A Circuit Design for Compact Sigma-Delta Domain Multiplier

Tsubasa Katao, Keita Hayashi, Hisato Fujisaka, Takeshi Kamio and Kazuhisa Haeiwa

Faculty of Information Sciences, Hiroshima City University
3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan
Email: fujisaka@im.hiroshima-cu.ac.jp

Abstract—In this paper we consider reducing logic gates used to build Sigma-Delta (SD) domain multipliers. The reduction brings not only compactness and low cost to every SD domain processors but also high precision even to middle scale processors. In inherently noisy SD domain, it is very important to attain highly precise multiplication. Ordinary multiplication procedure in SD domain is summation of sub-products. There exist integers that products of two integers $\in\{1, 2, \cdots, k\}$ never take in $\{1, 2, \cdots, k^2\}$. By decreasing the redundancy, it is possible to design SD domain multipliers with small adders for summing sub-products. The designed multipliers can be built of about a half of logic gates required to build the preceding multipliers.

1. Introduction

Sigma-Delta (SD) modulation is a popular technique for narrow band analog-to-digital and digital-to-analog conversions [1]. Recently direct processing of SD modulated signals has been proposed [2], which we call SD domain signal processing. The processing scheme has the following advantages:

(i) Decimation filters to convert SD modulated signals into Nyquist-rate multi-bit signals are not necessary.

(ii) Circuits operating on SD modulated signals are simple and connected to one another by a few signal lines.

(iii) Precision of processing increases in the polynomial order of oversampling ratio (OSR).

However, multipliers proposed for SD domain processing do not possess the properties (ii) and (iii). The multipliers are not always small in circuit scale. Their precision increases in proportion to OSR like Nyquist-rate arithmetic circuits.

In this paper we consider reducing logic gates used to build Sigma-Delta (SD) domain multipliers. The reduction brings not only compactness and low cost to every SD domain processors but also high precision even to middle scale processors. In inherently noisy SD domain, it is very important to accomplish highly precise multiplication. Ordinary multiplication procedure in SD domain is summation of sub-products There exist integers that products of two integers $\in\{1, 2, \cdots, k\}$ never take in $\{1, 2, \cdots, k^2\}$. By decreasing the redundancy, it seems possible to design SD domain multipliers with small adders for summing sub-products. This paper presents an attempt to reduce logic gates of the multipliers according to this strategy.

In this paper SD modulated signals to be multiplied are assumed to be outputs of first-order SD modulators with a one-bit quantizer.

2. Sub-multiplier

We first review the preceding SD domain multipliers [2]. Let two SD modulated signals be denoted by $x(n)$ and $y(n)\in\{-1, +1\}$, n: time index. Direct multiplication of $x(n)$ and $y(n)$ spreads their high frequency SD modulation noise over entire frequency band from DC to $f_s/2$, f_s: sampling frequency. Thus, signal components of product $x(n)y(n)$ are contaminated by the spread noise. This problem is solved by eliminating SD modulation noise before multiplication, that is by computing the following expression:

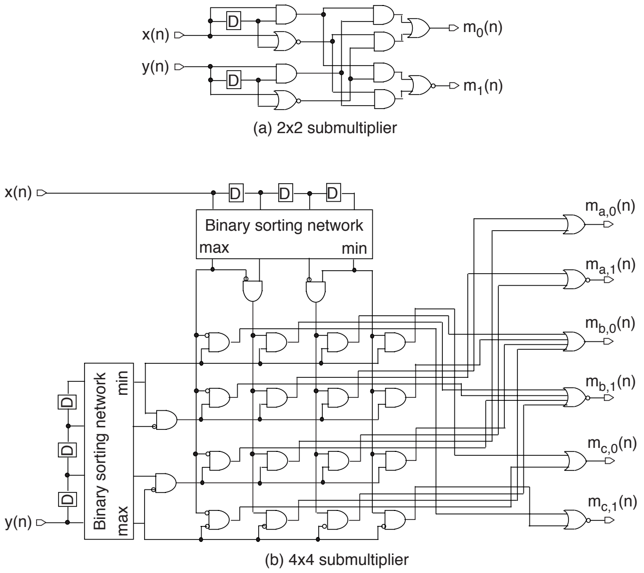
$$\left(\sum_{i=0}^{N-1} x(n-i)\right)\left(\sum_{j=0}^{N-1} y(n-j)\right) \tag{1}$$

However, a multi-bit multiplier is necessary to compute the expression. Let product-of-sums (1) be expanded to sum-of-products (2).

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(n-i)y(n-j) \tag{2}$$

Exclusive-OR (E-OR) gates compute sub-products $x(n-i)y(n-j)$. Sigma-Delta domain adders [2] sum up the sub-products. Although no multi-bit multiplier is necessary to compute expression (2), spread modulation noise of sub-products is not much attenuated if N in the expression is small. If N is large, a large number of logic gates are required to build a SD domain multiplier which operates according to expression (2). Reducing logic gates of the SD domain multiplier is very important to attain highly precise multiplication.

We draw up a strategy for the reduction. The SD domain adders for summing the sub-products occupy large part of the SD domain multiplier. Thus, it is



(a) 2x2 submultiplier

(b) 4x4 submultiplier

Figure 1: 2x2 and 4x4 sub-multipliers.

effective to reduce the adders in circuit scale reduction of the multipliers. In order to reduce the adders it is necessary to decrease the number of sub-products and express the sub-products by small number of bits. Expression (1) can be transformed to

$$\sum_{r=0}^{N/M-1} \sum_{s=0}^{N/M-1} \left(\sum_{i=rM}^{(r+1)M-1} x(n-i) \right) \left(\sum_{j=sM}^{(s+1)M-1} y(n-j) \right) \tag{3}$$

Although partial sums $\sum_i x(n-i) (\equiv S_r)$ and $\sum_j y(n-j)(\equiv S_s)$ are $(M+1)$-level signals, their product does not take all of $(M+1)^2$ levels. When $M=2$, S_r, $S_s \in \{-2, 0, +2\}$. Then, $S_r S_s \in \{-4, 0, +4\}$. Thus, the sub-products $S_r S_s$ can be expressed by only two bits m_1, $m_0 \in \{-1, +1\}$ as

$$S_r S_s = 2(m_0 + m_1) \tag{4}$$

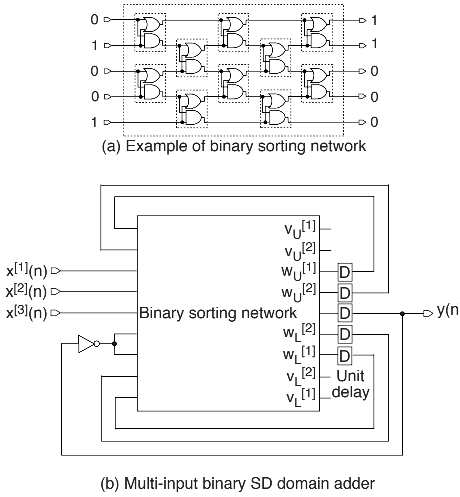
When $M=4$, partial sums are S_r, $S_s \in \{-4, -2, 0, +2, +4\}$. Then, sub-product is $S_r S_s \in \{-16, -8, -4, 0, +4, +8, +16\}$. The sub-products $S_r S_s$ can be expressed, for example, by 6 bits $m_{a,1}$, $m_{a,0}$, $m_{b,1}$, $m_{b,0}$, $m_{c,1}$, $m_{c,0}$ as

$$S_r S_s = 2(m_{a,1}+m_{a,0})+4(m_{b,1}+m_{b,0})+8(m_{c,1}+m_{c,0}) \tag{5}$$

Figure 1 shows sub-multipliers to compute sub-products $S_r S_s$.

3. SD Domain Adder

In this section we introduce SD domain adders based on binary sorting networks [4]. Figure 2(a) shows a binary sorting network. The network transfers "1"s in its input set to upper output terminals and "0"s to lower output terminals. Hereafter we assume that logic values "0" and "1" represent integers −1, +1.



(a) Example of binary sorting network

(b) Multi-input binary SD domain adder

Figure 2: L-input binary SD domain adder.

A SD domain adder which outputs sum of L binary inputs $x_i(n) \in \{-1, +1\}$, i=1, 2, $\cdots$, L, in binary SD modulated signal form is built by using the sorting network as shown in Fig. 2. Outputs $W_U^{[i]}$ and $W_L^{[i]}$, i=1, 2, $\cdots$, $L-1$, of the sorting network are connected to it inputs. Center output $y(n)$ is reversed and fed back to $L-1$ inputs. If the sorting network has $4L-3$ inputs, outputs $V_U^{[i]}$ and $V_U^{[i]}$, $i = 1, 2, \cdots, L-1$, always take +1 and −1 respectively. In this case the sum of all the outputs

$$\begin{aligned} u(n) \quad &= \quad \sum_{i=1}^{L-1} V_U^{[i]}(n) + \sum_{i=1}^{L-1} W_U^{[i]}(n) + y(n) \\ &+ \quad \sum_{i=1}^{L-1} W_L^{[i]}(n) + \sum_{i=1}^{L-1} V_L^{[i]}(n) \end{aligned} \tag{6}$$

satisfies the following equation:

$$u(n+1) = u(n) + \frac{1}{L} \sum_{i=1}^{L} x_i(n) - y(n) \tag{7}$$

Then, we realize that center output

$$y(n) = sgn(u(n)) \tag{8}$$

is the SD modulation of the sum of the inputs $x_i(n)$.

A SD domain adder which outputs sum of L K-level inputs in K-level SD modulated signal form is built mainly of a front sorting network and an internal L-input binary SD domain adder as shown in Fig. 3. The K-level inputs and output are composed of equally weighted $K-1$ bits variables $x_i^{[j]}$ and $y^{[j]}$, $i = 1, 2, \cdots, L$, $j = 1, 2, \cdots K-1$. Outputs from the front sorting network form $K-1$ groups each of which consists of L elements $(s_i^{[1]}, s_i^{[2]}, \cdots, s_i^{[L]})$, i=1, 2, $\cdots$, $K-1$, as shown in Fig. 3. The output is $y^{[i]}$=±1 if $s_i^{[1]}$=$s_i^{[2]}$= $\cdots$ =$s_i^{[L]}$=±1. The L elements of a group

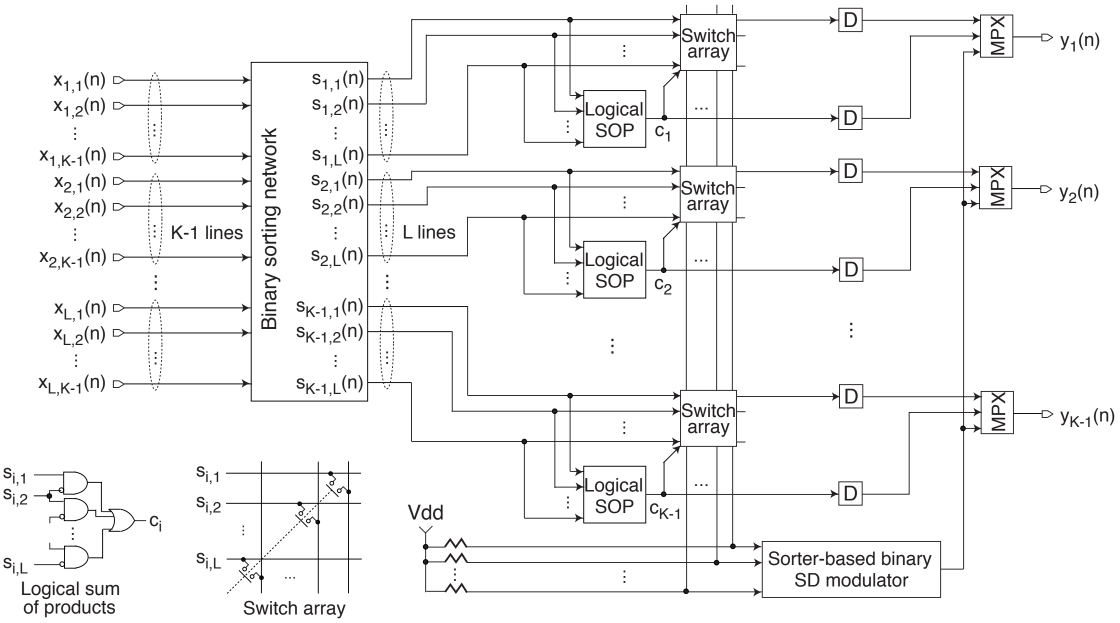


Figure 3: Multi-input multi-level SD domain adder.

are supplied to the internal L-input binary SD domain adder and the output of the internal adder is given to $y^{[i]}$ if the group contains both $+1$ and -1. It is shown in [4] that output set ($y^{[1]}$, $y^{[2]}$, $\cdots$, $y^{[K-1]}$) determined in this way is in K-level SD modulated signal form.

4. Design and Evaluation

We present examples of SD domain multipliers built of the sub-multipliers and the adders shown in previous sections. Figure 4 shows a (NxN=)4x4 SD domain multiplier consisting of four (MxM=)2x2 sub-multipliers, one 4-input 3-level SD domain adder, and one 2-input binary SD domain adder. Figure 5 shows an (NxN=)8x8 SD domain multiplier consisting of four (MxM=)4x4 sub-multipliers, three 8-input 2-level SD domain adders, and three 2-input binary SD domain adders. The outputs of the sub-multipliers can be weighted by cascading 2-input binary SD domain adders as given in Eq. (5).

We will evaluate 4x4, 8x8 and 16x16 SD domain multipliers with 2x2 and 4x4 sub-multipliers. Table 1 shows the gate counts of the multipliers. The upper and the lower numbers in each entry of the table respectively show the gate counts of a multiplier with only 2-input binary adders for summing sub-products and a multiplier with multi-input multi-level adders. Figure 6 presents noise power contained in the outputs of 8x8 multipliers and multipliers with 2x2 sub-multipliers.

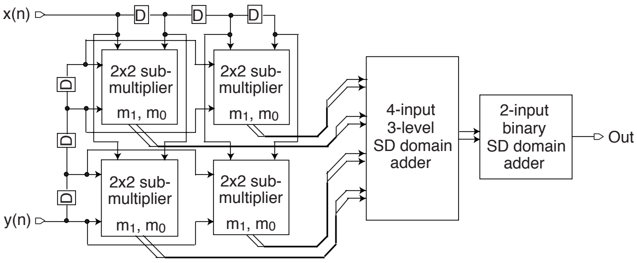

x(n)
y(n)
D
2x2 sub-multiplier
m_1, m_0
4-input 3-level SD domain adder
2-input binary SD domain adder
Out

Figure 4: A 4x4 multiplier.

5. Discussion

We find from Tab. 1 that the SD domain multipliers with 2x2 sub-multipliers and the multipliers with 4x4 sub-multipliers consume about 60 and 50% of the logic gates required to build preceding multipliers with E-OR gates for 1x1 sub-multiplication. The table also shows that using multi-input multi-level adders for summing sub-products does not always effectively reduce the gate counts. This is because the adder with many inputs has a large front sorting network.

Figure 6(a) shows that the output noise level of large NxN SD domain multiplier is low. The figure teaches us that the circuit scale reduction is very important in building practical scale and high precision multipliers. Figure 6(b) shows that the output noise level of SD domain multiplier with large MxM sub-multipliers is low. This is because such sub-multipliers can attenuate SD modulation noise before computing sub-products.

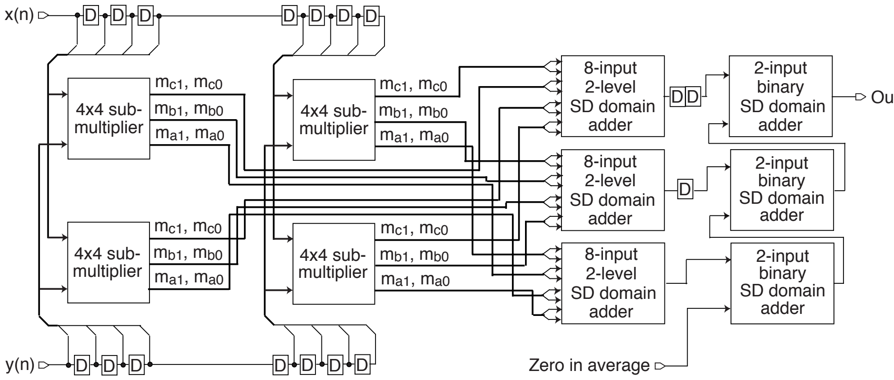


Figure 5: A 8x8 multiplier.

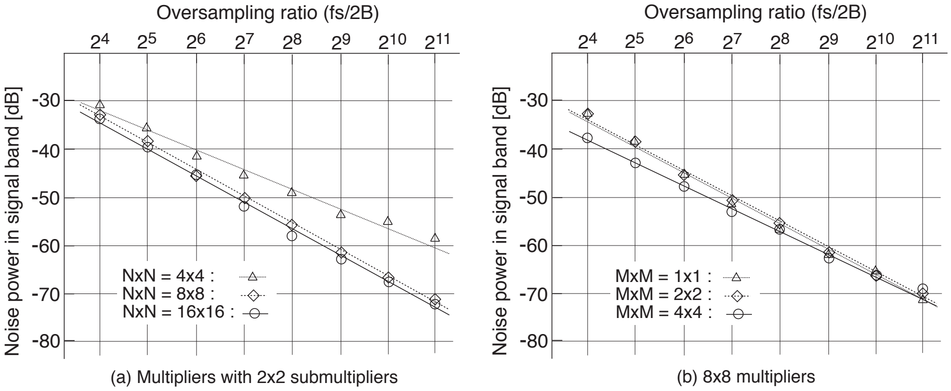


Figure 6: Output noise power of SD domain multipliers.

Table 1: Gate counts of SD domain multipliers with 2-input binary adders (upper) and with multi-input multi-level adders (lower).

NxN \ MxM	1x1	2x2	4x4
4x4	641 805	369 267	339 —
8x8	2625 3107	1537 1594	1246 943
16x16	10481 12335	6129 6823	4794 4968

6. Conclusions

We have presented a design scheme for compact SD domain multipliers. As shown in Tab. 1 we found that using large sub-multipliers was effective in reducing the gate counts. Developing sub-multipliers larger than 4x4 is one of our future works. The front sorting networks of multi-input multi-level adders consume large number of logic gates. The sorting networks can also be built of analog adders and analog threshold circuits. This analog design approach has a possibility to downsizes the sorting networks. To investigate the possibility is our another future subject.

Acknowledgment

The authors would like to thank Osaki Electric Company, Ltd. for the financial support for this research.

References

[1] J. C. Candy and G. C. Temes (eds.), “Oversampling Delta-Sigma Data Converters,” *IEEE Press*, 1992.

[2] H. Fujisaka, R. Kurata, M. Sakamoto and M. Morisue, “Bit-Stream Signal Processing and its Application to Communication Systems,” *IEE Proceedings, Circuits, Device and Systems*, vol. 149, No. 3, pp.159-166, 2002.

[3] I. Pitas and A. N. Venetsanopoulos, ”Nonlinear Digital Filters,” *Kluwer Academic Pubilishers*, 1990.

[4] T. Katao, K. Hayashi, H. Fujisaka, T. Kamio and K. Haeiwa, “Sorter-based Sigma-Delta Domain Arithmetic Circuits,” *Proceedings of ECCTD*, 2007.