Digital Cancellation of D/A Converter Noise in Pipelined A/D Converters

Ian Galton

***Abstract*—Pipelined analog-to-digital converters (ADCs) tend to be sensitive to component mismatches in their internal digital-to-analog converters (DACs). The component mismatches give rise to error, referred to as DAC noise, which is not attenuated or cancelled along the pipeline as are other types of noise. This paper describes an all-digital technique that significantly mitigates this problem. The technique continuously measures and cancels the portion of the ADC error arising from DAC noise during normal operation of the ADC, so no special calibration signal or auto-calibration phase is required. The details of the technique are described in the context of a nominal 14-bit pipelined ADC example at both the signal processing and register transfer levels. Through this example, the paper demonstrates that in the presence of realistic component matching limitations the technique can improve the overall ADC accuracy by several bits with only moderate digital hardware complexity.**

I. Introduction

UNLIKE other types of noise in a conventional pipelined ADC, noise introduced by the first-stage DAC is not attenuated or cancelled along the pipeline, so it tends to be the dominant contributor of overall ADC error [1], [2]. In typical switched-capacitor implementations, most of the DAC noise arises from static capacitor mismatches. With present VLSI circuit technology it is difficult to match capacitors to better than 0.1%. This translates into an A/D conversion limit of about 11 bits in pipelined ADC architectures without some form of error cancellation.

This paper describes a technique for digital cancellation of DAC noise arising from static analog errors such as capacitor mismatches. The technique is referred to as *DAC noise cancellation* (DNC). It differs from most other cancellation schemes in that it measures and cancels the DAC noise continuously during normal operation of the ADC; no special calibration signal or autocalibration phase is required prior to A/D conversion. Both the measurement and cancellation of DAC noise are performed entirely using digital logic, so no additional analog circuits are required over those of a conventional pipelined ADC. As demonstrated below, the digital complexity of the DNC processing is well within practical limits for typical CMOS and BiCMOS circuit technologies.

Throughout the paper, the DNC architecture, simulation results, and theory are presented in the context of a specific pipelined ADC architecture example. A conventional version of the example pipelined ADC architecture (i.e., without DNC) is described in Section II, and the modifications required to apply the DNC technique to the architecture are described in detail in Section III. An overview of the signal processing theory behind the DNC technique is presented in Section IV. Simulation results indicating the convergence rate of the DNC measurement process are presented in Section V.

II. A Conventional Pipelined ADC Example

The conventional version of the example pipelined ADC architecture is shown in Fig. 1. It consists of four pipeline stages. The first three stages each incorporate a 17-level flash ADC and a 17-level switched-capacitor DAC, and the fourth stage consists of a 33-level flash ADC. The three 17-level ADCs are nominally identical; each consists of 16 voltage comparators that compare the voltage at the input of the ADC to a set of 16 reference voltages nominally ranging from -0.75 V to 0.75 V in steps of 100 mV. The corresponding ADC *input no-overload range*, i.e., the range of input values for which the quantization error never exceeds half of the step-size, is -0.85 V to 0.85 V. The 33-level ADC is similar except that it consists of 32 comparators and its 32 reference voltages nominally range from -0.775 V to 0.775 V in steps of 50 mV. The corresponding input no-overload range is -0.825 V to 0.825 V.

The output of each 17-level ADC is the set of its 16 1-bit comparator outputs, and that of the 33-level ADC is the set of its 32 1-bit comparator outputs. This type of digital encoding is referred to as *thermometer encoding*, and the digital value of each ADC output is interpreted as the number of its comparator outputs that are high.

The three 17-level DACs are each implemented using 16 switched-capacitor 1-bit DACs that share a common summing node. In each case, the 16 1-bit DACs are driven directly by the 16 comparator outputs from the corresponding 17-level ADC. The nominal output voltage levels associated with each 1-bit DAC are ±50 mV. Therefore, the output of each 17-level DAC is nominally within 50 mV of the input to the corresponding ADC provided the no-overload range of the ADC is not exceeded.

In each stage except for the last, the difference between the ADC input and DAC output is amplified by an *interstage gain* of 8. In the absence of ADC and DAC errors, the interstage gains are such that just under half of the no-overload ranges of the ADCs in the second through last stages are ever used. That is, if a signal were applied to the input of the pipelined ADC that varied over the full-scale input range of -0.85 V to 0.85 V, then the resulting signals at the inputs to the ADCs in the second

Manuscript received April 1999; revised October 1999. This work was supported in part by the National Science Foundation under Grant MIP-9711331 and by the Office of Naval Research under Grant N00014–98–1–0830. This paper was recommended by Associate Editor H. Tanimoto.

The author is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA.

Publisher Item Identifier S 1057-7130(00)02388-0.

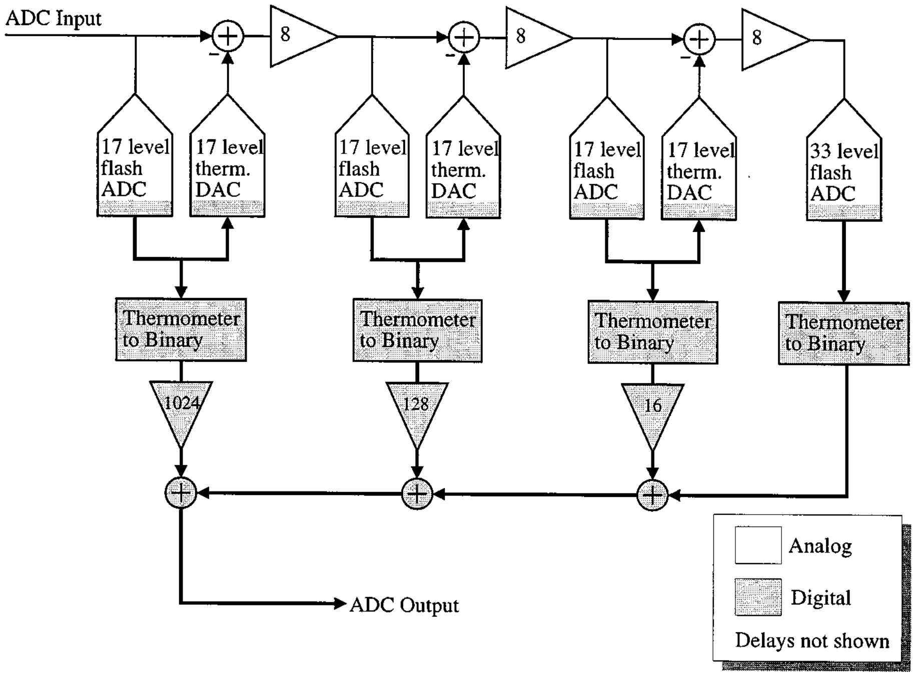


Fig. 1. Functional diagram of the 14-bit four-stage pipelined ADC architecture example.

through last stages would be restricted to the range: -0.4 V to 0.4 V. Therefore, the choice of 8 for the interstage gains ensures slightly more than a 100% margin for signals to *overrange* (i.e., exceed their ideal ranges) as a result of nonideal analog circuit behavior without overloading the ADCs.

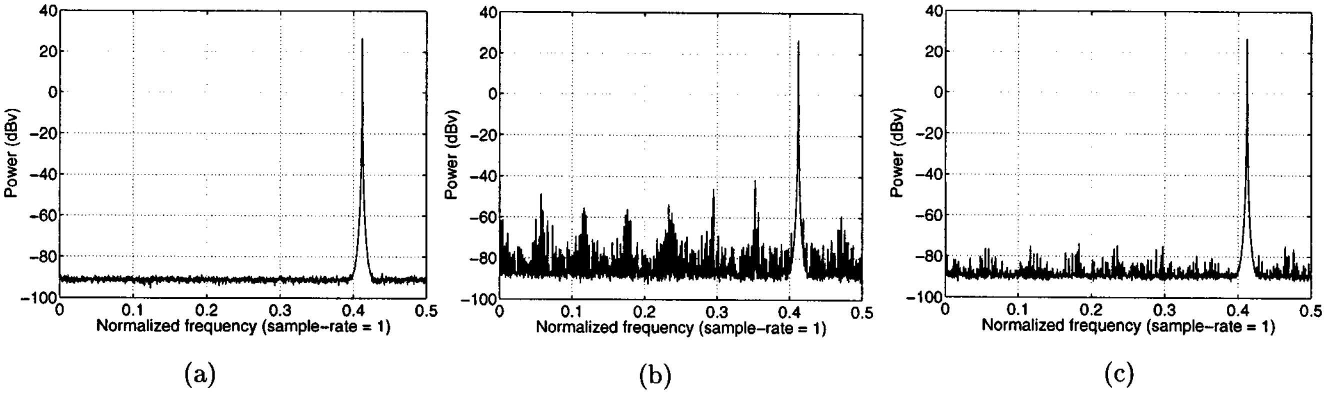
Prior to arithmetic processing, the output of each ADC is converted from thermometer encoded data to two's-complement binary encoded data. After the conversion, the digital signals from the ADCs are scaled and added as shown in Fig. 1. It can be verified that in the absence of nonideal circuit behavior the pipelined ADC of Fig. 1 performs uniform quantization with slightly more than 14 bits of precision. For example, if the input to the pipelined ADC increases from -0.8 V to 0.8 V (this range is slightly less than the full input no-overload range of the pipelined ADC which is equal to that of the first 17-level ADC: -0.85 V to 0.85 V), the output increases from -8192 to 8192. Consequently, the least significant bit (LSB) of the pipelined ADC output corresponds to an input value of $1.6 \cdot 2^{-14}$ V.

In practice, every analog component in the pipelined ADC will exhibit nonideal circuit behavior that will tend to degrade the overall conversion performance to less than 14 bits. However, the sensitivity of the pipelined ADC to nonideal circuit behavior of a given component is a function of where the component resides in the pipeline. It follows from Fig. 1 that in the first stage the gain between either input node of the analog differencer and the overall pipelined ADC output is 10240.[1] However, the corresponding gains in the subsequent two stages are 1280, and 160, respectively, and the gain from the input to output of the last stage is 20. Hence, the pipelined ADC performance tends to be most sensitive to nonideal circuit behavior in the first stage and the sensitivity decreases by a factor of 8 (18 dB) in each subsequent stage. Furthermore, the pipelined ADC is virtually insensitive to error introduced by each of the 17-level ADCs provided the error is not so large that the overrange margin of the next stage is exceeded. The reason is that in each case the signal from the ADC is converted back to analog, sign inverted, re-digitized by the subsequent stages of the pipeline, and added to the signal directly from the ADC, thereby cancelling the error.

To illustrate these concepts, three sets of simulation results are shown in Fig. 2. Fig. 2(a) shows the power spectral density (PSD) of the output sequence from the pipelined ADC of Fig. 1 implemented with ideal components. The ADC output was normalized to have unity gain prior to estimating the PSD which is shown in units of dBV. The input signal was a nearly full-scale sinusoid plus a white dither sequence uniformly distributed over the nominal LSB of the pipelined ADC (i.e., $1.6 \cdot 2^{-14}$ V). The purpose of the dither was to allow precise calculation of the overall ADC quantization noise power. As would be expected from an ideal uniform quantizer, only the desired signal plus white noise is evident in the PSD (the dither has been subtracted from the PSD in the figure so the white noise corresponds entirely to quantization noise). The noise floor is -01 dBV and this corresponds to 14.1 bits of precision as expected.

The PSD shown in Fig. 2(b) corresponds to the same input signal and pipelined ADC architecture except that various realistic levels of nonideal circuit behavior were included. Mismatches in the 17-level DACs were modeled by selecting the two output voltages corresponding to each 1-bit DAC with random errors of 0.3% standard deviation. The reference voltages for each ADC were generated by a simulated resistor ladder wherein each resistor was chosen with a random error of 0.3% standard deviation, and the offset voltage of each comparator in the ADC was chosen randomly with a standard deviation of 10 mV. The interstage gains were chosen with a random error of 0.3% standard deviation. In all cases, the errors

[1] With the LSB of each flash ADC interpreted as unity, the gain of the ADC is the inverse of its quantization step-size. Thus, the gain of each 17-level ADC is 10 and that of the 33-level ADC is 20.



(a) (b) (c)

Fig. 2. Power spectral density plots of the output of the pipelined ADC simulated with (a) ideal components, (b) nonideal components, and (c) nonideal components but with ideal post-ADC removal of the error components arising from the DAC noise introduced in the first two stages of the pipeline.

were chosen with a Gaussian distribution. As is evident from the numerous spurious tones and increased noise floor, these relatively small circuit nonidealities give rise to a dramatic reduction in performance. Integrating the PSD in the absence of the signal indicates a precision equivalent to 10.4 bits, a reduction of 3.7 bits below the ideal performance level.

The PSD shown in Fig. 2(c) was obtained by performing the same simulation that resulted in Fig. 2(b), except subtracting from the output of the pipelined ADC the error arising from DAC noise introduced by the 17-level DACs in the first two stages. A reduction in the power of the spurious tones and noise floor is evident. In this case, integrating the PSD in the absence of the signal indicates a precision equivalent to 13.5 bits. This result indicates that, as expected, most of the pipelined ADC error visible in the PSD of Fig. 2(b) is caused by the DAC noise introduced by the 17-level DACs in the first two stages. It can be verified that most of the remaining 0.6 bits of performance degradation relative to the ideal component case arises from the interstage gain error between the first and second stages of the pipeline.

III. Application of DNC to the Pipelined ADC Example

As demonstrated by the simulation results presented in the previous section, a significant performance improvement can be obtained by digitally removing from the pipelined ADC output the error components resulting from the noise introduced by the 17-level DACs in the first two pipeline stages. In the simulation software used to generate Fig. 2(c), this was done directly using a priori knowledge of the specific 1-bit DAC errors, which would not be practical in an actual pipelined ADC. However, the DNC technique achieves almost the same effect except in a practical fashion without requiring a priori knowledge of the 1-bit DAC errors. The modifications to the pipelined ADC of Fig. 1 that are necessary to apply the DNC technique to the first two pipeline stages are presented in this section.

The modified pipelined ADC with DNC applied to the first two pipeline stages is shown in Fig. 3. It differs from the conventional version in two respects. First, the 17-level DACs in the first two stages have been replaced by modified DACs referred to as *dynamic element matching* (DEM) DACs. The DEM DACs differ from the thermometer encoded DACs described in the previous section in that each contains a *digital encoder* that randomly permutes the connections between the 16 thermometer encoded input bits and the 16 switched-capacitor 1-bit DACs. Second, two blocks, labeled *DNC logic* in Fig. 3, have been included that each generate estimates of the error component in the ADC output arising from the noise introduced by the associated DEM DAC. These estimated error sequences are subtracted from what would otherwise be the output sequence in a conventional pipelined ADC.

As explained in detail in the next section, aside from gain and offset errors, the DAC noise introduced by each DEM DAC consists of a sum of 15 terms. Each term is the product of a constant and a unique modulation sequence. The constant depends only upon the 1-bit DAC errors. The modulation sequence is restricted to the values -1, 0, and 1, and its sign varies randomly such that it is uncorrelated with the other modulation sequences and with the pipelined ADC input signal. The modulation sequences associated with each DEM DAC are generated explicitly by the digital encoder, and are made available to the corresponding DNC logic block as digital signals.

Each DNC logic block performs two functions: 1) it estimates the 15 constants associated with the DAC noise from the corresponding DEM DAC and 2) it generates an estimate of the DAC noise by combining the 15 estimated constants and the 15 known modulation sequences. Each constant is estimated by multiplying the combined digital outputs of the appropriate pipeline stages by the corresponding modulation sequence and averaging the nonzero components of the resulting sequence. As shown in the next section, the average converges to the desired constant because the modulation sequence is uncorrelated to all but the term containing the constant.

The high-level topology of each DEM DAC is shown in Fig. 4. The digital encoder is comprised of the tree structure of 15 digital blocks labeled $S_{k,r}$ in the figure where $k = 1, \cdots, 4$ and $r = 1, \cdots, 8$. These blocks are referred to as *switching blocks*. The implementation details of the $S_{3,1}$ switching block are shown in Fig. 5 as an example of the general switching block architecture. As indicated in the figure, each pair of input bits is passed through a *swapper cell* that either passes the bits straight through to the output or interchanges their order depending upon whether the random control bit $q_{3,1}[n]$ is high or low [3]–[5]. All the swapper cells in a given switching

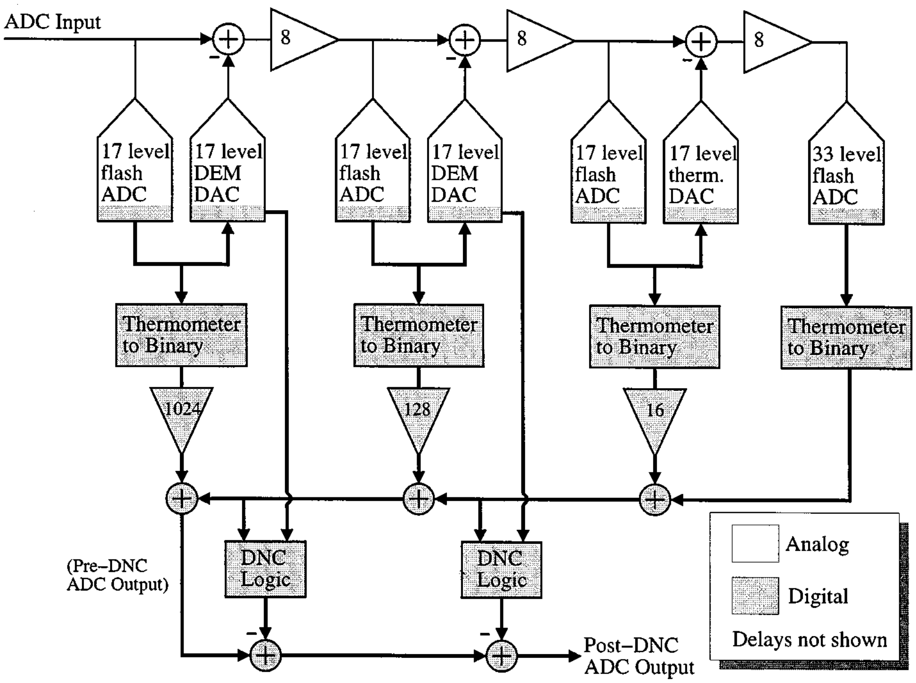


Fig. 3. Functional diagram of the example pipelined ADC with DNC applied to the first two stages.

block share the same random control bit, so the effect of each switching block is to pass all the even numbered input bits to the top output and all the odd numbered input bits to the bottom output, or vice versa, depending upon whether the random control bit is high or low. Each switching block, $S_{k,r}$, contains logic that sets the parity bit, $o_{k,r}[n]$, high if there are an odd number of input bits to the switching block that are high, and sets it low otherwise. The 15 parity bits and 15 random control bits associated with each DEM DAC are used by the DNC logic in estimating the error arising from the DAC noise as described shortly.

The switching blocks are all-digital devices, so they could be implemented using digital logic gates. However, in high-speed pipelined ADCs, the resulting propagation delay through the DAC might be unacceptably large. In such cases, a better approach is to use transmission gates for the switches in the swapper cells. Each transmission gate can be implemented using a pair of MOS transistors as shown in Fig. 6. The benefit of using transmission gates over conventional logic gates for the swapper cell switches is that they can be configured (i.e., each switch can be turned off or on) at a point in each clock cycle prior to the time at which the data is available from the flash ADCs. In this case conventional gate delays are avoided, and the delay through the digital encoder is determined mainly by the RC time constants resulting from the on-resistance and stray capacitances associated with each switch. Conventional digital combinational logic can be used to implement the even/odd detection logic in each switching block because propagation delay is not an issue for the parity bits.

As indicated in Fig. 3, the DNC logic block associated with the first stage has two input buses: one from the DEM DAC in the first stage, and one from the combined digital outputs of the three stages following the first stage. The data bus from the DEM DAC consists of the 15 random bits, $\{q_{k,r}[n]\}$, and the 15 parity bits, $\{o_{k,r}[n]\}$, described above. The other bus represents a value referred to as the *digitized residue* of the first stage. Similarly, the DNC logic block associated with the second stage has as its inputs a bus consisting of the 15 random bits and 15 parity bits from the DEM DAC in the second stage, and a bus representing the digitized residue of the second stage. It can be verified from the details of Fig. 3 that the bus representing the digitized residue of the first stage is 12 bits wide, and that representing the digitized residue of the second stage is 9 bits wide.

The details of the DNC logic block associated with the first stage of the pipeline are shown in Figs. 7 and 8. As indicated in Fig. 7, a 10-bit random number is added to the digitized residue, and the result is requantized to a three-level sequence according to

$$q[n] = \begin{cases} 1, & \text{if } s[n] \geq 512 \\ -1, & \text{if } s[n] < -512 \\ 0, & \text{otherwise} \end{cases}$$

where $s[n]$ and $q[n]$ are the input and output, respectively, of the requantizer. Thus, the quantizer acts as an ideal unity-gain mid-tread quantizer followed by a scale factor of 2^{-10}.

The remainder of the DNC logic consists of 15 parallel *channels* that all operate on the three level signal from the requantizer. The outputs of the channels are added together to generate the estimate of the error arising from the noise introduced by the first-stage DEM DAC.

Each channel corresponds to one of the switching blocks in the first-stage DEM DAC. As shown in the figure, the three-level input to each channel is first effectively multiplied by 1, 0, or -1 depending upon the states of the random bit and the parity bit from the corresponding switching block of the DEM DAC. The resulting sequence is operated on by the block labeled *averager* in Fig. 7, and the output of the averager is subjected to

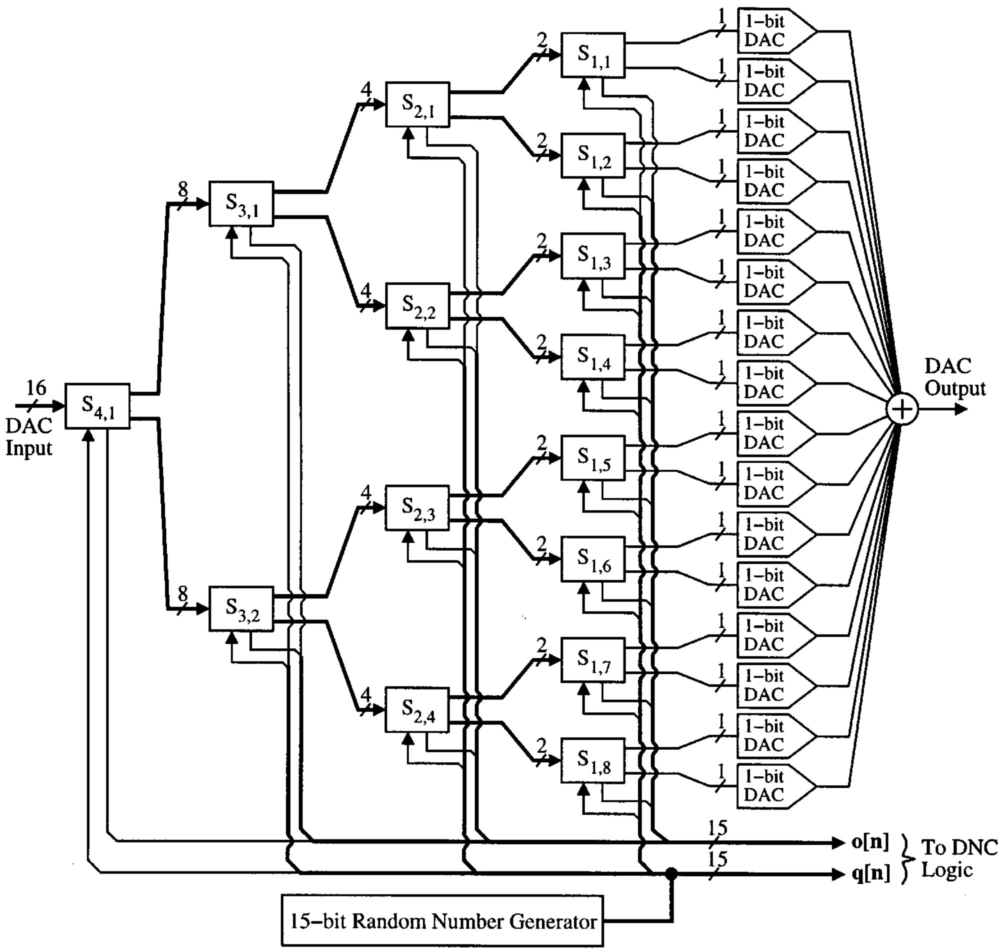


Fig. 4. Topology of the 17-level DEM DAC used in each of the first two stages of the pipelined ADC.

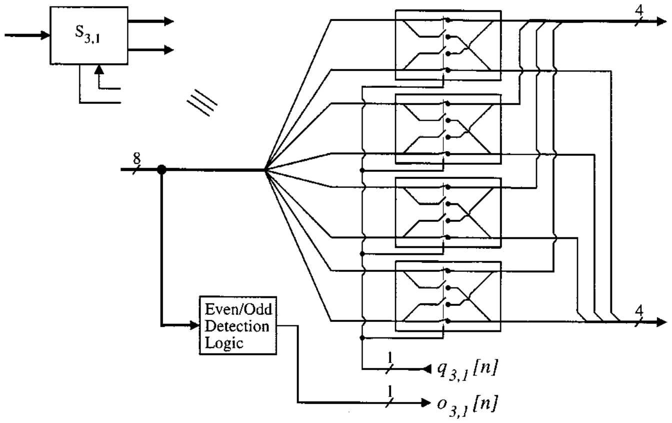


Fig. 5. Implementation details of the $S_{3,1}$ switching block as an example of the general switching block structure.

the same 1, 0, or -1 multiplication as applied to the sequence at the input to the channel. The averager samples its input sequence each time the parity bit $o_{k,r}[n]$ is high, and adds the value to that of an internal accumulator. Each time 2^{25} such samples have been accumulated, the averager updates its output register with the accumulated value divided by 2^{15} (i.e., right-shifted by 15-bit positions) and truncated to the five most significant bits (MSBs). It then clears the accumulator and begins the accumulation process again; the output register is not updated again until the next 2^{25} samples have been accumulated. The 2^{-10} scale factor imposed by the requantizer in combination with the 2^{-15} scale factor imposed by the averager results in a combined scale factor of 2^{-25}. Thus, a true average of each set of 2^{25} data samples is ultimately performed by each channel.

A hardware-efficient implementation of each averager in Fig. 7 is shown in Fig. 8. The three-level input sequence, labeled $v_{k,r}[n]$ in the figure, is interpreted as a two's complement value. The 18-bit up/down counter performs the accumulation, and the 25-bit up-counter determines when each complete set of 2^{25} samples has been accumulated.

The above discussion, and the details of Figs. 7 and 8 apply to the DNC logic associated with the first stage of the pipeline. The DNC logic associated with the second stage of the pipeline is identical except that most of the bus and register widths are smaller. Specifically, the digitized residue bus width is 9 bits, the random number bus width is 7 bits, the up-down counter width is 15 bits, and the up-counter width is 20 bits.

Fig. 9(a) shows a PSD plot of the output of the pipelined ADC simulated with the DNC technique as depicted in Figs. 3–8. The simulated circuit errors and input signal were the same as those used to obtain the results shown in Fig. 2. For comparison purposes, Fig. 2(b) and (c) are duplicated as Fig. 9(b) and (c), respectively; Fig. 9(b) shows results corresponding to Fig. 9(a) except without DNC, and Fig. 9(c) shows the effect of ideally

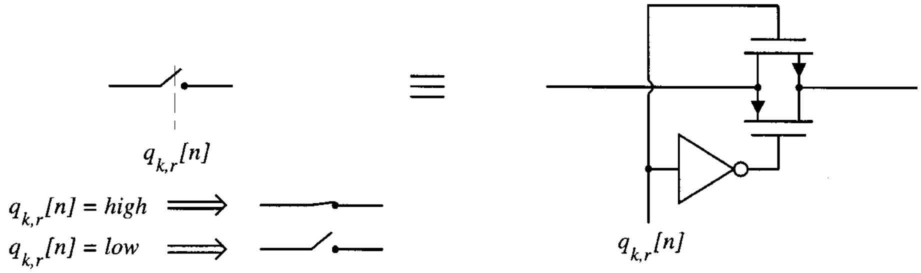


Fig. 6. Transmission gate implementation of the swapper cell switches (the switch with the opposite polarity is obtained by interchanging the n-channel and p-channel transistors).

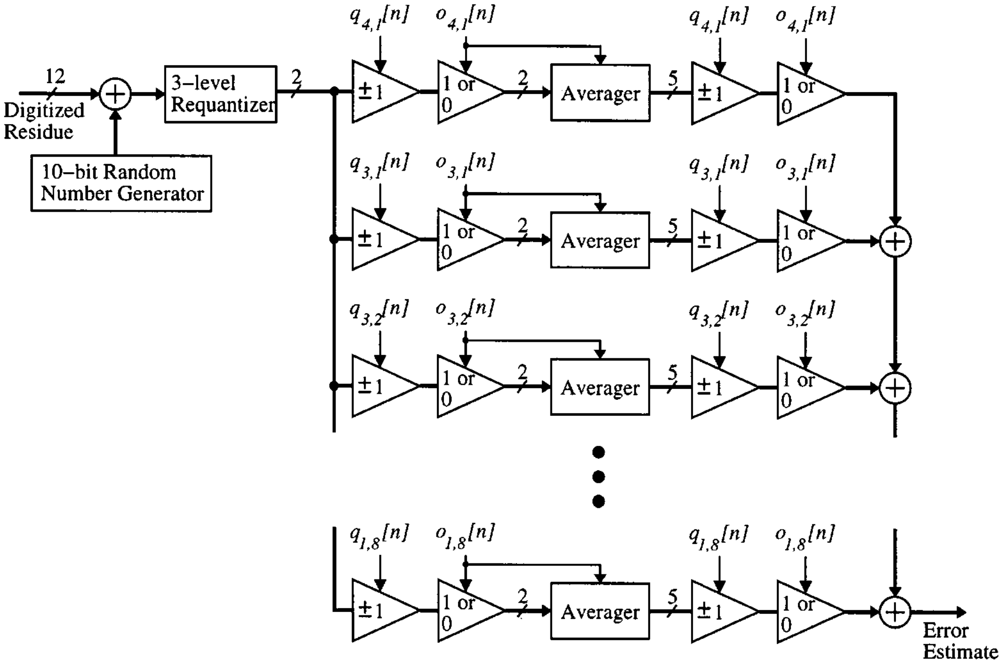


Fig. 7. Implementation details of the DNC logic for the first pipeline stage. Each of the 15 "channels" corresponds to one of the switching blocks of the DEM DAC.

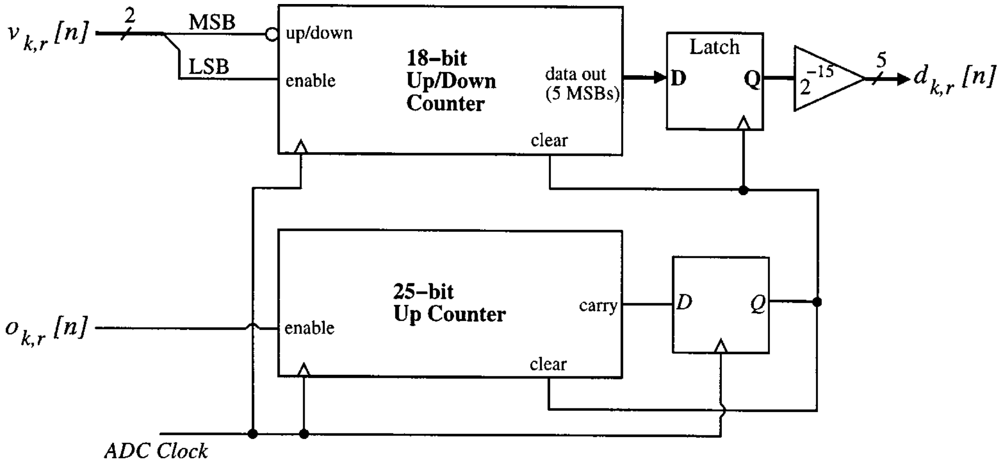


Fig. 8. Implementation details of the DNC averager for the first pipeline stage.

removing the error components associated with the DAC noise introduced in the first two pipeline stages. The A/D conversion precision associated with the results of Fig. 9(a) is 13.3 bits whereas those of Figs. 9(b) and (c) are 10.4 bits and 13.5 bits, respectively. Thus, the DNC technique resulted in an overall improvement of 2.9 bits relative to the case without DNC. The 13.5 bits of A/D conversion precision achieved by ideally removing the error sequences arising from DAC noise introduced in the first two pipeline stages represents the limit of the performance improvement that could be achieved by the DNC technique for

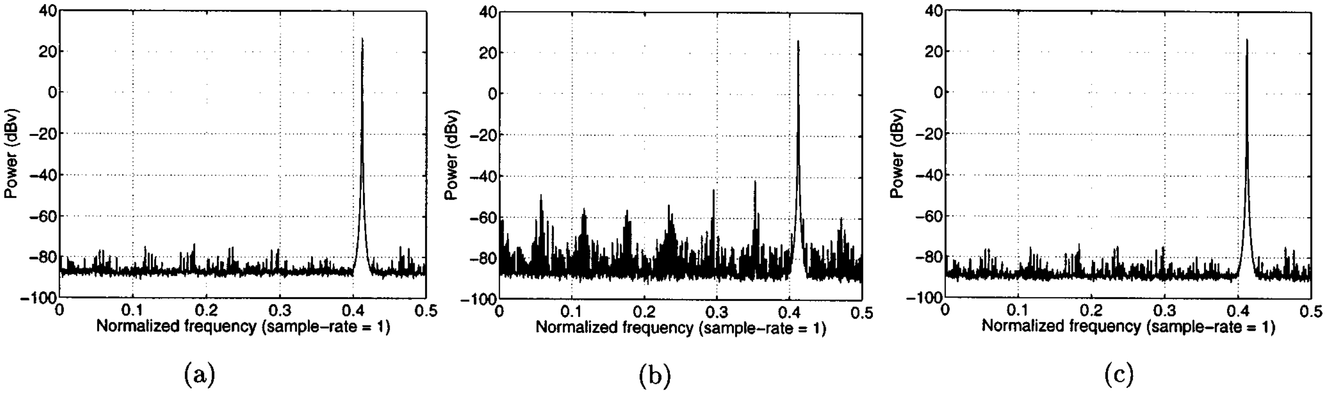


Fig. 9. Power spectral density plots of the output of the pipelined ADC simulated: (a) with the proposed DNC technique applied to the first two stages of the pipeline, (b) without DNC, and (c) with ideal post-ADC removal of the error components arising from the DAC noise introduced in the first two stages of the pipeline.

the input signal and circuit errors simulated. As explained in the next section, allowing the averagers in the DNC logic to average a larger number of samples causes the A/D conversion precision of the ADC with DNC to more closely approach this limit.

The simulation results shown in Fig. 9 correspond to a full-scale input sinusoid. Smaller amplitude input signals tend to result in faster convergence and smaller spurious components and this translates into a lower noise floor in the ADC output. Numerous simulations have been performed with various input signals and random errors. On average, the DNC technique implemented as described above results in a performance improvement of approximately 2.7 bits relative to the same pipelined ADC without DNC.

IV. Overview of the DNC Signal Processing Theory

This section describes the DNC technique from a signal processing point of view in the context of the pipelined ADC presented in the previous section. The signal processing details of the DNC technique as applied to the first and second stages of the pipelined ADC are almost identical, so only the first-stage DNC is considered in this section.

The last three stages of the pipelined ADC digitize the analog residue from the first stage of the pipeline, and therefore they can be viewed as an ADC in their own right. This is depicted in Fig. 10 wherein the pipelined ADC is shown with the second through fourth stages represented by an equivalent ADC, denoted as ADC_{2-4}. As described in the previous section, with ideal components the overall pipelined ADC has a quantization step-size of $1.6 \cdot 2^{-14}$ V, and a first interstage gain of eight. Therefore, the A/D conversion performed by the last three stages of the pipeline has a nominal quantization step-size of eight times that of the overall pipelined ADC, namely $1.6 \cdot 2^{-11}$ V. In the absence of nonideal circuit behavior, it follows that the digitized residue of the first stage and the scaled digital output of the first stage can be written as

$$r_1[n]|_{\mathrm{ideal}} = e_{\mathrm{ADC}_{2-4}}[n] - 2^{10} \cdot e_{\mathrm{ADC}_1}[n] \tag{1}$$

and

$$y_1[n]|_{\mathrm{ideal}} = \frac{2^{14}}{1.6} \cdot V_{\mathrm{in}}[n] + 2^{10} \cdot e_{\mathrm{ADC}_1}[n] \tag{2}$$

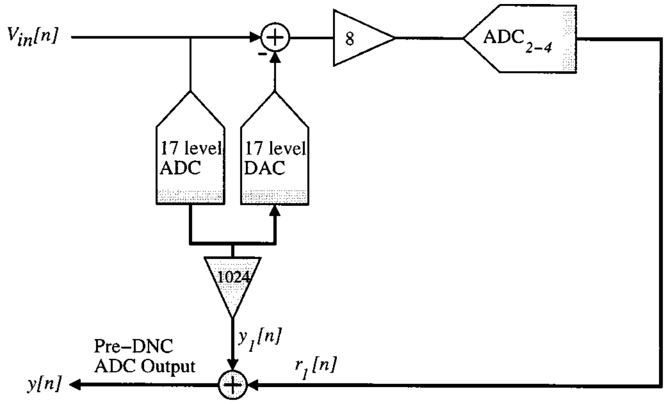


Fig. 10. The pipelined ADC with the last three stages represented by a single ADC denoted as ADC_{2-4}.

respectively, where $e_{\mathrm{ADC}_{2-4}}[n]$ is the quantization noise from the last three stages of the overall pipelined ADC, and $e_{\mathrm{ADC}_1}[n]$ is the quantization noise from the 17-level ADC in the first stage.[2] Therefore, in the absence of nonideal circuit behavior, the output of the overall pipelined ADC would be the sum of (1) and (2), which reduces to

$$y[n]|_{\mathrm{ideal}} = \frac{2^{14}}{1.6} \cdot V_{\mathrm{in}}[n] + e_{\mathrm{ADC}_{2-4}}[n].$$

It follows from the presentation in Section II that this is a 14-bit representation of the input signal, $V_{\mathrm{in}}[n]$.

In the presence of nonideal circuit behavior, the situation is more complicated. To quantify this statement, consider the pipelined ADC with the following types of nonideal component behavior: 1) interstage amplifier gain errors; 2) errors in the output levels of the 1-bit DACs; and 3) errors in the quantization threshold levels of the flash ADCs. In switched-capacitor circuits, these errors arise largely from capacitor ratio mismatches, so it is reasonable to assume that they are nearly invariant over time.

[2] Recall from Section II that the LSB of each A/D conversion is interpreted as unity which implies that the gain of the A/D conversion process is equal to the inverse of the quantization step-size. A consequence of this normalization is that $e_{\mathrm{ADC}_1}[n]$ and $e_{\mathrm{ADC}_{2-4}}[n]$ are both bounded in magnitude by 0.5 (half an LSB). By similar reasoning, the gain of each DAC is equal to the DAC step-size.

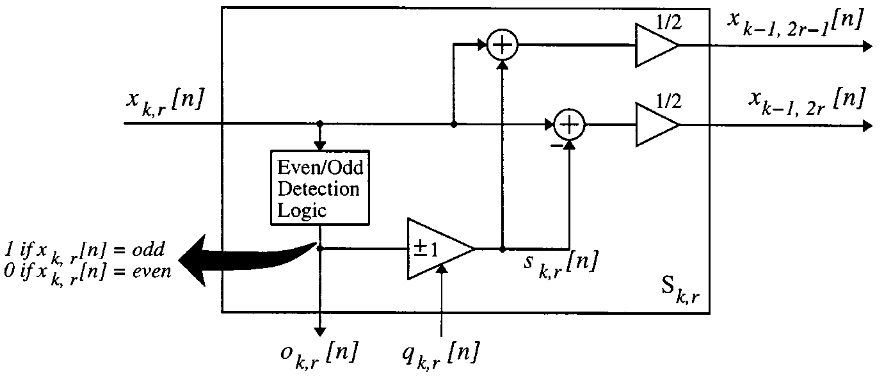


Fig. 11. The signal processing operations performed by each switching block (implemented as shown in Fig. 5).

For multi-bit DACs consisting of several 1-bit DACs, as considered in this paper, errors in the 1-bit DACs give rise to a gain error and a signal-dependent DAC noise component. Thus, the analog output sequence, say $x_a[n]$, of any such DAC can be written in terms of its digital input sequence, say $x_d[n]$, as

$$x_a[n] = \Delta_{\mathrm{DAC}}(1+\varepsilon_{\mathrm{DAC}})\cdot x_d[n] + e_{\mathrm{DAC}}[n] \tag{3}$$

where Δ_{DAC} is the nominal step-size of the DAC, $\varepsilon_{\mathrm{DAC}}$ is a constant *DAC gain error* and $e_{\mathrm{DAC}}[n]$ is the DAC noise. In the absence of 1-bit DAC errors, $\varepsilon_{\mathrm{DAC}}$ and $e_{\mathrm{DAC}}[n]$ would both be zero.

Similarly, the digital output sequence $y_d[n]$ of any ADC can be written in terms of its analog input sequence $y_a[n]$, as

$$y_d[n] = \frac{1}{\Delta_{\mathrm{ADC}}}(1+\varepsilon_{\mathrm{ADC}})\cdot y_a[n] + e_{\mathrm{ADC}}[n] \tag{4}$$

where Δ_{ADC} is the nominal step-size of the ADC, $\varepsilon_{\mathrm{ADC}}$ is a constant *ADC gain error*, and $e_{\mathrm{ADC}}[n]$ is the ADC noise. In the absence of nonideal circuit behavior, $\varepsilon_{\mathrm{ADC}}$ would be zero, and $e_{\mathrm{ADC}}[n]$ would be uniform quantization noise. However, errors in the quantization threshold levels cause the ADC gain error to be nonzero, and cause $e_{\mathrm{ADC}}[n]$ to represent nonuniform quantization noise.

It is straightforward to derive expressions for $r_1[n]$ and $y_1[n]$ in the system of Fig. 10 with the DAC and ADCs modeled using (3) and (4), respectively, and with the first interstage gain set to $8\cdot(1+\varepsilon_1)$, where ε_1 represents the interstage gain error. With respect to the DNC logic, $r_1[n]$ is the sequence of interest because 1-bit DAC errors in the first-stage DAC give rise to error in $r_1[n]$ but have no effect on $y_1[n]$. The expression for $r_1[n]$ is

$$r_1[n] = r_1[n]|_{\mathrm{ideal}} - \gamma\cdot e_{\mathrm{DAC}_1}[n] + v[n] \tag{5}$$

where γ is a constant, $e_{\mathrm{DAC}_1}[n]$ is the DAC noise introduced by the 17-level DAC in the first pipeline stage, and $v[n]$ is a function that does not depend upon $e_{\mathrm{DAC}_1}[n]$. Specifically

$$\gamma = \frac{2^{14}}{1.6}\cdot\left(1+\varepsilon_{\mathrm{ADC}_{2-4}}\right)\cdot(1+\varepsilon_1)$$

where $\varepsilon_{\mathrm{ADC}_{2-4}}$ is gain error of the A/D conversion performed by the last three pipeline stages, and

$$\begin{aligned} v[n] = {} & \left[2^{10} - 1.6\cdot 2^{-4}\cdot\gamma\cdot\left(1+\varepsilon_{\mathrm{DAC}_1}\right)\right]\cdot e_{\mathrm{ADC}_1}[n] \\ & -\gamma\cdot\left(\varepsilon_{\mathrm{ADC}_1}+\varepsilon_{\mathrm{DAC}_1}+\varepsilon_{\mathrm{ADC}_1}\varepsilon_{\mathrm{DAC}_1}\right)V_{\mathrm{in}}[n] \end{aligned} \tag{6}$$

where $\varepsilon_{\mathrm{ADC}_1}$ and $\varepsilon_{\mathrm{DAC}_1}$ are the gain errors of the first-stage ADC and DAC, respectively.

From (5) and (6), it is evident that both the DAC noise, $e_{\mathrm{DAC}_1}[n]$, and DAC gain error, $\varepsilon_{\mathrm{DAC}_1}$, arising from the 1-bit DAC errors in the first-stage DAC affect the pre-DNC pipelined ADC output, $y[n]$. Specifically, $r_1[n]$, and therefore $y[n]$, contains a term proportional to the first-stage DAC noise, and two terms corresponding to the first-stage DAC gain error. One of the terms corresponding to the first-stage DAC gain error is a scaled version of the pipelined ADC input signal, $V_{\mathrm{in}}[n]$, and the other term is a scaled version of the first-stage ADC error, $e_{\mathrm{ADC}_1}[n]$. The term proportional to $V_{\mathrm{in}}[n]$ only causes a slight gain error through the pipelined ADC, so typically it is not a problem. The term proportional to $e_{\mathrm{ADC}_1}[n]$ represents error identical in structure and similar in magnitude to that caused by the first interstage gain error, ε_1. Simulations indicate that the effect of the first-stage DAC noise is far more significant than that of the first-stage DAC gain error. For example, the first-stage DAC gain error was not corrected in obtaining the results shown in Figs. 2 and 9. Therefore, the objective of the first-stage DNC logic is to estimate $\gamma\cdot e_{\mathrm{DAC}_1}[n]$ so that it may be removed from the output of the pipelined ADC.

The DEM DACs are designed to modulate their DAC noise sequences in a fashion that facilitates estimation by the DNC logic of the $\gamma\cdot e_{\mathrm{DAC}_1}[n]$ term in $r_1[n]$. Interpreting the input to each switching block of the DEM DACs as a thermometer encoded sequence (e.g., see Fig. 5), it is straightforward to verify that the switching block performs the signal processing operations shown in Fig. 11. Thus, the outputs of the switching block can be written as

$$x_{k-1,2r-1}[n] = \frac{1}{2}(x_{k,r}[n] + s_{k,r}[n])$$

and

$$x_{k-1,2r}[n] = \frac{1}{2}(x_{k,r}[n] - s_{k,r}[n]) \tag{7}$$

where

$$s_{k,r}[n] = \begin{cases} 1, & \text{if } o_{k,r}[n]=1 \quad \text{and} \quad q_{k,r}[n]=1 \\ -1, & \text{if } o_{k,r}[n]=1 \quad \text{and} \quad q_{k,r}[n]=0 \\ 0, & \text{if } o_{k,r}[n]=0. \end{cases} \tag{8}$$

As shown in [6], the DAC noise introduced by DACs of the form shown in Fig. 4 with switching blocks that behave according to (7) has the form

$$e_{\mathrm{DAC}}[n] = \sum_{k=1}^{4}\sum_{r=1}^{2^{4-k}} \Delta_{k,r}s_{k,r}[n] + \beta \tag{9}$$

where $\Delta_{k,r}$ for $k = 1, \cdots, 4$ and $r = 1, \cdots, 8$ and β depend only on the errors introduced by the 1-bit DACs. If the errors in the two output levels of each 1-bit DAC do not change as a function of time, as assumed in this paper, then the $\Delta_{k,r}$ and β are all constants. The β term in (9) generally is not a problem because it contributes only a constant offset to the overall pipelined ADC output, but the $\Delta_{k,r}$ constants are modulated and therefore degrade the signal-to-noise-ratio (SNR) of the pre-DNC pipelined ADC output, $y[n]$.

Without the randomization introduced by the $q_{k,r}[n]$ sequences, the DAC noise would introduce harmonic distortion, and would thus limit the spurious-free-dynamic-range (SFDR) of the pipelined ADC. For example, with all the $q_{k,r}[n]$ sequences held low, the DEM DAC is equivalent to a thermometer encoded DAC, yet (8) and (9) remain valid. In this case, it follows from (8) that the $s_{k,r}[n]$ sequences are deterministic functions of $V_{\text{in}}[n]$. The $s_{k,r}[n]$ sequences modulate the $\Delta_{k,r}$ constants, so the DAC noise from a thermometer encoded DAC is a complicated, yet deterministic, function of $V_{\text{in}}[n]$, i.e., it represents pure harmonic distortion.

In contrast, if the $q_{k,r}[n]$ sequences well approximate independent, zero-mean, white random sequences restricted to the values 0 and 1, as tacitly assumed in Section III and throughout the remainder of the paper, it can be verified from (8) and (9) that the DAC noise is white and uncorrelated with $V_{\text{in}}[n]$. Even without DNC, an immediate benefit of this result is that the DAC noise does not contribute harmonic distortion to $y[n]$, so the DEM DACs do not limit the SFDR of the pipelined ADC.

The randomness properties of the $q_{k,r}[n]$ sequences also facilitate the estimation by the DNC logic of the DAC noise from $r_1[n]$ and the $s_{k,r}[n]$ sequences. It follows from the $q_{k,r}[n]$ sequence properties and (8) that the $s_{k,r}[n]$ sequences are white, zero-mean, uncorrelated with each other and with $V_{\text{in}}[n]$, and only assume values from the set $\{-1, 0, 1\}$. Therefore, it follows from (9) that the DAC noise has a form similar to the sum of 15 direct-sequence spread-spectrum signals wherein each constant, $\Delta_{k,r}$, is one of the 15 "message signals" and the corresponding $s_{k,r}[n]$ is the "spreading code." Thus, from (5) it follows that $r_1[n]$ can be viewed as the 15 spread-spectrum signals scaled by $-\gamma$ plus uncorrelated "interference signals." The DNC logic operates in a fashion similar to a spread-spectrum receiver to estimate each $-\gamma \cdot \Delta_{k,r}$ constant and uses the estimated constants to construct an estimate of $-\gamma \cdot e_{\text{DAC}_1}[n]$.

To estimate each $-\gamma \cdot \Delta_{k,r}$ constant, the DNC logic averages the product of the corresponding $s_{k,r}[n]$ sequence and the digitized residue of the first stage, $r_1[n]$, over a large number of sample times, n, for which $|s_{k,r}[n]| = 1$. Such an average taken over N time samples starting from an arbitrary sample time $n = n_0$ can be written as

$$d_{k,r}[N] = \frac{\sum_{i=0}^{N-1} r_1[n_0+i] \cdot s_{k,r}[n_0+i]}{\sum_{i=0}^{N-1} |s_{k,r}[n_0+i]|}. \qquad (10)$$

For brevity in the sequel, the following shorthand notation is used for this type of averaging operation:

$$d_{k,r}[N] = \underset{s_{k,r}[n]\neq 0}{\text{Average}} \{r_1[n] \cdot s_{k,r}[n]\}[N].$$

As described above, the $s_{k,r}[n]$ sequences are well approximated as independent, white, zero-mean sequences that are independent of $V_{\text{in}}[n]$ and are each restricted to the set of values $\{-1, 0, 1\}$. For the time being, suppose further that the average power of each $s_{k,r}[n]$ sequence is nonzero, so that

$$\sum_{i=0}^{N-1} |s_{k,r}[n_0+i]| \rightarrow \infty \qquad (11)$$

as $N \rightarrow \infty$. These properties imply

$$\underset{s_{k,r}[n]\neq 0}{\text{Average}} \{s_{k',r'}[n] \cdot s_{k,r}[n]\}[N] \rightarrow \begin{cases} 1, & \text{if } k = k' \quad \text{and} \quad r = r' \\ 0, & \text{otherwise} \end{cases}$$

as $N \rightarrow \infty$ by virtue of the Law of Large Numbers. By similar reasoning, for any sequence $f[n]$ that is uncorrelated with $s_{k,r}[n]$

$$\underset{s_{k,r}[n]\neq 0}{\text{Average}} \{f[n] \cdot s_{k,r}[n]\}[N] \rightarrow 0$$

as $N \rightarrow \infty$.

As is evident from (1), (5), (6), and (9), for a given k and r the only terms in $r_1[n]$ that are not completely uncorrelated with $s_{k,r}[n]$ are $-\gamma \cdot \Delta_{k,r}$ and $e_{\text{ADC}_{2-4}}[n]$. Since $e_{\text{ADC}_{2-4}}[n]$ has a slight dependence on the first stage DAC noise, it is likely to have some nonzero correlation with $s_{k,r}[n]$. However, the $e_{\text{ADC}_{2-4}}[n]$ term has a very small variance, and simulations indicate that its correlation to $s_{k,r}[n]$ is negligible compared to the 14-bit noise floor of the overall pipelined ADC. Therefore, to a very good approximation

$$d_{k,r}[N] \rightarrow -\gamma \cdot \Delta_{k,r} \qquad (12)$$

as $N \rightarrow \infty$.

The derivation leading to (12) relies upon (11), but there exist input sequences, $V_{\text{in}}[n]$, for which (11) does not hold for all values of k, and r. For example, if $V_{\text{in}}[n]$ were such that the output of the first-stage ADC is always an even number, then $s_{4,1}[n] = 0$ for all n so (11) would not hold for $k = 4$ and $r = 1$. However, whenever (11) does not hold the corresponding $s_{k,r}[n]$ term has zero average power. Therefore, the corresponding $-\gamma \cdot \Delta_{k,r}$ need not be estimated by the DNC logic, because the associated 1-bit DAC errors do not contribute average power to the DAC noise.

It follows that for sufficiently large values of N, $-\gamma \cdot e_{\text{DAC}_1}[n]$ is approximately equal to the sequence given by

$$c[n] = \sum_{k=1}^{4} \sum_{r=1}^{2^{4-k}} d_{k,r} s_{k,r}[n]. \qquad (13)$$

Consequently, subtracting $c[n]$ from the pre-DNC pipelined ADC output cancels most of the error arising from DAC noise introduced by the first stage DEM DAC. The accuracy with which the error is cancelled depends on accuracy with which each $d_{k,r}[N]$ converges to $-\gamma \cdot \Delta_{k,r}$ whenever the corresponding $s_{k,r}[n]$ satisfies (11). In each case, this depends upon the number of samples, M, for which $s_{k,r}[n] \neq 0$ in the set of sample times $n_0, n_0 + 1, \cdots, n_0 + N - 1$. For each k and r, convergence is only required when (11) holds, and in these cases $M \rightarrow \infty$ as $N \rightarrow \infty$.

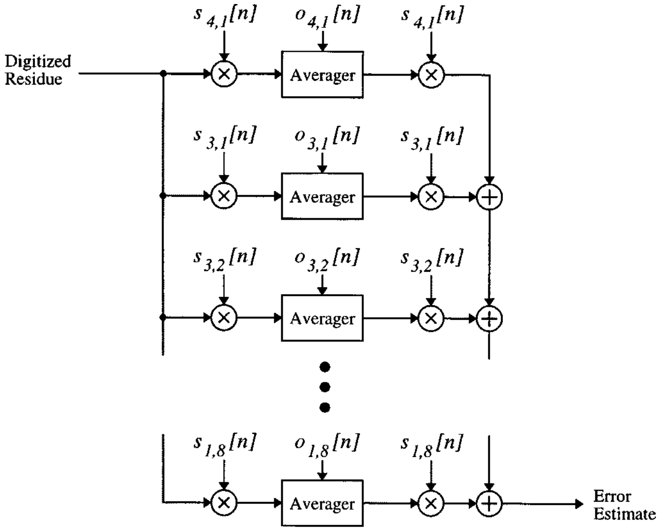


Fig. 12. The ideal signal processing operations performed by the DNC logic.

A block diagram that implements the signal processing operations described above is shown in Fig. 12. The first set of $s_{k,r}[n]$ multipliers together with the averagers calculate the $d_{k,r}[N]$ estimates, and the second set of $s_{k,r}[n]$ multipliers and the adders calculate $c[n]$. At the *startup* time, $n = n'$, no samples of $r_1[n]$ have been averaged, so $N = 0$, and $d_{k,r}[0] = 0$. At each sample time, $n > n'$, $d_{k,r}[N]$ is calculated using (10) with $n_0 = n'$ and $N = n - n'$. Thus ideal DAC noise cancellation is achieved in the limit as $n \rightarrow \infty$.

A key difference between the signal processing performed by the DNC logic described in the previous section and that shown in Fig. 12 is that the DNC logic performs *dithered* requantization of the digitized residue in order to simplify the averager hardware. The 10-bit random sequence added to the digitized residue is used as a *dither sequence* [7], [8]. To the extent that it well approximates a white random sequence that is uniformly distributed over its 1024 possible values and is independent of $r_1[n]$, it can be shown that the output of the three-level quantizer in the first stage DNC logic is given by $r_1'[n] = r_1[n] + u[n]$, where $u[n]$ is a white random sequence that is uncorrelated with $r_1[n]$. The DNC logic presented in Section III calculates each $d_{k,r}[N]$ using

$$d_{k,r}[N] = \frac{\sum_{i=0}^{N-1} r_1'[n_0 + i] \cdot s_{k,r}[n_0 + i]}{\sum_{i=0}^{N-1} |s_{k,r}[n_0 + i]|}. \tag{14}$$

instead of (10). Since $u[n]$ is uncorrelated with $r_1[n]$, and therefore with each $s_{k,r}[n]$, it follows from the arguments above that the $d_{k,r}[N]$ sequences given by (14) converge to the same values as those given by (10). However, the convergence rate is somewhat slower because the $u[n]$ sequence gives rise to a term in (14) that converges to zero at a finite rate and this term is not present in (10).

The only other difference between the signal processing performed by the DNC logic described in the previous section and that shown in Fig. 12 is that the DNC logic does not calculate the $d_{k,r}[N]$ values for ever-increasing values of N, so perfect convergence does not occur. Instead, in every contiguous set of sample-times during which $|s_{k,r}[n]| = 1$ exactly $M = 2^{25}$ times, the corresponding DNC averager calculates $d_{k,r}[N]$ as given by (14). The output of the averager is preset to zero, and is not updated until the first M samples for which $|s_{k,r}[n]| = 1$ have occurred. Once the M samples have occurred, the averager updates its output register with the calculated value of $d_{k,r}[N]$ and begins averaging the next set of input values. Thus, the actual number of samples, N, in each set over which the averager calculates $d_{k,r}[N]$ is a function of the $s_{k,r}[n]$ sequence. The reason for this design choice is that M is the value that determines the accuracy with which $d_{k,r}[N]$ approximates $-\gamma \cdot \Delta_{k,r}$ as described above.

Once the first set of $d_{k,r}[N]$ values have been calculated for all k and r, the DNC logic estimates the DAC noise to an accuracy that is independent of $V_{\text{in}}[n]$ provided the 1-bit DAC errors and the interstage gain errors do not change rapidly over time. In typical applications of CMOS switched-capacitor based pipelined ADCs, it is reasonable to assume that the 1-bit DAC errors and interstage gain errors do not change significantly over periods of time that are long compared to the period of the lowest signal frequency of interest. However, slow variations of these component errors may occur in response to environmental factors such as changes in temperature. To adjust to such variations, the DNC logic continuously calculates new $d_{k,r}[N]$ values during the normal operation of the pipelined ADC. Nevertheless, the DNC logic reaches its full level of accuracy once the first set of $d_{k,r}[N]$ values have been calculated for all k and r; at this point, the DNC logic continues to operate simply to maintain this level of accuracy.

V. Convergence Simulations

Simulation results that show the post-DNC pipelined ADC noise floor (i.e., the total mean-squared error of the pipelined ADC output after DAC noise cancellation) as a function of M for different choices of $V_{\text{in}}[n]$ are shown in Fig. 13. The same set of 1-bit DAC errors, interstage gain errors, and flash ADC errors as used in the simulations described in Sections II and III were used for all the simulations associated with Fig. 13. Each graph in Fig. 13 shows results corresponding to a sinusoidal $V_{\text{in}}[n]$ with a unique amplitude and frequency combination as labeled on the graph. The labeled values of M correspond to the first-stage DNC logic and range from 2^{17} to 2^{27}. In each simulation run, the value of M used in the second-stage DNC logic was 32 times smaller than that of the first stage. The different solid curves in each graph correspond to different initial conditions of the random number generators internal to the pipelined ADC; otherwise the simulations used to generate the curves were identical. The dashed line in each graph indicates the result of ideally removing from the pipelined ADC output the contribution from the DAC noise sequences introduced by the first-stage and second-stage DEM DACs. Thus, the dashed line indicates the ideal performance that could be expected from the DNC technique in the limit as $M \rightarrow \infty$.

As expected, the simulation results indicate the accuracy of the DNC technique approaches that of ideal DAC noise cancellation as M increases. The results indicate that increasing M above 2^{25} has a diminishing effect on the post-DNC pipelined ADC noise floor. It follows from the theory presented in the previous section that for such large values of M the post-DNC

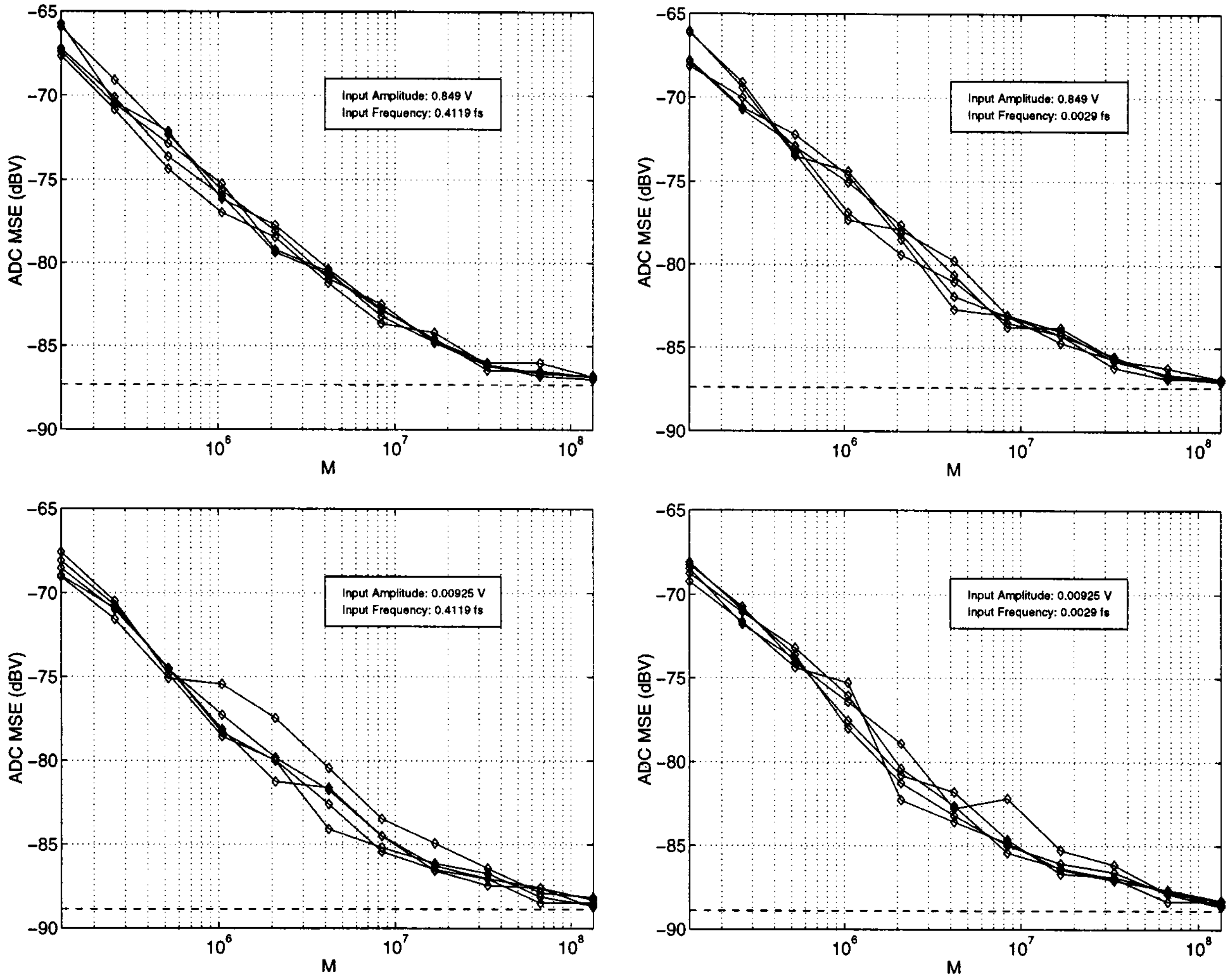


Fig. 13. The post-DNC pipelined ADC noise floor as a function of M for various sinusoidal $V_{\text{in}}[n]$ and various initial states of the random number generators associated with the DEM DACs and DNC logic.

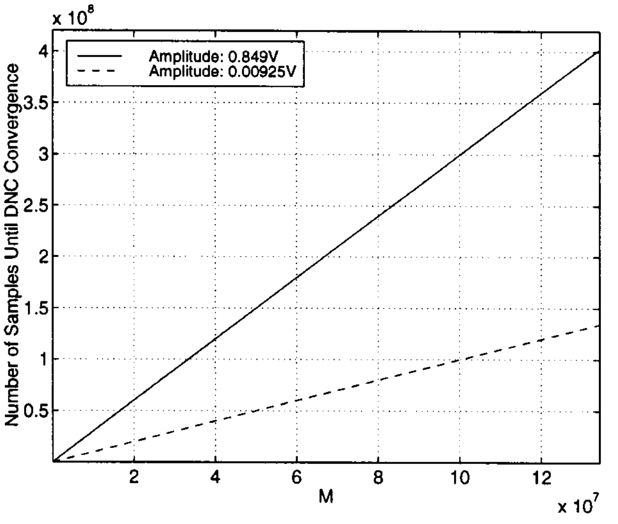


Fig. 14. The number of samples required to calculate all of the nonzero $d_{k,r}[N]$ values in the computer simulations associated with Fig. 13.

pipelined ADC noise floor is dominated by the effects of the various gain errors.

Fig. 14 shows the number of time samples required to calculate the first set of nonzero $d_{k,r}[N]$ values as a function of M for all the simulation runs associated with Fig. 13. The solid line in the figure corresponds to all of the simulation runs for which $V_{\text{in}}[n]$ had an amplitude of 0.849 V, and the dashed line corresponds to all of the simulation runs for which $V_{\text{in}}[n]$ had an amplitude of 0.009 25 V (the slight differences among the sets of data generated by these simulation runs are not visible given the large scale of the plot). The simulation results suggest that the number of time samples required to calculate the necessary $d_{k,r}[N]$ values increases linearly with M with a slope that depends on the amplitude of $V_{\text{in}}[n]$.

For the two sets of simulation runs in which $V_{\text{in}}[n]$ had an amplitude of 0.009 25 V, the $s_{k,r}[n]$ sequences for certain values of k and r in the first-stage DAC of the pipelined ADC were always zero, so the corresponding DNC averager outputs remained fixed at zero. For these values of k and r, $V_{\text{in}}[n]$ was such that (11) was not satisfied, so, as described above, it is reasonable to consider the DNC logic to have fully converged once all the nonzero $d_{k,r}[n]$ values are calculated. Of course, if after this time period $V_{\text{in}}[n]$ were to change such that (11) is satisfied for additional values of k and r, then the corresponding $d_{k,r}[n]$ values would be calculated by the DNC logic and the total convergence would time would be greater than that indicated by the dashed line in Fig. 14.

The simulation results shown in Figs. 13 and 14 are representative of the types of convergence behavior exhibited by the pipelined ADC with DNC presented in Section III. In general, the DNC convergence is a strong function of the input amplitude and offset, and tends to be a weaker function of the how rapidly the input signal varies.

VI. Conclusion

Switched-capacitor based pipelined ADCs tend to be highly sensitive to noise arising from component mismatches in their internal DACs. An all-digital technique, referred to as DNC, that continuously measures and cancels the A/D conversion error caused by such DAC noise has been presented. The details of the DNC approach have been described in the context of a particular pipelined ADC topology, although the approach is generally applicable to pipelined ADCs with multi-bit DACs. The ideal A/D conversion precision of the example pipelined ADC is 14.1 bits, but with realistic component matching and without the DNC technique, its typical simulated A/D conversion precision is 10.4 bits. With the DNC technique, its typical simulated A/D conversion precision increases to 13.3 bits. The hardware required to implement the DNC technique in the example pipelined ADC has been presented at both the signal processing level and the register-transfer-level. The associated digital hardware complexity has been shown to be modest by modern VLSI standards.

References

[1] 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, pp. 954–961, Dec. 1987.

[2] S. Sutarja and P. R. Gray, “A pipelined 13-bit 250-ks/s 5-V analog-to-digital converter,” *IEEE J. Solid State Circuits*, vol. 23, pp. 1316–1323, Dec. 1988.

[3] L. R. Carley and J. Kenney, “A 16-bit 4'th order noise-shaping D/A converter,” in *IEEE Proc. CICC*, 1988, pp. 21.7.1–21.7.4.

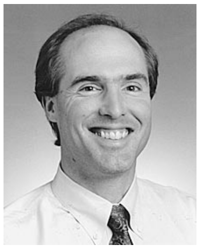
[4] L. R. Carley, “A noise shaping coder topology for $15+$ bits converters,” *IEEE J. Solid-State Circuits*, vol. 24, pp. 267–273, Apr. 1989.

[5] T. W. Kwan, R. W. Adams, and R. Libert, “A stereo multibit sigma delta DAC with asynchronous master-clock interface,” *IEEE J. Solid State Circuits*, vol. 31, pp. 1881–1887, Dec. 1996.

[6] I. Galton, “Spectral shaping of circuit errors in digital-to-analog converters,” *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 808–817, Oct. 1997.

[7] A. B. Sripad and D. L. Snyder, “A necessary and sufficient condition for quantization errors to be uniform and white,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 442–448, Oct. 1977.

[8] R. M. Gray and T. G. Stockham Jr., “Dithered quantizers,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 805–812, May 1993.



Ian Galton received the Sc.B. degree from Brown University, Providence, RI, in 1984, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, CA, in 1989 and 1992, respectively, all in electrical engineering.

He is currently an Associate Professor at the University of California at San Diego. He was formerly with the University of Californai at Irvine, NASA Jet Propulsion Laboratory, Acuson, and Mead Data Central, and has acted as a regular Consultant for several companies. His research interests involve integrated circuits and systems for communications, including ADCs, DACs, frequency-to-digital converters, and frequency synthesizers, and he holds four patents.

Dr. Galton was a recipient of the Caltech Charles Wilts Doctoral thesis prize.