Equivalent Relationship and Unified Indexing of FFT Algorithms

Chwen-Jye Ju, Ph.D.
Sharp Microelectronics Technology, Inc.
5700 NW Pacific Rim Blvd.
Camas, WA 98607, USA

Abstract—It is well-known that the twiddle factor matrix of discrete Fourier transform can be recursively factorized into the cascading of the basic butterfly stage matrices. The paper will show that the matrix can be further partitioned into three matrices practically specifying the input data, twiddle factor, and output data sequences of the FFT. Moreover, the equivalent relationship of these matrices is introduced. Thus, the equivalent relationship for a variety of the FFT algorithms can be obtained by equivalent matrix transformation. Furthermore, the paper shows that the multidimensional (M-D) FFT can be represented by the same vector-matrix form as the 1-D FFT. In addition, the addressing sequences of the M-D FFT is the subset of the 1-D FFT. Therefore, the signal flow graph of the 1-D FFT can be used to describe that of the M-D FFT and the 1-D FFT addressing sequences can be employed to implement the M-D FFT. Finally, the tremendous results of the proposed FFT approach simulated by the LH9124/LH9320 are given.

I. Introduction

In recent decades, the fast Fourier transform (FFT) algorithm has been a driving force to the progress of digital signal processing. With the advance of the VLSI technology, the FFT algorithm has been pushed further in solving the multidimensional array signal processing in real-time. However, there is no efficient addressing method for 1-D to M-D FFTs. Therefore, the paper will conquer this problem and propose a unified addressing for 1-D to M-D FFTs. All the M-D indexing can be simplified and implemented by 1-D indexing. The proposed approach has been implemented by many companies in their high-end systems such as radar, medical image recovery, etc.

It is well-known that the computing efficiency of the FFT comes from the recursive factorization of the twiddle factor matrix of the discrete Fourier transform (DFT) [1]. To derive the unified addressing for the 1-D to M-D FFT algorithm, we will factorize and represent twiddle factor matrix into a novel matrix form. Then, all the matrices have their physical meaning in the practical implementation. Each stage of the FFT is represented by three cascaded matrices. The right permutation matrix specifies the input interconnection and define the input data sequence. The left permutation matrix specifies the output interconnection and define the output data sequence. The middle diagonal block matrix performs the butterfly operation and define the twiddle factor sequence.

The equivalent relationship of these matrices are introduced in the paper. It is seen that one kind of the FFT algorithms can be derived from the other kind of the FFT algorithm through the equivalent transformation of the matrices. For example, the in-place bit-reverse inputs and linear outputs (BI/LO) FFT can be derived from the in-place linear inputs and bit-reverse outputs (LI/BO) FFT and vice versa. For definiteness, the paper discusses the decimation-in-time (DIT) FFT only. Essentially, all the results extend to the decimation-in-frequency (DIF) FFT in a straightforward manner.

From the novel vector-matrix representation, we can also derive the equivalent relationship between 1-D and M-D FFTs by employing the equivalent transformation of the matrices. Therefore, it can be obtained that the signal flow graph (SFG) structure of the 1-D FFT can be used to represent that of the M-D FFT regardless of the dimension if the total number of elements is the same. The paper only discusses the radix-2 FFT. Actually, the proposed approach can be extended to an arbitrary mixed radix FFT [2,3]

The unified indexing for 1-D to M-D FFT algorithms has been implemented in the array processor chip set LH9124/LH9320 developed by Sharp Microelectronics Technology. It can be seen from the chip set implementation that the computing time of the FFT is dependent on the total number of data in the array and is independent of the dimension of the FFT. Thus, both 256 by 256 2-D complex FFT and 64K 1-D complex FFT can be finished within 6.56 milliseconds.

II. 1-D Bit-Reverse Input and Linear Output FFT

The DFT of an N-point sequence {x(n)} is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \quad \text{for } 0 \le k < N-1 \quad \text{and} \quad W_N^{i} = e^{-2\pi j i/N} \tag{1}$$

and its parallel form can be represented by the vector-matrix equation as

$$\begin{bmatrix} X(0) \\ X(1) \\ \cdot \\ \cdot \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^1 & \cdot\,\cdot & W_N^{N-1} \\ W_N^0 & W_N^2 & \cdot\,\cdot & W_N^{2(N-1)} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ W_N^0 & W_N^{N-1} & \cdot\,\cdot & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \cdot \\ \cdot \\ x(N-1) \end{bmatrix} \tag{2a}$$

or

$$\underline{X} = W_N * \underline{x}\,. \tag{2b}$$

The structure of the FFT will be based upon how to factorize the twiddle factor matrix W_N. For the BI/LO FFT, W_N is factorized from right first by the bit-reverse matrix P_{br} as

$$W_N = W_N' * P_{br}\,. \tag{3a}$$

For the LI/BO FFT, W_N is factorized from left first as

$$W_N = P_{br} * W_N''\,. \tag{3b}$$

By further recursively decomposing the twiddle factor matrix W_N' [3,4], the vector-matrix form of the 1-D BI/LO FFT with the length $N=2^s$ can be derived as follows

$$\begin{aligned} \underline{X} &= FG_s(BI(s)) * FG_{s-1}(BI(s-1)) * \cdots * FG_1(BI(1)) * P_{br} * \underline{x} \\ &= fft' * \underline{x}_b \end{aligned} \tag{4}$$

where $\underline{X}$ and $\underline{x}_b$, respectively, denote the N-point linear output vector and N-point bit-reverse input vector. The matrix $FG_k(BI(k))$ denotes the k-th radix-2 butterfly stage of the BI/LO FFT. The paper will further partition the butterfly stage matrix into three matrices as

$$FG_k(BI(k)) = P_{lk} * BI(k) * P_{rk}\,. \tag{5}$$

Some essential physical meaning in FFT algorithm implementation can be found through the three-matrix representation of the butterfly stage. The right permutation matrix P_{rk} and the left permutation matrix P_{lk} can specify the input data sequence and the output data sequence of the stage, respectively. The center block diagonal matrix $BI(k)$ performs the radix-2 butterfly operations of the k-th stage and specifies the twiddle factor sequence. It is defined as

$$BI(k) = \begin{bmatrix} bi_k(0) & 0 & \cdot\,\cdot & 0 \\ 0 & bi_k(1) & \cdot\,\cdot & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \cdot\,\cdot & bi_k(N/2-1) \end{bmatrix} \tag{6}$$

where "0" is a 2 by 2 zero matrix and the radix-2 butterfly module $bi_k(n)$ along the diagonal of $BI(k)$ is defined as

$$bi_k(n) = \begin{bmatrix} W_N^0 & W_N^i \\ W_N^0 & -W_N^i \end{bmatrix} \quad \text{with } i = \frac{N}{2^k} * Int\left(\frac{n}{2^{s-k}}\right). \tag{7}$$

The function $Int(x)$ denotes the integer part of the real number x. P_{rk} specifies the interconnection between inputs and butterfly modules and is an N by N permutation matrix with its elements defined as

$$P_{rk}(n,m) = \begin{cases} 1 & \text{for } n = \dfrac{N}{2^{s+1-k}} * Mod(m)_{2^{s+1-k}} + Int\left(\dfrac{m}{2^{s+1-k}}\right) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

0-7803-1254-6/93$03.00 © 1993 IEEE

where $Mod(x)_y$ denotes the modulo operation on the number x with modulo length y and is defined as

$$Mod(x)_y = x - Int(x/y) * y \ . \tag{9}$$

Similarly, P_{lk} specifies the interconnection between outputs and butterfly modules with its elements defined as

$$P_{lk}(n,m) = \begin{cases} 1 & \text{for } m = \frac{N}{2^{s+1-k}} * Mod(n)_{2^{s+1-k}} + Int(\frac{n}{2^{s+1-k}}) \\ 0 & otherwise \end{cases} \tag{10}$$

The bit-reverse matrix is also an N by N permutation matrix with its element defined as

$$P_{br}(n,m) = \begin{cases} 1 & \text{for } n = br(m)_s \\ 0 & otherwise \end{cases} \tag{11}$$

where $br(m)_s$ denotes the bit-reverse operation on the index m with the number of bits s as follows:

$$br(m)_s = br(m_{s-1}m_{s-2} \cdots m_1m_0)_s = m_0m_1 \cdots m_{s-2}m_{s-1} \ . \tag{12}$$

III. 1-D Linear Input and Bit-Reverse Output FFT

The vector-matrix form of the 1-D LI/BO FFT with the length $N=2^s$ can be obtained by further factorizing the twiddle factor matrix W_N'' in (3b) as follows:

$$\underline{X} = P_{br} * FG_1(BL(s)) * FG_2(BL(s-1)) * \cdots * FG_s(BL(1)) * \underline{x} \ . \tag{13}$$

Multiplying P_{br} to both sides of (13), we have

$$\underline{X}_b = P_{br} * \underline{X} = fft'' * \underline{x} \ . \tag{14}$$

As the BI/LO FFT case, the k-th butterfly stage of the LI/BO FFT can be further decomposed into three cascaded matrices

$$FG_{(s+1-k)}(BL(k)) = P_{l(s+1-k)} * BL(k) * P_{r(s+1-k)} \ . \tag{15}$$

$BL(k)$ is the butterfly operation matrix of the k-th stage. Its input interconnection matrix is represented by $P_{r(s+1-k)}$ and its output interconnection matrix is $P_{l(s+1-k)}$. The fundamental difference between the BI/LO and LI/BO structures is the way of defining the butterfly operation matrix. The matrix for the LI/BO FFT is defined as

$$BL(j) = \begin{bmatrix} bl_k(0) & 0 & . \ . & 0 \\ 0 & bl_k(1) & . \ . & 0 \\ . & . & & . \\ . & . & & . \\ 0 & 0 & . \ . & bl_k(N/2-1) \end{bmatrix} \tag{16}$$

The radix-2 butterfly module $bl_k(n)$ of $BL(k)$ is defined as

$$bl_k(n) = \begin{bmatrix} W_N^0 & W_N^i \\ W_N^0 & -W_N^i \end{bmatrix} \quad with \quad i = br(Mod(n)_{2^{k-1}})_{s-1} \ . \tag{17}$$

IV. Equivalence of FFT Algorithms by Matrix Transformation

The previous two sections have discussed that each stage of BI/LO and LI/BO FFT algorithms can be represented by three cascaded matrices. In software or hardware realization of the FFT, these matrices can represent input, output, and twiddle factor addressing sequences. In this section, we will show some equivalent relationship of these matrices. Through these equivalent algorithms, it can be seen that one kind of the FFT structure can be derived from the other kind of the FFT structure. After transformation, the new three cascaded matrices also denote the three addressing sequences for the stage of the FFT. Section VI and VII employ these equivalent relationship to the M-D FFT. Some salient results can be obtained such as the unified 1-D FFT addressing sequences to implement the M-D FFT.

The following will list the theorems that describe the equivalent relationship of the input data, output data, bit-reverse, and twiddle factor matrices. The detailed proof can be found in [4].

Theorem 1: (Input Interconnection Operation)

$$P_{r(Mmod(i+j-1)_s)} = P_{ri} * P_{rj} \quad \text{for } 1 \le i \le s \text{ and } 1 \le j \le s \tag{18}$$

where $Mmod(x)$ is a modified modulo operation function defined as

$$Mmod(x)_y = x - Int(\frac{x-1}{y}) * y \ . \tag{19}$$

Theorem 2: (Output Interconnection Operation)

$$P_{l(Mmod(i+j-1)_s)} = P_{li} * P_{lj} \quad \text{for } 1 \le i \le s \text{ and } 1 \le j \le s \ . \tag{20}$$

Theorem 3: (Input and Output Interconnection Equivalent)

$$P_{rk} = P_{l(s+2-k)} \quad \text{and} \quad P_{lk} = P_{r(s+2-k)} \ . \tag{21}$$

Theorem 4: (Bit-Reverse Equivalent)

$$P_{lk} = P_{br} * P_{rk} * P_{br} \quad \text{and} \quad P_{rk} = P_{br} * P_{lk} * P_{br} \ . \tag{22}$$

Theorem 5: (Equivalence between BI/LO and LI/BO Butterfly Matrices)

$$BI(k) = P_{rs} * P_{br} * BL(k) * P_{br} * P_{ls} \tag{23a}$$

and

$$BL(k) = P_{rs} * P_{br} * BI(k) * P_{br} * P_{ls} \ . \tag{23b}$$

A variety of FFT algorithms can be obtained by changing the order of the input and the output data sequences or by changing the order of butterfly matrix computations. Two examples are given in the following.

A. Example of Deriving Constant Geometry FFT from In-Place FFT

The vector-matrix form of a 16-point in-place LI/BO FFT can be obtained from (13) and (15) and expressed as

$$\begin{aligned} P_{br} * \underline{X} = P_{l1} * BL(4) * P_{r1} * P_{l2} * BL(3) * P_{r2} \\ * P_{l3} * BL(2) * P_{r3} * P_{l4} * BL(1) * P_{r4} * \underline{x} \end{aligned} \tag{24}$$

It can be derived from Theorems 1, 2, and 3 that

$$P_{r1} * P_{l2} = P_{r2} * P_{l3} = P_{r3} * P_{l4} = P_{r4} * P_{l1} = P_{r4} \tag{25}$$

Thus, (24) becomes

$$\begin{aligned} P_{br} * \underline{X} = P_{l1} * BL(4) * P_{r4} * P_{l1} * BL(3) * P_{r4} \\ * P_{l1} * BL(2) * P_{r4} * P_{l1} * BL(1) * P_{r4} * \underline{x} \end{aligned} \tag{26}$$

(26) represents constant geometry 16-point LI/BO FFT. Similarly, constant geometry BI/LO FFTs can be derived from in-place BI/LO FFTs.

B. Example of Deriving In-Place LI/BO FFT from In-Place BI/LO FFT

The vector-matrix form of a 16-point in-place BI/LO FFT can be obtained from (4) and (5) and represented by

$$\begin{aligned} \underline{X} = P_{l4} * BI(4) * P_{r4} * P_{l3} * BI(3) * P_{r3} * P_{l2} * BI(2) \\ * P_{r2} * P_{l1} * BI(1) * P_{r1} * P_{br} * \underline{x} \end{aligned} \tag{27}$$

Using BL(i) to replace BI(i) by (23a) and using (19)-(22) to reduce the number of matrices, we can manipulate (27) into

$$\underline{X} = P_{br} * BL(4) * P_{r4} * BL(3) * P_{r4} * BL(2) * P_{r4} * BL(1) * P_{r4} * \underline{x} \tag{28}$$

Multiplying P_{br} to both sides of (28) and using the equivalent relationship of (25), we can obtain the equation of the 16-point in-place LI/BO FFT shown in (24).

V. Formulation of Vector Matrix Form of 2-D DFT

For a 2-D array (N_1,N_2), the 2-D DFT by the row-column approach over the region is defined as

$$X(k_1,k_2) = \sum_{n_2=0}^{N_2-1} \left[\sum_{n_1=0}^{N_1-1} x(n_1,n_2) W_{N_1}^{n_1k_1} \right] W_{N_2}^{n_2k_2} \tag{29}$$

$$\text{for } 0 \le k_1 < N_1 \text{ and } 0 \le k_2 < N_2$$

The parallel vector-matrix form of the 2-D DFT can be expressed as

$$\begin{bmatrix} X_{c0} \\ X_{c1} \\ . \\ . \\ X_{c(N_1-1)} \end{bmatrix} = \begin{bmatrix} TW_c & 0 & . \ . & 0 \\ 0 & TW_c & . \ . & 0 \\ . & . & & . \\ . & . & & . \\ 0 & 0 & . \ . & TW_c \end{bmatrix} P_T \begin{bmatrix} TW_r & 0 & . \ . & 0 \\ 0 & TW_r & . \ . & 0 \\ . & . & & . \\ . & . & & . \\ 0 & 0 & . \ . & TW_r \end{bmatrix} \begin{bmatrix} x_{r0}^T \\ x_{r1}^T \\ . \\ . \\ x_{r(N_2-1)}^T \end{bmatrix} \tag{30}$$

where TW_r and TW_c denote the twiddle factor matrices for the row DFT and column DFT, respectively. x_{ri} is the i-th row of the input array and X_{cj} is the j-th column of the output array. Set the row length $N_1=2^{s_1}$, column length $N_2=2^{s_2}$, and total elements $N=N_1*N_2$. The transpose matrix P_T is employed to transform the 2-D array from row-major order to column-major order and is also a permutation matrix expressed as

$$P_T = P_{r(s_1+1)} \quad \text{and} \quad P_T^{-1} = P_{l(s_1+1)} . \tag{31}$$

VI. 2-D Linear Input and Bit-Reverse Output FFT

The vector-matrix form for the 2-D LI/BO FFT implementation of (30) can be represented by

$$\begin{bmatrix} X_{bc0} \\ X_{bc1} \\ \cdot \\ \cdot \\ X_{bc(N_1-1)} \end{bmatrix} = \begin{bmatrix} cfft & 0 & .\,. & 0 \\ 0 & cfft & .\,. & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & .\,. & cfft \end{bmatrix} P_T \begin{bmatrix} rfft & 0 & .\,. & 0 \\ 0 & rfft & .\,. & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & .\,. & rfft \end{bmatrix} \begin{bmatrix} x_{r0}^T \\ x_{r1}^T \\ \cdot \\ \cdot \\ x_{r(N_2-1)}^T \end{bmatrix} \tag{32}$$

where X_{bci} is the i-th column of the 2-D bit-reverse output array. $rfft$ and $cfft$ can be represented by (13) with length N_1 and N_2, respectively. It can be derived that the N_2 $rfft$s can be implemented by the first s_1 stages of the N-point 1-D FFT and the N_1 $cfft$s by the first s_2 stages of the N-point 1-D FFT. Thus, (32) yields

$$\underline{X}_{bc} = FG_1(BL(s_2)) * FG_2(BL(s_2-1)) * \cdots * FG_{s_2}(BL(1)) * P_T$$
$$* FG_1(BL(s_1)) * FG_2(BL(s_1-1)) * \cdots * FG_{s_1}(BL(1)) * \underline{x}_r^T \tag{33}$$

where $\underline{X}_{bc}$ is the N-point output vector and $\underline{x}_r^T$ is the N-point input vector shown in (32). Combining the transpose matrix P_T with the row FFTs and using Theorems 1, 2, and 3, (33) can be transformed into a new form of the 2-D FFT structure

$$\underline{X}_{bc} = FG_1(BL(s_2)) * FG_2(BL(s_2-1)) * \cdots * FG_{s_2}(BL(1))$$
$$* FG_{s_2+1}(BL(s_1)) * FG_{s_2+2}(BL(s_1-1)) * \cdots * FG_{s_1+s_2}(BL(1)) * \underline{x}_c \tag{34}$$

where $\underline{x}_c$ is an N-point vector in the linear column-major order of the input array.

Comparing (34) with (13) and setting $s=s_1+s_2$, we can see that the N_1 by N_2 2-D FFT has the same interconnection structure as the N-point 1-D FFT. It implies they can have the same SFG structure. Fig. 1 shows the SFG structure of the traditional row-column 4 by 4 2-D LI/BO FFT implementation. Fig. 2 shows the LI/BO SFG structure of the 4 by 4 mapped 2-D FFT and 16-point 1-D FFT. The inputs, outputs, and twiddle factors are indicated upper for the 2-D case and lower for the 1-D case. The index "i" shown in the figure denotes the twiddle factor W_{16}^i. It can be seen that the twiddle factor matrices are the same for the k-th stage of the row FFT, column FFT, and 1-D FFT. The twiddle factor addressing sequence can be obtained from (17). The input and output interconnections are the same for the 1-D and 2-D FFTs with the same number of points and the addressing sequences can be obtained from (8) and (10).

VII. 2-D Bit-Reverse Input and Linear Output FFT

The vector-matrix form of the BI/LO 2-D FFT can be derived in the similar way as that of the LI/BO 2-D FFT. If the *TW sub r* and *TW sub c* of (30) are implemented by (4), then (30) can be transformed into the following form

$$\underline{X}_c = FG_{s_2}(BI(s_2)) * FG_{s_2-1}(BI(s_2-1)) * \cdots * FG_1(BI(1)) * P_T$$
$$* FG_{s_1}(BI(s_1)) * FG_{s_1-1}(BI(s_1-1)) * \cdots * FG_1(BI(1)) * \underline{x}_{br}^T \tag{35}$$

where $\underline{X}_c$ is an N-point vector in the column-major order of the linear output array and $\underline{x}_{br}^T$ is an N-point vector in the row-major order of the bit-reverse input array. Combining the transpose matrix P_T with the column FFTs and using Theorem 1, 2, and 3, (35) can become

$$\underline{X}_r^T = FG_{s_1+s_2}(BI(s_2)) * FG_{s_1+s_2-1}(BI(s_2-1)) * \cdots * FG_{s_1+1}(BI(1))$$
$$* FG_{s_1}(BI(s_1)) * FG_{s_1-1}(BI(s_1-1)) * \cdots * FG_1(BI(1)) * \underline{x}_{br}^T \tag{36}$$

where $\underline{X}_r^T$ is an N-point vector in the row-major order of the linear output array. As the LI/BO case by comparing (36) with (4), the N_1 by N_2 2-D FFT has the same interconnection structure as the N-point 1-D FFT. Moreover, the butterfly operation matrices are the same for the k-th stage of the row FFT, column FFT, and 1-D FFT. Fig. 3 shows the SFG structure of the traditional row-column 4 by 4 2-D BI/LO FFT implementation. Fig. 4 shows the LI/BO SFG structure of the 4 by 4 mapped 2-D FFT and 16-point 1-D FFT.

VIII. Algorithms Simulated by LH9124/LH9320

The proposed FFT algorithms with unified indexing have been implemented in the SMT's array processing chip set [2]. The LH9124 [5] is an execution unit with radix-2, radix-4, and radix-16 butterflies built in the highly pipelined data path. The radix-2, radix-4, and radix-16 butterflies can be implemented within 2, 4, and 16 cycles, respectively. The LH9320 [6] is a programmable address generator to provide the address patterns required by the LH9124. The unified indexing equations (7), (8), (10), (11), and (17) for the input/output data, twiddle factor, and bit-reverse sequences are built in the instruction set of the LH9320. The total number of machine cycles for an FFT implementation is calculated as

$$Cycles = \sum_{i=1}^{s} (N_i + PO_i) \tag{37}$$

where N_i and PO_i denote the data block size and the pipelined overhead of the i-th instruction, respectively. s is the number of instructions or butterfly stages.

TABLE I compares the performance of the 64K-point 1-D FFT with that of the 256 by 256 2-D FFT. It can be seen that both have the same performance because the data block size, the number of instructions, and the instruction overhead are all the same. It should be noted by the radix-2 butterfly instruction for the 2-D FFT operations that only 16 instructions are required for the proposed new 2-D FFT implementation instead of 4096 instructions required by the traditional 2-D FFT implementation. Therefore, the instruction pipelined overhead can be greatly reduced. With 25 nanoseconds machine cycle time, the 256 by 256 2-D complex FFT can be finished within 6.56 milliseconds.

IX. Conclusions

From the novel vector-matrix representation of the FFT algorithms, the paper derives the unified addressing for the 1-D and 2-D FFTs. Essentially, all the results extend to more general multidimensional FFTs in a straightforward manner. Table 2 shows the addressing equations implemented by FFT algorithms. (8) can be used as the input data addressing for 1-D to M-D FFTs and for both BI/LO and LI/BO FFTs. From the equivalent relationship of Theorem 3, the output data addressing can also be implemented by (8). It can be found in [7] that the M-D bit-reverse addressing can be implemented by the 1-D bit-reverse addressing described by (11). The twiddle factor sequences for the BI/LO FFTs are addressed by (7) and those for the LI/BO FFTs are addressed by (17).

There are several advantages for the proposed M-D FFT approaches. First, the program coding is simplified and the instruction overhead is reduced as discussed in Section VIII. Second, no data matrix transposition operations are required because the operations are combined with the input and output data transfer of the butterfly stage. Third, the chip architecture for the FFT addressing with arbitrary dimension is easy to define because the required addressing patterns are reduced and only 1-D indexing is necessary. The fourth advantage is especially for the block floating-point arithmetic.. There is no scaling problem by the proposed approach because the whole block data instead of sub-block data are calculated in each butterfly stage [2].

The proposed unified indexing can also be applied to an arbitrary mixed radix FFT algorithm [3,4]. The unified indexing concept for the M-D FFT implementation automatically solves the scaling problem for the block floating-point arithmetic. The concept can be extended to derive the efficient general DSP algorithms for block floating-point arithmetic such as IIR filtering, adaptive filtering, polyphase filter bank, and multi-channel DSP [8].

ACKNOWLEDGMENT

The author wishes to thank the System and Design groups of Sharp Microelectronics Technology for practically implementing the unified FFT algorithms to the array processing chip set.

REFERENCES

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

[2] C. Ju and M. Fleming, "Design concept of real-time array signal processors," *Proceeding of the International Conference on Signal Processing Applications and Technology*, Boston, pp.188-197, Nov. 1992.

[3] C. Ju, "Algorithms of defining 1-D indexing for M-D mixed radix FFT implementation," *Proceeding of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, May 1993.

[4] C. Ju, "Derivation and realization of fast Fourier transform," unpublished.

[5] *LH9124 Digital Signal Processor User's Guide*, Sharp Electronics Corporation.

[6] *LH9320 Address Generator User's Guide*, Sharp Electronics Corporation.

[7] C. Ju, *LH9124/LH9320 Fast Fourier Transform Application Note*, Sharp Electronics Corporation.

[8] C. Ju, "General DSP algorithms for block floating-point arithmetic," unpublished.

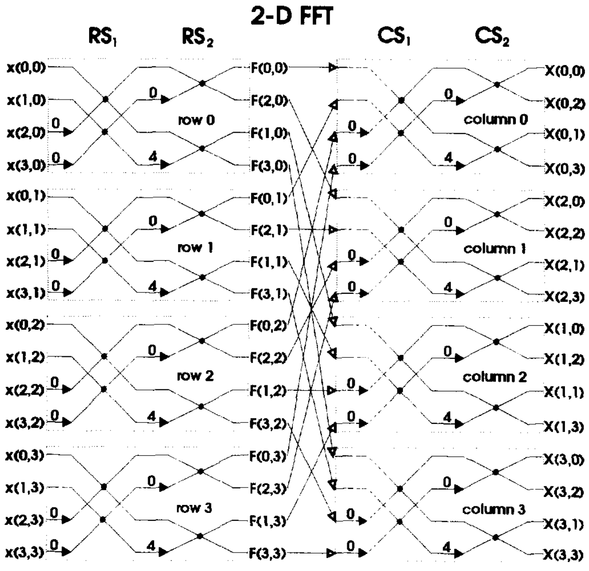

Fig. 1. Signal flow graph of traditional LI/BO 4 by 4 2-D FFT

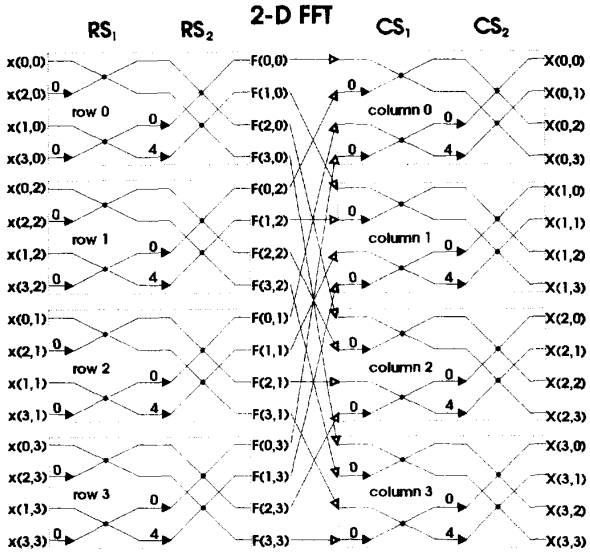

Fig. 3. Signal flow graph of traditional BI/LO 4 by 4 2-D FFT

TABLE I

UNIFIED FFT ALGORITHMS SIMULATED BY LH9124/LH9320

	64K-Point 1-D FFT			256 by 256 2-D FFT		
Butterfly	Stages	Cycles	msecs	Stages	Cycles	msecs
Radix-2	16	1048864	26.22	16	1048864	26.22
Radix-4	8	524432	13.11	8	524432	13.11
Radix-16	4	262416	6.56	4	262416	6.56

TABLE II

SUMMARY OF ADDRESSING EQUATIONS OF THE FFT ALGORITHMS

	1-D FFT		2-D FFT		M-D FFT	
Addressing	BI/LO	LI/BO	BI/LO	LI/BO	BI/LO	LI/BO
Inputs	(8)	(8)	(8)	(8)	(8)	(8)
Outputs	(10)	(10)	(10)	(10)	(10)	(10)
Bit-Reverse	(11)	(11)	(11)	(11)	(11)	(11)
Twiddle Factors	(7)	(17)	(7)	(17)	(7)	(17)

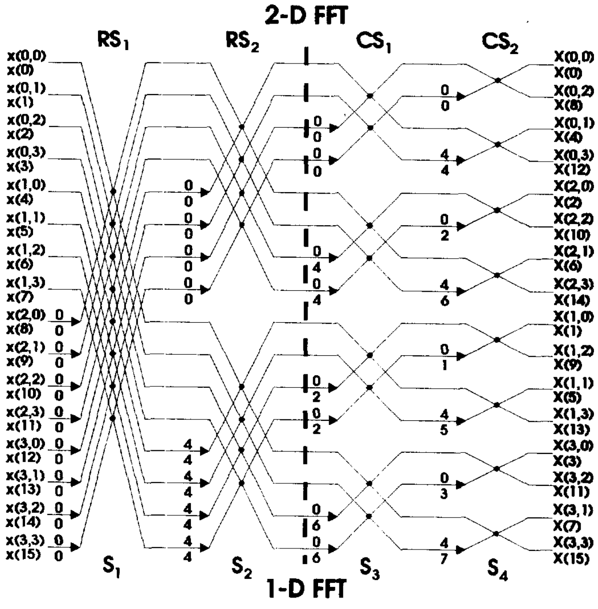

Fig. 2. Signal flow graph of LI/BO 4 by 4 2-D mapped & 16-point 1-D FFTs

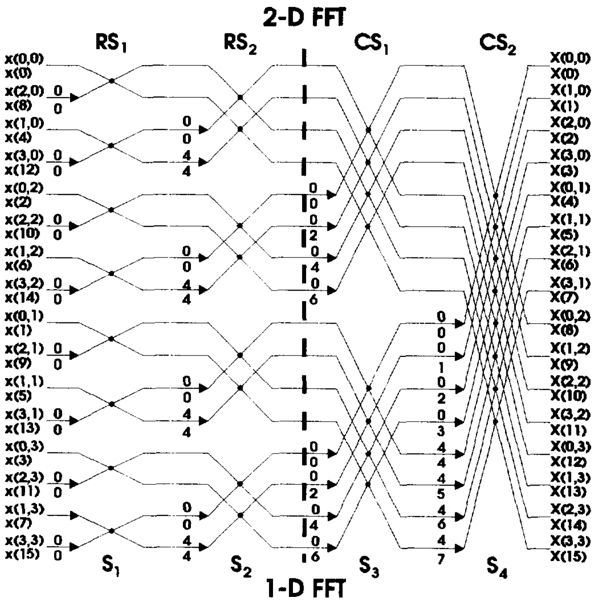

Fig. 4. Signal flow graph of BI/LO 4 by 4 2-D mapped & 16-point 1-D FFTs