

A Single-Bit Pseudo-Parallel Processing Low-Oversampling Delta-Sigma Modulator Suitable for SDR Wireless Transmitters

Safar Hatami, *Member, IEEE*; Mohamed Helaoui, *Member, IEEE*; Fadhel M. Ghannouchi, *Fellow, IEEE*; Massoud Pedram, *Fellow, IEEE*

Abstract—The oversampling requirement in a delta-sigma modulator (DSM) is considered one of the limiting factors toward its employment in today’s high-frequency applications, such as wireless software defined radio (SDR) systems. This paper advances that the critical requirement for DSMs is high-frequency processing and not a high oversampling ratio. A single-bit semi-parallel processing structure to accomplish the high-frequency processing is proposed in this paper. Using the suggested low-oversampling digital DSM architecture, high-speed, high-complexity computations, which are normally required for wireless applications, are executed in parallel. This facilitates the design of embedded SDR multi-standard transmitters using commercially available digital processors. The most favorable application of the proposed single-bit DSM is to build an RF transmitter that includes a one-bit quantifier with two-level switching power amplifier for both high linearity and high efficiency. Performance analysis was carried out by using MATLAB simulations, which showed a reduction of the oversampling ratio by a factor of 16 (for a baseline oversampling ratio of 256) with the same signal to noise (SNR) ratio. The proposed DSM was also implemented on a field-programmable gate array (FPGA) board and its performance was validated by using a code division multiple access (CDMA) signal. Bandwidth of the output signal was increased four times without increasing the processing frequency. Simultaneously, quality of the output signal remained the same but FPGA resource usage was increased by a factor of three.

Index Terms— Delta Sigma Modulation, Parallel Processing, FPGA, Oversampling

I. INTRODUCTION

OVERSAMPLING has become a popular technique for data conversion [1][2]. The outstanding linearity of delta-sigma modulators (DSMs) is the main reason for popularity of these modulators in modern electronic components such as data converters [3], frequency synthesizers [4], and switched-mode power supplies. However, achieving this degree of

linearity comes at the cost of a large oversampling ratio and, therefore, need for high-speed processing. The oversampling requirement in a DSM discourages its employment in today’s compute-intensive applications, such as software defined radio (SDR) systems.

Emerging applications have encouraged designers to develop highly linear converters with large input bandwidths [5][6][7][8]. One approach is through the use of higher order modulators and lower oversampling ratios. The disadvantage of this approach is the instability of high-order DSMs [15].

Several research works have utilized the concept of multi-rate signal processing to reduce the oversampling ratio. A Hadamard transform was used [10] [11] to decompose the input spectrum into several sub-bands, which were then applied to separate DSMs, whose outputs were subsequently recombined. This work used two DSMs per output bit, which is inefficient in terms of the die area when implemented using radio frequency integrated circuits (RFIC) technology.

An area-efficient architecture [12] was developed by combining multiple DSMs in parallel, along with analog preprocessing of the input signal and digital post-processing of the output signals. By using interconnected modulators working in parallel with each running at the same clock, a new Parallel processing DSM (PDSM) was proposed in [13]. A Time Interleaved Sigma-Delta architecture was used in [14] to increase bandwidth of the converter with a lower hardware complexity.

In this paper, an alternative approach, also based on parallel processing, is described. Here, however, multiple DSMs are not used. The proposed PDSM implements combined and simplified processing steps for n sequential clocks of a regular DSM (n closed loop computations.) A PDSM that combines n closed loops generates n bits per clock cycle. In fact the highest sampling frequency of the proposed PDSM is now shifted to one multiplexer, which is the same as the sampling frequency of the traditional single-bit DSM. The other processing element of PDSM work n times slower compared to traditional single-bit DSM.

The favorable application of the proposed PDSM is an RF transmitter which integrates a one-bit quantifier and a two-level switching power amplifier to attain high linearity. By using the proposed low-oversampling DSM, envelope signals in wireless applications, e.g., orthogonal frequency-division

Manuscript received April 02, 2012. The Informatics Circle of Research Excellence (iCORE), the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs (CRC) Program supported this work.

M. Helaoui and F. M. Ghannouchi are with Intelligent RF Radio Laboratory, Department of Electrical and Computer Eng., University of Calgary, 2500 University Drive NW, Calgary, AB, Canada, T2N 1N4. S. Hatami and M. Pedram are with Department of EE-Systems, University of Southern California, EEB-344 3740 McClintock Ave., Los Angeles CA, 90089. (e-mail: shatami@usc.edu, mhelaoui@ucalgary.ca; fghannouchi@ieee.org, Pedram@usc.edu.)

multiplexing (OFDM) and code division multiple access (CDMA), can be modulated to two-level signals. These signals can then be amplified with a switch-mode power amplifier (PA.) Theoretically, the switch-mode power amplifiers are able to obtain 100% power efficiency. Furthermore, two-level signals can ideally be processed without any errors (100% linearity.) Therefore, by combining the two-level DSM and switch-mode PA, it is expected that power efficiency and linearity can be simultaneously achieved.

Performance of the proposed technique has been validated through MATLAB simulations as well as field-programmable gate array (FPGA) implementation using a CDMA signal.

Section II provides a brief review of a regular oversampling DSM. Section III describes the proposed low-oversampling PDSM. Section IV reports the simulation results and discusses the advantages of the PDSM. The implementation and experimental results are presented in section V. The paper is concluded in Section VI.

II. REVIEW OF OVERSAMPLING LOW-PASS DSM

This section reviews low-pass digital DSM theory and provides an example of a third-order digital DSM.

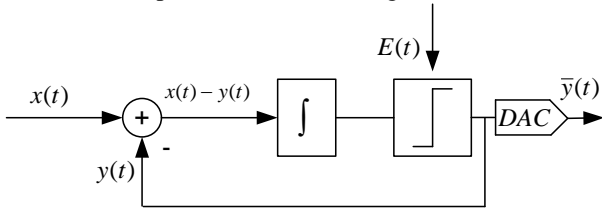


Fig. 1. General structure of a delta-sigma modulator.

The general structure of a DSM is depicted in Fig. 1. The input to the integrator is the difference between the input signal, $x(t)$, and the quantized output value, $y(t)$. The quantization noise is represented by the additive term, $E(t)$. This error is summed in the integrator and then quantized by a two-level quantizer. The output signal, $y(t)$, is held by DAC for a clock period of $T_s = 1/f_s$, which yields $\bar{y}(t)$. The inherent transfer function of the DAC is $h(t)$ and relates $y(t)$, and $\bar{y}(t)$ as follows:

$$\bar{y}(t) = h(t) \otimes y(t) \quad (1)$$

where \otimes denotes the convolution operator.

The output of a DSM is described in the z-domain by:

$$Y(z) = STF(z)X(z) + NTF(z)E(z) \quad (2)$$

where $X(z)$, $Y(z)$ and $E(z)$ represent the z-transforms of $x(t)$, $y(t)$, and $E(t)$ respectively. The signal transfer function, $STF(z)$, is applied to the signal at the desired frequency band whereas the noise transform function, $NTF(z)$, is applied to the quantization noise in order to

suppress it from the desired band.

A z-domain representation of a third-order low-pass (LP) DSM is depicted in Fig. 2. Details about the calculation of the modulator coefficients as well as the seventh order LP DSM architecture can be found in [15].

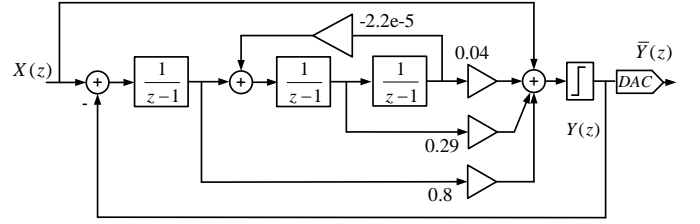


Fig. 2. Z-domain representation of a third-order low-pass DSM.

For this DSM, the signal and noise transfer functions are given in (3) and (4).

$$STF(z) = 1 \quad (3)$$

$$NTF(z) = \frac{(z-1)(z^2 - 2z + 1)}{(z - 0.6694)(z^2 - 1.531z + 0.6639)} \quad (4)$$

The frequency equivalent of the equation (2) is given by (5): $Y(f) = X(f) + NTF(f)E(f)$ (5)

The frequency domain depiction of this equation is illustrated in Fig. 3.

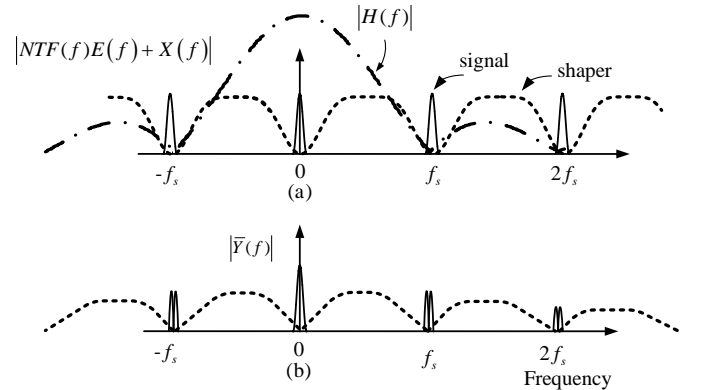


Fig. 3. (a) Signal $X(f)$ (solid), shaped noise $NTF(f)E(f)$ (dashed) and frequency response of the sample and hold $H(f)$ (dot-dashed); (b) sample and hold signal $\bar{Y}(f)$.

Fig. 3(a) shows $|Y(f)|$ and $|H(f)|$ while Fig. 3 (b) shows $|\bar{Y}(f)| = |H(f)Y(f)|$. As shown in Fig. 3(a), the shaped noise, $NTF(f)E(f)$, and signal, $X(f)$, are repeated at the harmonics of f_s . It is evident from Fig. 3(b) that, among all these signal replicas, the only undistorted signal is at zero frequency. All other replicas are distorted.

III. LOW-OVERSAMPLING PDSM

This section explains the idea behind the proposed low-oversampling architecture for generating two-level delta-sigma

output.

For the regular DSM, the sampling frequency of the input signal and clock frequency of the DSM are typically equal (this value is f_s for previous section.) Now, suppose the sampling frequency of the input signal is f_s while the clock frequency of the DSM is f'_s , which may not be equal. Furthermore assume that $f'_s > f_s$ and, for simplicity of analysis, f'_s/f_s is a positive integer value, N ; therefore, after the elapse of N clock cycles, the DSM processes one constant digital input. In next sections, f'_s refers to the PDSM output rate which is equivalent to the PDSM throughput and output multiplexer selection frequency. Also f'_s can be considered as the effective frequency of PDSM (considering parallel processing.) The effective frequency of the PDSM is the alternative name for the sampling frequency for the traditional DSM. The frequency of all processing elements in PDSM is f_s except for the frequency of multiplexer which is f'_s .

TABLE I compares the signals and transfer functions for regular DSM and PDSM. The frequency for the first row is the same for DSM and PDSM. However they have different frequencies for second, third, and fourth rows of the table.

TABLE I
DIFFERENT DELTA SIGMA ARCHITECTURES

Signal/ Transfer	DSM	PDSM
$ X(f) $	Repeated at harmonics of f_s , Fig. 3(a)	Repeated at harmonics of f_s , Fig. 4(a.)
$ E(f) $	Repeated at harmonics of f_s , Fig. 3(a)	Repeated at harmonics of f'_s , Fig. 4(b.)
$ H(f) $	Zero-crossings at kf_s , Fig. 3(a)	Zero-crossings at kf'_s , Fig. 4(c.)
$ NTF(f) $	Repeated at harmonics of f_s , Fig. 3(a)	Repeated at harmonics of f'_s , Fig. 4(b.)
$ \bar{Y}(f) $	Fig. 3(b)	Fig. 4(d.)

Note: k is all nonzero integer values.

Note that the DSM associated with Fig. 4 processes a constant input in N clock cycles (in this DSM, N is 2.) It is evident from Fig. 4 that, as long as f'_s is sufficiently larger than the Nyquist rate of the input signal, $x(t)$, the signal at the baseband is of high quality.

The oversampling ratio formula is given by (6). In (6), BW is the double-sided bandwidth of the signal, and OSR stands for the oversampling ratio.

$$OSR = \frac{f'_s}{BW} \quad (6)$$

Typically, f'_s is eight times greater than the Nyquist rate (.). In contrast, a regular DSM often has an oversampling ratio around 256 in order to generate a good quality signal at the

output of the DSM. Hence, the sampling frequency of the input signal of the DSM is lower by a factor of 32 or more.

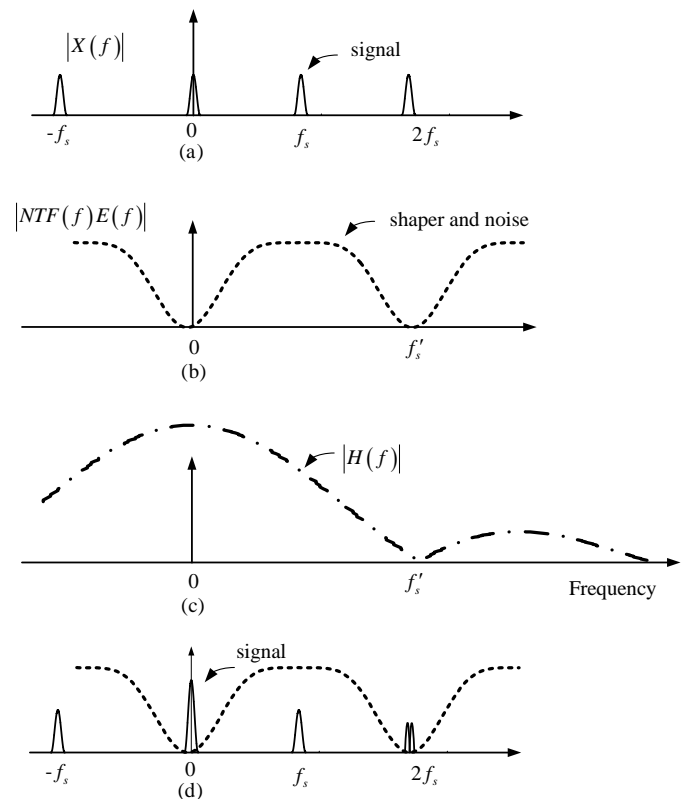


Fig. 4. (a) Signal $X(f)$ (solid); (b) shaped noise $NTF(f)E(f)$ (dashed); (c) frequency response of the sample and hold $H(f)$ (dot-dashed); (d) output signal when sampling frequency f'_s is different from DSM clock frequency f_s (in this figure, $N = f'_s/f_s = 2$).

It is clear that an oversampled input signal is not required for the DSM to produce a high quality output at baseband. However, it is crucial for the DSM to operate at a high frequency, say 256 times the Nyquist rate of the input signal, in order to stretch the quantization noise in a wide frequency range, and thereby, lower its level in the in-band of the useful signal.

The proposed PDSM takes advantage of the fact that the DSM can process constant input samples for N clock cycles. Therefore, a novel architecture that processes N constant samples in parallel by combining N closed loop processing of a regular DSM is presented. The order of the PDSM is the same as the order of the regular DSM that is used in the PDSM. Herein, N will be referred to as the unrolling factor of the PDSM.

In the next sections, a third-order PDSM with $N = 4$ is described; and, finally, the general derivation for the PDSM is provided.

A. Third-order and Four-unrolled PDSM Implementable on FPGA or ASIC Designs

This section proposes the parallel version of a third-order DSM when the unrolling factor, N , is 4. Fig. 5 shows a typical third-order and four-unrolled PDSM architecture. The different components of this architecture are introduced in Figures 7, 9, 10, 11 and 12. The input of delta sigma modulator is sampled with the frequency of f_s . All processing elements are working at clock frequency of f_s . The frequency of multiplexer, which gives the throughput of PDSM, is $f'_s = 4f_s$. In fact the effective frequency of PDSM is $f'_s = 4f_s$ because of parallel processing.

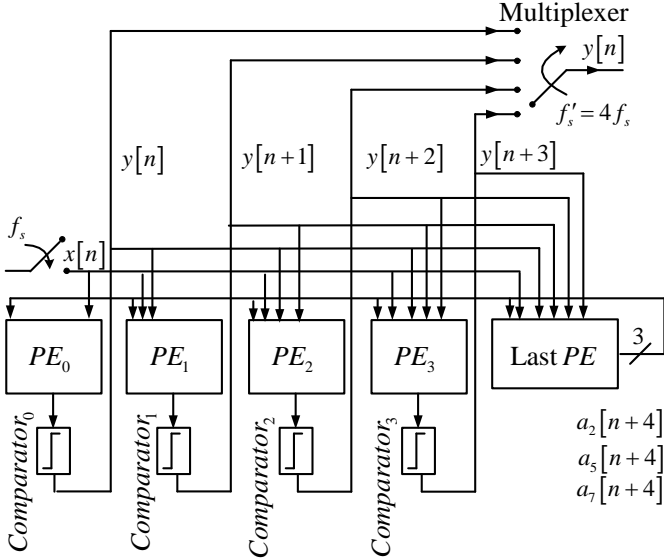


Fig. 5. A typical digital implementation of a third-order and four-unrolled PDSM.

Fig. 6 shows the parametric version of a third-order DSM, which is also shown in Fig. 2 ($v = 2.2e-005$, $p = 0.04$, $q = 0.29$ and $r = 0.8$.)

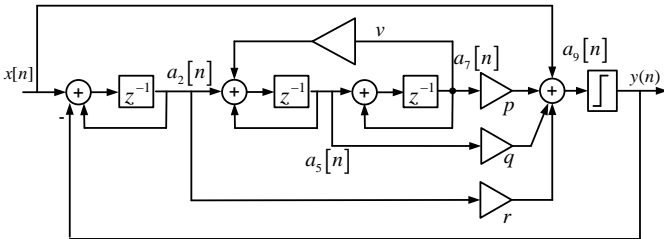


Fig. 6. Representation of a third-order low-pass DSM.

It is assumed that the signals $a_2[n]$, $a_5[n]$, $a_7[n]$, $a_9[n]$, $x[n]$ and $y[n]$ denote the signals at nodes a_2 , a_5 , a_7 , a_9 , x and y , respectively, at time sample n . The signals $x[n]$ and $y[n]$ refer to the input and output signals of the DSM. Since it is assumed that $N = 4$, the input signal is constant for four consequent clock cycles of f'_s , i.e. $x[n] = x[n+1] = x[n+2] = x[n+3]$, where n is a multiple of 4.

The expressions in (7) calculate the values of signals a_2 , a_5 and a_7 at time $n+1$, by using the signal values in the previous time sample, n . The signal value $a_9[n]$ is calculated directly from a_2 , a_5 , a_7 and x at time n . The two-level quantizer, Q , quantizes $a_9[n]$ into -1 or $+1$ at time n . The quantized value is $y[n]$.

$$\begin{aligned}
 a_2[n+1] &= a_2[n] + x[n] - y[n] \\
 a_5[n+1] &= a_2[n] + a_5[n] + va_7[n] \\
 a_7[n+1] &= a_5[n] + a_7[n] \\
 a_9[n] &= ra_2[n] + qa_5[n] + pa_7[n] + x[n] \quad \leftarrow PE_0 \\
 y[n] &= Q(a_9[n]) \quad \leftarrow Cmp_0
 \end{aligned} \tag{7}$$

Two last equations in (7) correspond to two parts of PDSM in Fig. 5: processing elements (PE_0) and the one-bit quantizer (comparator₀).

The expressions in (8) give the signals a_2 , a_5 and a_7 at time $n+2$, by utilizing the signal values at the previous time (previous clock cycle), assuming that the input signal, x , is constant at time n and $n+1$, i.e. $x[n] = x[n+1]$. In order to update the signal values for the next clock cycle of f'_s , the basic expressions in (7) were used, and only the time indexes were increased, as shown in (8).

$$\begin{aligned}
 a_2[n+2] &= a_2[n+1] + x[n+1] - y[n+1] \\
 &= a_2[n] + 2x[n] - y[n] - y[n+1] \\
 a_5[n+2] &= a_5[n+1] + va_7[n+1] + a_2[n+1] \\
 &= 2a_2[n] + (v+1)a_5[n] + 2va_7[n] + x[n] - y[n] \\
 a_7[n+2] &= a_7[n+1] + a_5[n+1] \\
 &= a_2[n] + 2a_5[n] + (v+1)a_7[n]
 \end{aligned} \tag{8}$$

The output signal, y , at time $n+1$ is simply updated as is given in (9).

$$\begin{aligned}
 a_9[n+1] &= pa_7[n+1] + qa_5[n+1] + ra_2[n+1] + x[n+1] \\
 &= \underbrace{(q+r)a_2[n] + (p+q)a_5[n] + (p+vq)a_7[n] + (r+1)x[n]}_{\text{First part}} + \underbrace{(-ry[n])}_{\text{Second part}} \\
 &= r_1a_2[n] + q_1a_5[n] + p_1a_7[n] + s_{11}x[n] + s_{12}y[n] \quad \leftarrow PE_1 \\
 y[n+1] &= Q(a_9[n+1]) \quad \leftarrow Cmp_1
 \end{aligned} \tag{9}$$

where $r_1 = q+r$, $q_1 = p+q$, $p_1 = p+vq$, $s_{11} = r+1$ and $s_{12} = -r$

Two equations in (9) correspond to two parts of PDSM in Fig. 5: processing elements (PE_1) and the one-bit quantizer (comparator₁), which calculate y at time $n+1$. It is clear from (9) that the process of calculating $a_9[n+1]$ can be divided in two parts. The first part is dependent on the signal values of a_2 , a_5 , a_7 and x at time n and can be processed at time n . The second part depends on $y[n]$, which is processed by PE_1 , and its process is started at time n . The second part is a two-level value, and its two possibilities can be pre-calculated and stored in two registers. Once $y[n]$ is ready, the second part is multiplexed from the two pre-calculated values available in the registers. It is noteworthy that the first part is computation intensive whose calculation is started at time n .

The only computation that depends on $y[n]$ is *summation* of the precalculated term $s_{12}y[n]$ and the calculation of the second part of (9). Fig. 7 shows the concept of pseudo-parallel processing for computing $y[n]$ and $y[n+1]$ together. The total delay associated with the parallel calculation is $T_p = t_m + 3t_s + 2t_c$, where t_m , t_s and t_c denote parameterized delays of the multiplier, adder, and comparator, respectively. The total calculation delay for regular DSM, T_r , is $T_r = 2t_m + 5t_s + 2t_c$. If $t_m = 4t_s$, $t_c = 0.2t_s$ then $T_r/T_p = 13.4/7.4 \approx 2$, which nearly provides a factor of two performance improvement for the parallel processing method for calculating $y[n]$ and $y[n+1]$.

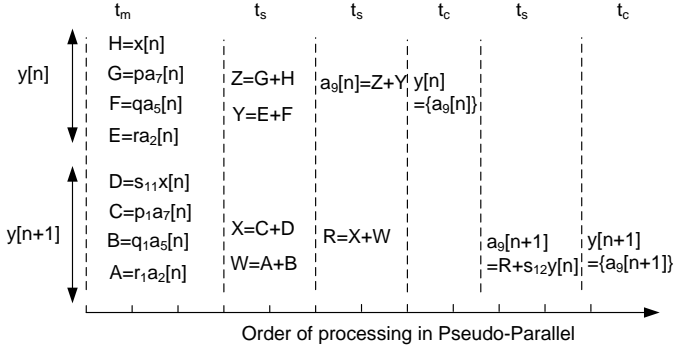


Fig. 7. A typical sequencing diagram for pseudo parallel processing.

In the next step, all signal values given in (8) and (9) are used to update the signal values for cycle time $n+3$. The updated signal values are given through the expressions in (10).

$$\begin{aligned}
 a_2[n+3] &= a_2[n] + 3x[n] - y[n] - y[n+1] - y[n+2] \\
 a_5[n+3] &= (v+3)a_2[n] + (3v+1)a_5[n] \\
 &\quad + (v^2+3v)a_7[n] + 3x[n] - 2y[n] - y[n+1] \\
 a_7[n+3] &= 3a_2[n] + (3+v)a_5[n] \\
 &\quad + (3v+1)a_7[n] + x[n] - y[n] \\
 a_9[n+2] &= r_2a_2[n] + q_2a_5[n] + p_2a_7[n] \\
 &\quad + s_{21}x[n] + s_{22}y[n] + s_{23}y[n+1] \quad \leftarrow PE_2 \\
 y[n+2] &= Q(a_9[n+2]) \quad \leftarrow Cmp_2
 \end{aligned} \tag{10}$$

where $r_2 = p + 2q + r$, $q_2 = 2p + (v+1)q$, $p_2 = (v+1)p + 2vq$ and $s_{21} = q + 2r + 1$, $s_{22} = -q - r$, and $s_{23} = -r$

It is evident from the two last expressions that the output signal at time $n+2$ is obtained from signals at time n and the output signals $y[n]$ and $y[n+1]$. Therefore, in a digital hardware implementation, the processing of signal $a_9[n+2]$ can be started at time n , instead of time $n+2$.

Furthermore, all significant computations for the $a_9[n+2]$ calculation are dependent on the signal values at time n . The only values from times $n+1$ and $n+2$ that contribute in the computing of $a_9[n+2]$ are the two-level values, $y[n]$ and $y[n+1]$, of which the two possibilities of the associated products, $s_{22}y[n]$ and $s_{23}y[n+1]$, can be pre-calculated and stored in two registers. Therefore, once $y[n]$ and $y[n+1]$ are ready, they can be used to evaluate $a_9[n+2]$. Two last

equations in (10) correspond two parts of PDSM in Fig. 5: processing elements (PE_2) and the one-bit quantizer (comparator₂), which processes $y[n+2]$.

Once again, the basic expressions of (7) are used to compute the output signal at time $n+3$, as given in (11).

$$\begin{aligned}
 a_9[n+3] &= r_3a_2[n] + q_3a_5[n] + p_3a_7[n] \quad \leftarrow PE_3 \\
 &\quad + s_{31}x[n] + s_{32}y[n] + s_{33}y[n+1] + s_{34}y[n+2] \tag{11} \\
 y[n+3] &= Q(a_9[n+3]) \quad \leftarrow Cmp_3
 \end{aligned}$$

where $s_{31} = p + 3q + 3r + 1$, $s_{32} = -p - 2q - r$, $s_{33} = -q - r$, $s_{34} = -r$, $p_3 = p(3v+1) + q(v^2+3v)$, $r_3 = 3p + q(v+3) + r$ and $q_3 = p(3+v) + q(3v+1)$.

In the signal derivation given in (11), it is supposed that the input signal at time $n+3$ is equal to the signal at time n ($x[n] = x[n+3]$.) The signal values a_9 and y at time $n+3$ are given in the two last expressions of (11). The evaluation process is started at time n and finished when the two-level values, $s_{32}y[n]$, $s_{33}y[n+1]$ and $s_{34}y[n+2]$, are available. Two equations in (11) correspond to two parts of PDSM in Fig. 5: processing elements (PE_3) and the one-bit quantizer (comparator₃), which compute $y[n+3]$.

In conclusion, the process of calculating four sequential outputs of PDSM can be started at the same time and accomplished in one clock cycle of f_s' . It is evident from (7) through (11) that the path delays of the four sequential outputs are in the same order as the regular DSM, as given in (7).

Figures 6, 7, 9 and 10 display how to calculate the four sequential outputs of PDSM. However, signals a_2 , a_5 and a_7 are computed through (12), which is to be used for the next four cycles.

$$\begin{aligned}
 a_2[n+4] &= a_2[n] + 4x[n] - y[n] - y[n+1] - y[n+2] - y[n+3] \\
 a_5[n+4] &= (4v+4)a_2[n] + (v^2+6v+1)a_5[n] + \\
 &\quad (4v^2+4v)a_7[n] + (6+v)x[n] + (-3-v)y[n] - 2y[n+1] - y[n+2] \\
 a_7[n+4] &= (v+6)a_2[n] + (4v+4)a_5[n] + (v^2+6v+1)a_7[n] \\
 &\quad + 4x[n] - 3y[n] - y[n+1]
 \end{aligned} \tag{12}$$

The equations are simply driven by updating the basic expression of (7) for time $n+4$ and utilizing signal values from (8) to (11). A typical implementation of (12) is depicted in Fig. 8. This hardware is referred to as the last processor element (last PE) which is part of Fig. 5.

Fig. 5 which shows block diagram of a third-order and four-unrolled PDSM architecture contains PE_0 , PE_1 , PE_2 , PE_3 and last PE . The result of an FPGA (field-programmable gate array) implementation of the PDSM shown in Fig. 5 is given in next section.

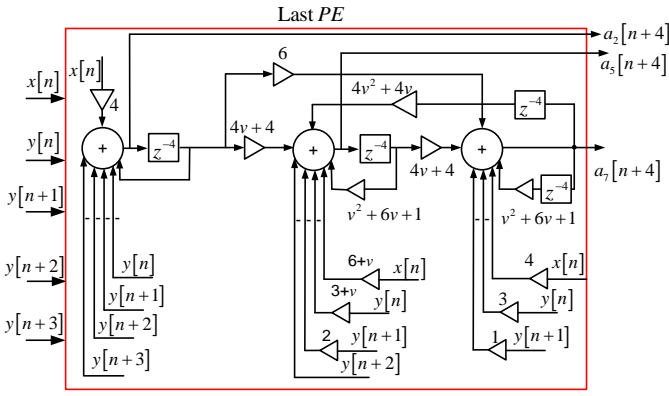


Fig. 8. Digital implementation of equation (12): last processing element of a third-order and four-unrolled PDSM.

B. n^{th} -Order and N -unrolled PDSM

This section proposes the general formulation for the PDSM. Suppose that the digital input sequence is x , where $x(i)$ is the i^{th} element of this sequence. The variable y is the two-level output of DSM, and y_a is its output before quantization to two levels, -1 and 1. The array $[m]_{n \times 1}$ indicates the values of registers in the DSM, where n is the order of the DSM. The matrices A , B and C describe the coefficients of the DSM.

The expressions in (13) present an n^{th} -order DSM, when the i^{th} input is fed to the modulator [15]. A gives the feedback values, whereas B describes the coefficients from the input and output to the registers. The output value is calculated from the input and the register values by using the coefficient matrix C .

$$[m(i+1)]_{n \times 1} = [A]_{n \times n} \times [m(i)]_{n \times 1} + [B]_{n \times 2} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}_{2 \times 1} \quad (13)$$

$$y_a(i) = [C]_{1 \times n} [m(i)]_{n \times 1} + x(i)$$

$$\text{where } y = Q(y_a) = \begin{cases} 1 & y_a > 0 \\ -1 & y_a < 0 \end{cases}$$

Let us assume that $f_s/f_s = N$, meaning that the input of the modulator is constant for each N clock cycle of f_s . We want to calculate the feedback values for clock cycle $N+1$ and all output values from the first clock cycle to the N^{th} clock cycle of f_s . In the PDSM structure, the output calculations for all N sequential outputs are started at the same time and carried out in one clock cycle of f_s .

First clock cycle of f_s

The following expressions describe signals for the first clock cycle of f_s .

$$[m(1)] = [A][m(0)] + [B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \quad (14)$$

$$y_a(0) = [C][m(0)] + x(0)$$

Second clock cycle of f_s

Assuming $x(0) = x(1)$, (13) and (14) are used to calculate signals for the second clock cycle of f_s , as given in (15):

$$x(1) = x(0)$$

$$\begin{aligned} [m(2)] &= [A][m(1)] + [B] \begin{bmatrix} x(1) \\ y(1) \end{bmatrix} \\ &= [A] \left([A][m(0)] + [B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \right) + [B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} \\ &= [A]^2 [m(0)] + [A][B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + [B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} \end{aligned} \quad (15)$$

$$\begin{aligned} y_a(1) &= [C] \left([A][m(0)] + [B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \right) + x(1) \\ &= [C][A][m(0)] + [C][B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + x(0) \end{aligned}$$

Third clock cycle of f_s

For the third clock cycle of f_s , the signal values can be calculated by using (13) to (15), as given in (16):

$$\begin{aligned} x(2) &= x(0) \\ [m(3)] &= [A][m(2)] + [B] \begin{bmatrix} x(2) \\ y(2) \end{bmatrix} \\ &= [A] \left([A]^2 [m(0)] + [A][B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + [B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} \right) + [B] \begin{bmatrix} x(0) \\ y(2) \end{bmatrix} \\ &= [A]^3 [m(0)] + [A]^2 [B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + [A][B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} + [B] \begin{bmatrix} x(0) \\ y(2) \end{bmatrix} \quad (16) \\ y_a(2) &= [C][m(2)] + x(2) \\ &= [C] \left([A]^2 [m(0)] + [A][B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + [B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} \right) + x(0) \\ &= [C][A]^2 [m(0)] + [C][A][B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + [C][B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} + x(0) \end{aligned}$$

Therefore, $y_a(N)$ and $m(N)$, the output and feedback values at the N^{th} clock period of f_s , respectively, can be expressed as in (17).

$$\begin{aligned} [m(N)] &= [A]^N [m(0)] + [A]^{N-1} [B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + \\ & [A]^{N-2} [B] \begin{bmatrix} x(0) \\ y(1) \end{bmatrix} + \dots + [A][B] \begin{bmatrix} x(0) \\ y(N-2) \end{bmatrix} + [B] \begin{bmatrix} x(0) \\ y(N-1) \end{bmatrix} \\ y_a(N) &= [C][A]^N [m(0)] + [C][A]^{N-1} [B] \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \\ & + \dots + [C][A][B] \begin{bmatrix} x(0) \\ y(N-2) \end{bmatrix} + [C][B] \begin{bmatrix} x(0) \\ y(N-1) \end{bmatrix} + x(0) \end{aligned} \quad (17)$$

The signals that should be calculated in the PDSM for N clock cycles of f_s are the output signal values, $y_a(i)$ ($i = 1, \dots, N$), and the feedback values at the N^{th} clock cycle of f_s , $m(N)$, which is used for cycle $N+1$. Considering the two last

equations, it is evident that $y_a(i)$ and $m(N)$ can be rewritten as given in (18), where w_i , b_i , ε_i and d_i are calculated from (17).

$$\begin{aligned}
 [m(N)] &= \underbrace{\omega_1 m_1(0) + \omega_2 m_2(0) + \dots + \omega_n m_n(0) + ax(0)}_{\text{first expression}} \\
 &\quad + \underbrace{(b_1 y(0) + b_2 y(1) + \dots + b_N y(N-1))}_{\text{second expression}} \\
 y_a(i) &= \underbrace{\varepsilon_1 m_1(0) + \varepsilon_2 m_2(0) + \dots + \varepsilon_n m_n(0) + cx(0)}_{\text{first expression}} \\
 &\quad + \underbrace{(d_1 y(0) + d_2 y(1) + \dots + d_{i-1} y(i-1))}_{\text{second expression}}
 \end{aligned} \tag{18}$$

The calculations of y_a and the last feedback values $[m(N)]$ are divided each into two expressions. The first expression only depends on the feedback values, $m(0)$, which can be processed in one clock cycle of f_s . Hence, in order to evaluate N sequential outputs of the DSM (N bits), for each bit, there are n multiplications for the feedback coefficients and one multiplication for the inputs that can be calculated in parallel. The results of the $n+1$ multiplications must be added by using n adders. The same situation is valid for the computation of the last feedback values.

In the second expression, $y(i)$ is a two-level value, so its multiplication is simple. The results for the two expressions must finally be added together. Notice that the calculation of $y_a(i)$ requires $y(i-1)$ from the last y_a calculation. Fig. 9 depicts a hardware implementation of a PDSM, based on (14) to (18). The architecture is an extension of the third-order and four-unrolled PDSM shown in Fig. 5. It shows that N processor elements calculate N outputs in parallel. One processor calculates the states of the registers for the cycle $N+1$. The frequency of input sampling and for processing elements is f_s . The PDSM output rate, which is equivalent to the PDSM throughput and output multiplexer selection frequency, is f'_s, f'_s can be called the effective frequency of PDSM, which considers parallel processing.

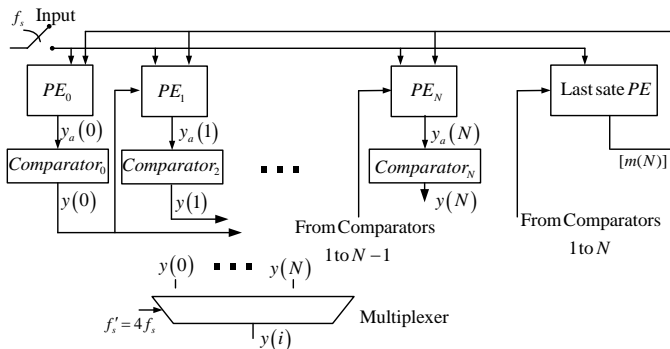


Fig. 9. Typical implementation of PDSM.

IV. SIMULATION RESULTS

This section gives simulation results for the proposed PDSM and compares these results with the conventional DSM. A hardware implementation of the DSM is presented in this

section. The suggested architecture is implementable with today's digital CMOS (complementary metal-oxide-semiconductor) technology and can be utilized in radio frequency (RF) wireless applications.

A. Simulation Results For Low-Pass DSM

The criterion for comparison is the signal-to-noise ratio (SNR) shown in (19), where MS is the mean square. SNR is defined as the ratio of the in-band signal power to the in-band and out-of-band noise power of modulated signal.

$$SNR = 10 \log \left(\frac{MS(\text{Signal})}{MS(\text{Noise})} \right) \tag{19}$$

TABLE II
SIMULATED SNR OF DS TRANSMITTERS WITH CDMA SIGNAL FOR DIFFERENT DSM ORDERS AND UNROLLING FACTOR VALUES

Exp.	Type	Order	SNR (dB)	OSR	N	f_s =Clock freq. (MHz)
1	DSM	2	54.7	512	1	2.097
2			37.8	256		2.097
3			22.7	128		2.097
4			9.7	64		2.097
5			2.0	32		2.097
6	PDSM	2	54.7	512	1	2.097
7			54.5	256	2	1.048
8			54.6	128	4	0.524
9			55.0	64	8	0.262
10			55.2	32	16	0.131
11	DSM	3	64.7	512	1	2.097
12			44.9	256		2.097
13			23.3	128		2.097
14			7.8	64		2.097
15			1.4	32		2.097
16	PDSM	3	66.7	512	1	2.097
17			66.4	256	2	1.048
18			67.0	128	4	0.524
19			67.7	64	8	0.262
20			67.6	32	16	0.131
21	DSM	5	65.9	512	1	2.097
22			40.0	256		2.097
23			8.2	128		2.097
24			4.2	64		2.097
25			1.1	32		2.097
26	PDSM	5	67.9	512	1	2.097
27			67.3	256	2	1.048
28			67.3	128	4	0.524
29			68.1	64	8	0.262
30			68.7	32	16	0.131

A PDSM and regular DSM have been implemented in MATLAB for first- to seventh-order DSMs. Simulations using a CDMA modulated signal were carried out and the results are shown in TABLE II for three different DSM orders for both the regular DSM and the proposed PDSM. N is the unrolling factor of the PDSM. The frequency column is the DSM and PDSM processing element clock frequency of processing, f_s . The multiplexer selection frequency is $f'_s = 5.12$ MHz and the

frequency bandwidth is 2.048 kHz. It is noteworthy that the frequency of processing for PDSM changes from 0.131 MHz to 2.097 MHz but its throughput is 2.097 MHz. For SNR calculation given in (19), the single sided bandwidth for in-band signal and out-of-band noise are 20kHz. For example, the table reports that the SNR of modulated signal for the second order DSM is 64.7 dB and reduces about 15dB for each folding of OSR (experiments 11 to 15). It also shows that SNR is about 65 dB for second order PDSM with $N \times OSR = 512$ for $N=2, 4, 8, 16$ (experiments 16 to 20). Similar results are included in TABLE II for the second, third and fifth order DSM and PDSM. Fig. 10 (a) and Fig. 10 (b) show the spectrum of the modulated CDMA signals at the output of the DSM for experiments 11 and 20, respectively. While PDSM allowed reducing the sampling frequency by 16 times by using parallel processing, the SNR remained at about the same level – 64.7 dB for DSM against 67.6 dB for PDSM.

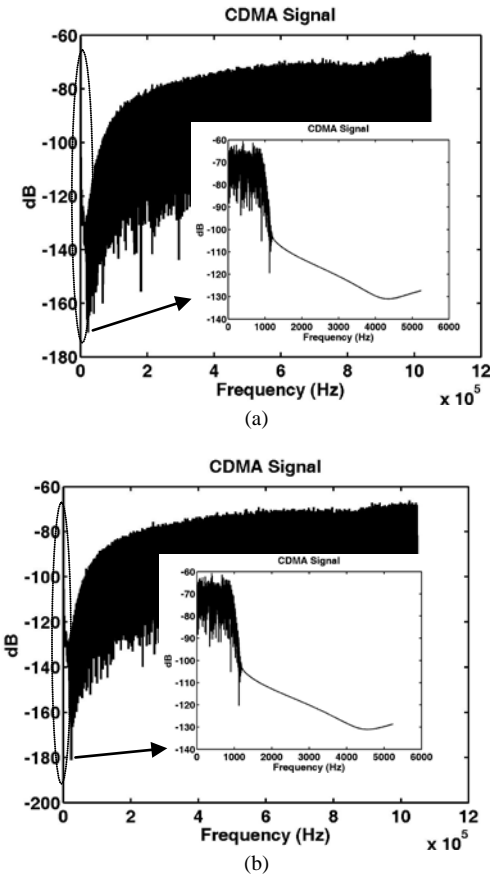


Fig. 10. Spectrum of 3th-order DSMs for a CDMA input signal: (a) DSM: $f'_s = 2.097$ MHz, OSR = 512, SNR = 64.7 dB; (b) PDSM: $f'_s = 2.097$ MHz and $f_s = 0.131$ MHz, OSR = 32, SNR = 67.6 dB and unrolling factor of 16.

V. EXPERIMENTAL VALIDATION USING DS TRANSMITTER

A GHz PDSM based transmitter was developed, prototyped and used to validate the approach proposed in this paper. Fig. 11 shows the block diagram of the demonstrator. The PDSM transmitter is implemented in two blocks. The baseband signal

processing part is implemented using a FPGA block. The modulation and up-conversion is implemented using a high-speed dedicated logic stage.

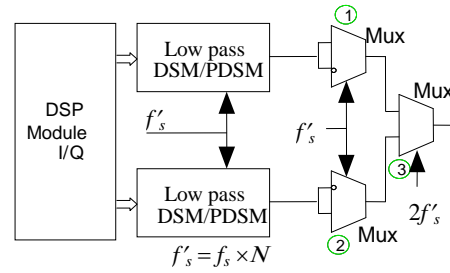


Fig. 11. Block diagram delta-sigma based transmitter.

The two third-order DSMs and PDSMs shown in Fig. 6 and Fig. 5 were implemented on a Stratix II EP2S60 DSP development board [16] and tested with a CDMA signal. The baseband in-phase (I) and quadrature (Q) signals were read from two on-board memories and fed through the low-pass DSMs/PDSMs. Three multiplexers were used for up-conversion and in- I/Q modulation at carrier frequency. The binary RF output signal was fed to a vector signal analyzer (VSA), which was used to capture, filter and analyze the signal.

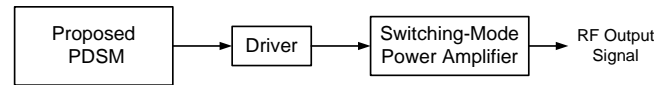


Fig. 12. Block diagram of setup to test PDSM-based transmitter

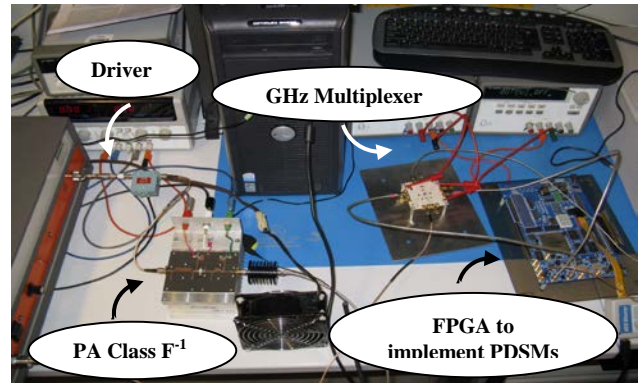


Fig. 13. Setup used to test PDSM-based transmitter

The main advantage of PDSM is to achieve higher SNR output signal using lower processing frequency compared to a regular DSM. One of the most favorable applications of the proposed single-bit PDSM is to make an RF transmitter which includes a one-bit quantizer delta sigma and two-level switching power amplifier, which results in a high efficiency and high linear transmitter. A two-level switching Power Amplifier Class D, E, F, F⁻¹ or S can be driven with the two-level output of PDSM [1][2][5][6]. Fig. 12 shows a block diagram of the setup used to evaluate the performance of the PDSM-based transmitter. Fig. 13 shows a photo of the

measurement setup for a prototype DSM/PDSM based RF transmitter.

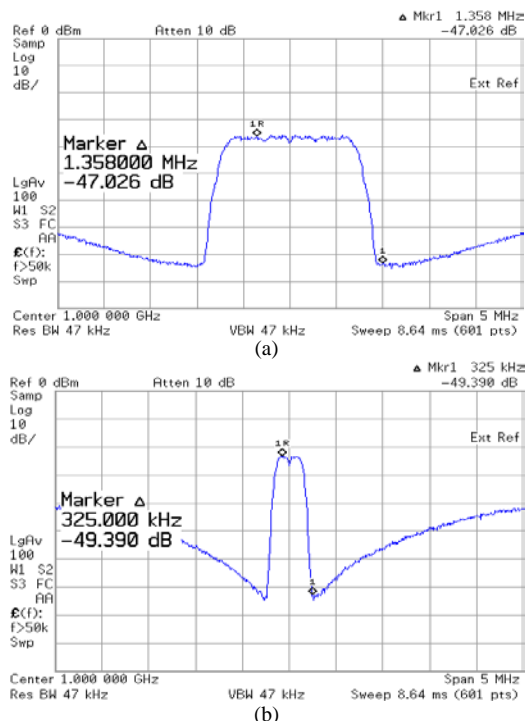


Fig. 14. Spectrum of the output signal (the signal BW for the PDSM is four times the signal BW for the DSM): (a) 3rd-order PDSM with an unrolling factor of 4; (b) 3rd-order DSM.

The unrolling factor of the implemented PDSM was selected to be four. The PDSM and DSM were fed by CDMA signals with bandwidths of 1600 kHz and 400 kHz, respectively. The clock frequencies (sampling frequency) of the DSM and PDSM were 25 MHz. As shown in Fig. 14, with the help of parallel processing, the PDSM allows for an increase of the modulation bandwidth by a factor of 4 compared to DSM, while maintaining a comparable noise shaping performance. In fact, the SNRs of the output signals for both cases were approximately the same level, 49 dB for DSM and 47 dB for PDSM, and are given in TABLE III.

TABLE III

SNR COMPARISON OF THE THIRD-ORDER DSM AND PDSM (THE SIGNAL BW FOR PDSM IS FOUR TIMES THE SIGNAL BW FOR DSM)

Structure	Processing Clock	SNR	BW
DSM	25 MHz	49 dB	400 kHz
PDSM	25 MHz	47 dB	1600 kHz

Area and Power:

TABLE IV shows the evaluation of the resources occupied in the FPGA, in terms of number of logic cells for gates, register and arithmetic logic units (ALUs.) The improvement in performances in the PDSM architecture ($N = 4$) comes with an increase in the resources required for implementation. It is shown that the resources are increased about three times for $N=4$. In general, based on the methodology presented in

Section III.A, the architecture of a PDSM with unrolling factor N is obtained by unrolling structures of N regular DSM. As shown in equations (7) to (12) the summation and multiplication operations are simplified and optimized. Therefore hardware of a PDSM with unrolling factor N is smaller than $N \times A_{\Delta\Sigma}$, where $A_{\Delta\Sigma}$ refers to the area of regular DSM with same order of noise shaping. The FPGA area for $N=4$ and 3rd-order PDSM as reported in TABLE IV is approximately $(3/4) \times N \times A_{\Delta\Sigma}$.

The power consumption of the development board and multiplexer is of the order of 100 mW. This includes PDSM components and other unused components on FPGA development board. The power consumption of power amplifier in a PDSM based transmitter is of the order of 10 Watt. Therefore power consumption of PDSM is negligible compared to total power consumption of the transmitter.

TABLE IV

RESOURCE UTILIZATION OF THE THIRD-ORDER DSM AND PDSM ($N=4$)

Structure	Other Logic Cells	Logic Cell for Registers	Logic Cell for ALU
DSM	95	376	334
PDSM	62	1246	970

Comparison:

TABLE V compares different single-bit parallel processing DSM structures. For comparison, it is assumed the order of different delta sigma architectures is the same. References [12] and [13] implemented PDSM using analog circuits and references [10] and [11] only reported simulation results. Since each design was implemented on different technology, design areas are compared parametrically. The power consumption values are not available for every referenced design. The first row of TABLE V considers a regular DSM with processing frequency f_s' and area $A_{\Delta\Sigma}$. The throughput of different delta sigma architectures is f_s' . However processing frequencies of different designs are different which are reported in third column. Gain of SNR for each design compared to regular DSM is reported in fourth column. The structure proposed in [10], [11] and [12] have larger area compared to the proposed PDSM. These designs need FIR filters and Hadamard modulators which make PDSM design more complicated. Design of [13] has larger area compared to the proposed PDSM but it perform 20 db better than regular DSM. Using architecture proposed in [14] the input signal is decimated by a factor of N and is processed through N parallel channels. Input signal of each channel is interpolated by a factor of M . This architecture results in a scalable scheme. Although the unrolling methodology proposed in III.A is general, the coefficient calculation is different for different unrolling factors and different DSM orders. This makes the proposed PDSM less scalable compared to other referenced designs.

TABLE V
DIFFERENT DELTA SIGMA ARCHITECTURES

Structure	Area	Proc. Freq.	PDSM _{SNR} -DSM _{SNR} (db)	Complexity
DSM	$A_{\Delta\Sigma}$	f'_s	0	low
PDSM [10]	$N \times (A_{\Delta\Sigma} + 7\text{Mul} + 6\text{Add} + 3\text{Del})$	f'_s/N	< -4	high
PDSM [11] [12]	$N \times (A_{\Delta\Sigma} + 1\text{Mul} + 5.5\text{Add} + 20\text{Del})$	f'_s/N	0	high
PDSM [13]	$\approx N^2 \times A_{\Delta\Sigma}$	f'_s/N	20	high
PDSM [14]	$N \times (A_{\Delta\Sigma} + \text{Mul} + \text{Demux})$	Mf'_s/N	-	middle
Proposed PDSM	$< N \times A_{\Delta\Sigma}$ $\approx (3/4) \times N \times A_{\Delta\Sigma}$ (for PDSM in Fig. 5)	f'_s/N	2~22	middle

It is worthwhile to mention that there are *multi-bit* DSM structures which lower required processing frequency. However the focus of this paper is *single-bit* DSM which are most applicable in RF transmitter with one-bit quantizer and two-level switching mode power amplifier. For example multi-bit quantizer delta sigma is a structure that ensures linearity with a lower processing frequency and a lower OSR value compared to regular DSM. The multi-stage noise shaping (MASH) structure is also an alternative delta sigma structure which is simple for implementation and it is unconditionally stable [15].

VI. CONCLUSION

A new DSM architecture has been introduced. This structure performs delta-sigma modulation with a smaller oversampling rate. The proposed architecture uses the concept of parallel processing to achieve the effect of oversampling without the need for a high sampling frequency. The analysis presented is general and is applicable for LP and band-pass DSMs. The proposed structure has been validated through MATLAB simulation. Simulation results show that for a DSM with OSR = 256, the proposed structure is able to fold the required OSR 16 times while maintaining the same signal to noise (SNR) ratio. A 1 GHz carrier frequency transmitter with a CDMA signal was implemented on FPGA using pseudo-parallel processing low-oversampling DSM and regular DSM. The proposed architecture was able to increase the bandwidth of the output signal four times without increasing the processing frequency while producing the same quality of output signal.

REFERENCES

- [1] Y. Wang, "A class-S RF amplifier architecture with envelope delta-sigma modulation," *IEEE Radio and Wireless Conference*, pp. 177-179, 2002.
- [2] F. M. Ghannouchi, S. Hatami, P. Aflaki, M. Helaoui, and F. M. Ghannouchi, "Multistandard GHz Wireless RF Transmitter Using a Delta-Sigma Modulator and Switch-Mode Power Amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 11, pp. 2811-2819, Nov. 2010.
- [3] X. Wu, V. A. Chouliaras, J. L. Nunez and R. M. Goodall, "A novel $\Delta\Sigma$ control system processor and its VLSI implementation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 3, March 2008, pp. 217-228.
- [4] D. Yang, F. F. Dai, W. Ni, Y. Shi, and R. C. Jaeger, "Delta-Sigma modulation for direct digital frequency synthesis," *IEEE Transactions Very Large Scale Integrated (VLSI) Systems*, vol. 17, no. 6, pp. 793-802, Jun. 2009.
- [5] M. Helaoui, S. Hatami, R. Negra, and F.M. Ghannouchi, "A Novel Architecture of Delta-Sigma Modulator Enabling All-Digital Multiband Multistandard RF Transmitters Design," *IEEE Transactions Circuits and Systems II*, vol. 55, no. 11, pp. 1129-1133, Nov. 2008.
- [6] S. Hatami, M. Helaoui, R. Negra, and F.M. Ghannouchi, "Multiband Multistandard Delta-Sigma-based RF Transmitters," *Software Defined Radio Technical Conference (SDR'07 Tech Conf)*, Denver, CO, Nov. 2007.
- [7] J.S. Keyzer, J.M. Hinrichs, A.G. Metzger, M. Iwamoto, I. Galton, and P.M. Asbeck, "Digital generation of RF signals for wireless communications with band-pass delta-sigma modulation," *IEEE MTT-S International Microwave Symposium Digest*, vol. 3, pp. 2127-2130, 2001.
- [8] J. Rode, J. Hinrichs, and P. Asbeck, "Transmitter architecture using digital generation of RF signals," *IEEE Radio and Wireless Conference*, 2003, pp. 245-248.
- [9] R. Schreier and G.C. Temes, *Understanding Delta-Sigma Data Converters*, IEEE Press, Piscataway NJ, 2005.
- [10] I. Galton, H.T. Jensen, "Delta-Sigma modulator based A/D conversion without oversampling," *IEEE Transactions Circuits and Systems II*, vol. 42, no. 12, 1995.
- [11] I. Galton and H.T. Jensen, "Oversampling parallel delta-sigma modulator A/D conversion," *IEEE Transactions Circuits and Systems II*, vol. 43, no. 12, 1996.
- [12] E.T. King, A. Eshraghi, I. Galton, T.S. Fiez, "A Nyquist-Rate Delta-Sigma A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 1, 1998.
- [13] R. Khoini-poorfard, E.B. Lim, D.A. Johns, "Time-interleaved oversampling A/D converters: theory and practice," *IEEE Transactions Circuits and Systems II*, vol. 44, no. 8, pp. 634, 1997.
- [14] Ch. Jabbour, D. Camarero, V. T. Nguyen, P. Loumeau, "A 1 V 65 nm CMOS Reconfigurable Time Interleaved High Pass Sigma Delta ADC," *ISCAS*, pp. 1557-1560, 2009.
- [15] S.R. Norsworthy, R. Schreier and G.C. Temes, *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation*, 3rd edition, IEEE Press, Piscataway NJ, 1997.
- [16] Altera Corporation. *Altera Stratix II EP2S60 DSP Development Board*, San Jose, CA, USA. (May 2005) [Online]. Available: http://www.altera.com.cn/literature/ds/ds_stratixII_dsp_dev_board.pdf