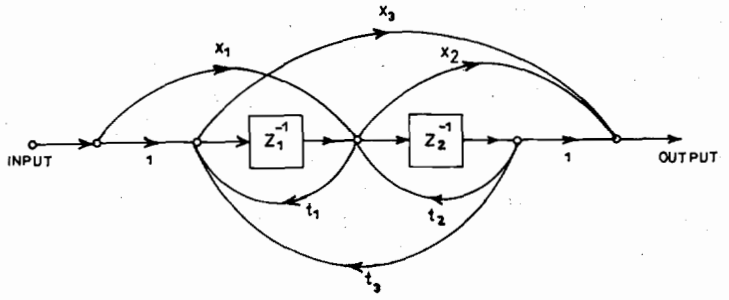

Fig. 1. Signal flow graph to realize a first-order 2-D all-pass digital filter.

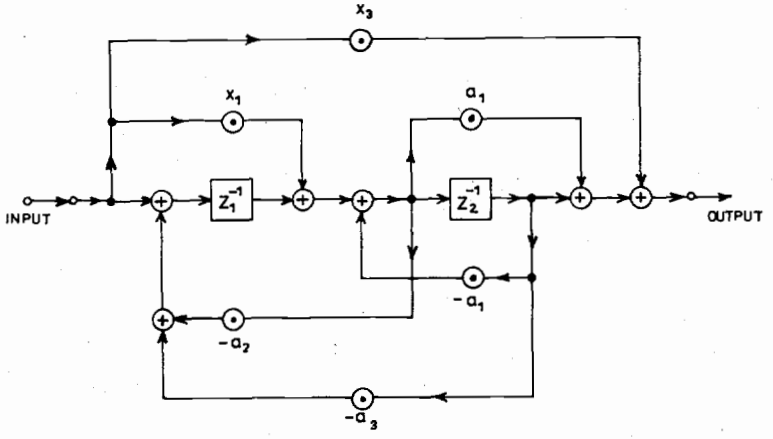

Fig. 2. Minimal delay realization of a first-order 2-D all-pass filter.

$$(a_0 - 1)^2 > (a_1 - a_2)^2$$
$$a_0^2 + 1 - 2a_0 > a_1^2 + a_2^2 - 2a_1a_2$$
$$1 + a_0^2 - a_1^2 - a_2^2 > 2(a_0 - a_1a_2). \quad (17)$$

From inequality (17), it is evident that inequality (15) is always satisfied for stable first-order 2-D all-pass filters. Hence, x_3 in (14) will have real solutions. The value(s) of branch transmittance x_1 is obtained by substituting the value(s) of x_3 into (9) or (10). The minimal delay configuration of the realized 2-D all-pass filter is shown in Fig. 2. Another signal flow graph configuration which also gives minimal-delay realization of the 2-D all-pass filter is the transpose of Fig. 1.

IV. Conclusions

In this paper, we have shown that first-order 2-D digital filters of the all-pass type are realizable with six real-gain multipliers and two delays. The configuration chosen to realize the filter is without a delay-free loop and, hence, reduces the complexity of the network and the computational labor. The realization obtained is minimal with respect to the number of delays.

References

[1] E. Fornasini, "On the relevance of noncommutative power series spatial filters realization," *IEEE Trans. Circuits, Syst.*, vol. CAS-22, pp. 290–299, May 1978.

[2] S. Chakrabarti and S. K. Mitra, "Decision methods and realization of 2-D digital filters using minimum number of delay elements," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 657–666, Aug. 1980.

[3] —, "Corrections to 'Decision methods and realization of 2-D digital filters using minimum number of delay elements,'" *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 262–263, Mar. 1981.

[4] R. C. Joshi, H. Singh, and S. Rai, "First-order 2-D all-pass network realization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1089–1091, Oct. 1981.

[5] S. J. Mason and H. J. Zimmerman, *Electronic Circuits Signals and Systems.* New York: Wiley, 1960.

[6] M. Goodman, "A design technique for circularly symmetric low-pass filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 290–304, Aug. 1978.

[7] G. A. Maria and M. M. Fahmy, "An l_p design technique for two-dimensional digital recursive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 17–21, Feb. 1974.

Memory Intensive Multipliers for Signal Processing

FRED J. TAYLOR

Abstract—**Digital signal processing is an arithmetic-intensive operation. Over the past decade, several memory-intensive multiplication algorithms have been reported. They offer a high-speed alternative to traditional methods. In this correspondence, an error analysis is performed on this class of multiplier and an extended precision version is**

Manuscript received December 31, 1981; revised September 1, 1982 and April 15, 1983. This work was supported by a National Science Foundation ECS grant.

The author was with the University of Cincinnati, Cincinnati, OH 45221. He is now with the Departments of Electrical Engineering and CIS, University of Florida, Gainesville, FL 32611.

derived. The new multiplier is capable of adding several additional bits of precision to a fixed point product without significantly increasing hardware complexity.

I. Introduction

A memory-intensive multiplier is one which uses memory-lookup operations to replace conventional arithmetic. Unfortunately, for fixed point multiplication, the direct-lookup methods of an $n \times n$ product, rounded to its n most significant bits, would require a ROM (or RAM) of size $2^{2n} \times n$ bits. With respect to practical high-speed memory size limitations, this figure is unrealistic for n greater than 6. However, during the past decade, two algorithms have been developed which substantially reduce this burden. They are the following [1]-[7].

Quarter Square Algorithm (QSA) [2]-[6]:

$$xy = \phi(x+y) - \phi(x-y); \quad \phi(s) = (s/2)^2. \tag{1}$$

Bluestein's Identity (BI)[1]*:*

$$xy = \theta(x+y) - \theta(x) - \theta(y); \quad \theta(s) = s^2/2. \tag{2}$$

Using the theorem found in Taylor [7], and an observation made by Johnson [4], it can be verified that no fractional bits must be considered in the division by 2 or 4 in (1) and (2). In order to realize a QSA, two 2^{n+1} word memory units are required, while the BI needs two 2^n words plus one 2^{n+1} word devices.

In this correspondence, the roundoff-error properties of the table-lookup multipliers will be reported. In addition, an extended precision version of these fixed-point sign-magnitude multipliers will also be presented.

II. Error Properties

The table-lookup mappings found in (1) and (2) are a nonlinear function of a single variable [i.e., $\phi(s)$ and $\theta(s)$]. Therefore, the standard-assumption uniform error distribution model may not be applicable [8]. This question was examined using numerical simulation. The argument s in (1) will be assumed to be an integer belonging to Z_N, $N = 2^n$. Over all $s \in Z_N$, a real $\phi(s)$ and a $\phi(s)$ rounded to an m-bit fractional number were computed and analyzed. It was both interesting and comforting to note that the experimentally determined error variance exhibited a distinct $Q^2/12$ character. This means that the classic fixed-point error models used in roundoff error studies of digital systems and filters can be assumed to apply to the memory-intensive multiplier as well.

III. Extended Precision Algorithm (EPA)

The multiplication error can be reduced by increasing the information found in the input field by shifting the contents of an input register left d times until a logical "1" appears in the most significant magnitude digit location. If, for example, a variable z has a string of d "0's" before the first logical "1," then let $x^* = z/2^d$. Under, $\phi(z) = (z^*)^2/4 = \phi(z)/2^{2d}$. Using a left shift of d-bits at the input and a right shift of $2d$-bits at the output, equality can be maintained over an integer ring.

We may now generalize the example problem to the more complex case. Suppose we consider rounding the modulus of an n bit input, say s, to its m most significant bits to form $|\hat{s}|$. Recall that $\theta(|\hat{s}|)$ and $\theta(|\hat{s}|)$ are mappings $\phi: Z_t \to Z_t$, $t = 2^m$, and are mechanized using a $2^m \times n$ bit ROM or RAM. If, for example, $|s| \leqslant 2^{-(n-m)}$, then $|\hat{s}|$ is formed without any rounding error, if the system has the ability to shift an input data string $(n-m)$ bits to the left and an output string $2(n-m)$ bits to the right. From the theory of a random variable, the address presented to a lookup table, for x and y uniformly distributed over (0, 1), is triangularly distributed over (0, 2). Therefore, the table address and system data are of the same wordlength. For P_i denoting the probability prob$(x \leqslant \frac{1}{2}^i)$, $i = 0, 1, \cdots, n$, it can be directly computed that $P_i = (\frac{3}{2})(\frac{1}{4}^i)$ for all $i > 0$ and $P_0 = \frac{1}{2}$. The error variance associated with each transaction is modeled as the familiar $\sigma^2 = Q^2/12$. For the original system, the error variance is approximately

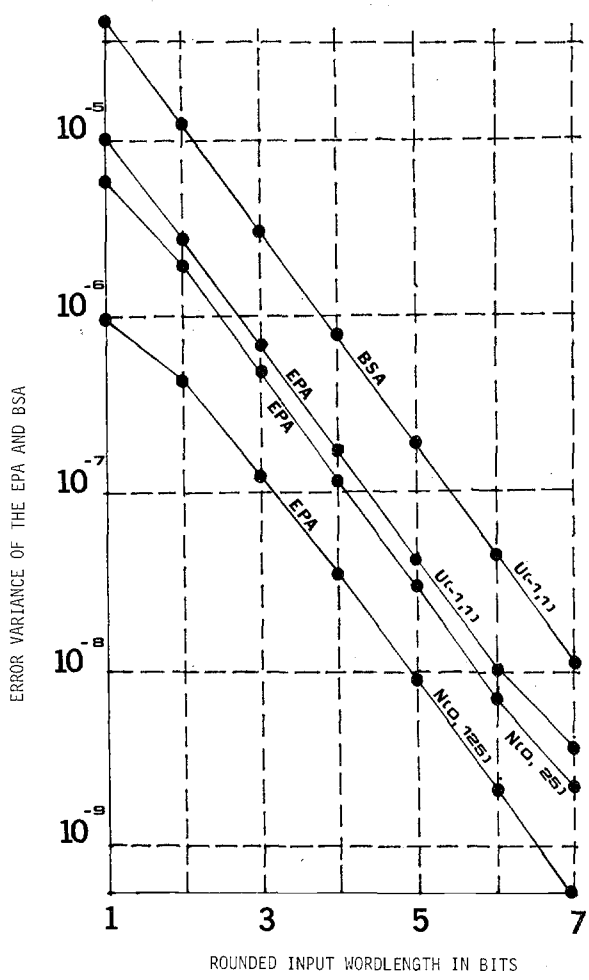

Fig. 1.

$$\sigma_e^2(\text{QSA-BI}) = \sum_{i=0}^{n} P_i \sigma^2 = \sum_{i=1}^{n} \left(\frac{Q^2}{12}\right)\left(\frac{3}{2}\right)\left(\frac{1}{4^i}\right) + \left(\frac{1}{2}\right)\left(\frac{Q^2}{12}\right). \tag{3}$$

For n large, (3) can be evaluated in terms of the well-known identity $\Sigma_{i=0}^{\infty} x^i = 1/(1-|x|)$ if $|x| < 1$. It follows that σ_e^2 can be approximated

$$\sigma_e^2(\text{QSA-BI}) \cong \left(\left(\frac{3}{2}\right)\left(\frac{1}{1-(\frac{1}{4})}\right) - 1\right)\frac{Q^2}{12} = \left(\frac{3}{2}\cdot\frac{4}{3} - \frac{1}{2}\right)\frac{Q^2}{12} = \frac{Q^2}{12} \tag{4}$$

[1] L. I. Bluestein, *IEEE Trans Audio Electroacoust.*, vol. AU-18, Dec. 1970.

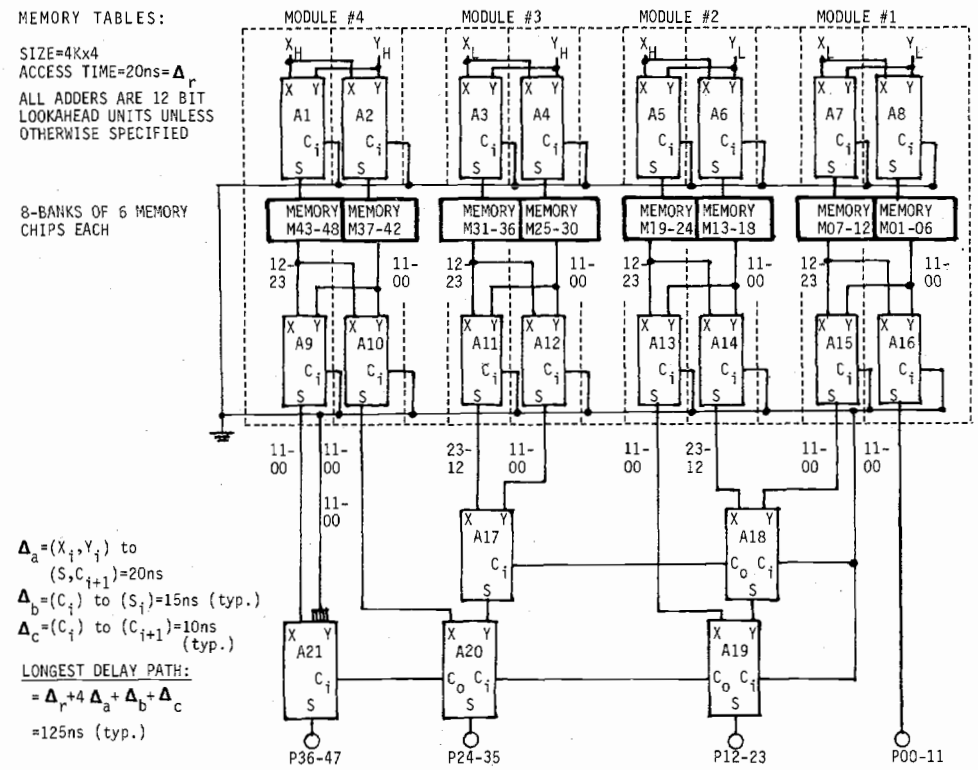

Fig. 2.

as expected. For the *extended precision algorithm* (EPA) system, the analysis for the MSB case is unchanged. In addition, P_i is unchanged for $i > 0$, as well. However, due to the EPA's QSA–BI autoscaling capability, the error variance will be reduced from $Q^2/12$ to $(Q^2/12)/2^{2i}$. Continuing, the error variance for the EPA model becomes

$$\sigma^2_{\text{EPA}}(\text{QSA–BI}) = \sum_{i=1}^{n} \left(P_i \frac{Q^2}{12} \Big/ (1/2^{2i}) \right) + \frac{1}{2} \frac{Q^2}{12} \tag{5}$$

or, for n large,

$$\sigma^2_{\text{EPA}}(\text{QSA–BI}) = \frac{Q^2}{12} \left(\frac{3}{2}\right) \left(\frac{1}{1-(\frac{1}{4})}\right) \Big/ \left(\frac{1}{1-(\frac{1}{4})}\right)$$

$$- \frac{Q^2}{12} = \sigma^2/2. \tag{6}$$

It can be observed that the EPA has approximately one bit [i.e., $\log_2(\frac{1}{2})$] additional accuracy over the basic QSA–BI structure.

IV. Experimentation

In many applications (for example, signal processing, recursive algorithms, etc.), system variables are normally distributed rather than uniformly distributed. If we assume x and y are $N(0, \sigma^2)$, then the values of xy will be most often clustered near zero (i.e., a preamble of logical zeros) (see Fig. 1). Referring to Fig. 1, one notes that over a uniformly distributed database, the EPA is about one bit superior to the QSA, as predicted. Also, for an $N(0, 0.125)$ database, a two-bit improvement can be realized over the QSA–BI, and so on.

V. Hardware Consideration

The QSA is more efficient, in that the memory architecture required of $(x + y)$ is that of $(x - y)$, while the BI exhibits two distinct architectures. Using 35 ns 4K × 4 RAM's, a QSA 12-bit fixed-point multiplier module would require six memory chips and an additional 12-bit adder (typ 15–20 ns). Pipelined throughput rates of 20 ns or less would result (which is superior to those reported by VLSI units). Furthermore, the EPA requires the use of longer shift registers of size $n + 2d_m$, where d_m is the maximal allowable EPA left shift length (chosen by the designer).

VI. Cellular Arrays [9]

Due to a technology-imposed address space limitation (for high-speed memory), a 12-bit multiplier is obtainable. Based on a 12 × 12 multiplier model, a 24 × 24-bit full-precision multiplier can be configured as the array shown in Fig. 2. The displayed array is designed using standard 8K × 4 static RAM's and 12-bit lookahead adders. Based on a 35 ns memory access time, the multiplier throughput can be estimated to be 140 ns. Pipelining would result in a 35 ns (28.6 MMPS) data rate.

VII. Conclusion

The error statistics of two previously reported memory-intensive multipliers, denoted QSA and BI, were determined. It was found that the error variance properties of those multipliers are comparable to other conventional fixed-point multipliers. A new extended-precision version of this class of multiplier was also reported. It was shown that several bits of additional precision can be realized without increasing the memory addressing space.

References

[1] S. R. Logan, "A square-summing high speed multiplier," *Comput. Design*, June 1971.
[2] H. Nussbaumer, "Digital filters using read-only memories," *Electron. Lett.*, vol. 11, 1976.
[3] S. M. Pollard, "Implementation of number-theoretic transforms," *Electron. Lett.*, vol. 12, July 1976.
[4] E. Johnson, "A digital quarter square multiplier," *IEEE Trans. Comput.*, vol. C-29, Mar. 1980.

[5] F. J. Taylor, "Large moduli multipliers," in *Proc. ICASSP*, Denver, CO, Apr. 1980, p. 80.
[6] —, "Large VLSI moduli multipliers," in *Proc. IEEE Circuits Syst. Conf.*, Houston, TX, Apr. 1980.
[7] —, "Large moduli multipliers for signal processing," *IEEE Trans. Circuits Syst.*, July 1981.
[8] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[9] K. Hwang, *Computer Arithmetic*. New York: Wiley, 1970.

Comparative Study of Nonlinear Time Warping Techniques in Isolated Word Speech Recognition Systems

A. WAIBEL AND B. YEGNANARAYANA

Abstract—**In this paper, the effects of two major design choices on the performance of an isolated word speech recognition system are examined in detail. They are: 1) the choice of a warping algorithm among the Itakura asymmetric, the Sakoe and Chiba symmetric, and the Sakoe and Chiba asymmetric, and 2) the size of the warping window to reduce computation time. Two vocabularies were used: the digits (zero, one, · · · , nine) and a highly confusable subset of the alphabet (b, c, d, e, g, p, t, v, z). The Itakura asymmetric warping algorithm appears to be slightly better than the other two for the confusable vocabulary. We discuss the reasons why the performance of the algorithms is vocabulary dependent. Finally, for the data used in our experiments, a warping window of about 100 ms appears to be optimal.**

I. INTRODUCTION

In this correspondence, we present a comparative study of the performance of three different nonlinear time warping algorithms used in isolated word speech recognition systems. The objective is to carefully study the effects of some design choices on the recognition accuracy and to determine factors responsible for the residual errors in the current recognition system. A complete discussion of the various experiments undertaken is given in [1] (also, see [5]). Here, we consider in detail two major design issues for the matching algorithm, namely, 1) choice of warping algorithm and 2) choice of an appropriate search window for the warping algorithm. Results of experiments on a large database for different vocabularies are analyzed, and the factors responsible for significant errors in the recognition are identified.

II. DESCRIPTION OF MATCHING ALGORITHMS

Dynamic programming (DP) consists of mapping the time axis of a speech pattern (test utterance) onto the time axis of another speech pattern (reference utterence) in such a way that the resulting dissimilarity is minimized. The goal of nonlinear time warping is to find the best path (with path index k) through the search space of all possible frame to frame distances $\{d(i(k), j(k))\}$ between the test and reference patterns, where $i(k)$ and $j(k)$ represent the test and reference frame index, respectively. The thick line path connecting points A and D in Fig. 1 is a typical DP search path. Adopting the notation of Sakoe and Chiba [2], the path is given by the minimum cumulative distance score D over all allowable paths:

$$D = \min_f \left[\sum_{k=1}^{K} d(i(k), j(k))\, \omega(k) \Big/ \sum_{k=1}^{K} \omega(k) \right] \tag{1}$$

where f represents all possible paths through the warping plane and $\omega(k)$ is a weighting function. The expression in the denominator serves to normalize the dissimilarity score, to make it independent of the number of points on the search path.

We consider the following three warping algorithms in our studies.

Warp 1—The asymmetric algorithm of Itakura [3].
Warp 2—The best symmetric algorithm of Sakoe and Chiba [2].
Warp 3—The best asymmetric algorithm of Sakoe and Chiba [2].

The warping algorithms span a search space in the shape of a parallelogram ($ABDC$ in Fig. 1) by virtue of the slope constraints. It is reasonable to assume that the paths leading through the corner regions B and C are highly unlikely to occur in reality. If the search space is restricted too severely, then the recognition accuracy may deteriorate. On the other hand, the number of grid points in the search space is directly proportional to the cost of computation. So in general, the cost of computation can be traded with the recognition accuracy. By superimposing a rectangular window onto the parallelogram of the warping search space, we obtain a reduction in search space shown by the dotted area in Fig. 1. The effect of the window width t (shown in Fig. 1) on the recognition accuracy is studied by considering five different values for t, namely, 0, 3, 5, 8, and infinity. The values 0 and infinity correspond to linear time normalization case and no window case, respectively, whereas $t = 3$, 5, and 8 correspond to window tolerance 30 ms, 50 ms, and 80 ms, respectively, for the

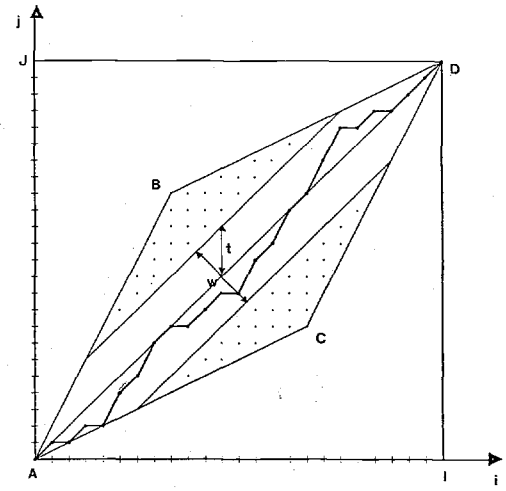

Fig. 1. Restriction of the search via an adjustment window. The dotted area indicates computational savings through the use of the window constraint. Tolerance T is used as a measure of the width as well as the saving acheived.

Manuscript received May 18, 1981; revised August 24, 1982, April 25, 1983, and June 27, 1983. This work was supported by the National Science Foundation and the Advanced Research Project Agency. Any views or conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the above Agencies and Institutions.

A. Waibel is with the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.

B. Yegnanarayana was with the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213. He is now with the Computer Centre, Indian Institute of Technology, Madras-600 036, India.

0096-3518/83/1200-1582$01.00