

Research Article

Realization of Ternary Sigma-Delta Modulated Arithmetic Processing Modules

Amin Z. Sadik and Peter J. O'Shea

School of Engineering Systems, Queensland University of Technology (QUT), 2 George Street, Brisbane, QLD 4000, Australia

Correspondence should be addressed to Peter J. O'Shea, pj.oshea@qut.edu.au

Received 22 May 2008; Revised 5 December 2008; Accepted 11 February 2009

Recommended by Bernhard Wess

Sigma-delta modulated systems have a number of very appealing properties and are, therefore, heavily used in analog to digital converters, amplifiers, and modulators. This paper presents new results which indicate that they may also have significant potential for general purpose arithmetic processing. The paper introduces new arithmetic processing structures for ternary (i.e., +1, 0, or -1) sigma-delta modulated signals. Simulations show that these new structures can be implemented very efficiently and have relatively good accuracy.

Copyright © 2009 A. Z. Sadik and P. J. O'Shea. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

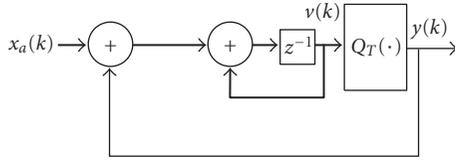
1. Introduction

Oversampled sigma-delta modulation (SDM) signal representations have several key advantages over traditional Nyquist rate pulse code modulated formats. When signals are put into SDM format, they typically have very short word-lengths (e.g., binary or ternary). This very simple representation creates the potential for reduced hardware complexity, simpler signal bus routing, and resilience against electronic component inaccuracies [1, 2]. There is yet another advantage to the use of SDM (or bitstream) systems. If signals are maintained in bitstream form all the way through the processing chain, one does not need format conversions (or the associated interpolation and decimation filters). This is so because within many systems, the "front end" analog to digital converters and "back end" digital to analog converters use SDM bitstream format, but the intermediate processing stages are typically implemented in multibit format. If the intermediate processing stages of a system can be operated in binary or ternary format as well, then the format conversions are unnecessary.

Several works have proposed digital bitstream arithmetic processing using pulse width modulation, particularly in digital neural networks, for example, [3, 4]. O'Leary and Maloberti [5] also presented a binary bitstream adder in which the sum is stored and fed back to the adder to reduce

the truncation error (i.e., the carries of the full-adder). This same approach has been adopted by [6] to implement a ternary bitstream adder using 2's complement format. However, the possibility of compensating the ignored carries is confined to the immediate next sample. The compensation fails when the next sample addition generates a carry as well. Another well-cited work on bitstream arithmetic is that conducted in [7], where various binary arithmetic circuits were proposed. However, most of the arithmetic circuits proposed in [7] suffer from two main drawbacks. First, many of these structures do not operate fully in the short word-length domain; they partially operate in the multibit domain. In particular, many of these structures use integrators which consist of a recursive multibit adder followed by an SDM requantizer. The second major drawback in the arithmetic units within [7] was the limited accuracy of the structures.

This paper attempts to address both of the above listed limitations. Ternary quantized SDM processing is the assumed format throughout the paper. Ternary format (i.e., +1, 0, -1) is used rather than binary because the extra zero state reduces quantisation error and so enables greater accuracy. At the same time, the zero state often corresponds to a no hardware operation, and so ternary formats often require minimal extra hardware components [8].



$Q_T(\cdot)$: Ternary quantizer
 \rightarrow 1-bit
 \rightarrow Multi-bit

FIGURE 1: The structure of a first-order ternary sigma-delta modulator.

In this paper, *ternary* arithmetic processing modules are proposed, and an attempt is made to provide a measure of the accuracy of these systems. This is done by determining the resolution or number of bits in a multibit counterpart with similar accuracy. Both DC and slowly varying input signals are considered in this paper.

2. Basic 1-Trit Ternary Adder/Subtractor

The adder is a critically important component of an arithmetic processor since it is a fundamental building block upon which many other processing operations are built. Therefore, it is highly desirable to create adders with minimal complexity. Let $x(k)$ and $y(k)$ be two ternary bitstreams. It is assumed that the inputs to the proposed adder are obtained with ternary sigma-delta modulators (TSDMs). A sample first-order TSDM is shown in Figure 1. These modulators may be well incorporated into analog to digital converter hardware. The signal $y(k)$ is

$$y(k) = \begin{cases} +1, & \text{for } v(k) \geq \frac{\lambda}{4}, \\ 0, & \text{for } -\frac{\lambda}{4} < v(k) < \frac{\lambda}{4}, \\ -1, & \text{for } v(k) \leq -\frac{\lambda}{4}, \end{cases} \quad (1)$$

where $[-\lambda/2, +\lambda/2]$ is the allowable range of the input. $x(k)$ is defined similarly. Assume that the desired output of the proposed ternary adder is $s(k) = x(k) + y(k)$, such that all $x(k)$, $y(k)$, and $s(k) \in \{1, 0, -1\}$. The task of designing the adder involves a tradeoff between implementation complexity and accuracy. One can implement a very simple adder with relatively large quantization by simply truncating any results greater than ± 1 . The resulting quantization error is then relatively large. This simple adder would be defined as

$$s(k) = \begin{cases} \frac{(x(k) + y(k))}{2}, & \text{for } x(k) = y(k), \\ x(k) + y(k), & \text{for } x(k) \neq y(k). \end{cases} \quad (2)$$

The above “basic” ternary adder can be implemented using a traditional ternary half-adder. It is perfectly accurate except when the two input signal values are identical, in which case a carry is generated (and neglected).

The quantization noise at the output of the basic ternary adder has two components: (i) the quantization noise inherent in the two input signals and (ii) the quantization noise due to the truncation operation which occurs when the two inputs have identical values. These two components are quantified below.

- (i) If there was a white uniformly distributed quantization noise in each of the inputs, then the power spectral density (PSD) of each input would be

$$P_n = \frac{\Delta^2}{12f_s}, \quad (3)$$

where f_s is the sampling frequency, and Δ is the quantization step. The quantization noise will not be white, however, because of the noise shaping inherent in SDM signal formats [9, 10]. Assume, say, that 1st-order TSDMs have been used to create $x(k)$ and $y(k)$, then the power spectral density of the quantization noise in both $x(k)$ and $y(k)$ would be

$$\begin{aligned} P_q(f) &= P_n \left| 2 \sin \left(\pi \frac{f}{f_s} \right) \right|^2 \\ &= \frac{1}{3f_s} \left| \sin \left(\pi \frac{f}{f_s} \right) \right|^2. \end{aligned} \quad (4)$$

It is worth noting that we are dealing here with the whole oversampled spectrum (from 0 to f_s Hz) since the operations are achieved within the ternary domain and there is no decimation process.

- (ii) Now, it is necessary to determine an expression for the quantization noise corresponding to the truncation introduced by the ternary adder. We denote the truncation error signal as $p(k)$. This signal will only be nonzero when $x(k) = y(k) = \pm 1$. This condition would be expected to occur on average about two samples out of every nine, assuming that the probability of obtaining a value of +1, -1, or 0 is equal to 1/3. The total average power of $p(k)$ would, therefore, be about 2/9. The spectral shape of $p(k)$ depends on the correlation between $x(k)$ and $y(k)$. If $x(k)$ and $y(k)$ are perfectly correlated, then the spectrum has a delta function at DC. If the signals are uncorrelated, then the spectrum tends to be white.

Then, the expected PSD due to the truncation process is

$$E[P_{\text{trun}}(f)] = \bar{P}_t(f) = E\{|F\{p(k)\}|^2\}, \quad (5)$$

where $E[\cdot]$ is the expectation operator, and F is the Fourier transform operator. The total power spectral density at the output of the adder will be the sum of the quantization noise corresponding to the two input signals ($2P_q(f)$) and $P_{\text{trun}}(f)$. That is, the total quantization power spectrum is given by

$$P_t(f) = \frac{2}{3f_s} \left| \sin \left(\pi \frac{f}{f_s} \right) \right|^2 + E\{|F\{p(k)\}|^2\}. \quad (6)$$

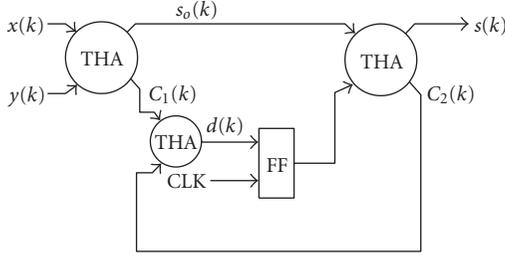


FIGURE 2: An improved version of the proposed adder (TA) constructed using three ternary half-adders (THAs).

The subtraction operation can be easily accomplished by negating one of the ternary bitstreams and using the same proposed adder.

3. Improved Adder

Knowing the source of errors in the ternary adder specified in (2), one may alleviate this error using a simple technique. If the lost carries are compensated for whenever possible in the next samples, then in the average sense, the adder would have improved accuracy. This can be done by introducing a ternary flip-flop $d(k)$ in the adder circuit to store any carry overflows and propagate this carry information to subsequent samples. Figure 2 shows a block diagram of the improved ternary adder version. The rationale behind the new adder circuit is that any carry arising from the addition of the current two input samples should be stored in a flip-flop and added to the next output sample. If there is any carry generated from doing this addition to the next output sample, then that resulting carry should also be fed back and stored. The operation of the circuit in Figure 2 can be described mathematically as

$$\begin{aligned} s_o(k) &= \text{sgn}[x(k) + y(k)], \\ s(k) &= \text{sgn}[s_o(k) + d(k-1)], \\ c_1(k) &= x(k) + y(k) - s_o(k), \\ c_2(k) &= s_o(k) + d(k-1) - s(k), \\ d(k) &= \text{sgn}[c_1(k) + c_2(k)]. \end{aligned} \quad (7)$$

The improved adder can be implemented by using three ternary half-adder (THA) modules and one delay element. Each THA performs according to the truth-table shown in Table 1. The ternary adder (TA) defined in (7) can easily be implemented with either conventional digital gates (e.g., [11]) or with multiple-valued logic (e.g., [12, 13]).

According to (7), $d(k)$ can be re-expressed as

$$d(k) = \begin{cases} \frac{1}{2}(c_1(k) + c_2(k)), & \text{for } c_1(k) = c_2(k) \neq 0, \\ c_1(k) + c_2(k), & \text{otherwise,} \end{cases} \quad (8)$$

noting that the condition $c_1(k) = c_2(k) \neq 0$ corresponds to the event of $x(k) = y(k) = d(k-1) \neq 0$.

By recombining lines 3 and 4 of (7), one obtains the following expression for $s(k)$:

$$s(k) = x(k) + y(k) + \underbrace{d(k-1) - [c_1(k) + c_2(k)]}_{\text{error term, } e_c(k)}, \quad (9)$$

Recalling that $d(k) = \text{sgn}[c_1(k) + c_2(k)]$. It should be noted that the error term is only nonzero when both $c_1(k)$ and $c_2(k)$ are equal and nonzero (or equivalently, $x(k) = y(k) = d(k-1) \neq 0$). Implicitly, this leads to the fact that an error will only occur when $x(k) = y(k) = x(k-1) = y(k-1) \neq 0$ (i.e., with two consecutive +1s (-1s)). Then, the error $e_c(k)$ is given as

$$e_c(k) = \begin{cases} \pm 1, & \text{for } x(k) = y(k) = x(k-1) = y(k-1) \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Note that if the probability of +1, -1, and 0 is assumed equal for a trit, then the probability of $x(k) = y(k) = x(k-1) = y(k-1) = +1$ is $1/3^4$ and the probability of $x(k) = y(k) = x(k-1) = y(k-1) = -1$ is also $1/3^4$. Therefore, the total probability that $x(k) = y(k) = x(k-1) = y(k-1) = \pm 1$ would be $2/81$.

Now, assuming ergodicity, the average value for $e_c(k)$ can be calculated as

$$c(0) = \frac{1}{N} \sum_{k=0}^{N-1} e_c^2(k), \quad (11)$$

and the autocorrelation function for $e_c(k)$ is given by

$$c(m) = \frac{1}{N} \sum_{k=0}^{N-1} e_c(k) e_c(k-m). \quad (12)$$

The precise form for the autocorrelation function will depend on the nature of the signals and in particular the correlation between them. By taking the Fourier transform of (12), one can determine the power spectral density $P_c(f)$ of $e_c(k)$ as

$$P_c(f) = \mathcal{F}\{c(m)\} = \sum_{m=0}^{\infty} c(m) \cos(2\pi m f / f_s). \quad (13)$$

Note that in the above equation the discrete Fourier transform uses cosine basis functions rather than complex exponential ones because c_m is an autocorrelation function and is, therefore, even.

An alternative approach can be used by recombining lines 3, 4, and 5 of (7), then one obtains the following expression for $d(k)$:

$$\begin{aligned} d(k) &= \text{sgn}[x(k) + y(k) + d(k-1) - s(k)] \\ &= x(k) + y(k) + d(k-1) - s(k) + q(k), \end{aligned} \quad (14)$$

where $q(k)$ is the truncation error in $d(k)$ due to “uncompensatable” carries. $q(k)$ is given by $q(k) = \text{sgn}[x(k) + y(k) + d(k-1) - s(k)] - [x(k) + y(k) + d(k-1) - s(k)]$.

TABLE 1: Truth-table of the proposed THA.

x	y	c	s
0	0	0	0
1	0	0	1
0	1	0	1
-1	0	0	-1
0	-1	0	-1
1	1	1	1
-1	1	0	0
1	-1	0	0
-1	-1	-1	-1

Rearrangement the above expression for $d(k)$ yields an expression for $s(k)$ as follows:

$$s(k) = x(k) + y(k) + d(k-1) - d(k) + q(k). \quad (15)$$

Taking the z -transform of (15), the output can be expressed as

$$S(z) = X(z) + Y(z) - (1 - z^{-1})D(z) + Q(z). \quad (16)$$

Examination of the above equation reveals that $S(z)$ comprises a true component ($X(z) + Y(z)$) and an error term ($-(1 - z^{-1})D(z) + Q(z)$). The error term is, in turn, comprised of two components, the first corresponding to carries which are eventually “compensated” ($-(1 - z^{-1})D(z)$) and the second due to uncompensated or “lost” carries ($Q(z)$). The structure of the adder causes the compensated carry error component to be high-pass filtered, as per the $(1 - z^{-1})D(z)$ term. This high-pass filtering causes significant attenuation of the error term and accounts for the improvement provided by the adder. The uncompensated carries error term does not get attenuated, but fortunately, it is relatively low in power because it tends to be nonzero in an average probability of about 2/81.

Because of the high-pass filtering of the error term which is inherent in this adder, a significant reduction in the average quantization error can be achieved. This is illustrated in simulations in Section 5.

4. Format Conversion via an SDM with Ternary Integrator

As discussed earlier, ternary arithmetic is significantly more accurate than binary arithmetic, at least for pulse width modulation type signal formats. To implement practical ternary arithmetic, it may sometimes be necessary to convert incoming signals from binary format to ternary format. It is obviously desirable that this must be done efficiently. This section proposes an efficient new structure for binary to ternary format conversion, with this new structure involving an SDM whose internal integrator is formed from the adder proposed in the previous section.

In the short word-length literature, digital integrators have generally been constructed from a multibit subsystem (such as up-down counters) followed by a 1-bit noise shaper

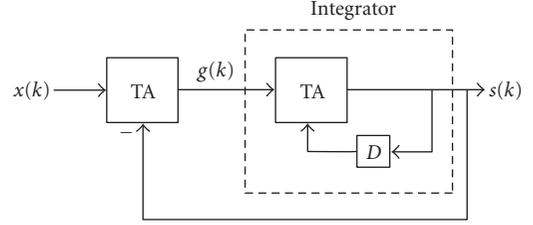


FIGURE 3: Realization of a first-order SDM using the proposed adder (TA) and the integrator.

to restore the format to the short word-length domain [7]. This approach is computationally intensive. By using the ternary adder (TA) from the previous section, a novel integrator is proposed that operates entirely in the ternary domain (see Figure 3, inside the box)

Simulation results presented in Section 5 show that the new 1-bit ternary integrator outperforms the traditional counterpart in [7].

Having devised a digital integrator the next step is to construct an SDM-based format conversion structure. This structure uses both the proposed adder and new integrator and is shown in Figure 3.

Consider first the leftmost TA in Figure 3. Equation (16) provides a general expression for the output of an arbitrary TA, and using this result, one can obtain the output for the leftmost TA as follows:

$$G(z) = X(z) - z^{-1}S(z) - (1 - z^{-1})D_1(z) + Q_1(z), \quad (17)$$

where $(1 - z^{-1})D_1(z)$ and $Q_1(z)$ are the errors due to the *compensated* and *uncompensated* carries, respectively, in the leftmost TA. (Note that because of the synchronous clocking which is used, there is effectively a single sample delay in the feedback path of the SDM in Figure 3. This effective delay is not explicitly shown in Figure 3 because of convention—SDMs representations normally do not explicitly show a delay).

Now consider the rightmost TA. Again, using the result in (16), one can obtain the Z -transform of the output as

$$S(z) = G(z) + z^{-1}S(z) - (1 - z^{-1})D_2(z) + Q_2(z), \quad (18)$$

where $(1 - z^{-1})D_2(z)$ is the error due to compensated carries, and $Q_2(z)$ is the error due to uncompensated carries in the rightmost TA. Combining the above two equations yields

$$S(z) = X(z) - (1 - z^{-1})[D_1(z) + D_2(z)] + Q_1(z) + Q_2(z). \quad (19)$$

The performance of this new format conversion structure is evaluated in the next section.

5. Simulations

As we are dealing with arithmetic processing, there is a need to determine the resolution of the new ternary bitstream structures. To make a reasonable comparison with the

TABLE 2: Comparison among bitstream adders.

Technique	mse, ($\times 10^{-5}$)	No. of bits
used in [6]	4.9	6.4
used in [7]	45.2	4.8
proposed	0.72	7.75

multibit domain, the output stream has to be windowed and averaged to determine an equivalent multibit value. The length of the time window should be greater than or equal to the oversampling ratio (OSR) to ensure a fair comparison. The SNR of this averaged output ternary bitstream is then calculated, and an equivalent resolution (equivalent number of bits) can be obtained.

5.1. Simulation Results for the Proposed Adders. Four different input signals were considered in this work. The first two were sinusoids corrupted by additive white Gaussian noise with an SNR of about 25 dB. These sinusoids had the forms $x_a(k) = 0.5 \sin(2\pi k) + \nu_1(k)$ and $y_a(k) = 0.5 \sin(4\pi k) + \nu_2(k)$, respectively, with $\nu_1(k)$ and $\nu_2(k)$ representing the additive noise. These two sinusoids were mapped to the symmetrical ternary domain using ternary quantizer sigma-delta modulators. Figure 4 shows the spectra of the bitstreams $x_a(k)$, $y_a(k)$, and the summation of these bitstreams with the adder is specified in (7). Additional simulations have revealed that the adder is quite robust to the presence of DC components. That is, spurious tones do not appear when there is a DC component present.

The final two types of input signals considered were the DC signal and the ramp signal. The former is considered to be one of the most challenging signals for SDMs to deal with—it can easily produce limit cycles [14, 15]. It was specified by $x_a = -0.5$. The ramp signal was specified by $y_a(k) = 2^{-10}k$, where $k = 0, 1, 2, \dots, 2^{10}$. Figure 5 shows plots of the sum of $x_a(k)$ and $y_a(k)$ obtained with (i) the basic adder defined in (2), (ii) the improved adder defined in (7), and (iii) a 32 bit precision multibit adder. As seen in Figure 5, the curves corresponding to the 32 bit precision adder and the improved ternary adder are almost indistinguishable. The average output signal error power was calculated by subtracting the true signal value from a multibit reconstruction of the ternary signal representation. This reconstruction was achieved by filtering the ternary signal with an L -point moving average filter. In these simulations, $L = \text{OSR} = 128$. The mean squared error of the basic adder (method 1) and the improved adder (method 2) were 8.3×10^{-4} and 7.2×10^{-6} , respectively. This corresponds to an equivalent multibit resolution of 4.4 and 7.75 bits, respectively. An improvement of 3.35 bits has, therefore, been achieved by using the improved adder rather than the basic adder. This result is consistent with the expectation expressed in Section 3.

To compare the resolutions achieved with the proposed adder versus those of the adding technique presented in [7] (1-bit adder), [6] (ternary), the same DC and ramp inputs

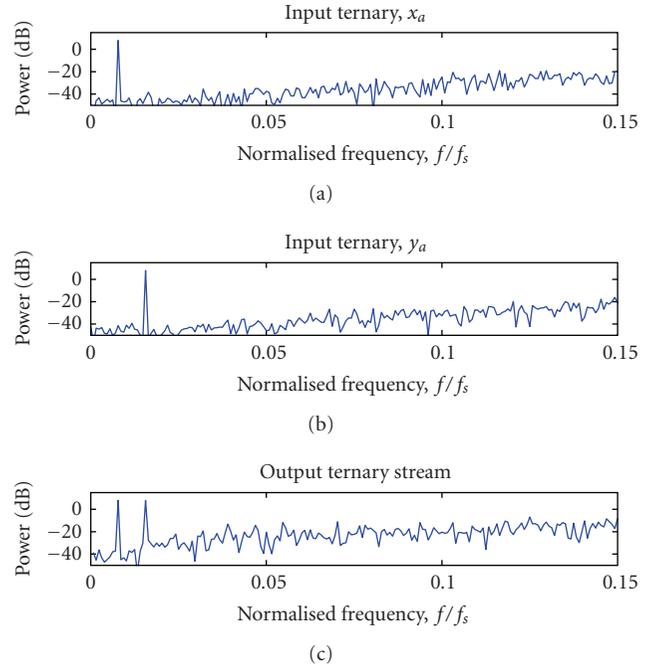


FIGURE 4: The spectra of the input ternary streams and their ternary summation.

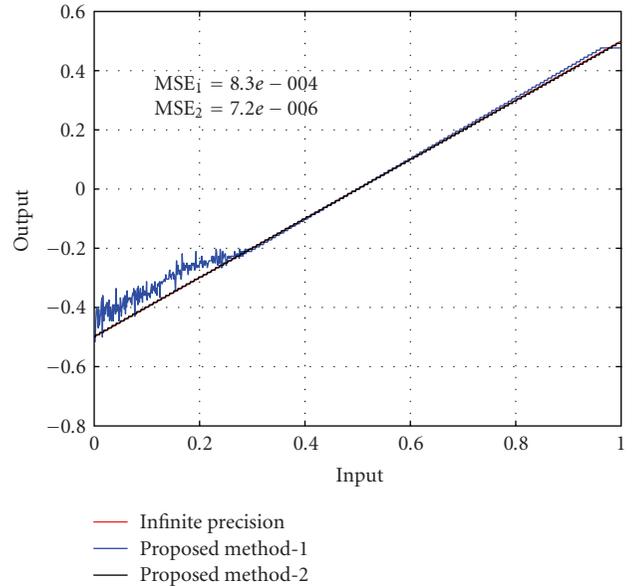


FIGURE 5: A comparison between the proposed ternary adders with DC ramp input, (red) infinite-precision, (blue) basic (method 1), and (black) improved (method 2).

as above were used. The adder output was averaged over $\text{OSR} = 128$ samples, and the mean squared error (MSE) was calculated and compared with an equivalent N -bit quantizer that produced the same value of MSE (for the same dynamic range $(-0.5-0.5)$). Table 2 summarizes the outcomes. The improved adder proposed in this paper clearly outperforms the existing adders.

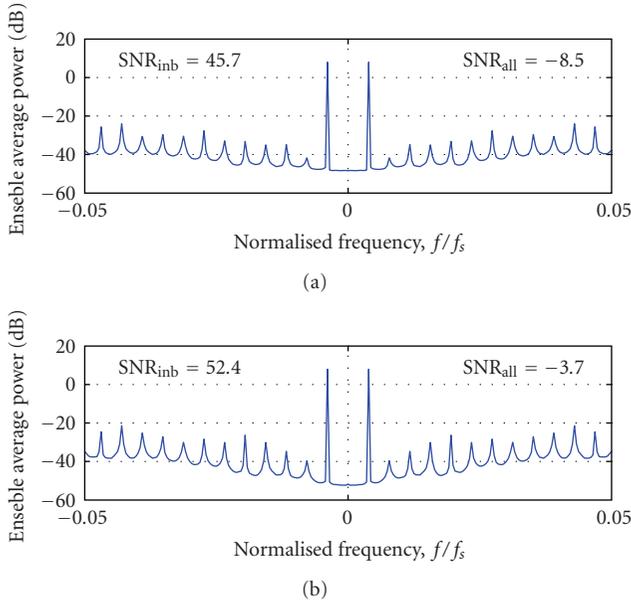


FIGURE 6: Output of the proposed format conversion structure in Figure 3 and output of its traditional counterpart. The oversampling ratio is always 128.

5.2. The Proposed Binary to Ternary Format Conversion Structure. It should be noted that as long as we are dealing with short word-length systems (i.e., with no need to go back and forth between decimation/interpolation stages), one must be concerned about the whole range of the frequency spectrum, that is, $[0, f_s]$.

For the simulations in this section, a 16-bit PCM signal $x_a(k) = 0.5 \sin(2\pi f_o k) + v_1(k)$ with an SNR of 45 dB was modulated using an SDM to produce $x(k) \in \{1, -1\}$. For simulation, this binary bitstream was used as input to the newly proposed SDM format conversion structure (shown in Figure 3). The power spectrum of the output is shown in Figure 6 with the normalized in-band region assumed to be $\pm(f_o/f_s) = \pm 0.0078$. Also in Figure 6 is shown the output obtained if the same format conversion structure and inputs are used, but with a traditional integrator (of the form proposed in [7]) instead of the integrator proposed in Section 3.

For the traditional and proposed format conversion structures, the ensemble-average (1000 runs) of the in-band SNRs (SNR_{inb}) was found to be +45.7 dB and +52.2, respectively. While the whole of band SNR (SNR_{all}) was -3.7 dB for the new integrator and -8.5 dB for the traditional one. That is, improvements of about 6.5 dB in SNR_{inb} and about 5 dB were obtained by using the new integrator within the format converter structure. This improvement is a promising finding as integrators are common structures in many digital electronic circuits. Moreover, the proposed format converter structure not only outperforms its traditional counterpart but also permits more efficient hardware implementation.

5.3. Realization of Exponential/Trigonometric Functions. This section illustrates the use of the improved adder as a building

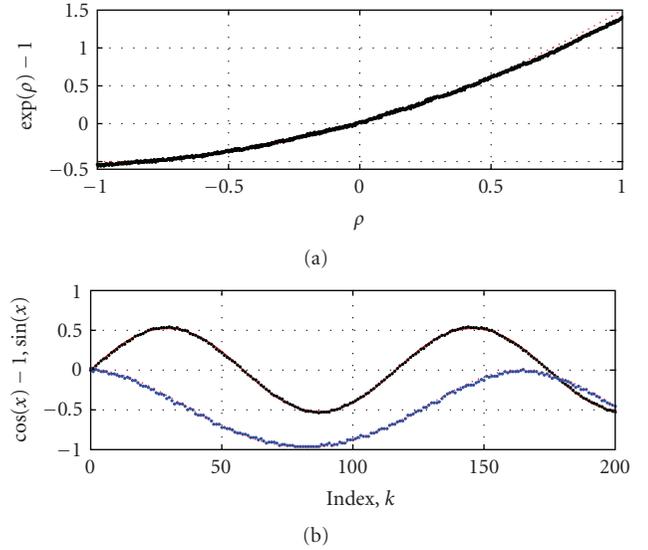


FIGURE 7: Realization of series expansion ternary bitstream (dotted) functions compared with their 32 bit precision counterparts (red): (a) Exponential function; (b) sine and cosine functions.

block for realising practically important functions such as exponential and trigonometric functions. To create these functions, the improved adder was first used to create a multiplier according to the model in [7]. That is, the multiplier in [7] was realized by simply replacing the original adder components with the new adder introduced in Section 3. Once the multiplier was constructed, the exponential and trigonometric functions were able to be created by using two non-DC terms of their series expansions (i.e., $e^x - 1 = x + x^2/2!$ and $\cos(x) - 1 = x^2/2! - x^4/4!$).

Figure 7(a) shows the averaged (with a 128 sample moving average filter) ternary realization of the function $e^x - 1$ for a ramp DC input (extending between -1 and $+1$) compared with its 2-term infinite-precision counterpart. The input was varied in steps of 2×2^{-10} . A mean squared error of 6×10^{-4} was obtained for a dynamic range of 2.0234 which is equivalent to the quantization noise of ~ 5.6 bit system.

Figure 7(b) shows averaged ternary sine and cosine functions which were created using the same approach as was used for the exponential function. These functions are drawn versus the index k of the input signal $x(k) = A \sin[8\pi\epsilon k]$ so as to provide a clear visual assessment of the outcome. The input signal $x(k)$ was varied in steps of $\epsilon = 2 \times 2^{-9}$. For a dynamic range of 1, the mses were 3.29×10^{-5} for the sine and 8.4×10^{-5} for the cos, with these being equivalent to 6.76 and 5.7 resolutions bits, respectively. The reduction in accuracy in the cosine function is attributed to the quadruple term implementation in its expansion series.

6. Conclusions

Novel 1-bit ternary arithmetic structures have been proposed in this paper for adders, integrators, and format converters. The internal processing and the output for these new

structures are all kept entirely in the ternary domain. The operation of the proposed adder is assessed in terms of the accuracy (expressed as the equivalent number of bits in corresponding multibit system). Simulations show that both structures are surprisingly efficient and, therefore, have the potential to realize multiplication, division, and exponential/trigonometric functions.

References

- [1] P. M. Aziz, H. V. Sorensen, and J. van der Spiegel, "An overview of sigma-delta converters," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 61–84, 1996.
- [2] B. Steele and P. J. O'Shea, "Performance estimation for lowpass ternary filters," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 12, pp. 1250–1256, 2003.
- [3] J. E. Tomberg and K. K. K. Kaski, "Pulse-density modulation technique in VLSI implementations of neural network algorithms," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 5, pp. 1277–1286, 1990.
- [4] A. F. Murray and A. V. W. Smith, "Asynchronous VLSI neural networks using pulse-stream arithmetic," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 3, pp. 688–697, 1988.
- [5] P. O'Leary and F. Maloberti, "Bit stream adder for oversampling coded data," *Electronics Letters*, vol. 26, no. 20, pp. 1708–1709, 1990.
- [6] C.-W. Ng, N. Wong, and T.-S. Ng, "Bit-stream adders and multipliers for tri-level sigma—delta modulators," *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 12, pp. 1082–1086, 2007.
- [7] H. Fujisaka, R. Kurata, M. Sakamoto, and M. Morisue, "Bit-stream signal processing and its application to communication systems," *IEE Proceedings: Circuits, Devices and Systems*, vol. 149, no. 3, pp. 159–166, 2002.
- [8] B. Hayes, "Third base," *American Scientist*, vol. 89, no. 6, pp. 490–494, 2001.
- [9] R. M. Gray, "Over sampled sigma-delta modulation," *IEEE Transactions on Communications*, vol. 35, no. 5, pp. 481–489, 1987.
- [10] R. M. Gray, "Quantization noise spectra," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1220–1244, 1990.
- [11] H. T. Koanantakool, "Implementation of ternary identity cells using c.m.o.s. integrated circuits," *Electronics Letters*, vol. 14, no. 15, pp. 462–464, 1978.
- [12] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron Devices*, vol. 39, no. 6, pp. 1444–1455, 1992.
- [13] H. Gundersen and Y. Berg, "A balanced ternary multiplication circuit using recharged semi-floating gate devices," in *Proceedings of the 24th Norchip Conference*, pp. 205–208, Linköping, Sweden, November 2006.
- [14] D. Reefman, J. Reiss, E. Janssen, and M. Sandler, "Description of limit cycles in sigma-delta modulators," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 6, pp. 1211–1223, 2005.
- [15] A. Z. Sadik, Z. M. Hussain, X. Yu, and P. J. O'Shea, "An approach for stability analysis of a single-bit high-order digital sigma-delta modulator," *Digital Signal Processing*, vol. 17, no. 6, pp. 1040–1054, 2007.