Chapter 3

Walsh Transformation

IIIA Definition of the Walsh transform

The definition for the Walsh functions given in Section IIA may be restated by saying that every function $f(t)$ which is integrable (in the Lebesque sense) is capable of being represented by a Walsh series defined over the open interval (0, 1) as

$$x(t) = a_0 + a_1 \,\mathrm{WAL}(1, t) + a_2 \,\mathrm{WAL}(2, t) + \ldots \tag{3.1}$$

where the coefficients are given by

$$a_k = \int_0^1 f(t) \,\mathrm{WAL}(k, t)\, dt \tag{3.2}$$

From this we are able to define a transform pair,

$$f(t) = \sum_{k=0}^{\infty} F(k) \,\mathrm{WAL}(k, t) \tag{3.3}$$

$$F(k) = \int_0^1 f(t) \,\mathrm{WAL}(k, t)\, dt \tag{3.4}$$

This definition applies to a continuous function limited in time over the interval $0 \geqslant t \geqslant 1$. For numerical use it is convenient to consider a discrete series of N terms set up by sampling the continuous functions at N equally spaced points over the open interval (0, 1). In order that the properties of the continuous and discrete systems should correspond then we must make N equal to a power of 2, i.e. $N = 2^p$.

he integration shown in equation (3.4) may then be replaced by summa-
, using the trapezium rule on N sampling points, x_i, and we can write the
te discrete Walsh transform pair as

$$X_n = \frac{1}{N} \sum_{i=0}^{N-1} x_i \, \mathrm{WAL}(n, i)$$

$$n = 0, 1, 2 \ldots N-1 \tag{3.5}$$

$$x_i = \sum_{n=0}^{N-1} X_n \, \mathrm{WAL}(n, i)$$

$$i = 0, 1, 2 \ldots N-1 \tag{3.6}$$

ilar transforms, $X_c(k)$ and $X_s(k)$ can be obtained for a time series, x_i
ng Harmuth's CAL and SAL functions viz.

$$X_c(k) = \frac{1}{N} \sum_{i=0}^{N-1} x_i \, \mathrm{CAL}(k, i) \tag{3.7}$$

$$X_s(k) = \frac{1}{N} \sum_{i=0}^{N-1} x_i \, \mathrm{SAL}(k, i) \tag{3.8}$$

As with their functions the transforms are linear so that if

$$x_i \leftrightarrow X_n \quad \text{and} \quad y_i \leftrightarrow Y_n$$

n

$$ax_i + by_i \leftrightarrow aX_n + bY_n$$

ere a and b are real constants and $\leftrightarrow$ denotes a transform operator.
Also the transform is symmetrical (equation 2.2). Since $\mathrm{WAL}(n, i)$ is
metrical about the mid-point of the sequence, $i = 0, 1, 2 \ldots N-1$ when
s even, and anti-symmetric when n is odd, then it also follows from
ation (2.1) that a sequence x_i will have a transform composed only of
n-order Walsh function coefficients (CAL function) if it is symmetric
ut its mid-point and be composed only of odd-order (SAL function)
fficients if the series is inversely symmetric.

Comparison with the discrete Fourier transform

e transform and its inverse given by equations (3.5) and (3.6) may be
ained by matrix multiplication using the digital computer. Since the

matrices are symmetrical for the Walsh transform (unlike the Fourier transform) then both transform and inverse transform are identical, except for a scaling factor, $1/N$.

If we compare equation (3.5) with the corresponding discrete Fourier transform

$$X_f = \frac{1}{N} \sum_{i=0}^{N-1} x_i \exp(-j2\pi if/N)$$
$$f = 0, 1, 2 \ldots N-1 \qquad (3.9)$$

we note that whilst WAL(n, i) is real and limited to values ± 1 the kernal, $K = \exp(-j2\pi if/N)$, is complex and can assume N different values for each coefficient. As a direct consequence of this the Walsh transform proves considerably easier and faster to calculate using digital methods.

It may also be noted that since the sine and cosine functions cannot be represented exactly by a finite number of bits then a source of truncation noise is introduced by the discrete Fourier transform which involves repeated multiplication by a complex number. The Walsh transform, on the other hand, involves only addition and subtraction and precise representation is possible, so that the transform is not a noisy one.

IIIC Effects of circular time shift

The discrete Fourier transform is invariant to the phase of the input signal so that the same spectral decomposition can be obtained independently of the phase or circular time-shift of the input signal. This is not the case for the discrete Walsh transform.

Walsh transform signals conform to Parseval's theorem where the energy in the time and sequency domains are shown to be equivalent. Expressing this energy in terms of the sum of the squared coefficients for a discrete time series, x_i, and its transformed coefficients, X_n, a special and simplified case of Parseval's theorem may be expressed as

$$\frac{1}{N} \sum_{i=0}^{N-1} (x_i)^2 = \sum_{n=0}^{N-1} (X_n)^2 \qquad (3.10)$$

If the transformed coefficients, X_n, are expressed terms of CAL and SAL transformed coefficients (see Section VE), we can also write

$$\frac{1}{N} \sum_{i=0}^{N-1} x_i^2 = X_i^2(0, t) + \sum_{k=1}^{N/2-1} (X_c^2(k, t)) + \sum_{k=1}^{N/2-1} (X_s^2(k, t)) + X_s^2(N/2, t))$$
$$k = 1, 2 \ldots (N/2-1)$$
$$i = 0, 1, 2 \ldots (N-1)$$
$$0 \leqslant t \leqslant 1 \qquad (3.11)$$

ere $X_c(k, t)$ and $X_s(k, t)$ are the CAL and SAL function coefficients for x_i. ow x_i is circularly-shifted forming

$$y_i = x_{i+p} \tag{3.12}$$

n, as noted by Whelchel and Guinn[1],

$$Y_c^2(k, t) + Y_s^2(k, t) \neq X_c^2(k, t) + X_s^2(k, t) \tag{3.13}$$

wever, Pichler[2] has shown that if the time-shift is obtained through a dic translation,

$$z_i = x_{i \oplus p} \tag{3.14}$$

ere $\oplus$ indicates Modulo-2 addition for the binary representations of i p) then

$$Z_c^2(k, t) + Z_s^2(k, t) = X_c^2(k, t) + X_s^2(k, t) \tag{3.15}$$

the sequency spectrum is invariant under dyadic time-shift of the input al. This also indicates that a relationship must exist between the uency values obtained, namely

$$Z(k, t) = X(k \oplus p, t) \tag{3.16}$$

is shown in the following example.

0	0.663	0.063	0	0	−0.263	0.025	0
0	−0.052	−0.006	0	0	−0.126	0.013	0
0	−0.013	−0.002	0	0	0.006	0	0
0	−0.025	−0.002	0	0	−0.062	0.006	0

TABLE 3.1 (a). Walsh transform coefficients for a simple sine waveform. $N = 32$.

0	−0.063	0.663	0	0	0.025	0.263	0
0	0.006	−0.052	0	0	0.013	0.126	0
0	0.001	−0.013	0	0	0	−0.006	0
0	0.002	−0.025	0	0	0.006	0.062	0

TABLE 3.1 (b). Walsh transform coefficients for the sine waveform shifted by 90°.

able 3.1 shows the discrete Walsh transform of a single cycle of a pled sinusoidal waveform having 32 values, which is compared with a nsform of the same waveform circularly-shifted through $\pi/2$ radians. If take the shift index, p, as $p = \log_2 8 = 3$, then we obtain the following ex values for the sequency coefficients k and $k \oplus p$:

Conversion between discrete Walsh and Fourier transformation

n equations (3.6) and (3.9) we can express a sampled time series, x_i, of N, in terms of its Fourier transform series, X_f, or its Walsh transform s, X_n, viz.

$$x_i = \sum_{f=0}^{N-1} X_f \exp(j2\pi if/N) = \sum_{n=0}^{N-1} X_n \, \mathrm{WAL}(n, i) \tag{3.26}$$

re $f, i, n = 0, 1 \ldots (N-1)$.
ence conversion from Fourier to Walsh transformation is

$$X_n = \frac{1}{N} \sum_{i=0}^{N-1} x_i \, \mathrm{WAL}(n, i)$$

$$= \frac{1}{N} \sum_{f=0}^{N-1} X_f \left(\sum_{i=0}^{N-1} \mathrm{WAL}(n, i) \exp(j2\pi if/N) \right) \tag{3.27}$$

conversion from Walsh to Fourier transformation is

$$X_f = \frac{1}{N} \sum_{i=0}^{N-1} x_i \exp(-j2\pi if/N)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} X_n \left(\sum_{i=0}^{N-1} \mathrm{WAL}(n, i) \exp(-j2\pi if/N) \right) \tag{3.28}$$

limitation here is, of course, that N be sufficiently large so that an rate representation of the sampled time series is possible from a limited ber of terms. This was discussed in Section IIF where the number of s required was shown to be dependent on the shape of the original inuous time function, $f(t)$.

Summary of Walsh transform characteristics

characteristics of the Walsh function and its transform described in the eding text are compared with the Fourier function and summarised in e 3.3. The relationships given refer to a discrete time series where the ber of terms N is expressed as a power of 2.
we consider the discrete Walsh transform, given by equation (3.5), as a sh function matrix obtained by sampling the continuous function of tion (3.4) then the relationship

$$\mathbf{W} \cdot \mathbf{W}^{-1} = \mathbf{W}^{-1} \cdot \mathbf{W} = N \cdot \mathbf{I} \tag{3.29}$$

ys applies.

Here $\mathbf{W}$ and $\mathbf{W}^{-1}$ are the direct and transposed Walsh function matri
and $\boldsymbol{I}$ is an identity matrix (i.e. an N by N unit matrix). This derives fro
general property of the Hadamard matrix (discussed in Chapter 2) whic
related simply to the Walsh matrix by means of a permutation of the rows
a symmetrical matrix. The relationship given in equation (3.29) is fun
mental to the derivation of the fast Walsh transform which is examined in
next section.

IIIH The fast Walsh transform

The basis for efficient implementation of the transformations discussed h
is the high degree of redundancy present in the transform matrix. If
redundancy can be removed by using matrix factorisation then the efficie
of transformation will be improved. Such a technique was described
Good[5] and has resulted in the development of a fast Fourier transfo
(F.F.T.)[6]. Similar computation algorithms are available for many ot
orthogonal transformations including the Walsh and Haar transforms.

Computation of the discrete Walsh transform given in equation (3
requires N^2 mathematical operations, where an operation is either
addition or a subtraction. Using matrix factorisation techniques
algorithm may be found to enable the transformation to be carried out us
only $N \log_2 N$ operations. This is known as the fast Walsh transfo
(F.W.T.).

A procedure for obtaining a fast Walsh transform algorithm is describ
in this section. This follows the well-known Cooley–Tukey algorithm us
for fast Fourier transformation[6] and has similar limitations. In particular
value of N must be a power of two which will be seen later to be an essent
condition for the factorisation method used in the algorithm. It is wo
noting that certain fast Fourier transform programs may be converted to
Walsh transform by simply setting all the trigonometric values to ± 1 a
removing the complex part of the transformation (since the Walsh transfo
mation is a real one). The ordering of the Walsh functions obtained with t
method will, however, correspond to natural order or bit-reversed natu
order.

IIIH1 Sequency-ordered transforms

A derivation of a fast Walsh transform algorithm having sequency-order
coefficients is most conveniently obtained from the continued produ
representation given by Pratt *et al.*[7]. This was defined earlier for a series

$= 2^p$ terms as

$$\mathrm{WAL}(n, i) = \prod_{r=0}^{p-1} (-1)^{n_{p-1-r}(i_r + i_{r+1})}$$

$$i, n = 0, 1, 2 \ldots N-1$$

$$r = 0, 1, 2 \ldots p \qquad (3.30)$$

ere i, n are expressed in terms of their binary digits, i_r and n_r, e.g.

$$i = (i_{p-1}, i_{p-2} \ldots i_1, i_0)_2 \qquad (3.31)$$

The algorithm is developed by substitution of equation (3.30) into equa-
n (3.5) and factorising the calculation into p separate stages. Carrying out
is substitution we obtain a product-sum expression for the discrete Walsh
ansform as,

$$X_n = \sum_{i=0}^{N-1} x_i \,\mathrm{WAL}(n, i) = \prod_{r=0}^{p-1} \sum_{i_r=0}^{1} (-1)^{n_{p-1-r}(i_r + i_{r+1})} x_{(i_{p-1} \ldots i_0)} \qquad (3.32)$$

eglecting scaling by N).

Here x_i is expressed as $x_{(i_{p-1} \ldots i_o)}$ and X_n is expressed as $X_{(n_{p-1} \ldots n_0)}$ where i_p
d n_p are the binary bits of i and n with $r = 0, 1, 2 \ldots p$.

The calculation of the fast Walsh transform is carried out in a series of
ages, one stage for each power of 2 for N. The first calculation stage is to
erive a partial transformation series, A_n, from the input series, x_i, by placing
$= 0$ in equation (3.32) giving

$$A(n_{p-1}, i_{p-1} \ldots i_1) = \sum_{i_0=0}^{1} (-1)^{n_{p-1}(i_0 + i_1)} x_{(i_{p-1} \ldots i_0)} \qquad (3.33)$$

To see how this is calculated the case of $N = 16$ can be studied. We take
ne first adjacent pair of data samples and look at the i_1 bit in x_i. Four values
f

$$(-1)^{n_{p-1}(i_0 + i_1)} \quad \text{for} \quad i_0 = 0 \text{ or } 1$$

$$n_{p-1} = 0 \text{ or } 1$$

re obtained.

These indicate whether to add or subtract the two adjacent values of x_i
nd x_{i+1} (where i and $i+1$ are decimal values). From these sums and
ifferences the two intermediate transformed values are obtained as

$$x_i + x_{i+1}$$

nd either

$$x_i - x_{i+1} \quad \text{or} \quad x_{i+1} - x_i$$

depending on the sign for $(-1)^{n_{p-1}(i_0+i_1)}$ worked out earlier. This process is continued for the remaining consecutive pairs of x_i coefficients.

Further stages of calculation proceed serially by using the results of the preceding stage as input for the next stage. This may be expressed in the general expression for the intermediate transformation as

$$A_r(n_{p-1}, n_{p-2}, \ldots n_{p-r}; i_{p-1}, i_{p-2} \ldots i_r)$$
$$= \sum_{i_{r-1}=0}^{1} (-1)^{n_{p-r}(i_r+i_{r-1})} A_{r-1}(n_{p-1} \ldots n_{p-r+1}, i_{p-1} \ldots i_{r-1}) \qquad (3.34)$$

with the values of A_1 to A_p retained in temporary storage during the course of the calculation.

Finally we need to normalise the result through division by N, viz.

$$X_{(n_{p-1}, n_{p-2}, \ldots n_0)} = \frac{1}{N} A_{p(n_{p-1}, n_{p-2}, \ldots n_0)} \qquad (3.35)$$

Thus it is seen that the complete transform may be obtained in $N \log_2 N$ addition and subtraction operations rather than N^2 operations demanded by the direct method of calculation for equation (3.5).

A summary of the mathematical operations required in the evaluation of several orthogonal transformations is given in Table 3.4. These compare the number and type of operations needed to calculate the transform for the discrete and fast forms of the algorithm. A fast transform algorithm does not exist for the Karhunen–Loeve transform.

Transform	Mathematical operations
Discrete Fourier	N^2 complex multiply-additions
Fast Fourier	$N \log_2 N$ complex multiply-additions
Discrete Walsh	N^2 addition/subtractions
Fast Walsh	$N \log_2 N$ additions/subtractions
Discrete Haar	N^2 additions/subtractions
Fast Haar	$2(N-1)$ additions/subtractions
Discrete Karhunen–Loeve	N^2 multiply-additions

TABLE 3.4. Summary of transformation operations.

IIIH2 Signal flow diagrams

Computation for the fast transform algorithms can be described conveniently by means of the signal flow diagram. This consists of a series of nodes, each representing a variable which is itself expressed as the sum of other variables originating from the left of the diagram, with the nodes connected together by means of straight lines. The weighting of these

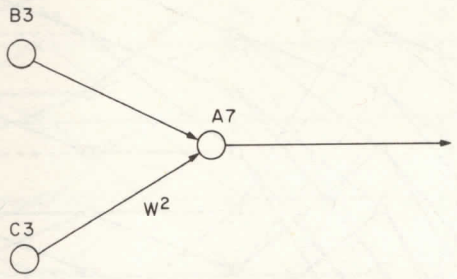

FIG. 3.2. A signal flow diagram.

itions is indicated by a number appearing at the side of an indicating w shown on these connecting lines. Thus from Fig. 3.2, the variable $A7$ erived from variables originating at nodes $B3$ and $C3$, with the latter ghted by W^2, so that we can write

$$A7 = B3 + W^2 \cdot C3 \tag{3.36}$$

ıre 3.3 shows a signal flow diagram for a sequency-ordered Walsh sform[8] which corresponds to the mathematical development given in tion IIIH1. The solid lines in the diagram indicate that the calculated ıe is to be carried forward to the addition at the next node with the same , and a dotted line indicates multiplication by -1 before addition takes ce.

3 "In-place" algorithms

e final calculation stage, illustrated in the signal flow diagram of Fig. 3.3, is resting since it shows a series of identical self-contained operations ied out on two values only. This is reproduced in Fig. 3.4 and has been rred to as a "butterfly" diagram. Here the output value D_1 is obtained n a linear combination of C_1 and C_2, whilst D_2 is obtained from the nbination C_1 and C_2 with the latter modified by a sign inversion.

These butterflies have the inherent advantage that in each of the transfor- tion steps once a pair of data locations have been read then they can be rwritten by the calculated pair of output values. This is the key to -place" algorithms in which memory storage for intermediate stage culations is not needed since the calculated values can be placed back into mory locations occupied previously by the initial data values. A signal v diagram for an "in-place" algorithm is shown in Fig. 3.5. This gives ults in bit-reversed order so that an auxiliary sorting sequence is required achieve binary reversal and hence linear sequency order. A study of t v diagram will show that calculation can proceed as a series of sin tterfly" calculations requiring no intermediate storage locations.

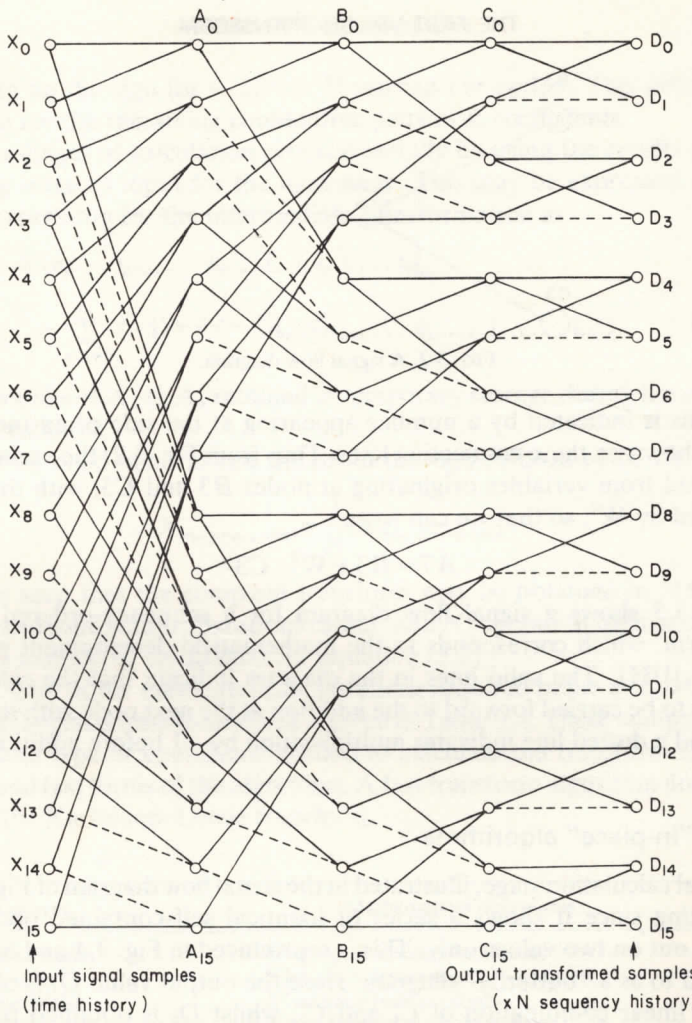

FIG. 3.3. A signal flow diagram for a sequency-ordered discrete Walsh transform (F.W.T.).

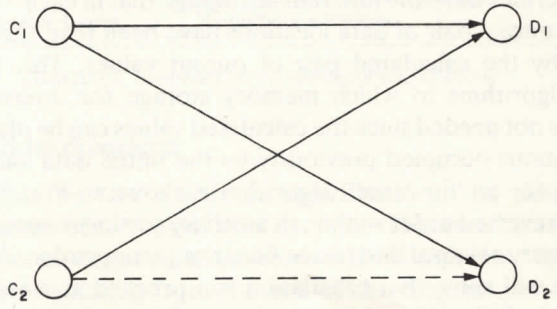

FIG. 3.4. A "Butterfly" of a signal flow diagram.

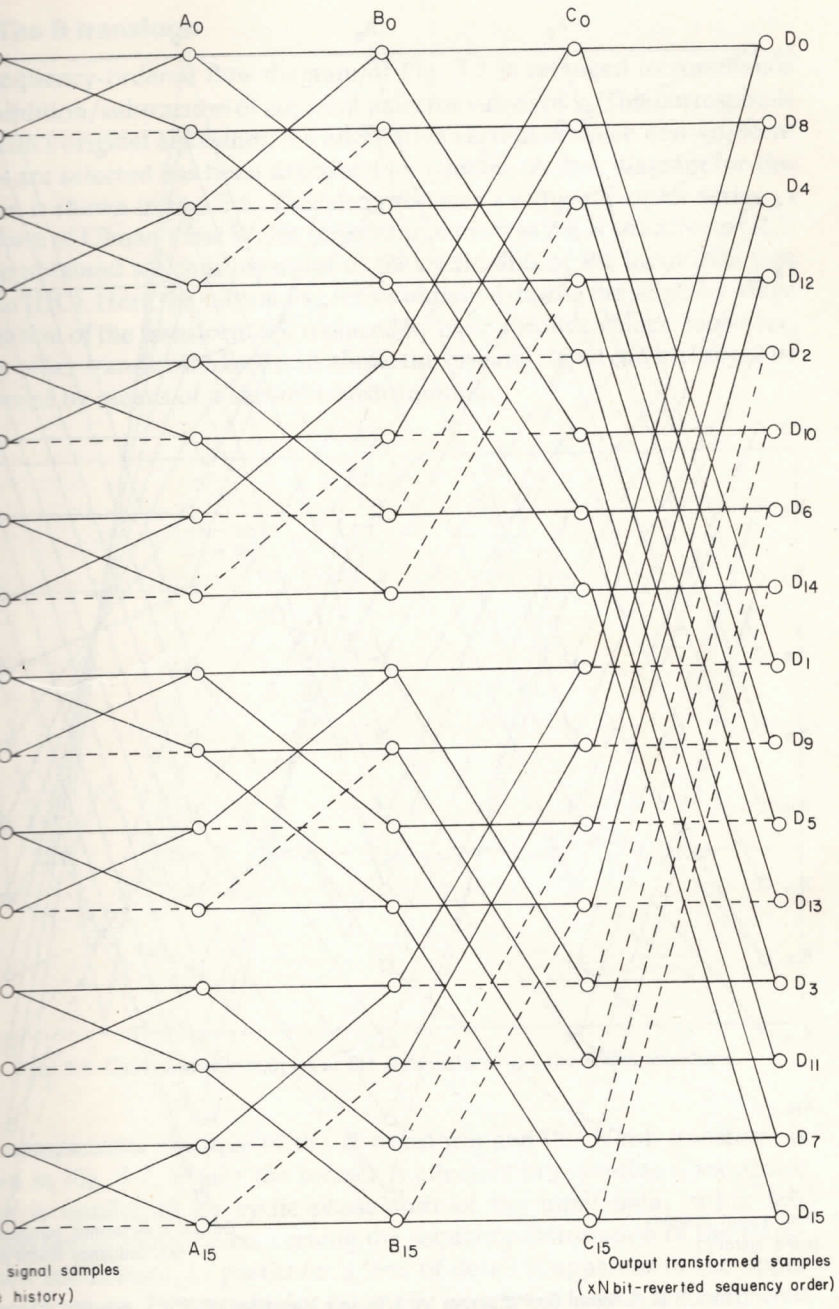

FIG. 3.5. A signal flow diagram for an "in-place" algorithm (F.F.W.T.).

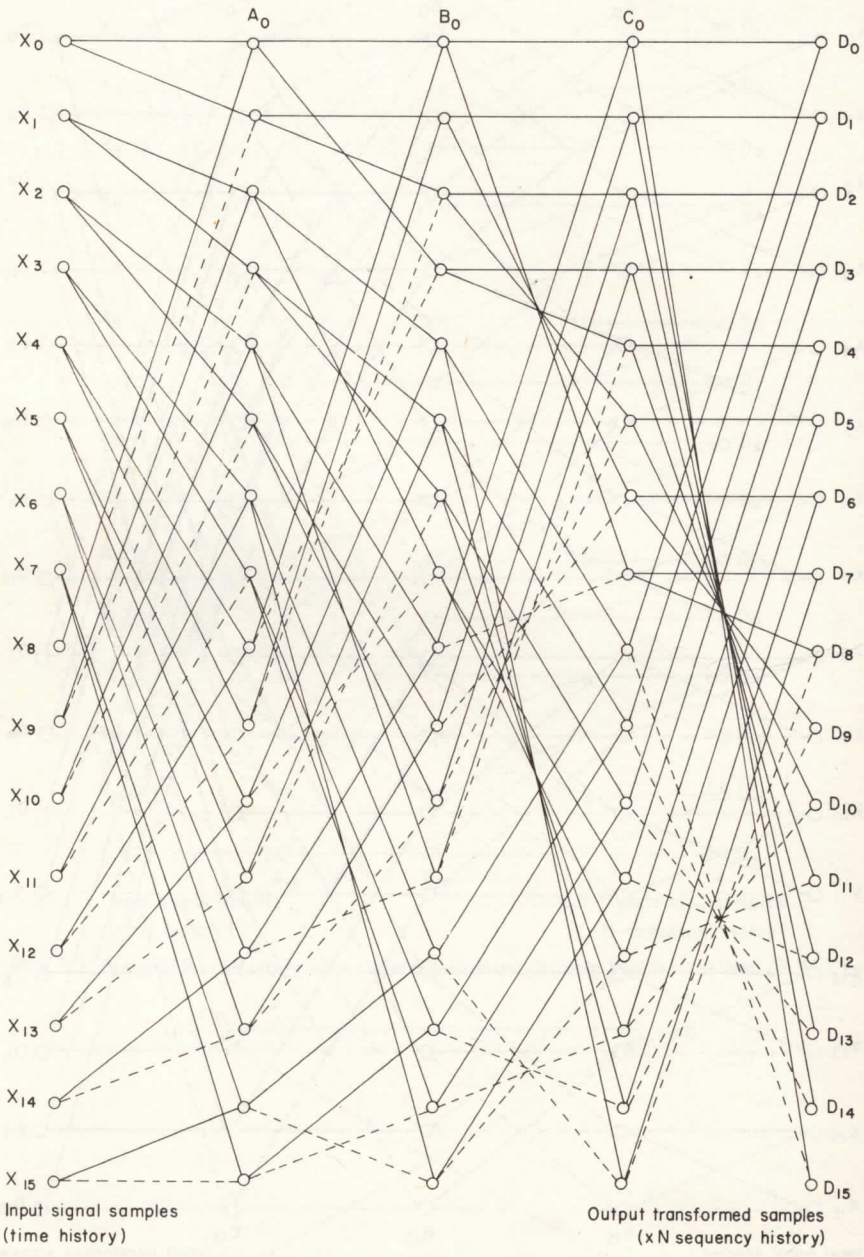

FIG. 3.6. A signal flow diagram for Ulman's algorithm (F.H.T.). $N = 2^4$.