

A Parallel Radix-4 Fast Fourier Transform Computer

MICHAEL J. CORINTHIOS, MEMBER, IEEE, KENNETH C. SMITH, MEMBER, IEEE,

AND JUI L. YEN, MEMBER, IEEE

Abstract—The organization and functional design of a parallel radix-4 fast Fourier transform (FFT) computer for real-time signal processing of wide-band signals is introduced.

Several machine oriented FFT algorithms obtained by factoring the discrete Fourier transform (DFT) to an arbitrary radix and which are well suited for the organization of parallel wired-in processors are considered. This class of machine oriented algorithms is distinguished by the fact that it yields machine implementations ranging from fully hard-wired serial sequential processors to partially wired-in cascade processors with no feedback and with a level of parallelism that is proportional to the radix of implementation. Moreover, this class of algorithms can yield the computed Fourier coefficients in a properly ascending order without the need for pre- or postordering of data.

The organization of a system for digital-spectral and time-series analysis implementing a high-speed algorithm is then outlined. It is shown that doubling the processing speed by simultaneous processing of two real-valued time series in such parallel wired-in computers is possible. System considerations for reducing roundoff errors and for performing other processes based on the Fourier transform are discussed.

The organization and design of a radix-4 256-word synchronous sequential FFT signal processor which has been constructed and which performs real-time processing of signals sampled at a rate of up to 1.6 million samples/s is described. Outlined are the basic concepts which have been developed and used to minimize logic circuitry in the different units of the machine. Oscilloscope displays showing the whole sequence of computations involved in real-time Fourier transformation at a sampling frequency of 1.6 MHz are included. The FFT of 256 complex samples is computed in 160 μ s.

Finally, configurations of systems for signal processing using other linear transforms of generalized spectral analysis are described.

Index Terms—Computer architecture, convolution, correlation, digital filtering, digital processing of signals, fast Fourier transform (FFT), special-purpose computer, spectral analysis, time-series analysis.

I. INTRODUCTION

THIS PAPER outlines the basic concepts which have been developed and employed in the organization and construction of a high-speed fast Fourier transform (FFT) signal processing computer.

Manuscript received December 15, 1972; revised November 15, 1973. This work was supported in part by the National Research Council of Canada under Grants A3148, A3951, and A8448.

M. J. Corinthios is with the Department of Electrical Engineering, Ecole Polytechnique, Université de Montréal, Montreal, P.Q., Canada.

K. C. Smith and J. L. Yen are with the Department of Electrical Engineering, University of Toronto, Toronto, Ont., Canada.

FFT algorithm is an efficient way of computing the finite Fourier transform of a time series [2]. The computational saving introduced by the FFT algorithm has rendered digital real-time signal processing potentially feasible in many areas of research including vibration analysis [3], the detection and analysis of radio spectral lines [4], speech processing and communication [5], seismic exploration [6] and electroencephalogram and electrocardiogram analysis [7]. In addition, the technique of factoring the discrete Fourier transform (DFT) to obtain a "fast" algorithm has been shown to be applicable to generalized spectral analysis [8], and to a more general and abstract class of problems, as to finite Abelian groups [9].

The objective of the research reported in this paper has been the organization of a processor for the spectral analysis of wide-band signals (in the MHz range) in real time. The high-processing speed thus called for necessitated the search for algorithms which would be well adapted for parallel machine architecture and wired-in organization of a special purpose computer. The problem is then one of computer architecture in which a proper match is sought between the implemented algorithm and the building blocks of the machine.

Several considerations have guided the search for FFT algorithms suitable for implementation by a special purpose machine. Among these considerations were the emphasis on elimination or reduction of addressing of data, the storage of data in long sequentially accessed queues, the possibility of partitioning the memory into a number of submemories, the words of which to be simultaneously accessed for parallel processing, and the advantage of incorporating a properly ordered set of weighting coefficients and obtaining properly ordered Fourier coefficients without preordering the input. In addition, a sequential rather than cascade processor was believed to be of more feasible cost, since the latter would incorporate a number of arithmetic units (AU's) that is proportional to the number of iterations of the algorithms, i.e., proportional to $\log_2 N$, where N is the number of samples in the input record. Moreover, it was considered important to employ algorithms that can be easily implemented by a cascade processor when higher speeds of processing are essential enough to justify the cost. With these considerations in mind several algorithms have been suggested by Corinthios [10]–[12].

The processor described in this paper is a high-speed radix-4 machine implementing one of a class of algorithms that allows full-time utilization of the AU. A member of this class of algorithms, which will be referred to as the "high-speed algorithms" has been introduced in [12]. This class of algorithms is described in Section II. The organization of a system for on-line digital spectral analysis is then outlined. System considerations for reducing round off errors and for performing other processes based on Fourier transform are discussed.

The organization of the different units of the radix-4 256-word synchronous sequential FFT processor which has been constructed and which performs real-time processing of signals sampled at a rate of up to 1.6 million samples/s is then described. Machine configuration and test results are then outlined. Oscilloscope displays showing the whole sequence of computations involved in real-time Fourier transformation at a sampling frequency of 1.6 MHz are included.

II. HIGH-SPEED FFT ALGORITHMS

In the following, two high-speed algorithms are described. The first is an "asymmetric algorithm" which yields an asymmetric machine where a complex multiplier is included in each but one of its parallel channels [11]. The second is a symmetric one which yields a higher processing speed, but requires one more complex multiplier than the asymmetric one.

Let f_s denote the s th sample of the time series obtained by sampling a generally complex time function $f(t)$ for a duration T . For N such samples the DFT is defined by

$$F_r = \frac{1}{N} \sum_{s=0}^{N-1} (\exp 2\pi jrs/N) f_s \quad (1)$$

where F_r is the r th Fourier coefficient and $j = (-1)^{1/2}$. Both the time increment (s) and frequency increment (r) range between 0 and $N - 1$.

If we denote the sets f_s and F_r , respectively, by the column vectors

$$f = \text{col} (f_0, f_1, \dots, f_{N-1}), \quad (2)$$

and

$$F = \text{col} (F_0, F_1, \dots, F_{N-1}); \quad (3)$$

and if we define a matrix T_N of coefficients given by

$$(T_N)_{rs} = \exp (2\pi jrs/N) \quad (4)$$

$$= w^{rs} \quad (5)$$

where

$$w = \exp (2\pi j/N) \quad (6)$$

then (1) can be written in the form

$$F = (1/N) T_N f. \quad (7)$$

To simplify the notation, as in the papers [10]–[12], we preserve only the exponent of w . That is, we write k in

place of w^k . Then T_N can be written as

$$T_N = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 2 & 3 & \dots & N-1 \\ 0 & 2 & 4 & 6 & \dots & 2(N-1) \\ 0 & 3 & 6 & 9 & \dots & 3(N-1) \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & N-1 & 2(N-1) & 3(N-1) & \dots & (N-1)^2 \end{bmatrix}. \quad (8)$$

In this notation, multiplication of entries becomes addition.

In the following the number of samples N is restricted to values that are multiples of an integer, referred to in the following as the radix r , i.e., $N = r^n$. Let $P_K^{(r)}$ be the ideal-shuffle-base- r permutation matrix of dimension K . The finite Fourier transformation matrix T_N can be partitioned and factored yielding the high-speed ordered-input ordered-output (OIOO) asymmetric algorithm [12]

$$T_N = \prod_{m=1}^n (\mu_m^{(r)} S_m^{(r)}). \quad (9)$$

The pertinent definitions of matrices as derived in [12] are stated in the following, where the symbol x stands for the Kronecker product of matrices and $p_m^{(r)}$ is a permutation defined by

$$p_i^{(r)} = I_{r^{n-i}} \times P_{r,i}^{(r)} \quad (10)$$

and

$$p_1 = \mu_1 = I_N \quad (11)$$

where the notation I_K denotes the identity matrix of dimension K . $\mu_m^{(r)}$ is the weighting or twiddle operator and is given by

$$\mu_i^{(r)} = I_{r^{n-i}} \times D_{r,i}^{(r)} \quad (12)$$

where

$$D_{N/k}^{(r)} = \text{quasidiag} (I_{N/rk}, K_k, K_{2k}, K_{3k}, \dots, K_{(r-1)k}) \quad (13)$$

and

$$K_m = \text{diag} \left\{ 0, m, 2m, 3m, \dots, \left(\frac{N}{rk} - 1 \right) m \right\}; \quad (14)$$

in general

$$S_{m-1}^{(r)} = S^{(r)} p_m^{(r)}; \quad m = 2, 3, \dots, n \quad (15)$$

with

$$S_1^{(r)} = S^{(r)} = (I_{N/r} \times T_r) \quad (16)$$

and

$$T_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N/r & 2N/r & 3N/r & \cdots & (r-1)N/r \\ 0 & 2N/r & 4N/r & 6N/r & \cdots & 2(r-1)N/r \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & (r-1)N/r & \cdot & \cdot & \cdots & (r-1)^2N/r \end{bmatrix} \quad (17)$$

An example of the high-speed OIOO asymmetric algorithm for $N = 8$, $r = 2$ is shown in graph form in Fig. 1.

In a way similar to that which has been utilized in [12] for obtaining the high-speed asymmetric algorithm we can arrive at the high-speed OIOO symmetric algorithm which yields higher speed at the expense of an additional multiplier

$$T_N = \prod_{m=1}^n (\sigma_m U_m) \quad (18)$$

where

$$U_i = I_{r^{i-1}} \times D_{N/r^{i-1}} \quad (19)$$

$$\sigma_m = R_{m-1} S, \quad (20)$$

and

$$R_m = I_{r^{m-1}} \times P'_{N/r^{m-1}}; \quad (21)$$

where

$$P_K' = P_K^{-1}. \quad (22)$$

III. SYSTEM ORGANIZATION OF A DIGITAL SPECTRUM ANALYZER

Fig. 2 shows in block form a possible configuration of a global system for digital spectral analysis. The system incorporates a high-speed processor as its basic central processing unit. In addition, the system comprises an A/D converter, an input buffer memory and means for simultaneous processing of two real records [1], for obtaining averaged power spectra and performing other processes of time-series analysis. Means for automatic array scaling is incorporated in the control unit.

The input function is fed to the system at the input terminal marked IN in the figure. It is sampled and quantized using the A/D converter the output of which is stored sequentially into the input buffer memory marked IB in the figure. IB thus accumulates the elements of the input vector f to be transformed. When N samples have been accumulated, IB contains the N -word input record ready for processing.

The contents of IB are then unloaded into memory MEM1 of the central processor. The central processor performs a high-speed FFT algorithm and yields the output Fourier coefficients at the terminal marked OUT in the figure.

A. Automatic Array Scaling

Truncation or roundoff errors may occur as a result of performing cumulative fixed-point arithmetic in a machine of limited word length. Array scaling, a compromise between fixed-point and floating-point computation, employs a feature that is similar to normalization in floating-point arithmetic. By shifting the bits of a word to the left until all leading zeros are eliminated when the word is positive, and all one's eliminated when the word is negative and two's complement representation is employed, normalization helps preserve least significant bits resulting from arithmetic operations followed by a truncation operation.

As shown in Fig. 2, the control unit includes two boxes for detecting the number of leading zeros at the input to IB and the output of the AU. The first detects the size of data in the input time series, such that when the N th point has been observed the number of leading zeros in the word of maximum size is determined. The second detector performs a similar function throughout the processing iterations. Thus at any iteration, other than the last, the words of the array at the output of the arithmetic unit are monitored and the leading zeros in the maximum word recorded. The control unit, moreover, includes an accumulator for summing the number of left shifts performed at each iteration and presenting it at the end of processing as a scale factor associated with the output array.

B. Machine Organization for Performing Other Processes Based on Fourier Transforms

As shown in Fig. 2, the Fourier coefficients at the output marked OUT of the central processing unit are fed into an auxiliary memory. In order to perform operations which call for the multiplication of two transforms in real time such auxiliary memory would be added to the central processing unit. Operations such as cross correlation and convolution can thus be performed in real time using the auxiliary memory for temporarily storing the first transform until the second is computed.

The remaining part of Fig. 2 shows a method for simultaneous processing of two real-valued time series and obtaining averaged power spectra. The algorithm for processing two real-valued functions simultaneously and obtaining their transforms separately is developed in [1]. This algorithm calls for simultaneous accessing of pairs of words in the F array that are symmetrically located around its middle point. The auxiliary memory is thus divided into two halves, into the first of which the Fourier coefficients $F_0, F_1, \dots, F_{N/2-1}$ are stored; the second stores $F_{N/2}, F_{N/2+1}, \dots, F_{N-1}$. The decoder in the figure includes simply two adders and two subtractors, each of which is followed by a division by 2 in the form of one-bit shift-right operation. Separation of the components of the two transforms is accomplished by successively performing the addition, subtraction, and division operations on the real and imaginary components of each pair of words of the

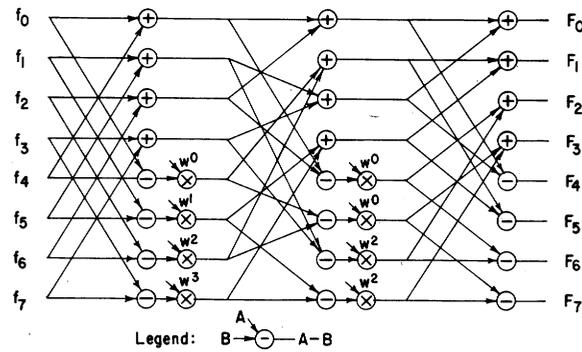
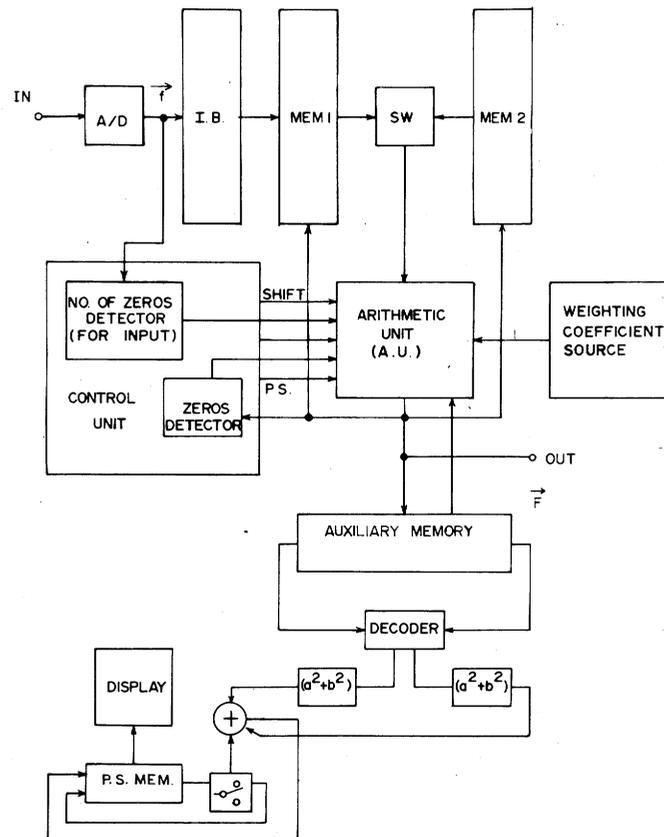
Fig. 1. High-speed OIOO asymmetric algorithm for $N = 8$, $r = 2$.

Fig. 2. System for digital spectrum analysis.

array F . Simultaneous accessing of the proper pairs of words is achieved by right-shift signals applied to the first half of the auxiliary memory simultaneously with left-shift signals applied to the second half. The second half of the auxiliary memory should allow such left-shift operation in addition to the right-shift operation utilized in storing the second half of the array F . Right-shift left-shift registers or random access memories can be used in constructing the second half of the auxiliary memory.

Fig. 2 shows interconnections for computing averaged power spectra. The real and imaginary components a and b , respectively, of each of the decoder outputs are squared and added and the resulting power spectra accumulated into an $N/2$ -word memory, P.S. MEM in the figure, which

may be in the form of recirculating shift registers. The accumulated power spectrum may then be divided by the number of accumulated power spectra to yield the averaged power spectrum, or this number may be associated with the assumed power spectrum as a scale factor.

Finally, we should observe that the scale factor associated with the output array F has to be accounted for in all the operations described in this section.

IV. FUNCTIONAL DESIGN OF A RADIX-4 HIGH-SPEED FOURIER PROCESSOR

The organization and functional design of a radix-4 real-time high-speed Fourier processor which has been

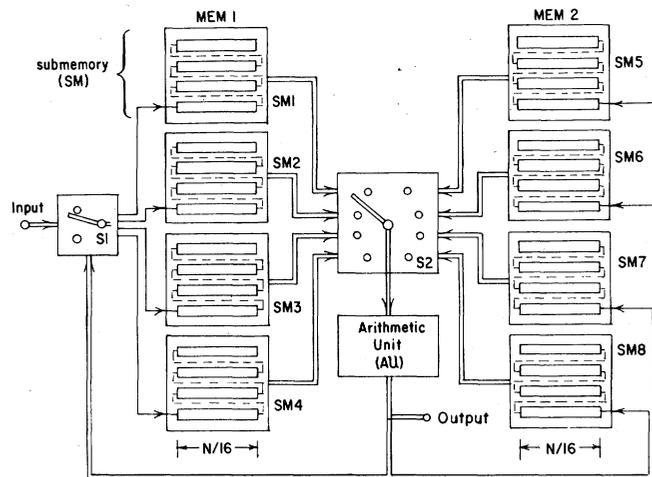


Fig. 3. Radix-4 high-speed processor.

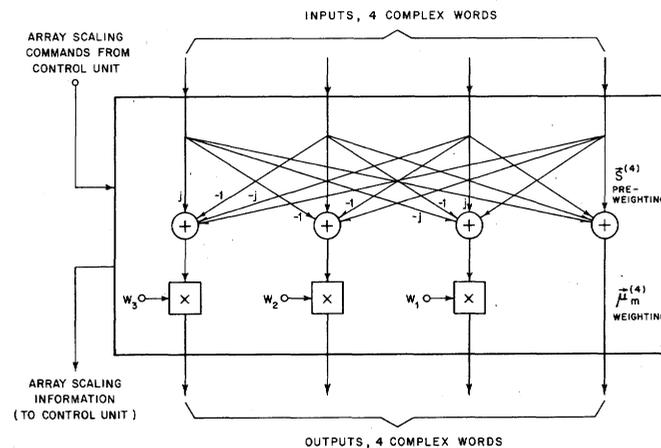


Fig. 4. Schematic of an arithmetic unit for an asymmetric radix-4 processor. Circles including plus sign indicate 4-word adders; squares with X sign indicate complex multipliers.

constructed will now be described. The processor is an asymmetric type machine designed to compute Fourier transforms of 256-word input records in real time.

A. Basic Organization of High-Speed Processors

The Appendix shows the speed of machines implementing the high-speed algorithm relative to those implementing the fully wired-in algorithms described in [11]. Since the difference in implementation cost is minor, a high-speed algorithm was chosen for implementation. The sequential operation of a basic processor will now be described. Reference is made to Fig. 3 which shows a block diagram of an example of a radix-4 machine, even though the description applies to a general-radix machine. As shown in Fig. 3 the machine includes two memories MEM1 and MEM2, each storing N words, an AU and two switches. The AU includes preweighters and weighters as is shown schematically in Fig. 4. Both MEM1 and MEM2 are divided into r submemories (SM), each of which is again divided into r equal length

queues. Switch $S2$ gates r words at each clock pulse to the AU. In all but the first iteration these words constitute the data at the tops of the r queues of a selected SM. In the first iteration, the queues of each SM are connected to form a long queue, and the r words at the tops of the thus formed queues are fed to AU through $S2$. When the input data to the AU are selected from MEM1, designated then as a "source," the output r words of the AU are stored in MEM2, designated "sink," and vice versa. When either MEM1 or MEM2 is in the "sink" mode, the r queues in each of its SM's are connected to form one long queue, to the "rear" of which data is fed.

The n th iteration calls for preweighting only. Thus the data at the output of the preweighter are gated out of the processor during the n th iteration, and the Fourier coefficients are in proper order. We also note that, since the last iteration includes no multiplication the power spectrum can be evaluated during the n th iteration by making use of the otherwise idle multipliers. Power spectra can thus be computed in the same time as that required to

perform the Fourier transformation. The power spectrum is also obtained in proper order which eliminates the need of reordering that may be otherwise be necessary.

B. The AU

The AU shown in Fig. 4 performs the preweighting and weighting operations which involve mainly complex addition, complex subtraction and complex multiplication. In addition, the AU comprises means for performing right- and left-shifts, for avoiding overflow and for array scaling. Each complex number, coded in two's complement is 24-bits long and comprises a real and an imaginary component of 11 bits plus a sign bit each.

In the following we assume the MEM1 contains the input record, the machine is ready to perform the first iteration of the algorithm and the input buffer memory IB is accumulating the samples of the subsequent record. The processor implements the OIOO asymmetric algorithm given by (9) above with $r = 4$. We note that for this value of the radix the operator T_r has the value

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}. \quad (23)$$

The Prewighting Unit: The preweighting unit accepts four input complex words, 24 bits each, and yields at its output four complex words of the same length. It applies the preweighting operator S_m of (15) and (16), to the input operands. The preweighting unit includes eight four-word adders/subtractors the inputs to which are the inputs to the arithmetic unit after being weighted by the values $\pm j$.

We adopt a binary fraction representation of numbers. Array scaling decisions are based in the first iteration upon scanning of the array words upon entry into MEM1. In any of the subsequent iterations the decision is based upon the result of scanning the array words during the preceding iteration.

Let us denote the four selected and properly ordered input words by

$$z_k = z_{kr} + jz_{ki}; \quad k = 0,1,2,3 \quad (24)$$

where the subscripts r and i stand for real and imaginary components, respectively. Let us further denote the outputs of the preweighters by

$$v_k = v_{kr} + jv_{ki}; \quad k = 0,1,2,3. \quad (25)$$

The input-output transformation performed by the preweighting unit can be written in the matrix form

$$v = T_4 z \quad (26)$$

where z and v are vectors whose elements are z_k and v_k ,

respectively, for $k = 0,1,2,3$ and T_4 is given by (23).

It is possible to implement the transformation given by (26) directly. However, a more efficient implementation can be achieved by a factorization of the matrix T_4 that is identical to the original factorization that has been performed on the matrix T_N . This is a further application of the FFT algorithm to the computation of the preweighting transformation matrix. It can be shown that the result of factorization of T_4 may be written in the form

$$T_4 = C_4 E_2 (I_2 \times T_2) \quad (27)$$

$$= \begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \\ 1 & -1 & & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & j \end{bmatrix} \begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \\ 1 & -1 & & \\ & & 1 & -1 \end{bmatrix} \quad (28)$$

It can be shown that through such factorization only four two-word adders/subtractors are needed to compute outputs such as v_{2r} and v_{4r} , as compared to six such units had straightforward evaluation without the factorization of T_4 been performed. This applies for the real and imaginary components of each of the four channels.

The Weighting Unit: The weighting unit comprises three complex multipliers for multiplying the input complex words v_1 , v_2 , and v_3 by the input weighting coefficients w_1 , w_2 , and w_3 , respectively. Each of these complex multipliers calls for four multipliers for real numbers, an adder and a subtractor.

The word lengths in the different units of the machine is selected to yield an absolute output error relative to full-scale input amplitude of less than 1 percent. Results of an error analysis using a computer simulation program of the machine with a set of different word lengths were compared to floating point computation of the transform on the 360/65 IBM machine for different input functions such as the random, sine, square, and ramp functions. It was found that with a quantization to 8 bits of the input function, with the particular array scaling mechanism utilized, the error criterion was satisfied when the memory of the machine has a word length of 12 bits for each of the real and imaginary components. Hardware and feasibility considerations have shown that a word length of 11 bits (10 bits + sign) for the operands to the multiplier were a convenient choice. With such constraints, and with the 20-bit plus sign output of the multipliers kept at that double precision length until after the addition and subtraction operations were performed to yield the complex outputs of the weighting unit, the error criterion was found to be still satisfied. These word lengths were therefore utilized in the construction of the machine.

A Multiplier for Real Numbers: The real-numbers multiplier has a three-dimensional form in the form of parallel

planes, as has already been described in [11]. Its structure is that of a binary tree each node of which contains a parallel n -bit adder, where n is the word length. Higher speeds could be achieved by synchronizing the arrival of sums and carries to each 4-bit parallel adder, through propagation of carries between planes.

It is noted that the multiplier organization lends itself well to pipelining. Thus by the incorporation of latches between the multiplier's planes we can obtain speeds of multiplication approaching the speed of addition within a plane.

The Real-To-Complex (RTC) Unit: The inputs to the RTC unit are the outputs of the four multipliers, namely $v_r w_r$, $v_i w_i$, $v_r w_i$, $v_i w_r$. The inputs are thus four words, each represented by 20 bits plus a sign bit and a control signal for conditional left shift, constitute the inputs to the RTC unit. The outputs of the RTC unit are monitored for the detection of the number of leading zeros as a means for forming the array scaling decisions of the subsequent iteration. The RTC unit completes the complex multiplication operation by adding and subtracting the outputs of the real-numbers multipliers.

C. The Storage and Switching Units

The storage and switching units contain memories MEM1 and MEM2 used for the storage of data, and the switching units incorporated in the processor, namely, units S1 and S2 in Fig. 2 above. The storage unit receives the arithmetic unit outputs AU01 to AU04. Its output is fed to the switching unit S2. The storage units queues were implemented as long-shift registers. Multiplexers were used for implementing the switching unit.

D. The Weighting Coefficients Storage Unit

The weighting coefficients W_1 , W_2 , and W_3 are stored in three read-only memories. Each of the real and imaginary component is quantized to 10 bits plus a sign bit. One extra bit is added to some component, to act as a flag that commands bypassing the multiplier.

E. The Input Buffer Memory

The input buffer memory IB is divided into four sub-memories in the form of long queues, each of which is in turn divided into four shorter queues. It accepts input data quantized to 11 bits plus a sign bit.

The four outputs of IB, denoted IB1, IB2, IB3, and IB4, are connected to the switching unit S1.

F. Sequencing and Control Signals

The sequencing operations called for by the implemented algorithm and the scaling of data throughout processing are performed by signals generated by the control unit. In [1] a concise description of the different events that occur sequentially in the processor and the control signals employed for achieving them is outlined.

In addition to generating signals for moving data between the memories, the arithmetic unit generates signals

for writing and reading data into and out of the input buffer memory, for generating the addressing signals of the ROM's and for array scaling.

The addressing for the ROM's is obtained through the utilization of the six least significant bits of the machine's master counter, to the input of which the main clock is connected. These least significant bits are gated into a weighting-coefficients-address latch at intervals during processing, as specified by the operator μ_m of the high-speed asymmetric algorithm. An accumulator incorporated in the control unit adds the number of shifts performed during one transform and stores the result as a scale factor associated with the output array.

V. MACHINE CONFIGURATION AND PERFORMANCE

A. Machine Configuration

Except for memories MEM1 and MEM2 the circuits of the machine employ SSI and MSI TTL integrated circuits. Bipolar random access memories are used in implementing the input buffer memory circuitry, and bipolar programmable read only memories (PROM's) are used as the storage medium for the weighting coefficients. MEM1 and MEM2 utilize MOSFET shift registers.

Fig. 5 shows a photograph of the different circuits of the machine before final assembly. These include 12 real-number multipliers, memories MEM1, MEM2, switches S1 and S2 and the preweighting unit. In addition, the figure shows the weighting coefficients unit, the control unit, the input buffer memory, and three RTC units. The processor employs a total number of 1800 integrated circuit packages. Figs. 6 and 7 show different views of the assembled processor.

B. Performance of the Assembled Fourier Processor

In this section the performance of the assembled processor when performing Fourier transformation in real time is described. A periodic ramp function the samples of which have the values 0,1,2,...,255 is the input testing function. The outputs of the AU were observed throughout the four iterations and compared bit-by-bit with the results of a computer simulation program. The test was carried at different sampling frequencies ranging from static conditions to the frequency 1.6 MHz which was observed to be nearly the maximum sampling frequency for correct operation. The shift registers employed in the machine are specified to have a maximum cutoff frequency of 2 MHz. Their response to a shifting clock is delayed by 120 to 160 ns, but delays as large as 200 ns were observed. It is the belief of the authors that this varying amount of delay between registers is the main reason for the cutoff frequency of the machine not being closer to 2 MHz. The total computation time per clock pulse is estimated not to exceed 350 ns. Thus the arithmetic unit would allow a sampling frequency that is near to 3 MHz.

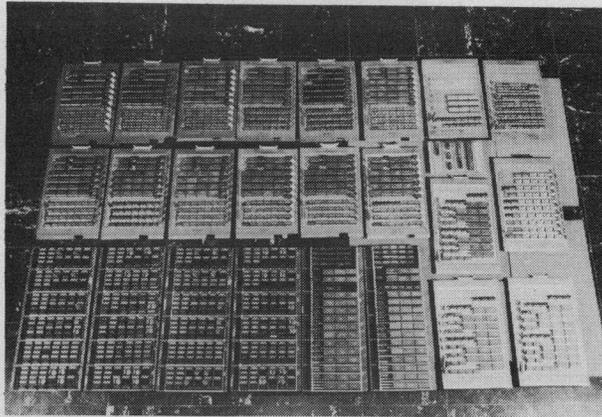


Fig. 5. Top view of all the panels in the machine before final assembly.

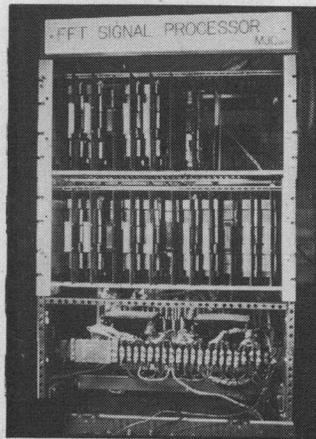


Fig. 6. Front view of the radix-4 high-speed Fourier processor.

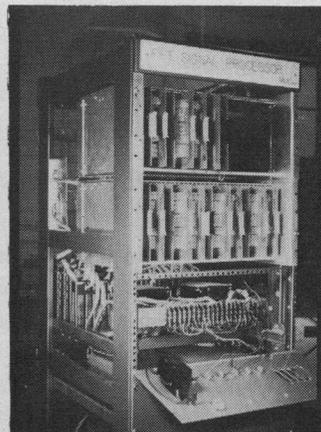


Fig. 7. Side view of the high-speed Fourier processor.

Figs. 8 and 9 show the sequence of rotations and attenuations undergone by the two AU outputs AU01 and AU04 in the complex plane during each of the 256 clocks of real-time operations. These displays have been obtained by attaching D/A converters to the real and imaginary component of each AU output and feed-

ing these two components to the X and Y deflections, respectively, of the oscilloscope. Fig. 8(a) shows the fourth iteration of AU01, while Fig. 8(b) is a simultaneous display of all iterations of AU01. Similarly, Fig. 9(a) and (b) show respectively the fourth iteration and a superposition of all iterations of AU04.

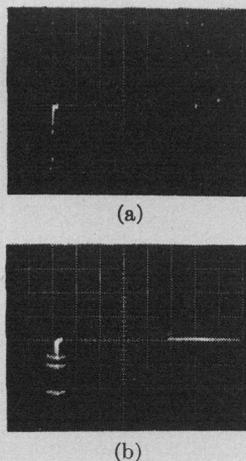


Fig. 8. (a) Iteration 4 of AU01. (b) All iterations of AU01.

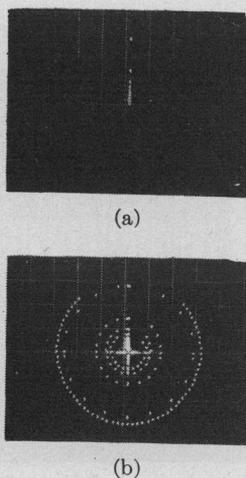


Fig. 9. (a) Iteration 4 of AU04. (b) All iterations of AU04.

To verify these results the outputs of a radix-4 FFT computer program employing floating point arithmetic were plotted in the complex plane and proved identical with the oscilloscope photographs. Figs. 10 and 11 show two of these computer outputs for AU01 and AU04 and are seen to be identical to the oscilloscope displays of these outputs [Figs. 8(b) and 9(b)].

VI. CONCLUSION

By choosing a value of four for the radix of factorization of the DFT, and by implementing a high-speed machine oriented algorithm, we were able to combine parallel machine architecture with wired-in design. Algorithms obtained by factorization to higher radices call for a fewer number of multiplications. Moreover, the reduction of truncation or roundoff errors associated with them, due to the reduction in the number of iterations or stages of processing, constitutes an additional advantage which is significant in determining the word length and hence the size of the machine.

The time of computing the Fourier transform of 256

generally complex samples in real time has been shown to equal $160 \mu\text{s}$ reducible to $128 \mu\text{s}$ and to even less time if fast memory is used. For 1024 samples the corresponding time equals $640 \mu\text{s}$.

A. *The Class of Machines as General Signal Processors*

The class of algorithms and implementing machines outlined above has been discussed in relation to Fourier transforms and power spectrum computation of signals. Such a limitation needs not be imposed on the scope of applications of these processors, however. In fact, these processors should be able to perform the general task of applying a highly symmetric transformation matrix to an input vector. The machines are thus array processors in which the input is a vector in the form of an array which has to be transformed into a new vector through the application of some particular transformation, such as the Walsh, Hadamard, or other transforms of generalized spectral analysis. This transformation matrix is highly symmetric and can be factored into a series of matrix Kronecker products, in a similar way to that applied in factoring the DFT to obtain FFT algorithms. The fac-

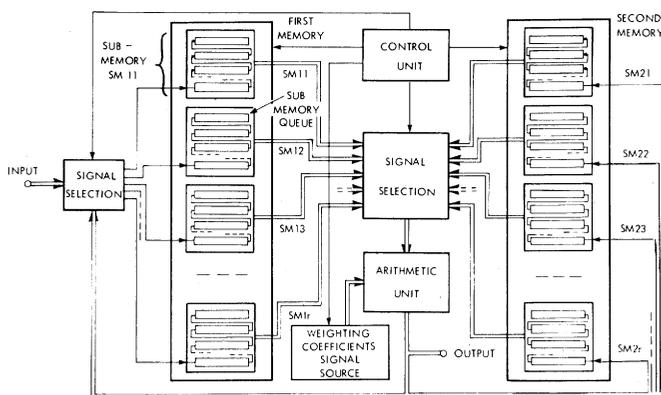


Fig. 12. Generalized high-speed signal processor.

TABLE I
COMPUTATION TIME t_c AND MAXIMUM SAMPLING FREQUENCY f_{\max} FOR REAL-TIME FOURIER TRANSFORMATION OF $N = 4096$ SAMPLES

Radix r	Fully Wired-In Machines				High-Speed Asymmetric Machines	
	Asymmetric Machines		Symmetric Machines		t_p (ms)	f_{\max} (samples/s)
	t_c (ms)	t_{\max} (samples/s)	t_c (ms)	f_{\max} (samples/s)		
2	15.32	265 000	12.84	316 000	6.43	637 000
4	5.63	714 000	5.37	748 000	1.64	2 500 000
8	3.04	1 303 000	3.01	1 317 000	0.69	5 970 000

The assumptions made in this table are as follows.

Assumption 1: $t_{sh} = 0.20 \mu\text{s}$.

Assumption 2: For $r = 2$, $t_p = 0.06 \mu\text{s}$, $t_m = 0.28 \mu\text{s}$.

Assumption 3: For $r = 4$, $t_p = 0.10 \mu\text{s}$, $t_m = 0.30 \mu\text{s}$.

Assumption 4: For $r = 8$, $t_p = 0.20 \mu\text{s}$, $t_m = 0.38 \mu\text{s}$.

torization produces an iterative computation algorithm in the form of a product of matrices. The weighting coefficients involved in the computation are related to the original transformation matrix before factorization, and need be generated according to the implemented algorithm and fed to the arithmetic unit of the processor. Fig. 12 shows the organization of the high-speed signal processor for general spectral analysis. Design and sequencing of operations of these machines should be similar to the ones described in here in relation to the particular application of Fourier transformation.

APPENDIX

Comparison of Maximum Sampling Frequency for Different Machine Organizations

We summarize here the results of a comparison that has been made [1] to evaluate the relative speed between the fully wired-in machines and the high-speed machines. Table I lists the results of this comparison for a record length of 4096 samples. Equations defining the maximum speed for each machine organization can be found in [1]. The assumptions made in Table I involve the following symbols: the time t_{sh} indicates the reciprocal of the maximum frequency of shifting data in memory; the time t_p

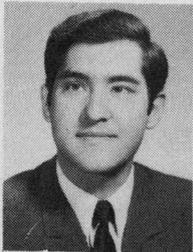
indicates the preweighting time; t_m indicates the total arithmetic time, i.e., the time of preweighting followed by weighting.

Table I shows the higher processing speeds possessed by machines of higher radices. The table also shows the higher sampling frequencies associated with high-speed machines as compared to the fully wired-in machines, for the same radix of factorization of the DFT. The high-speed *symmetric* machines are not included in the table. Their speed of processing is higher than the asymmetric ones, and this difference in processing speed reduces for higher radices; as is the case for fully wired-in machines.

REFERENCES

- [1] M. J. Corinthios, "A class of fast Fourier transform computers for high speed signal processing," Ph.D. dissertation, Dep. Elec. Eng., Univ. Toronto, Toronto, Ont., Canada, Dec. 1971.
- [2] W. T. Cochran *et al.*, "What is the fast Fourier transform?," *Proc. IEEE*, vol. 55, pp. 1664-1674, Oct. 1967.
- [3] C. A. Glew, "The octave band vibration analyzer as a machinery defect indicator," presented at the Amer. Soc. Mechanical Engineers Int. Conf. Vibrations and Design Automation, Toronto, Ont., Canada, Sept. 8-10, 1970, Paper 71.
- [4] J. L. Yen, "A variable resolution radio frequency spectrometer employing time scaling," *Proc. IEEE (Lett.)*, vol. 58, pp. 1373-1374, Sept. 1970.
- [5] J. F. Kaiser, "The digital filter and speech communication," *IEEE Trans. Audio Electroacoust. (Special Issue on Speech Communication and Processing—Part I)*, vol. AU-16, pp. 180-183, June 1968.
- [6] D. Silverman, "The digital processing of seismic data," *Geophysics*, vol. XXXII, pp. 998-1002, Dec. 1967.

- [7] G. Dumermuth and H. Fluhler, "Some modern aspects of numerical spectral analysis of multi channel electroencephalographic data," *Med. Elec. Biol. Eng.*, vol. 15, pp. 319-331, 1967
- [8] H. C. Andrews and K. L. Caspari, "A generalized technique for spectral analysis," *IEEE Trans. Comput.*, vol. C-19, pp. 16-25, Jan. 1970.
- [9] T. W. Cairns, "On the fast Fourier transform on finite Abelian groups," *IEEE Trans. Comput.* (Short Notes), vol. C-20, pp. 559-571, May 1971.
- [10] M. J. Corinthios, "A time-series analyzer," in *Proc. Symp. Computer Processing in Communication*, MRI Symp. Ser., vol. 19. New York: Polytechnic Press, 1969.
- [11] —, "The design of a class of fast Fourier transform computers," *IEEE Trans. Comput.*, vol. C-20, pp. 617-623, June 1971.
- [12] —, "A fast Fourier transform for high speed signal processing," *IEEE Trans. Comput.*, vol. C-20, pp. 843-846, Aug. 1971.
- [13] R. R. Shively, "A digital processor to generate spectra in real time," *IEEE Trans. Comput.*, vol. C-17, pp. 485-491, May 1968.



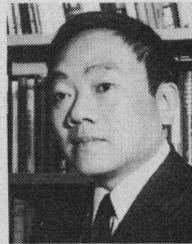
Michael J. Corinthios (S'68-M'69) was born in Cairo, Egypt, on January 19, 1941. He received the B.Sc. degree in electrical engineering from Ain Shams University, Cairo, in 1962, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, Ont., Canada, in 1968 and 1971, respectively.

From 1962 to 1965 he was employed by the Radio Communication Department, United Arab Republic Government, where he worked at the (Marconi) Radio Transmission Stations at Abu Zaabal, Cairo. During 1965 he was with Bell Telephone of Canada Ltd. In the period 1968 to 1969 he was employed by Litton Systems (Canada) Ltd. From 1970 to 1971 he worked as a Consultant Engineer for Hunttec Geological Exploration, Ltd. In October 1971 he joined the Département de Génie Electrique, Ecole Polytechnique de Montréal, Montréal, P.Q., Canada, where he is presently Professeur Adjoint in this department. Recently, he has constructed a radix-4 parallel fast Fourier transform processor. His current research interests include computer organization and design for real-time applications, signal processing and time-series analysis.



Kenneth C. Smith (S'53-A'64-M'60) was born in Toronto, Ont., Canada, on May 8, 1932. He received the B.A.Sc. degree in engineering physics, the M.A.Sc. degree in electrical engineering, and the Ph.D. degree in physics, from the University of Toronto, Toronto, Ont., Canada, in 1954, 1956, and 1960, respectively.

From 1954 to 1955 he served with Canadian National Telegraph as a Transmission Engineer. In 1956 he joined the Computation Center, University of Toronto, as a Research Engineer assigned to assist in the development of high-speed computers at the Digital Computer Laboratory, University of Illinois, Urbana. In 1960 he joined the staff of the Department of Electrical Engineering at the University of Toronto as an Assistant Professor. In 1961 he returned to the University of Illinois as Assistant Professor of Electrical Engineering where he became Chief Engineer for Illiac II and attained the rank of Associate Professor of Computer Science. In 1965 he returned to Toronto where he is Professor in both the Departments of Computer Science and Electrical Engineering. As a Director of Electrical Engineering Consociates Limited, he has participated in development programs in a number of Canadian and U. S. industries. He is currently engaged in the design of digital and linear instrumentation systems.



Jui L. Yen (S'53-A'54-M'60) was born in Canton, China, on November 29, 1925. He received the B.S. degree in electrical engineering from Chiao Tung University, Shanghai, in 1948, and M.A.Sc. and Ph.D. degrees from University of Toronto, Ont., Canada, in 1950 and 1953, respectively.

In 1953 he joined the staff of the University of Toronto, where he is now a Professor of Electrical Engineering. He was a Visiting Associate in Radio Astronomy at the California Institute of Technology, Pasadena, during 1972-1973. He is currently working in radio astronomy, antennas and signal processing.

Dr. Yen is a Fellow of the Royal Society of Canada, and a member of the International Astronomical Union. He was a chairman of Commission VI of the Canadian National Committee for the International Scientific Radio Union. In 1971 he was a co-recipient of the Rumford Award for his contributions in long-baseline interferometry.