IEICE TRANS. FUNDAMENTALS, VOL.E90–A, NO.10 OCTOBER 2007

PAPER

A Digitally Assisted Gain and Offset Error Cancellation Technique for a CMOS Pipelined ADC with a 1.5-bit Bit-Block Architecture

Hiroki SAKURAI[†], *Student Member*, **Shigeto TANAKA**[†], *Nonmember*, *and* **Yasuhiro SUGIMOTO**[††a)], *Member*

SUMMARY This paper proposes a very simple method of eliminating the gain and offset errors caused by mismatches of elements, such as capacitors, for a high-speed CMOS pipelined ADC with a 1.5-bit architecture. The gain and offset errors in a bit-block due to capacitor mismatch are analog-to-digital (A-D) converted without correcting errors, but by exchanging capacitors at every clock. The obtained results are digital codes at the output of the ADC, and they contain positive and negative errors in turn. The two consecutive codes are then added in digital form, thus canceling the errors. This results in the two-fold oversampling operation. As the distortion component arises when the input signal frequency increases, a front-end SHA is used to completely eliminate distortion up to the Nyquist frequency. The behavioral simulation of a 14-bit ADC reveals that this CMOS pipelined ADC with a 1.5-bit bit-block architecture, even without a front-end SHA, has more than 70 dB of spurious-free dynamic range (SFDR) for up to an 8 MHz input signal when each of the upper three bit-blocks has gain and offset errors of +0.8% when the clock frequency is 102.4 MHz. Using an SHA in front further improves the SFDR to 95 dB up to the signal frequency bandwidth of 25.6 MHz.

key words: *capacitor mismatch, CMOS pipelined ADC, 1.5-bit bit-blocks, averaging in digital domain, Oversampling ADC*

1. Introduction

The accuracy of a high-speed CMOS pipelined ADC is generally limited to around a 10-bit equivalent due to mismatches of devices such as capacitors. The capacitor-ratio mismatch causes gain and offset errors in each bit-block of the ADC [1].

The capacitor-ratio mismatch can be compensated for by exchanging capacitors in a bit-block. When the capacitors are exchanged, the ratio error between two capacitors changes by the same amount but with reversed polarity, provided that the error is small. In the ADC in [2], two errors of the same amount but with opposite polarities are produced sequentially in time by exchanging capacitors, and each error plus the input signal is stored in one of the pair of capacitors. Then, two stored signals with errors are added in analog form to cancel the error components. This operation is called 'averaging'. However, an additional op-amp, two capacitors, and extra time for averaging are needed.

The need for an additional op-amp can be eliminated by using a fixed voltage source to which one end of one of the two additional capacitors is connected sequentially in time. As the output signal of a bit-block is stored on two additional capacitors, these can be replaced by the two sample capacitors in the next bit-block [3], [4]. In [4], high performance was realized, such as over 100 dB of SFDR, as was a small power-divided-by-speed ratio in comparison with prior techniques. However, additional transistors to cancel signal-dependent feedthrough errors caused by the stray capacitances of switches between stages are needed. Also needed was a complicated control of clock signals for the switches in order to protect kickback-like errors that result from the increased functional complexity.

On the other hand, in the oversampling method shown in [5], the exchange of the differential input signal and multiplication to the output digital data in digital form are done at every clock to realize the equivalent exchange-capacitor function. This suggests that the digital function helps the analog function reach a minimum and improves the ADC's performance. However, this method is not applicable to ADCs with a 1.5-bit bit-block architecture, which is commonly used in the CMOS pipelined ADC, because the error in the bit-block in [5] is restricted to the form of even function.

The purpose of this paper is, therefore, to offer an effective method for eliminating the gain and offset errors produced by capacitor mismatches in a bit-block of an ADC with a 1.5-bit bit-block architecture by using the digital function without performing a complex averaging procedure in analog form.

In Sect. 2, the major prior techniques of the capacitor averaging method, in both analog and digital forms, are reviewed. Section 3 introduces the proposed digitally assisted capacitor error-averaging method, and Sect. 4 realizes the CMOS bit-block circuit applicable to the proposed ADC with a 1.5-bit bit-block architecture. Section 5 shows the behavioral simulation results of a 14-bit ADC to which the proposed method is applied, and Section 6 concludes this study.

2. Error-Averaging Methods in Prior Techniques

2.1 The Passive Capacitor Error-Averaging Method

We first review representative prior techniques to achieve high precision in CMOS pipelined ADCs [3]–[5]. Those are methods to eliminate bit-block gain and offset errors, which arise mainly from capacitor mismatches. Figure 1 shows

Manuscript received August 28, 2006.
Manuscript revised March 30, 2007.
Final manuscript received July 2, 2007.

[†]The authors are with the Graduate School of E., E., and C. Eng., Chuo University, Tokyo, 112-8551 Japan.
[††]The author is with the Dept. of E., E., and C. Eng., Chuo University, Tokyo, 112-8551 Japan.
a) E-mail: sugimoto@sugi.elect.chuo-u.ac.jp

DOI: 10.1093/ietfec/e90–a.10.2272

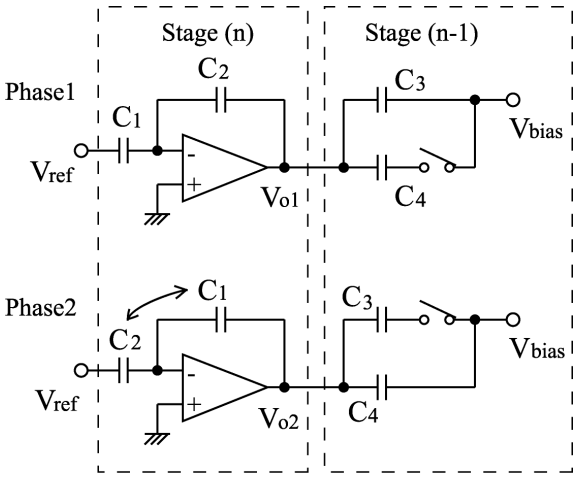


Fig. 1 The passive capacitor error-averaging method.

the passive capacitor error-averaging method reported in [4]. Although the figure shows the circuit of a 1-bit bit-block architecture, it is easily extended to accommodate a 1.5-bit bit-block architecture, which is the standard for CMOS pipeline ADCs. In Fig. 1, two sequential phases, phase 1 and phase 2, in which capacitors C1 and C2 are exchanged, are used in the amplification mode of stage (n). The output residue voltages Vo1 in phase 1 and Vo2 in phase 2 of stage (n) contain gain and offset errors of equal amounts but opposite polarities. Vo1 and Vo2 are stored in the next-stage capacitors, C3 and C4, separately and are automatically merged during the next amplification mode of the stage (n-1) to achieve averaging.

In Fig. 1, we assume capacitor mismatches of C1, C2, C3, and C4 to be $C_1 = C(1+\delta_1)$, $C_2 = C(1+\delta_2)$, $C_3 = C(1+\delta_3)$ and $C_4 = C(1+\delta_4)$ where δ_x $(x = 1 \sim 4)$ is the Gaussian random variable of relative mismatch error with zero mean and variance σ^2, and is a small quantity compared with 1. Then, Vo1 of stage (n) in phase 1 is

$$\begin{aligned} V_{o1} &= \left(1+\frac{C_1}{C_2}\right)V_{in} - \frac{C_1}{C_2}V_{ref} \\ &= \left(2+\frac{\delta_1-\delta_2}{1+\delta_2}\right)V_{in} - \left(1+\frac{\delta_1-\delta_2}{1+\delta_2}\right)V_{ref} \end{aligned} \quad (1)$$

This voltage charges capacitor C3 as shown in Fig. 1, and $Q_{C3} = C(1+\delta_3)(V_{o1} - V_{bias})$.

In phase 2, C1 and C2 are exchanged while preserving the total charges stored in them. Vo2 of stage (n) is

$$\begin{aligned} V_{o2} &= \left(1+\frac{C_2}{C_1}\right)V_{in} - \frac{C_2}{C_1}V_{ref} \\ &= \left(2+\frac{-\delta_1+\delta_2}{1+\delta_1}\right)V_{in} - \left(1+\frac{-\delta_1+\delta_2}{1+\delta_1}\right)V_{ref} \end{aligned} \quad (2)$$

The charge stored in capacitor C4 in phase 2 becomes $Q_{C4} = C(1+\delta_4)(V_{o2} - V_{bias})$. The averaging operation is equivalent to connection between C3 and C4 in parallel, although the averaging is done in the next amplification mode in [4]. Assuming C3 and C4 are connected in parallel, the voltage across C3 and C4 becomes

$$\frac{Q_{C3}+Q_{C4}}{C(2+\delta_3+\delta_4)} = (2+\epsilon)V_{in} - (1+\epsilon)V_{ref} - V_{bias} \quad (3)$$

where the gain error term is

$$\epsilon \approx \frac{1}{2}(\delta_1-\delta_2)(\delta_1-\delta_2+\delta_3-\delta_4) \quad (4)$$

Equation (4) reveals that the gain and offset errors consist of the second-order products of δ_x and is therefore very small. Assume $\delta_1, \cdots \delta_4$ are uncorrelated; we have $E(\epsilon) = \sigma^2$ and $var(\epsilon) = 3\sigma^4$ as indicated in [3].

The averaging method appearing in [3] and [4] worked well and more than 90 dB of THD was obtained up to a 40 MHz input signal frequency in [4]. However, the ADC in [4] needed additional transistors to cancel the unexpected signal-feedthrough error through stray capacitors of switches between stages, additional Vbias voltage, complicated control sequences to protect kickback-like errors due to the increase in functional complexity, and a certain possibility for op-amps to become temporarily floating.

Here, a fundamental question arises: why did those authors find it necessary to store errors in the next-stage capacitors C3 and C4 in Fig. 1, and why did they perform averaging afterward in analog form? Suppose that the erroneous bit-block, i.e., stage (n) in Fig. 1, is followed by the ideal pipeline stages. As errors with equal amounts but in opposite polarity are produced alternately at the output of stage (n) with exchanging capacitors C1 and C2, the output digital data also contain errors with equal amounts but in opposite polarity when the output of stage (n) is digitized in each phase. What if we were to take the average of these two digital data?

2.2 The Oversampling Method

The oversampling method proposed in [5] and [6] produces gain and offset errors of equal amounts but opposite polarities in a similar way as in a logic operation. Input differential signals at the input of a bit-block are exchanged by using analog switches, and the multiplication of minus one to the output digital data from a bit-block is done in alternate clock periods. This is equivalent to exchange capacitors in a bit-block. Unfortunately, authors of [5] and [6] did not take the average of the output digital data of the ADC, instead performing the complicated control of switches in a statistical manner and digital signal processing to improve the frequency characteristics.

Figure 2 explains the concept of this technique. This ADC has input and output characteristics, as shown in Fig. 2(a). The horizontal line shows the input signal level while the vertical line corresponds to the output digital code. As the input and output relationship is not linear, it introduces distortion when the converter's digital output is reconstructed into the analog signal by using a DAC. The quantity of error components in this converter, which is measured as

the distance from the straight dashed line in Fig. 2(a), becomes symmetrical between the input signal level greater than half of the input signal range and that less than half of the input signal range. The error function then becomes an even function.

Suppose that we invert the x- and y-axes together in the graph in Fig. 2(a). We then obtain the graph shown in Fig. 2(b). Figure 2(b) has reciprocal error characteristics compared with Fig. 2(a), and it is clear that the errors will disappear when two graphs in Fig. 2(a) and 2(b) are added together. The reference 6 produced the reciprocal error by inverting the input signal and the output digital code as described above. The inversion of the input signal is equivalent to the inversion of the x-axis, and the inversion of the output digital code is equivalent to the inversion of the y-axis in Fig. 2(a). The block representation of the ADC in [6] is shown in Fig. 2(c). Multiplying $(-1)^n$ means inversion at every clock cycle. The A-D converted data alternate between that in Fig. 2(a) and that in 2(b) in every clock cycle.

It is preferable for a highly accurate and high-speed ADC under a low supply-voltage operating condition to avoid the use of a sensitive and complicated analog operation. The method shown in Fig. 2 only switches differential input signals and performs a simple calculation in the digital domain, yet realizes the equivalent capacitor-exchanging function which is similar to that in Fig. 1. However, the realization of an ADC with 1.5-bit bit-block architecture, which is the standard for CMOS pipelined ADCs, is needed. Although the errors in Fig. 2(a) and (b) were in the form of even functions, the errors in a bit-block of an ADC with 1.5-bit bit-block architecture is in the form of odd functions. We cannot apply the same method as that shown in Fig. 2 to an ADC with 1.5-bit bit-block architecture. A new method that resolves this issue is needed.

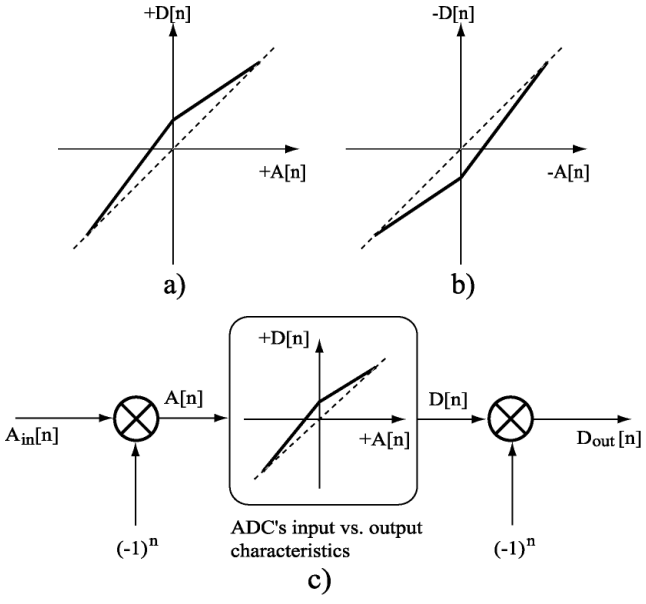


Fig. 2 Oversampling converter method (a) Input vs. output characteristics of a bit-block, (b) Axes-inverted version of (a), (c) Block representation of the oversampling method.

3. The Proposed Capacitor Error-Averaging Method with Simple Control Sequences

We propose to exchange capacitors in a bit-block, which is done only by adding two switches, and to take the average of the ADC's output data in two consecutive clock periods in digital form. This concept is illustrated in Fig. 3. The concept of averaging in digital form is already seen in [7], but two different ADCs are used there. Our approach uses only one ADC. In Fig. 3(a), stage (n) is in the amplification mode, while stage (n-1) is in the sample mode of the residue signal from stage (n). The feedback capacitor in stage (n) is C2. Suppose that this operation in Fig. 3(a) occurs in the $(2N-1)$th clock period. We further assume that the capacitors from stage (n-1) to stage (1) have no mismatch errors. Then, the output digital data from bit-blocks are delayed in some clock periods, are corrected in digital correction logic, and appear at the output of the ADC that contains the A-D converted capacitor mismatch error between C2 and C1. In the $(2N)$th clock period, the conversion changes to that shown in Fig. 3(b), where the roles of C1 and C2 in stage (n) have been exchanged. The ADC's output digital data thus contain a capacitor mismatch error between C1 and C2.

Following the analytical procedure of [6], the operation in Fig. 3(a) and 3(b) is considered to be expressed as

$$D[m] = A[m] + N_w[m] + N_{(1/f)}[m] + (-1)^m g_n(A[m]) \tag{5}$$

where m is the m-th clock period; $A[m]$ and $D[m]$ represent the input signal and the output digital code, respectively;

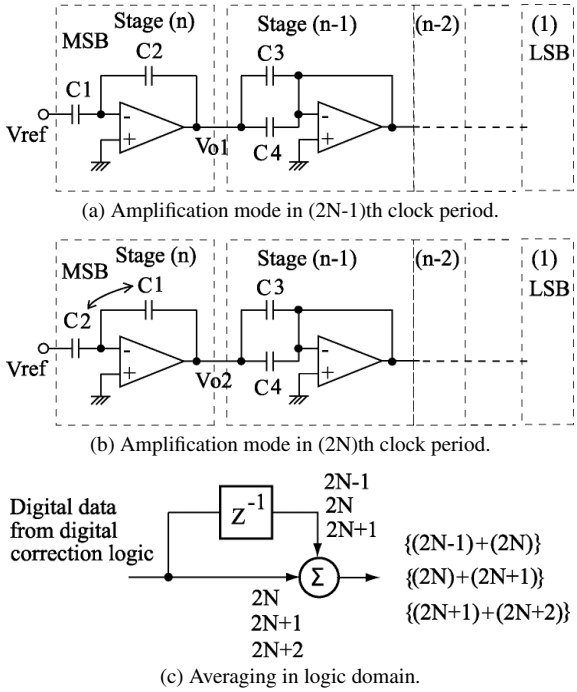


Fig. 3 The proposed digitally assisted capacitor averaging method.

$N_w[m]$ is white noise such as quantization and thermal noise; $N_{(1/f)}[m]$ is the 1/f noise; and $g_n(\cdot)$ models the digital representation of the error generated in stage (n). $(-1)^m$ becomes -1 for the $(2N-1)$th clock period and 1 for the $(2N)$th clock period, or vice versa.

Equation (5) indicates that the error component in stage (n) is inverted at alternate clocks and that all of the distortion components associated with the error are modulated by half the sampling clock frequency. As major distortion components exist in the vicinity of the signal frequency, they are moved to the vicinity of half the clock frequency. By limiting the frequency bandwidth of the ADC to one-fourth the sampling clock frequency, which is the result of the averaging operation shown in Fig. 3(c), the major distortion components are swept out. When $D[2N-1]$ and $D[2N]$ in Eq. (5) are added, a new output digital code, $D_{new}[m]$ of the ADC, is obtained:

$$\begin{aligned} D_{new}[m] \;=\; & A[2N-1] + A[2N] \\ & + N_w[2N-1] + N_w[2N] \\ & + N_{(1/f)}[2N-1] + N_{(1/f)}[2N] \\ & - g_n\left(A[2N-1]\right) + g_n\left(A[2N]\right) \end{aligned} \tag{6}$$

Equation (6) shows that the addition of data at two consecutive clock periods eliminates error components if the input signal frequency is low and A[2N-1] is equal to A[2N]. As the capacitor mismatch error is considered to be the fixed error at each stage even though it differs from stage to stage, errors at all stages except for the final stage are included in the output digital data. Usually, the final stage consists of only a comparator to judge whether the signal level is more than half of its input signal range. The error correction of the first stage is done by using the output digital data of all the rest of the bit-blocks from the second to the last, and the error correction itself comes to have (n-1) bit accuracy where n is the total number of bit-blocks. The operation in Eq. (6) is considered as the two-fold oversampling. Equation (5) is similar to that in [5]; however, the white and 1/f noises in [5] were modulated by half the sampling clock frequency, and such a modulation doesn't occur in the case of Fig. 3.

The transfer function of the digital averaging in Fig. 3(c) is expressed by using Z-transform:

$$H(z) = 1 + Z^{-1} \tag{7}$$

Substituting $Z = e^{j\omega T}$ into Eq. (7), where T is the time interval of the sampling clock periods, the following equation is obtained:

$$|H(j\omega)| = |1 + e^{-j\omega T}| = \sqrt{2\{1 + \cos(\omega T)\}} \tag{8}$$

As Eq. (8) becomes zero when

$$f = \frac{(2k-1)}{2T} \tag{9}$$

where k is an integer, the frequency characteristics of the circuit in Fig. 3(c) are null at odd integer multiples of half the sampling clock frequency. Therefore, all the distortion components associated with the error are modulated by half the sampling clock frequency and thus are greatly reduced by the null.

For comparison with the passive capacitor error-averaging method shown in Fig. 1, we calculate the error component in the output voltage of the bit-block assuming that the capacitor mismatch exists only in the n-th bit-block as in the case of the ADC shown in Fig. 3. The output voltage of the n-th bit-block with C2 as a feedback capacitor as shown in Fig. 3(a) becomes equal to V_{o1} in Eq. (1), and that of the n-th bit-block with C1 as a feedback capacitor as shown in Fig. 3(b) becomes equal to V_{o2} in Eq. (2). The voltage across C3 and C4 connected in parallel is not affected by the capacitor mismatch between C3 and C4 in the case of Fig. 3. In the ADC shown in Fig. 3, the A-D converted digital codes at $(2N-1)$th, and the $(2N)$th clock interval is added to take twice the averaged value. This new value becomes the new ADC output code. By a simple calculation, the error component ϵ becomes,

$$\epsilon \approx \frac{1}{2}(\sigma_1 - \sigma_2)^2 \tag{10}$$

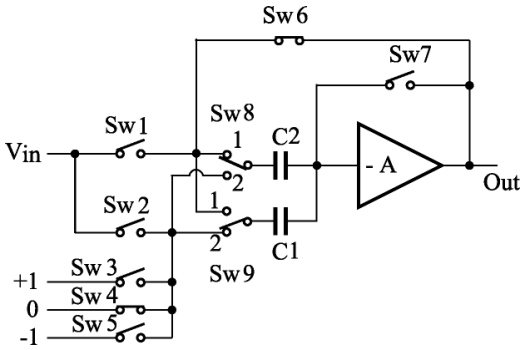
Assume δ_1 and δ_2 are uncorrelated; we have $E(\epsilon) = \sigma^2$ and $var(\epsilon) = 2\sigma^4$. Variance in Eq. (10) is smaller than that in Eq. (4).

4. The Proposed Bit-Block Circuit of the CMOS Pipelined ADC with the 1.5-bit Bit-Block Architecture

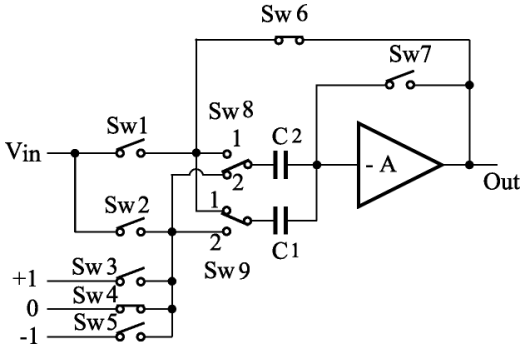
In order to adopt the proposed architecture shown in Fig. 3 to the 1.5-bit bit-block architecture, which is very popular in CMOS pipelined ADC, the circuit as shown in Fig. 4 is used as a bit-block circuit. The circuit in Fig. 4 consists of an S/H circuit, a sub-DAC, and a residue amplifier. The only difference between the circuit shown in Fig. 4 and the conventional one [1] is the addition of switches Sw8 and Sw9. Figure 4(a) shows the circuit in amplification mode in the (2N-1)th clock period. The input signal is stored in advance in sample mode in both capacitors C1 and C2. Because the charge stored in the capacitors is equal in both the sample and amplification modes, Eq. (1) is derived for the circuit shown in Fig. 4(a), where Sw8 and Sw9 are in the first and second switch positions, respectively. Figure 4(b) has a configuration such that the feedback capacitor of the amplifier is C1 instead of C2, in contrast to the configuration shown in Fig. 4(a). In this case, Eq. (2) is derived and Sw8 and Sw9 are in the second and first switch positions, respectively.

5. Behavioral Simulation Results

To demonstrate the effectiveness of the proposed ADC shown in Fig.3 and 4, we performed behavioral simulations using a C program. The program realizes 13 1.5-bit bit-blocks plus one 1-bit bit-block, delay units, digital correction logic, a one-clock delay unit (Z^{-1}), and an adder (+) to form a 14-bit ADC like that shown in Fig. 5. Gain and offset



(a) Amplification mode in (2N-1)th clock period.



(b) Amplification mode in (2N)th clock period.

Fig. 4 The possible CMOS bit-block circuit for the 1.5-bit bit-block architecture.

errors caused by capacitor mismatches are set to +0.8% for each of the first three bit-blocks. The rest of the bit-blocks are assumed to be ideal. The input signal frequency for all the simulations, except for that in Fig.9, is 1.00625 MHz, and the clock frequency is 102.4 MHz for all. Clock timing is assumed to be aligned, and the timing jitter has not been taken into account.

Figure 6 shows the Fast-Fourier-Transform (FFT) result of the output data. The input signal voltage is ±2 V. In this case, the program disables the capacitor-exchanging function and the function to take the average of the output data at every clock. We describe this as the conversion of a conventional ADC. There are 16,384 collected data points. As seen in Fig. 6, gain and offset errors given to the first three bit-blocks produce harmonic components in the vicinity of the fundamental signal frequency, that is, 1 MHz. Odd multiples of harmonics such as 3 MHz, 5 MHz, and so on, become eminent because the gain and offset errors in the 1.5-bit bit-block become odd functions. The signal frequency bandwidth is 51.2 MHz, and the SFDR is 57.5 dB.

When the capacitors in the first three bit-blocks at every clock have been exchanged in the ways shown in Figs. 3(a) and (b), the frequency spectrum changes, as shown in Fig. 7. The averaging function at the output has not been enabled yet. As Eq. (5) has predicted, the error components are modulated by half of the clock frequency. It can be clearly seen that the harmonic components become aligned in reverse order, in contrast to Fig. 6, in the frequency domain starting from 51.2 MHz. The SFDR remains 56.5 dB because only

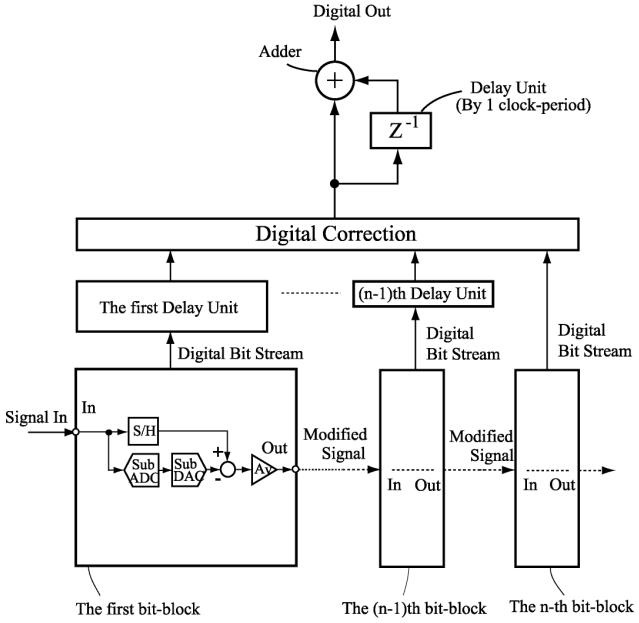


Fig. 5 The simulated 14-bit ADC configuration.

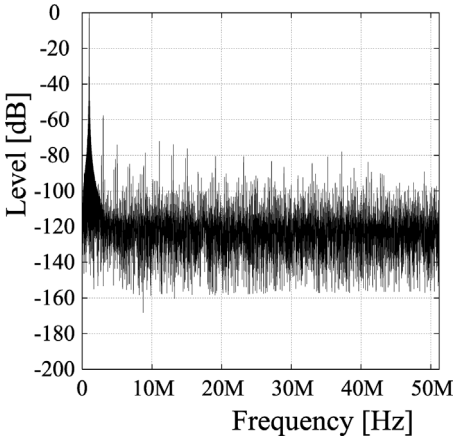


Fig. 6 The frequency spectrum of the simulated conventional 14-bit ADC.

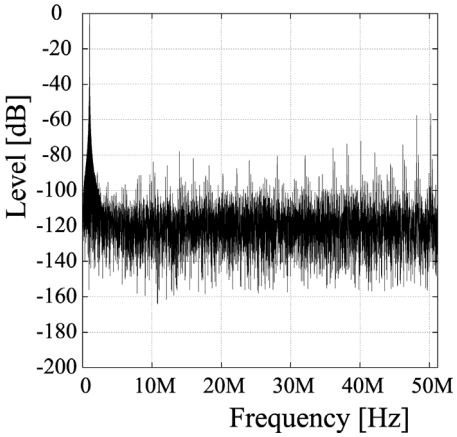


Fig. 7 The frequency spectrum of the simulated 14-bit ADC with exchanging capacitors.

the re-alignment of distortion components within the bandwidth occurred. No averaging was taken, and the signal frequency bandwidth does not change, but remains 51.2 MHz.

Further improvement in the SFDR is achieved by taking the sum of the ADC outputs at every clock using a unit delay function and an adder, as shown in Fig. 3(c) and Fig. 5. The average is obtained by taking the upper 14-bit data from the resultant 15-bit data at the adder output. This is because the addition of two 14-bit data produces 15-bit data and the divide-by-two operation is equal to the right shift of bits in a binary number. The offset error caused by exchanging capacitors has a frequency component at half the clock frequency, and is eliminated by the null of Eq. (9). However, the introduction of the null reduces the signal frequency bandwidth to one-fourth of the clock frequency.

Figure 8 shows the result of introducing the averaging function to the model shown in Fig. 7. The offset error at 51.2 MHz is completely eliminated and the distortion components in the vicinity of 51.2 MHz are reduced by the introduction of the null. The signal frequency bandwidth becomes half and is thus 25.6 MHz due to averaging. Both the signal and noise in low frequencies increase by 6 dB by the addition and are reduced to 0 dB by the divide-by-two operation. The result is that the SNR increased 3 dB, and the SFDR is measured to 78.7 dB for the bandwidth of 25.6 MHz. The SNR improvement is automatic due to the bandwidth reduction; however, it is not automatic for the SFDR. It is improved by the combination of both the introduction of the null and the reduction of the bandwidth, that is, the averaging function. Therefore, we focused on the SFDR here. The frequency range of 25.6 MHz is wide enough to display, but it is extended to 51.2 MHz in Fig. 8 in order to see the effect of the null.

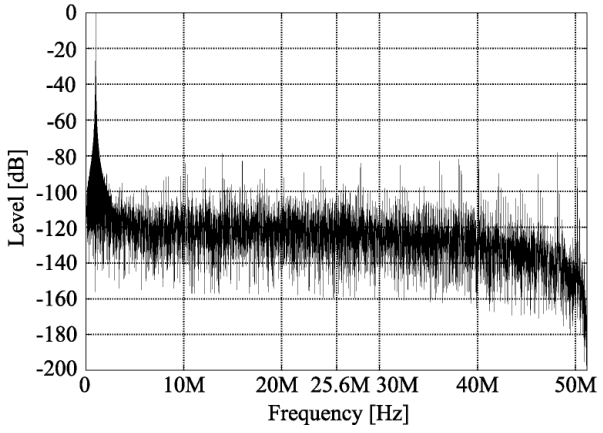


Fig. 8 The frequency spectrum of the simulated 14-bit ADC including capacitor exchange and averaging at the output.

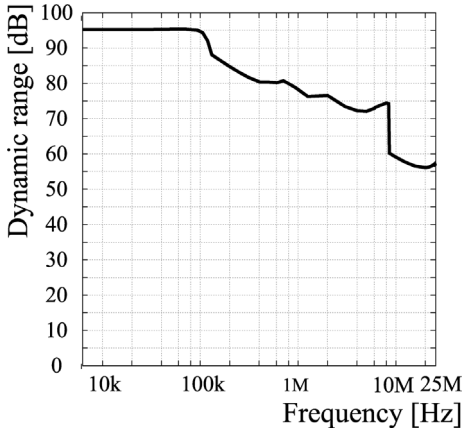


Fig. 9 The input signal frequency vs. SFDR characteristics of the ADC in Fig. 5.

When the input signal frequency increases, however, the cancellation of gain and offset errors becomes less effective due to the input signal change during two consecutive clock periods. Figure 9 shows the input signal frequency vs. SFDR characteristics. The SFDR at low frequencies is excellent. However, the SFDR decreases to approximately 78.7 dB at 1.00625 MHz and to 55 dB at 25.6 MHz, which is one-fourth the clock frequency. Due to the complex error function and the fixed phase alignment between the input signal and sampling clock in the simulation, the SFDR degradation does not become linear. However, it is evident that the SFDR declines gradually as the input signal frequency increases.

If the input signal to the pipelined ADC is held constant during the $(2N-1)$th and $(2N)$th clock periods, then the complete cancellation of the harmonics or distortion components is achieved and no degradation of SFDR for high frequency input signals arises. The S/H circuit can be used for completely cancell the harmonics. In this case, the S/H circuit should sample the input signal at half the ADC's clock frequency; in other words, two ADC clock periods are necessary to complete the operation. Moreover, the S/H samples the input signal for the period of half of one ADC's clock and holds the signal for a period of one and a half of the ADC's clock, so that the ADC can take out two of the same signal levels from the S/H output in the timing shown in Fig. 10. In this case, two consecutive output data that were A-D converted while the S/H output was held constant should be added correctly. If the output data of the ADC is obtained at the time when the state expressed by "ADC in Fig. 11" shown in Fig. 10 changes from S to H, then we should not add the output data of the second and third state changes from S to H in "ADC in Fig. 11" in Fig. 10. As the correct data with the sum of the first and second state changes and the incorrect data with the sum of the second and third state changes appear alternately at the adder output in synchronizing with a clock, a master-slave D-latch that operates in half the clock frequency is placed there to

Clock	1	0	1	0	1	0	1	0
S/H output	S	H	H	H	S	H	H	H
ADC in Fig.11	H	Ⓢ	H	Ⓢ	H	Ⓢ	H	Ⓢ

Fig. 10 Timing diagram for connection between a front-end S/H and an ADC.

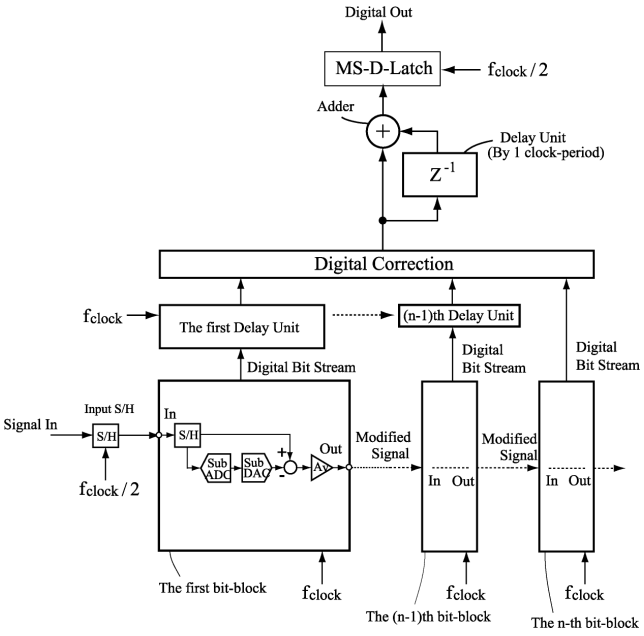


Fig. 11 The block diagram of the simulated 14-bit ADC with a front-end S/H.

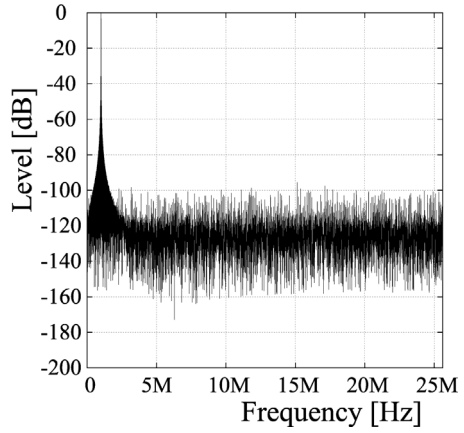


Fig. 12 The frequency spectrum of the simulated 14-bit ADC with a front-end S/H.

obtain the correct sum, as shown in Fig. 11.

Figure 12 shows the output frequency spectrum of the ADC with an S/H function shown in Fig. 11. The frequency range in the figure is limited to 25.6 MHz because the input S/H now samples the input signal at half the clock frequency of the ADC. The sampling frequency is 51.2 MHz and the clock frequency is 102.4 MHz in this case. In the hold mode of the S/H circuit, two A-D conversions are performed by exchanging capacitors at the $(2N-1)$th and $(2N)$th clock periods. Now the input is constant, and the gain and offset errors caused by the capacitor mismatches contained in the two conversion data cancel each other out. With the perfect S/H circuit and with aligned clocks, the SFDR reaches 95.6 dB, as seen in Fig. 12, even when gain and offset errors of +0.8% exist in each of the first three bit-blocks of the ADC. The SFDR value doesn't change even if the gain and offset errors change to −0.8% in each of these three bit-blocks, or if the input signal frequency changes.

Table 1 ADC performance comparison.

ADCs	BW	fs	SFDR (1.00625 MHz Input)
Conventional	51.2 MHz	102.4 MHz	57.5 dB
Exchange Only	51.2 MHz	102.4 MHz	56.5 dB
ADC in Fig. 5	25.6 MHz	102.4 MHz	78.7 dB
ADC in Fig. 11	25.6 MHz	51.2 MHz	95.6 dB

Table 1 summarizes the input signal bandwidth (BW), sampling frequency (fs), and SFDR of several ADCs. By alternating capacitors in the first three bit-blocks at alternate clocks, and by taking the average of the output digital code, the oversampling pipelined ADC with 1.5-bit bit-blocks is realized. The SFDR of the ADC in Fig. 5 is improved by the sacrifice of frequency bandwidth reduction to one-fourth of the clock frequency. By using an S/H in front, the SFDR is remarkably improved with the same frequency bandwidth as that of the ADC in Fig. 5.

6. Conclusion

A new pipelined ADC with 1.5-bit bit-blocks is introduced using a digitally assisted capacitor averaging method. As the averaging is performed in the digital domain, the increase of the circuit complexity comprises a mere two switches in a bit-block. A behavioral simulation verified that the SFDR reaches 95.6 dB with the S/H in front at a sacrifice of input-signal frequency bandwidth of one half.

References

[1] B. Razavi, Principles of Data Conversion System Design, pp.218–222, IEEE Press, 1995.
[2] B.S. Song, M.F. Tompsett, and K.R. Lakshmikumar, "A 12-bit 1-Msamples/s capacitor error-averaging pipelined A/D converter," IEEE J. Solid-State Circuits, vol.23, no.12, pp.1324–1333, Dec. 1988.
[3] Y. Chiu, "Inherently linear capacitor error-averaging techniques for pipelined A/D conversion," IEEE Trans. Circuits Syst. II, vol.47, no.3, pp.229–232, March 2000.
[4] Y. Chiu, P.R. Gray, and B. Nikolic, "A 14-b 12-MS/s CMOS pipeline ADC with over 100-dB SFDR," IEEE J. Solid-State Circuits, vol.39, no.12, pp.2139–2151, Dec. 2004.
[5] A. Shabra, H.S. Lee, and L. Hernandez, "Oversampled pipeline A/D converters with mismatch shaping," Electron. Lett., vol.34, no.6, pp.508–509, March 1998.
[6] A. Shabra and H.S. Lee, "Oversampled pipeline A/D converters with mismatch shaping," IEEE J. Solid-State Circuits, vol.37, no.5, pp.566–578, May 2002.
[7] K. Nakamura, M. Hotta, L.R. Carley, and D.J. Allstot, "An 85 mW, 10 b, 40 Msample/s CMOS parallel-pipelined ADC," IEEE J. Solid-State Circuits, vol.30, no.3, pp.173–183, March 1995.
[8] P.C. Yu and H.S. Lee, "A pipelined A/D conversion technique with near-inherent monotonicity," IEEE Trans. Circuits Syst. II, vol.42, no.7, pp.500–502, July 2000.

Hiroki Sakurai received the B.E. and M.E. degrees in 2002 and 2004 from Chuo University, Tokyo, Japan. He is currently the Ph.D. student at the Graduate School of Electrical, Electronic, and Communication Engineering of Chuo University. He is interesting in the design and new circuit development especially in mixed-signal LSIs such as high-speed and high-resolution D-to-A converters and low-voltage step-down DC-DC converters.

Shigeto Tanaka received the B.E. degree in 2005 from Chuo University, Tokyo, Japan. He is currently the Master student at the Graduate School of Electrical, Electronic, and Communication Engineering of Chuo University. He is interesting in the design and new circuit development especially in mixed-signal LSIs such as high-speed and high-resolution A-to-D converters.

Yasuhiro Sugimoto received the B.E. degree from Tokyo Institute of Technology, Tokyo, Japan, M.E. degree from University of Michigan, Ann Arbor, Michigan, and Doctor of Engineering degree from Tokyo Institute of Technology, Tokyo, Japan, in 1973, 1980, and 1991, respectively. He joined Toshiba Semiconductor Group in 1973, and engaged in the development of analog VLSIs. Since 1992, he has been with the Faculty of Science and Engineering, Chuo University where he is now a professor in the Department of Electrical, Electronic, and Communication Engineering. His main interest is the design and development of new circuits in mixed-signal and RF LSIs. He is the recipient of the 1989 Best Papers Award of European Solid-State Circuits Conference and the 1998 IEICE Best Papers Award. He is the author of three books. Dr. Sugimoto is the senior member of the Institute of Electrical and Electronics Engineers, INC., the members of the Japan Consulting Engineers Association.