MSB}^{(1)}, C_{MSB}^{(1)}\right) &= H_{HM}\left(C_{MSB}^{(1)}, C_{MSB}^{(1)}\right) + 1 \\ H_{HM}\left(C_{MSB}^{(2)}, C_{MSB}^{(2)}\right) &= H_{HM}\left(C_{MSB}^{(2)}, C_{MSB}^{(2)}\right) + 1 \\ H_{HM}\left(C_{MSB}^{(1)}, C_{MSB}^{(2)}\right) &= H_{HM}\left(C_{MSB}^{(1)}, C_{MSB}^{(2)}\right) - 1 \\ H_{HM}\left(C_{MSB}^{(2)}, C_{MSB}^{(1)}\right) &= H_{HM}\left(C_{MSB}^{(2)}, C_{MSB}^{(1)}\right) - 1 \\ H_{ydM}\left(C_{MSB}^{(1)}\right) &= H_{ydM}\left(C_{MSB}^{(1)}\right) + y_{dk} \\ H_{ydM}\left(C_{MSB}^{(2)}\right) &= H_{ydM}\left(C_{MSB}^{(2)}\right) - y_{dk} \end{aligned} \tag{16}$$

The corresponding matrices for ISB and LSB will be also updated in a similar fashion. This operation only requires simple additions and increments which can be finished before the next set of output codes are ready. After all M sets of codes have been obtained, the complete matrix can then be formed and the MSB, ISB and LSB error terms can then be evaluated one by one. Since H_{HM}, H_{HI} and H_{HL} are positive definite matrices, Cholesky decomposition can be used to evaluate the least square. The actual hardware implementation will be explained in section IV.

In all the equations above, we assume that the value of the offset between the pair of input signals is known. However, in practice, it is difficult to measure the offset directly so the offset value is estimated by calculating the average difference between the two sets of ADC output codes. During the data acquisition time, the average difference is also unknown. In order to calculate it, a default offset value $a^{(def)}$is preset at the beginning. There will be a difference between the default offset $a^{(def)}$ and the actual offset a which is summed and stored in a variable which is then used to store the summation of the output code difference to compute the average offset value. The H_{ydM}, H_{ydI} and H_{ydL} values will be updated after all the data is collected and the average output code difference is computed. During this process, the ADC output codes will not be stored in memory and only H_{HM}, H_{HI}, H_{HL} and H_{ydM}, H_{ydI}, H_{ydL} are stored. The memory required for the algorithm is thus significantly reduced and the subsequent computation time is also reduced since the matrix is already formed. This

adaption from the USER-SMILE enables the on-chip ADC BIST implementation.

IV. ADC BIST SUBSYSTEM DESIGN

The block diagram of the ADC BIST subsystem is shown in Fig. 1. The 12-bit redundant SAR ADC is the test object. An 12-bit resistive DAC is used here as test signal generator which does not have sufficient resolution and linearity accuracy to test 12-bit ADC using traditional histogram method. The ADC BIST and calibration logic is designed as a standalone digital block that performs all required control, computing and calibration tasks. A dedicated test mode is defined for the ADC BIST block to take over the essential control over ADC and DAC when the mode is activated. Each block is reviewed in details below.

A. 12-bit resistive DAC

A signal generator for this BIST solution needs to cover the majority of the input range of ADC to ensure most of the codes are hit. Although its linearity performance requirement is removed, it needs to provide sufficient resolution to drive ADC to generate sufficient number of code pairs (with and without offset) for accurate INL approximation by the algorithm. 1 hit per code is demonstrated to be sufficient as seen in simulation, therefore a 12-bit DAC is required. The chosen 28nm automotive microcontroller happens to carry a 12-bit DAC using resistor-ladder architecture which meets the requirements. When the BIST test mode is enabled, a 12-bit counter inside the BIST controller is sending control word to the DAC which output is sampled by the ADC. Every one DAC output after settling is sampled by ADC twice, one without offset and one with offset. This is to make sure that the ADC is converting same input voltage where the offset is applied within ADC as detailed in next section. Benefiting from the USER-SMILE algorithm, the DAC doesn't need to be tested on its linearity performance. Therefore, the checking of a dead DAC (ADC output code does not change), missing input range (no code found in specific segments) and invalid offset are implemented in ADC BIST sub-block to screen out catastrophic DAC failures only.

B. 12-bit redundant SAR ADC and voltage offset creation

The ADC under test is a single-ended charge redistribution SAR ADC. As shown in Fig. 2, the p-side CDAC array (CDACP) is the main array being used for conversion. The m-side CDAC array (CDACM) maintains common mode voltage on its top plate to provide reference voltage and capacitor matching to the comparator. A split-capacitor structure is implemented to limit the die size of CDACP. Therefore, the CDACP is partitioned into 3 binary weighted sub-DACs based on unit size capacitors C_u. The MSB capacitor array is controlled by 5 bits (bit 12 to 9 and bit s), the ISB capacitor array is controlled by 4 bits (bit 8 to 5), and the LSB capacitor array is controlled by 6 bits (bit 4 to 0 and bit terminal). To correct the capacitor mismatches, a calibration scheme is implemented. To distinguish this with later proposed

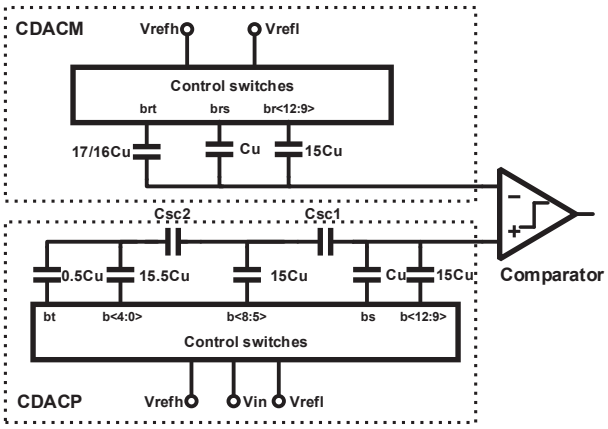

Fig. 2: CDAC Array of SARADC

BIST enabled calibration, this calibration scheme is referred as structural capacitor-weight-identification (CWI) calibration in the remaining sections. The scheme is implemented like [27] where lower weighted capacitors are exercised to measure weight of each individual capacitor in MSB array progressively to determine correction codes. Split capacitor Csc1 is over-sized to $1.125C_u$ to introduce redundancy to ensure monotonicity. The missing codes resulted from the redundancy is removed by this CWI calibration scheme, however it reduces the code range or resolution of the ADC. An additional unit sized capacitor C_s is added into MSB array which is controlled by bit s to extend the code range. The C_s induced gain error is calculated based on the CWI calibration code and is applied to exclude certain capacitors from sampling V_{in} during sampling phase for compensation. The bit evaluation phase involves 14 cycles that includes evaluation of bit 12 to 0 and bit s. 12 bit results are reported as the final conversion results after CWI calibration, extraction and rounding.

The accuracy of linearity approximation provided by the USER-SMILE algorithm is critically dependent on the constancy of the voltage offset [24]. Previous publications have focused on creating the voltage offset within a signal generator e.g. [25]. This significantly complicates the design of the signal generator and may still not provide the required constancy across the entire input range of the ADC. As presented by [26], an additional capacitor is inserted into the existing SAR capacitor array to create the voltage offset which is only determined by the capacitor ratio and a constant reference voltage supply. This is believed to provide superior constancy and simplicity to prior arts.

In this work, the creation of constant voltage offset shares the same principle as [26]. Instead of adding an additional capacitor, the CDACM array is leveraged to create a different reference voltage for the comparator. When a voltage offset is wanted, the bottom plate of capacitor C_{rt} which is controlled by bit brt is toggled to V_{refh} during evaluation phase where the bottom plates of other capacitors in CDACM remain connected to V_{refl}. A few disadvantages that were encountered

in [26] have been avoided. No additional capacitor is required in this implementation. It does not reduce the step size seen by the comparator. The top plate voltage of CDACM does not surpass safeguard voltage which avoided a potential device reliability risk. This essentially created a voltage offset shown in Eq. 17.

$$V_{Offset} = (V_{refh} - V_{refl}) \cdot \frac{C_{rt}}{C_M} \cdot (2^N + W(C_s) - C^{cal}) \quad (17)$$

In the above equation, V_{refh} and V_{refl} are the high reference voltage and low reference voltage respectively. C_M is the sum of total capacitors in CDACM array. N is the resolution of ADC. $W(C_s)$ is the weight of C_s where C^{cal} is the total sum of CWI calibration codes of MSB capacitors. The constancy of the offset across all possible codes is determined by the constancy of capacitor ratio C_{rt}/C_M, the stability of reference voltages V_{refh} and V_{refl}, and the constancy of offset voltage of the comparator. CDACM does not participate the bit evaluation while merely providing reference voltage to the comparator. The constancy of the capacitor ratio should hold regardless of the code being converted. The reference voltages are supplied from dedicated and precise tester instrument which short term drift performance is guaranteed. A large amount of decoupling capacitors are added on the reference voltage supplies to reduce noise and maintain voltage level. The voltage offset of the comparator is small due to the high gain of comparator and fixed due to fixed inputs of the comparator. Substituting true design values into the equation and representing V_{Offset} in LSB, the expected amount of offset is between 263 and 264 LSB depending on actual value of C^{cal}. Although there is no direct way to measure the constancy of the offset, we use the expected value and the silicon correlation results in section V to measure it indirectly.

C. BIST and BIST enabled calibration

The BIST block is consisted of a finite sate machine (FSM) sub-block, the hardened USER-SMILE algorithm unit, a small memory to hold intermediate and final test data, a small test MUX sub-block, and a calibration sub-block enabled by BIST. The FSM sub-block is designed control the operation sequence of the subsystem. When the operation kicks off by activating BIST test mode, the FSM logic starts a 12-bit counter that sends control words to the 12bit DAC. It also toggles the offset enable control signal to the ADC for voltage offset creation. Once valid ADC conversion results are available, the FSM logic kicks off the USER-SMILE algorithm unit for post processing. The entire operation continues until all possible control words of 12-bit DAC have been covered and valid BIST pass/fail status have been reported. The test MUX is a simple logic block that is designed to bypass the normal control and data path of the DAC and the ADC for the BIST block to take over. The algorithm unit is the hardened block implementation of the USER-SMILE algorithm. A small dedicated memory is paired with the algorithm unit to hold intermediate and final data. It would be possible to share a small portion of the system memory within the SoC for this purpose to reduce the die size of the solution. We choose to implement a dedicated memory due to its small size and simplicity of integration. Below sections highlight the design of algorithm unit and BIST enabled calibration logic.

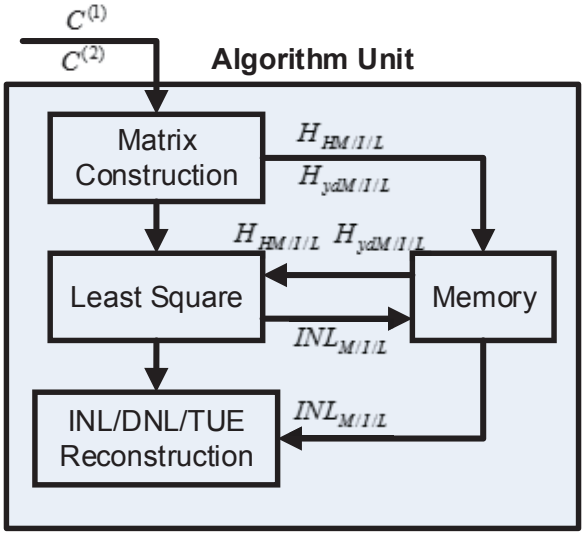

Fig. 3: Algorithm Unit Design

1) High level synthesis assisted algorithm unit design: Per the USER-SMILE algorithm, the nonlinearity of SAR ADC is modeled as coefficients in MSB, ISB and LSB segments which are represented as INL_M, INL_I and INL_L. After running through all ADC output codes, a matrix equation is constructed in the form of the equation below.

$$H_{Hx} \cdot INL_x = H_{ydx}, x \in [M, I, L]. \quad (18)$$

H_{Hx} and Hyd_x are obtained through equation 16. The hardened algorithm unit is implemented as shown in Fig. 3. The matrix construction block receives the pair of ADC output codes (with and without voltage offset applied): $C^{(1)}$ and $C^{(2)}$, and constructs a matrix HH_x and Hyd_x, and then saves the matrix into memory until all valid ADC conversions have been completed. The least square block then uses the linear least square method to solve this over-determined system to get INL_x. Once all of the INL coefficients have been identified, they will be used to reconstruct final INL, DNL and TUE parameters. A high level synthesis (HLS) technique was used to realize this design. HLS significantly simplifies the design, speeds up the design process while yielding satisfactory results.

2) BIST enabled Calibration: The SAR ADC has its own structural CWI calibration based on capacitor mismatch measurement as detailed in section IV-B. However, calibrating capacitor mismatch may not provide best DNL/INL results as the total errors contributing to INL is not only from the capacitor mismatches. All these structure-based test methods

have such limitations as they cannot remove other error sources. The goal for ADC static linearity is to achieve minimum INL. With the full-code INL/DNL information, we can calibrate it correspondingly. Unlike other structural test which have some assumptions, full-code INL doesn't assume any structure-specific errors. Therefore, it is preferred to obtain the full-code INL information for calibration purpose. As the BIST solution readily provides the full-code INL information, it is only natural to extend the BIST results to further correct ADC linearity errors.

INL is defined as the deviation of code transition from its ideal location. The INL at code C is:

$$INL(C) = \big[T(C) - T^i(C)\big]/V_{lsb} \tag{19}$$

where $T(C)$ is the actual transition voltage from code $C-1$ to C; $T^i(C)$ is the ideal transition voltage; and V_{lsb} is the ideal 1 LSB in voltage.

For digital calibration, the final output code is:

$$C^{final} = C^{adc} + C^{cal} \tag{20}$$

where C^{adc} is the adc codes from shift registers and C^{cal} is the calibration code we want to generate. The INL after calibration is expressed as:

$$\begin{aligned} INL(C^{final}) &= \big[T(C^{adc}) - T^i(C^{final})\big]/V_{lsb} \\ &= \big\{T(C^{adc}) - [T^i(C^{adc}) + C^{cal} \cdot V_{lsb}]\big\}/V_{lsb} \\ &= INL(C^{adc}) - C^{cal} \end{aligned} \tag{21}$$

As indicated in equation 21, to minimize $INL(C^{final})$, C^{cal} should be equal to $INL(C^{adc})$. Considering the most significant error is from the MSB segment, in the current implementation, only MSB errors are calibrated. Therefore, the calibration code is:

$$C^{cal} = round\big(e_M(C^{adc}_{MSB})\big) \tag{22}$$

where C^{cal}_{MSB} is the MSB segment codes of C^{cal} and the $round$ function is to round the calibration code to the ADC's minimum quantization bit. Assuming the error from ISB and LSB segment is small, the INL after calibration is the rounding error due to finite resolution.

V. Measurement Results

The ADC BIST and BIST enabled calibration logic consumes 18K gates after logic synthesis. The 4KB memory array occupies $0.011mm^2$. With 75% utilization rate, the total BIST solution occupies $0.028mm^2$ on 28nm CMOS technology. Compared to a minimum 20 hits per code that is required by traditional histogram ramp test method in production, this BIST solution only requires 2 measurements per code. The computation time of the BIST takes 1ms with a 80MHz clock which is negligible in the overall test time. A minimum 10X tester time reduction is achieved which significantly reduces the ADC's test cost. Although cost metric (test cost per second, die cost per mm^2 and etc.) varies and deemed business confidential, the cost of increased die size is estimated to be only 1/8 of the saved cost from tester time alone.

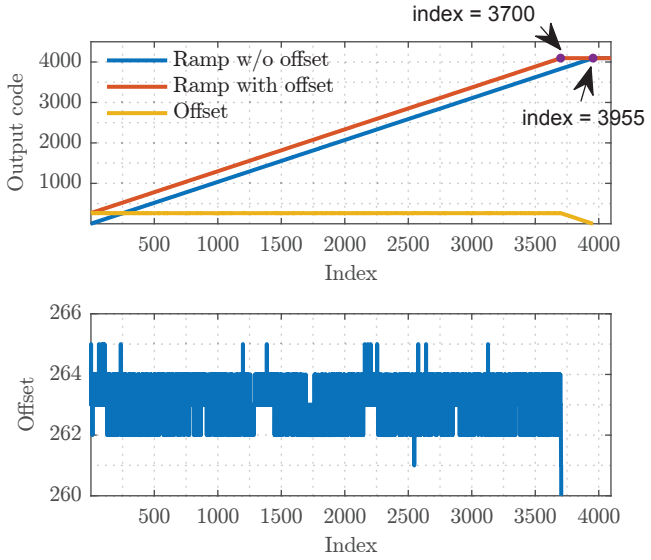

Fig. 4: ADC output

A. BIST test results

In order to verify the test accuracy of the ADC BIST solution, a 200 HPC histogram ramp test is developed for comparison. The ramp signal is delivered by a high-performance tester instrument which is specified to be sufficiently linear. The ADC BIST only uses two 1 hit per code ramps generated using the on-chip DAC. 5-4-3 segmentation is chosen for the definition of INL error coefficients to mimic the segmentation of SARADC capacitor array, thus only 56 ($2^5 + 2^4 + 2^3$) error coefficients need to be identified. After INL or DNL performance data for 4096 codes is derived, they are saved into on-chip memory for correlation or characterization.

Raw ADC data with and without offset are captured and shown in Fig. 4. The amount of offset is indeed centering around 263 LSB as calculated in Eq. 17. As opposed to a clean and constant value where only an ideal noise-free and error-free ADC could achieve, the offset clearly incorporates the noise and nonlinearity errors which are the exact info needed by the USER-SMILE algorithm. A minor issue is exposed by Fig. 4 where the ramp with offset saturates at code index 3700. This reduces the effective number of code pairs from 4096 to 3700 to construct the matrix which may negatively impacts the test accuracy of BIST solution. The root cause of the issue is found out to be the C_s induced gain error which is not properly compensated in ADC BIST test mode. An fix to the issue is either to remove the gain error completely or to introduce a negative gain error which would maximize the available number of code pairs. However, it has to be pointed out that all segments of the ADC are covered from 0 to 4095 despite this issue as shown clearly in Fig. 4.

The DNL and INL curves measured on one unit using the 200 HPC histogram method and ADC BIST method are plotted in Fig. 5. Fig. 5.c showed the Δ of INL between histogram method and BIST solution. Despite spikes at a few locations, most differences are within +/-0.5LSB which

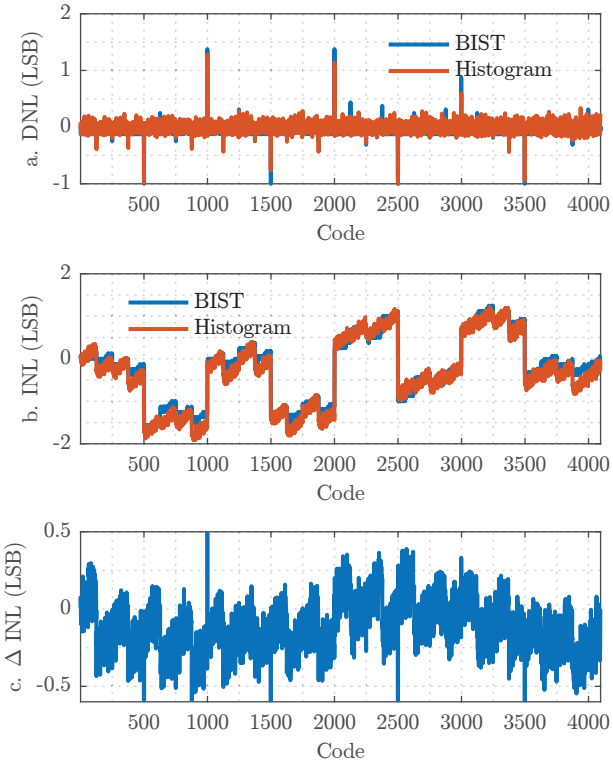

Fig. 5: DNL/INL correlation

indicates the level of BIST test accuracy. This level of test accuracy is deemed sufficient as demonstrated in following correlation exercises. However, a few things are believed to attribute to the differences:

- In USER-SMILE algorithm, the noise term including input referred noise and quantization noise is assumed to be averaged to 0 when certain criteria is met. In reality, this may not be the case which causes estimation error. Proper dithering is needed.
- The amount of offset in this design is more than 2 MSB segments (<128 code for each MSB segment) which causes fewer correlated code pairs between adjacent even and odd MSB segments. In Fig. 5.c, the "toggling" of the differences in even and odd MSB segments can be attributed to this. Choosing appropriate offset is required to address this issue.
- The aforementioned C_s induced gain error not being compensated reduces the available number of code pairs for algorithm data processing, thus reducing the estimation accuracy.

Fig. 6 zoomed in on one of the code locations where a spike is noticed. As the graph clearly indicates, the error is due to the misalignment of transition edge between two MSB segments. This happens during reconstruction of the full code INL when the transition edges of MSB segments is approximated without knowing their exact code location. Those edges are determined

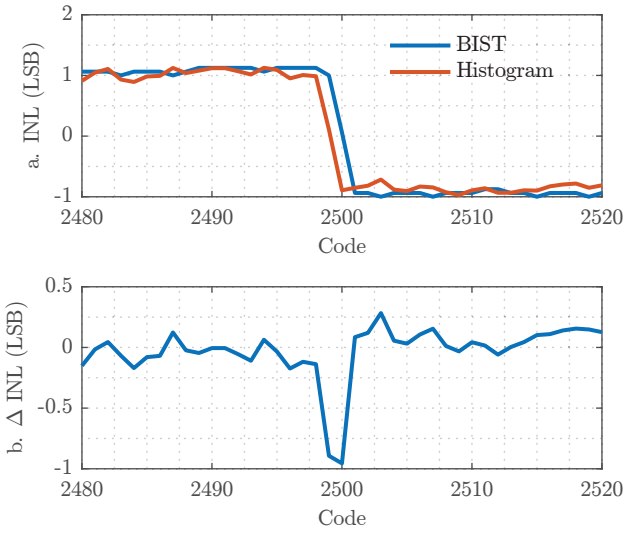

Fig. 6: Zoom in on one spike

by the CWI calibration codes, however the codes are not made directly available to the BIST sub-block where approximation has to be made. The fix is straightforward which is to make the CWI calibration codes accessible to BIST block. As this is a misalignment issue, it does not impact the maximum DNL and INL errors which are reported to determine the pass/fail status of the ADC test.

To further verify the correlation between histogram and ADC BIST and to establish the error tolerance of the BIST solution, a large number (> 700) of parts from different corner processes are tested across maximum and minimum operating voltages and temperatures. The maximum INL and minimum INL are obtained for comparison. Fig. 7 plots the INL data obtained from histogram test with different number of hits per code. It shows good correlation between the two with +/- 0.5LSB as a clear boundary of the variance.

Fig. 8 plots the correlation of INL between the histogram method using 200 HPC and the BIST solution. The same +/- 0.5LSB boundary is applied for comparison. Although the correlation between those two sets of data is slightly worse than the correlation in Fig. 7 with a few data points falling out of the boundary, it clearly indicates similar level of test accuracy of BIST solution when compared to that of traditional histogram method. There appeared to be 4 outlier points, but since the tester does not save intermediate test data to save tester time, attempting to identify the outlier parts or causes would require long test time and high cost that are not warranted since the errors are small. With the ADC INL specification limits set at +/-3LSB, the BIST accuracy is deemed to be sufficient with no good unit being rejected or bad unit escaping. One unit that failed on both histogram tests is correctly identified and rejected by BIST solution.

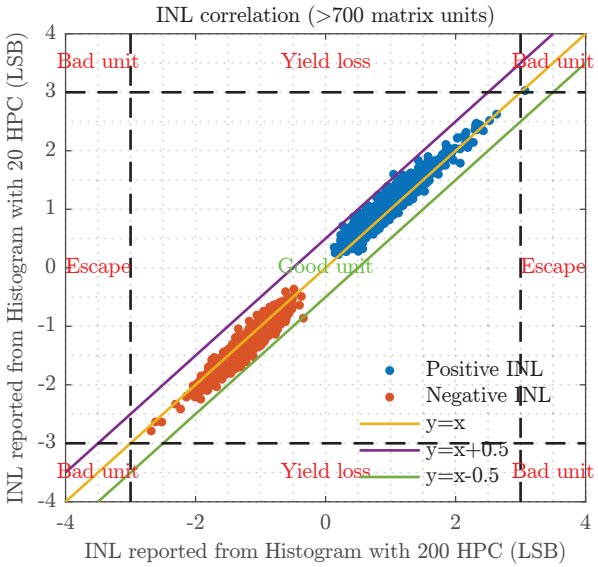

Fig. 7: Histogram INL correlation between 200 HPC and 20 HPC

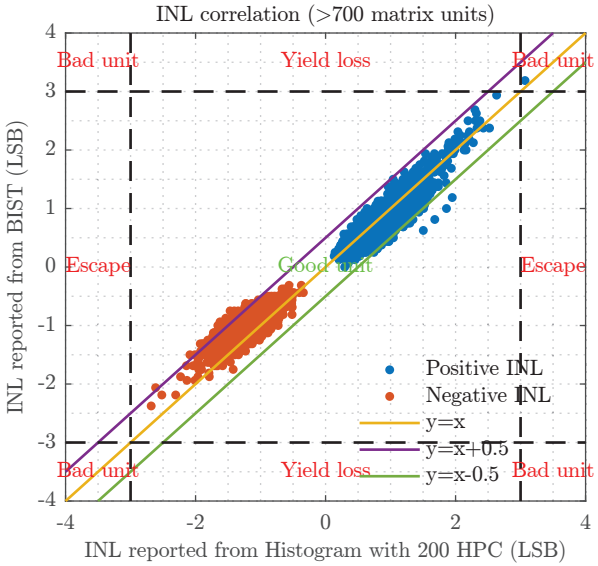

Fig. 8: INL correlation between 200 HPC and BIST

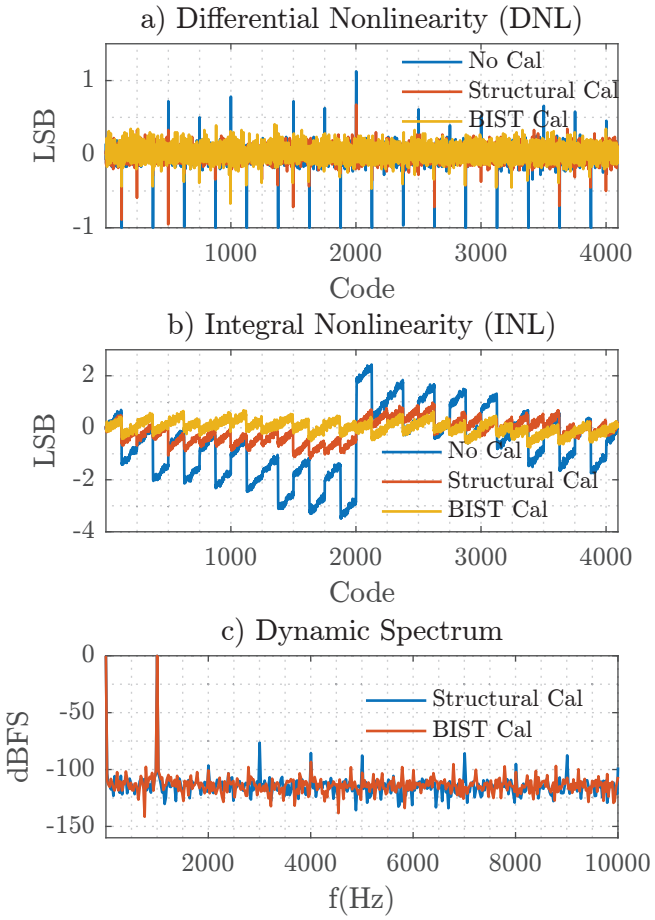

Fig. 9: Performance Comparison

B. Calibration results

To prove the effectiveness of this BIST enabled calibration scheme, the ADC is tested on a high end mixed signal tester. Traditional histogram based test is performed to find its static linearity performance. Fig. 9a) and 9b) showed a typical device's DNL and INL under different calibration conditions. Both structural CWI calibration and BIST enabled calibration can bring the ADC static performance to a satisfactory level. However, noticeable performance enhancement of DNL and INL is achieved by the BIST enabled calibration. As only MSB errors are calibrated, the INL error is now dominated by the gain error in each MSB segment and noise. The aforementioned code misalignment issue does not impact the calibration accuracy here since the MSB bits are monitored and used to decode and retrieve calibration codes from BIST memory where CWI calibration codes need not be known. Since the low frequency dynamic performance of a ADC is closely tied to its static linearity, a dynamic test is performed to obtain the dynamic spectrum as shown in Fig. 9c). The detailed performance data of one device is shown in Table I.

TABLE I: ADC performance comparison

	No Cal	Structural CWI Cal	BIST Cal
DNL (LSB)	1.1253/-1	0.6726/-0.9531	0.4066/-0.6718
INL (LSB)	2.4324/-3.4909	0.9636/-1.1818	0.6633/-0.7572
SNR (dB)	62.3533	68.6602	69.7948
THD (dB)	-64.8619	-74.937	-84.4755
SNDR (dB)	60.4186	67.7412	69.6494
SFDR (dB)	66.2537	75.4594	86.5137
ENOB (bit)	9.744	10.9603	11.2773

A >0.3bit ENOB improvement is achieved over structural CWI calibration with the THD and SFDR contributing the most with each parameter gaining >10dB improvement. The performance improvement is significant even not considering this calibration being essentially free. Applying this BIST

enabled calibration scheme does not exclude the BIST solution from testing the ADC again as the fundamentals of the BIST algorithm are not changed by the calibration.

VI. Conclusion

An ADC BIST solution and calibration scheme has been presented and reviewed. It removes the dependency to a high-performance external test instrument, and also significantly reduces test time. Extensive correlation has been performed to demonstrate the test accuracy of the BIST solution. The presented technique is extended to calibrate an ADC for superior static and dynamic linearity performance. It opens the door for in-field ADC performance testing that may be required to meet stringent functional safety requirement in certain application e.g. autonomous driving. The BIST circuit's die size cost is negligible compared to the test cost savings being realized, both recurring and non-recurring.

VII. Acknowledgment

The authors would like to thank Allan Scott at NXP Semiconductors for helping collect tester data of BIST and histogram test for extensive correlation.

References

[1] International Technology Roadmap for Semiconductors, 2011 edition, [Online]. Available: http://www.itrs.net/

[2] T. Kuyel, "Linearity Testing Issues Of Analog To Digital Converters," ITC International Test Conference, pp. 747-756, 1999.

[3] IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters, in IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000) , vol., no., pp.1-139, Jan. 14 2011

[4] J. Duan, D. Chen and R. Geiger, "A low cost method for testing offset and gain error for ADC BIST," Circuits and Systems (ISCAS), 2012 IEEE International Symposium on, Seoul, 2012, pp. 2023-2026.

[5] F. Azais, S. Bernard, Y. Bertrand, X. Michel and M. Renovell, "A low-cost adaptive ramp generator for analog BIST applications," VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001, Marina Del Rey, CA, 2001, pp. 266-271.

[6] B. Provost and E. Sanchez-Sinencio, "On-chip ramp generators for mixed-signal BIST and ADC self-test," in IEEE Journal of Solid-State Circuits, vol. 38, no. 2, pp. 263-273, Feb 2003.

[7] G. Renaud, M. J. Barragan and S. Mir, "Design of an on-chip stepwise ramp generator for ADC static BIST applications," 2015 IEEE 20th International Mixed-Signals Testing Workshop (IMSTW), Paris, 2015, pp. 1-6.

[8] L. Jin, et al, "Accurate Testing of Analog-to-Digital Converters Using Low Linearity Signals With Stimulus Error Identification and Removal," IEEE Paper 10.2 INTERNATIONAL TEST CONFERENCE 10 Trans. On Instrumentation And Measurement, Vol. 54, No. 3, pp. 1188-1199, June 2005.

[9] L. Jin, et al, "SEIR Linearity Testing of Precision A/D Converters in Nonstationary Environments With Center-Symmetric Interleaving," IEEE Transactions On Instrumentation And Measurement, Vol. 56, No. 5, pp. 1776-1785, October 2007.

[10] F. Adamo, et al, "FFT Test of A/D Converters to Determine the Integral Nonlinearity," IEEE Trans. On Instrumentation and Measurement, Vol. 51, No. 5, pp. 1050-1054, October 2002.

[11] F. Attivissimo, et al, "INL reconstruction of A/D converters via parametric spectral estimation," IEEE Trans. Instrum. Meas., vol. 53, no. 4, pp. 940946, Aug. 2004.

[12] A. Cruz Serra, et al, "Combined Spectral and Histogram Analysis for Fast ADC Testing," IEEE Transactions On Instrumentation And Measurement, Vol. 54, No. 4, pp. 1617-1623, August 2005.

[13] Z. Yu, et al, "Pipeline ADC Linearity Testing with Dramatically Reduced Data Capture Time," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 792-795, 2005.

[14] L. Jin, et el, "Linearity Test of A/D Converters Using Kalman Filtering," IEEE International Test Conference, Paper 28.3, Santa Clara, CA, pp. 1-9, Oct. 2006.

[15] B. Vasan, et al, "Linearity Testing of ADCs Using Low Linearity Stimulus and Kalman Filtering," IEEE International Symposium on Circuits and System, pp. 3032-3035, June 2010.

[16] S. Kook, et al, "Low-Resolution DAC-Driven Linearity Testing of Higher Resolution ADCs Using Polynomial Fitting Measurements," to appear in IEEE Transactions On Very Large Scale Integration (VLSI) Systems, 2012.

[17] J. Duan, et al, "INL Based Dynamic Performance Estimation for ADC BIST," IEEE International Symposium on Circuits and System, pp. 3028-3031, June 2010.

[18] S. Goyal, et al, "Test Time Reduction of Successive Approximation Register A/D Converter By Selective Code Measurement," International Test Conference, Nov 2005.

[19] H. Chang, et al, "Calibration and Testing Time Reduction Techniques for a Digitally-Calibrated Pipelined ADC," VLSI Test Symposium, 2009.

[20] S. Uemori, et al, "ADC linearity test signal generation algorithm," IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2010.

[21] Z. Yu and D. Chen, "Algorithm for Dramatically Improved Efficiency in ADC Linearity Test," Proceedings International Test Conference 2012. 2012.

[22] T. Chen and D. Chen, "Ultrafast stimulus error removal algorithm for ADC linearity test," VLSI Test Symposium (VTS), 2015 IEEE 33rd, Napa, CA, 2015, pp. 1-5.

[23] L. Schuchman, "Dither Signals and Their Effect on Quantization Noise," IEEE Trans. Commun. Technol. , vol. COM-12, pp. 162-165 (1964).

[24] J. Duan, B. Vasan, C. Zhao, D. Chen and R. Geiger, "Stimulus generator for SEIR method based ADC BIST," Proceedings of the IEEE 2009 National Aerospace and Electronics Conference (NAECON), Dayton, OH, 2009, pp. 251-255.

[25] J. Duan, D. Chen and R. Geiger, "Cost effective signal generators for ADC BIST," Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on, Taipei, 2009, pp. 13-16.

[26] X. Jin and N. Sun, "Low-cost high-quality constant offset injection for SEIR-based ADC built-in-self-test," Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, Melbourne VIC, 2014, pp. 285-288.

[27] H. S. Lee, D. A. Hodges and P. R. Gray, "A self-calibrating 15 bit CMOS A/D converter," in IEEE Journal of Solid-State Circuits, vol. 19, no. 6, pp. 813-819, Dec. 1984.

[28] W. Liu, P. Huang and Y. Chiu, "A 12-bit, 45-MS/s, 3-mW Redundant Successive-Approximation-Register Analog-to-Digital Converter With Digital Calibration," in IEEE Journal of Solid-State Circuits, vol. 46, no. 11, pp. 2661-2672, Nov. 2011.

[29] J. A. McNeill, K. Y. Chan, M. C. W. Coln, C. L. David and C. Brenneman, "All-Digital Background Calibration of a Successive Approximation ADC Using the 'Split ADC' Architecture," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 58, no. 10, pp. 2355-2365, Oct. 2011.