Efficient systolic solution for a new prime factor discrete Hartley transform algorithm

P.K. Meher
J.K. Satapathy
G. Panda

Abstract: Recently, a novel systolic structure has been proposed for the computation of DFT for transform length $N = 4M$, M being prime to 4. In this paper, we have proposed a similar structure for the computation of DHT by prime factor decomposition. A new recursive algorithm is also proposed for computing DHT using a linear systolic array of cordic processing elements. The proposed structure has nearly the same hardware requirement as that of the corresponding DFT structure for real-valued data; but it yields significantly higher throughput.

1 Introduction

The discrete Hartley transform (DHT) of a real-valued sequence $\{x(n)\}$ is defined [1] as

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)\left(\cos \frac{2\pi kn}{N} + \sin \frac{2\pi kn}{N}\right)$$

$$\text{for } k = 0, 1, \ldots, N-1 \quad (1)$$

where $1/N$ is the scale factor. *Note:* $1/N$ is ignored in the rest of the paper.

This transform has been emerging as an useful alternative to the discrete Fourier transform (DFT) to avoid the complex arithmetic in various signal processing applications. Several attempts have therefore been made to develop efficient DHT algorithms to increase the computational speed but, only a few reports have been made so far of hardware implementation of the DHT. Boussakta and Holt [2] have proposed a method for computing the DHT by Fermat number transform (FNT) using a VLSI chip. Chakrabarti and Ja'Ja' [3] have presented a bit serial solution for small DHT modules. Also they have suggested a systolic architecture for the prime factor DHT [4] which is computed via four temporary outputs.

Recently, Jones [5] has proposed a novel systolic structure for the computation of DFT for transform length $N = 4M$, M being prime to 4. In this paper, we have proposed a structure similar to that of Reference 5 for the computation of prime factor DHT. The scheme of

prime factor decomposition used in this paper is different from that used in Reference 4. A new recursive algorithm is also proposed for efficient computation of the DHT by a linear systolic array of cordic processing elements (PE). It is shown that the proposed structure has nearly the same hardware requirement as the corresponding DFT structure [5] for real-valued data; but it yields significantly higher throughput.

2 The recursive DHT algorithm and its systolic solution

The DHT defined by eqn. 1 may be expressed as

$$X(k) = x(0) + \sum_{n=1}^{(N-1)/2} [\text{Re Pt } \{z(n)\alpha^{kn}\} + \text{Im Pt } \{z(n)\alpha^{kn}\}]$$

$$(2a)$$

and

$$X(N-k) = x(0) + \sum_{n=1}^{(N-1)/2} [\text{Re Pt } \{z(n)\alpha^{-kn}\}$$
$$+ \text{Im Pt } \{z(n)\alpha^{-kn}\}] \quad (2b)$$

where $z(n) = x(n) + jx(N-n)$, $\alpha = e^{j2\pi/N}$ and $X(N) = X(0)$ for $k = 0, 1, \ldots, (N-1)/2$, $j = \sqrt{(-1)}$ and N is an odd number. *Note:* The upper limit of the summation index n of eqn. 2 would be $N/2$ when N is even.

The DHT components given by eqn. 2 may be written as

$$X(k) = \text{Re Pt } Y(k) + \text{Im Pt } Y(k) \quad (3a)$$

and

$$X(N-k) = \text{Re Pt } Y(N-k) + \text{Im Pt } Y(N-k) \quad (3b)$$

where

$$Y(k) = \left(\cdots\left(\left(z\left(\frac{N-1}{2}\right)\alpha^k + z\left(\frac{N-3}{2}\right)\right)\alpha^k\right.\right.$$
$$\left.\left. + z\left(\frac{N-5}{2}\right)\right)\alpha^k + \cdots + z(1)\right)\alpha^k + x(0) \quad (4a)$$

and

$$Y(N-k) = \left(\cdots\left(\left(z\left(\frac{N-1}{2}\right)\alpha^{-k} + z\left(\frac{N-3}{2}\right)\right)\alpha^{-k}\right.\right.$$
$$\left.\left. + z\left(\frac{N-5}{2}\right)\right)\alpha^{-k} + \cdots + z(1)\right)\alpha^{-k}$$
$$+ x(0) \quad (4b)$$

$Y(k)$ and $Y(N-k)$ of eqns. 4a and 4b, respectively, for each k $(k = 0, 1, 2, \ldots, (N-1)/2)$ can be obtained by

$(N + 1)/2$ identical recursions; where each recursion is composed of a pair of complex multiplications followed by a complex addition.

It is shown [6] that a vector P_m may be obtained from a vector P_0 by m successive phase rotations, given by

$$P_m = M_m M_{m-1} \cdots M_2 M_1 P_0 \qquad (5a)$$

where

$$M_i = \begin{bmatrix} 1 & \delta_i \\ -\delta_i & 1 \end{bmatrix}$$

$$P_0 = [x_0, y_0]^T$$

and

$$P_m = [x_m, y_m]^T \qquad (5b)$$

such that

$$x_m = K(x_0 \cos \alpha + y_0 \sin \alpha) \qquad (5c)$$

and

$$y_m = K(y_0 \cos \alpha - x_0 \sin \alpha) \qquad (5d)$$

for

$$\alpha = \sum_{i=0}^{m-1} \alpha_i \qquad (5e)$$

$$K = \prod_{i=0}^{m-1} (1 + \delta_i^2)^{1/2} \qquad (5f)$$

and

$$\alpha_i = \tan^{-1} \delta_i \qquad (5g)$$

It may be seen from eqns. 5c and 5d that

$$(x_0 + jy_0)e^{-j\alpha} = \frac{1}{K}(x_m + jy_m) \qquad (6a)$$

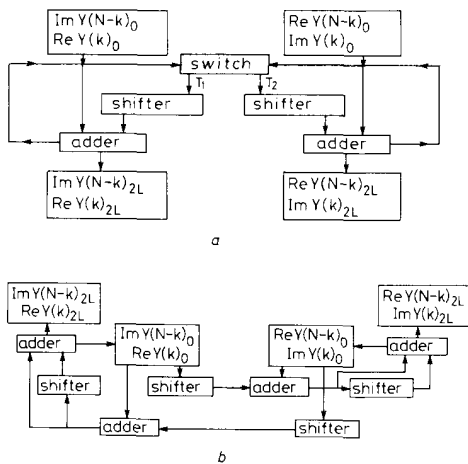$$(y_0 + jx_0)e^{j\alpha} = \frac{1}{K}(y_m + jx_m) \qquad (6b)$$

a

b

Fig. 1 *Cordic circuits for processing elements*

a cordic circuit I
b cordic circuit II
$Y(k)_0$ and $Y(N - k)_0$ are inputs
$Y(k)_{2L}$ and $Y(N - k)_{2L}$ are outputs after 'L' iterations
For rotations: T_1 is Im $Y(k)$ followed by Re $Y(N - k)$ and T_2 is Re $Y(k)$ followed by Im $Y(N - k)$
For scalings: T_1 is Re $Y(k)$ followed by Im $Y(N - k)$ and T_2 is Im $Y(k)$ followed by Re $Y(N - k)$

Eqns. 5 and 6 imply that the complex multiplications of the form $(x_0 + jy_0)e^{-j\alpha}$ as well as $(y_0 + jx_0)e^{j\alpha}$ may be computed via phase rotations and scaling. By setting δ_i to be a power of 2, phase rotations can be achieved by repeated shift-add operations. It is shown [7] that scaling can also be performed by repeated shift-add operations. The pair of complex multiplications required in each recursion to compute $Y(k)$ and $Y(N - k)$ may therefore be performed in a cordic circuit by the same sequence of shift-add operations. Two different cordic circuits to be used for these multiplications are depicted in Fig. 1. The systolic array consisting of $(N + 1)/2$ cordic PEs to compute N-point DHT is shown in Fig. 2a. The function of the $(k + 1)$th PE is described in Fig. 2b. The addition of the real parts with the corresponding imaginary parts of the outputs of the $(k + 1)$th PE yields the kth and $(N - k)$th DHT components.

3 The prime factor DHT algorithm and its implementation

For the transform length $N = N_1 \times N_2$, where N_1 and N_2 are relatively prime, the indices k and n in eqn. 1 may be mapped into pairs of indices (k_1, k_2) and (n_1, n_2), respectively, according to the following equations [8]

$$k = (k_1 N_2 s_1 + k_2 N_1 s_2) \bmod N \qquad (7a)$$

$$n = (n_1 N_2 + n_2 N_1) \bmod N \qquad (7b)$$

for

$$k_1 \quad \text{and} \quad n_1 = 0, 1, \ldots, (N_1 - 1)$$

and

$$k_2 \quad \text{and} \quad n_2 = 0, 1, \ldots, (N_2 - 1)$$

where

$$N_2 s_1 = 1 \bmod N_1 \qquad (8a)$$

and

$$N_1 s_2 = 1 \bmod N_2 \qquad (8b)$$

As a result, eqn. 1 assumes a two-dimensional form

$$X(k_1, k_2) = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x(n_1, n_2)\left[\cos 2\pi\left(\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2}\right) \right.$$
$$\left. + \sin 2\pi\left(\frac{k_1 n_1}{N_1} + \frac{k_2 n_2}{N_2}\right)\right] \qquad (9)$$

The arguments of sine and cosine functions of eqn. 9 may be expanded to yield

$$X(k_1, k_2) = \sum_{n_2=0}^{N_2-1} w(k_1, n_2)$$
$$\times \left[\cos 2\pi \frac{k_2 n_2}{N_2} + \sin 2\pi \frac{k_2 n_2}{N_2}\right] \qquad (10a)$$

where

$$w(k_1, n_2) = u(k_1, n_2) + v(k_1, N_2 - n_2)$$
$$\text{for } 1 \leqslant n_2 \leqslant N_2 - 1 \qquad (10b)$$

$$u(k_1, n_2) = \sum_{n_1=0}^{N_1-1} x(n_1, n_2) \cos 2\pi \frac{k_1 n_1}{N_1} \qquad (10c)$$

$$v(k_1, n_2) = \sum_{n_1=0}^{N_1-1} x(n_1, n_2) \sin 2\pi \frac{k_1 n_1}{N_1} \qquad (10d)$$

and

$$w(k_1, 0) = u(k_1, 0) + v(k_1, 0) \qquad (10e)$$

$u(k_1, n_2)$ and $v(k_1, n_2)$ in eqns. 10c and 10d represent the even and the odd parts of the DHT of elements of the n_2th column of $x(n_1, n_2)$, respectively. $w(k_1, 0)$ in eqn. 10e denotes the DHT of the zeroth column of $x(n_1, n_2)$.

into the next section. The second section of the structure is a linear array of $(N_2 + 1)/2$ PEs (discussed in Section 2).

If each of the complex adders and delay cells, as well as the PEs, are driven by the same clock, the DHT of the first row of $w(k_1, n_2)$ can be obtained in $(N_2 + 3)$ time steps. One time step is considered to be the time taken by
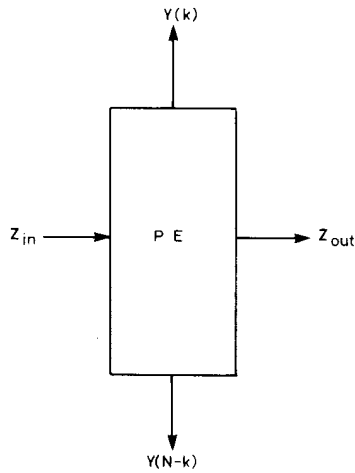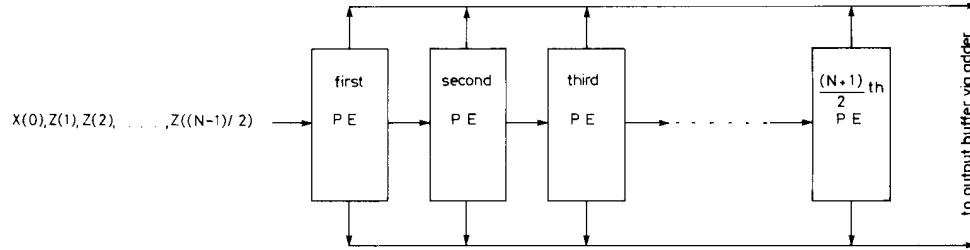

a

b

Fig. 2 *Linear systolic array for computation of DHT*

a Linear array
b function of $(k + 1)$th processing element
Algorithm:
if count $= (N + 1)/2$ then
$Y(K) = Z(K)$
$Y(N - k) = Z(N - k)$
$Z(K) = 0$
$Z(N - k) = 0$
count $= 0$
else
$Z(k) = Z_{in} + Z(K)\alpha^k$
$Z(N - k) = Z_{in} + Z(N - k)\alpha^{-k}$
$Z_{out} = Z_{in}$
count $=$ count $+ 1$
endif

The proposed structure for a two-factor DHT, where $N_1 = 4$ and N_2 is prime to 4 is shown in Fig. 3A. It consists of two sections. The first section of the structure consists of 12 complex adders and three delay cells (depicted in Fig. 3B). Four rows of $w(k_1, n_2)$ are pipelined out from it, and get queued through a buffer and a channel selector, to be further pipelined in proper order

a PE to perform both the multiplications required to be computed in each recursion. The successive output corresponding to the successive rows of $w(k_1, n_2)$ as input can be obtained from the linear array in $(N_2 + 1)/2$ time steps interval. Thus, the DHTs of all four rows of $w(k_1, n_2)$ are computed in a total of $[(5/8)N + 9/2]$ time steps; and the DHT of successive sets of N-point data may be obtained in an interval of $2(N_2 + 1) = (N/2 + 2)$ time steps.

4 Throughput and hardware considerations

As mentioned earlier, during each recursion, a PE performs two complex multiplications, where each of these multiplications is implemented through L iterations. Each iteration comprises one phase rotation followed by a scaling. Furthermore, either a rotation or a scaling involves two additions and two shifts. All these additions and shifts are performed by two pairs of parallel adders and shifters if cordic circuit I (Fig. 1a) is used by PEs. The total computational load per pair of adder and shifter then comes out to be $4L$ additions and $4L$ shifts. To maintain a precision of L correct bits at the output, the input data needs additional $\log_2 L$ bits as guard bits to account for the internal cordic arithmetic [9]. Assuming that addition of two single bits takes one microcycle, a total of $(L + \log_2 L)$ microcycles are necessary for the addition of two $(L + \log_2 L)$ bit words. It may be noted here that the additions corresponding to one multiplication and the shift operations corresponding to the other multiplication are performed simultaneously. Hence, no extra time is required for the shift operations. The total time required per recursion in cordic circuit I may then be calculated to be $4L(L + \log_2 L)$ microcycles. It is stated in Section 3 that the first N-point DHT requires $[(5/8)N + 9/2]$ recursions (or time steps) and the successive N-point DHTs require $(N/2 + 2)$ recursions. Thus the total time required to compute an N-point DHT is given by

$$T = (2N + 8)(L + \log_2 L)L \text{ microcycles} \qquad (11a)$$

If the transform length $N \gg 4$, this may be approximated to

$$T = 2NL(L + \log_2 L) \text{ microcycles} \qquad (11b)$$

In cordic circuit II (Fig. 1b), the additions and shifts are performed by four pairs of parallel adders and shifters. Therefore, the duration of each time step in cordic circuit II is half that of cordic circuit I (i.e. $2L(L + \log_2 L)$

microcycles). Using cordic circuit II in the PEs of the linear array, the computation time of N-point DHT may therefore be reduced to

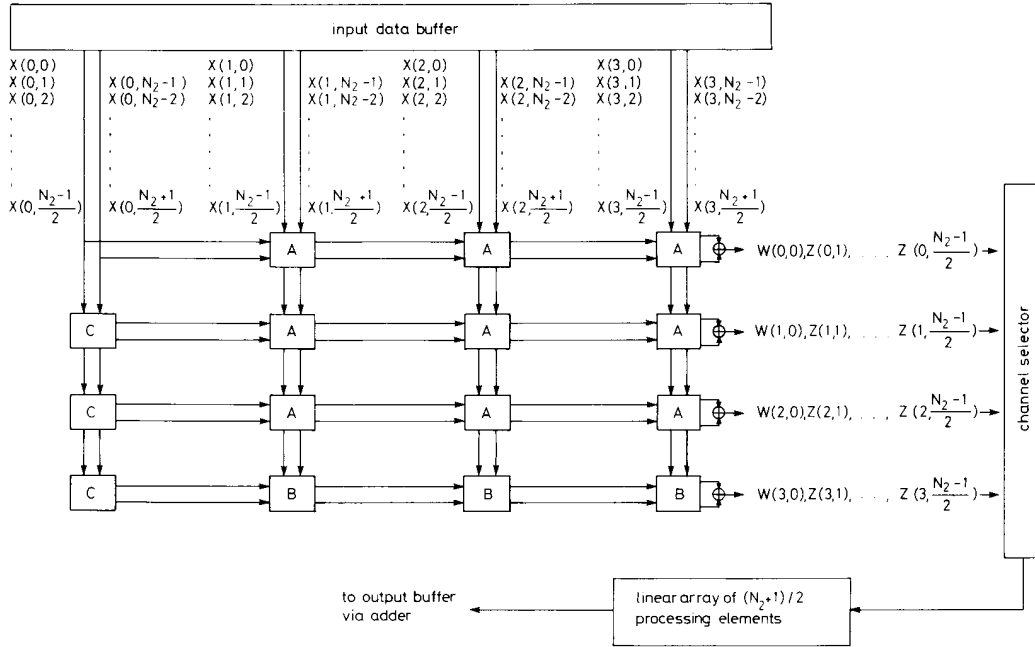$$T = NL(L + \log_2 L) \text{ microcycles} \qquad (12)$$

Fig. 3A *Systolic structure for two-factor DHT computation*
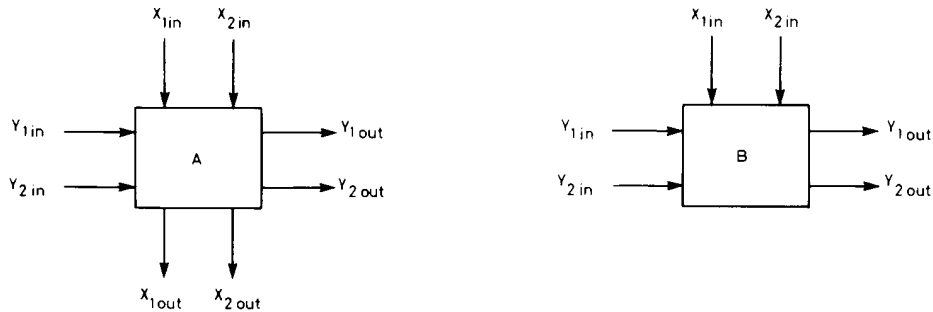
$Z(k, n) = W(k, n) + jW(k, N_2 - n)$

Fig. 3B *Function of complex adders and delay cells*

Adders are tailored to compute $W(k, n)$ and $W(k, N_2 - n)$ according to eqn. 10 for $N_1 = 4$ by choosing proper values of a, b, c, d and e

a and $b \in (1, 0, -1)$

c, d and $e \in (1, 0, -1, j, -j)$

for internal adder cells A

$Y_{1out} = Y_{1in} + aX_{1in} + bX_{2in}$

$Y_{2out} = cY_{2in} + dX_{1in} + eX_{2in}$

$X_{1out} = X_{1in}$

$X_{2out} = X_{2in}$

for boundary adder cells B

$Y_{1out} = Y_{1in} + aX_{1in} + bX_{2in}$

$Y_{2out} = cY_{2in} + dX_{1in} + eX_{2in}$

for delay cells C

$X_{1out} = X_{1in} = Y_{1out}$

$X_{2out} = X_{2in} = Y_{2out}$

The DFT of real-valued data, and hence the DHT, may also be computed by the DFT structure [5] using the following two methods.

Method 1: For real-valued input, the DFT may be obtained from the first $(N_2 + 1)/2$ components of each 'column DFT' [5] of the intermediate output. Therefore, only $(N_2 + 1)/2$ PEs may be used in the linear array of the structure [5]. The modified structure would then yield the first $(N_2 + 1)/2$ components of the first column DFT after $[3 + (N_2 - 1)/2 + N_2]$ time steps and those of the successive column DFTs in N_2 time steps interval. The first N-point DFT may thus be computed in $[3 + (N_2 - 1)/2 + 4N_2]$ time steps, and the successive DFTs may be obtained in the interval of $4N_2$ time steps, where each time step is $L(3L + 2 \log_2 L)$ microcycles [5]. The computation time of the DFT of an N-point real-valued sequence is therefore given by

$$T = NL(3L + 2 \log_2 L) \text{ microcycles} \tag{13}$$

Method 2: When the input sequence is real, the third N_2 point 'column DFT' of the intermediate output may be obtained from those of the second N_2 point 'column DFT'. Therefore, only three column DFTs are required to be computed in the final stage so that the first set of final DFT output may be computed in $[3 + (N_2 - 1) + 3N_2]$ time steps and the successive DFTs in $3N_2$ time steps interval. Hence, the computation time of the DFT of an N-point real-valued sequence may be given by

$$T = (3/4)NL(3L + 2 \log_2 L) \text{ microcycles} \tag{14}$$

The hardware requirement of the first section of the proposed structure is the same as that of the DFT structure [5] for Method 1. For Method 2, however, three complex adders and a delay cell may be avoided in the first section of the structure [5]; because only three column DFTs are required to be computed in this case at the output stage. Moreover, the amount of hardware used by the second section of the proposed structure, using cordic circuit I and cordic circuit II is nearly the same as those of the DFT structure for real valued data by Method 1 and Method 2, respectively.

5   Results and discussion

An efficient systolic structure is proposed for computing the DHT by prime factor decomposition for transform length $N = N_1 \times N_2$, where $N_1 = 4$ and N_2 is prime to 4. A recursive algorithm is also proposed to compute N point DHT by a systolic array of $(N + 1)/2$ cordic processing elements. The computation times of the prime factor DHT by the proposed structure, and the times by the DFT structure [5] for real-valued data using two different methods, are plotted in Fig. 4. It is found that the proposed structure requires significantly less computation time than the other. It is also observed that the hardware requirement of the proposed structure, using cordic circuit I and cordic circuit II is nearly the same as that of the DFT structure [5] for Method 1 and Method 2, respectively. The adders and shifters of the cordic cir-

cuits are utilised more efficiently by the proposed scheme compared to that of Reference 5. This is achieved by performing the additions for one multiplication, and the shift operations for the other multiplication simultaneously. Cordic circuit I of the proposed structure is similar to the cordic circuit proposed in Reference 5; but cordic circuit II is different in that it does not require any control switch to regulate the data.
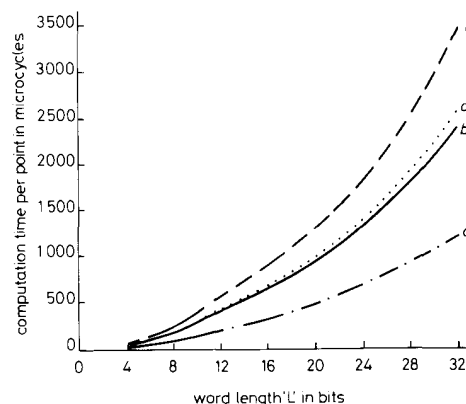

Fig. 4 *Comparison of computation time of DHT by proposed structure with that of DFT [5] of real-valued data*

a and *b*: proposed structure using cordic circuit I and cordic circuit II, respectively
c and *d*: DFT structures [5] if modified for real data to compute by Method 1 and Method 2, respectively

The higher throughput obtained by the proposed DHT structure over the corresponding DFT structure [5] for real-valued data is mainly due to the scheme of prime factor decomposition, and to the efficient use of the cordic circuits by the recursive DHT algorithm.

6   References

1 BRACEWELL, R.N.: 'The discrete Hartley transform', *J. Opt. Soc. Am.*, 1983, **73**, pp. 1832–1835
2 BOUSSAKTA, S., and HOLT, A.G.J.: 'Calculation of the discrete Hartley transform via the Fermat number transform using a VLSI chip', *IEE Proc. G*, 1988, **135**, (3), pp. 101–103
3 CHAKRABARTI, C., and JA'JA', J.: 'Optimal systolic designs for computing DHT and DCT', in BRODERSEN, R., and MOSCOVITZ, H. (Eds.): 'VLSI signal processing' (IEEE Press, New York, 1988), Vol. 3, pp. 411–422
4 CHAKRABARTI, C., and JA'JA', J.: 'Systolic architecture for the computation of the discrete Hartley and discrete cosine transforms based on prime factor decomposition', *IEEE Trans.*, 1990, **COMP-39**, pp. 1359–1368
5 JONES, K.J.: 'High-throughput, reduced hardware systolic solution to prime-factor discrete Fourier transform algorithm', *IEE Proc. E*, 1990, **137**, (3), pp. 191–196
6 WILLEY, T., CHAPMAN, R., YOHO, H., DURANI, T.S., and PREIS, D.: 'Systolic implementations for deconvolution, DFT and FFT', *IEE Proc. F*, 1985, **132**, (6), pp. 466–472
7 KUNG, S.Y., WHITEHOUSE, H.J., and KAILATH, T. (Eds.): 'VLSI and modern signal processing' (Prentice-Hall, 1985)
8 BURRUS, C.S.: 'Index mappings for multidimensional formulation of DFT and convolution', *IEEE Trans.*, 1977, **ASSP-25**, pp. 239–242
9 JONES, K.J.: '2D systolic solution to discrete Fourier transform', *IEE Proc. E*, 1989, **136**, (3), pp. 211–216