On the Accuracy of ROM Stored Square Multipliers

TARUNA TJAHJADI, STUDENT MEMBER, IEEE, AND WILLEM J. STEENAART, SENIOR MEMBER, IEEE

Abstract—**A variant on the stored product digital filter (SPDF) is the stored square digital filter (SSDF) where the contents of all ROM's are identical. An error analysis is given and the accuracy obtained is compared with that of a multiplier digital filter.**

I. Introduction

As LARGER ROM's at low cost are now available, it is expected that in the future digital storage will prevail over digital arithmetic in terms of cost, speed and availability. An analysis of all possible forms of ROM elements replacing multipliers in digital processing is therefore warranted.

The use of ROM elements replacing multipliers in digital filters has been considered using distributed arithmetic [1], [2] and stored product techniques [3], [4]. The use of a stored square ROM was recently suggested [5], [6]. For use in digital filtering this technique requires additions before and after the ROM table look-up. The advantage of having identical ROM contents, independent of the multiplier coefficients, is so attractive that this technique merits an evaluation. It also makes effective time-sharing of the ROM's possible.

An analysis is given of the accuracy obtainable with this technique, as compared with that of binary multiplication, for fixed point and floating point arithmetic, assuming product roundoff error. Signal and coefficient quantization error can be shown to be equal to that obtained in binary multiplication.

The size of the ROM required for various wordlengths can then be formulated as a function of the required accuracy, and of the signal wordlength.

The application of these ROM stored square multipliers in digital signal processing will give the stored square digital filter (SSDF) to compete with the stored product digital filter (SPDF) at lower and medium speed clock rates in applications where multiplexing and the variability of the coefficients are important.

II. The Concept

A multiplication can be expressed as [6]

$$x(n)\cdot c=\left[\frac{(x(n)+c)^2}{4}-\frac{(x(n)-c)^2}{4}\right] \tag{1}$$

where $x(n)$ and c are signal sample and coefficient, respectively. Based on (1) a stored square ROM multiplier (SSRM) is developed. The process involves storing the squared sum of signal and coefficient divided by 4 and the squared difference divided by 4. The quantities $x(n)+c$ and $x(n)-c$ will be used as addresses to the ROM. The overall multiplication consists of additions, subtractions and squaring, where the squares are obtained by a ROM look-up table technique. This technique will be evaluated for roundoff error performance and compared with digital multiplication for both fixed and floating point arithmetic.

III. Error Analysis

A. Fixed Point Arithmetic

Consider a multiplication of two binary numbers using fixed point arithmetic. Assume that the signal, the coefficient, and the output are each represented by N bits and the stored squares are represented by M bits ($N \leq M \leq 2N$) with all numbers rounded and two's complement numbers used for negative values. The block diagram of this multiplication process is shown in Fig. 1. An "absolute value" circuit will change the sign of negative address numbers, and thus keep the number of stored bits to a minimum.

The roundoff errors generated in the process are represented in Fig. 2. The rounding errors of the squared numbers are bounded by

$$-\frac{2^{-M}}{2}\leq \epsilon_{ri}\leq \frac{2^{-M}}{2}, \qquad i=1,2. \tag{2}$$

The roundoff error of the output is bounded by

$$-\frac{2^{-N}}{2}\leq \epsilon_{r3}\leq \frac{2^{-N}}{2}. \tag{3}$$

Using (1), the actual output is

$$y(n)=x(n)\cdot c+\epsilon_{r1}(n)-\epsilon_{r2}(n)+\epsilon_{r3}(n)=P(n)+e_0(n) \tag{4}$$

with the output error

$$e_0(n)=\epsilon_{r1}(n)-\epsilon_{r2}(n)+\epsilon_{r3}(n) \tag{5}$$

and the product

$$P(n)=x(n)\cdot c. \tag{6}$$

The maximum output error to maximum product ratio is, with $P(n)_{\max}=1$

$$\frac{\max|e_0|}{\max|P|}=\frac{2^{-N}}{2}+2^{-M}. \tag{7}$$

Manuscript received June 4, 1981; revised December 18, 1981. This work was supported by the Natural Sciences and Engineering Research Council under Grant A8572.

The authors are with the Electrical Engineering Department, University of Ottawa, Ottawa, Ont., Canada K1N 6N5.

0098-4094/82/0700-0441$00.75

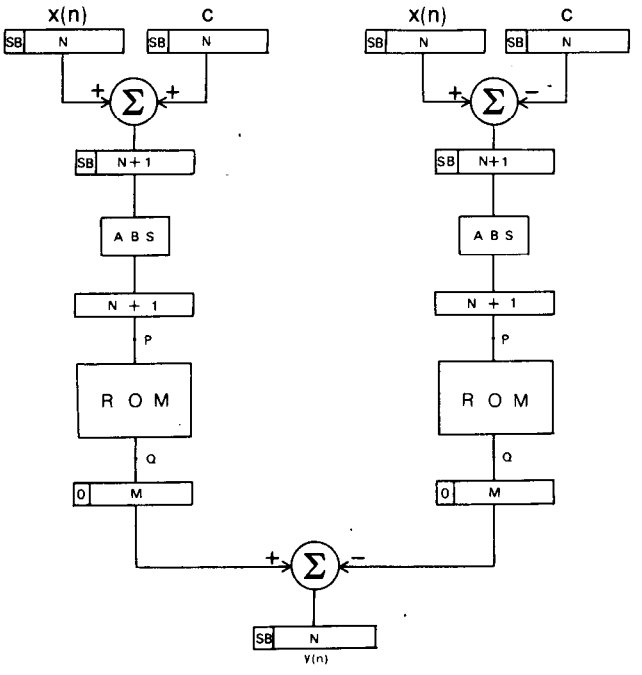

Fig. 1. Block diagram of the multiplication process with fixed point arithmetic.

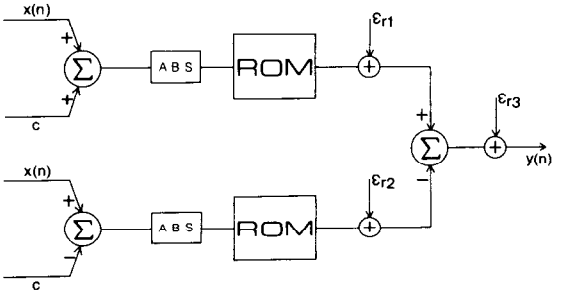

Fig. 2. Block diagram of the multiplication process with roundoff errors, for fixed point arithmetic.

Assuming that the roundoff error sources are statistically independent, the variance of e_0 will be

$$\sigma_{e_0}^2 = \frac{2^{-2N}}{12} + \frac{2^{-2M}}{6}. \tag{8}$$

With $x(n)$ and c assumed to be uncorrelated, white and each of uniform probability density over $(-1, +1)$, the noise to product power ratio is

$$\frac{\sigma_{e_0}^2}{\sigma_P^2} = \frac{3}{4}2^{-2N} + \frac{3}{2}2^{-2M}. \tag{9}$$

In direct multiplication, the roundoff error of the product will be

$$e_{0m} = \epsilon_{rm} \tag{10}$$

where ϵ_{rm} is uniformly distributed between $-\frac{2}{2}^{-N}$ to $\frac{2}{2}^{-N}$. Then, the maximum output error to maximum product is

$$\frac{\max|e_{0m}|}{\max|P|} = \frac{2^{-N}}{2}. \tag{11}$$

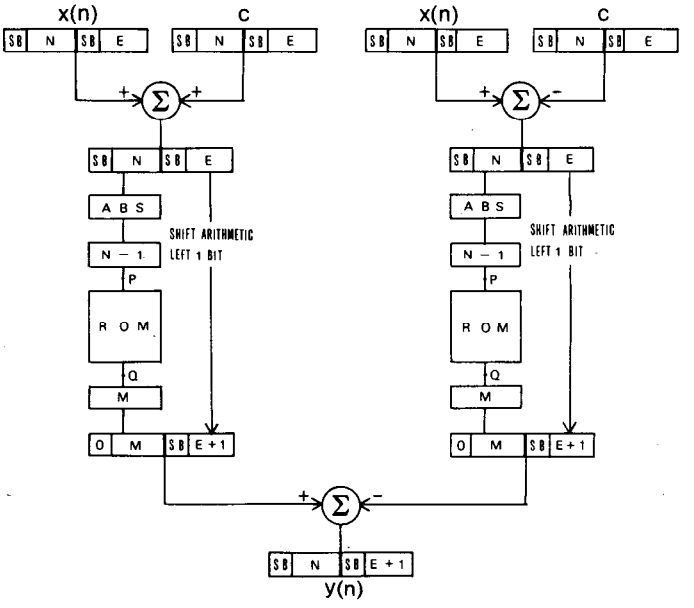

Fig. 3. The block diagram of floating point multiplication process.

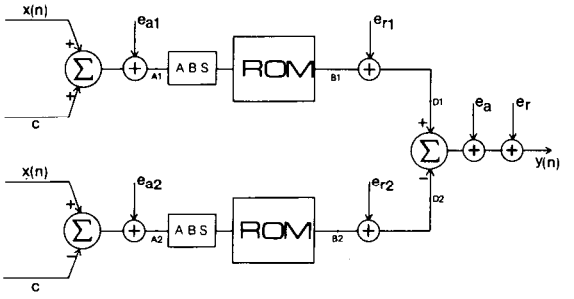

Fig. 4. Block diagram of the multiplication with roundoff errors, for floating point arithmetic.

The variance of the output error is

$$\sigma_{e_{0m}}^2 = \frac{2^{-2N}}{12} \tag{12}$$

and the output noise to product power ratio is

$$\frac{\sigma_{e_{0m}}^2}{\sigma_P^2} = \frac{3}{4}2^{-2N}. \tag{13}$$

The comparisons between stored square ROM multiplier and direct multiplier with fixed point are plotted in Figs. 6 and 8.

B. Floating Point Arithmetic

In floating point arithmetic a number is represented by mantissa and exponent. The mantissa has a normalized magnitude in the range between 0.5 and 1. Let the mantissa of the signal and of the coefficient be N bits and the exponent be E bits, and let the mantissa of the stored squares be M bits. Fig. 3 shows the process in block diagram form. Note that the address to the ROM is $(N-1)$ bits since the address will always have a positive value between 0.5 and 1. The errors that are generated in the process are

1) The errors due to addition or subtraction operation, denoted by e_{ai}.

2) The errors due to rounding, denoted by e_{ri}.

The multiplication process, along with the errors generated, is represented in Fig. 4.

In the following it is assumed that all quantization is by rounding and relative errors are used, denoted by ϵ. All errors are assumed to have the characteristics of white noise, and have uniform probability density.

At nodes $A1$, $A2$:

$$a_i(n) = (x(n) \pm c)\left[1 + \epsilon_{a_i}(n)\right], \qquad i = 1,2. \tag{14}$$

At nodes $B1$, $B2$:

$$b_i(n) = \frac{(x(n) \pm c)^2}{4}\left[1 + \epsilon_{a_i}(n)\right]^2 \cong \frac{(x(n) \pm c)^2}{4}\left[1 + 2\epsilon_{a_i}(n)\right], \qquad i = 1,2. \tag{15}$$

At nodes $D1$, $D2$:

$$d_i(n) = \frac{(x(n) \pm c)^2}{4}\left[1 + 2\epsilon_{a_i}(n)\right]\left[1 + \epsilon_{r_i}(n)\right] \cong \frac{(x(n) \pm c)^2}{4}\left[1 + 2\epsilon_{a_i}(n) + \epsilon_{r_i}(n)\right], \qquad i = 1,2. \tag{16}$$

The actual product is

$$y(n) = \frac{(x(n)+c)^2}{4}\left[1 + 2\epsilon_{a1}(n) + \epsilon_{r1}(n) + \epsilon_a(n) + \epsilon_r(n)\right] - \frac{(x(n)-c)^2}{4}\left[1 + 2\epsilon_{a2}(n) + \epsilon_{r2}(n) + \epsilon_a(n) + \epsilon_r(n)\right]. \tag{17}$$

Let

$$\epsilon_{Ti}(n) = 2\epsilon_{a_i}(n) + \epsilon_{r_i}(n) + \epsilon_a(n) + \epsilon_r(n), \qquad i = 1,2.$$

The output error is

$$e_0(n) = \frac{(x(n)+c)^2}{4}\epsilon_{T1}(n) - \frac{(x(n)-c)^2}{4}\epsilon_{T2}(n). \tag{18}$$

The errors ϵ_{a1}, ϵ_{a2}, and ϵ_r are bounded by

$$-2^{-N} \leq \epsilon_{a1}, \epsilon_{a2}, \epsilon_r \leq 2^{-N}. \tag{19}$$

The errors ϵ_{r1}, ϵ_{r2}, and ϵ_a are bounded by

$$-2^{-M} \leq \epsilon_{r1}, \epsilon_{r2}, \epsilon_a \leq 2^{-M}. \tag{20}$$

The maximum output error is

$$\max|e_0| = [3 \cdot 2^{-N} + 2^{-M+1}] \cdot \max|P| \tag{21}$$

and the maximum output error to maximum product ratio is

$$\frac{\max|e_0|}{\max|P|} = 3 \cdot 2^{-N} + 2^{-M+1}. \tag{22}$$

Let

$$\frac{(x(n)+c)^2}{4} = \alpha(n)$$

and

$$\frac{(x(n)-c)^2}{4} = \beta(n).$$

Assuming that $\alpha(n)$ and $\beta(n)$ are uncorrelated with $\epsilon_{T1}(n)$ and $\epsilon_{T2}(n)$. Assume also that $x(n)$ and c are uniformly distributed between $-t$ and t, it can be shown [7] that

$$\bar{\alpha} = \frac{t^2}{6} \qquad \bar{\beta} = \frac{t^2}{6}$$
$$\overline{\alpha^2} = \frac{t^4}{15} \qquad \overline{\beta^2} = \frac{t^4}{15}. \tag{23}$$

From (18) it follows that

$$\sigma_{e_0}^2 = \sigma_{\epsilon_{T1}}^2 \overline{\alpha^2} + \sigma_{\epsilon_{T2}}^2 \overline{\beta^2}. \tag{24}$$

Since $\sigma_{\epsilon_{T1}}^2 = \sigma_{\epsilon_{T2}}^2$, so

$$\sigma_{e_0}^2 = \sigma_{\epsilon_{T1}}^2\left(\overline{\alpha^2} + \overline{\beta^2}\right). \tag{25}$$

We have

$$P(n) = \alpha(n) + \beta(n) \tag{26}$$

and

$$\bar{P} = \bar{\alpha} - \bar{\beta} = 0 \tag{27}$$

$$\overline{P^2} = \sigma_P^2 = \overline{\alpha^2} + \overline{\beta^2} - 2\overline{\alpha\beta} = \frac{t^4}{9}. \tag{28}$$

Then (25) becomes

$$\sigma_{e_0}^2 = \sigma_{\epsilon_{T1}}^2\left(\sigma_P^2 + 2\overline{\alpha\beta}\right) = \sigma_{\epsilon_{T1}}^2 \cdot \sigma_P^2\left(1 + \frac{2\overline{\alpha\beta}}{\sigma_P^2}\right). \tag{29}$$

From (23), (28), and (29), the output noise to product power ratio is

$$\frac{\sigma_{e_0}^2}{\sigma_P^2} = \sigma_{\epsilon_{T1}}^2(1 + 0.2) = 2 \cdot 2^{-2N} + \frac{4}{5} \cdot 2^{-2M}. \tag{30}$$

With direct multiplication, the maximum output error to maximum product ratio will be

$$\frac{\max|e_{0m}|}{\max|P|} = 2^{-N} \tag{31}$$

and the output noise to product power ratio is

$$\frac{\sigma_{e_{0m}}^2}{\sigma_P^2} = \frac{1}{3} \cdot 2^{-2N}. \tag{32}$$

The maximum output error to maximum product ratio

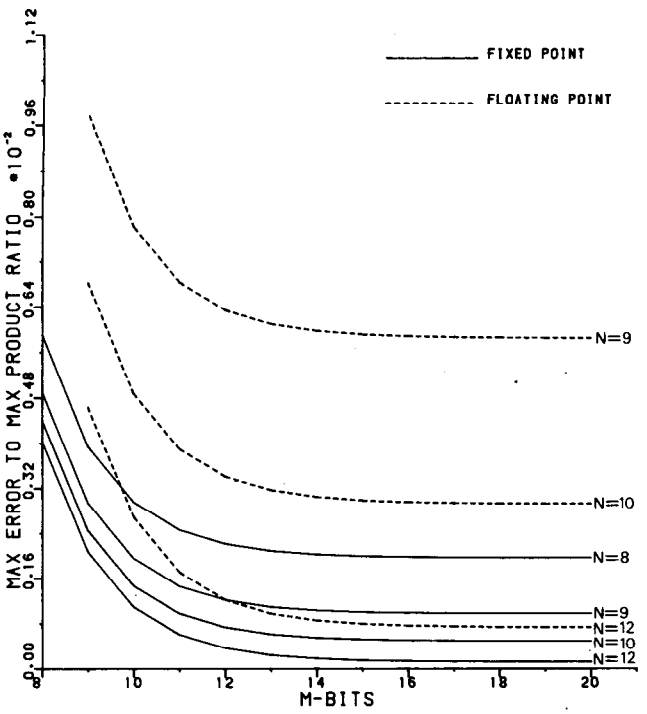

Fig. 5. The max output error to max product ratio for fixed and floating point arithmetic plotted as a function M for various N.

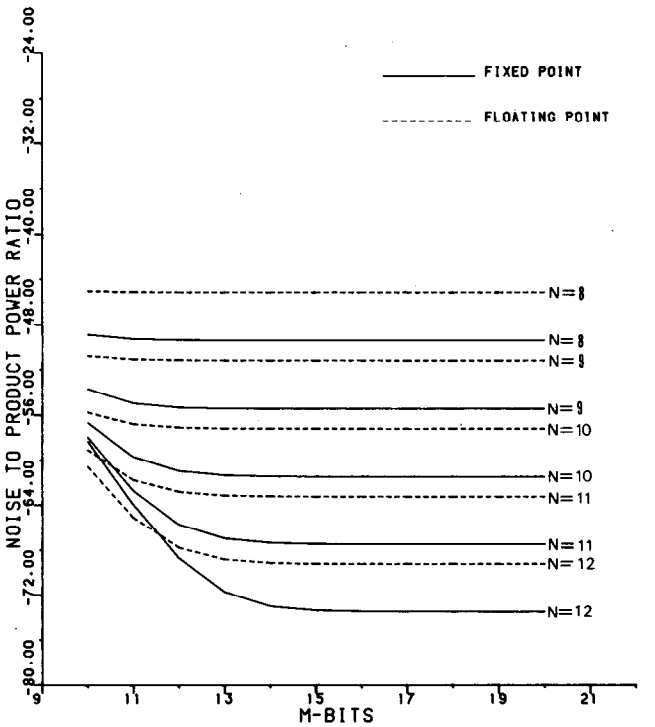

Fig. 7. The output noise to product power ratio for fixed and floating point arithmetic plotted as a function M for various N.

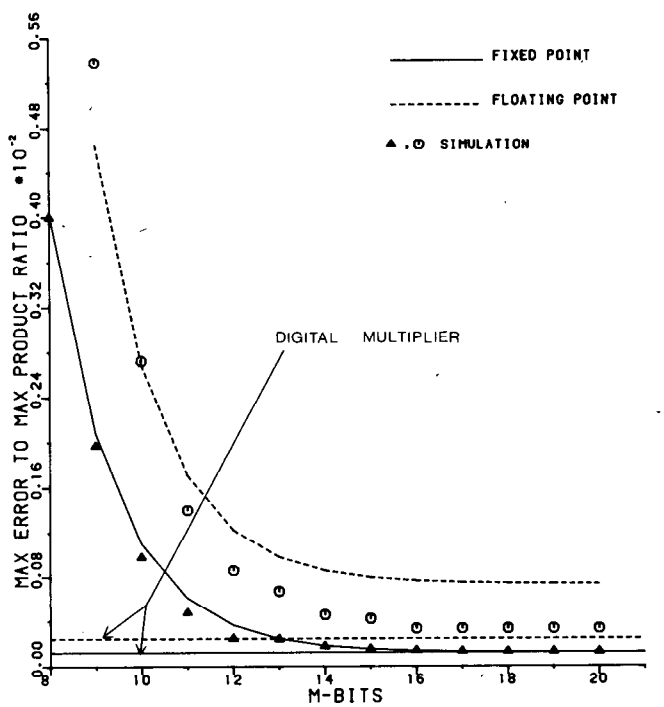

Fig. 6. The plot of max output error to max product ratio with $N=12$.

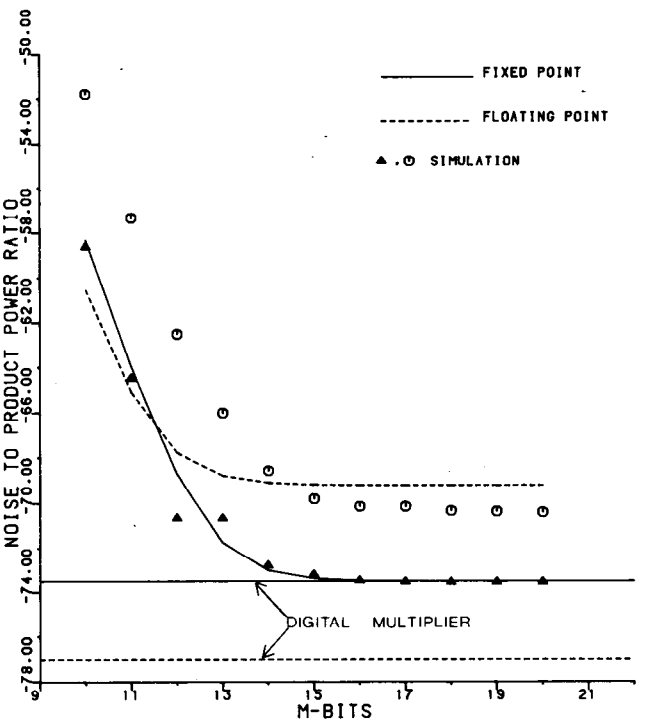

Fig. 8. The plot of output noise to product power ratio with $N=12$.

and the output noise to product power ratio are plotted in Figs. 5–8 for various values of N and M for both fixed- and floating point arithmetic. The simulation result on floating point SSRM shows that the maximum output error to maximum product ratio is about one-half of the theoretical bound.

IV. The Effect of Input Signal and Coefficient Quantization

The analysis is based on the equations derived in Section III. In the previous sections we assumed that the input signal and the coefficient are quantized; here we will take into account the quantization errors and compare these with errors obtained in direct multiplication.

In the following discussion, let $x(n)$ and c be the continuous input signal samples and coefficient, respectively, and $x_q(n)$ and c_q denote the quantized values.

A. The Effect of the Fixed Point SSRM on Quantization Noise

With input signal and coefficient quantized, the actual product (4) becomes

$$y(n)=\frac{\left(x_q(n)+c_q\right)^2}{4}-\frac{\left(x_q(n)-c_q\right)^2}{4}+\epsilon_{r1}(n)-\epsilon_{r2}(n)+\epsilon_{r3}(n) \quad (33)$$

but $x_q(n)=x(n)+\epsilon_x(n)$ and $c_q=c+\epsilon_c$, where $\epsilon_x(n)$ and ϵ_c are the quantization errors. Then

$$y(n)=x(n)\cdot c+x(n)\cdot\epsilon_c+c\cdot\epsilon_x(n)+\epsilon_x(n)\cdot\epsilon_c+\epsilon_{r1}(n)-\epsilon_{r2}(n)+\epsilon_{r3}(n). \quad (34)$$

The error introduced due to input signal and coefficient quantization is

$$e_q(n)=x(n)\cdot\epsilon_c+c\cdot\epsilon_x(n)+\epsilon_x\cdot(n)\epsilon_c. \quad (35)$$

In direct multiplication, the product of $x_q(n)\cdot c_q$ is

$$y(n)=x_q(n)\cdot c_q=\left(x(n)+\epsilon_x(n)\right)\cdot(c+\epsilon_c)+\epsilon_{rm}=x(n)\cdot c+x(n)\cdot\epsilon_c+c\cdot\epsilon_x(n)+\epsilon_x(n)\cdot\epsilon_c+\epsilon_{rm}. \quad (36)$$

So, the error due to input signal and coefficient quantization for on fixed point SSRM is equal to that obtained with direct multiplication.

B. The Effect of the Floating Point SSRM on Quantization Noise

A similar analysis can be carried out for floating point arithmetic. The result, as expected, will give the same input quantization effect on output error for floating point SSRM as with direct multiplication.

V. Comparison Between Fixed and Floating Point Error

If we let $M\to\infty$, then for fixed point

$$\frac{\max|e_0|}{\max|P|}=\frac{1}{2}\cdot 2^{-N} \quad (37)$$

and

$$\frac{\sigma_{e_0}^2}{\sigma_P^2}=\frac{3}{4}\cdot 2^{-2N} \quad (38)$$

which has the same maximum output error to maximum product ratio and the output noise to product power ratio as in digital multiplier.

For floating point

$$\frac{\max|e_0|}{\max|P|}=3\cdot 2^{-N} \quad (39)$$

and

$$\frac{\sigma_{e_0}^2}{\sigma_P^2}=2\cdot 2^{-2N} \quad (40)$$

where the maximum output error to maximum product ratio is three times larger than that of digital multiplier, but in simulation it gives less than half of that. The output noise to product power ratio is twice that of digital multiplier.

Based on Figs. 5–8 it is concluded that the floating point SSRM requires approximately one bit more in the mantissa than the fixed point SSRM wordlength in order to have compatible output noise to product power ratio.

The ROM storage requirements are

a) $2[2^{N+1}\times M]$ bits for fixed point;

b) $2[2^{N-1}\times M]$ bits for floating point ($N=$ mantissa wordlength)

where $M_{\max}=2N$.

VI. Implementation

The proposed technique presents a solution to the replacement of binary multipliers with ROM's. An analysis of how many bits are required to be stored shows that for address wordlength of up to 11 bits this leads to numbers presently realizable at reasonable cost.

For wordlength of 12 bits and larger a possible reduction of the numbers of bits stored is mandatory, and this can be accomplished by using a variation of the split-address technique of stored product digital filtering [8], [9].

For the stored square ROM multiplier this is done by replacing the ROM elements of Figs. 1 and 3 by the "Reduced ROM Squaring Circuit" of Fig. 9.

Consider an address wordlength of K bits, split into two parts of K_1 least significant bits and $K_2=K-K_1$ most significant bits. The magnitude of the address can be expressed as

$$|K|=\sum_{i=1}^{K_2}b_i2^{-i}+\sum_{i=K_2+1}^{K}b_i2^{-i} \quad (41)$$

where $b_i=[0,1]$ (binary number).

The square of (41) will be

$$|K|^2=\left(\sum_{i=1}^{K_2}b_i2^{-i}\right)^2+\left(\sum_{i=K_2+1}^{K}b_i2^{-i}\right)^2+2\left(\sum_{i=1}^{K_2}b_i2^{-i}\right)\left(\sum_{i=K_2+1}^{K}b_i2^{-i}\right). \quad (42)$$

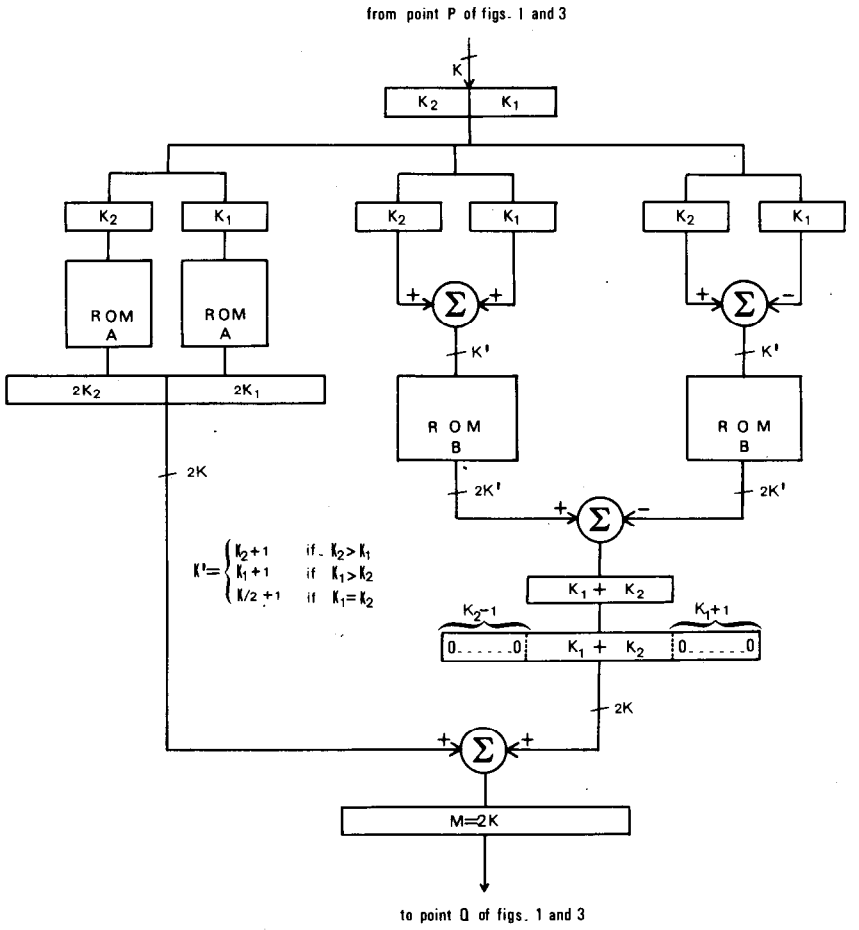

Fig. 9. Block diagram of "reduced ROM squaring circuit." ROM's *A* are squaring ROM's. ROM's *B* are squaring and divide by 4 ROM's.

Rewriting (42), we have

$$|K|^2 = \left(\sum_{i=1}^{K_2} b_i 2^{-i}\right)^2 + \left(\sum_{i=K_2+1}^{K} b_i 2^{-i}\right)^2 + 2^{-(K_2-1)}\left(\sum_{j=1}^{K_1} b_{j+K_2} 2^{-j} \sum_{i=1}^{K_2} b_i 2^{-i}\right). \quad (43)$$

Fig. 9 shows the realization of (43) in block diagram form.

In general, with split address technique the ROM storage requirement is

$$2^{K_2} \times 2K_2 + 2^{K_1} \times 2K_1 + 2(2^{K'} \times 2K')$$

bits per squaring circuit.

For minimal storage requirement, it is found that K_1 and K_2 have to be of the same length or as close as possible. With $K_1 = K_2$, then the storage requirement will be

$$2\left[2^{K/2} \times K\right] + 4\left[2^{K/2+1}\left(\frac{K}{2}+1\right)\right]$$

bits per squaring circuit.

A fixed-point stored square ROM multiplier with $N = 11$ and $M = 24$, if implemented using split address technique will require a ROM storage of

$$2\{2[2^{12/2} \times 12] + 4[2^7 \times 7]\} = 10 \text{ kbits}.$$

TABLE I
COMPARISON OF ROM STORAGE REQUIREMENT

	Storage Size (Bits)	
N	Direct Impl.	Split Address Impl.
7	8K	1.75K
9	40K	4.5K
11	192K	10K
15	4M	52K

Comparing with direct implementation which requires 192 kbits, there is a reduction of almost 20 times and further with $N = 15$ and $M = 32$ we would have a reduction of almost 79 times; however, the use of split address technique will slow down the overall multiplication time somewhat, but with a large reduction of the storage size. Table I shows the comparison of the ROM's size requirement between direct and split address implementation for fixed point with $M = M_{\max}$.

VII. CONCLUSION

The error analysis on the stored square ROM multiplier has been evaluated for fixed and floating point arithmetic. The stored squares are to be of length $M > N$ in order to approach the error obtainable from direct binary multipli-

cation where the product is rounded to length N. After addition of the stored squares the product obtained, however, is also rounded to N bits. In general $M > N+3$ for the product rounding to be comparable, however, for the floating point case the output noise to product power ratio does not approach the one obtainable by direct multiplication and it is found to remain larger even for the case when $M \gg N$.

In order that the fixed and floating point SSRM have comparable output noise to product power ratio, the latter requires one bit more in the mantissa. As for equal input signal wordlength (N) the required ROM storage for floating point is one fourth of that required for fixed point, then with one bit more the floating point storage requirement is still one half of that of fixed point.

For address wordlengths of 12 bits and larger the ROM storage requirement can be reduced by means of "reduced ROM squaring circuit." A large reduction of the storage size can be obtained with this technique.

REFERENCES

[1] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456–462, Dec. 1974.

[2] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 3, 1973.

[3] O. Monkewich and W. Steenaart, "Companding for digital filters," in *Proc. 1975, IEEE ISCAS*, pp. 68–71.

[4] O. Monkewich and W. Steenaart, "Stored product digital filtering with non-linear quantization," in *Proc. 1976, IEEE ISCAS*, pp. 157–160.

[5] M. A. Soderstrand and E. L. Fields, "Multipliers for residue number arithmetic digital filters," *Electron. Lett.*, vol. 13, pp. 164–166, Mar. 17, 1977.

[6] E. L. Johnson, "A digital quarter square multiplier," *IEEE Trans. Comp.*, vol. C-29, pp. 258–261, Mar. 1980.

[7] J. B. Thomas, *An Introduction to Statistical Communication Theory* New York: Wiley, pp. 30–31, 1968.

[8] W. Steenaart, D. Dubois, and O. Monkewich, "Digital filtering, structure, potential and applications," in *Proc. 1981 European Conf. Circuit Theory and Design*, pp. 118–126.

[9] D. Dubois and W. Steenaart, "High speed stored product recursive digital filters," *IEEE Trans. Circuit Syst.*, vol. 29, pp. 390–393, June, 1982.

Taruna Tjahjadi (S'80) was born in Jakarta, Indonesia, on June 24, 1956. He received the B.A.Sc. degree in electrical engineering from the University of Ottawa, Ottawa, Ont. in 1980. He is currently working toward his M.A.Sc. degree at the same university.

His research interests include digital signal processing and microprocessor based systems.

Willem J. Steenaart (A'55–M'57–SM'70) was born in Jakarta, Indonesia, on March 24, 1928. He received the Electrical Engineering degree in 1953 and the Ph.D. degree in 1965, both from the Technological University of Delft, the Netherlands.

Between 1953 and 1958 he worked at Philips Research Laboratories, Eindhoven, the Netherlands; the Northern Electric Company, Montreal, Canada, and at Computing Devices of Canada, Ottawa, Canada, respectively. In 1958 he became a Member of the Technical Staff at Bell Telephone Laboratories, Murray Hill, NJ, where he was engaged in exploratory development work on transmission systems. From 1963 to 1965 he was a research associate at the Technological University of Eindhoven, where he did research on microwave circuits. From 1965 to 1972, he was an Associate Professor at Rensselaer Polytechnic Institute of Troy, NY. In 1972 he joined the faculty of the University of Ottawa where he is a Professor of Electrical Engineering. His interests include digital signal processing and digital filter realization techniques with applications to digital communication systems; communication signal and system optimization for digital signal transmission; microwave communication systems.

Dr. Steenaart is a Professional Engineer in the Province of Ontario, a member of Sigma Xi, and a member of the Radio Institute of the Netherlands. He has served as Chairman of the IEEE Microwave groups at Schenectady and at Ottawa. He has served as Program Chairman of the 1978 International Microwave Symposium at Ottawa, Canada, and as Guest Editor of the Symposium issue, December 1978, of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES. During the 1979–1980 academic year he was an invited professor at the Ecole Polytechnique Fédérale de Lausanne, Switzerland. He has served as Program Chairman for the 1981 IEEE Electrical and Electronic Measurement and Test Instrument Conference, Ottawa, and is Guest-Editor of the March 1982 Conference issue of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENTS.