A 12-bit 80-Msample/s Pipelined ADC with Bootstrapped Digital Calibration

Carl R. Grace, Paul J. Hurst, and Stephen H. Lewis

C. R. Grace was with UC Davis. He is now with ClariPhy Communications Inc., Irvine CA 92618 USA. P. J. Hurst and S. H. Lewis are with the Solid-State Circuits Research Laboratory, Department of Electrical and Computer Engineering, University of California, Davis, CA 95616 USA.

This work was supported by SRC Grant 2000-HJ-787 and by UC MICRO Grant 02-040.

Abstract

This paper presents a prototype analog-to-digital converter (ADC) that uses a calibration algorithm to adaptively overcome constant closed-loop gain errors, closed-loop gain variation, and slew-rate limiting. The prototype consists of an input sample-and-hold amplifier (SHA) that can serve as a calibration queue, a 12-bit 80-Msample/s pipelined ADC, a digital-to-analog converter (DAC) for calibration, and an embedded custom microprocessor, which carries out the calibration algorithm. The calibration is bootstrapped in the sense that the DAC is used to calibrate the ADC, and the ADC is used to calibrate the DAC. With foreground calibration, test results show that the peak differential nonlinearity (DNL) is -0.09 least significant bits (LSB), and the peak integral nonlinearity (INL) is -0.24 LSB. Also, the maximum signal-to-noise-and-distortion ratio (SNDR) and spurious-free dynamic range (SFDR) are 71.0 dB and 79.6 dB with a 40-MHz sinusoidal input, respectively. The prototype occupies 22.6 mm^2 in a 0.25 μm CMOS technology and dissipates 755 mW from a 2.5 V supply.

Keywords

Adaptive systems, analog-to-digital conversion, digital background calibration, bootstrapped calibration, CMOS analog integrated circuits.

I. Introduction

Some high-performance digital communication systems require high-speed, high-accuracy ADCs. Modern wireless basestations, for example, require fast converters to digitize large numbers of channels simultaneously, and they require highly linear converters to provide the large SFDR needed to discern weak, distant signals in the presence of strong, nearby signals. In practice, wireless basestation converters require speeds of at least 65 Msample/s and SFDR of at least 80 dB across the entire operating range of the converter [1].

Pipelined ADCs can potentially provide this performance but are limited by their reliance on precision analog signal processing [2]. With a 1.5-bit/stage architecture, pipelined ADC linearity depends almost entirely on the accuracy and variation of the gain of the SHAs, which are increasingly difficult to build in scaled CMOS technologies.

Most previous amplifier calibration algorithms corrected constant SHA gain errors [3–12]. Calibration to overcome signal-dependent gain error was introduced recently [13]. The overall linearity of that ADC was limited by uncalibrated nonlinearity in its input SHA

*This work was supported by SRC Grant 2000-HJ-787 and by UC MICRO Grant 02-040.

and its backend ADC. Also, the time required for the calibration to converge in that work is large because the input signal appears as uncorrelated noise much larger in amplitude than the calibration signal to the calibration loop, requiring millions of samples to average out the input. This paper describes a prototype pipelined ADC in which every stage, including the input SHA, is calibrated, allowing the requirements of cellular basestation converters to be met without any high-gain opamps [14]. Furthermore, the calibration reduces nonlinearity from finite slew rate, and the number of samples required to converge the calibration loops is reduced because the calibration test signal is applied when the input signal is not being converted. A highly accurate test signal generator is not required for the algorithm because the calibration is bootstrapped.

This paper is organized as follows. The calibration architecture is introduced in Section II. The proposed calibration algorithm is presented in Section III. Section IV describes the prototype. Measured results from the prototype are presented in Section V, and Section VI gives the conclusion.

II. Calibration Architecture

A. Queue-Based Calibration

Fig. 1(a) shows a block diagram of an ADC with a calibration queue [4]. The queue consists of one or more SHAs. The input $x(t)$ is sampled at a rate of f_S, and the ADC is clocked at a rate of f_C. To do background calibration, $f_C > f_S$ is chosen. Fig. 1(b) shows a simplified timing diagram. Impulses in "Sample" show times when the queue begins sampling the input. Impulses in "Convert" show times when the ADC begins a conversion. "Full" is high when the queue holds an input sample. Once a sample is completely transferred to the ADC, Full is reset. If $f_C > f_S$, the interval during which samples are held in the queue decreases over time at first. When the ADC is available to begin a conversion while the queue is empty, "CAL" rises, and the ADC instead converts a calibration signal. ADC outputs corresponding to samples of the calibration signal are separated from ADC outputs corresponding to samples of $x(t)$. The samples of the calibration signal are used to calibrate the ADC, and the ADC appears to uniformly sample the input $x(t)$ at a rate of f_S while doing background calibration at a rate of $f_C - f_S$. If

$f_C = f_S$, calibration does not occur in the background and is done in the foreground by interrupting the normal input $x(t)$.

The idea of an input queue was originally proposed and implemented with two SHAs [4]. The queue was shortened to only one SHA in [5] to reduce the input noise. Both previous implementations of the calibration queue were applied to algorithmic ADCs. In this work, the calibration queue is applied to a pipelined ADC for the first time. Applying a queue with one SHA to a pipelined ADC requires that f_C is at least one and a half times larger than f_S [5]. This calibration architecture allows the input and calibration signals to be quantized separately by the ADC. As a result, the input does not have to be averaged out during the calibration, reducing the time required for convergence.

B. SHA Closed-Loop Gain in Pipelined ADCs

Fig. 2 shows the block diagram of a general pipelined ADC. Each stage except the last contains an analog-to-digital subconverter (ADSC), a digital-to-analog subconverter (DASC), an analog subtracter, and a SHA with gain G. Once the pipeline is full, the SHAs in the stages allow all the stages to operate concurrently on analog residues that correspond to different samples of the input, allowing a high throughput rate. With redundancy and digital correction in a 1.5-bit/stage configuration, pipelined converters are insensitive to offset errors in their ADSCs and residue amplifiers, and errors mainly stem from gain errors in the SHAs [15].

In a general feedback amplifier with open-loop gain A_{ol} and feedback factor f, the closed-loop gain $A_{cl} \approx 1/f$ if $A_{ol}f \gg 1$. Under this condition, the closed-loop gain is primarily determined by the feedback factor, which is well controlled in practice. When $A_{ol}f$ is not much larger than one, however, variation in A_{ol} can have a significant effect on the variation of A_{cl}, as shown by the following well-known expression [16]:

$$\frac{dA_{cl}}{A_{cl}} = \frac{1}{(1 + A_{ol}f)} \frac{dA_{ol}}{A_{ol}}. \tag{1}$$

In modern CMOS processes, scaling reduces power-supply voltages, decreasing open-loop gain and increasing both the closed-loop gain error and the closed-loop gain variation in part by reducing the number of levels of cascodes that can be used. Traditionally, these problems have been overcome by increasing analog circuit complexity. In this work,

simple analog circuits are used to maximize the speed, and gain errors and gain variation are overcome using digital calibration. This approach shifts the design complexity from the analog to the digital domain, reducing cost in modern processes.

A single-stage, telescopic cascode opamp using the same structure as the opamp in [4] is used in this work. SPICE simulation of small-signal opamp open-loop gain versus differential opamp output shows that the maximum gain is about 360 for zero differential output and drops by about 10% at full swing of ±0.5V. This gain variation stems from transistors moving toward the triode region as the magnitude of the differential output voltage increases. Reducing the supply voltage increases the gain variation. This problem is often overcome with a multistage opamp [17,18], but single-stage opamps are faster than multistage opamps for a given bias current. Without calibration, the gain of 360 would limit the ADC linearity to about 8 bits, and the gain variation would further reduce the ADC linearity to about 7 bits.

Fig. 3 shows a plot of V_{od}/V_{ref} versus time from SPICE simulation. V_{od}/V_{ref} is the differential residue-amplifier output normalized to the reference voltage. The plot assumes that the input to the residue amplifier is stepped from 0 to its maximum value at $t = 0$. The output jumps down at first because of capacitive feedforward and then gradually approaches its final value of unity. Sampling the output at time $t = T_S$ before the output reaches its final value creates an error as labeled in the figure. If V_{od} is a linear function of the input, the error is simply a constant gain error. Interstage gain calibration has been used for several years to compensate for this error [6]. On the other hand, if the output is slew-rate limited, then V_{od} is not a linear function of the input, and the error is not simply a constant gain error. Instead, it can be modeled as a gain variation.

Fig. 4 shows SPICE simulation results with a simple opamp model demonstrating this effect. It is a plot of the effective gain versus the final output normalized to the reference voltage. The effective gain is the ratio of the sampled output to the input change and includes the effect of incomplete settling. In this simulation, the opamp had a constant dc gain and the capacitors were ideal, so the curvature seen here stems from finite slew rate. Increasing the output magnitude increases the time spent slewing and reduces the effective gain. The amount of gain variation here increases as the sampling rate increases,

and this effective gain can be modeled as an even-order function of the opamp output voltage, and terms higher than second order are significant here.

C. Modeling the Residue Amplifier

A simplified half-circuit of the fully differential residue amplifier is shown in Fig. 5 [2]. This topology is used rather than the more commonly used feedforward topology [8, 19] because the subtraction of the DASC output from the input signal in Fig. 5 occurs before the gain for this residue amplifier, so the signal and the DASC output see the same gain. This is important because it simplifies the modeling of the residue amplifier.

The open-loop gain of the opamp in Fig. 5 is modeled as a second-order function of its output because simulations show that the second-order term is dominant. In practice, feedback reduces but does not eliminate nonlinearity, and second-order variation in the open-loop gain causes mainly third-order nonlinearity in the closed-loop residue amplifier. However, calculating this nonlinearity from open-loop parameters is not necessary from a calibration standpoint. Instead, the goal is to linearize the ADC. A simple way to proceed is to model the closed-loop gain as a second-order function of the residue-amplifier output. This approach is reasonable because it also leads to mainly third-order nonlinearity in the residue amplifier. Then since the residue amplifier in a given stage of a pipelined ADC operates on the difference between the stage input and the corresponding DASC output:

$$V_{od} = (V_{id} - V_{DASC})(G + BV_{od}^2), \tag{2}$$

where V_{od} is the differential output, V_{id} is the differential input, and V_{DASC} is the differential DASC output in the pipelined stage under calibration (ideally 0 or ± 0.5 V in a 1.5-bit stage with ± 1V as full scale). $G + BV_{od}^2$ is the closed-loop gain of the stage. G models the zero-output closed-loop gain, and B models the closed-loop gain variation. The calibration concept here is to digitize the residue output V_{od} of the stage under calibration using the backend of the pipeline and then calculate the input corresponding to the digitized output based on accurate estimates of G and B [3, 4, 12]. Since (2) models closed-loop gain as a second-order function of the output, and since finite slew rate generates second-order and higher gain nonlinearity terms, this model can be used to partially calibrate for errors stemming from finite slew rate.

III. Calibration Algorithm

The ADC has 18 stages. Stages 1 and 2 have independent correction for constant and signal-dependent gain errors. Stages 3 through 6 have independent correction for constant gain errors, and the last 12 stages share one common correction for constant gain errors. Also, the input SHA or queue is calibrated by using coefficients from the SHA in the first ADC stage. The accuracy of this approach is limited by mismatch between these two SHAs. To reduce the mismatch, the input SHA is designed to have a nominal gain of two instead of one that is usually used. However, increasing the gain of the input SHA from one to two reduces the maximum signal-to-noise ratio that can be achieved.

A. Estimation of Model Parameter G

Fig. 6 is a plot of the residue characteristic of a 1.5-bit stage. The constant-gain estimate G is found by modifying a method presented by Karanicolas et al. [3]. The input during calibration is set about equal to a comparator threshold. In Fig. 6, let $V_{id}/V_{ref} \approx -0.25$. The comparator output is forced both high and low in successive calibration cycles, yielding digital outputs $D1$ and $D0$ as shown in the diagram. The gain estimate is updated using

$$G_e[j+1] = G_e[j] - \mu_g(D1 - D0). \tag{3}$$

Negative feedback drives the difference between $D1$ and $D0$ to zero. This difference should be zero because the redundancy in the 1.5-bit/stage architecture makes the ADC output independent of the result of this comparator for inputs near the comparator threshold [12]. Also, both comparator thresholds in each stage are tested, and the resultant outputs are averaged to make a first-order correction for inaccuracies caused by common-mode to differential-mode conversion [12].

The above algorithm corrects for gain error due to capacitor mismatch and low opamp gain, but cannot correct errors due to non-constant opamp gain. It is capable of providing at least 15-bit linearity in cases where gain variation is insignificant [3]. Here, the linearity after calibration is limited by gain variation. Information about the gain variation can be used to improve the accuracy of the calibration. This information can be acquired using a separate calibration loop.

B. Estimation of Model Parameter B

Fig. 7 shows two ADC transfer characteristics. The top plot is ideal, and the bottom plot shows an example with nonlinearity, where the nonlinearity is exaggerated for clarity.

To estimate B, a calibration DAC applies test signals to the input of the stage under calibration, and the backend of the ADC measures the analog output of the stage. The outputs of the calibration DAC should be as far as possible from the outputs used to measure constant gain errors because the goal here is to measure the variation in gain caused by a variation in the input. The nonlinearity is maximum at full scale. However, with full-scale test signals, nonzero offset would cause clipping for one of the measurements. (Offset would also upset the symmetry of the gain curvature, but this can be corrected by measuring the offset of each stage digitally and then subtracting it during normal operation.) To maximize the test signal amplitude while avoiding such clipping, test signals of ± 0.875 of full scale are used. The resulting measurements are called $D3$ and $D2$, and the difference $D3-D2$ is labeled in Fig. 7. If the DAC is ideal, the ideal difference between $D3$ and $D2$ is known in advance and is called $BDIST$. $D3-D2$ is then compared to $BDIST$ and used to update the estimate of B as follows

$$B_e[j+1] = B_e[j] + \mu_b(D3 - D2 - BDIST). \tag{4}$$

A problem with this approach is that errors in the calibration DAC will change the $\pm 0.875 V_{ref}$ outputs and limit the accuracy of the B estimate.

C. Bootstrapped Calibration

To overcome this problem, the calibration is bootstrapped, which means that the DAC is used to calibrate the ADC, and the ADC is used to calibrate the DAC. Bootstrapping reduces the accuracy requirements on the calibration DAC.

As mentioned at the end of Section II-C, the input is calculated from the digitized output to calibrate a stage. When the calibration includes both constant and signal-dependent gain as in (2), the G and B values must be known accurately. Since the values of G and B are not known in advance, estimates G_e and B_e are used instead. These estimates are forced to converge to the values of G and B by an adaptive process using (3) and (4). At

each step in the process, the most recent values of G_e and B_e are used. A calibration cycle consists of a number of updates of the estimates G_e and B_e. The first calibration cycle consists of four updates of G_e followed by a measurement of $BDIST$ but no updates of B_e. All other calibration cycles consist of 1280 calibration operations: 1024 updates of G_e (based on measurements of $D1$ and $D0$ at both thresholds) followed by 256 updates of B_e (based on measurements of $D2$, $D3$, and $BDIST$). The values 1024 and 256 were chosen empirically to minimize the time required for convergence.

Fig. 8 shows the results of a C language simulation of the convergence of G_e and B_e for the second stage with $G = 1.961$ and $B = -0.031$, respectively. At first, both G_e and B_e are far from their correct values, so $D1 - D0$ and $D3 - D2$ are far from ideal, and the estimates change significantly. As the calibration continues, improved estimates reduce $D1 - D0$ and $D3 - D2 - BDIST$ until they both are equal to zero, and the algorithm has converged. The process by which the estimates converge is described next.

Using the G_e and B_e estimates to solve (2) for V_{id} gives

$$V_{id(cal)} = \frac{V_{od}}{G_e + B_e V_{od}^2} + V_{DASC} \tag{5}$$

This result is called $V_{id(cal)}$ to distinguish it from V_{id}, which represents the analog input to the stage for this calibration. Fig. 9 shows a flow chart of the bootstrapped calibration algorithm. In step ①, the loop that finds G_e is activated with $B_e = 0$. The calibration DAC sets $V_{id} \approx \pm 0.25 V_{ref}$ (a comparator threshold). This loop is not sensitive to the exact value of V_{id} because the calibration principle is to adjust G_e until the ADC output does not depend on the comparator decision [12]. This principle holds when the input is within the digital correction range of the comparator threshold. Assume at first that the initial value of $G_e < G$ as shown in the example in Fig. 8. Then, updating G_e increases it to a value that is closer to but less than G.

In step ② of Fig. 9, the calibration DAC sets $V_{id} = V_{cd+} \approx 0.875 V_{ref}$. The stage output is digitized and used to calculate the corresponding value of $V_{id(cal)}$. Since B_e=0 the first time step ② is started, $V_{id(cal)} = V_{od}/G_e + V_{DASC}$ from (5). This result is called $V_{id(cal)+}$. Then the calibration DAC sets $V_{id} = V_{cd-} \approx -0.875 V_{ref}$ and calculates another value of $V_{id(cal)}$, called $V_{id(cal)-}$. $BDIST = V_{id(cal)+} - V_{id(cal)-}$ is calculated, giving an offset-insensitive measurement of the outputs of the calibration DAC.

In step ③, the loop that finds G_e is reactivated, and the G_e estimate increases as G_e is updated 1024 times. Then in step ④, the loop that finds B_e is activated, and $V_{id(cal)+}$ and $V_{id(cal)-}$ are found again using the new, larger value of G_e. In this case, $V_{id(cal)+} - V_{id(cal)-}$ is called $D3-D2$. The estimate B_e is updated using (4) and the value of $BDIST$ that was measured in step ②. Then $V_{id(cal)+}$ and $V_{id(cal)-}$ are measured 255 more times, updating B_e after each measurement using the value of $BDIST$ measured in step ②. Since G_e has increased from its value when $BDIST$ was measured, $D3 - D2 < BDIST$, and B_e becomes negative as a result of these updates as shown in Fig. 8.

In step ⑤, the G_e loop is activated again, with the new, negative B_e. Assuming the actual circuit gain when $|V_{id}| \approx 0.25V_{ref}$ is unchanged if V_{id} is unchanged, the negative B_e causes the total gain estimate $(G_e+B_eV_{od}^2)$ to decrease. Therefore, 1024 updates of G_e now produces a larger value of G_e than when $B_e = 0$. Then $BDIST$ is measured again in step ②, G_e is updated in step ③, and the loop that updates B_e is reactivated in step ④. If G_e increases in step ③, the total gain estimate with $V_{id} = V_{cd+}$ or $V_{id} = V_{cd-}$ has increased. Again assuming the actual gain of the circuit has not changed, the B_e loop now produces a more negative value of B_e. G_e and B_e continue to evolve until $D3 - D2 = BDIST$, which happens when G_e is the same for the $BDIST$ and $D3 - D2$ calculations.

The above reasoning assumes that $G_e < G$ at first, but this assumption is not required. If $G_e > G$ is assumed initially, then the G_e loop will decrease G_e at first. Therefore, $BDIST < D3 - D2$ after this change of G_e assuming $G_e > G$. As a result, $B_e > 0$ after the first update of B_e. Positive B_e increases the total gain estimate. Because the G_e loop tries to compensate for B_e, G_e eventually becomes less than G in order to lower the total gain estimate. As a result, B_e decreases and eventually becomes negative. Then the system converges as described above. However, if B_e passes through zero during convergence as in this case, adaptation could stop at that point with G_e not equal to its desired value G. This situation is similar to the presence of undesired operating points in self-biased reference circuits [22], which are avoided by startup circuits in practice [16]. One way to overcome this problem would be to restart the calibration with a lower initial G_e estimate if B_e crosses zero during convergence.

D. Algorithm Convergence

The convergence of the G estimate is similar to that in [23]. In that work, the gain estimate of an algorithmic ADC was shown to converge for $G_e > 1.7$ if $\mu_g < 0.514$. The convergence of the B estimate can also be shown in a similar way. B_e can be shown to converge for $G_e - B_e > 1.7$ if $\mu_b < 10.3$. Such a large μ_b is possible here for two reasons. First, μ_b is multiplied by a small number ($D3 - D2 - BDIST$), so the actual update step is small. Second, because B_e is small, it has a small effect on the total gain of the stage, so large step sizes can be used without destabilizing the calibration loops. In addition to the individual convergence of G_e and B_e, the co-convergence of G_e and B_e must be considered. To use a calibration DAC with relaxed performance, bootstrapped calibration is used as described in the previous section. Because bootstrapped calibration relies on the measurement of G_e to improve the measurement of B_e and vice-versa, it must be shown that this co-dependent convergence avoids local minima to find the optimal gain for the system. To show this, it is sufficient to show that unless $D1 - D0 = 0$ and $D3 - D2 = BDIST$ (i.e. the system is converged) then a change in B_e will cause a change in G_e and vice-versa. For N updates of B_e and a step size of μ_b, then $\mu_b N > 1$ LSB is required for G_e to be altered when $D3 - D2 \neq BDIST$. Otherwise a changing B_e does not guarantee G_e will change and convergence may terminate at a local minimum. In the prototype various values of μ_b and N can be used. For N=256, for example, $\mu_b > 0.004$ ensures that local minima will be avoided. Similar reasoning shows that local minima will be avoided if $\mu_g > 0.004$ provided that M, the number of G loop iterations, is 256.

The prototype uses $\mu_b = \mu_g = 2^{-5}$. These values are chosen by simulation and are significantly smaller than the maxima found above. Increasing μ_b and μ_g allows faster convergence and tracking [21] but reduces the effectiveness of bootstrapping because the loops converge so quickly they are not able to interact as much.

E. Required Component Accuracy

This calibration approach uses the backend of the ADC to make measurements on the stage under calibration. Traditionally, the backend has had to be at least as accurate in an average sense as the result required in the stage under calibration [6]. One useful aspect

of this algorithm is that it can partially correct residual backend nonlinearity because the backend is used to measure gain variation in the stage under calibration. So the calibration result depends in part on residual backend nonlinearity. The B estimate loop corrects third-order INL errors in the stage under calibration, so third-order residual errors in the backend can be corrected. The algorithm does this by converging its gain estimates to values that satisfy the convergence criteria, not to the true gain values. As a result, the algorithm can adjust its estimates of the parameters that model the gain of a given stage to correct errors in downstream stages. Therefore, the backend can be less accurate than its resolution, and simulations show that only 7-bit opamp gain accuracy is required to achieve a 12-bit linear ADC.

IV. Prototype

Fig. 10 shows a block diagram of the prototype. The analog components in the ADC are shown inside the dashed box. The decisions from the 16 stages that receive calibration for constant-gain errors only go through the Radix Converter to convert this raw digital output to base 2. The output of the Radix Converter is a digital estimate of the output of the second pipeline stage, which is the last stage that receives a correction for signal-dependent gain error. The 14 MSBs of the estimate are sent to an off-chip dual-port SRAM, which is used as a lookup table. These 14 MSBs, when combined with the decision bits of the first two stages, address the lookup table to find the calibrated ADC output D_{OUT}. The backend estimate is also sent to the Calibration Engine, an on-chip 28-bit custom microprocessor that has two functions. First, it calculates the powers of G used in the Radix Converter to correct the backend stages, and it calculates the values stored in the lookup table. Second, between updates, it performs all the calculations required to converge on the gain estimates for all the stages. By designing a microprocessor and implementing the algorithm in software, many different calibration algorithms can be tested simply by changing the software, simplifying the design of the prototype for testability.

The DAC used to generate the test signals is shown in Fig. 11. A single-ended structure is shown for simplicity. On the prototype, this DAC and all the analog circuits are fully differential. To generate a voltage of $0.875V_{ref}$, seven capacitors are charged to V_{ref} and one capacitor is discharged during ϕ_1. All eight capacitors are then switched to the output

during ϕ_2. A different capacitor is discharged each time a test signal is required. Variation in the test signal voltage arises from this rotation due to capacitor mismatch [20]. Test-signal variation does not overcome errors from finite opamp gain in Fig. 11. Instead, it dithers the calibration system, helping the algorithm converge to an accurate B estimate with limited resolution in the remaining stages.

A die photograph of the prototype is shown in Fig. 12. The prototype was fabricated in 0.25-μm CMOS. In the prototype, digital circuits use about 75% of the area because they were designed to be highly reliable and programmable to allow thorough testing of the prototype in the first and only silicon. The area of the digital circuits could be reduced by about 85% and the power dissipation by about 75% by using a nonprogrammable approach and a more aggressive design in the same process technology [24].

V. Measured Results

All measurements correspond to one of three modes for the prototype ADC. The first mode is no calibration, where no stages are calibrated. The second mode is constant-gain calibration, where $B_e = 0$ for the first two stages, so the first six stages receive independent constant-gain estimates and the final 12 stages share a single constant-gain estimate. With constant-gain calibration, the gain error in the input SHA is corrected, assuming its gain is the same as the gain of the SHA in the first ADC stage. The final mode is full calibration, where the first two stages use independent estimates for gain variation and constant-gain errors, the next four stages use independent constant-gain estimates and the final 12 stages share a single constant-gain estimate. With full calibration, the input SHA is also corrected using G_e and B_e from the first ADC stage.

Code-density testing was used to determine the linearity of the prototype [25]. The measured DNL at a 12-bit level is shown in Fig. 13(a) for the case of no calibration, in Fig. 13(b) for constant-gain calibration only, and in Fig. 13(c) for full calibration. The linear input range of the converter is limited to the range of the calibration test signal used to measure stage nonlinearity in Fig. 7. The codes with DNL$= -1$ LSB in the no-calibration and constant-gain calibration cases are eliminated in the full-calibration case. Peak DNL in the full-calibration case is 0.09 LSB at a 12-bit level, which indicates the calibration has achieved 14-bit DNL performance.

The measured INL at a 12-bit level is shown in Fig. 14(a) for the case of no calibration, in Fig. 14(b) for constant-gain calibration only, and in Fig. 14(c) for full calibration. Significant improvement is seen due to calibration. Peak INL in the full-calibration case is 0.24 LSB at a 12-bit level. The INL profiles have odd symmetry, and the cases for no calibration and constant-gain calibration are dominated by their third-order components, as would be expected with a second-order gain variation. The case with full calibration is not dominated by its third-order component, which indicates the prototype has been successful in eliminating errors caused by the second-order gain variation.

A. Foreground Calibration

FFT plots of the ADC are shown in 15(a) for the case of no calibration, Fig. 15(b) for constant-gain calibration only, and in Fig. 15(c) for full calibration. In these tests, the input is a 0.5 V peak-to-peak 50-kHz sinusoid, and the sampling rate is 80 MS/s. Also, in these tests, the ADC is calibrated in the foreground (which means that the sample rate of the input SHA, f_S, is equal to the conversion rate of the ADC, f_C). The plot extends to 625 kHz because the output is downsampled by a factor of 64 on-chip to enable the use of a serial interface. The SNDR improves by 13 dB between the no-calibration and constant-gain calibration cases and by another 9 dB when full calibration is applied. By design, the SNDR here is limited by noise and achieves almost full 12-bit performance. Simulations show that this SNDR cannot be achieved with mismatch of more than 1% between the input SHA and the SHA in the first ADC stage. The SFDR increases by 14 dB between the no-calibration and constant-gain calibration cases, but it is still limited by strong tones at harmonics of the input frequency. Full calibration improves the SFDR by another 18 dB to 84.5 dB, which is sufficient to meet the specifications for wireless basestation converters. (Note that the SFDR is measured for the downsampled output. Downsampling reduces the frequency resolution and could alter the SFDR that would have been measured without downsampling.)

The SNDR and SFDR as a function of input frequency are shown in Fig. 16. The 3 dB loss in SNDR at Nyquist (40 MHz) is caused by about 10 ps rms jitter in the sampling clock. As the input frequency increases, the SFDR shows somewhat more degradation than the SNDR. Table I summarizes measured performance with foreground calibration.

B. Background Calibration

FFTs with background calibration are shown in Fig. 17. To allow background calibration with a calibration queue, the ADC conversion rate f_C has to be at least 50% faster than the input sampling rate f_S [5]. In the top plot, $f_S = 20$ MS/s, and $f_C = 30$ MS/s. The performance here is similar to that achieved with foreground calibration. However, if the sampling rate is increased beyond this point with background calibration, the performance falls.

For example, the bottom plot shows the performance with $f_S = 53.3$ MS/s and $f_C = 80$ MS/s. The SNDR and SFDR degradation stems from calibrating the input SHA with coefficients from the SHA in the first ADC stage while these two SHAs operate at different rates (f_S for the input SHA and f_C for the SHA in the first ADC stage). Under these conditions, the portion of the settling time used for slewing in the input SHA and in the first stage differ, causing a miscalibration of the input SHA at high rates with background calibration. Since the problem does not occur with foreground calibration where $f_S = f_C$, slew-rate limiting is partially corrected in the prototype with foreground calibration.

VI. Conclusion

This paper presents a 12-bit 80-MS/s pipelined ADC with digital background calibration. The calibration algorithm is the first to demonstrate correction of errors caused by finite slew rate. The algorithm uses bootstrapping, where the calibration DAC is calibrated by the ADC and vice-versa, which relaxes the requirements on the DAC. Also, the algorithm corrects for third-order residual nonlinearity, which means the backend ADC need not be as accurate as its resolution. The prototype uses the 1.5-bit/stage architecture, and all the stages, including the input SHA, receive digital calibration. As a result, a high-performance pipelined ADC has been built without any precision analog components.

Acknowledgments

The authors thank K. Dyer, V. Gopinathan, S. Jamal, J. Keane, J. Ming, and X. Wang for helpful technical discussions.

References

[1] L. Singer, S. Ho, M. Timko, and D. Kelly "A 12b 65MSample/s CMOS ADC with 82dB SFDR at 120MHz," *IEEE International Solid-State Circuits Conference*, pp. 38–39, Feb. 2000.

[2] S. H. Lewis and P. R. Gray, "A Pipelined 5-Msample/s 9-bit Analog-to-Digital Converter," *IEEE J. Solid-State Circuits*, vol. SC-22, no.6, pp. 954–961, Dec. 1987.

[3] A. Karanicolas, H.S. Lee, and K. Bacrania, "A 15-b 1Msample/s Digitally Self-Calibrated Pipeline ADC," *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1207–1215, Dec. 1993.

[4] O. E. Erdoğan, P. J. Hurst, and S. H. Lewis, "A 12b Digital-Background-Calibrated Algorithmic ADC with -90dB THD," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1812–1820, Dec. 1999.

[5] E. B. Blecker, T. M. McDonald, O. E. Erdoğan, P. J. Hurst, and S. H. Lewis, "Digital Background Calibration of an Algorithmic Analog-to-Digital Converter Using a Simplified Queue," *IEEE J. of Solid-State Circuits*, vol. 38, no. 6, pp. 1059–1062, Jun. 2003.

[6] J. Ming and S. H. Lewis, "An 8-bit 80-Msample/s Pipelined ADC With Background Calibration," *IEEE J. of Solid-State Circuits*, vol. 36, no. 10, pp. 1489–1497, Oct. 2001.

[7] J. M. Ingino and B. A. Wooley, "A Continuously Calibrated 12-b 10-MS/s, 3.3-V A/D Converter," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1920–1931, Dec. 1998.

[8] B.-S. Song, M. Tompsett, and K. Lakshmikumar, "A 12-bit 1-Msample/s capacitor error-averaging pipelined A/D converter," *IEEE J. Solid-State Circuits*, vol. 23, no.6, pp. 1324–1333, Dec. 1988.

[9] T.-H. Shu, B.-S. Song, and K. Bacrania, "A 13-b 10-Msample/s ADC Digitally calibrated with Oversampling Delta-Sigma Converter," *IEEE J. Solid-State Circuits*, vol. 30, no.4, pp. 443–452, Apr. 1995.

[10] R. Jewett, K. Poulton, K.-C. Hsieh, and J. Doernberg, "A 12b 128 MSample/s ADC with 0.05 LSB DNL," *International Solid-State Circuits Conference*, pp. 138–139, Feb. 1997.

[11] D. Fu, K. C. Dyer, S. H. Lewis, and P. J. Hurst, "A Digital Background Calibration Technique for Time-Interleaved Analog-to-Digital Converters," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1904–1911, Dec. 1998.

[12] X. Wang, P. J. Hurst, and S. H. Lewis, "A 12-bit 20-MS/s Pipelined ADC with Nested Digital Background Calibration," *IEEE Custom IC Conf.*, pp. 409–412, Sept. 2003.

[13] B. Murmann, and B. Boser "A 12b 75MS/s Pipelined ADC using Open-Loop Residue Amplification," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp.2040–2050, Dec. 2003.

[14] C. R. Grace, P. J. Hurst, and S. H. Lewis, "A 12b 80MS/s Pipelined ADC with Bootstrapped Digital Calibration," *IEEE International Solid-State Circuits Conference*, pp. 460–461, Feb. 2004.

[15] S. H. Lewis, H. S. Fetterman, G. F. Gross, Jr., R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-Msample/s Analog-to-Digital Converter," *IEEE J. Solid-State Circuits*, vol. 27, no. 3, pp. 351–357, Mar. 1992.

[16] P. R. Gray, P.J. Hurst, S.H. Lewis, and R. G. Meyer "Analysis and Design and Analog Integrated Circuits," New York, Wiley 2001.

[17] E. M. Cherry, "A New Result in Negative-Feedback Theory, and Its Application to Audio Power Amplifiers," *Int. J. Circuit Theory,* vol. 6, pp. 265–288, July 1978.

[18] J. H. Huijsing and D. Linebarger, "Low-Voltage Operational Amplifier with Rail-to-Rail Input and Output Ranges," *IEEE J. Solid-State Circuits,* vol. SC-20, pp. 1144–1150, Dec. 1985.

[19] S. Sutarja, "High-Resolution Pipelined Analog-to-Digital Conversion," UC Berkeley Doctoral Thesis, Memorandum No. UCB/ERL M88/27, pp. 40–42, May 1988.

[20] R. van de Plassche "Dynamic Element Matching for High-Accuracy Monolithic D/A Converters" *IEEE J. Solid-State Circuits*, vol. 11, no. 6, pp. 795–800, Dec. 1976.

[21] B. Widrow and S. D. Stearns, *Adaptive Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1985.

[22] V. Gopinathan, private communication.

[23] O. E. Erdoğan, "Digital Background Calibration of Analog-to-Digital Converters Using a Calibration Queue," UC Davis Doctoral Thesis, Memorandum No. UCD/SSCRL M99-5, Sept. 1999.

[24] C. R. Grace, "Digital Calibration of Interstage Gain Errors and Signal-Dependent Variations in Pipelined Analog-to-Digital Converters," UC Davis Doctoral Thesis, Mar. 2004.

[25] J. Doernberg, H.-S. Lee, and D. Hodges, "Full-Speed Testing of A/D Converters," *IEEE J. Solid-State Circuits*, vol. 19, no. 6, pp. 820–827, Dec. 1984.

Figure and Table Captions

Fig. 1: (a) ADC with calibration queue and (b) simplified queue timing diagram with background calibration (where $f_C > f_S$).

Fig. 2: Block diagram of pipelined ADC.

Fig. 3: Simulated step response of residue amplifier.

Fig. 4: Simulated effective residue amplifier gain with slewing. The closed-loop gain of the amplifier at $V_{od} = 0$ is 1.9256.

Fig. 5: Pipelined ADC residue amplifier.

Fig. 6: Residue characteristic of 1.5-bit stage.

Fig. 7: ADC transfer characteristic without (top) and with (bottom) nonlinearity.

Fig. 8: Simulated convergence of the gain estimates. The G and B values used in the simulation are 1.961 and -0.031, respectively.

Fig. 9: Flow chart of bootstrapped calibration.

Fig. 10: Block diagram of ADC and calibration system.

Fig. 11: Calibration DAC.

Fig. 12: Die photograph.

Fig. 13: Differential nonlinearity (DNL) with foreground calibration: (a) with no calibration, (b) with constant-gain calibration only, and (c) with full calibration. Sampling rate is 80 MS/s.

Fig. 14: Integral nonlinearity (INL) with foreground calibration: (a) with no calibration, (b) with constant-gain calibration only, and (c) with full calibration. Sampling rate is 80 MS/s.

Fig. 15: ADC output spectra with foreground calibration: (a) with no calibration, (b) with constant-gain calibration only, and (c) with full calibration. Sampling rate is 80 MS/s. Output downsampled by factor of 64.

Fig. 16: SNDR and SFDR versus input frequency. Sampling rate is 80 MS/s.

Table I: Performance Summary with Foreground Calibration (2.5 V, 25°C).

Fig. 17: ADC output spectra with background calibration: (a) Sampling rate is 20 MS/s, (b) Sampling rate is 53.33 MS/s. Output downsampled by factor of 64.

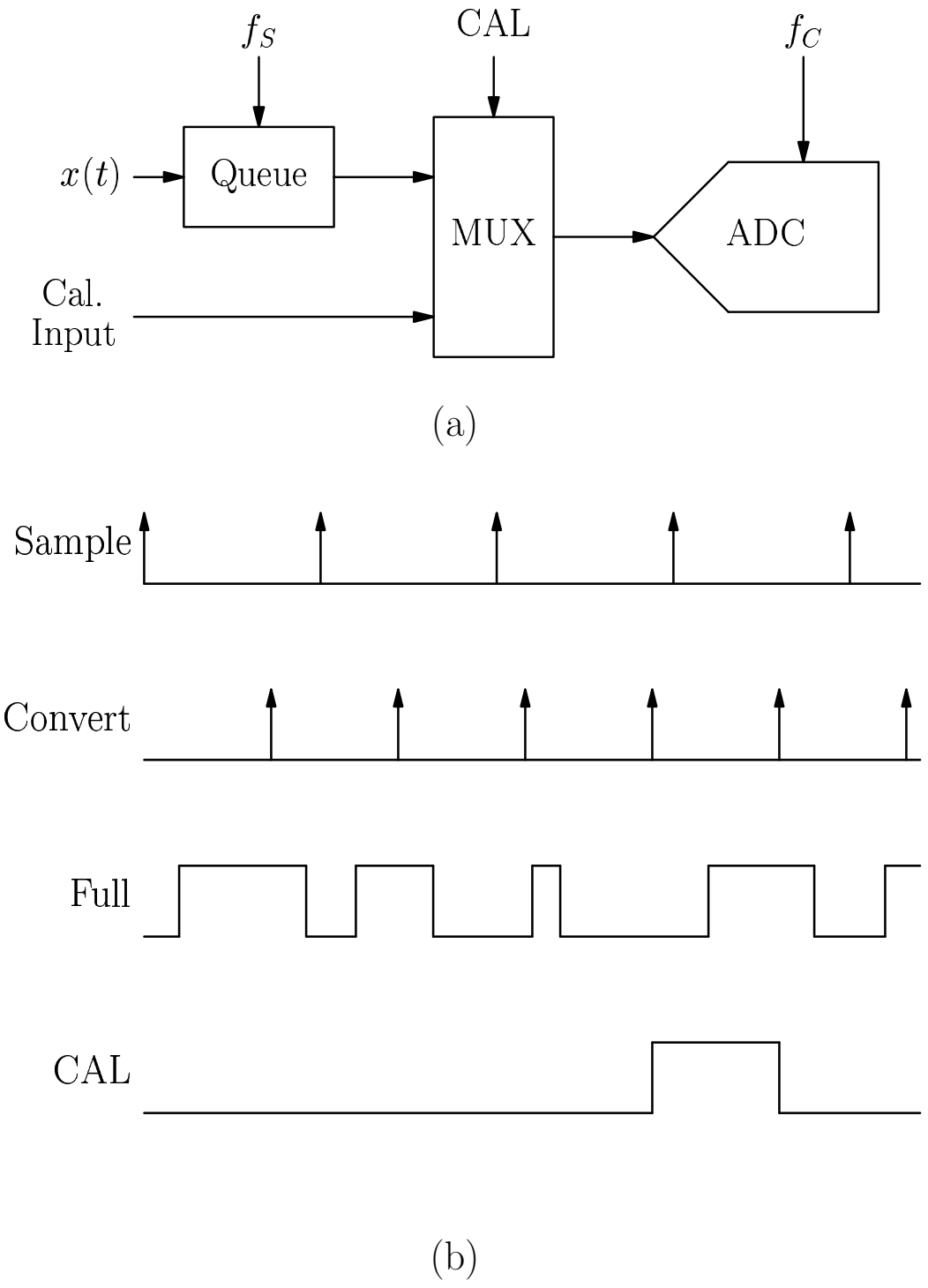


Fig. 1. (a) ADC with calibration queue and (b) simplified queue timing diagram with background calibration (where $f_C > f_S$).

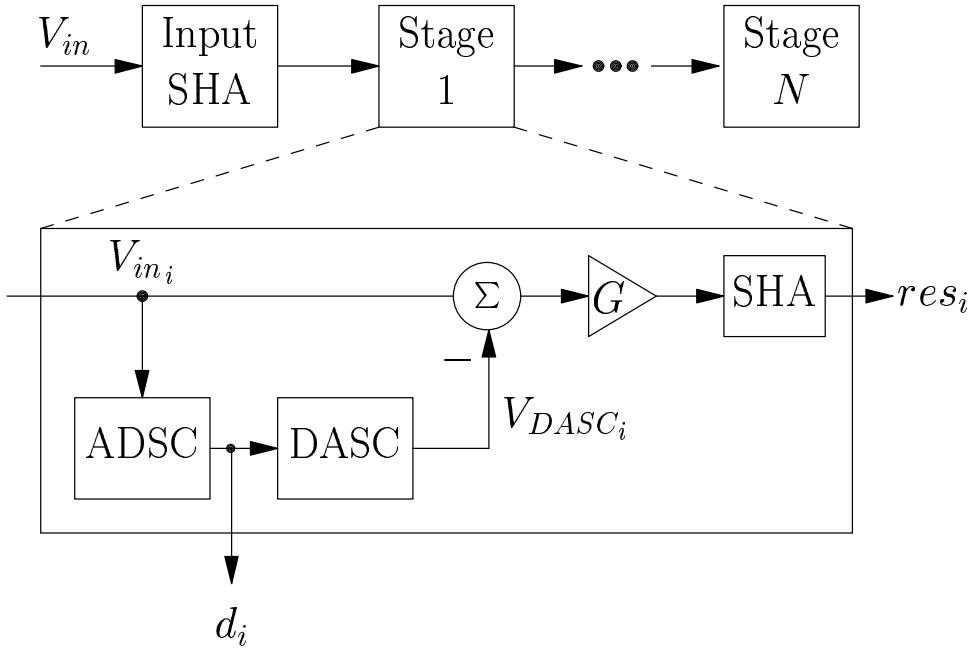


Fig. 2. Block diagram of pipelined ADC.

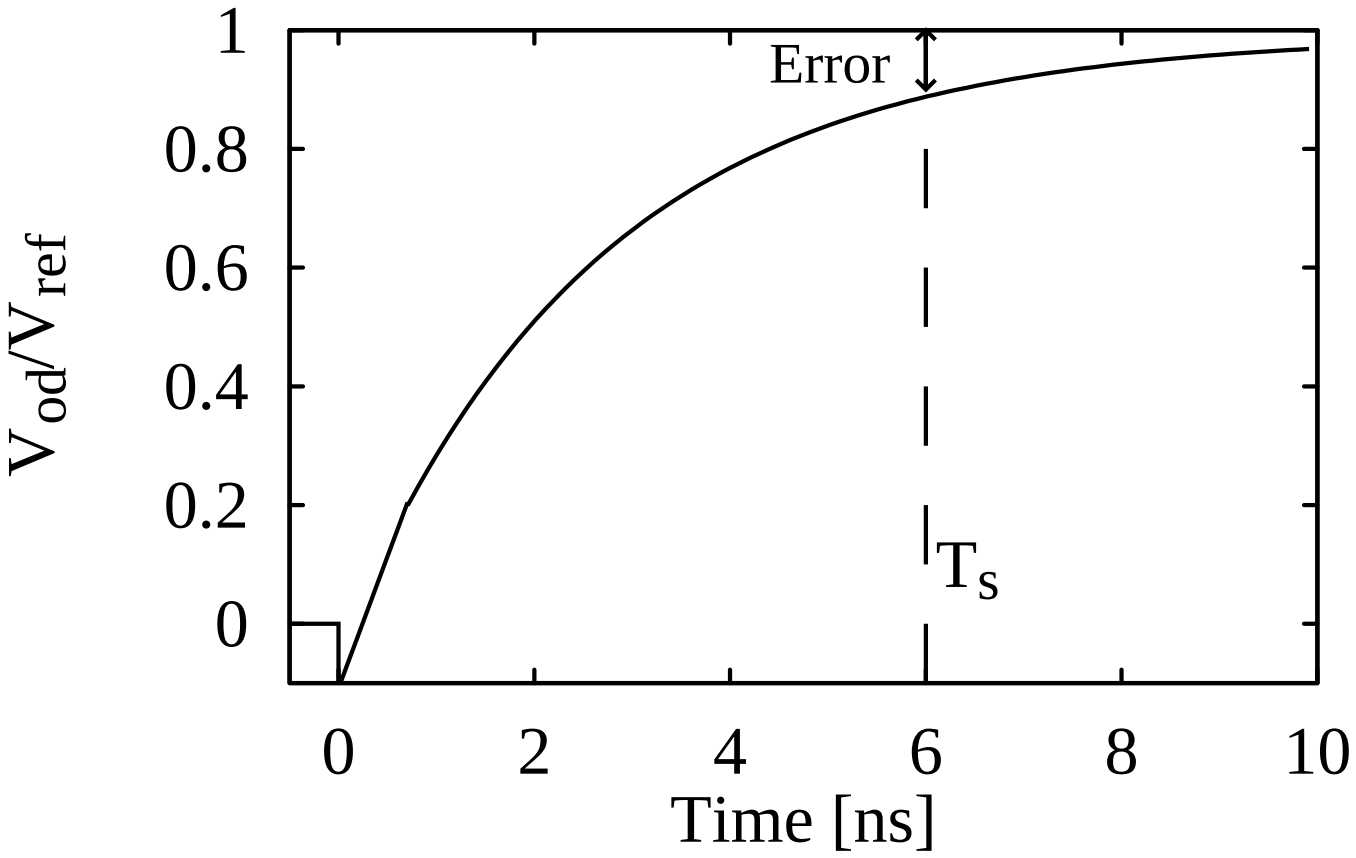


Fig. 3. Simulated step response of residue amplifier.

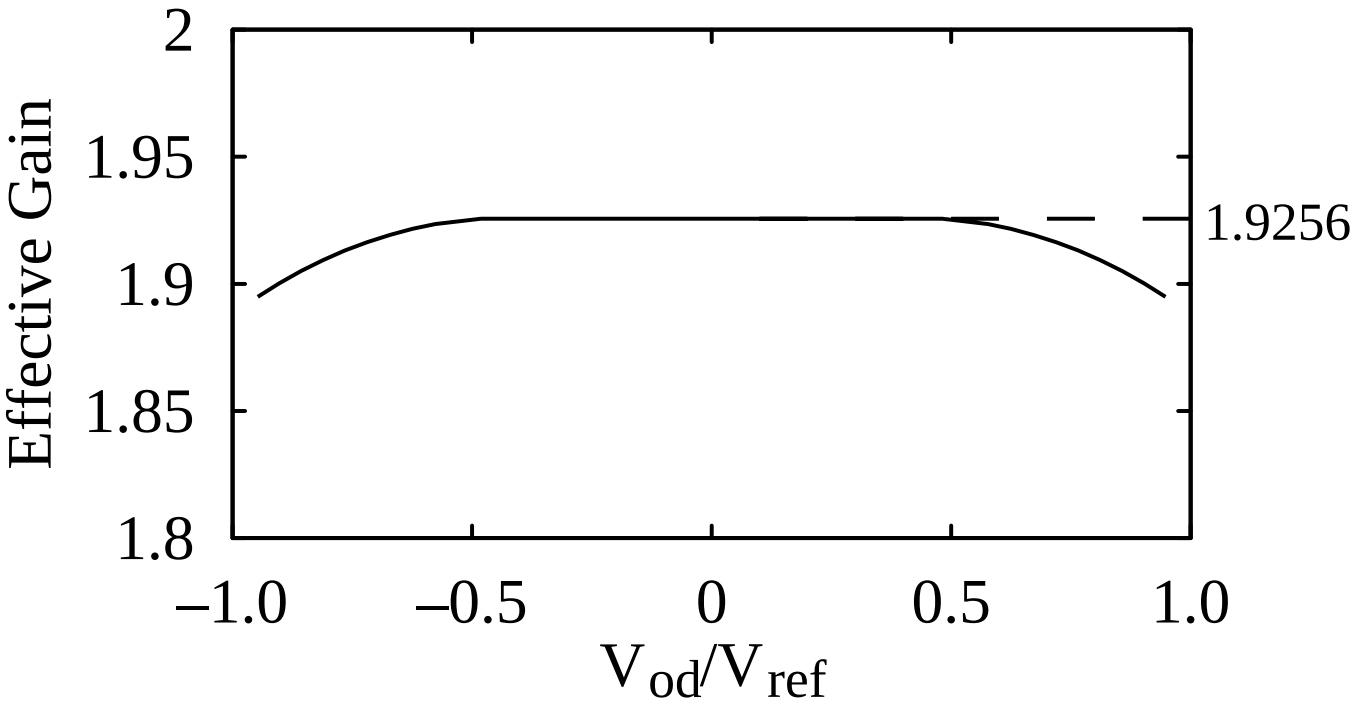


Fig. 4. Simulated effective residue amplifier gain with slewing. The closed-loop gain of the amplifier at $V_{od} = 0$ is 1.9256.

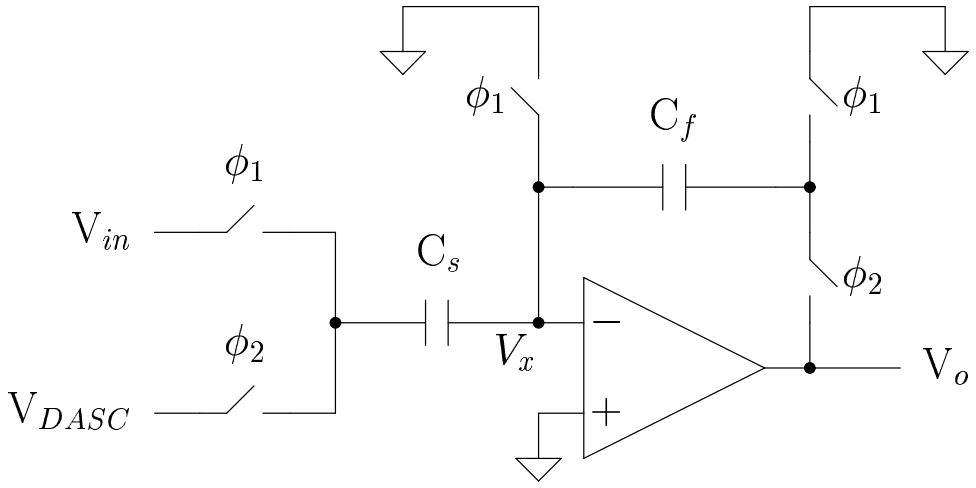


Fig. 5. Pipelined ADC residue amplifier.

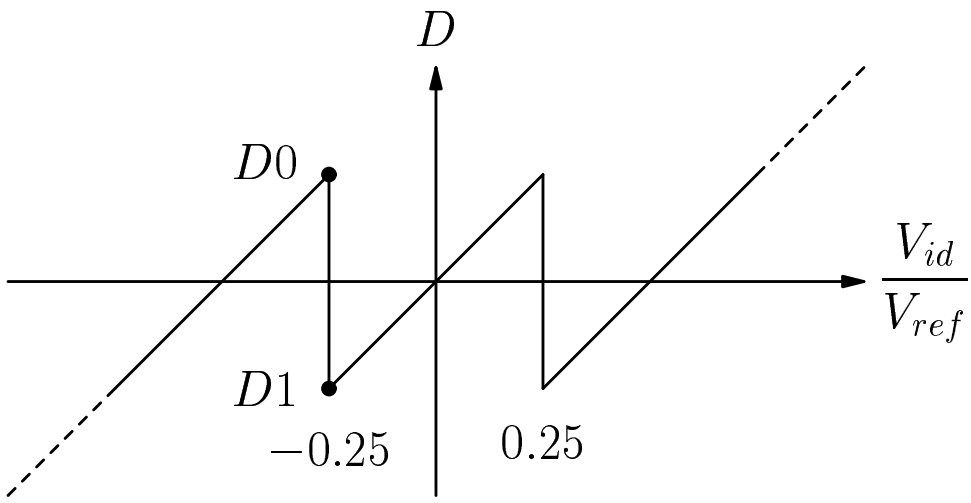


Fig. 6. Residue characteristic of 1.5-bit stage.

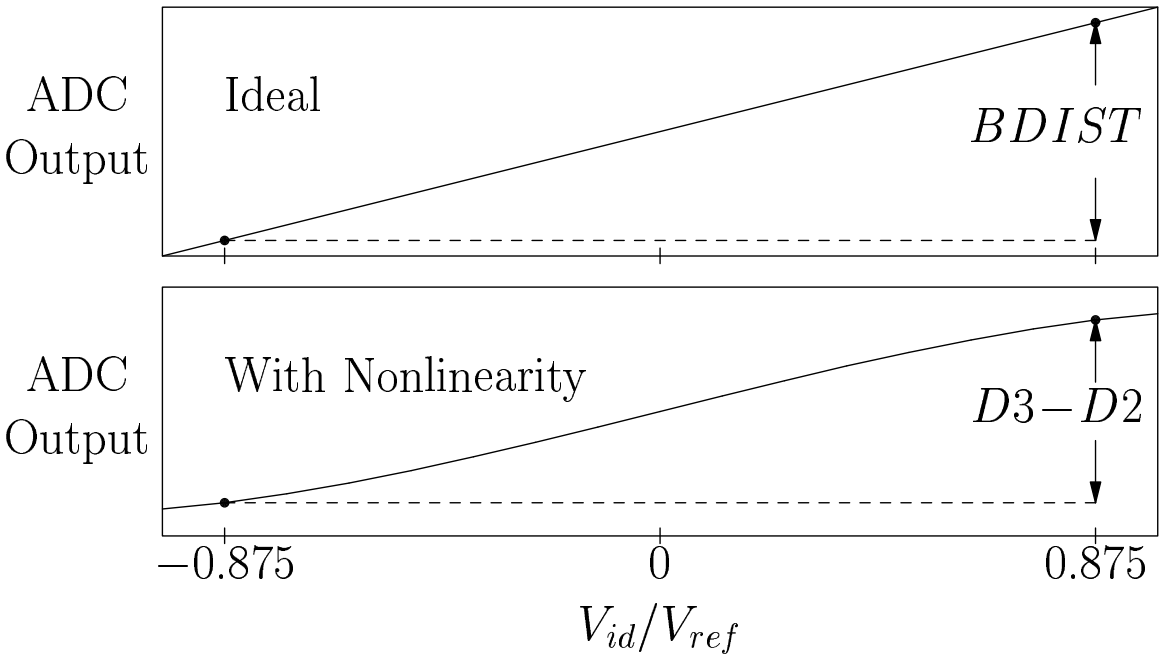


Fig. 7. ADC transfer characteristic without (top) and with (bottom) nonlinearity.

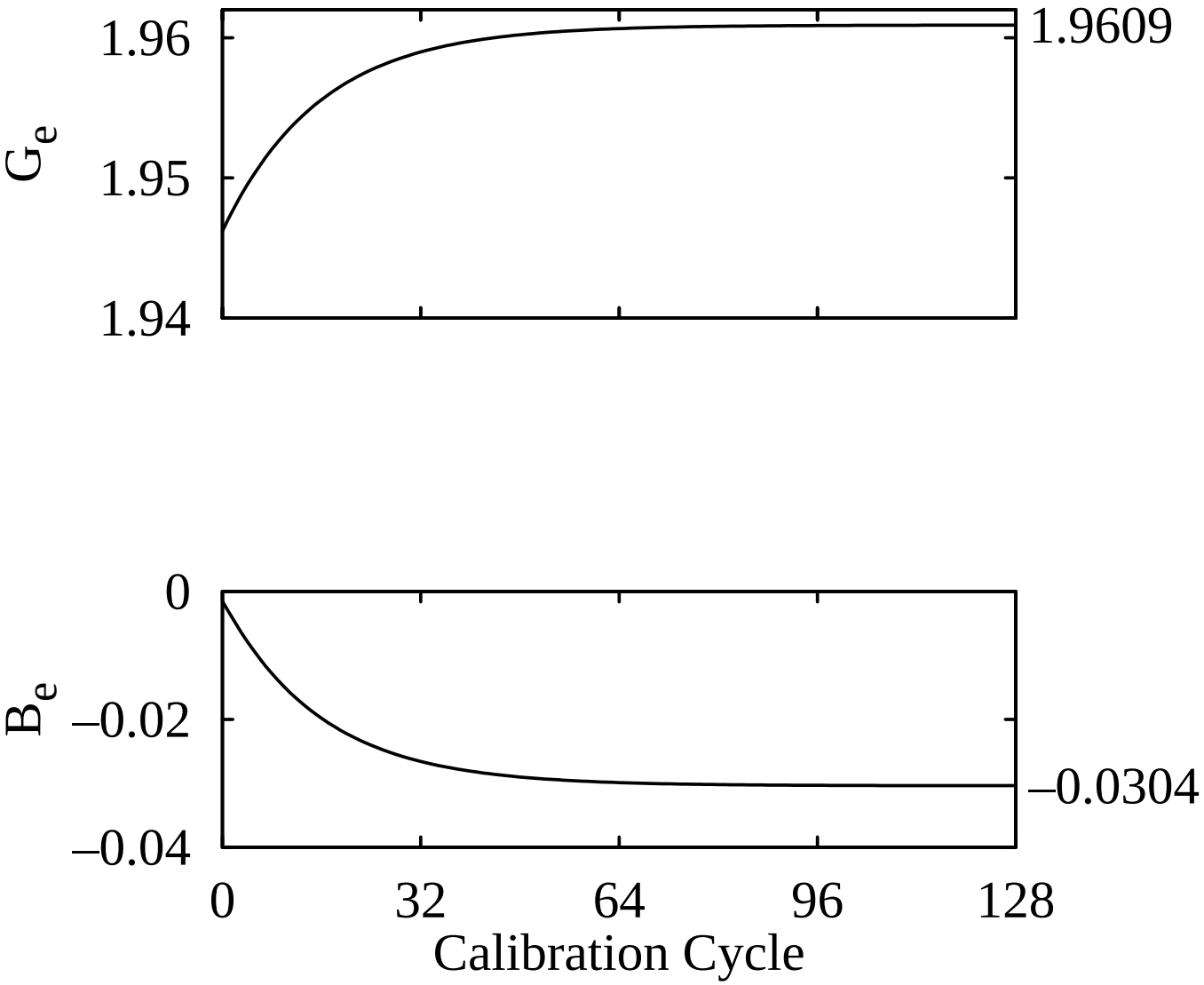


Fig. 8. Simulated convergence of the gain estimates. The G and B values used in the simulation are 1.961 and -0.031, respectively.

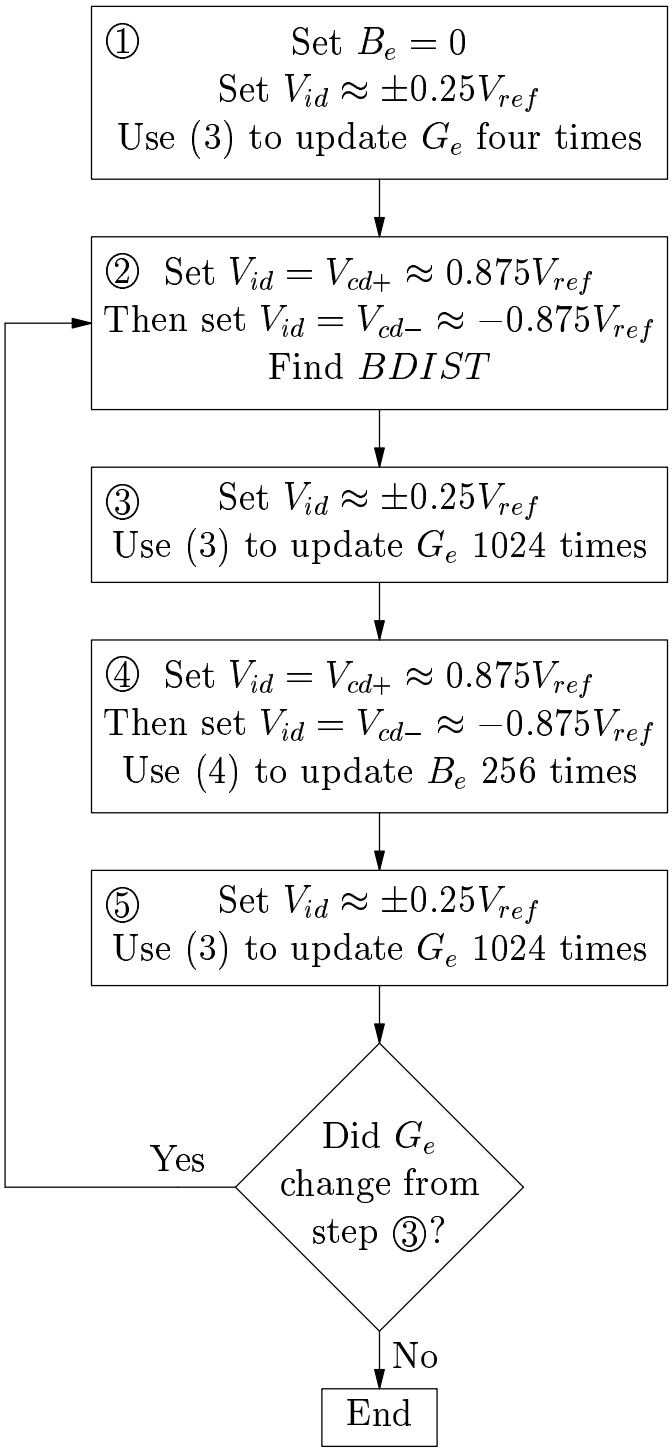


Fig. 9. Flow chart of bootstrapped calibration.

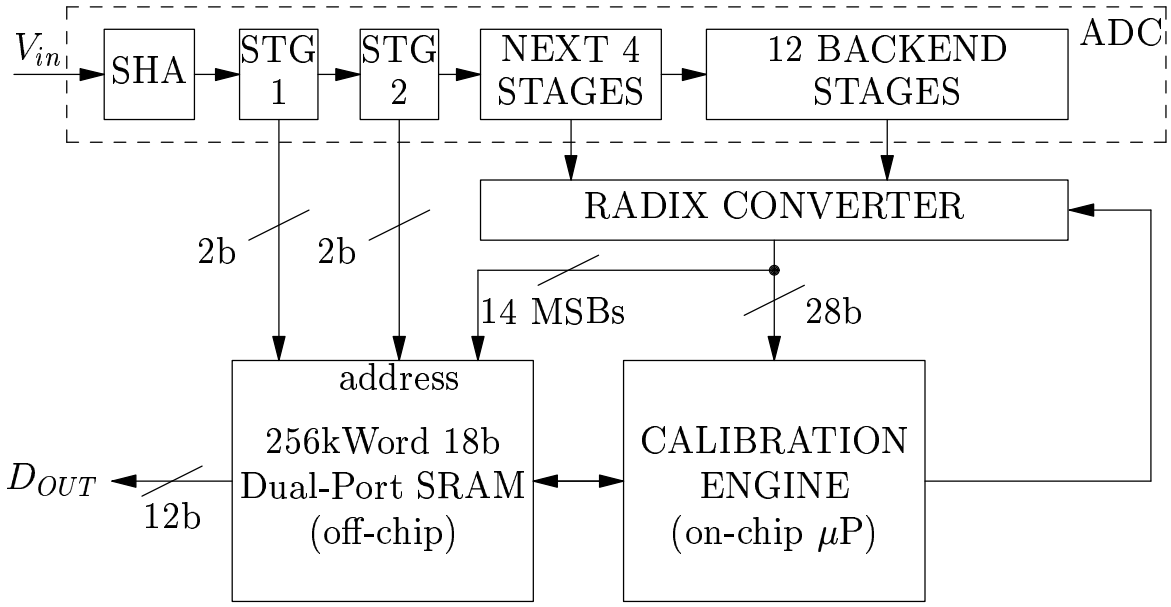


Fig. 10. Block diagram of ADC and calibration system.

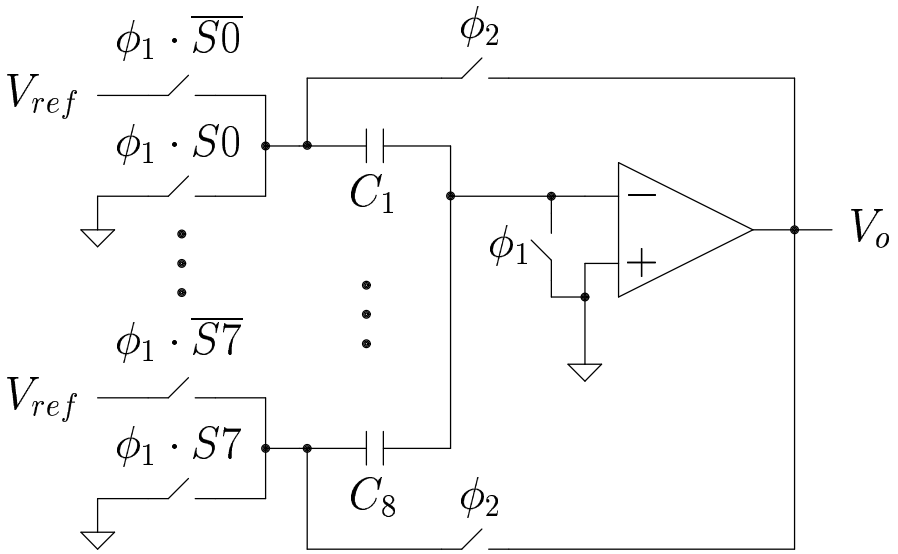


Fig. 11. Calibration DAC.

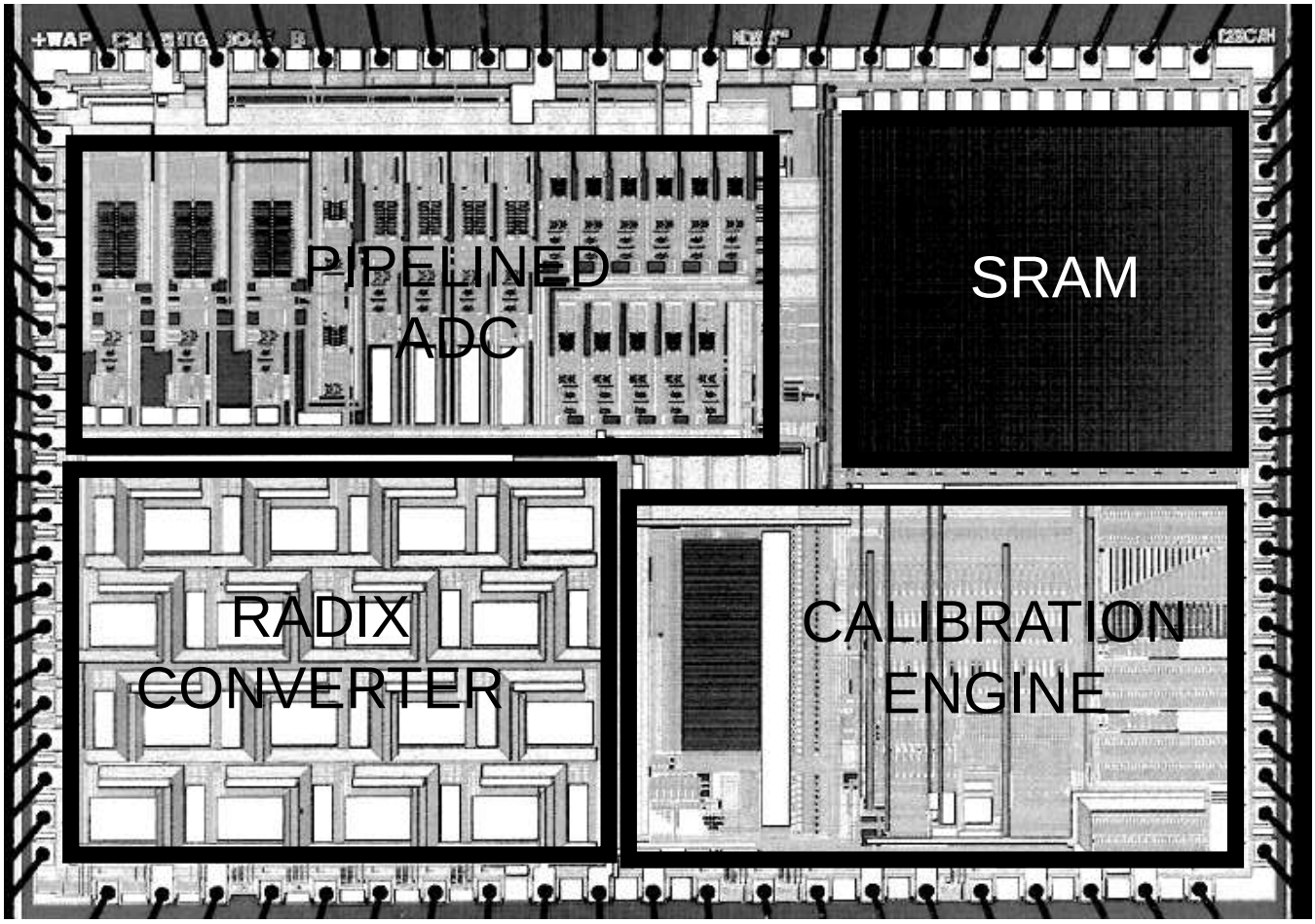


Fig. 12. Die photograph.

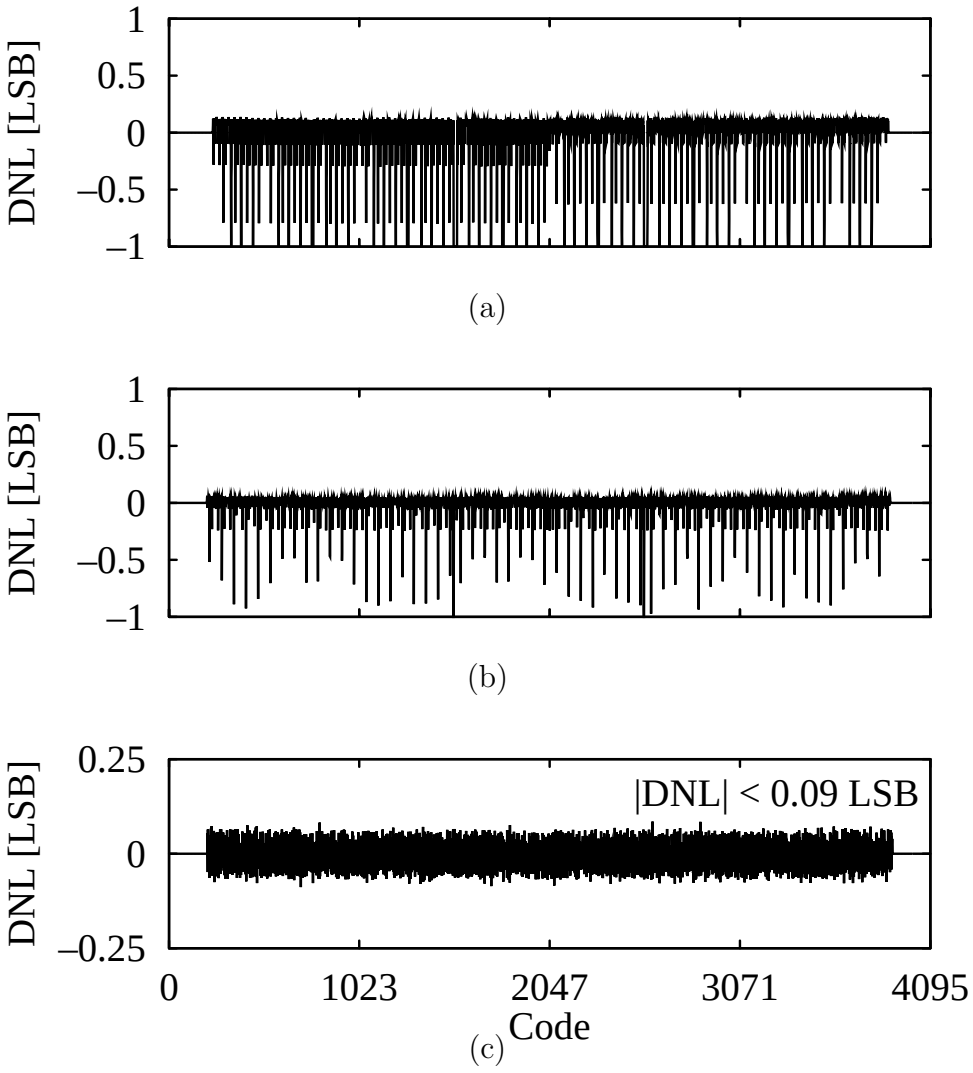


Fig. 13. Differential nonlinearity (DNL) with foreground calibration: (*a*) with no calibration, (*b*) with constant-gain calibration only, and (*c*) with full calibration. Sampling rate is 80 MS/s.

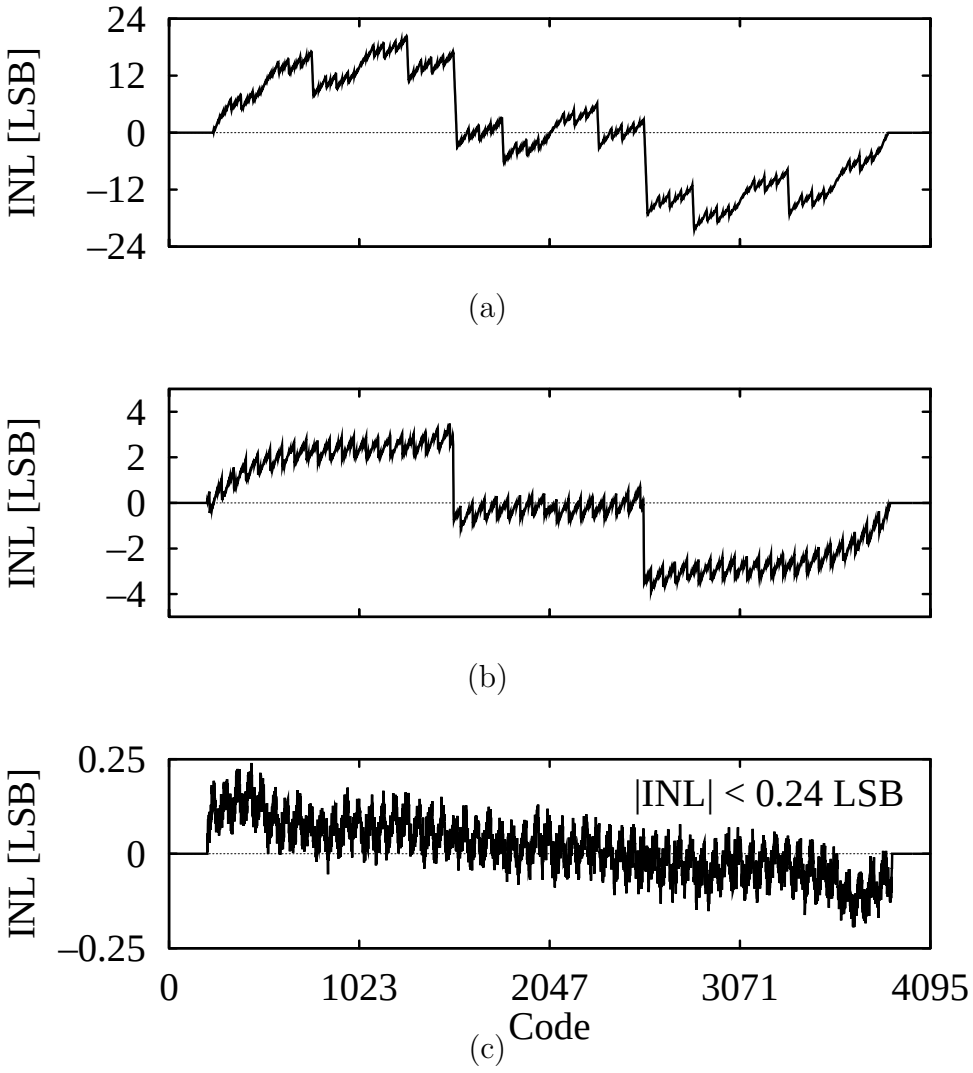


Fig. 14. Integral nonlinearity (INL) with foreground calibration: (*a*) with no calibration, (*b*) with constant-gain calibration only, and (*c*) with full calibration. Sampling rate is 80 MS/s.

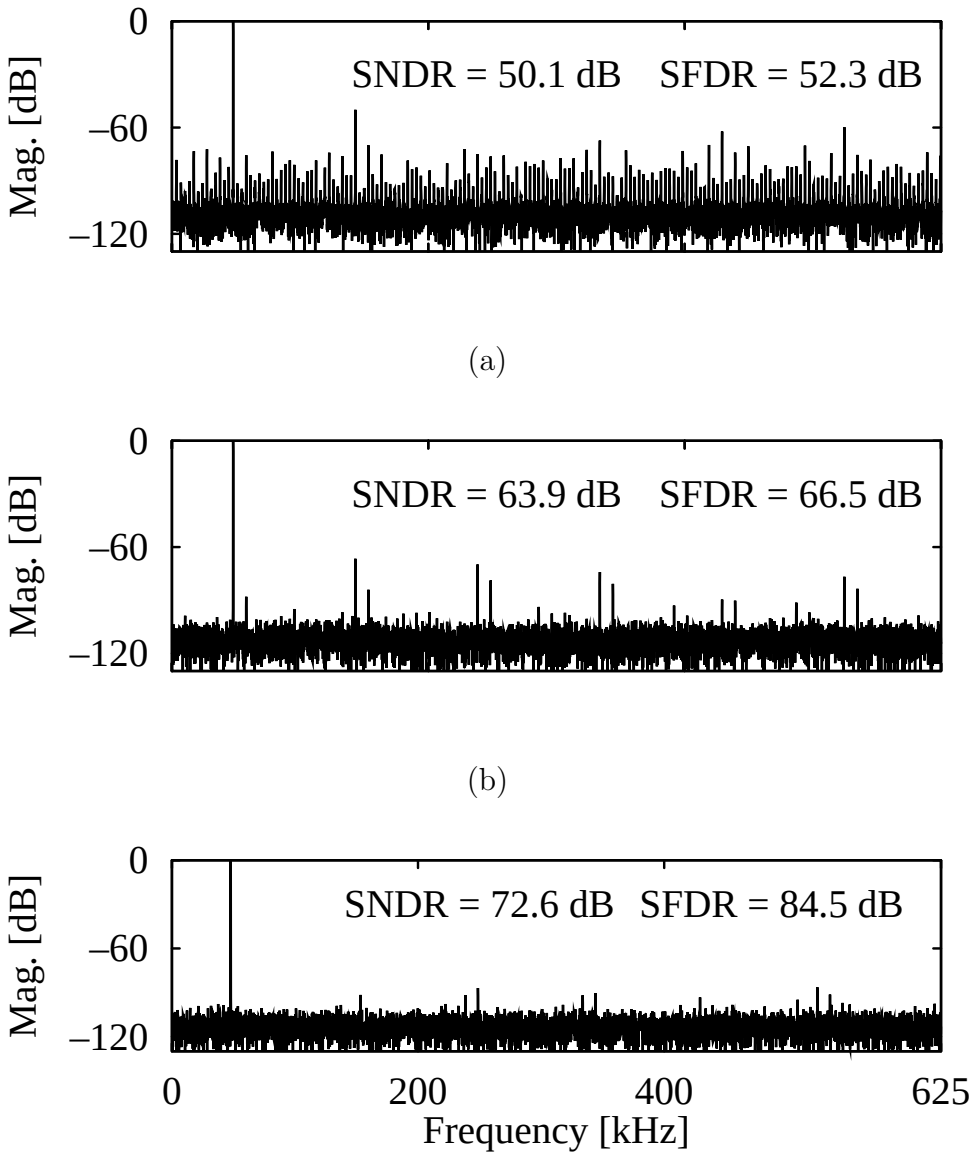


Fig. 15. ADC output spectra with foreground calibration: (*a*) with no calibration, (*b*) with constant-gain calibration only, and (*c*) with full calibration. Sampling rate is 80 MS/s. Output downsampled by factor of 64.

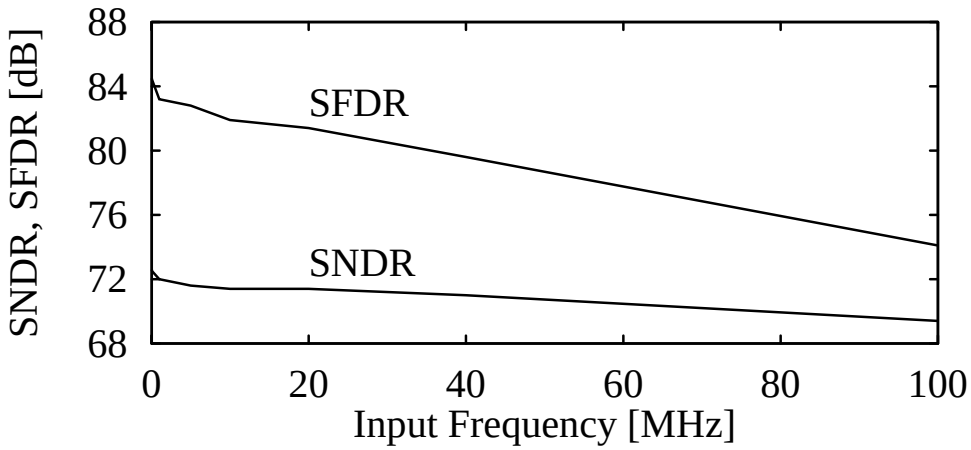
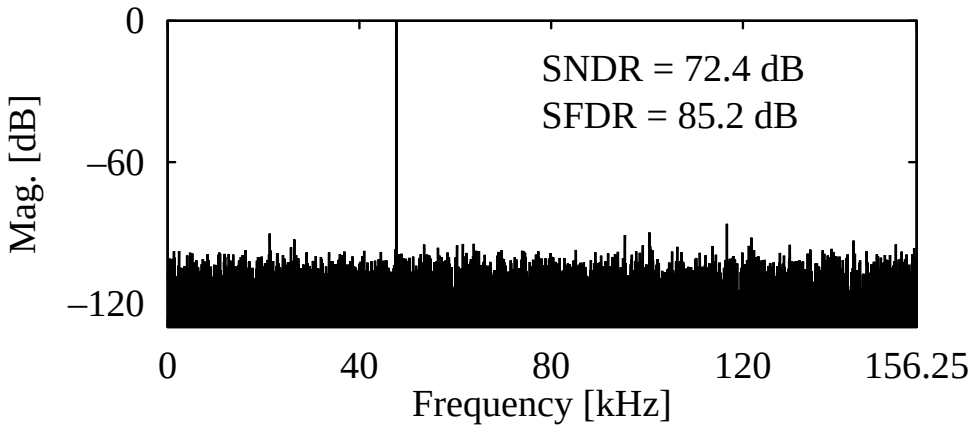


Fig. 16. SNDR and SFDR versus input frequency. Sampling rate is 80 MS/s.

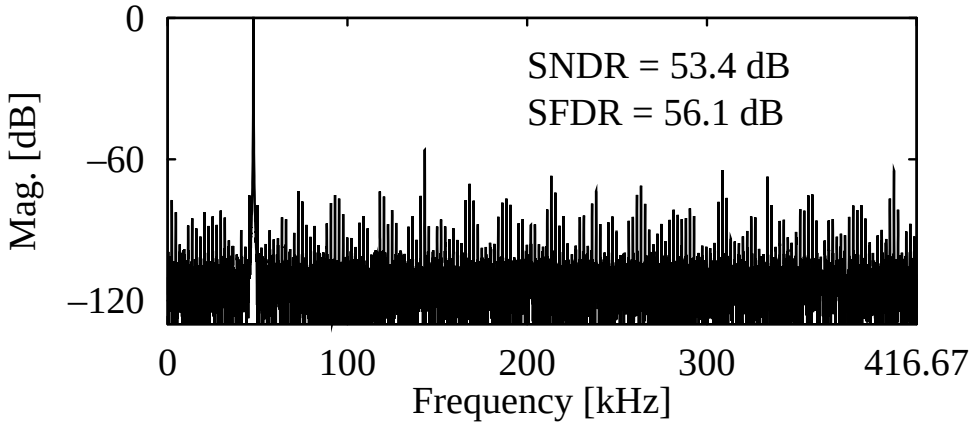
	No Cal.	Constant-Gain Cal.	Full Cal.
Resolution	12b		
Sampling Rate	80 MS/s		
Technology	0.25 μm CMOS		
Power [mW]	340	755 actual, 450 projected	
Area [mm^2]	5.9	22.6 actual, 9.1 projected	
Peak DNL [LSB]	1.0	1.0	0.09
Peak INL [LSB]	20.5	3.9	0.24
SNDR [dB]	50.1	63.9	72.6
SFDR [dB]	52.3	66.5	84.5
ENOB [b]	8.0	10.3	11.8

TABLE I

Performance Summary with Foreground Calibration (2.5 V, 25°C).



(a)



(b)

Fig. 17. ADC output spectra with background calibration: (*a*) Sampling rate is 20 MS/s, (*b*) Sampling rate is 53.33 MS/s. Output downsampled by factor of 64.