J. Low Power Electron. Appl. **2015**, *5*, 234-256; doi:10.3390/jlpea5040234

OPEN ACCESS

Journal of
Low Power Electronics and Applications
ISSN 2079-9268
www.mdpi.com/journal/jlpea

Article

Hardware-Efficient Delta Sigma-Based Digital Signal Processing Circuits for the Internet-of-Things

Yifei Liu, Paul M. Furth and Wei Tang *

Klipsch School of Electrical and Computer Engineering, New Mexico State University,
1125 Frenger Mall, Las Cruces, NM 88003, USA; E-Mails: yifeiliu@nmsu.edu (Y.L.);
pfurth@ad.nmsu.edu (P.M.F.)

* Author to whom correspondence should be addressed; E-Mail: wtang@nmsu.edu;
Tel.: +1-575-646-5768.

Academic Editor: Alexander Fish

Received: 22 July 2015 / Accepted: 26 October 2015 / Published: 9 November 2015

Abstract: This paper presents hardware-efficient Delta Sigma linear processing circuits for the next generation low-power VLSI devices in the Internet-of-things (IoT). We first propose the *P-N* (positive-negative) pair method to manipulate both the analog value and length of a first-order Delta Sigma bit sequence. We then present a binary counter method. Based on these methods, we develop Delta Sigma domain on-the-fly digital signal-processing circuits: the Delta Sigma sum adder, average adder and coefficient multiplier. The counter-based average adder can work with both first-order and higher-order Delta Sigma modulators and can also be used as a coefficient multiplier. The functionalities of the proposed circuits are verified by MATLAB simulation and FPGA implementation. We also compare the area and power between the proposed Delta Sigma adders and a conventional multi-bit adder by synthesizing both circuits in the IBM 0.18-µm technology. Synthesis results show that the proposed Delta Sigma processing circuits can extensively reduce circuit area and power. With 100 inputs, a Delta Sigma average adder saves 94% of the silicon area and 96% of the power compared to a multi-bit binary adder. The proposed circuits have the potential to be widely used in future IoT circuits.

Keywords: VLSI; Delta Sigma modulation; digital signal processing; adder; coefficient multiplier; low-power low-complexity circuits

1. Introduction

The Internet-of-Things (IoT) demands ubiquitous devices that collect and process sensed data and transmit back to the central node through various network interfaces [1]. Since radio devices usually dominate the power consumption of the system [2], in biomedical or physiological monitoring, the sensor data at each node should be preprocessed before transmission [3]. In particular, biomedical sensors in the IoT targets an important application of the "unmanned medical nursing system" [1], which could alleviate increasing medical costs as the baby boomers enter retirement [4]. Such wearable and implantable biomedical sensors demand low-complexity, low-power, high-resolution and high-reliability digital signal processing (DSP) circuits, which present the need to enhance the currently-applied solution. In a typical biomedical sensor, the recorded analog signal is first converted into digital format and then processed by an application-specific integrated circuit (ASIC) processor running a pre-implemented DSP algorithm, as shown in Figure 1a. If the algorithm determines that the incoming signal is abnormal, then the ASIC sends an alert signal and calls for help through wireless portable devices via the IoT. In such a system, the major challenges are the relatively large circuit area and high power of the DSP circuits performing multiply-and-accumulate (MAC) operations [5], which are adders and coefficient multipliers in digital filters. In order to miniaturize the physical size and extend the battery life of the circuits, low-complexity low-power designs are very important. As an example, in [5], an eight-channel 10-bit resolution DSP seizure detector processor consumes more than 50% of the chip area and power in a 5 mm × 5 mm system-on-a-chip (SOC) implemented in a 65-nm process, even though a look-up table (LUT) method is applied, which saved more than 50% of the total gate count in the digital filters [6]. Besides circuit complexity and power consumption, such circuits demand higher resolution and reliability, since higher resolution improves accuracy, and reliability is the key issue in biomedical applications. However, if a higher resolution circuit is implemented, then the power, chip area and cost of the DSP circuits also increase.

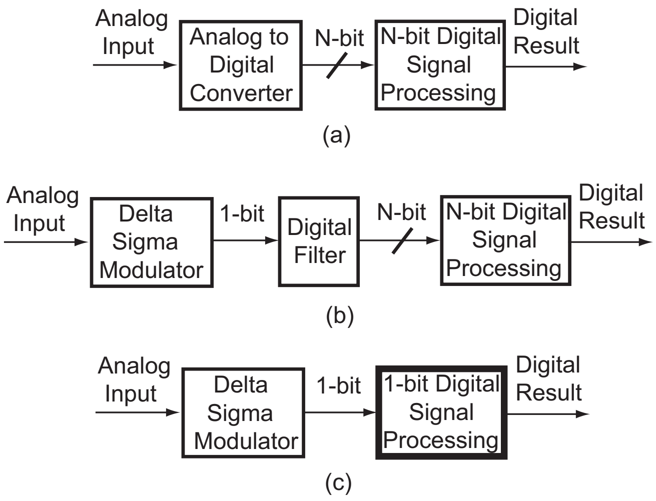


Figure 1. (**a**) Traditional signal processing architecture; (**b**) traditional signal processing after a Delta Sigma ADC; (**c**) the proposed signal processing using a Delta Sigma digital signal processing (DSP).

Delta Sigma technology can be an attractive solution for the aforementioned challenges and has been widely applied in biomedical circuits and systems [7–12] because of its unique properties. First, it is well known that a Delta Sigma analog-to-digital converter (ADC) has much less circuit complexity and lower power consumption than multi-bit ADCs. Second, in the Delta Sigma domain, the instantaneous analog value is represented by the moving average of the Delta Sigma bit-stream. Therefore, a few bit errors in the bit-stream do not disturb the signal level greatly [13]. Therefore, Delta Sigma systems have a higher tolerance to bit errors than multi-bit binary systems, which means Delta Sigma systems have higher reliability. Third, an oversampling Delta Sigma ADC can achieve a higher resolution (up to 24 bits) compared to a Nyquist rate multi-bit ADC (usually less than 18-bits). The Delta Sigma ADCs also relax the requirements on the analog anti-aliasing filter. Moreover, the resolution of a Delta Sigma system can be flexibly changed by adjusting its clock rate, without modifying the hardware. On the other hand, concerns for using a Delta Sigma system usually arise from its higher sampling rate and larger data size. However, in biomedical applications, the input signal usually has a bandwidth less than 2 kHz. Furthermore, the storage of raw data is usually neither necessary nor easily implementable. Thus, in such applications, using Delta Sigma bit-streams to represent and process information is both applicable and desirable.

Although Delta Sigma technology has been widely used in converting analog signals to digital signals [7,9,11], it has not been commonly applied in DSP, because of the lack of Delta Sigma-based DSP circuits. Currently, the modulated Delta Sigma bit-stream is demodulated into a multi-bit stream and then processed by conventional multi-bit DSP circuits, as shown in Figure 1b [14]. However, hardware complexity, power and reliability issues limit the performance of the multi-bit DSP system. Here, we study the alternative method using a one-bit DSP system to process Delta Sigma bit-streams directly, as shown in Figure 1c.

Currently, Delta Sigma-based DSP circuits are not sufficiently studied. In DSP systems, the most important linear circuits are sum adders, average adders and coefficient multipliers. To build those circuits with low noise and complexity, pure digital implementations are preferable. However, some previous Delta Sigma processing methods focused on modifying the Delta Sigma modulators [15–17] instead of processing the modulated bit-stream. Moreover, some solutions are hindered by involving analog operations [18–20] or tri-state digital circuits [21], which increase the complexity of the circuits. Other designs are impeded by their ability to process only first-order Delta Sigma bit-streams [22–33], which do not fit the widely-used higher-order Delta Sigma modulators. Furthermore, the adders in [22] can only take two inputs. One method of processing higher order Delta Sigma bit-streams is the digital Delta Sigma modulator proposed. However, a digital Delta Sigma modulator designed with a complicated sorting network [13] requires enormous circuit resources, which is not applicable for low-power devices.

In this paper, we propose novel methods and circuits for Delta Sigma-based DSP systems. Based on the digital Delta Sigma modulator method and the proposed *P-N* (positive-negative) pair method with a signed binary counter, we design the Delta Sigma sum adder, average adder and coefficient multiplier. We also present simulation and measurement results of the proposed circuits and provide a comparison of the synthesis results between Delta Sigma adders and multi-bit binary adders. This paper is organized as follows: Section 2 presents the properties of a first-order Delta Sigma modulated bit-stream and the basic processing techniques using the proposed *P-N* pair method. In Section 3, we describe the design of the Delta Sigma sum adder and the average adder. Section 4 provides simulation and analysis of the Delta Sigma adders. Section 5 describes a Delta Sigma compressor design. Section 6 presents the implementation and measurement results of the Delta Sigma average adder. Section 7 compares the proposed Delta Sigma adders to conventional multi-bit adders. Section 8 concludes the paper.

2. Delta Sigma Bit-Stream and Basic Signal Processing Techniques

In this section, we first briefly explain the properties of a first-order Delta Sigma-modulated bit-stream. Based on these properties, we propose the *P-N* pair method to process the Delta Sigma-modulated bit-streams.

2.1. Properties of First-Order Delta Sigma-Modulated Bit-Streams

A Delta Sigma modulator converts an analog input signal into a Delta Sigma-modulated digital bit-stream. A first-order Delta Sigma modulator consists of an analog integrator, a one-bit ADC, and a one-bit digital-to-analog converter (DAC), as shown in Figure 2. In the Delta Sigma modulator circuit, the analog input port and the digital output port can use either dual power supplies or a single power supply. To avoid confusion, we use *P* and *N* to represent the polarity of the output digital signals. *P* means a digital positive value of "true", while *N* means a digital negative value of "false". We use numbers to represent digital or analog levels. For example, -1 and $+1$ represent the dual power supply level *VSS* and *VDD*, respectively. Zero means analog ground, which is zero volts in a dual power system.

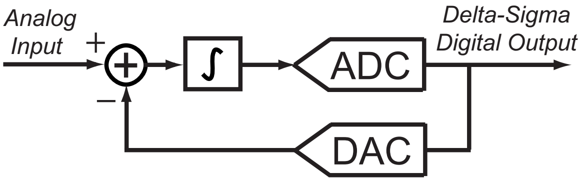


Figure 2. Schematic diagram of a first-order Delta Sigma modulator.

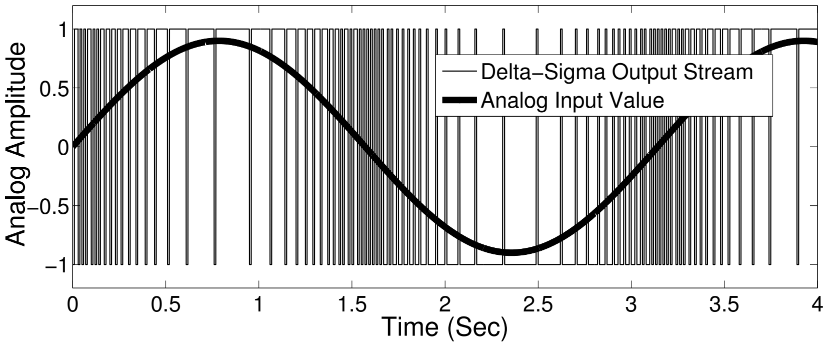


Figure 3. Waveforms of the input analog signal and the output Delta Sigma-modulated bit-stream of a first-order Delta Sigma modulator.

Figure 3 shows the input analog waveform and the output digital waveform of a first-order Delta Sigma modulator. When the input is positive, the digital output bit-stream contains more *P* bits than *N* bits. There can be consecutive *P* bits, but each *N* bit is separated by at least one *P* bit. Thus, there are no adjacent *N* bits when the input is positive. Similarly, when the input is negative, the output bit-stream consists of single *P* bits separated by one or more *N* bits. A Delta Sigma bit-stream toggles equally between *N* bits and *P* bits when its analog value is zero. Delta Sigma modulation is a type of pulse density modulation (PDM), which means the output digital bit-stream can be converted back to an analog value using a digital moving-average filter. Therefore, suppose Delta Sigma-modulated bit sequence a_i has a total of *W* bits, if *n* of which have the value *N* and *p* of which have the value *P*. Then the analog value *y* of bit sequence a_i can be written as:

$$y = \frac{1}{W}\sum_{i=0}^{W-1} a_i = \frac{-n+p}{W} \tag{1}$$

where $W = n + p$. For any bit sequence of a finite length, the analog value is determined by all bits in the window, where all bits have the same weight. The analog value of a Delta Sigma bit-stream can be simply negated by reversing the polarity of every bit in the stream. A few examples of first-order Delta Sigma-modulated (DSM) bit-streams are given in Table 1.

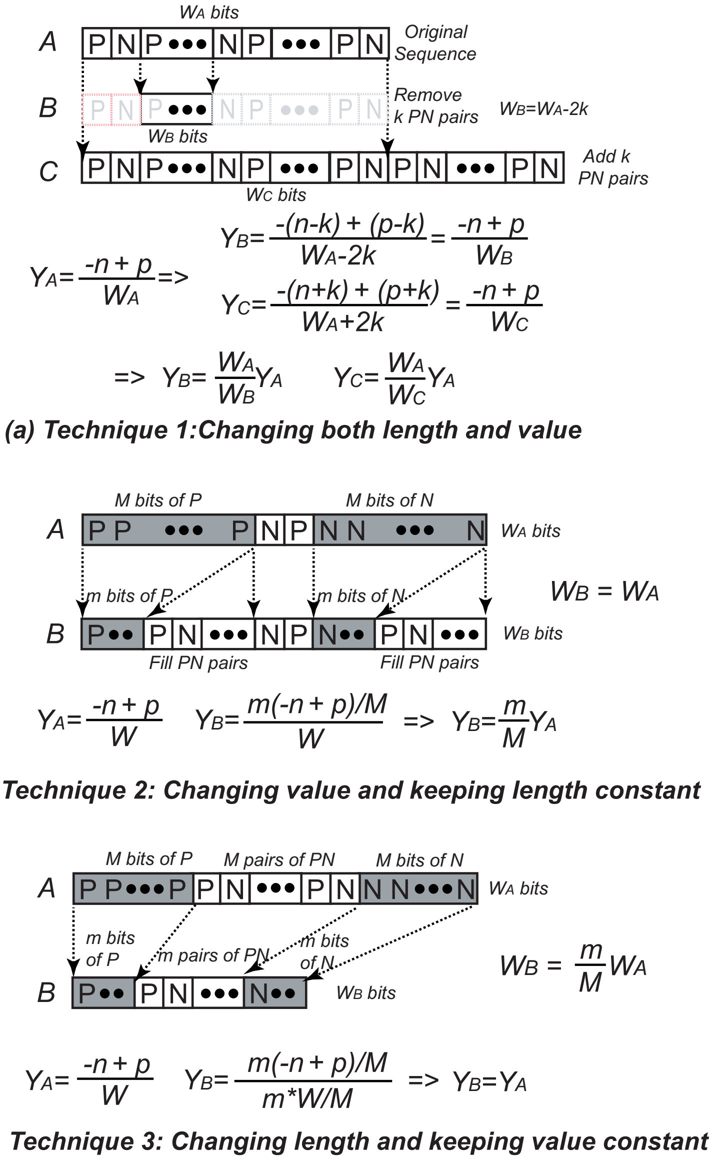
Table 1. Examples of first-order Delta Sigma-modulated (DSM) bit-streams. P, positive; N, negative.

Analog Value	Delta Sigma Stream								
1	P	P	P	P	P	P	P	P	...
0.5	P	P	P	N	P	P	P	N	...
0	P	N	P	N	P	N	P	N	...
−0.5	P	N	N	N	P	N	N	N	...
−1	N	N	N	N	N	N	N	N	...

The instantaneous analog value of a running Delta Sigma bit-stream can also be calculated using Equation (1) with a pre-determined window length, e.g., *W* bits. Thus, Equation (1) can be applied to both a bit sequence of finite length and a running bit-stream of infinite length. This result applies to any-order Delta Sigma modulated bit-streams, including first-order and higher-order Delta Sigma modulated bit-streams.

2.2. Techniques for Modifying Instantaneous Analog Values and/or the Length of the Delta Sigma-Modulated Bit-Streams

We present techniques for processing Delta Sigma bit-streams based on Equation (1). Assume that we have a Delta Sigma-modulated bit sequence of finite length. From Equation (1), we see that one *P* bit and one *N* bit in a Delta Sigma bit sequence cancel each other and do not contribute to the value of the numerator, but undoubtedly contributed to the length of the window, *i.e.*, the value of the denominator. In other words, removing or adding *P*-*N* pairs in a Delta Sigma bit sequence changes its analog value by the ratio of the original length over the new length, as shown in Figure 4a. In another case, assume that *m* and *M* are integers and *m* is smaller than *M*. If we merge every *M P* bits and every *M N* bits, excluding *P*-*N* pairs, into *m* bits each and use *P*-*N* pairs to fill the space and keep the length of the bit sequence constant, then the analog value of the new bit-stream is reduced by *m/M*, as shown in Figure 4b. Moreover, if we merge every *M P* and every *M N* bits into *m* bits each and merge *M P*-*N* pairs from the original sequence into *m P*-*N* pairs, then the length of the Delta Sigma bit sequence is reduced by *m/M*, while its analog value is preserved, as shown in Figure 4c.


WA bits
A
P N P ••• N P ••• P N
Original Sequence
B
P N P ••• N P ••• P N
Remove k PN pairs
WB=WA-2k
WB bits
C
P N P ••• N P ••• P N P N ••• P N
Add k PN pairs
WC bits
YA= (-n + p)/WA =>
YB= (-(n-k) + (p-k))/(WA-2k) = (-n + p)/WB
YC= (-(n+k) + (p+k))/(WA+2k) = (-n + p)/WC
=> YB= (WA/WB) YA   YC= (WA/WC) YA

(a) Technique 1:Changing both length and value

(b) Technique 2: Changing value and keeping length constant

(c) Technique 3: Changing length and keeping value constant

Figure 4. *P*-*N* pair method: Delta Sigma signal-processing techniques to adjust the length and value of a Delta Sigma bit sequence. *A* is the original Delta Sigma bit sequence with W_A bits, while *B* and *C* are the adjusted Delta Sigma bit sequence with lengths of W_B and W_C, respectively. (**a**) Changing the value by changing the length; (**b**) changing the value while keeping the length constant; and (**c**) changing the length while keeping the value constant.

In summary, we have created three techniques to represent these operations, which are based on the *P-N* pair method:

- Technique 1: Create a new Delta Sigma bit sequence by adding or removing *P-N* pairs from the original Delta Sigma bit sequence, changing its length from W_A bits to W_B bits. The new bit sequence has an analog value of W_A/W_B of the original bit sequence.
- Technique 2: Create a new Delta Sigma bit sequence by merging every *M* homogeneous bits from the original Delta Sigma bit sequence into *m* bits and using *P-N* pairs to fill the space and keep the length of the sequence constant. The new Delta Sigma bit sequence has an analog value of *m/M* of the original sequence.
- Technique 3: Create a new Delta Sigma bit sequence by merging every *M* homogeneous bits from the original Delta Sigma bit sequence into *m* bits and merging *M P-N* pairs into *m P-N* pairs. By doing so, the new sequence has the same analog value as the original sequence, while the length of the new bit sequence is reduced to *m/M* of its original length.

Our proposed Delta Sigma circuits are designed based on the above techniques.

3. Delta Sigma Adders

In this section, we introduce Delta Sigma adders using the proposed *P-N* pair method. Adding is a basic linear operation, which is indispensable in DSP. There are two types of adders in DSP: sum adders and average adders. In Delta Sigma processing, a sum adder generates a new Delta Sigma bit-stream based on the input bit-streams. The analog value of the output bit-stream is equal to the sum of the analog values of the addend bit-streams. On the other hand, an average adder generates a bit-stream whose analog value is equal to the average of all of the analog values of the addend bit-streams.

Here, we show a basic example of adding Delta Sigma bit-streams. An average bit-stream can be simply created by mixing all of the addend bit-streams bit by bit. As shown in Figure 5, the output bit sequence *Z* is generated by mixing input bit sequences *A*, *B* and *C*. Here, the analog value of *Z* is 1/3 of the sum ($y_A + y_B + y_C$). However, its length is three-times that of one addend. Therefore, *Z* requires a three-times faster clock than the addend bit-streams. In order to obtain the output bit-stream with the same data rate of the input bit-streams, the length of the mixed bit-stream must be reduced. This can be done using Technique 3 in Section 2 by deleting *P-N* pairs and homogeneous bits from the mixed bit sequence until the output bit sequence has the same length of the addends. This keeps the analog value while reducing the length of the bit sequence. On the other hand, by applying Technique 1 in Section 2, we can reduce the length of the bit sequence by three while increasing its analog value by three, thus removing the coefficient 1/3 and obtaining the sum bit sequence.

An on-the-fly processing circuit can be designed to perform the aforementioned processing of long combined bit-streams. The basic idea of an on-the-fly Delta Sigma adder is to remove *P-N* pairs simultaneously from the addend Delta Sigma bit-streams in order to keep the data rate of the sum bit-stream the same as that of the addends. This requires counting the number of *P* and *N* bits from all addends in each clock cycle. A counter performs the on-the-fly operation based on:

$$Y = \sum_{i=1}^{M} X_i \tag{2}$$

In Equation (2), Y is the counting result. X_i is the instantaneous bit of the i-th input bit-stream, and M is the number of addends. To make a *P* bit cancel an *N* bit, X_i equals $+1$ if the input bit is *P*, while X_i equals -1 if the input bit is *N*. This guarantees that *P*-*N* pairs can be removed from the result. Y represents the number of the surplus of either *P* bits or *N* bits. Thus, the result Y is either a zero, a positive number or a negative number. We use a signed binary number, for example the two's complement format, to represent the result *Y*.

Nevertheless, since $+1$ and -1 are applied in Equation (2), the system requires dual power supplies or a tri-state digital system similar to [21], which is not compatible with current two-state digital systems. We need a counter that can work in a two-state digital system. To make X_i compatible with the commonly-used binary system, we modify Equation (2) to get:

$$Y' = 2 \times \sum_{i=1}^{M} X_i - M \tag{3}$$

In Equation (3) X_i equals one if the input bit is *P* or equals zero if the input bit is *N*. A binary counter for calculating Equation (3) can be realized using an M-input one-bit binary adder in a two-state digital system. Here, a positive value of the *Y'* means *Y' P* bits remain after removing *P*-*N* pairs from all of the input bits, while a negative value of the *Y'* means $-$*Y' N* bits remain.

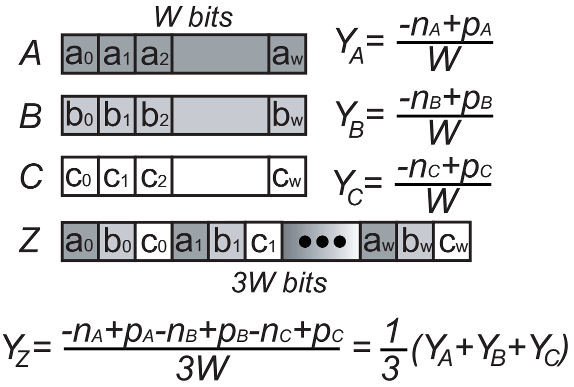

W bits
A
a0 a1 a2 aw
$Y_A = \frac{-n_A+p_A}{W}$
B
b0 b1 b2 bw
$Y_B = \frac{-n_B+p_B}{W}$
C
c0 c1 c2 cw
$Y_C = \frac{-n_C+p_C}{W}$
Z
a0 b0 c0 a1 b1 c1 ••• aw bw cw
3W bits
$Y_Z = \frac{-n_A+p_A-n_B+p_B-n_C+p_C}{3W} = \frac{1}{3}(Y_A+Y_B+Y_C)$

Figure 5. Averaging of the Delta Sigma bit-streams by mixing bits.

3.1. Delta Sigma Sum Adder

Based on Equation (3), we design a Delta Sigma sum adder. A sum adder takes M Delta Sigma bit-streams as inputs and generates a sum bit-stream. M is an arbitrary positive integer. In each clock cycle, the sum adder uses the counter in Equation (3) to count all of the input bits and obtains the result in a signed binary format. Besides the current result in the counter, the adder also needs to consider the previous value of the counter. An adder adds the previous value, which is stored in memory, to the current counter value. If the added result is positive (negative), then the sum output is one bit *P* (*N*). If the added result is greater than $+1$ or less than -1, then the added result is updated and stored in memory for the next clock cycle. The value stored in memory has the same polarity of the added result, but its absolute value is reduced by one from the added result. This is because the output takes one bit from the value. If the added result is zero, then the stored value is zero, and the output takes one bit from a toggling signal. This is because a toggling bit-stream has an analog value of zero, so adding toggling bits does not affect the analog value of the sum bit-stream.

A simplified schematic of a Delta Sigma sum adder is shown in Figure 6. X_1 to X_M are Delta Sigma bit-streams as addends and inputs to a *Counter*. The *Counter* operates using Equation (3) and generates a signed binary number *Y'*. *Y'* is added to the value in *Q* by a binary adder. Here, *Q* is a register storing the previous results from the *Counter*. *Q* has the same number of bits as *Y'*. The added result of *Y'* and *Q* determines the output bit through *C*. Here, *C* is a digital comparator, whose output is one *P* bit if the input binary number is above zero or *N* if the input is less than zero. If the added result of *Y'* and *Q* is zero, then the output takes the value from a toggling bit *T*, and *T* toggles for the next cycle, *i.e.*, T = NOT(T). If the added result of *Y'* and *Q* is not zero, a new value of *Q* is calculated and stored. The truth table of the circuit is summarized in Table 2. To avoid overflow errors, when *Q* reaches its maximum or minimum value, it should keep the value instead of rolling back to zero.

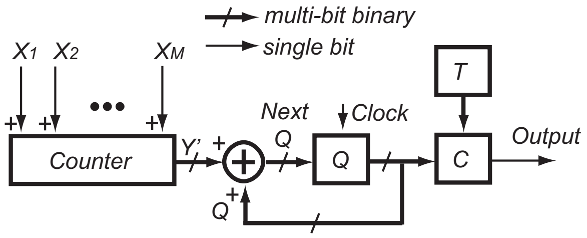


Figure 6. Simplified schematic of the Delta Sigma sum adder.

Table 2. Truth table of the sum adder.

Q + Y'	**≥1**	**≤−1**	**0**
Next Q	Q + Y' − 1	Q + Y' + 1	0
Output	P	N	T
T	T	T	NOT(T)

3.2. Delta Sigma Average Adder

Based on the Delta Sigma sum adder, we design the Delta Sigma average adder. The average adder generates an output Delta Sigma bit-stream, whose analog value is the average of all input Delta Sigma bit-streams. The average adder is essentially a digital Delta Sigma modulator, which can be realized in first-order or higher-order. The order number of the digital Delta Sigma modulator should be equal to or higher than the order number of the input bit-streams to maintain the signal-to-noise and distortion ratio (SNDR). For example, if the input bit-streams are generated by second-order Delta Sigma modulators, then the digital Delta Sigma modulator in the average adder should be at least second-order. Otherwise, the SNDR of the output bit-stream decreases from that of the input.

Simplified schematics of the Delta Sigma average adders are shown in Figure 7. Figure 7a shows a first-order average adder, which has a similar structure to the sum adder in Figure 6. It also shares the same truth table with the sum adder. The difference is that the average adder has a feedback path from the output to the *Counter*. The equation of the *Counter* in an average adder is:

$$\begin{aligned} Y' &= 2 \times \sum_{i=1}^{M} X_i - M - M \times (2 \times Output - 1) \\ &= 2 \times \sum_{i=1}^{M} X_i - 2 \times M \times Output \end{aligned} \tag{4}$$

where *Output* equals one (zero) if it is *P* (*N*). Thus, in each clock cycle, the input bits are added to the feedback bits from the output. The maximum value of feedback is the same as the sum of the maximum value of all input bits. The feedback bits are negated from the output bit, *i.e.*, every feedback bit is *P* if the output bit is *N*, or *vice versa*. The added result of *Y'* and *Q* determines the output bit and the next *Q* according to Table 2. We can also extend the first-order average adder to a second-order average adder, as shown in Figure 7b.

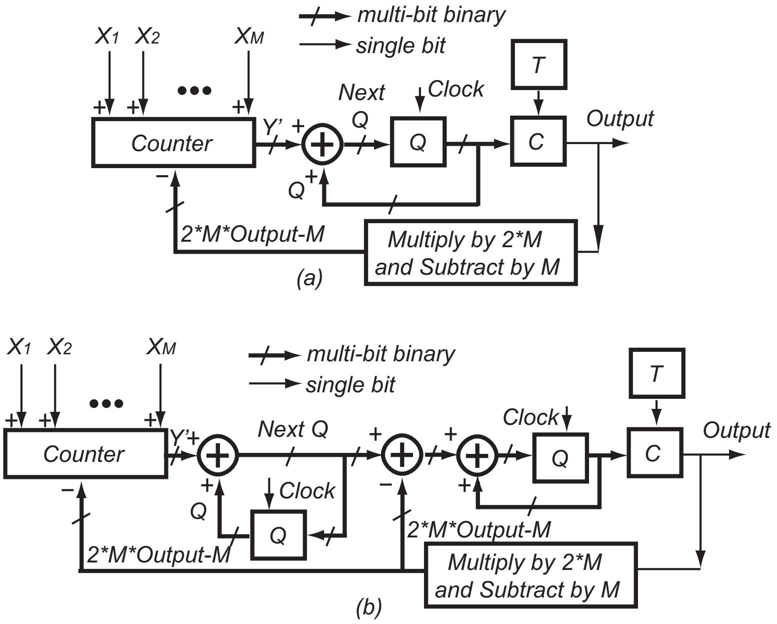


Figure 7. Simplified schematic of the Delta Sigma average adders. (**a**) First-order average adder; and (**b**) second-order average adder.

The average adder can also be used as a coefficient multiplier by a coefficient less than unity. The coefficient multiplier generates a new Delta Sigma bit-steam, whose analog value equals the analog value of the input bit-stream multiplied by a pre-defined coefficient. For example, the coefficient can be *m*/*M*, where both *m* and *M* are integers and *m* is smaller than *M*. This can be realized by using an M-input average adder with *m* of its inputs connected to the input bit-stream, while the other $M - m$ inputs are connected to a toggling bit-stream, as shown in Figure 8a, where M = 4 and m = 3. The toggling bit-stream, which has an analog value of zero, is running at the same bit rate of the input bit-stream. When a higher resolution of the coefficient is required, a cascade structure can be applied using more stages of average adders, as shown in Figure 8b.

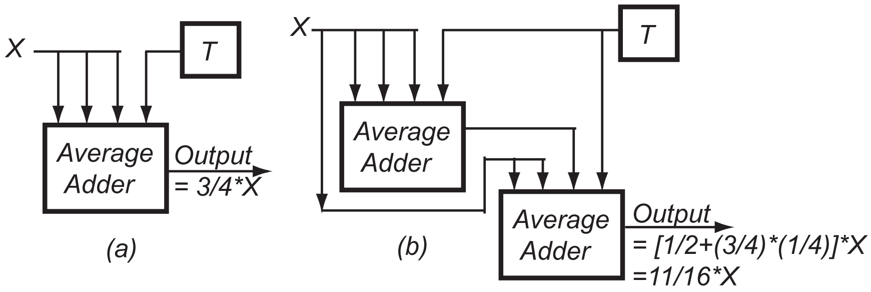


Figure 8. Simplified schematic of the Delta Sigma coefficient multiplier based on the Delta Sigma average adder: (**a**) one stage; and (**b**) cascade structure.

The proposed Delta Sigma adders have several advantages. First, they can take any number of input addend bit-streams compared to the two-input Delta Sigma adders in [22]. Second, the circuit complexity and power consumption are lower than other Delta Sigma adders. This is because when adding more addends, it only requires a linear increase in the size of the counter and memory. Therefore, the complexity of the proposed Delta Sigma adder circuits of N inputs is $\Theta(N)$, which is better than $\Theta(N^2)$ when circuits using sorting networks are applied [13]. Furthermore, compared to multi-bit binary adders, the proposed Delta Sigma adder can take addends at different clock rates, as long as the adder clock is equal to or faster than the highest clock rate among all addends.

4. Simulation of Delta Sigma Adders

This section presents simulation results of the Delta Sigma sum adder and the average adder. We simulated the proposed sum adder and the average adder in MATLAB using the Delta Sigma Toolbox [34]. Figure 9 shows the simulated SNDR plotted against the input power of a two-input average adder and a two-input sum adder both at an oversampling ratio (OSR) = 1024, where the inputs are Delta Sigma-modulated sine waves. Because the sum adder does not have feedback, the previous value Q representing the "temporal overflow" bits may not be able to be completely assimilated by the output bit-stream. The temporal overflow remains in the system until the input bit-stream changes the polarity of its analog value. Therefore, the temporal overflow can stay in the system for a long time and affects the output results. This process introduces extra noise in the output bit-stream. This noise increases when the input amplitude is high, because there are more "temporal overflow" bits at a higher amplitude. Furthermore, this noise increases with the number of addends. This is because when the number of addends is large, there are higher chances of having temporal overflow when input signals are in-phase. Figure 10 shows MATLAB-simulated SNDR as a function of OSR and the number of inputs in a Delta Sigma sum adder, where the input signals are sine waves modulated by second-order Delta Sigma modulators.

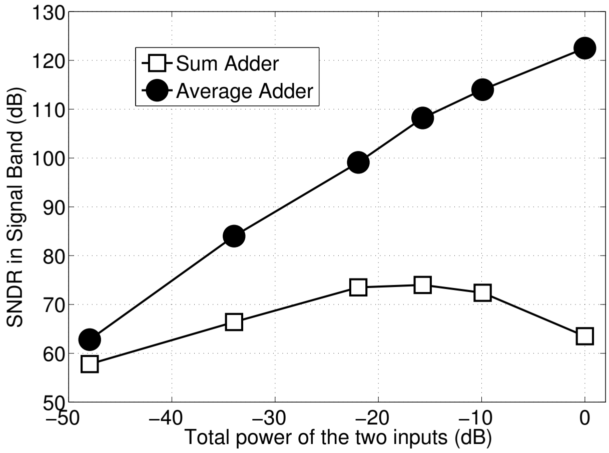


Figure 9. Simulated output signal-to-noise and distortion ratio (SNDR) *v.s.* input power in a two-input sum adder and a two-input second-order average adder both with oversampling ratio (OSR) = 1024.

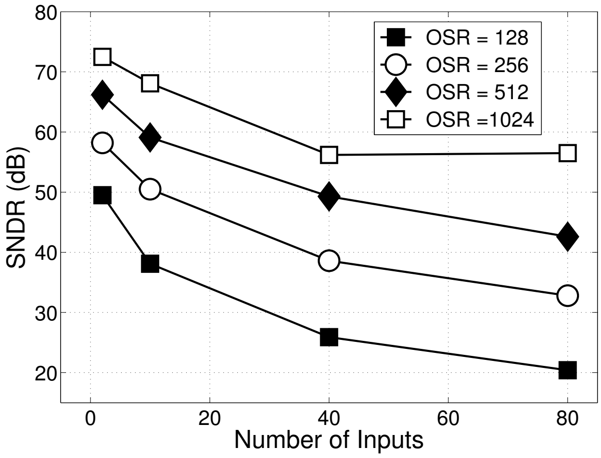


Figure 10. Simulated output SNDR in a Delta Sigma sum adder at different numbers of inputs and OSR.

We simulated the output bit-stream spectrums of the sum and average adders, as shown in Figure 11. Compared to the sum adder, the average adder has a feedback path that is able to completely merge its previous value with the current input bits within one clock cycle. Thus, there are no "temporal overflow" bits. Moreover, the feedback path enables the possibility of building higher-order digital Delta Sigma modulators that can further improve the noise shaping performance of the output bit-stream. In Figure 11, we use ten sine waves with the same amplitude and different frequencies for the adders. The ten inputs are second-order bit-streams of sine waves with OSR = 1024. The frequencies are normalized to the sampling frequency of the Delta Sigma modulators. Figure 11 shows that the noise floor of the sum adder is higher than the average adder by 50 dB. Furthermore, the noise shaping slopes in the sum adder and the first-order average adder are less than 30 dB/dec, while the output of the second-order average adder maintains a 40-dB/dec noise shaping slope, which is the same as the spectrum of the ideal second-order sum bit-stream.

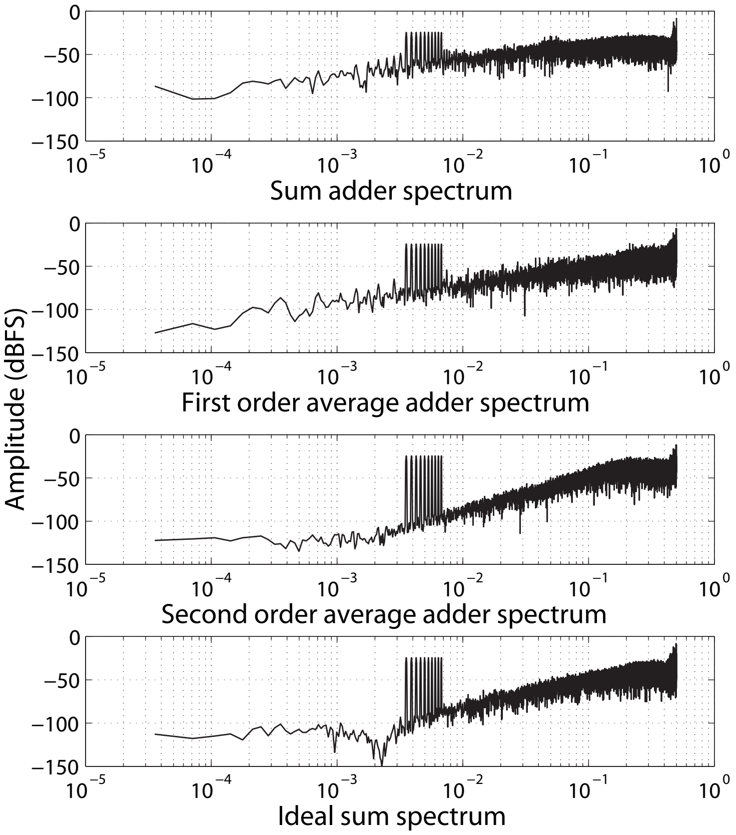


Figure 11. Simulated spectrums of Delta Sigma adders with ten addends.

In summary, while in a multi-bit binary DSP system, either a sum adder or an average adder can be used in digital filters, in a Delta Sigma DSP system, the average adder is preferred because of better noise performance.

5. Downsampling and Delta Sigma Compressor

In this section, we discuss the techniques of changing the length of a Delta Sigma bit sequence while preserving its analog value. This is equivalent to adjusting the clock rate of a Delta Sigma bit-stream. We are particularly interested in downsampling a Delta Sigma bit-stream. However, we cannot simply use a lower speed clock to sample the Delta Sigma bit-stream and hope to preserve its analog value. Downsampling of a Delta Sigma bit-stream is equivalent to data compression. An algorithm is necessary to preserve the analog value while reducing the number of bits. Unlike complicated compressing algorithms in the multi-bit system, this work proposes a simple method based on Technique 3 in Section 2. Essentially, we reduce the number of homogeneous bits and *P*-*N* pairs with the same ratio, for instance *K*. Then, the new bit sequence has the same analog value as the original bit sequence, and the length of the new bit sequence is $1/K$ of the original bit sequence. This algorithm can be realized with simple circuitry using on-the-fly processing.

Here, we give an example of a Delta Sigma compressor circuit to reduce the data rate by 1/2. The circuit is shown in Figure 12. Assume that a Delta Sigma bit-stream is coming into the compressor at a clock frequency f. We use an $f/2$ clock to sample two incoming bits at once. By doing so, any bit is only checked once in a two-bit window. We mark the two bits in the window as a_0 and a_1, where a_0 is the preceding bit, while a_1 is the succeeding bit. If the two bits are homogenous, then the output bit X_0 should be the same polarity of the inputs. If the two incoming bits are heterogeneous, we take the value from a toggling bit T as the output bit. The toggling bit T toggles whenever the incoming two bits are heterogeneous. Thus, both the number of *P-N* pairs and the number of homogeneous bits are reduced by 1/2 at the output. The value of X_0 at frequency $f/2$ is the output Delta Sigma bit-stream. The Delta Sigma compressor can also be realized using the Delta Sigma average adder by adding sub-sequences from the original sequence, at the cost of increase of circuit complexity.

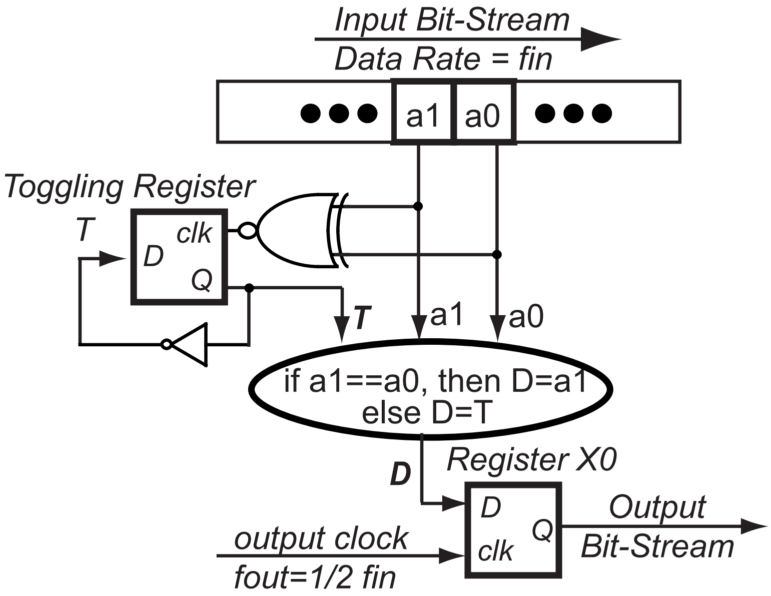


Figure 12. Simplified circuit schematic of the Delta Sigma compressor.

We simulated the SNDR of the compressor in MATLAB. Figure 13 shows the SNDR of the input and output of the Delta Sigma compressor *versus* OSR. The input is created by a second-order Delta Sigma modulator. Although the algorithm keeps the output analog value the same as the input analog value, Figure 13 shows that the compressor reduces the SNDR because the OSR of the output bit-stream is half that of the input bit-stream. Figure 14 shows the simulated input and output spectra of the Delta Sigma compressor with a second-order input bit-stream and OSR = 1024. The noise floor of the output spectrum increased by 50 dB compared to the input spectrum. Furthermore, the noise shaping slope of the output spectrum reduced to 20 dB/dec, which makes the output a first-order bit-stream. The compressor based on the first-order sum adder can only preserve the noise shaping of a first-order Delta Sigma bit-stream.

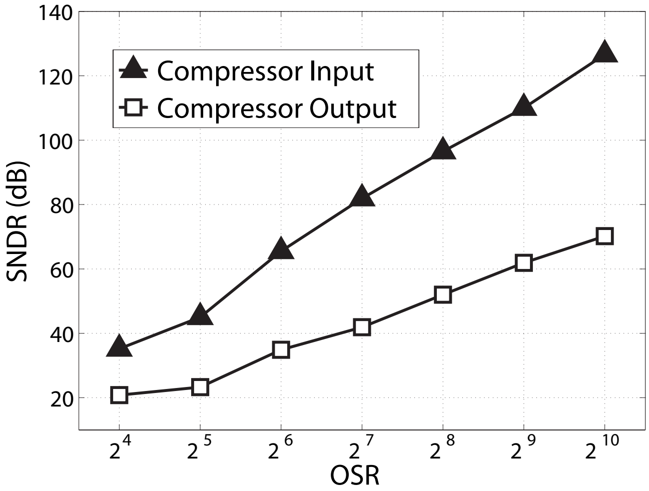


Figure 13. Simulated signal-to-noise-ratio of the Delta Sigma compressor.

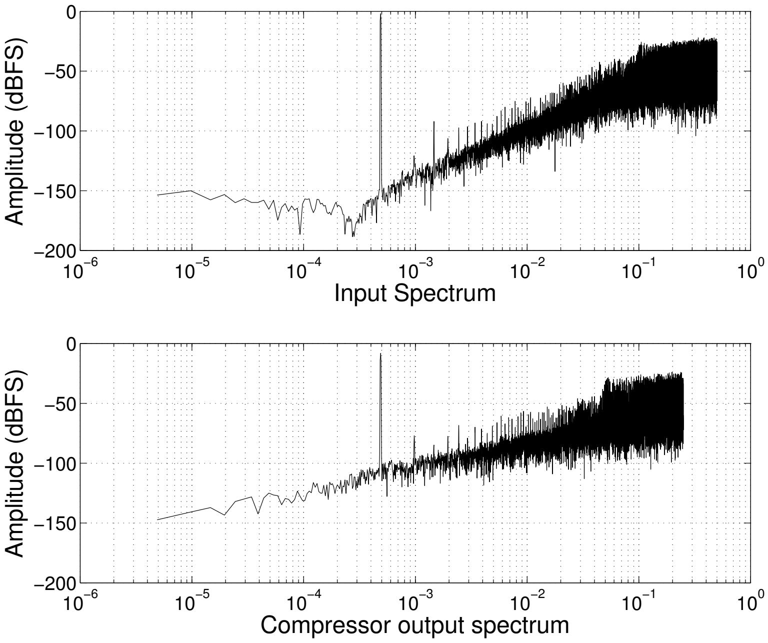


Figure 14. Simulated input and output spectrum of the Delta Sigma compressor.

6. Measurement Results

We tested the proposed Delta Sigma signal processing circuits using a field programmable gate array (FPGA). The testbed is shown in Figure 15. We used second-order Delta Sigma modulators AD7401 [35] to generate the Delta Sigma bit-streams from analog signal generators. The Delta Sigma bit-streams are fed into the proposed Delta Sigma processing circuits, which include the sum adder and the average adder. The processing circuits are coded in Verilog and simulated by the Xilinx ISE isim simulator and then implemented on an Opal Kelly Spartan-3 FPGA board [36]. In our experiment, an external function generator provided the clock signals to both the Delta Sigma modulator and the Delta Sigma processing circuit. Fourth-order analog filters recovered analog waveforms from both the Delta Sigma modulator and the Delta Sigma processing circuit. Oscilloscopes calculated the spectra of the digital signals using the FFT function. Both the AD7401's output port and the Opal Kelly FPGA are powered by a 3.3-V power supply.

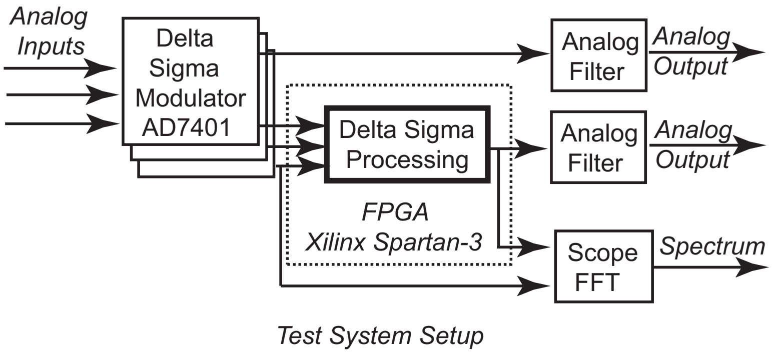


Figure 15. Testbed setup for testing the Delta Sigma processing circuits.

Figure 16 shows the input and output waveforms of a three-input second-order average adder. The three input signals are 113-Hz, 157-Hz and 213-Hz sine waves all with a 200-mVpp amplitude. The output waveform is recovered from a fourth-order analog filter. The output amplitude is higher than the amplitude of the inputs because of the internal gain of the AD7401. Figure 17 shows the FFT result of the second-order average adder output, which is calculated by the oscilloscope. In Figures 16 and 17, the functionality of the three-input average adder is verified. Furthermore, the 40-dB/dec noise shaping slope in Figure 17 shows that the processed signal is a second-order Delta Sigma bit-stream. In order to get a better view of the noise shaping slope, the OSR is set at approximately 100 for the three signals, which makes the sampling clock of the AD7401 and the clock of the Delta Sigma average adder both at 4 kHz.

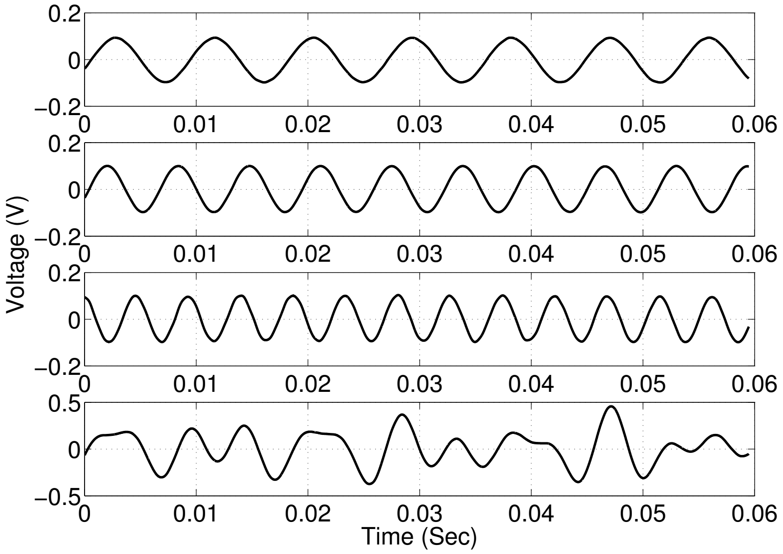


Figure 16. Measured time-domain waveforms of a three-input Delta Sigma second-order average adder recovered by analog filters. From top to bottom: Input 1, Input 2, Input 3 and output.

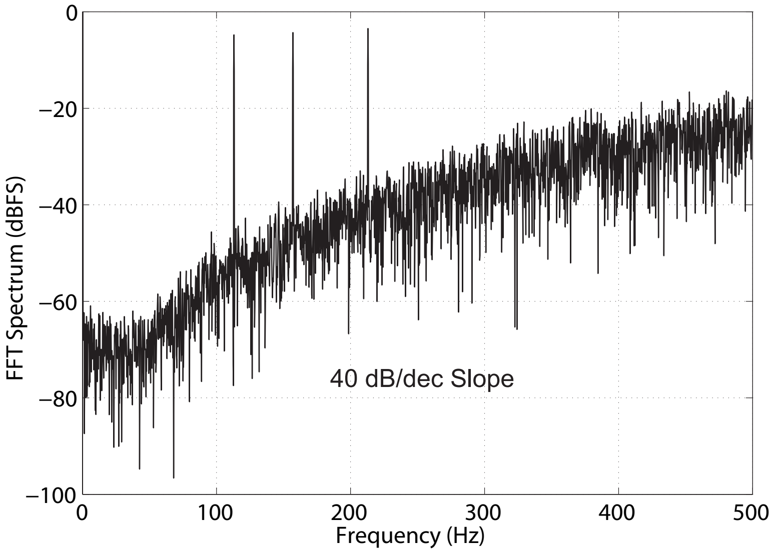


Figure 17. Measured FFT spectrum of the output of a three-input Delta Sigma second-order average adder. The noise shaping slope is 40 dB/dec.

We also recorded the digital input and output waveforms of the Delta Sigma compressor, as shown in Figure 18. In the test, we give a DC input of 500 mV and use a sampling rate of 500 kHz to obtain the input second-order Delta Sigma bit-stream. The output of the Delta Sigma compressor has half the data rate of the input Delta Sigma bit-stream. The input and output bit-streams have the same average duty cycle, which means the analog value of the input is preserved in the output.

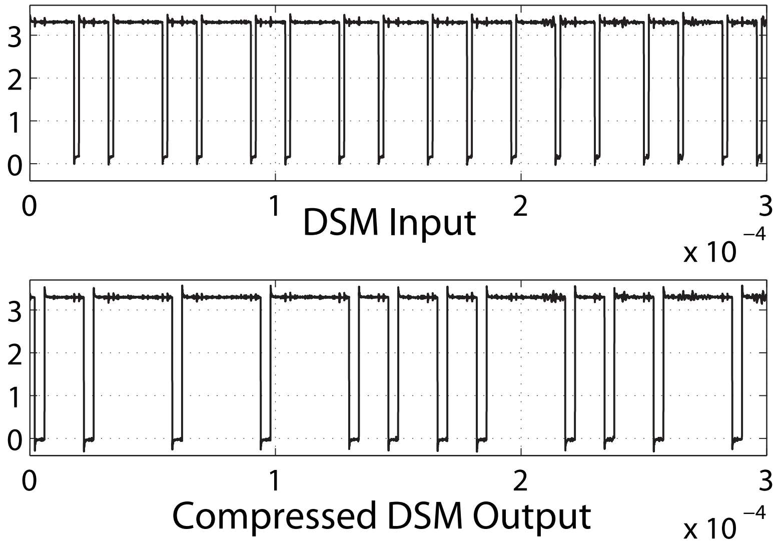


Figure 18. Measured time-domain input and output waveforms of the Delta Sigma compressor.

7. Comparison between Delta Sigma Adders and Multi-Bit Adders

In linear DSP systems, higher-order FIR filters require adding many addends together at once. The proposed Delta Sigma processing circuits can significantly reduce circuit complexity compared to multi-bit processing circuits, especially when the number of inputs in an adder is large. In Figure 19, we tested the integrated circuit area of the Delta Sigma sum adder, average adder and a multi-bit ripple-carry adder. The Delta Sigma sum adder has a simpler structure than the average adder. However, since the sum adder introduces extra noise, it is fair to compare the multi-bit binary adder and the Delta Sigma average adder, since neither circuit adds noise. We synthesized the Verilog code in the IBM 0.18-µm CMOS process using the Cadence RTL Compiler. The results show that if we replace the binary adder with the Delta Sigma average adder, the circuit area is reduced by 94% when the number of inputs is 100.

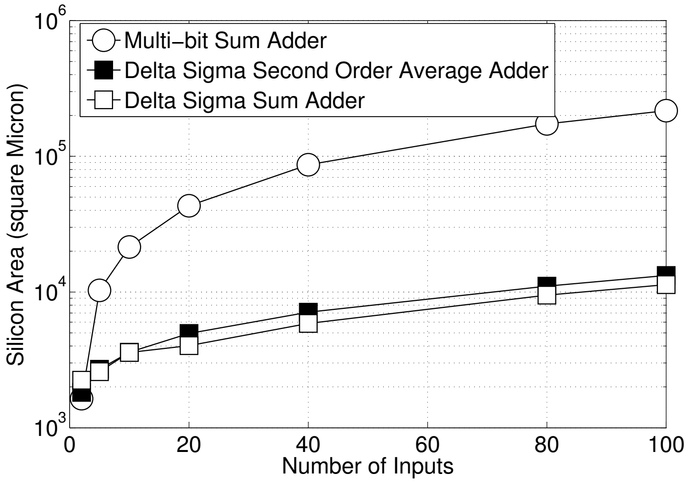


Figure 19. Comparison of silicon area between the Delta Sigma adders and the binary adder.

We also compared the power consumption between the proposed Delta Sigma adders and the multi-bit adder. In order to make a fair comparison, we refer to Texas Instruments ADS131E04 [14]. It contains a Delta Sigma ADC made of a second-order Delta Sigma modulator, which operates at a fixed clock of 1024 kHz. The output sample rate of the ADC is 16 kSamples/s with 24-bit resolution. This means that, if we use the Delta Sigma modulator in the ADS131E04 as the ADC and make the Delta Sigma sampling rate 1024 kSamples/s, the result is equivalent to having a 24-bit binary ADC with a sampling rate of 16 kSamples/s. Thus, utilizing the same resolution and sampling rate of the ADC, when we design DSP circuits, it is fair to compare the power consumption of the proposed Delta Sigma average adder at a 1024-kHz clock rate to a 24-bit adder at a 16-kHz clock rate. We compared the power consumption for different numbers of inputs between the Delta Sigma adders and multi-bit binary adders. The comparison includes leakage power, net power and switching power, which are shown in Figures 20–22, respectively. For a 100-input adder, the proposed Delta Sigma average adder reduces the switching power by more than 96% compared to the multi-bit adder.

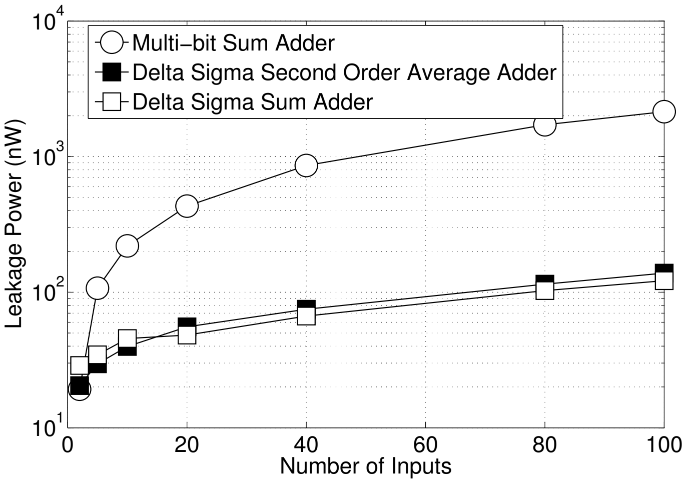


Figure 20. Comparison of leakage power between the Delta Sigma adders and a binary adder.

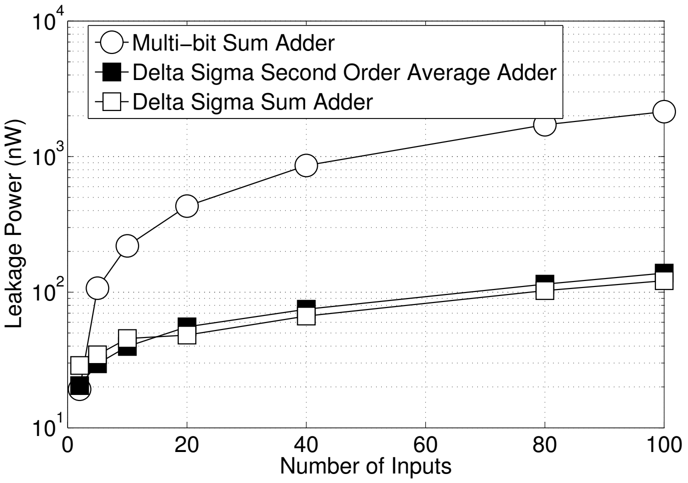


Figure 21. Comparison of net power between the Delta Sigma adders and a binary adders.

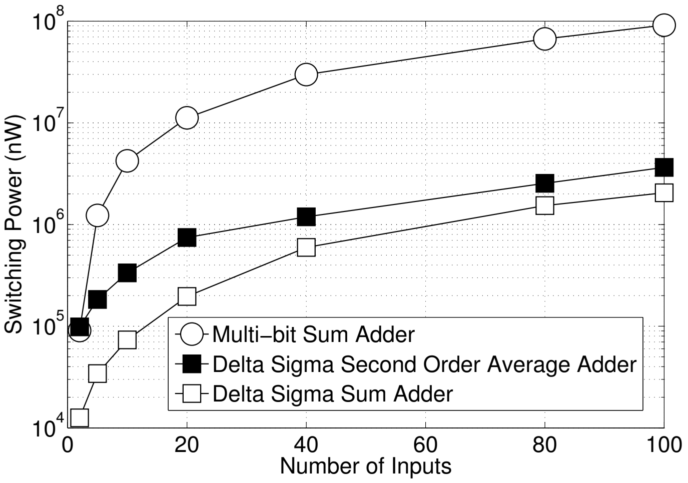


Figure 22. Comparison of switching power between the Delta Sigma adders and a binary adder.

8. Conclusions

In this paper, we presented several low-complexity low-power Delta Sigma processing circuits for future IoT sensors, including the Delta Sigma sum adder, average adder and coefficient multiplier. These on-the-fly processing circuits take the input Delta Sigma bit-streams and generate an output Delta Sigma bit-stream in real time. These circuits are compatible with CMOS digital logic. A comparison between our proposed circuit and other methods is shown in Table 3. We simulated the Delta Sigma adders in MATLAB and tested the functionality of the proposed circuits on an FPGA. We also presented synthesis results in the IBM 0.18-μm process. We compared silicon area and power consumption between Delta Sigma average adders and conventional multi-bit binary adders. In our comparison of Delta Sigma and multi-bit adders, we chose 100 addends. FIR (finite impulse response) filters with 100 or more taps are not uncommon in DSP algorithms, where each tap represents one multiplication and one addend in the physical realization of the filter. Synthesis results showed that the proposed Delta Sigma adders can significantly reduce silicon area and power consumption, which makes them competitive candidates for future low-power, low-complexity DSP circuits for IoT.

Table 3. Comparison between Circuits.

Comparison	[22] **1978**	[21] **2009**	[13] **2012**	**This Work** **2015**
Circuit Structure	Binary Adder	Tri-State Digital	Sorting Network	Counter
DSM Order	First-Order Only	First-Order Only	First- and Higher Order	First- and Higher Order
Circuit Complexity	Θ(N)	Θ(N)	$\Theta(\mathrm{N}log(\mathrm{N}^2))$	Θ(N))

Author Contributions

Wei Tang conceived of the idea and directed the work. Yifei Liu performed the simulation, the experiment, the synthesis and obtained the data. Wei Tang wrote the manuscript. Paul M. Furth helped with earlier versions of the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Sai, V.; Mickle, M. Exploring energy efficient architectures in passive wireless nodes for IoT applications. *IEEE Circuits Syst. Mag.* **2014**, *14*, 48–54.

2. Bachmann, C.; Ashouei, M.; Pop, V.; Vidojkovic, M.; Groot, H.; Gyselinckx, B. Low-power wireless sensor nodes for ubiquitous long-term biomedical signal monitoring. *IEEE Commun. Mag.* **2012**, *50*, 20–27.
3. Chen, G.; Hanson, S.; Blaauw, D.; Sylvester, D. Circuit design advances for wireless sensing applications. *Proc. IEEE* **2010**, *98*, 1808–1827.
4. *Fiscal Year 2014 Historical Tables Budget of the U.S. Government*; Office of Management and Budget: Alexandria, VA, USA, 2014.
5. Yoo, J.; Yan, L.; El-Damak, D.; Altaf, M.; Shoeb, A.; Chandrakasan, A. An 8-channel scalable EEG acquisition SoC with patient-specific seizure classification and recording processor. *IEEE J. Solid-State Circuits* **2013**, *48*, 214–228.
6. Yoo, J.; Yan, L.; El-Damak, D.; Bin Altaf, M.; Shoeb, A.; Yoo, H.J.; Chandrakasan, A. An 8-channel scalable EEG acquisition SoC with fully integrated patient-specific seizure classification and recording processor. In Proceedings of the 2012 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 19–23 February 2012; pp. 292–294.
7. Qian, C.; Shi, J.; Parramon, J.; Sanchez-Sinencio, E. A low-power configurable neural recording system for epileptic seizure detection. *IEEE Trans. Biomed. Circuits Syst.* **2013**, *7*, 499–512.
8. Custodio, J.; Goes, J.; Paulino, N.; Oliveira, J.; Bruun, E. A 1.2-V 165-µW 0.29-mm^2 multibit sigma-delta ADC for hearing aids using nonlinear DACs and with over 91 dB dynamic-range. *IEEE Trans. Biomed. Circuits Syst.* **2013**, *7*, 376–385.
9. Porrazzo, S.; Morgado, A.; Segundo Bello, D.; Cannillo, F.; van Hoof, C.; Yazicioglu, A.; van Roermund, A.; Cantatore, E. A 155 µW 88-dB DR discrete-time $\Delta\Sigma$ modulator for digital hearing aids exploiting a summing SAR ADC quantizer. *IEEE Trans. Biomed. Circuits Syst.* **2013**, *7*, 573–582.
10. Deligoz, I.; Naqvi, S.; Copani, T.; Kiaei, S.; Bakkaloglu, B.; Je, S.S.; Chae, J. A MEMS-based power-scalable hearing aid analog front end. *IEEE Trans. Biomed. Circuits Syst.* **2011**, *5*, 201–213.
11. Lopez-Morillo, E.; Carvajal, R.; Munoz, F.; El Gmili, H.; Lopez-Martin, A.; Ramirez-Angulo, J.; Rodriguez-Villegas, E. A 1.2-V 140-nW 10-bit sigma-delta modulator for electroencephalogram applications. *IEEE Trans. Biomed. Circuits Syst.* **2008**, *2*, 223–230.
12. Stanacevic, M.; Murari, K.; Rege, A.; Cauwenberghs, G.; Thakor, N. VLSI potentiostat array with oversampling gain modulation for wide-range neurotransmitter sensing. *IEEE Trans. Biomed. Circuits Syst.* **2007**, *1*, 63–72.
13. Fujisaka, H.; Kamio, T.; Ahn, C.J.; Sakamoto, M.; Haeiwa, K. Sorter-based arithmetic circuits for sigma-delta domain signal processing—Part I: Addition, approximate transcendental functions, and log-domain operations. *IEEE Trans. Biomed. Circuits Syst.* **2012**, *59*, 1952–1965.
14. *ADS131E04 Analog Front-End for Power Monitoring, Control and Protection*; Technical Report; Texas Instruments: Harris, TX, USA, 2013.
15. Melanson, J. Signal Processing System Using Delta-Sigma Modulation Having An Internal Stabilizer Path With Direct Output-to-Integrator Connection. U.S. Patent 7,696,913, 13 April 2010. Available online: https://www.google.com/patents/US7696913.

16. Choi, J.; Han, S.; Jang, J.; Yu, H. Structure of Delta-Sigma Fractional Divider. U.S. Patent App. 10/179,840, 12 June 2003. Available online: https://www.google.com/patents/US20030108143.
17. Cho, M.; Kim, Y.; Kwon, J. Coefficient Multiplier and Digital Delta-Sigma Modulator Using the Same. U.S. Patent App. 12/783,294, 16 June 2011. Available online: https://www.google.com/patents/US20110140940.
18. Niwa, A.; Ueno, Y. Delta-Sigma Modulator and Signal Processing System. U.S. Patent 8,581,763, 12 November 2013. Available online: https://www.google.com/patents/US8581763.
19. Moue, T. Delta-Sigma Modulator and Signal Processing System. U.S. Patent 8,581,764, 12 November 2013. Available online: https://www.google.com/patents/US8581764.
20. Dias, V.F. Signal processing in the sigma-delta domain. *Microelectron. J.* **1995**, *26*, 543–562.
21. Ng, C.W.; Wong, N.; Ng, T.S. Bit-stream adders and multipliers for tri-level sigma delta modulators. *IEEE Trans. Circuits Syst. Express Briefs* **2007**, *54*, 1082–1086.
22. Kouvaras, N. Operations on delta-modulated signals and their application in the realization of digital filters. *Radio Electron. Eng.* **1978**, *48*, 431–438.
23. Kouvaras, N. Some novel elements for delta-modulated signal processing. *Radio Electron. Eng.* **1981**, *51*, 241–250.
24. Kouvaras, N.; Karakatsanis, J. A technique for a substantial reduction of the quantization noise in the direct processing of delta-modulated signals. *Signal Process.* **1985**, *8*, 107–119.
25. Kouvaras, N. Modular network for direct complete addition of delta-modulated signals with minimum quantization error. *Int. J. Electron.* **1985**, *59*, 587–595.
26. Hidaka, Y.; Fujisaka, H.; Sakamoto, M.; Morisue, M. Piecewise linear operations on sigma-delta modulated signals. In Proceedings of the 9th International Conference on Electronics, Circuits and Systems, Dubrovnik, Croatia, 15–18 September 2002; Volume 3, pp. 983–986.
27. Zrilic, D. Method and Apparatus for Mixed Analog and Digital Processing Of Delta Modulated Pulse Streams Including Digital-to-Analog Conversion of A Digital Input Signal. U.S. Patent 5,349,353, 20 September 1994. Available online: https://www.google.com/patents/US5349353.
28. Circuits and Methods for Functional Processing of Delta Modulated Pulse Density Stream. U.S. Patent 6,285,306, 4 September 2001. Available online: https://www.google.com/patents/US6285306.
29. Horianopoulos, S.; Anastassopoulos, V.; Deliyannis, T. Design technique for hardware reduction in delta modulation fir filters. *Int. J. Electron.* **1991**, *71*, 93–106.
30. Pneumatikakis, A.; Anastassopoulos, V.; Deliyannis, T. Realization of a high-order iir delta sigma filter. *Int. J. Electron.* **1995**, *78*, 1071–1089.
31. O'Leary, P.; Maloberti, F. Bit stream adder for oversampling coded data. *Electron. Lett.* **1990**, *26*, 1708–1709.
32. Liang, Y.; Wang, Z.; Meng, Q.; Guo, X. Design of high speed high snr bit-stream adder based on delta sigma modulation. *Electron. Lett.* **2010**, *46*, 752–753.
33. McGrath, D. Method and Apparatus for Processing Sigma-Delta Modulated Signals. U.S. Patent 5,990,818, 23 November 1999. Available online: https://www.google.com/patents/US5990818.

34. Schreier, R. Delta Sigma Toolbox: High-Level Design and Simulation Of Delta-Sigma Modulators. Matlab Toolbox. Available online: http://www.mathworks.com/matlabcentral/fileexchange/19-delta-sigma-toolbox (accessed on 29 December 2011).
35. *AD7401 Isolated Sigma-Delta Modular*; Technical Report; Analog Device Inc.: Norwood, MA, USA, 2011.
36. *Opal Kelly XEM 3001v2 User's Manual*; Technical Report; Opal Kelly Inc.: Portland, OR, USA, 2004.