References

[1] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications.* New York: North Holland, 1979.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco, CA: Freeman, 1979.

[3] T. C. Hu, *Combinatorial Algorithms,* Reading, MA: Addison-Wesley, 1982.

[4] T. C. Hu and E. S. Kuh, *VLSI Circuit Layout Theory and Design.* New York: IEEE Press, 1985.

[5] C. P. Hsu, private communication, 1987.

[6] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning of electrical circuits," *Bell Syst. Tech. J.,* Feb. 1970.

[7] C. L. Liu, *Introduction to Combinatorial Mathematics.* New York: McGraw-Hill, 1968.

[8] G. Persky, D. N. Deutsch, and D. G. Schweikert, "LTX—A minicomputer-based system for automatic LSI layout," in *Proc. 14th Design Automat. Conf.,* May 1977, pp. 217–255.

[9] B. T. Preas and P. G. Karger, "Automatic placement a review of current techniques," in *Proc. 23rd Design Automat. Conf.,* 1986, pp. 622–629.

[10] C. Sechen, "Placement and global routing of integrated circuits using simulation annealing," Ph.D. dissertation, Univ. of California, Berkeley, 1986.

[11] J. Soukup, "Circuit layout," *Proc. IEEE,* Oct. 1981.

[12] J. E. Stevens, "Fast heuristic techniques for placing and wiring printed circuit boards," Ph.D. dissertation, Dep. Comput. Sci., Univ. of Illinois, 1972.

[13] *The TTL Data Book,* Vol. 1, Texas Instruments 1984.

[14] A. Tucker, *Applied Combinatorics.* New York: Wiley, 1980, pp. 109–112.

[15] X. Yao, M, Yamada, and C. L. Liu, "A new approach to the pin assignment problem," *IEEE Trans. Computer-Aided Design,* pp. 999–1006, Sept. 1989.

Generation of Digit Reversed Address Sequences for Fast Fourier Transforms

T. C. Choinski and T. T. Tylaska

***Abstract*—Digit reversed sequences for fast Fourier transforms (FFT) can be generated using firmware resident look-up tables, software algorithms, or application specific hardware. The integrated circuit described herein uses a digital circuit design approach to selectively generate digit reversed sequences for either radix-2, radix-4, or radix-2/4 fast Fourier transforms.**

***Index Terms*—Bit reversal, digit reversal, digital signal processor, fast Fourier transform (FFT), FFT prescramble, integrated circuit.**

I. Introduction

An important part of any fast Fourier transform (FFT) algorithm is unscrambling data sequences using a digit reversal procedure. The reversal procedure rearranges the address sequence used to access data by reordering the bits of a binary representation of the address according to the radix of the FFT. The bit reordering procedure can be realized in firmware, software, or hardware. In any case, the object is to perform the digit reversal procedure in the shortest time with minimal hardware.

Firmware implementations consist of determining the digit reversed address sequence *a priori* and storing the sequence in a table in read only memory (ROM). During execution of the FFT, the digit reversed address sequence is accessed from memory. However, each FFT processed in the computing machine, which differs in size or radix, requires a unique table. The size of the memory required to store the table is directly related to the number of data points transformed by the FFT. A significant increase in memory size can be encountered with the firmware approach.

The digit reversal procedure can also be accomplished using software methods. One software method requires modification of the order of the program code so that the FFT is performed "out of place." This method requires an increase in the size of random access memory in the computer. Another software method reorders the bits of an address pointer by shifting the bits out and into a pointer register. This latter method requires software shifting operations which are very time consuming.

A hardware realization of the digit reversal process reorders the address sequence generated by a binary counter. The rearrangement of the digits is accomplished by selectively routing the bits from the output of the binary counter to an address register. Bit reversal, as opposed to digit reversal, methods have been developed for radix-2 FFT applications [1], [2]. A digit reversal technique for FFT's with different radices can be realized by utilizing several routing networks with the single binary counter. Thus, the digit reversal procedure can be performed for various types of FFT algorithms quickly with minimal hardware. This approach is described in this paper. Specifically, the design of a digit reversed counter for radix-2, radix-4, and radix-2/4 fast Fourier transform algorithms is discussed and presented in detail. The design can also be extended to radix-8 and radix-16 FFT's when appropriate sample sizes are available. Higher radix FFT's offer greater efficiency at reduced versatility.

The digit reversed counter is capable of generating address sequences for fast Fourier transforms varying in size from 4 to 64 K data points. The circuit design can also be adapted for higher sampling sizes. The counter can be used in conjunction with the interstage address sequencer developed by Advance Micro Devices to perform all the address sequencing required by an FFT algorithm in hardware [3].

II. Digit Reversal Concepts for the FFT

Data have to be accessed in a digit reversed order at the input or output stage of an FFT algorithm. The stage depends on whether a decimation-in-time or a decimation-in-frequency FFT algorithm is used [4]. The digit reversal procedure will require that any data found in memory stored in address j be accessed as though it were located in address j', the digit reversed address. Singleton gives a generic definition for j and j' which explains the digit reversal procedure [5]:

$$j = j_m n_{m-1} n_{m-2} \cdots n_1 + \cdots + j_2 n_1 + j_1 \qquad (1)$$

$$j' = j_1 n_2 n_3 \cdots n_m + j_2 n_3 n_4 \cdots n_m + \cdots + j_m \qquad (2)$$

where N is the number of data points to be transformed, m is the number of factors of N, n_i are the factors of N, j_i are the respective digits of j and $i = 1, \cdots, m$.

The generic description of the digit reversal procedure given by (1) and (2) can be easily applied to any number of data points, N, and any FFT radix. These equations will be specifically applied to radix-2, radix-4, and mixed radix-2/4 FFT algorithms. The results from these three applications will then be used to design a digit reversed

Manuscript received September 5, 1988; revised January 29, 1990. This work was supported by NAVSEA PMS-412. The device described in this paper has been registered as a statutory invention at the U.S. Patent Office (Registration Number H570, January 3, 1989).

The authors are with Code 2151 of the Naval Underwater Systems Center, New London, CT 06320.

IEEE Log Number 9041282.

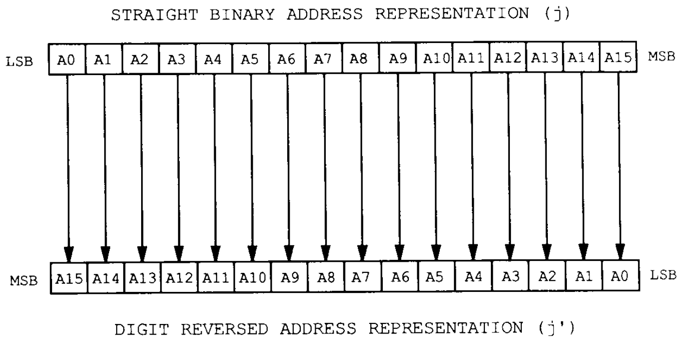
STRAIGHT BINARY ADDRESS REPRESENTATION (j)
LSB
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
MSB
MSB
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
LSB
DIGIT REVERSED ADDRESS REPRESENTATION (j')

Fig. 1 Radix-2 digit reversal map.

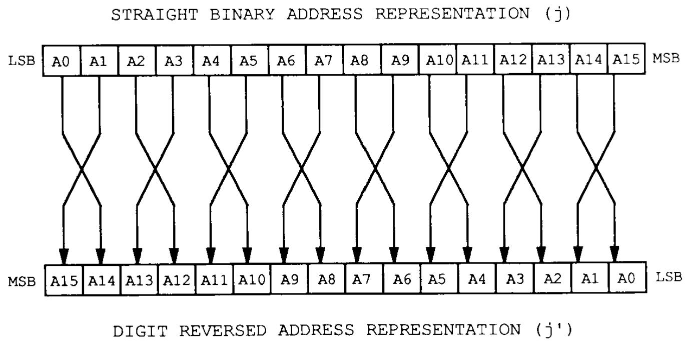
STRAIGHT BINARY ADDRESS REPRESENTATION (j)
LSB
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
MSB
MSB
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
LSB
DIGIT REVERSED ADDRESS REPRESENTATION (j')

Fig. 2 Radix-4 digit reversal map.

counter which can be selected to operate in a radix-2, radix-4, or mixed radix-2/4 mode.

III. Radix-2 FFT Digit Reversal

The bit reversal procedure indicated by (1) and (2) can be best exemplified when N is a power of 2 (i.e., $N = 2^m$) and a radix-2 FFT algorithm is used to perform the transform. For this case j and j' are

$$j = j_m 2^{(m-1)} + \cdots + j_2 2 + j_1 \tag{3}$$

$$j' = j_1 2^{(m-1)} + j_2 2^{(m-2)} + \cdots + j_m. \tag{4}$$

Thus, the order of each radix-2 digit in the binary address sequence is reversed. The most significant bit (MSB) is exchanged with the least significant bit (LSB). Likewise, the second most significant bit is exchanged with the second least significant bit, and so on. In effect, the binary representation of the address is reversed as shown in Fig. 1 for a 16 bit binary pointer.

IV. Radix-4 FFT Digit Reversal

Several authors have previously demonstrated the efficiency of higher radix FFT algorithms and in particular the radix-4 FFT [4], [6], [7]. Singleton's equations for j and j' can be easily used to determine the digit reversal procedure for a radix-4 fast Fourier transform algorithm. The radix-4 FFT requires $N = 4^m$. So j and j' will be

$$j = j_m 4^{(m-1)} + \cdots + j_2 + j_1 \tag{5}$$

$$j' = j_1 4^{(m-1)} + j_2 4^{(m-2)} + \cdots + j_m \tag{6}$$

where $j_1 = 0$, 1, 2 or 3 and $i = 1, 2, \cdots, m$.

If a binary number is used to represent the address j, the bit reversal procedure switches radix-4 digits (i.e., groups of two binary bits) in the address. Therefore, the two most significant binary bits (the most significant radix-4 digit) is exchanged with the two least significant binary bits (the least significant radix-4 digit), and so on. In each exchange of groups of two binary bits the significance of each bit within the group must be preserved as shown in Fig. 2 for a 16 bit address pointer.

V. Mixed Radix-2/4 Digit Reversal

Mixed radix-2/4 FFT algorithms allow users to take advantage of the efficiency of the radix-4 algorithm for data sets which are powers of two but not powers of four. For these data sets, the power of two can be factored out so $N = (2)4^{(m-1)}$. Therefore, the FFT can be performed with one radix-2 stage and the remaining stages as radix-4 stages. The radix-2 stage can be put anywhere in the FFT flow graph.

If the radix-2 stage is performed first in the FFT flow graph then j and j' will be calculated to be

$$j = j_m 2\, 4^{(m-2)} + \cdots + j_2\, 2 + j_1 \tag{7}$$

$$j' = j_1\, 4^{(m-1)} + j_2\, 4^{(m-2)} + \cdots j_m \tag{8}$$

where $j_1 = 0$, 1, 2, or 3 for $i = 2, \cdots, m$ and $j_1 = 0$ or 1. Fig. 3 shows the digit reversal procedure for 16 bit address pointer, given a mixed radix-2/4 case when the radix-2 stage is performed first. Note that all the digit exchanges are performed with groups of two binary bits except for the LSB of j.

When the radix-2 stage is performed in the last stage of the FFT signal flow graph, the digit reversal procedure will define j and j'

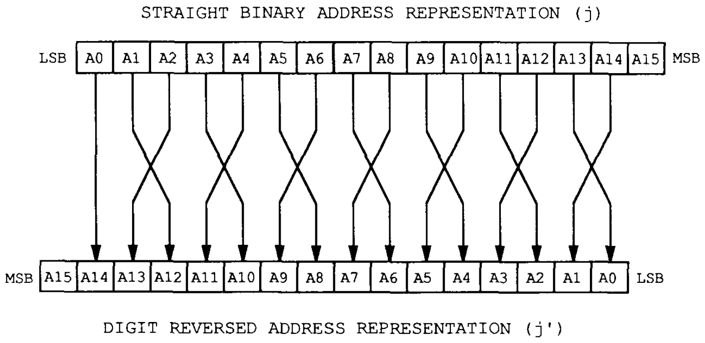
STRAIGHT BINARY ADDRESS REPRESENTATION (j)
LSB
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
MSB
MSB
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
LSB
DIGIT REVERSED ADDRESS REPRESENTATION (j')

Fig. 3. Radix-2/4 digit reversal map.

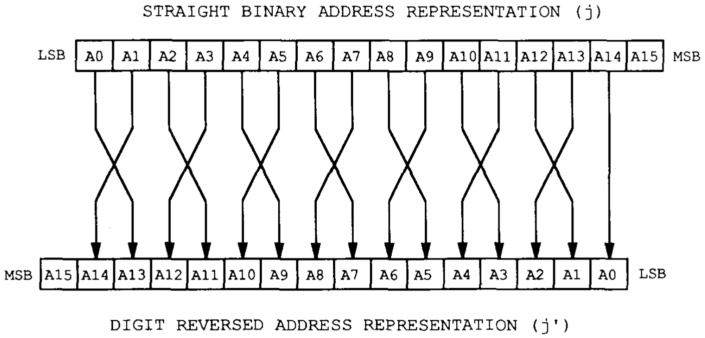
STRAIGHT BINARY ADDRESS REPRESENTATION (j)
LSB
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
MSB
MSB
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
LSB
DIGIT REVERSED ADDRESS REPRESENTATION (j')

Fig. 4 Radix-4/2 digit reversal map.

to be

$$j = j_m \; 4^{(m-1)} + \cdots + j_2 \; 4 + j_1 \tag{9}$$

$$j' = j_1 \; 2 \; 4^{(m-2)} + j_2 \; 4^{(m-3)} + \cdots + j_m \tag{10}$$

where $j_i = 0, 1, 2$ or 3 for $i = 1, 2, 3, \cdots, (m-1)$ and $j_m = 0$ or 1. Fig. 4 shows the digit reversal procedure for the binary representation of a 16 bit address when the radix-2 stage is performed last (radix-4/2). All the exchanges are performed with groups of two binary bits except for the MSB of j.

VI. Digit Reversed Counter Design Description

The design of the logic hardware for the digit reversed counter is very simple and straightforward once the proper digit reversal procedures have been determined. The design centers around a straight binary counter. The counter will cycle its count from zero to $N-1$. The binary bits which are output from the counter are used to generate the digit reversed address. Basically, the binary bits are physically routed to an address pointer using the mapping procedures depicted in Figs. 1–4.

A block diagram for a digit reversed counter which is capable of providing any one of the three radix selections via two external select pins is presented in Fig. 5. The selectable radix digit reversed counter is realized by inserting a multiplexing or switching network between the binary counter and buffers which feed the address data. A photograph of this circuit is presented in Fig. 6.

The multiplexing circuit consists of sets of NAND gates. Either the radix-2, radix-4, or radix-2/4 routing network can be fed to the output address buffers by selecting the appropriate values on the select lines $S0$ and $S1$ (Table I). The multiplexing circuit is illustrated in Fig. 7.

TABLE I
Radix Selection Logic

RADIX	S1	S0
RADIX-2/4	0	0
RADIX-4	0	1
RADIX-2	1	0
X	1	1

The multiplexer circuit can be modified to include higher radix FFT's (e.g., radix-8 or radix-16). Typically, these FFT sizes offer greater efficiency at reduced versatility. These radices can be included in the design by changing the three input NAND gate function in the multiplexer to a five input NAND gate function. Two dual input NAND gates must also be added for each output bit of the counter. The two dual input NAND gates will feed the additional inputs of the five input NAND gate function. The routing to the dual input gates is determined from (1) and (2). The radix selection logic must also be modified to account for the two new radix selections.

The selectable radix digit reversed counter shown in Fig. 6 uses approximately 200 gates to implement all the logic. The gate count can be significantly reduced by using toggle flip flops instead of *JK* flip flops, and pass transistor switches in place of the NAND gate multiplexing circuit. The selectable digit reversed counter circuit can also accommodate smaller or larger FFT sizes if the size for the particular application is known *a priori*. The circuit shown in Fig. 6

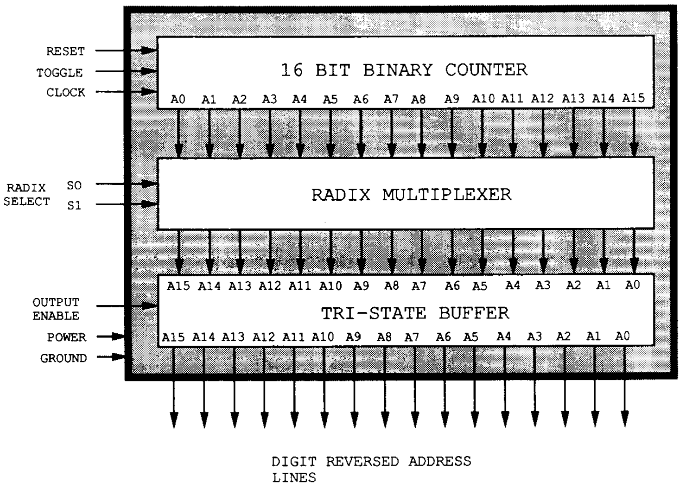
RESET
TOGGLE
CLOCK
16 BIT BINARY COUNTER
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
RADIX SELECT
S0
S1
RADIX MULTIPLEXER
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
OUTPUT ENABLE
TRI-STATE BUFFER
POWER
GROUND
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
DIGIT REVERSED ADDRESS LINES

Fig. 5. Radix selectable FFT prescrambler.

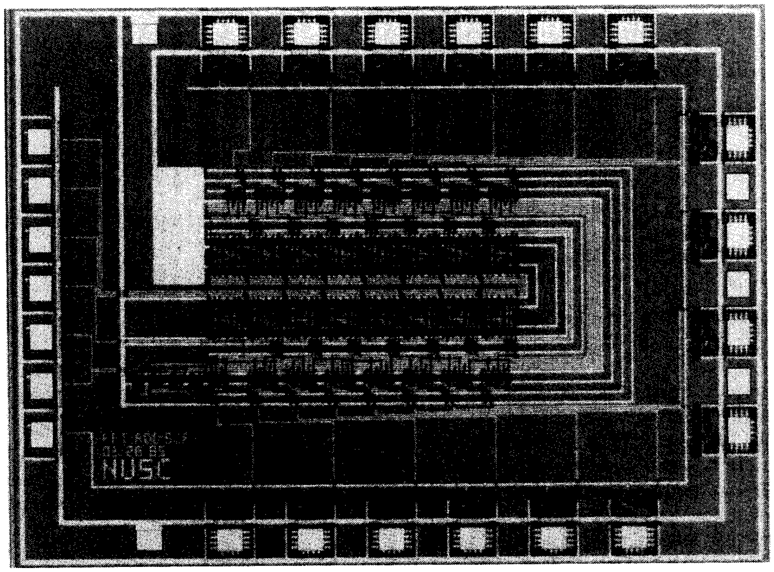

Fig. 6. Prescrambler IC photograph.

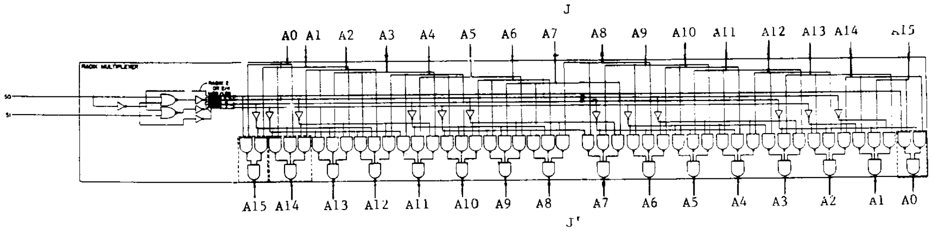
J
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
S0
S1
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
J'

Fig. 7. Radix multiplexer.

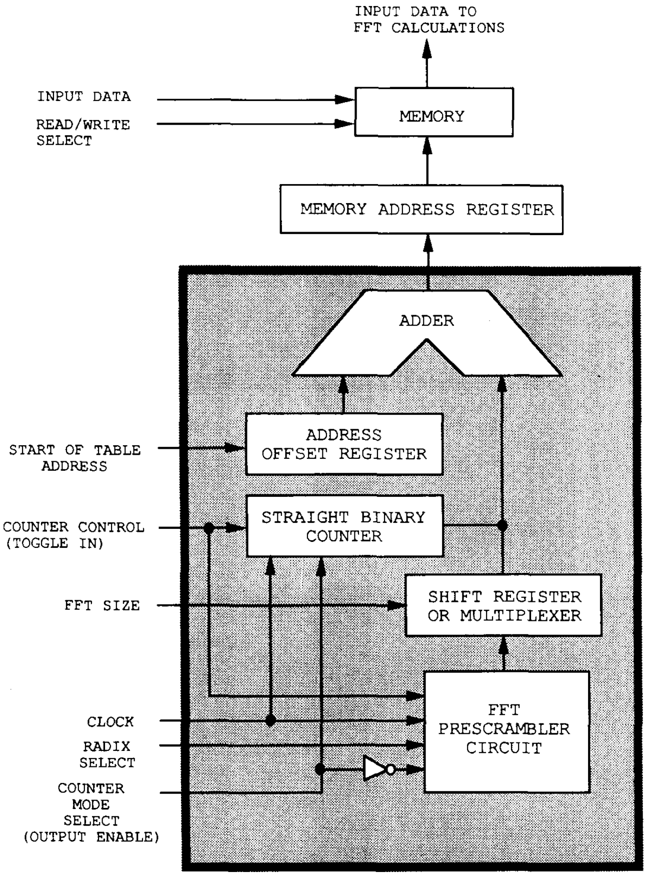

Fig. 8. FFT prescrambled address generation with memory offset.

works for any FFT size between 4 and 64 K merely by tapping the higher order bits at the output according to the FFT size.

If the FFT data are offset in memory, which is the usual case, the digit reversed address must be adjusted external to the digit reversed counter circuitry. Basically, an adder is used to offset the digit reversed address sequence to the appropriate location in memory. A shift register or multiplexing network must be included to align the output bits from the prescrambled address with the adder. The alignment is a function of FFT size. This concept is illustrated in Fig. 8. Note the shift register (or multiplexing network) which creates the programmable FFT size option.

VII. Conclusion

The hardware design of a circuit capable of producing digit reversed sequences for radix-2, radix-4, and mixed radix-2/4 fast Fourier transform algorithms was presented in detail. The design requires selectively routing the output of a binary counter to the output address pointer used during the execution of the FFT.

The selectable radix digit reversed counter offers a highly efficient alternative to firmware and software methods. In addition, its physical size and gate count is small. If this design is used in conjunction with commercially available FFT address sequencing chips all the address generation required during FFT execution can be implemented with hardware techniques.

Acknowledgment

The authors would like to acknowledge T. Min who assisted in the fabrication of the integrated circuit, M. McGowan for providing NUSC's internal review and the other technical reviewers of this paper for their useful comments and suggestions.

References

[1] F. W. Thies, "Method and apparatus for reordering data," U. S. Patent 3,731,284, May 1, 1973.

[2] M. J. Collins and M. T. S. Ching, "Bit reversing apparatus," U. S. Patent 4,181,976, Jan. 1, 1980.

[3] D. Quong and R. Perlman, "One chip sequencer shapes up addressing for large FFTs," *Electron. Design,* July 12, 1984.

[4] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

[5] R. C. Singleton, "An algorithm for computing the mixed radix fast Fourier transform," *IEEE Trans. Audio Electroacoust.,* vol. AU-17, no. 1, June 1969.

[6] R. R. Shively, "Architecture of a programmable digital signal processor," *IEEE Trans. Comput.,* vol. C-31, no. 1, Jan. 1982.

[7] T. C. Choinski and T. T. Tylaska, "An improved FFT for a four parallel pipe SIMD arithmetic processor," Naval Underwater Systems Center Tech. Rep. 7363, May 1985.