Transaction Briefs

A New Algorithm for Cyclic and Pipeline Data Conversion

KENZO WATANABE, GABOR C. TEMES,
AND TOMOHISA TAGAMI

Abstract—A new algorithm is proposed for digital-to-analog conversion. The conversion starts with the most significant bit. This algorithm can be implemented using unity-gain buffers, and thus permits high-speed data conversion. Switched-capacitor D/A and A/D converter architectures based on this algorithm are described, and experimental results are presented to demonstrate its validity. Error analysis shows that a conversion accuracy of at least 9 bits is obtainable with a monolithic implementation. Computer simulations indicate that video-frequency operation may be possible if fine-line CMOS, bipolar or BiCMOS technology is used.

I. INTRODUCTION

In applications such as audio and video systems, digital signal processing is increasingly becoming prevalent, and hence analog-to-digital (A/D) and digital-to-analog (D/A) converters are needed for interfacing the real world with the digital world of signal processing. To meet the demands of applications, a wide variety of data converters has been developed [1]. Table I lists some of the most commonly used data conversion techniques [2], [3].

Among these techniques, the flash or parallel technique is used almost exclusively for A/D conversion in the video-frequency range. This type of converter, however, requires $2^n - 1$ comparators for an n -bit resolution, and thus it is difficult to incorporate it into ASIC's, or to extend the resolution beyond 8 bits. High-speed A/D conversion is possible also using time-multiplexed successive approximation or pipeline converters [4]. These converters achieve an n -bit resolution using only n comparators, but require auxiliary D/A converters which, starting with the most significant bit determined in the first stage, produce the threshold voltages of the comparators cyclically or sequentially. Such a cyclic D/A converter is currently not available, except for algorithmic converters used mostly in low-speed applications [5], [6]. Therefore, most successive approximation A/D converters use parallel D/A converters consisting of binary-weighted current sources. This results in a complicated circuit and high cost.

This paper describes a new algorithm which lends itself readily to cyclic and pipeline D/A conversion, and which can also provide fast, economical and accurate D/A subconverters in A/D converters. The basic algorithm, the resulting architectures, accuracy analysis, and some experimental results are described in the subsequent sections.

Manuscript received August 12, 1988; revised March 3, 1989. G. C. Temes was supported in part by the State of California, by the Xerox Corporation, and by the National Aeronautical and Space Administration under Grant MICRO 87-022 and Grant NCC 2-374. This paper was recommended by Associate Editor C. A. T. Salama.

K. Watanabe and T. Tagami are with the Research Institute of Electronics, Shizuoka University, Hamamatsu 432, Japan.

G. C. Temes is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90024.

IEEE Log Number 8931731.

TABLE I
COMMONLY USED DATA CONVERSION TECHNIQUES

Speed	D/A		A/D	
	Technique	Type	Technique	Type
Low	Pick-Off Ramp Pick-Off	Counter Counter	Dual-Slope Charge-Balance	Counter Counter
Moderate	Cyclic	Serial: LSB to MSB	Successive- Approximation Cyclic	Serial: MSB to LSB MSB to LSB
High	Resistive- Division	Parallel	Flash	Parallel
	Weighted- Current	Parallel	Pipeline	Serial: MSB to LSB
	R-2R Ladder	Parallel		

II. THE ITERATIVE D/A CONVERSION ALGORITHM

The bits b_i of an n -bit binary input number b and the corresponding analog voltage $V_a(n)$ in D/A conversion are related by

$$V_a(n) = (b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n}) V_r \quad (1)$$

where V_r is the reference voltage, and b_1 and b_n are the most significant bit (MSB) and least significant bit (LSB) of b , respectively. A cyclic or pipeline D/A conversion starting with the MSB generates the analog voltage sequence

$$V_a(i) = V_a(i-1) + b_i 2^{-i} V_r, \quad i=1, 2, \dots, n \quad (2)$$

where $V_a(0) = 0$. Introducing the reference voltage sequence $V_r(i)$ defined by the relation

$$V_r(i) - V_a(i) = \frac{V_r(i-1) - V_a(i-1)}{2} = 2^{-i} V_r \quad (3)$$

where $V_r(0) = V_r$, and noting that $V_a(i-1) = (b_i + \bar{b}_i) V_a(i-1)$, where $\bar{b}_i$ is the complement of b_i , one can rewrite (2) in the form

$$V_a(i) = \bar{b}_i V_a(i-1) + b_i \frac{V_r(i-1) + V_a(i-1)}{2} \quad (4)$$

Substituting (4) into (3), we get the matrix representation of the new D/A conversion algorithm:

$$\begin{pmatrix} V_r(i) \\ V_a(i) \end{pmatrix} = \begin{pmatrix} b_i + \bar{b}_i/2 & \bar{b}_i \\ b_i/2 & \bar{b}_i + b_i/2 \end{pmatrix} \begin{pmatrix} V_r(i-1) \\ V_a(i-1) \end{pmatrix} \quad (5)$$

Note that $V_r(i)$ and $V_a(i)$ represent upper and lower bounds on $V_a(n)$. These bounds converge as $i \rightarrow n$.

The schematic diagram of a possible 1-bit D/A converter circuit executing (5) is shown in Fig. 1. Many types of data converters, such as cyclic and pipeline D/A converters, and successive-approximation and pipeline A/D converters, can be realized using this unit configuration iteratively or in cascade. Their basic architectures were described in [7] and [8]. In CMOS realization, voltage averaging using capacitors is preferable. Such architectures will be given in the next section.

Fig. 1. The block diagram of the 1-bit D/A converter.

Fig. 2. The switched-capacitor cyclic D/A converter.

III. SWITCHED-CAPACITOR REALIZATION FOR THE CONVERTERS

Fig. 2 shows the circuit diagram of a cyclic D/A converter, which uses matched capacitors rather than resistors. Here, C_1 , C_2 , and C_3 are matched capacitors. C_1 stores the reference voltages $V_r(i)$, while C_3 holds the analog voltages $V_a(i)$. ϕ and $\bar{\phi}$ are the nonoverlapping two phase clocks. ϕ_R is the reset pulse used to set the initial values $V_r(0) = V_r$ and $V_a(0) = 0$ across C_1 and C_3 , respectively. If the value of b_i is 1, then C_2 is charged to $V_r(i-1)$ in the $\phi = "1"$ phase, and is then connected in parallel with C_3 in the subsequent $\bar{\phi} = "1"$ phase to update $V_a(i-1)$. During this time, C_1 continues to hold the previous reference voltage value $V_r(i-1)$. If, on the other hand, b_i is 0, then C_2 is charged to $V_a(i-1)$ and then connected in parallel with C_1 to update $V_r(i-1)$, while C_3 holds the previous value $V_a(i-1)$. Thus this stage in one cycle executes the algorithm of (5). D/A conversion starting with the LSB [9] is also possible, by modifying slightly the switching sequence.

The average voltage $V_i(i)$ between $V_r(i)$ and $V_a(i)$ is given by

$$V_i(i) = \frac{V_r(i) + V_a(i)}{2} = \frac{V_r}{2} - \sum_{j=1}^i (-1)^{b_j} 2^{-(j+1)} V_r. \quad (6)$$

This is just the threshold voltage with which the input analog voltage must be compared in the i th cycle of a successive-approximation A/D conversion. Expressing it in a recursive form:

$$V_i(i) = \frac{V_i(i-1) + b_i V_r(i-1) + \bar{b}_i V_a(i-1)}{2} \quad (7)$$

results, where $V_i(0) = V_r/2$. The reference and analog voltage sequences can then be rewritten as

$$V_r(i) = b_i V_r(i-1) + \bar{b}_i V_r(i-1) \quad (8)$$

and

$$V_a(i) = \bar{b}_i V_a(i-1) + b_i V_r(i-1) \quad (9)$$

Fig. 3. The switched-capacitor successive-approximation A/D converter.

respectively. A switched-capacitor successive-approximation A/D converter based on (7)–(9) is shown in Fig. 3. The matched capacitors C_3 and C_4 share charges in the $\bar{\phi} = "1"$ phase to produce the threshold voltage sequence $V_i(i)$ given by (7). Depending on the value of b_i determined earlier by the comparator, either $V_r(i)$ (stored across C_1) or $V_a(i)$ (across C_2) is then updated according to (8) and (9). Note that the values of C_1 and C_2 are not critical. The circuit can be simplified further by using a single unity-gain buffer to recharge C_3 either to $V_r(i)$ or to $V_a(i)$.

IV. ACCURACY AND SPEED ANALYSIS

The accuracy of the data converters proposed in the previous section is limited by the offset voltages and gain errors of the unity-gain buffers, by capacitance mismatch, and by clock feedthrough due to the analog switches. All unity-gain buffers are used in S/H circuits in conjunction with analog switches, and thus the combined errors associated with unity-gain buffers and analog switches can be compensated for by using existing techniques developed for improved S/H circuits. Such a compensation scheme is shown in Fig. 4 [4], [10], [11], together with the timing diagram of the clock signals. The gain error δ and the offset voltage V_{os} of the buffer are compensated by storing an error voltage $V_{in}\delta - V_{os}$ across C_a at t_s and then adding it to V_{in} at t_3 . At this instant, most of the feedthrough charges Q injected from M_2 and M_3 into nodes (b) and (c) at t_2 and t_1 , respectively, are absorbed, because parasitic capacitances C_p and C_s are recharged to $V_{in} + V_a$ and V_{in} , respectively, where V_a is the voltage across C_a . The feedthrough charge Q injected from M_1 into C_h at t_5 is cancelled at t_h by transferring the feedthrough charge Q stored in C_c at t_4 into C_h . The output voltage at t_h is

$$[V_{out}]_{t=t_h} = \left(1 - \frac{\delta(\Delta_p + \delta)}{1 + \Delta_p}\right) V_{in} + (1 - \delta) \left(\frac{\delta + \Delta_p}{1 + \Delta_p} V_{os} + \frac{Q}{C_a} \cdot \frac{\Delta_s}{(1 + \Delta_s)(1 + \Delta_p)} \right) \quad (10)$$

where $\Delta_p = C_p/C_a$ and $\Delta_s = C_s/C_a$. In deriving (10), it was assumed that M_1 , M_2 , M_3 , and M_5 inject the same amount of charge Q into nodes (a), (b), (c), and (d), respectively, when

Fig. 4. An offset- and gain-compensated S/H circuit.

turned off. This assumption is reasonable because these switches are subjected to almost the same voltages. It is clear from (10) that the effects of δ and V_{os} are reduced. The effect of parasitic capacitances is also second-order small. For realistic values (say, $\delta \approx 0.01$, $V_{os} \approx 5$ mV, $Q/C_a \approx 5$ mV, $\Delta_p \approx 0.01$, $\Delta_s \approx 0.05$), the gain error can be reduced to a negligibly small value, and the residual offset voltage is 0.3 mV or less. Let the residual offset be V_{off} . Then, the recursive algorithm executed by the D/A converter shown in Fig. 2 is modified to

$$\begin{pmatrix} V_r(i) \\ V_a(i) \end{pmatrix} = \begin{pmatrix} b_i + \bar{b}_i/2 & \bar{b}_i/2 \\ b_i/2 & \bar{b}_i + b_i/2 \end{pmatrix} \begin{pmatrix} V_r(i-1) + \bar{b}_i V_{off} \\ V_a(i-1) + b_i V_{off} \end{pmatrix} \quad (11)$$

where $V_r(i)$ and $V_a(i)$ are referred to the input terminals of unity-gain buffers B_1 and B_2 , respectively. Simulation based on (11) shows that the error in $V_a(n)$ becomes maximum when $b_i \oplus b_{i+1} = 1$ ($i = 1, 2, \dots, n-1$), amounting to 1 mV when $V_{off} = 0.3$ mV and $n = 10$. Simulation, based on (7)–(9) including the offset terms, also shows that the error in $V_r(n)$ in Fig. 3 becomes maximum when $V_a = (1/3)V_r$ or $(2/3)V_r$, and amounts to 2 mV when $V_{off} = 0.3$ mV and $n = 10$. Therefore, the residual offset voltage limits the resolutions of the D/A and A/D converters to 10 and 9 bits, respectively, when $V_r = 2$ V.

Let the capacitance mismatch between C_1 and C_2 in Fig. 2 (including the top-plate parasitic capacitances) be ϵ . Because of circuit symmetry, it is reasonable to assume that the same mismatch exists between C_2 and C_3 . Then, the 2×2 matrix representation of the conversion algorithm becomes

$$\begin{pmatrix} V_r(i) \\ V_a(i) \end{pmatrix} = \begin{pmatrix} b_i + \frac{\bar{b}_i}{2+\epsilon} & \bar{b}_i \frac{1+\epsilon}{2+\epsilon} \\ \frac{b_i}{2+\epsilon} & \bar{b}_i + b_i \frac{1+\epsilon}{2+\epsilon} \end{pmatrix} \begin{pmatrix} V_r(i-1) \\ V_a(i-1) \end{pmatrix}. \quad (12)$$

Executing the algorithm n times, the n -bit D/A converter produces the analog voltage:

$$V_a' = \sum_{i=1}^n \frac{b_i}{(2+\epsilon)^i} \prod_{j=1}^{i-1} (1+b_j) V_r. \quad (13)$$

The integral nonlinearity is largest in the vicinity of the major carry, when all the bits except the MSB assume the value 1. For

Fig. 5. Experimentally observed reset signal (top trace), clock signal (upper middle trace), voltage across C_2 (lower middle trace), and analog voltage sequence (bottom trace) when the prototype switched-capacitor cyclic D/A converter converted the binary number $b = (11111111)_B$ into the analog voltage with a voltage reference $V_r = 2$ V. Vertical scales: 5 V/div (top trace), 10 V/div (upper middle trace) and 1 V/div (lower middle and bottom traces). Horizontal scale: 10 μ s/div.

the nonlinear error to be less than $(1/2)$ LSB, ϵ should satisfy the condition

$$\epsilon \leq 2^{-(n-1)}. \quad (14)$$

Because the conversion process is monotonic, the differential nonlinearity is always smaller than the integral nonlinearity in this scheme. Hence, the conversion generates no missing code and is accurate to within $(1/2)$ LSB if ϵ satisfies (14). In standard IC technologies, a matching accuracy better than 0.2 percent is feasible [13], [14]. It follows therefore that a conversion accuracy higher than 9 or 10 bits can be expected from the monolithic implementation of the cyclic D/A converter shown in Fig. 2. This also holds true for the successive-approximation A/D converter shown in Fig. 3.

The conversion rate of the proposed data converters is limited by the 0.1 percent settling time t_s of the unity-gain buffers, the propagation delay t_d of the digital circuitry, and the response time t_r of the comparators. According to SPICE simulations, $t_s \approx t_d \approx 40$ ns and $t_r \approx 20$ ns can be expected if the converter is fabricated using a 3- μ m CMOS process. Therefore, a conversion rate up to 10 Mb/s can be expected from such a CMOS monolithic implementation, and higher rates for finer linewidths. Because of the small device-count, cascading the 1-bit A/D converter shown in Fig. 3 for pipeline conversion and its monolithic implementation using fine-line CMOS, BiCMOS or bipolar process are possible. A conversion rate suitable for video signals could thus be attained.

Another important aspect in a monolithic implementation is power dissipation. In the SPICE simulation, a unity-gain buffer with $V_{DD} = -V_{SS} = 5$ V and a slew rate 250 V/ μ s with a 2-pF load capacitor was assumed. Since the 1-bit A/D converter stage includes three S/H circuits, the power dissipated in the 8-bit pipeline A/D converter is estimated to be about 120 mW, which is four times smaller than that in a conventional 8-bit CMOS flash converter.

V. EXPERIMENTAL RESULTS

Prototype cyclic D/A and successive-approximation A/D converters based on Figs. 2 and 3, respectively, were built using LF356 commercially available op-amps as buffers and MC14016 analog switches. 1-nF capacitors with matching errors less than 1 percent were used as unit capacitors.

Fig. 5 shows the experimentally observed waveforms in the circuit of Fig. 2 during the conversion of the 8-bit word $b =$

Fig. 6. Experimentally observed reset signal (top trace), clock signal (upper middle trace), binary output b (lower middle trace), and threshold voltage sequence (bottom trace) when the prototype switched-capacitor successive-approximation A/D converter converted the analog voltage 0.67 V into a 9-bit binary number with a voltage reference $V_r = 2$ V. Vertical scales: 10 V/div (top, upper and lower middle traces) and 0.5 V/div (bottom trace). Horizontal scale: 10 μ s/div.

$(11111111)_B$ into an analog voltage, with a voltage reference $V_r = 2$ V. Since all bits have the value 1, capacitor C_2 is charged to V_r in every $\phi = "1"$ phase (lower middle trace), while, starting with 0 V, the analog voltage sequence $V_a(i)$ (bottom trace) increases in each cycle by $V_r/2$, $V_r/4$, and so on, due to the charge redistribution between C_2 and C_3 in the $\phi = "1"$ phase. The final analog voltage is 2 V, which is the correct value for this digital input word. Fig. 6 shows the experimentally observed waveforms when the prototype successive approximation A/D converter based on the circuit of Fig. 3 converted the analog voltage $V_a = 0.67$ V into a 9-bit number, with a voltage reference $V_r = 2$ V. The bottom trace shows the threshold voltage sequence which increases or decreases depending on the value b_i , to approach the analog input voltage. The result $b = (010101011)_B$, indicated in the lower middle trace, is the correct digital equivalent of the analog input voltage.

VII. CONCLUSIONS

A novel algorithm for D/A conversion was described. It uses the MSB first and the LSB last. It is applicable to cyclic, successive approximation, sequential-flash, and pipeline conversion. An error analysis and experimental results were also presented to verify the usefulness of the new algorithm. The results indicate that a conversion accuracy higher than 9 bits and a conversion rate suitable for video applications can be expected from their pipeline realization using advanced IC technologies. The power dissipation is also estimated to be 4 times smaller than that of a conventional CMOS flash A/D converter.

ACKNOWLEDGMENT

The authors thank Y. Moriuchi and H. Matsumoto for useful discussions and experimental works.

REFERENCES

- [1] D. H. Sheingold, Ed., *Analog-Digital Conversion Handbook*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
- [2] B. Loriferne, *Analog-Digital and Digital-Analog Conversion*. London, England: Heydon, 1982.
- [3] D. J. Dooley, Ed., *Data Conversion Integrated Circuits*. New York: IEEE Press, 1980.
- [4] G. C. Temes, "High accuracy pipeline A/D converter configuration," *Electron. Lett.*, vol. 21, pp. 762-763, Aug. 1985.
- [5] K. Watanabe and H. Matsumoto, "Switched-capacitor algorithmic digital-to-analog and analog-to-digital converters," in *IEEE Proc. Int. Conf. on Circuits and Systems Proc.*, Kyoto, Japan, pp. 331-332, 1985.

- [6] H. Matsumoto and K. Watanabe, "Switched-capacitor algorithmic digital-to-analog converters," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 721-724, July 1986.
- [7] K. Watanabe, G. C. Temes, and Y. Moriuchi, "A new digital-to-analog conversion algorithm," in *Proc. IEEE Int. Symp. Circuits and Systems*, Espoo, kpp. 2817-2820, 1988.
- [8] G. C. Temes, F.-J. Wang, and K. Watanabe, "Novel pipeline data converters," in *Proc. IEEE Int. Symp. Circuits and Systems*, Espoo, pp. 1943-1946, 1988.
- [9] E. Suarez, P. R. Gray, and D. A. Hodges, "All-MOS charge redistribution analog-to-digital conversion techniques—Part II," *IEEE J. Solid-State Circuits*, vol. SC-10, pp. 379-385, Dec. 1975.
- [10] K. Watanabe and S. Ogawa, "Clock-feedthrough compensated sample/hold circuits," *Electron. Lett.*, vol. 24, pp. 1226-1228, Sept. 1988.
- [11] L. Kunsagi and G. C. Temes, "Buffer-based switched-capacitor gain stages," *Electron. Lett.*, vol. 24, pp. 254-255, Mar. 1988.
- [12] Y. Kuramochi, private communication.
- [13] J. B. Shyu, G. C. Temes, and F. Krummenacher, "Random error effects in matched MOS capacitors and current sources," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 948-955, Dec. 1984.
- [14] R. Gregorian and G. C. Temes, *MOS Analog Integrated Circuits for Signal Processing*. New York: Wiley, 1986, chap. 7.

Realization of 2-D State-Space Filters with Fewer Multipliers

PRADEEP MISRA AND ARNAB K. SHAW

Abstract—It is shown that a reduction in the number of multipliers for hardware realizations of two-dimensional (2-D) state-space digital filters can be achieved. This is enabled by transformation of the given system to a "block diagonal canonical structure."

Keywords—2-D filters, State-space representation, VLSI implementation.

I. INTRODUCTION

In recent years several authors have proposed different realizations of 2-D digital filters described by their transfer functions as well as state-space representations [1]–[6]. An account of relative merits and demerits of various realizations with respect to their modularity, regularity, parallelism, and the number of multiplier and/or adder requirement can be found in [6]. It is obvious that the fewer the number of adders and multipliers, the more economical it is to implement them using the VLSI technology.

In this note, we will consider 2-D filters in their state-space representation [7–8]:

$$\begin{bmatrix} x_1(\mu + \sigma_1) \\ x_2(\mu + \sigma_2) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(\mu) \\ x_2(\mu) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(\mu) \quad (1.1a)$$

$$y(\mu) = \begin{bmatrix} c_1^T & c_2^T \end{bmatrix} \begin{bmatrix} x_1(\mu) \\ x_2(\mu) \end{bmatrix} + du(\mu) \quad (1.1b)$$

where $x_1 \in \mathbb{R}^n$, $x_2 \in \mathbb{R}^m$; σ_1, σ_2 are the forward shifts and $(n + m)$ is the minimum number of delays required to implement the above filter, provided the system of (1.1) has minimal order [9].

In [6], it is shown that by transforming A_{11} and A_{22} to their canonical forms, the number of multipliers required to implement

Manuscript received November 29, 1988; revised April 18, 1989. This work was supported in part by WRDC/WPAFB under Grant F33615-88-C-3605. This paper was recommended by Associate Editor D. M. Goodman.

The authors are with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435.
IEEE Log Number 8931732.