PERMUTATION PROPERTIES OF SIGNAL PROCESSING TRANSFORMS

A THESIS
Submitted to the
UNIVERSITY OF ROORKEE
for the award of the degree
of
DOCTOR OF PHILOSOPHY
in
Electronics & Communication Engineering

By

SHIVA KANT VARMA

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247672 (INDIA)

September, 1979

C E R T I F I C A T E

Certified that the thesis entitled "PERMUTATION PROPERTIES OF SIGNAL PROCESSING TRANSFORMS" which is being submitted by Mr. Shiva Kant Varma in fulfilment of the requirements for the award of the Degree of DOCTOR OF PHILOSOPHY in Electronics and Communication Engineering of the University of Roorkee is a record of student's own work carried out by him under our supervision and guidance. The matter embodied in this thesis has not been submitted for the award of any other degree.

This is further to certify that he has worked for the prescribed period under Ph.D.regulations at this University for preparing this thesis.

P.S. Moharir
13-9-79
(Dr. P.S.Moharir)
Reader
Department of Earth Sciences
University of Roorkee,
Roorkee.

N C Jain
13-9-79
(Dr. N.C.Jain)
Professor and Head
Department of Electronics
and Communication Engg.
University of Roorkee,
Roorkee.

A B S T R A C T

The research work which lead to the preparation of this thesis was undertaken with the objective of defining some new transforms which could be used for signal (message, picture or data) processing and to study the permutation properties of the proposed signal processing transforms. The work contained in this thesis includes generation of higher order orthonormal transform kernels from lower order orthonormal transform kernels, proposing new two-dimensional transforms and studying their permutation properties, modification of some of the existing transforms for pattern recognition to transforms which could be used for transmission of message, picture and data, and defining a new class of systems which is invariant to some prescribed permutation.

It has been observed that the discrete finite system matrices for the proposed class of permutation invariant systen are not necessarily matrices with ranks equal to their orders. Conditions have been stipulated under which the resulting system matrices would have ranks equal to their orders. But this, however, needs further investigation,

Two-dimensional transforms could be frequently thought of as two one-dimensional transforms. By taking

various combinations of two one-dimensional orthonormal transform kernels one can define a class of two-dimensional transform kernels. The permutation properties of such transform can be deduced from the permutation properties of the component orthonormal transforms.

It is known that Kronecker product of two lower order orthonormal matrices results in an orthonormal matrix of higher order. The algebra for Kronecker product is well developed. But it does not commutative. A new matrix product, Chinese product, has been proposed. This product is defined only when the respective dimensions of the two-component matrices are coprimes. The matrix resulting from this product has all the properties of the matrix resulting from Kronecker product of the same component matrices. In addition this matrix product commutes. In fact the former is a rowwise and columnwise permuted version of the latter. Expressions have been derived for permutation matrices which can help in getting one from another. The notions of these matrix products and partitioning of matrices have been exploited to obtain higher order orthonormal transform kernels from lower order orthonormal transform kernels.

Many of the known transforms which find application in pattern recognition are nonlinear in nature. If these transforms could be inverted by some modification then the modified transforms could be useful for message, picture and data signals. It has been proposed that the additional knowledge about the labels at each functional block in the transmitter could lead to the recovery at receiver of the input signal samples at the transmitter. The class of thus modified transforms has been named as labelled symmetric function transform.

The thesis ends, as is customary, with references to some problems which could be taken up in future as an extension of this work.

ACKNOWLEDGEMENTS

It has indeed been a pleasure to work under the supervision and guidance of Dr. P.S.Moharir and Dr. N.C.Jain. This work could not have been completed but for their valuable advice, sincere guidance and friendly behaviour. The gratitude to them cannot be expressed in words but only felt.

It is a pleasure to remember the help, academic and otherwise, rendered by friends and well wishers.

Thanks are due to Shri U.K.Mishra, for cutting the stencils, Shri R.C.Vaish for making the diagrams and Shri Hari Ram for running the stencils.

This acknowledgement would be incomplete without a mention of gratitude to the members of the author's family, especially his wife and children, who had to bear his neglect of household during the period of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$a(i)$	Sequence of input samples
$a_t(i)$	a(i) after linear translation of samples
$a_r(i)$	a(i) written as two-dimensional array, read row by row, after rowwise permutation
$a_c(i)$	a(i) written as two-dimensional array, read row by row, after columnwise permutation
$a_{rc}(i)$	a(i) written as two-dimensional array , read row by row, after rowwise and columnwise permutation
$a(i,j)$	elements of matrix A
A,B,C	Matrices
$A(i_1,j_1)$	Submatrices of dimension mxn
$B(i_2,j_2)$	Submatrices of dimension nxp
$C(i,j)$	Submatrices of dimension mxp
C_c	Matrix obtained by Chinese product of component matrices
C_k	Matrix obtained by Kronecker product of component matrices
C_{ck}	Matrix obtained by Chinese-Kronecker product of component matrices
C_{kc}	Matrix obtained by Kronecker-Chinese product of component matrices
$A(I)$	Transform samples of $a(i)$
$A_t(I)$	Transform samples of $a_t(i)$
$A_r(I)$	Transform samples of $a_r(i)$
$A_c(I)$	Transform samples of $a_c(i)$
$A_{rc}(I)$	Transform samples of $a_{rc}(i)$

G	Transitive abelian permutation group of order N and degree N
P_i	Permutation/Permutation matrix
P'	Reciprocal of P with respect to a particular transform
P^{-1}	Inverse of P
$P(m)$	Selective-bit complementation permutation operator
$P(p,N)P(q,N)$	Modular permutation operator pair
$P(a\ b\ldots k)$	Bit-plane permutation operator
R^N	N-dimensional vector space of N-tuples
T	Transformation kernel
S	Finite discrete system matrix
$\otimes_c$	Chinese product
$\otimes_k$	Kronecker product
$\otimes_{ck}$	Chinese-Kronecker product
$\otimes_{kc}$	Kronecker-Chinese product
$((x))_N$	x modulo N
δ	Kronecker delta
$\equiv$	Congruent to
$\updownarrow$	equivalent
H	Hadamard transform operator
$h_{i_2}^{(I_1)}$	Element at the I_1^{th} row and i_2^{th} column of the matrix obtained by columnwise HT of a(i)
$H_{i_2}^{(I_1)}$	$h_{i_2}^{(I_1)}$ after multiplication with twiddle factor
$s(\ ,\ )$	elements of matrix S
$P_i^{(j)}$	Permutation matrix P_i of transitive abelian permutation group G_j

CHAPTER - 1

INTRODUCTION

In recent years there has been a growing interest regarding study of orthogonal transforms in the area of digital signal processing. This is primarily due to the impact of high speed digital computers and the rapid advances in digital technology and consequent development of special purpose digital processors. The applications of such transforms include image processing, speech processing, pattern recognition, spectroscopy etc. In signal processing and, in particular, speech and picture processing a wide class of transforms including Fourier, Walsh, Hadamard, Haar, slant and discrete cosine has come to be widely used in recent years.

Linear transforms can be used to obtain alternative descriptions of signals. These alternative descriptions can have many uses and most of the applications are based on the exploitation of the fact that linear transformation is a way of changing statistical and spectral characteristics of the signals. Thus it is advantageous to have a wide class of linear transforms from which a particular choice could be made for specific application.

Many transform kernels have a large amount of structural redundancy. In many cases the structural redundancy of the transform kernel is such that certain lower-order transform kernels are embedded in the higher-order transform kernels, provided certain elementary relations hold between these orders.

Non linear or even noninvertible transforms can be useful for applications such as classification and pattern recognition. These transforms exploit the fact that they could suppress certain aspects of the input signals which are irrelevant and focus attention on relevant parameters, relevance being defined in the context of a particular application. Efficient computational algorithms are known for a number of linear transforms. An efficient computational algorithm would be a desirable characteristic of nonlinear and noninvertible transforms also.

The work carried out for the preparation of this thesis relates to generation of higher order orthonormal kernels starting with lower order orthonormal kernels, definition of some new transform kernels, study of permutation properties of the kernels and inversion of some nonlinear transforms using additional labels. The thesis has been written in an unconventional way in the sense that instead of devoting a couple of chapters in the beginning for review of the existing literature the review and background necessary for understanding the text have been incorporated in each chapter. Thus each chapter in this thesis is mostly self contained in the sense that for its understanding one does not have to refer to other chapters.

The results in Chapter-2 are regarding possible exploitation of groups known as transitive abelian permutation group. Siddiqui [57] and Rao [52] have studied such groups to define a class of systems which is invariant under some prescribed permutation. The class of systems given by them has been named as permutation-invariant (P.I.) systems. It has been mentioned by these workers that such groups can be found out using results from finite group theory. But the search of literature on finite group theory which would enable one to write all possible transitive abelian permutation groups of order N and degree N which can be formed from a permutation group of order N! resulted in the negative. This problem is twofold : first to find out the exact number of such groups and second to write all the elements of all such groups. The second part of the problem in fact boils down to the task of finding any one primitive element of the group as all the elements in any group are the powers of the primitive element in the group. The work in this direction started with the listing of all such groups of lower degree so as to get some clue regarding any possible algorithm to solve the problem of writing the complete groups. All such groups have been listed for degree four and five. The complexity increases many times as one tries groups of higher degree. With these available results and the P.I.systems defined by earlier worker a new system which is invariant to some other permutation has been defined. Some inferences have been reported regarding conditions under which such systems would exist but they are inadequate in the sense that they are based on the results of examples worked out in respect of transitive abelian permutation groups of order and degree four and five only.

In Chapter 3 the known permutation properties of discrete Fourier transform and Hadamard transform have been exploited to study the permutation properties of a new transform namely Fourier twiddled H-DF transform. The idea of defining such a transform came from the results known for finding out the D F T of a sequence by writing it as a two-dimensional array read row by row. The D F T of the sequence is equal to the columnwise D F T followed by twiddling and then rowwise D F T . It was thought that if the columnwise D F T is replaced by columnwise H T and the twiddling factor modified then the resulting transform should exhibit certain permutation properties. When it was tried some nice permutation properties resulted. The permutation on the input sequence, written as two-dimensional array and read row by row, was effected by treating each row (column) as an element. The permutation properties for D F T and H T are well known for one-dimensional sequence and these were applied and transform taken. The permutation properties of the transform samples were found to depend upon permutation operator pairs, as expected. The technique developed in this chapter could be used to define a family of new transforms by choosing various pairs of orthonormal transforms in place of H T and D F T. The permutation properties of such transforms, if any, would depend upon the reciprocal permutation operator pairs for component transforms and the twiddling scheme. P, P' are said to be reciprocal permutation operator pairs with respect to some transform T if $T\,Pa(i) = P'Ta(i)$ where $a(i)$, $i = 0,1,2,\ldots,N-1$ is the input signal sample sequence of length N.

It is well known that the matrix resulting from the Kronecher product of two orthonormal matrices is again an orthonormal matrix. In Chapter 4 three more matrix products viz. Chinese product, Chinese-Kronecker product and Kronecker-Chinese product have been introduced. These are basically some modifications over the known Kronecker product of matrices. In all cases the dimension of the resulting matrix is the product of the corresponding dimensions of the component matrices. Unlike Kronecker product of matrices which is defined for all dimensions of the component matrices the matrix products introduced are defined only when the corresponding dimensions of the component matrices bear some relationships. The Chinese product is defined when the dimension of the rows (columns) of the component matrices are coprimes, i.e., they have no factor in common. The Chinese-Kronecker and Kronecker-Chinese products are defined only when the dimensions of the rows and columns respectively of the two component matrices are coprimes. It has been shown that the matrices obtained by taking these matrix products are orthonormal if the component matrices are orthonormal. In fact the matrices obtained in these cases are the row-wise or/and columnwise permuted versions of the matrix obtained by the Kronecker product of the same component matrices. This being the case it was hoped that one should be obtainable from the other by premultiplication and postmultiplication by suitable permutation matrices. Relationships have been deduced to obtain Chinese product matrix from Kronecker product matrix and vice-versa , and Chinese-Kronecker product matrix from Kronecker-Chinese product matrix and vice-versa provided both the products are defined. Thus by defining some more matrix products a method has been suggested for generating higher order orthonormal transform kernels starting with lower order orthonornal transform kernels.

The notions of Kronecker product and Chinese product of matrices have been made use of in Chapter 5 to define two new matrix products namely partitioned matrix Kronecker product and partitioned matrix Chienese product. In both of them the two component matrices are partitioned into submatrices so that the submatrices of one are conformable for ordinary matrix multiplication to submatrices of the other. The resulting submatrices are indexed as elements. Matrices are then obtained by taking Kronecker (Chinese) products of these component matrices wherein submatrices are treated as elements. The ordinary product of elements in Kronecker product of matricies is replaced by ordinary matrix product of submatrices. The matrices resulting from such multiplications are orthonormal if the component matrices are orthonormal.

Another method of getting higher order orthonormal matrices starting with lower order submatrices is exploiting the results available for construction of Hadamard arrays. In particular if the submatrices are real, symmetric and circulant then higher order orthonormal matrices of various orders can be obtained using Williamson design, Baumert-Hall design and Baumert-Hall-Welch design.

One important area of application of transforms is character recognition. Many transforms are known which have been successfully applied for such purposes. Most of these transforms are nonlinear and noninvertible. Since the location of the character in the pattern domain is of little interest, some of the transforms finding application in character recognition are translation invariant. Wagh [65] has given a class of such transforms, namely symmetric function transforms (S F T), and studied their usefulness in pattern recognition. The permutation properties of some of the transforms which are translation invariant in the sense that if the input is cyclically shifted the transform samples do not change have been studied in Chapter 6. Some new transforms have been proposed which could find application for such purposes. Some of them are superior in the sense that hardware needed is simpler and that many of the transform samples are zeroes in which case the average energy required for transmission would be less assuming that transmission of zeroes needs no energy. In this chapter an effort has been made to invert the nonlinear transforms. The additional information required for achieving this objective is labels at various functional blocks of the scheme which effects transformation. These labels would be different in different transformation schemes. Thus the known nonlinear transformations alongwith the knowledge of the labels at various functional blocks could be used for unique recovery of input samples from a knowledge of the transform samples and labels. The transformations basically developed for pattern recognition purposes can now be successfully employed for message or picture signals.

Though the results obtained in this thesis have been illustrated with examples of lower orders they can also be used with higher orders. As a matter of fact, as the order increases there is more flexibility in selecting the parameters like permutation operator, order of the submatrices etc.

CHAPTER - 2

SYSTEMS WITH PRESCRIBED PERMUTATION PROPERTIES

This chapter gives a critical review of the earlier work in the areas of one-dimensional and two-dimensional permutation invariant systems. Some modifications to the notions of permutation invariance have been suggested. With the modified notions the resulting system matrices may have lesser rank than order leading to degenerate cases. It may be interesting to find out the conditions under which degenerate cases do not arise.

2.1 PERMUTATION-INVARIANT SYSTEMS

A method for high speed computation of correlation and convolution of finite discrete signals has been given by Stockham [59]. Cyclic or circular convolution is the name assigned to such convolutions. Finite discrete linear systems defined by cyclic convolutional relationship between the input and output sequences are termed as cyclic convolution systems [22] and they have characteristics similar to linear time-invariant systems. The cyclic convolution systems have been widely used indirectly in many areas involving fast Fourier transform (F F T) implementations of ordinary convolution and correlation of finite discrete signals [1,4,5,21,23,51,59]. The notion of convolution was extended when Walsh functions [19,45,68] were applied for signal processing. Gibbs introduced the concept of logical convolution, which is presently known as dyadic convolution [16,17,22,26,47], of finite discrete signals of length equal to an integer power of two. The linear systems characterized by dyadic convolutions, termed as dyadic-invariant systems by Pichler[46], have features similar to linear time-invariant systems.

Rosenbloom [54] applied theory of groups [13,14, 20,25,27,28,29,31,56,69] to dyadic systems. He has pointed out that character vectors and regular representations of finite abelian groups play respectively the roles that complex exponential functions and time translations have in the theory of linear time-invariant systems. His discussion, however, was limited to dyadic systems corresponding to dyadic groups. Gethöffer [22] observed that cyclic, dyadic as well as ordinary discrete convolutions have similar structures. He investigated mutual mappings amongst these systems with particular emphasis on cyclic and dyadic systems.

Definitions of some of the terms from group theory which find frequent references in the chapter are:

ORDER - If G is a finite group then the number of elements in G is known as the order of the group.

DEGREE - If $\sum$ represents a finite set of objects and then objects of $\sum$ are denoted by the integers 1,2,...,N then a map of $\sum$ onto itself is called a permutation of degree N.

PERMUTATION GROUP - There are N! arrangements which map a set of objects denoted by the integers 1,2,..,N onto itself. The set containing all such arrangements is known as a permutation group. Thus a permutation group of degree N is of order N!

TRANSITIVE GROUP - A group of permutations is said to be transitive if, given any pair of letters a,b (which need not be distinct), there exists at least one permutation in the group which transforms a into b. Othwrwise the group is intransitive.

ABELIAN GROUP - A group which has the additional property that for every two of its elements $a * b = b * a$ is called an Abelian (or commutative) group.

TRANSITIVE ABELIAN PERMUTATION GROUP OF ORDER N AND DEGREE N -

This is a subgroup of order N formed out of a permutation group of order N! and degree N such that for any pair of elements P_i , P_j ; i , $j = 0,1,2,..., N-1$

i) there exists only one permutation in the subgroup which transforms P_i into P_j.

ii) $P_i * P_j = P_j * P_i$

CALEY TABLE - For a finite set S, a binary operation * on the set can be defined by means of a table. Caley table or multiplication table is one such table wherein (i^{th} entry on the left)*(j^{th} entry on the top) = (entry in the i^{th} row and j^{th} column of the table)

It is obvious from the definition of transitive abelian permutation group of order N and degree N that a variety of such groups can be constructed for any given N. There would always be a cyclic group for all values of N and a dyadic group if N is equal to an integer power of two.

By a finite discrete system S is meant a mapping from R^N to R^N and written

$$y = S x$$

where $x \varepsilon R^N$ and $y \varepsilon R^N$ are the system input and system output respectively. R^N is the N-dimensional vector space of N-tuples. If the system is linear it has the following matrix representation with respect to the standard basis E:

$$\begin{bmatrix} s(0,0) & s(0,1) & \cdots & s(0,N-1) \\ s(1,0) & s(1,1) & \cdots & s(1,N-1) \\ \vdots & \vdots & \cdots & \vdots \\ s(N-1,0) & s(N-1,1) & \cdots & s(N-1,N-1) \end{bmatrix} \quad \ldots(2.1)$$

The input and output signals x and y have N x 1 matrix representations $[x(0)\ x(1)\ \ldots\ x(N-1)]^T$ and $[y(0)\ y(1)\ \ldots\ u(N-1)]^T$ respectively.

If G be a transitive abelian permutation group of order N and degree N and P_i some element in G, $P_i \in G$, then a finite discrete system S is said to be permutation - invariant (P-I) relative to G if, for any signal x in R^N,

$$P_i\ (S\ x) = S\ (P_i\ x) \quad P_i \in G \quad \ldots(2.2)$$

The set of all such systems, relative to a given G, is termed as a class of P.I. system of dimension N, where N is the length of the signal. The number of various classes of P.I. systems of a given dimension is equal to the number of G's defined.

Siddiqui [57] has obtained a general formula which generates pertinent permuted signals belonging to various classes of P-I systems. This has been accomplished by suitably ordering the group elements in accordance with the notion of representing numbers with respect to mixed radices [60]. It has been shown that systems in each class are fully characterized by their unit sample response. P-I systems are represented by matrices known as P-I matrices. Some of the important properties of P-I matrices are:

1) P-I matrices representing any particular class of P-I systems constitute a vector space of dimension equal to the dimension of the signal space on which the P-I systems of the class operate , the set of permutation matrices representing the permutations of the transitive abelian permutation group with respect to which the class of P-I systems is defined serves as a basis of this vector space.

2) The eigen vectors of P-I matrices, and hence the eigen signals of the P-I systems, are the discrete versions of Levy's generalized Walsh functions [30]. The corresponding modal matrices belong to the family of generalized Hadamard matrices. The eigen values of P-I matrices are the components of generalized Walsh-Hadamard transform of their generating vector (zeroth column).

3) P-I matrices are closed under inversion and multiplication which is commutative.

Siddiqui [57] has applied his results on P-I systems for spectral shaping (filtering) of finite discrete signals to develope a theory of P-I filters wherein the role of the complex exponential signals in classical filter theory is taken over by the eigen signals of the particular class of P-I systems. He has, in particular, discussed the filtering of finite discrete signals with the help of dyadic P-I systems and cyclic P-I systems.

In case of many seperable systems, the extension of concepts to multi-dimensional situation is trivial and for dyadic systems, many one-dimensional results are obtained by taking recourse to the fact that one-dimensional dyadic system is equivalent to multi-dimensional dyadic systems through Kronecker products [36]. Of the various classes of 1-D P-I systems, only the cyclic and dyadic classes have so far been found to have a significant role in the processing of finite discrete data. It has been shown by Rao [52] that many of those I-D P-I systems which belong neither to the cyclic nor to the dyadic class are, in fact, the 1-D equivalents of 2-D or multi-dimensional cyclic or dyadic P-I systems. Such 1-D P-I systems are thus of indirect practical use in the processing of 2-D and multi-dimensional finite discrete data. Rao [52] has shown that when the data to be processed are finite, exact 1-D realization of 2-D filters can be obtained using the P-I system approach. The results reported by Rao [52] are essentially an extension of Siddiqui's [57] work. He has reported certain generalizations of P-I linear systems. The generalizations reported by him pertain to the following three new categories of P-I systems :

1) 2-D P-I systems which have finite 2-D arrays of reals as their input signals.
2) P-I systems on finite fields, i.e., those 1-D P-I systems whose finite length input sequences have their entries drawn from finite fields.
3) P-I systems on rings, i.e., those 1-D P-I systems whose finite length input sequences have their entries from rings of residue class integers.

2.2 SOME NEW RESULTS ON P-I SYSTEMS

All possible arrangements of $(x_0\ x_1\ x_2 \ldots x_{N-1})$ form a group of order $N!$ and degree N under permutation multiplication. Many transitive abelian permutation groups of order N and degree N can be formed out of this group. Any permutation can be equivalently represented as a premultiplication by a permutation matrix. Unless otherwise stated the same symbol P would be used for permutation as well as permutation matrix. When $N = 4$ there are four possible transitive abelian permutation groups of order four and degree four, say G_0, G_1, G_2

and G_3. Various elements of each such group alongwith their Caley table are listed below:

1. Group G_0 Permutation P_0 $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_0 & x_1 & x_2 & x_3 \end{matrix}$

Permutation P_1 $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_3 & x_0 & x_1 & x_2 \end{matrix}$

Permutation P_2 $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_2 & x_3 & x_0 & x_1 \end{matrix}$

Permutation P_3 $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_1 & x_2 & x_3 & x_0 \end{matrix}$

Caley Table

*	P_0	P_1	P_2	P_3
P_0	P_0	P_1	P_2	P_3
P_1	P_1	P_2	P_3	P_0
P_2	P_2	P_3	P_0	P_1
P_3	P_3	P_0	P_1	P_2

where * is permutation multiplication and P_0 is the identity element.

2. Group G_1 Permutation P_0 $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_0 & x_1 & x_2 & x_3 \end{matrix}$

Permutation P_1 $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_1 & x_0 & x_3 & x_2 \end{matrix}$

Permutation P_2

x_o	x_1	x_2	x_3
x_3	x_2	x_o	x_1

Permutation P_3

x_o	x_1	x_2	x_3
x_2	x_3	x_1	x_o

Caley Table

*	P_o	P_1	P_2	P_3
P_o	P_o	P_1	P_2	P_3
P_1	P_1	P_o	P_3	P_2
P_2	P_2	P_3	P_1	P_o
P_3	P_3	P_2	P_o	P_1

3. Group G_2

Permutation P_o

x_o	x_1	x_2	x_3
x_o	x_1	x_2	x_3

Permutation P_1

x_o	x_1	x_2	x_3
x_2	x_o	x_3	x_1

Permutation P_2

x_o	x_1	x_2	x_3
x_1	x_3	x_o	x_2

Permutation P_3

x_o	x_1	x_2	x_3
x_3	x_2	x_1	x_o

Caley Table

*	P_o	P_1	P_2	P_3
P_o	P_o	P_1	P_2	P_3
P_1	P_1	P_3	P_o	P_2
P_2	P_2	P_o	P_3	P_1
P_3	P_3	P_2	P_1	P_1

4. Group G_3

Permutation	P_o	x_o	x_1	x_2	x_3
		x_o	x_1	x_2	x_3
Permutation	P_1	x_o	x_1	x_2	x_3
		x_1	x_o	x_3	x_2
Permutation	P_2	x_o	x_1	x_2	x_3
		x_2	x_3	x_o	x_1
Permutation	P_3	x_o	x_1	x_2	x_3
		x_3	x_2	x_1	x_o

Caley Table

*	P_o	P_1	P_2	P_3
P_o	P_o	P_1	P_2	P_3
P_1	P_1	P_o	P_3	P_2
P_2	P_2	P_3	P_o	P_1
P_3	P_3	P_2	P_1	P_o

Group G_o is cyclic and group G_3 is dyadic-all its elements being their own inverses. Such a group exists only when the order of the group N is an integer power of two.

If S be a finite discrete linear system represented by Eq. (2.1) then this is said to be permutation-invariant with respect to some group G if matrix S commutes with the permutation matrices corresponding to all the elements of the group G, i.e.

$$SP_o = P_oS \qquad P_i \in G \quad i=0,1,2,3$$
$$SP_1 = P_1S$$
$$SP_2 = P_2S$$
$$SP_3 = P_3S$$

Since P_o is identity matrix so effectively only three conditions are to be satisfied. The system matrices representing P.I. systems with respect to the groups defined earlier are listed below:

1. Group G_o

$$S = \begin{bmatrix} s_o & s_3 & s_2 & s_1 \\ s_1 & s_o & s_3 & s_2 \\ s_2 & s_1 & s_o & s_3 \\ s_3 & s_2 & s_1 & s_o \end{bmatrix}$$

This has cyclic structure as would be expected.

2. Group G_1

$$S = \begin{bmatrix} s_o & s_1 & s_3 & s_2 \\ s_1 & s_o & s_o & s_3 \\ s_2 & s_3 & s_o & s_1 \\ s_3 & s_2 & s_1 & s_o \end{bmatrix}$$

3. Group G_2

$$S = \begin{bmatrix} s_o & s_2 & s_1 & s_3 \\ s_1 & s_o & s_3 & s_2 \\ s_2 & s_3 & s_o & s_1 \\ s_3 & s_1 & s_2 & s_o \end{bmatrix}$$

4. Group G_3

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_1 & s_0 & s_3 & s_2 \\ s_2 & s_3 & s_0 & s_1 \\ s_3 & s_2 & s_1 & s_0 \end{bmatrix}$$

This has dyadic structure as would be expected. When $1 \leq N \leq 3$ there is only one P-I system and that has cyclic structure. In case of $N = 4$ it has been shown that there are four possible P-I systems : one cyclic, one dyadic and the rest having no special names. The transitive abelian permutation groups and the resulting P-I system matrices are listed below for $N = 5$.

1. Group G_0

Permutation	P_0	x_0	x_1	x_2	x_3	x_4
		x_0	x_1	x_2	x_3	x_4
Permutation	P_1	x_0	x_1	x_2	x_3	x_4
		x_4	x_0	x_1	x_2	x_3
Permutation	P_2	x_0	x_1	x_2	x_3	x_4
		x_3	x_4	x_0	x_1	x_2
Permutation	P_3	x_0	x_1	x_2	x_3	x_4
		x_2	x_3	x_4	x_0	x_1
Permutation	P_4	x_0	x_1	x_2	x_3	x_4
		x_1	x_2	x_3	x_4	x_0

Caley Table

*	P_0	P_1	P_2	P_3	P_4
P_0	P_0	P_1	P_2	P_3	P_4
P_1	P_1	P_2	P_3	P_4	P_0
P_2	P_2	P_3	P_4	P_0	P_1
P_3	P_3	P_4	P_0	P_1	P_2
P_4	P_4	P_0	P_1	P_2	P_3

$$S = \begin{bmatrix} s_0 & s_4 & s_3 & s_2 & s_1 \\ s_1 & s_0 & s_4 & s_3 & s_2 \\ s_2 & s_1 & s_0 & s_4 & s_3 \\ s_3 & s_2 & s_1 & s_0 & s_4 \\ s_4 & s_3 & s_2 & s_1 & s_0 \end{bmatrix}$$

This has cyclic structure as would be expected.

2. Group G_1

Permutation P_0 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_0 & x_1 & x_2 & x_3 & x_4 \end{matrix}$

Permutation P_1 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_2 & x_0 & x_3 & x_4 & x_1 \end{matrix}$

Permutation P_2 $\begin{matrix} x_0 & x_1 & z_2 & x_3 & x_4 \\ x_1 & x_4 & x_0 & x_2 & x_3 \end{matrix}$

Permutation P_3 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_4 & x_3 & x_1 & x_0 & x_2 \end{matrix}$

Permutation P_4 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_3 & x_2 & x_4 & x_1 & x_0 \end{matrix}$

Caley Table

*	P_o	P_1	P_2	P_3	P_4
P_o	P_o	P_1	P_2	P_3	P_4
P_1	P_1	P_4	P_o	P_2	P_3
P_2	P_2	P_o	P_3	P_4	P_1
P_3	P_3	P_2	P_4	P_1	P_o
P_4	P_4	P_3	P_1	P_o	P_2

$$S = \begin{bmatrix} s_o & s_2 & s_1 & s_4 & s_3 \\ s_1 & s_o & s_4 & s_3 & s_2 \\ s_2 & s_3 & s_o & s_1 & s_4 \\ s_3 & s_4 & s_2 & s_o & s_1 \\ s_4 & s_1 & s_3 & s_2 & s_o \end{bmatrix}$$

3. Group G_2 Permutation P_o $\quad \begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_o & x_1 & x_2 & x_3 & x_4 \end{matrix}$

Permutation P_1 $\quad \begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_2 & x_o & x_4 & x_1 & x_3 \end{matrix}$

Permutation P_2 $\quad \begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_1 & x_3 & x_o & x_4 & x_2 \end{matrix}$

Permutation P_3 $\quad \begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_4 & x_2 & x_3 & x_o & x_1 \end{matrix}$

Permutation P_4 $\quad \begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_3 & x_4 & x_1 & x_2 & x_o \end{matrix}$

Caley Table

*	P_0	P_1	P_2	P_3	P_4
P_0	P_0	P_1	P_2	P_3	P_4
P_1	P_1	P_3	P_0	P_4	P_2
P_2	P_2	P_0	P_4	P_1	P_3
P_3	P_3	P_4	P_1	P_2	P_0
P_4	P_4	P_2	P_3	P_0	P_1

$$S = \begin{bmatrix} s_0 & s_2 & s_1 & s_4 & s_3 \\ s_1 & s_0 & s_3 & s_2 & s_4 \\ s_2 & s_4 & s_0 & s_3 & s_1 \\ s_3 & s_1 & s_4 & s_0 & s_2 \\ s_4 & s_3 & s_2 & s_1 & s_0 \end{bmatrix}$$

4. Group G_3 Permutation P_0 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_0 & x_1 & x_2 & x_3 & x_4 \end{matrix}$

Permutation P_1 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_3 & x_0 & x_1 & x_4 & x_2 \end{matrix}$

Permutation P_2 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_4 & x_0 & x_3 \end{matrix}$

Permutation P_3 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_4 & x_3 & x_0 & x_2 & x_1 \end{matrix}$

Permutation P_4 $\begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_2 & x_4 & x_3 & x_1 & x_0 \end{matrix}$

Caley Table

*	P_o	P_1	P_2	P_3	P_4
P_o	P_o	P_1	P_2	P_3	P_4
P_1	P_1	P_3	P_o	P_4	P_2
P_2	P_2	P_o	P_4	P_1	P_3
P_3	P_3	P_4	P_1	P_2	P_o
P_4	P_4	P_2	P_3	P_o	P_1

$$S = \begin{bmatrix} s_o & s_3 & s_4 & s_1 & s_2 \\ s_1 & s_o & s_3 & s_2 & s_4 \\ s_2 & s_1 & s_o & s_4 & s_3 \\ s_3 & s_4 & s_2 & s_o & s_1 \\ x_4 & s_2 & s_1 & s_3 & s_o \end{bmatrix}$$

5. Group G_4 Permutation P_o $\begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_o & x_1 & x_2 & x_3 & x_4 \end{matrix}$

Permutation P_1 $\begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_4 & x_o & x_3 & x_1 & x_2 \end{matrix}$

Permutation P_2 $\begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_1 & x_3 & x_4 & x_2 & x_o \end{matrix}$

Permutation P_3 $\begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_3 & x_2 & x_o & x_4 & x_1 \end{matrix}$

Permutation P_4 $\begin{matrix} x_o & x_1 & x_2 & x_3 & x_4 \\ x_2 & x_4 & x_1 & x_o & x_3 \end{matrix}$

Caley Table

*	P_0	P_1	P_2	P_3	P_4
P_0	P_0	P_1	P_2	P_3	P_4
P_1	P_1	P_4	P_0	P_2	P_3
P_2	P_2	P_0	P_3	P_4	P_1
P_3	P_3	P_2	P_4	P_1	P_0
P_4	P_4	P_3	P_1	P_0	P_2

$$S = \begin{bmatrix} s_0 & s_4 & s_3 & s_2 & s_1 \\ s_1 & s_0 & s_2 & s_4 & s_3 \\ s_2 & s_3 & s_0 & s_1 & s_4 \\ s_3 & s_1 & s_4 & s_0 & s_2 \\ s_4 & s_2 & s_1 & s_3 & s_0 \end{bmatrix}$$

Obtaining results on permutation properties of linear systems does not per se need listing of permutation groups. The results should be obtainable from the properties of the group. But in actual practice it is common that the nature of the permutation properties is studied from results obtained by actual listing. The pattern of results so obtained gives an idea about the theorem and which subsequently is derived from analytical considerations. In this case because of enormous amount of computation needed it was not possible to have results which could suggest substantially consistant permutation properties. It was at this stage that detailed investigations had to be given up.

When an attempt was made to write all transitive abelian permutation groups for N = 6 it turned out that this number would be fairly large and almost beyond manual computation. The existing literature on finite group theory was reviewed to find some fast algorithms for listing such groups so that efficient computer programme could be prepared. A computer programme based on the definition of the transitive abedian permutation group would be highly complex in nature and would have enormous time and memory requirements even for relatively small values of N. The available literature on finite group theory has Sylow theorems which could be of some help. But this in our case would give the number of subgroups of order N and degree N provided N! = Nm where N is a prime and does not divide m. The number of subgroups of order N and degree N would be congruent to 1 (modulo N) and divides N!. It is important to note that all such subgroups would not be transitive abelian permutation groups of order N and degree N. Identification of such groups from possible number of subgroups is not easy. Once this identification has somehow been achieved the actual problem is to list all the groups. Though all the elements of the transitive abelian permutation group can be written even if one of its primitive elements were known, no theorem was available which could help isolate such elements and consequently the groups. Thus the Sylow theorems have two limitations from our standpoint.

1) They have application only when N! can be written in a particular form.

2) The maximum number of subgroups possible is very large and hence direct enumeration is not practicable.

It might be of interest in future to develope fast algorithms for the listing of transitive abelian permutation groups for any given N.

2.3 RECIPROCAL-PERMUTATION SYSTEMS

The same permutation matrix P_i appears on either sides of equation (2.2) which defines a P-I system relative to some transitive abelian permutation group of order N and degree N. A natural question is what would happen if different permutation matrices, but both belonging to the same group G, appear on two sides in the equation. This scheme was worked out to see if there is any regular pattern in the resulting finite discrete system S under prescribed permutation properties relative to some group G. The P-I system S resulting from the definition given in equation (2.2) would satisfy the following conditions when N=4

$$\begin{aligned} SP_1 &= P_1S \\ SP_2 &= P_2S \\ SP_3 &= P_3S \end{aligned} \qquad P_1, P_2, P_3 \in G \qquad \ldots (2.3)$$

Let us define a finite discrete system S under the following permutation properties :

$$SP_1 = P_uS \qquad u,v,w = 1,2,3 \qquad \ldots (2.4)$$
$$SP_2 = P_vS$$
$$SP_3 = P_wS$$

The set of such system matrices S would be termed as a class of reciprocal-permutation system (R-P system) and P_1, P_u; P_2, P_v ; and P_3, P_w are called reciprocal-permutation pairs with respect to S. When u = 1, v = 2 and w = 3 the R-P system becomes P-I system. The finite discrete system matrices S for R-P system were obtained for groups G_o, G_1 and G_3 when N = 4. The conditions under which realizable S were obtained are listed below along-with the corresponding R-P system matrices S. The system matrices S corresponding to P-I system have not been given here as they have already been listed in the earlier section.

1. $SP_1 = P_3S \qquad P_1, P_2, P_3 \in G_o$
   $SP_2 = P_2S$
   $SP_3 = P_1S$

$$S = \begin{bmatrix} s_o & s_1 & s_2 & s_3 \\ s_1 & s_2 & s_3 & s_o \\ s_2 & s_3 & s_o & s_1 \\ s_3 & s_o & s_1 & s_2 \end{bmatrix}$$

S is a back circulant matrix

2. $SP_1 = P_1S \qquad P_1, P_2, P_3 \in G_1$
   $SP_2 = P_3S$
   $SP_3 = P_2S$

$$S = \begin{bmatrix} s_o & s_1 & s_2 & s_3 \\ s_1 & s_o & s_3 & s_2 \\ s & s_3 & s_1 & s_o \\ s_3 & s_2 & s_o & s_1 \end{bmatrix}$$

3(a) $SP_1 = P_1S$ $\quad P_1, P_2, P_3 \in G_3$

$SP_2 = P_3S$

$SP_3 = P_2S$

$$S = \begin{bmatrix} s_0 & s_1 & s_3 & s_2 \\ s_1 & s_0 & s_2 & s_3 \\ s_2 & s_3 & s_1 & s_0 \\ s_3 & s_2 & s_0 & s_1 \end{bmatrix}$$

This has dyadic structure rowwise in the sense that all subsequent rows can be derived from the leading row by dyadic shifts.

(b) $SP_1 = P_2S$ $\quad P_1, P_2, P_3 \in G_3$

$SP_2 = P_1S$

$SP_3 = P_3S$

$$S = \begin{bmatrix} s_0 & s_2 & s_1 & s_3 \\ s_1 & s_3 & s_0 & s_2 \\ s_2 & s_0 & s_3 & s_1 \\ s_3 & s_1 & s_2 & s_0 \end{bmatrix}$$

This has dyadic structure.

(c) $SP_1 = P_2S$ $\quad P_1, P_2, P_3 \in G_3$

$SP_2 = P_3S$

$SP_3 = P_1S$

$$S = \begin{bmatrix} s_0 & s_2 & s_3 & s_1 \\ s_1 & s_3 & s_2 & s_0 \\ s_2 & s_0 & s_1 & s_3 \\ s_3 & s_1 & s_0 & s_2 \end{bmatrix}$$

This has dyadic structure.

(d) $SP_1 = P_3S$ $\quad P_1, P_2, P_3 \in G_3$

$SP_2 = P_1S$

$SP_3 = P_2S$

$$S = \begin{bmatrix} s_0 & s_3 & s_1 & s_2 \\ s_1 & s_2 & s_0 & s_3 \\ s_2 & s_1 & s_3 & s_0 \\ s_3 & s_0 & s_2 & s_1 \end{bmatrix}$$

This has dyadic structure.

(e) $SP_1 = P_3S \qquad P_1, P_2, P_3 \in G_3$

$SP_2 = P_2S$

$SP_3 = P_1S$

$$S = \begin{bmatrix} s_0 & s_3 & s_2 & s_1 \\ s_1 & s_2 & s_3 & s_0 \\ s_2 & s_1 & s_0 & s_3 \\ s_3 & s_0 & s_1 & s_2 \end{bmatrix}$$

This has dyadic structure.

The following inferences can be drawn from the above results:

1) The finite discrete system matrix S would have rank equal to its order if the suffixes of P's on the left hand side and the right hand side of at least one of the conditions is the same and that this permutation forms a subgroup of order two with the identity element P_o. The product of P's on the L H S and R H S of the remaining conditions must be equal to either the identity element P_o or a permutation which is its own inverse. Thus in case of group G_2 the finite discrete system matrix S would be of rank four when

$$SP_1 = P_2S$$
$$SP_2 = P_1S$$
$$SP_3 = P_3S$$

This can be easily verified.

2) The set of system matrices which satisfy equation (2.4) constitute a class of systems with reciprocal-permutation property. This may be termed as 1-D R-P system. The resulting realizable system matrices are matrices which are columnwise permuted version of each other with zeroth column remaining fixed. In such permuted matrices the 1st, 2nd and 3rd columns are the u,v and w columns of the system matrix corresponding to the P-I system.

The inferences enumerated above are based on studies on transitive abelian permutation groups of order four and degree four only. It may be of interest to investigate cases corresponding to groups of higher order and enumerate more comprehensive results.

The class of 1-D R-P system can be enlarged by generalizing the definition of 1-D R-P system as given by Eq. (2.4) as

$$\begin{aligned} SP_1^{(i)} &= P_u^{(j)}S \qquad u,v,w = 1,2,3 \\ SP_2^{(i)} &= P_v^{(j)}S \qquad i,j = 0,1,2,3 \\ SP_3^{(i)} &= P_w^{(j)}S \end{aligned} \qquad \ldots (2.5)$$

where $P_1^{(k)}$ is the permutation matrix P_1 of group G_k.

An attempt was made to find out whether finite discrete system matrices S, having rank equal to its order, would exist under this prescribed permutation property. If so, under what conditions? The scheme was worked out for several pairs of groups and the conditions under which realizable S were obtained are listed below alongwith the corresponding generalized R-P system matrices S.

1(a) $SP_1^{(1)} = P_3^{(2)}S$

$SP_2^{(1)} = P_1^{(2)}S$

$SP_3^{(1)} = P_2^{(2)}S$

$$S = \begin{bmatrix} s_0 & s_3 & s_2 & s_1 \\ s_1 & s_2 & s_0 & s_3 \\ s_2 & s_1 & s_3 & s_0 \\ s_3 & s_0 & s_1 & s_2 \end{bmatrix}$$

(b) $SP_1^{(1)} = P_3^{(2)}S$

$SP_2^{(1)} = P_2^{(2)}S$

$SP_3^{(1)} = P_1^{(2)}S$

$$S = \begin{bmatrix} s_0 & s_3 & s_1 & s_2 \\ s_1 & s_2 & s_3 & s_0 \\ s_2 & s_1 & s_0 & s_3 \\ s_3 & s_0 & s_2 & s_1 \end{bmatrix}$$

(c) $SP_1^{(2)} = P_2^{(1)}S$

$SP_2^{(2)} = P_3^{(1)}S$

$SP_3^{(2)} = P_1^{(1)}S$

$$S = \begin{bmatrix} s_0 & s_3 & s_2 & s_1 \\ s_1 & s_2 & s_3 & s_0 \\ s_2 & s_0 & s_1 & s_3 \\ s_3 & s_1 & s_0 & s_2 \end{bmatrix}$$

(d) $SP_1^{(2)} = P_3^{(1)}S$

$SP_2^{(2)} = P_2^{(1)}S$

$SP_3^{(2)} = P_1^{(1)}S$

$$S = \begin{bmatrix} s_0 & s_2 & s_3 & s_1 \\ s_1 & s_3 & s_2 & s_0 \\ s_2 & s_1 & s_0 & s_3 \\ s_3 & s_0 & s_1 & s_2 \end{bmatrix}$$

It may be pointed out that $P_1^{(1)}$ and $P_3^{(2)}$ are their own inverses, and the groups G_1, G_2 are neither dyadic nor cyclic. Further, $P_2^{(1)}*P_2^{(1)} = P_3^{(1)}*P_3^{(1)}=P_1^{(1)}$ and $P_1^{(2)}*P_1^{(2)} = P_2^{(2)} * P_2^{(2)} = P_3^{(2)}$. Also conditions (c) and (d) are obtained by commuting S and P in(a) and (b) respectively.

2(a)
$$SP_1^{(0)} = P_2^{(1)}S \qquad SP_2^{(0)} = P_1^{(1)}S \qquad SP_3^{(0)} = P_3^{(1)}S$$
$$S = \begin{bmatrix} s_0 & s_3 & s_1 & s_2 \\ s_1 & s_2 & s_0 & s_3 \\ s_2 & s_0 & s_3 & s_1 \\ s_3 & s_1 & s_2 & s_0 \end{bmatrix}$$

(b)
$$SP_1^{(0)} = P_3^{(1)}S \qquad SP_2^{(0)} = P_1^{(1)}S \qquad SP_3^{(0)} = P_2^{(1)}S$$
$$S = \begin{bmatrix} s_0 & s_2 & s_1 & s_3 \\ s_1 & s_3 & s_0 & s_2 \\ s_2 & s_1 & s_3 & s_0 \\ s_3 & s_0 & s_2 & s_1 \end{bmatrix}$$

(c)
$$SP_1^{(1)} = P_2^{(0)}S \qquad SP_2^{(1)} = P_1^{(0)}S \qquad SP_3^{(1)} = P_3^{(0)}S$$
$$S = \begin{bmatrix} s_0 & s_2 & s_3 & s_1 \\ s_1 & s_3 & s_0 & s_2 \\ s_2 & s_0 & s_1 & s_3 \\ s_3 & s_1 & s_2 & s_0 \end{bmatrix}$$

(d)
$$SP_1^{(1)} = P_2^{(0)}S \qquad SP_2^{(1)} = P_3^{(0)}S \qquad SP_3^{(1)} = P_1^{(0)}S$$
$$S = \begin{bmatrix} s_0 & s_2 & s_1 & s_3 \\ s_1 & s_3 & s_2 & s_0 \\ s_2 & s_0 & s_3 & s_1 \\ s_3 & s_1 & s_0 & s_2 \end{bmatrix}$$

It may be pointed out that $P_2^{(0)}$ and $P_1^{(1)}$ are their own inverses and the group G_0 is cyclic. Further $P_1^{(0)}*P_1^{(0)}=P_3^{(0)}*P_3^{(0)}=P_2^{(0)}$. Also conditions (c) and (d) are obtained by commuting S and P in (a) and (b) respectively. Similar results may be expected with groups G_0 and G_2.

When groups G_0 and G_3, i.e. cyclic and dyadic, were taken and investigated it was found that a system matrix S having rank four could not be obtained under any condition. Since cyclic group G_0 gives some system matrices S of rank four with groups which are not dyadic it may be said that a dyadic group G_3 when considered alongwith any other group would not yield any system matrix S of rank four.

Thus it has been shown that it is possible to enlarge the class of 1-D R-P system if it be defined according to Eq. (2.5). It may be of interest to find out the conditions under which degenerate cases would not arise.

A two-dimensional discrete signal $x \in V$ is an array with M rows and N columns. Let T be a 2-D finite discrete linear system on V and G_i, G_j be transitive abelian permutation groups of order M and N respectively. Then T is said to be a 2-D P-I system relative to G_i and G_j if for every $x \in V$, every $P_k \in G_i$, $k \in Z_M$ and every $P_l \in G_j$, $l \in Z_N$,

$$T(P_k x) = P_k (Tx) \quad \ldots (2.6)$$

and

$$T(x P_l^T) = (T x) P_l^T \quad \ldots (2.7)$$

where P_k is the M x M matrix that results from permuting by P_k the rows of the identity matrix of size M , and P_l is the N X N matrix that results from permuting by P_l the columns of the identity matrix of size N.

The set of all such systems which satisfy Eq. (2.6) and (2.7) constitute a class of 2-D P-I system relative to groups G_i and G_j. The two conditions can also be written as

$$T (P_k x P_l^T) = P_k(T x) P_l^T \quad \ldots (2.8)$$

or

$$T(P_k x P_l^T) = P_k y P_l^T \quad \ldots (2.9)$$

where $Tx = y$

This relation expresses the fact that the effect of permuting the rows of the input signal to a 2-D P-I system by members of group G_i and its columns by members of group G_j is to permute the rows and columns of the output signal exactly in the same manner.

If the defining Eqs. (2.6) and (2.7) be modified in accordance with Eq. (2.4) as

$$T_1 (P_k x) = P_k'(T_1 x) \qquad P_k , P_k' \in G_i \quad \ldots (2.10)$$

and

$$(x P_l^T) T_2 = (x T_2)P_l'^T , P_l, P_l' \in G_j \quad \ldots (2.11)$$

Then the set of all such systems T_1 and T_2 which satisfy Eqs. (2.10) and (2.11) constitute as class of two-dimensional system having some prescribed permutation properties. Such systems may be termed as two-dimensional reciprocal-permutation system (2-D R-P system) relative to groups G_i and G_j. As has already been pointed out in case of 1-D R-P systems not all combinations of P_k, P_k' ε G_i and P_l, P_l' ε G_j would yield T_1 and T_2 of ranks equal to their orders. The 2-D R-P system would result in 2-D P-I system when $P_k = P_k'$ and $P_l = P_l'$. It may be of interest to find out the conditions under which T_1 and T_2 would have ranks equal to their orders.

A wider class of 2-D R-P system, say generalized 2-D R-P system, could be obtained by modifying Eqs. (2.6) and (2.7) in accordance with Eq. (2.5). In most of the cases the resulting system matrices T_1 and/or T_2 would not have ranks equal to their orders. It may be interesting to derive the conditions under which degenerate cases would not arise.

2.4 SYNCHRONOUS TRANSLATION INVARIANT TRANSFORMS

In this section some more systems are derived which are permutation-invariant to a class of permutations. If circulant matrices B with the property that

$$B\,B^T = I \text{ modulo } 2 \qquad \ldots (2.12)$$

are obtainable they could be used as transform kernels to transform binary signals with 0 and 1 as possible values with the understanding that matrix algebra is defined over G F (2) . In other words the sum is replaced by EOR and multiplication by AND in the definition of a transform by matrix multiplication.

A (v, k, λ) cyclic difference set [10,55] is normally characterized by its incidence matrix A and that A is a square matrix of order v with 0 and 1 as entries and satisfies the relations [10]

$$A A^T = A^T A = (k - \lambda) I + J \quad \ldots (2.13)$$

$$A J = J A = k J \quad \ldots (2.14)$$

where J is a matrix of all ones.

If $\lambda \equiv 0$ modulo 2 and $k \equiv 1$ modulo 2, A can be used as matrix B of Eq. (2.12). That is, orthonormal circulants of Eq. (2.12) can be obtained from some $(v, k\ \lambda)$ cyclic difference sets directly. If $k \equiv 0$ modulo 2 and $\lambda \equiv 1$ modulo 2, the desired matrix B can be obtained from A as

$$B = \begin{bmatrix} 0 & j \\ j^T & A \end{bmatrix}, \quad j = (1, 1, \ldots, 1) \quad \ldots (2.15)$$

That is, orthonormal circulants of Eq. (2.12) can be obtained from some (v, K, λ) cyclic difference sets by a minor modification.

A (v, p_k, p_λ) cyclic difference set is an arrangement of v elements $x_0, x_1, \ldots, x_{v-1}$ into v sets $S_0, S_1, \ldots, S_{v-1}$ such that S_i contains k_i elements and S_i, S_j have λ_{ij} elements in common such that [44]

$$\lambda_{ij} \equiv p_\lambda \text{ modulo } 2 \qquad i,j=0,1,2,\ldots,v-1 \qquad \ldots(2.16)$$

$$k_i \equiv p_k \text{ modulo } 2$$

If A is an incidence matrix of the (v, p_k, p_λ) cyclic difference set analagous to Eqs. (2.13) and (2.14) then

$$A A^T = A^T A = ((p_k - p_\lambda)I + p_\lambda J) \text{ modulo } 2 \quad \ldots(2.17)$$

$$AJ = JA = p_k J \text{ modulo } 2 \qquad \ldots(2.18)$$

Two important cases arise.

Case I

If $p_k = 1$ and $p_\lambda = 0$ ...(2.19)

then Eq. (2.17) becomes

$$A A^T = A^T A = I \text{ modulo } 2 \qquad \ldots(2.20)$$

so that the incidence matrix A can be used as matrix B of Eq. (2.12).

Case II

If $p_k = 0$ and $p_\lambda = 1$ ...(2.21)

then Eq. (2.17) becomes

$$A\,A^T = A^T A = (I+J) \text{ modulo } 2 \qquad \ldots (2.22)$$

Then

$$B = \begin{bmatrix} 0 & j \\ j^T & A \end{bmatrix}, \quad j = (1,1,\ldots,1) \qquad \ldots (2.23)$$

where j is a row vector of all 1's satisfies Eq. (2.12) provided $v \equiv 1$ modulo 2. The (v, p_k, p_λ) cyclic difference sets can be used to generate similar sets of higher order.

A (v, p_k, p_λ) difference set is said to be of type I if

$$v \equiv 0 \text{ modulo } 2,\; p_k = 1,\; p_\lambda = 0 \qquad \ldots (2.24)$$

and of type II if

$$v \equiv 1 \text{ modulo } 2,\; p_k = 0,\; p_\lambda = 1 \qquad \ldots (2.25)$$

If A_1 and A_2 are incidence matrices of $(v_1, p_{k_1}, p_{\lambda_1})$ and $(v_2, p_{k_2}, p_{\lambda_2})$ cyclic difference sets respectively then [44]

$$X = \begin{bmatrix} A_1 & J \\ J^T & A_2 \end{bmatrix} \qquad \ldots (2.26)$$

would be an incidence matrix of a $(v_1 + v_2, p_k, p_\lambda)$ cyclic difference set of type I if $(v_1, p_{k_1}, p_{\lambda_1})$ and $(v_2, p_{k_2}, p_{\lambda_2})$ cyclic difference sets belong to the same type and X would be an incidence matrix of a $(v_1 + v_2, p_k, p_\lambda)$ cyclic difference set of type II if

$(v_1, p_{k_1}, p_{\lambda_1})$ and $(v_2, p_{k_2}, p_{\lambda_2})$ cyclic difference sets belong to different types.

If X is an incidence matrix of a $(v_1+v_2, p_k, p_\lambda)$ cyclic difference set of type I then X can be used as matrix B of Eq. (2.12). Similarly if X is an incidence matrix of a $(v_1 + v_2, p_k, p_\lambda)$ cyclic difference set of type II then

$$y = \begin{bmatrix} 0 & j \\ j^T & X \end{bmatrix} \qquad \ldots (2.27)$$

can be used as matrix B of Eq. (2.12). Eq. (2.27) can be taken as a general method of obtaining a $(v+1, p_k, p_\lambda)$ cyclic difference set of type I from a (v, p_k, p_λ) cyclic difference set of type II. If

$$\begin{aligned} v &\equiv 0 \text{ modulo } 2 \\ k &\equiv 1 \text{ modulo } 2 \\ \lambda &\equiv 0 \text{ modulo } 2 \end{aligned} \qquad \ldots (2.28)$$

then a (v, k, λ) cyclic difference set qualifies to be a (v, p_k, p_λ) cyclic difference set of type I and if

$$\begin{aligned} v &\equiv 1 \text{ modulo } 2 \\ k &\equiv 0 \text{ modulo } 2 \\ \lambda &\equiv 1 \text{ modulo } 2 \end{aligned} \qquad \ldots (2.29)$$

then a (v, k, λ) cyclic difference set qualifies to be a (v, p_k, p_λ) cyclic difference set of type II. A list of known cyclic difference sets is available [10].

If $[S_1, S_2]^T$ be an input vector of length $(v_1 + v_2)$ and matrix X given by Eq. (2.26) be used as a transform kernel of order $(v_1 + v_2)$ then the output vector $[R_1, R_2]^T$ will also be of length $(v_1 + v_2)$, where S_i, R_i are sequences of lengths v_i at input and output respectively. It can be infered from the structure of the kernel used that the sequence R_i is decided by A_i alone. Thus if the input sequences S_1 and S_2 are cyclically shifted within themselves then the resulting output sequences R_1 and R_2 would also undergo identical cyclic shifts within themselves. Similar permutation properties can be derived when other orthonormal B matrices derived from (v, p_k, p_λ) sets (which in turn are derived from (v, k, λ) sets) are used as transform kernels.

It can be noted for completeness that circulant matrices B of Eq. (2.12) are useful in defining self-dual codes [44].

CHAPTER - 3

PERMUTATION PROPERTIES OF SOME TRANSFORMS

This chapter begins with a review of some of the permutation properties of discrete Fourier transform (D F T) and Hadamard transform (H T). The relation between D F T of one-dimensional sequence written as a two-dimensional array is well known. It envolves columnwise D F T, twiddling and rowwise D F T. A new transform has been defined where the first operation of columnwise D F T is replaced by columnwise H T and twiddling factors are suitably chosen. Since the transform has H T and D F T as component transforms it can be said that this transform would exhibit permutation properties which could be derived from a knowledge of the permutation properties of H T and D F T. It is known that D F T exhibits modular permutation property and H T exhibits bit-plane permutation-property among others.

3.1 MODULAR PERMUTATION

The 'discrete Fourier transform', abbreviated as D F T, A(I) of a sequence of N samples a(i), i = 0,1,2, .., N-1 is

$$A(I) = \sum_{i=0}^{N-1} a(i) W_N^{iI} \qquad I=0,1,2,\dots,N-1 \quad \dots (3.1)$$

$$= \sum_{i=0}^{N-1} a(i) \exp\left(- j \frac{2\pi i I}{N}\right)$$

where $W_N = \exp(-j\frac{2\pi}{N})$, $j = \sqrt{-1}$

There exists an inverse D F T (I D F T), which is

$$a(i) = \frac{1}{N}\sum_{I=0}^{N-1} A(I)W_N^{iI} \quad i=0,1,2,\ldots,N-1 \quad \ldots (3.2)$$

The I D F T of A(I) can also be written as D F T by making the substitution $k = N-I$ and writing

$$\frac{1}{N}\sum_{I=0}^{N-1} A(I)W_N^{iI} = \frac{1}{N}\sum_{k=0}^{N-1} A(N-k)W_N^{i(N-k)}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} A(N-k)\, W_N^{-ik} \quad \ldots (3.3)$$

This relation can be interpreted to mean that the I D F T of a sequence is $\frac{1}{N}$ times the D F T of the reverse of the sequence.

It is well known that D F T exhibits permutation properties, i.e., if the sequence of signal samples a(i), i=0,1,2,...., N-1 is permuted according to modular (defined presently) permutation operator and D F T taken then the resulting sequence would be as if the sequence of transform samples A(I), I = 0,1,2,..,N-1 are permuted according to some other modular permutation operator. The modular permutation operator P(p,N) is defined as [33]

$$P(p,N)\, a(i) = a\,[\,pi \text{ modulo } N] \quad i=0,1,2,\ldots,N-1 \quad \ldots (3.4)$$

where p is an integer such that it has no factors in common with N. Modular permutation is a sumpsimus for what otherwise is called decimation [23].

If P(q,N) be the inverse modular permutation operator then

$$P(q,N)\, P(p,N)\, a(i) = a(i) \qquad i=0,1,2,..,N-1 \qquad \dots (3.5)$$

Now,

$$P(q,N)\, P(p,N)\, a(i) = a[q(p\ i \text{ modulo } N) \text{ modulo } N]$$
$$= a\,[p\ q\ i \text{ modulo } N] \qquad \dots (3.6)$$

From Eqs. (3.5) and (3.6) one gets

$$p\ q \text{ modulo } N = 1$$

It is clear that for any value of p, q is an unique integer on the range $0 \leq q \leq N-1$. If the D F T operator is represented by F then

$$F[\,P(p,N)\, a(i)] = P(q,N)\, A(I)\ ,\ A(I) = Fa(i)$$
$$i,I = 0,1,2,..,N-1$$

or

$$F\ [a((pi))_N] = A((q\ I))_N \qquad \dots (3.7)$$

where $((x))_N$ is x modulo N

A particular case of modular permutation is that when p = q. The resulting modular permutation operator is called self-inverting permutation operator and gives

$$F[a((p'i))_N] = A\ ((p'I))_N \qquad p = q = p' \qquad \dots (3.8)$$

Some typical self-inverting permutation operators are P(1,16), P(7,16), P(9,16) and P(N-1, N).

3.2 BIT-PLANE PERMUTATION

It is well known that Hadamard matrices of order $M = 2^n$, n being a nonzero positive integer, always exist. A normalized Hadamard matrix of order M can be written in 'natural form' or 'ordered form'. In the natural form it is written as

$$H(X,Y) = (-1)^{\sum_{t=0}^{n-1} X^t Y^t \text{ modulo } 2} \qquad \ldots (3.9)$$

where $X_{decimal} = (X^{n-1}\, X^{n-2} \ldots X^t \ldots X^1\, X^0)_{binary}$

and $Y_{decimal} = (Y^{n-1}\, Y^{n-2} \ldots Y^t \ldots Y^1\, Y^0)_{binary}$

In ordered form it is written as

$$H(X, Y) = (-1)^{\sum_{t=0}^{n-1} g^t (X)\, Y^t \text{ modulo } 2} \qquad \ldots (3.10)$$

where

$$\begin{aligned} g^0(X) &= X^{n-1} \\ g^1(X) &= X^{n-1} + X^{n-2} \\ g^2(X) &= X^{n-2} + X^{n-3} \\ &\vdots \\ g^t(X) &= X^{n-t} + X^{n-t-1} \\ &\vdots \\ g^{n-1}(X) &= X^1 + X^0 \end{aligned}$$

In this form the number of sign changes in any row is more than that in the preceeding rows. Hadamard transform of order $M = 2^n$ is also known as Walsh-Fourier transform. In what follows a review would be made of known permutation operators P and reciprocal (with respect to HT) permutation operators P^r such that

$$H[P\ a(i)] = P^r[H\ a(i)] \quad i,I=0,1,2,\ldots,M-1 \quad \ldots (3.11)$$
$$= P^r\ A(I)$$

where H is the Hadamard transform operator in natural form so that

$$H\ a(i) = A(I) \qquad i,I=0,1,2,\ldots,M-1 \quad \ldots (3.12)$$

i.e., $A(I)$, $I=0,1,2,\ldots,M-1$ are the HT samples of the signal sequence $a(i)$, $i=0,1,2,\ldots,M-1$. An inverse permutation operator, P^{-1}, can be defined as

$$P^{-1}\ P\ a(i)=a(i) \qquad i=0,1,2,\ldots,M-1 \quad \ldots (3.13)$$

A permutation operator is termed as self-inverting permutation operator if it satisfies $P^{-1}=P$. Moharir [36] has defined two permutation operators, namely, selective bit-complementation permutation operator and bit-plane permutation operator.

Let $a(i)$, $i=0,1,2, \ldots, 2^n-1$ be the sequence of signal samples and $P(m)$, $m = 0,1,2,,\ldots,2^n-1$ be the selective bit-complementation permutation operator. If the indices i and m are represented in n-digit binary notation, the operator $P(m)$, $m = 0,1,2,\ldots,2^n-1$ prescribes complementation in those locations in binary representation of i in which the binary representation of $P(m)$ has

1's. It is evident that P(m) is self-inverting. This permutation has been illustrated for n=3 by Moharir [36] and is reproduced in Table 3.1. If a(i), $i=0,1,2,\ldots,2^n-1$ be the sequence of input signal samples and A(I), $I=0,1,2,\ldots,2^n-1$ its H T samples in 'natural order' then it has been shown by Moharir [36] that permuting a(i) according to the above permutation operator before H T has the effect of changing the signs of some of the H T samples A(I) but leaving the sequence unpermuted. In other words A(I) is invariant to this permutation operator. In fact H T samples A(I) get multiplied by $(-1)^{f(m,I)}$ where

$$f(m,I) = \sum_{t=0}^{n-1} m^t\, I^t \qquad \ldots (3.14)$$

and $m_{decimal} = (m^{n-1}\, m^{n-2} \ldots m^t \ldots m^1\, m^0)_{binary}$

$$I_{decimal} = (I^{n-1}\, I^{n-2} \ldots I^t \ldots I^1\, I^0)_{binary}$$

Another permutation property exhibited by H T is bit-plane permutation. The bit-plane permutation operator P(a b ... k) permutes $a(i^{n-1}\, i^{n-2} \ldots i^1\, i^0)_{binary}$ into $a(i^a\, i^b \ldots i^k)_{binary}$ where $a, b, \ldots, k = 0,1,2,\ldots,n-1$ and $a \neq b \ldots \neq k$. It has been shown [36] that bit-plane permutation operator is self-reciprocal w.r.t. H.T. of order 2^n. Then

$$H\, P(a\, b \ldots k)\, a(i) = P(a\, b \ldots k)\, H\, a(i) \qquad \ldots (3.15)$$

$$= P(a\, b \ldots k)\, A(I)$$

$$i, I = 0,1,2,\ldots,2^n-1$$

TABLE-3.1 : Selective Bit Complementation Permutation

$P(m)_{dec.}$ / $a(i)_{dec}$	$P(m)_{bin}$ / $a(i)_{bin.}$	P(0) P(000)	P(1) P(001)	P(2) P(010)	P(3) P(011)	P(4) P(100)	P(5) P(101)	P(6) P(110)	P(7) P(111)
a(0)	a(000)	a(000)	a(001)	a(010)	a(011)	a(100)	a(101)	a(110)	a(111)
a(1)	a(001)	a(001)	a(000)	a(011)	a(010)	a(101)	a(100)	a(111)	a(110)
a(2)	a(010)	a(010)	a(011)	a(000)	a(001)	a(110)	a(111)	a(100)	a(101)
a(3)	a(011)	a(011)	a(010)	a(001)	a(000)	a(111)	a(110)	a(101)	a(100)
a(4)	a(100)	a(100)	a(101)	a(110)	a(111)	a(000)	a(001)	a(010)	a(011)
a(5)	a(101)	a(101)	a(100)	a(111)	a(110)	a(001)	a(000)	a(011)	a(010)
a(6)	a(110)	a(110)	a(111)	a(100)	a(101)	a(010)	a(011)	a(000)	a(001)
a(7)	a(111)	a(111)	a(110)	a(101)	a(100)	a(011)	a(010)	a(001)	a(000)

where H is the Hadamard transform operator in natural form. The above equation can be written as

$$H\,P(a\ b\ \dots\ k)\,a(i^{n-1}i^{n-2}\ \dots\ i^{0})_{binary}$$

$$= P(a\ b\ \dots\ k)\ A(I^{n-1}\ I^{n-2}\dots I^{0})_{binary}$$

$$H\ a(i^{a}\ i^{b}\ \dots i^{k})_{binary} = A(I^{a}\ I^{b}\dots\ I^{k})_{binary} \qquad \dots (3.16)$$

Thus permuting the signal sample sequence a(i), $i=0,1,2,..,2^n-1$ by bit-plane permutation operator and then taking H T of this permuted sequence is equivalent to subjecting H T samples A(I) of unpermuted sequence a(i) to identical bit-plane permutation operator. It is known that bit-plane permutation operators are not necessarily self-inverting , but the inverse of any bit-plane permutation operator is some bit-plane permutation operator. This permutation has been illustrated for n=3 by Moharir [36] and is reproduced in Table-3.2.

The simple relationship between the permutation operator and the reciprocal permutation operator, when the length of the signal sample sequence a(i) is 2^n, n being an integer, opens the possibility of permuting a(i) before H T at the 'sending end' of a communication system, and doing the reciprocal permutation on received transform samples followed by inverse Hadamard transformation at the 'receiving end' of the communication system . If the length

TABLE-3.2 : Bit-Plane Permutation

$a(i)_{binary}$ \ P(ab...k)	P(210)	P(201)	P(120)	P(102)	P(021)	P(012)
a(000)	a(000)	a(000)	a(000)	a(000)	a(000)	a(000)
a(001)	a(001)	a(010)	a(001)	a(010)	a(100)	a(100)
a(010)	a(010)	a(001)	a(100)	a(100)	a(001)	a(010)
a(011)	a(011)	a(011)	a(101)	a(110)	a(101)	a(110)
a(100)	a(100)	a(100)	a(010)	a(001)	a(010)	a(001)
a(101)	a(101)	a(110)	a(011)	a(011)	a(110)	a(101)
a(110)	a(110)	a(101)	a(110)	a(101)	a(011)	a(011)
a(111)	a(111)	a(111)	a(111)	a(111)	a(111)	a(111)

of the signal sample sequence a(i) is $M = 2^n$ there are n! possible bit plane permutation operators-of course one of these would leave the sequence unpermuted. Instead of transmitting the H T samples of a(i) one can think of transmitting the H T samples of permuted a(i). The information regarding the bit-plane permutation operator used for permuting a(i) could be made known to the authorized receiver with the help of a synchronous shift-register sequence generator.

The selective-bit complementation permutation is not suitable from application point of view as the modulus of H T samples is invariant to this permutation on input signal sample sequence.

3.3 FOURIER-TWIDDLED KRONECKER PRODUCTS

If a transform kernel could be expressed as matrix product of component transform kernels with many zero entries then it is known that computationally efficient algorithms exist for the transform [6,8,15]. Computationally efficient algorithms for D F T of composite order are based on this fact. Alternatively, if the transform kernel could be expressed as a Kronecker product of the component transform kernels then also computationally efficient algorithms for the transform are known to exist [7,37,39]. Computationally efficient algorithms are based on this Kronecker decomposition are known for many discrete transforms.

Let $a(i)$, $i = 0,1,2,\ldots,MN-1$ be a discrete signal sample sequence of length MN and $A(I)$, $I = 0,1,2,\ldots,MN-1$ its D F T, then [43]

$$A(I) = \sum_{i=0}^{MN-1} a(i)\, W_{MN}^{iI} \qquad \ldots (3.17)$$

where $W_{MN} = \exp\left(-j \frac{2\pi}{MN}\right)$

Writing i, I as

$$\begin{aligned} i &= i_1 N + i_2 \qquad i_1, I_1 = 0,1,2,\ldots,M-1 \\ I &= I_1 + I_2 M \qquad i_2, I_2 = 0,1,2,\ldots,N-1 \end{aligned} \qquad \ldots (3.18)$$

where i_1 and I_2 are integeral parts of (i/N) and (I/M) respectively so that

$$\begin{aligned} i_1 &= [i/N]\ ,\ i_2 = i - N[i/N] \\ I_2 &= [I/M]\ ,\ I_1 = I - M[I/M] \end{aligned} \qquad \ldots (3.19)$$

where $[x/y]$ is the integral part of (x/y).

Eq. (3.17) can be rewritten as

$$\begin{aligned} A(I_2 M + I_1) &= \sum_{i=0}^{MN-1} a(i)\, W_{MN}^{(I_2M+I_1)i} \\ &= \sum_{i_1=0}^{M-1} \sum_{i_2=0}^{N-1} a(i_1N+i_2) W_{MN}^{(I_2M+I_1)(i_1N+i_2)} \end{aligned}$$

$$= \sum_{i_2=0}^{N-1} W_{MN}^{i_2 I_2 M} [W_{MN}^{i_2 I_1} \sum_{i_1=0}^{M-1} a(i_1 N+i_2) W_{MN}^{i_1 I_1 N}], W_{MN}^{i_1 I_2 MN} = 1$$

$$= \sum_{i_2=0}^{N-1} W_N^{i_2 I_2} [W_{MN}^{i_2 I_1} \sum_{i_1=0}^{M-1} a(i_1 N+i_2) W_M^{i_1 I_1}] \qquad \ldots (3.20)$$

$$= \text{DFT} \; W_{MN}^{i_2 I_1} \times [\text{DFT} [a(i_1 N+i_2)]] \qquad \ldots (3.21)$$

N points	M points
mapping	mapping
$i_2 \rightarrow I_2$	$i_1 \rightarrow I_1$

Let $a(i)$, $i = 0,1,2,\ldots,MN-1$ be considered to have been chopped into several sequences and written into an array of M rows of N terms each. Further let $A(I)$, $I = 0,1,2,\ldots, MN-1$ be also chopped into several sequences and written into an array of N columns of M terms each. Then i_1 and I_1 can be considered as indices along columns , and i_2 and I_2 as indices along rows. The inner summation over i_1 in Eq. (3.20) can be thought of as columnwise M term D F T of the array of a (i) with i_2 held constant. This would give an 'intermediate array' of dimension M X N. The element in the I_1^{th} row and i_2^{th} column is multiplied by $W_{MN}^{i_2 I_1}$. The factors $W_{MN}^{i_2 I_1}$ have been called 'twiddle factors'. This operation can also be represented in terms of the Hadamard product of matrices. The Hadamard product $C = [c(i,j)]$ of matrices $A = [a(i,j)]$ and

B = [b(i,j)] is defined by the relation c(i,j) = a(i,j) b(i,j). Obviously the matrices A,B and C have the same dimensions [52]. The outer summation over i_2 in Eq. (3.20) can be thought of as rowwise N terms D F T of the 'intermediate array' obtained after columnwise M term D F T of a (i), holding i_2 as constant, and twiddling by factors $W_{MN}^{i_2 I_1}$. The Fourier coefficients are obtained by reading the final array column by column. If the input signal sample sequence a(i), i = 0,1,2,...., MN-1, can be written as a two-dimensional array of M rows and N columns read row by row as

$$\begin{bmatrix} a(0) & a(1) & a(2) & \cdots & a(N-1) \\ a(N) & a(N+1) & a(N+2) & \cdots & a(2N-1) \\ a(2N) & a(2N+1) & a(2N+2) & \cdots & a(3N-1) \\ \vdots & \vdots & \vdots & & \vdots \\ a(\overline{M-2}N) & a(\overline{M-2}N+1) & a(\overline{M-2}N+2) & \cdots & a(\overline{M-1}N-1) \\ a(\overline{M-1}N) & a(\overline{M-1}N+1) & a(\overline{M-1}N+2) & \cdots & a(MN-1) \end{bmatrix}$$

The Fourier transform samples A(I) = DFT a(i), I = 0,1,2,....,MN-1 are obtained as a two-dimensional array of M rows and N columns read column by column as

$$\begin{bmatrix} A(0) & A(M) & A(2M) & \cdots & A(\overline{N-1}M) \\ A(1) & A(M+1) & A(2M+1) & \cdots & A(\overline{N-1}M+1) \\ A(2) & A(M+2) & A(2M+2) & \cdots & A(\overline{N-1}M+2) \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ A(M-2) & A(2M-2) & A(3M-2) & \cdots & A(MN-2) \\ A(M-1) & A(2M-1) & A(3M-1) & \cdots & A(MN-1) \end{bmatrix}$$

The process of obtaining FT samples $A(I)$, $I = 0,1,2,\ldots,MN-1$ of input signal sample sequence $a(i)$, $i = 0,1,2,\ldots,MN-1$ as given by Eq. (3.21) is illustrated in Fig. 3.1 [23]. If the twiddle factors $W_{MN}^{i_2 I_1}$ are ignored, then Eq. (3.21) can be interpreted as columnwise M-term D F T followed by rowwise N-term D F T. It has been shown by Moharir [42] that this can be thought of as a transformation based on a kernel which is a Kronecker product of two component kernels : kernel of M-term D F T and kernel of N-term D F T. In addition the transform samples would now be read row by row as against column by column when twiddle factors are taken into consideration. Except for this difference MN-term DFT can be obtained by Fourier-twiddling the Kronecker product of M-term D F T and N-term D F T.

This result has very far reaching consequences

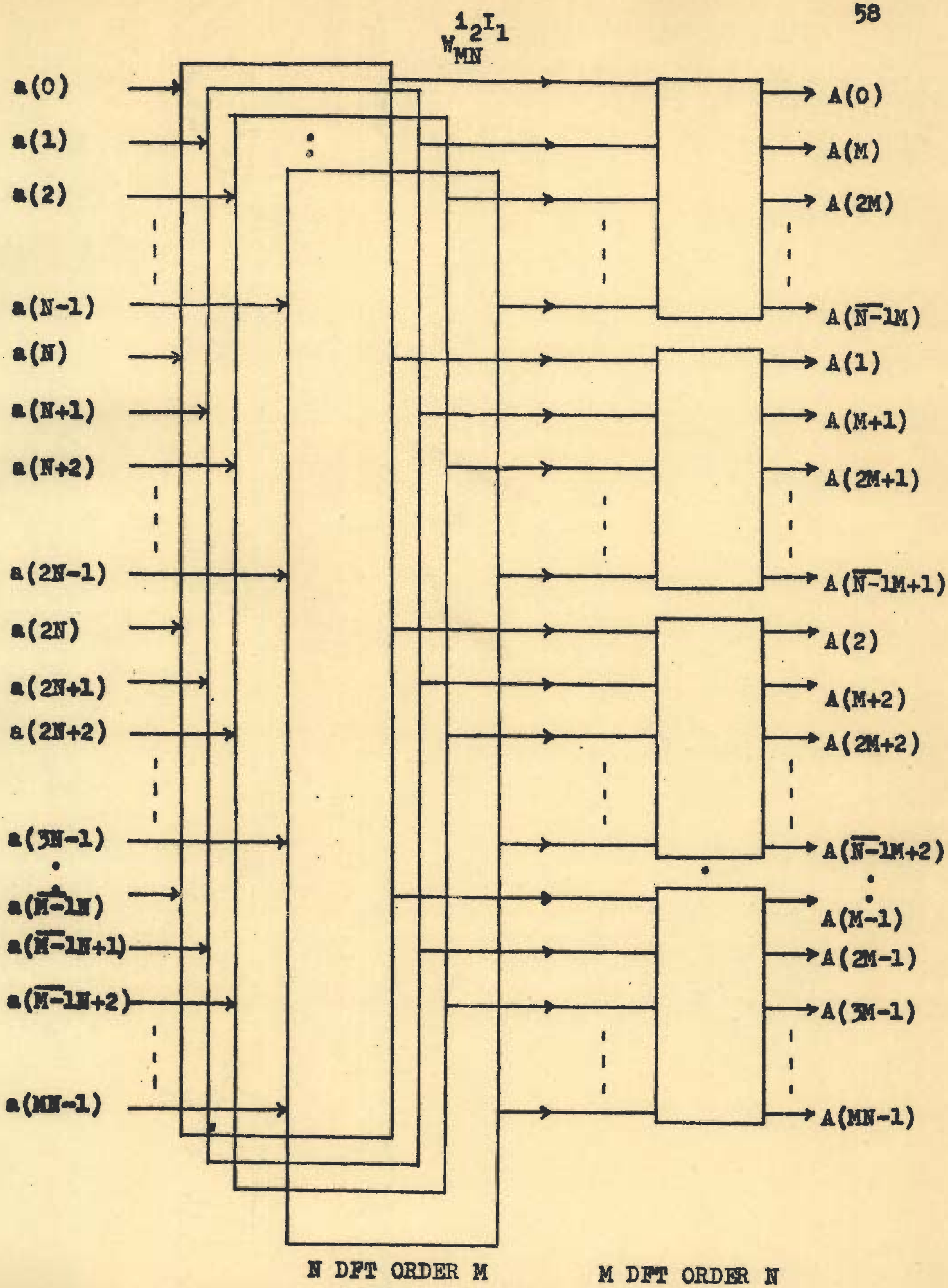


FIG.3.1 TREE-GRAPH FOR DFT OF 2-D ARRAY

and in fact can be exploited to define a wide class of orthonormal transforms. If the M-term D F T and N-term D F T are replaced by any other M-term and N-term discrete orthonormal transforms having kernels $Q_M(I_1,i_1)$ and $Q_N(I_2,i_2)$ respectively. Further if $W_x^{i_2I_1}$, x not necessarily equal to MN, be the twiddling factors instead of $W_{MN}^{i_2I_1}$ then one can get [43]

$$B(I_2M+I_1)=\sum_{i_2=0}^{N-1} Q_N(I_2,i_2)\ [W_x^{i_2I_1} \sum_{i_1=0}^{M-1} b(i_1N+i_2)Q_M(I_1,i_1)] \qquad \ldots (3.22)$$

It has been reported by Moharier [43] that a wide class of complete orthonormal transforms can be obtained by Fourier-twiddling the Kronecker product of orthonormal kernels as in Eq. (3.22).

3.41 FOURIER-TWIDDLED H-DF TRANSFORM

In section 3.3 a method for fast computation of FT samples of a sequence of length MN has been given. The method consists of writing the one-dimensional array of length MN as a two-dimensional array of M rows and N columns read row by row. The first step is to write M-term columnwise D F T. The elements of the resulting two-dimensional array are then multiplied by twiddling factors which depend upon M,N and the row and column in which the particular element is contained, or, in other words the Hadamard product of the resulting array and the array of twiddle factors is taken. The array so obtained is then subjected to N-term rowwise D F T to get FT samples as a two-dimensional array of M rows and N columns read column by column. It has also been pointed out that this method of obtaining the FT samples can be used to compute other transform samples by replacing the two D F T operations with suitable orthonormal transforms. In this section a new transform would be defined by doing M-term columnwise transformation with Hadamard transformation instead of D F T. The N-term rowwise transformation would remain D F T.

Let a(i), i = 0,1,2,..., MN-1 be signal sample sequence, $M = 2^s$. This is written as a two-dimensional array of M rows and N columns as

$$[a(i)] = [a(i_1 N + i_2)] = \begin{array}{c|ccccc} i_1 \downarrow \; \rightarrow i_2 & 0 & 1 & 2 & \cdots & N-1 \\ \hline 0 & a(0) & a(1) & a(2) & \cdots & a(N-1) \\ 1 & a(N) & a(N+1) & a(N+2) & \cdots & a(2N-1) \\ 2 & a(2N) & a(2N+1) & a(2N+2) & \cdots & a(3N-1) \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ M-2 & a(\overline{M-2}N) & a(\overline{M-2}N+1) & a(\overline{M-2}N+2) & \cdots & a(\overline{M-1}N-1) \\ M-1 & a(\overline{M-1}N) & a(\overline{M-1}N+1) & a(\overline{M-1}N+2) & \cdots & a(MN-1) \end{array} \quad \ldots (3.23)$$

where i_1, i_2 are defined by Eq. (3.19).

The normalized Hadamard matrix in the 'natural form' as given in Eq. (3.9) can be rewritten as

$$H(i_1, I_1) = (-1)^{\sum_{t=0}^{s-1} i_1^t I_1^t \text{ modulo } 2} \qquad i_1 I_1 = 0,1,2,\ldots,M-1 \qquad \ldots (3.24)$$

This would be a square two-dimensional array of order M. This would be used as transformation kernel to achieve M-term columnwise Hadamard transformation. The resulting two-dimensional 'intermediate array' $[h_{i_2}^{(I_1)}]$ would be of dimension MXN and can be written as

$$[h_{i_2}^{(I_1)}] = \begin{array}{c|ccccc} \downarrow I_1 \backslash i_2 \rightarrow & 0 & 1 & 2 & \ldots & N-1 \\ \hline 0 & h_0^{(0)} & h_1^{(0)} & h_2^{(0)} & \ldots & h_{N-1}^{(0)} \\ 1 & h_0^{(1)} & h_1^{(1)} & h_2^{(1)} & \ldots & h_{N-1}^{(1)} \\ 2 & h_0^{(2)} & h_1^{(2)} & h_2^{(2)} & \ldots & h_{N-1}^{(2)} \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ M-2 & h_0^{(M-2)} & h_1^{(M-2)} & h_2^{(M-2)} & & h_{N-1}^{(M-2)} \\ M-1 & h_0^{(M-1)} & h_1^{(M-1)} & h_2^{(M-1)} & \ldots & h_{N-1}^{(M-1)} \end{array} \qquad \ldots (3.25)$$

Mathematically any element of the above intermediate array can be expressed as

$$h_{i_2}^{(I_1)} = \sum_{i_1=0}^{M-1} a(i_1N+i_2)(-1)^{\sum_{t=0}^{s-1} i_1^t I_1^t \text{ modulo } 2} \qquad \ldots (3.26)$$

The next step is to multiply each element of this 'intermediate array' by twiddling factors $W_{MN}^{i_2 I_1}$ to get a two-dimensional array, $[H_{i_2}^{(I_1)}]$ of dimension MXN.

$$H_{i_2}^{(I_1)} = h_{i_2}^{(I_1)} W_{MN}^{i_2 I_1} \qquad \ldots (3.27)$$

$$H_{i_2}^{(I_1)} = W_{MN}^{i_2 I_1} \sum_{i_1=0}^{M-1} a(i_1N+i_2)(-1)^{\sum_{t=0}^{s-1} i_1^t I_1^t \text{ modulo } 2} \qquad \ldots (3.28)$$

$$[H_{i_2}^{(I_1)}] = \begin{bmatrix} I_1 \downarrow \; i_2 \rightarrow & 0 & 1 & \ldots & N-1 \\ 0 & W_{MN}^{0} h_o^{(0)} & W_{MN}^{0} h_1^{(0)} & \ldots & W_{MN}^{0} h_{N-1}^{0} \\ 1 & W_{MN}^{0} h_o^{(1)} & W_{MN}^{1} h_1^{(1)} & \ldots & W_{MN}^{(N-1)} h_{N-1}^{(1)} \\ 2 & W_{MN}^{0} h_o^{(2)} & W_{MN}^{2} h_1^{(2)} & \ldots & W_{MN}^{2(N-1)} h_{N-1}^{(2)} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ M-2 & W_{MN}^{0} h_o^{(M-2)} & W_{MN}^{(M-2)} h_1^{(M-2)} & \ldots & W_{MN}^{(M-2)(N-1)} h_{N-1}^{(M-2)} \\ M-1 & W_{MN}^{0} h_o^{(M-1)} & W_{MN}^{(M-1)} h_1^{(M-1)} & \ldots & W_{MN}^{(M-1)(N-1)} h_{N-1}^{(M-1)} \end{bmatrix}$$

$$\ldots (3.29)$$

$$
= \begin{array}{c|ccccc|}
{}_{I_1}\downarrow \overset{\rightarrow i_2}{} & 0 & 1 & 2 & \cdots & N-1 \\
\hline
0 & H_o^{(0)} & H_1^{(0)} & H_2^{(0)} & \cdots & H_{N-1}^{(0)} \\
1 & H_o^{(1)} & H_1^{(1)} & H_2^{(1)} & \cdots & H_{N-1}^{(1)} \\
2 & H_o^{(2)} & H_1^{(2)} & H_2^{(2)} & \cdots & H_{N-1}^{(2)} \\
\cdot & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & & \cdot \\
M-2 & H_o^{(M-2)} & H_1^{(M-2)} & H_2^{(M-2)} & \cdots & H_{N-1}^{(M-2)} \\
M-1 & H_o^{(M-1)} & H_1^{(M-1)} & H_2^{(M-1)} & \cdots & H_{N-1}^{(M-1)}
\end{array} \quad \ldots(3.30)
$$

The final step in the calculation of transform samples A(I) involves the N-term rowwise D F T. This D F T kernel $Q(I_2,i_2)$ would be a square matrix of order N.

$$
Q(I_2,i_2) = \begin{array}{c|ccccc|}
{}_{I_2}\downarrow \overset{\rightarrow i_2}{} & 0 & 1 & 2 & \cdots & N-1 \\
\hline
0 & W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\
1 & W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\
2 & W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\
\cdot & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & & \cdot \\
N-2 & W_N^0 & W_N^{(N-2)} & W_N^{2(N-2)} & & W_N^{(N-1)(N-2)} \\
N-1 & W_N^0 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)}
\end{array} \quad \ldots(3.31)
$$

where $W_N = \exp(-j\frac{2\pi}{N})$ and powers of W_N are taken modulo N.

The transform samples A(I) can be written as

$$A(I) = A(I_2M+I_1) = \sum_{i_2=0}^{N-1} W_N^{i_2I_2} H_{i_2}^{(I_1)}$$

$$= \sum_{i_2=0}^{N-1} W_N^{i_2I_2} W_{MN}^{i_2I_1} \sum_{i_1=0}^{M-1} a(i_1N+i_2)(-1)^{\sum_{=0}^{s-1} i_1^t I_1^t \text{ modulo } 2} \qquad \ldots (3.32)$$

The transform samples A(I) can be written as a two-dimensional array of M rows and N columns read column by column.

$$A(I) = [A(I_2M+I_1)] = \begin{array}{c|ccccc|} I_1 \downarrow \; \rightarrow I_2 & 0 & 1 & 2 & \ldots & N-1 \\ 0 & A(0) & A(M) & A(2M) & \ldots & A(\overline{N-1}M) \\ 1 & A(1) & A(M+1) & A(2M+1) & \ldots & A(\overline{N-1}M+1) \\ 2 & A(2) & A(M+2) & A(2M+2) & \ldots & A(\overline{N-1}M+2) \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ M-2 & A(M-2) & A(2M-2) & A(3M-2) & \ldots & A(MN-2) \\ M-1 & A(M-1) & A(2M-1) & A(3M-1) & \ldots & A(MN-1) \end{array} \qquad \ldots (3.33)$$

Example Let M=4, N=5 and a(i), i=0,1,2,....,19 be given as

[1 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1]

$$[a(i_1 5+i_2)]=\begin{bmatrix}1&0&0&1&0\\0&1&0&0&0\\1&0&1&0&1\\0&1&0&0&1\end{bmatrix}$$

$$[H(i_1,I_1)]=\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}$$

$$[h_{i_2}^{(I_1)}]=\begin{bmatrix}2&2&1&1&2\\2&-2&1&1&0\\0&0&-1&1&-2\\0&0&-1&1&0\end{bmatrix}$$

$$[H_{i_2}^{(I_1)}]=\begin{bmatrix}2+j0&2+j0&1+j0&1+j0&2+j0\\2+j0&-1.9022+j0.618&0.809-j0.5878&0.5878-j0.809&0+j0\\0+j0&0+j0&-0.309+j0.9511&-0.309-j0.9511&1.618+j1.1756\\0+j0&0+j0&0.309+j0.9511&-0.9511-j0.309&0+j0\end{bmatrix}$$

$$A(I) = [A(I_2 4 + I_1)] = \begin{bmatrix} 8 + j0 & 1.618 + j0 & -0.618 + j0 & 0.618 + j0 & 1.618 + j0 \\ 1.4946 - j0.7788 & 1 + j3.0001 & 4.1234 + j0.3969 & 3.8176 - j2.2601 & -0.4356 - j0.3582 \\ 1 + j1.1756 & 1 + j1.9021 & -4 + j0 & 1 - j1.9021 & 1 - j1.1756 \\ -0.6421 + j0.6421 & 1.2601 - j1.2601 & -1.3969 + j1.3969 & 1 - j1 & -0.2212 + j0.2212 \end{bmatrix} \quad \ldots (3.34)$$

where the D F T kernel of order five is

$$\begin{bmatrix} W_5^0 & W_5^0 & W_5^0 & W_5^0 & W_5^0 \\ W_5^0 & W_5^1 & W_5^2 & W_5^3 & W_5^4 \\ W_5^0 & W_5^2 & W_5^4 & W_5^1 & W_5^3 \\ W_5^0 & W_5^3 & W_5^1 & W_5^4 & W_5^2 \\ W_5^0 & W_5^4 & W_5^3 & W_5^2 & W_5^1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + j0 & 1 + j0 & 1 + j0 & 1 + j0 & 1 + j0 \\ 1 + j0 & 0.3090 - j0.9511 & -0.8090 - j0.5878 & -0.8090 + j0.5878 & 0.309 + j0.9511 \\ 1 + j0 & -0.809 - j0.5878 & 0.309 + j0.9511 & 0.309 - j0.9511 & -0.809 + j0.5878 \\ 1 + j0 & -0.809 + j0.5878 & 0.309 - j0.9511 & 0.309 + j0.9511 & -0.809 - j0.5878 \\ 1 + j0 & 0.309 + j0.9511 & -0.809 + j0.5878 & -0.809 - j0.5878 & 0.309 - j0.9511 \end{bmatrix}$$

In the later sections of this chapter investigations have been reported regarding the permuted locations of transform samples in the two-dimensional array when the rows or columns or both of input signal sample sequence a(i), written as a two-dimensional array, have been subjected to some prescribed permutations. It has been observed that prescribing permutations is not possible with twiddling factors $W_{MN}^{i_2 I_1}$. Instead if twiddling factors are taken as $W_N^{i_2 I_1}$ the permutation properties exhibited by this transform are quite interesting. In that case Eq. (3.32) would become

$$A(I)=A(I_2M+I_1)=\sum_{i_2=0}^{N-1} W_N^{i_2 I_2} \quad W_N^{i_2 I_1} \sum_{i_1=0}^{M-1} a(i_1N+i_2)(-1)^{\sum_{t=0}^{s-1} i_1^t I_1^t \bmod 2} \qquad \ldots (3.35)$$

This would give the two-dimensional arrays $[H_{i_2}^{(I_1)}]$ and $[A(I)]$ for the example as

$$[H_{i_2}^{(I_1)}]=\begin{bmatrix} 2+j0 & 2+j0 & 1+j0 & 1+j0 & 2+j0 \\ 2+j0 & -0.618+j1.9022 & -0.809-j0.5878 & -0.809+j0.5878 & 0+j0 \\ 0+j0 & 0+j0 & -0.309-j0.9511 & 0.309-j0.9511 & 1.618-j1.1756 \\ 0+j0 & 0+j0 & -0.309+j0.9511 & 0.309+j0.9511 & 0+j0 \end{bmatrix}$$

$$
A(I) = [A(I_2, I_1)] = \begin{bmatrix}
8 + j0 & 1.618 + j0 & -0.618 + j0 & -0.618 + j0 & 1.618 + j0 \\
-0.236 + j1.9022 & 4.2362 + j1.1756 & 4.2362 - j1.1756 & -0.236 - j1.9022 & 2 + j0 \\
1.618 - j3.0778 & 1.618 + j3.0778 & -0.618 + j0.7266 & -2 + j0 & -0.618 - j0.7266 \\
0 + j1.9022 & 0 - j1.1756 & 0 + j0 & 0 + j1.1756 & 0 - j1.9022
\end{bmatrix} \quad \ldots (3.36)
$$

The two-dimensional array representing A(I) has a peculiar pattern. The elements along any row appear in conjugate pairs. In case N is odd one of the elements in each row would be real. In the zeroth row transform samples (1.618+j0) and (0.618+j0) occur twice. It is because of the particlar a(i) chosen. This is also the reason for all the elements in the zeroth row to be real except A(0) which would be always real.

3.42 MODULAR PERMUTATION OF COLUMNS

It is known that DFT exhibits modular permutation property. In the transform defined in Section (3.41) N-term rowwise DFT is taken. Let the N columns of a(i), i = 0,1,2,...,MN-1 written as a two-dimensional array of M rows and N columns and read row by row be subjected to modular permutation operator P(p,N) treating each of the N columns as an element. Then the resulting transform samples, $A_c(I)$, would be given by

$$A_c(I) = A(I_2'M+I_1')$$

$$= \sum_{i_2=0}^{N-1} W_N^{i_2I_2} W_N^{i_2I_1} \sum_{i_1=0}^{M-1} a(i_1N+((i_2p))_N)(-1)^{\sum_{t=0}^{s-1} i_1^t I_1^t \text{ modulo } 2} \quad \ldots (3.37)$$

$$I = 0,1,2,\ldots,MN-1$$

where I_1' and I_2' are the new values of indices I_1 and I_2 respectively.

and the permuted input signal sample seqnence, $a_c(i)$, would be given by

$$a_c(i)=a(i_1N+((pi_2))_N) \text{ , } i = 0,1,2,\ldots,MN-1 \quad \ldots (3.38)$$

$$A_c(I) = A(I_2'M+I_1')$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2I_2))_N} W_N^{((i_2I_1))_N} h_{((pi_2))_N}^{(I_1)}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2I_2))_N} W_N^{((I_1pi_2))_N} h_{((pi_2))_N}^{(I_1)} W_N^{((i_2I_1))_N-((I_1pi_2))_N}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2I_2))_N+((i_2I_1))_N-((I_1pi_2))_N} H_{((pi_2))_N}^{(I_1)}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((I_2'pi_2))_N} H_{((pi_2))_N}^{(I_1')} \text{ , say} \quad \ldots (3.39)$$

Then

$$((I_2'pi_2))_N = ((i_2I_2))_N + ((i_2I_1))_N - ((I_1pi_2))_N$$

If P(q,N) be the inverse modular permutation operator so that

$$((p\ q))_N = 1$$

then

$$((I_2' p i_2))_N = ((pq))_N ((i_2 I_2))_N + ((pq))_N ((i_2 I_1))_N - ((I_1 p i_2))_N$$
$$= ((pqi_2 I_2))_N + ((pqi_2 I_1))_N - ((I_1 p i_2))_N$$
$$= ((\ ((qI_2 + qI_1 - I_1))_N\ p i_2))_N$$

that is

$$I_2' = ((q(I_1+I_2) - I_1))_N \qquad \dots (3.40)$$

Also from Eq. (3.39) it is obvious that

$$I_1' = I_1$$

So if the columns of $a(i) = a\ (i_1N+i_2)$ are permuted so that the input signal sample sequence is given by $a_c(i) = a(i_1N+((pi_2))_N)$ then the transform samples $A(I) = A(I_2M+I_1)$ are given by
$A_c(I) = A(M((q(I_1+I_2) - I_1))_N+I_1)$

This result is very interesting and shows that a columnwise modular permutation on $a(i) = a(i_1N+i_2)$ first and then taking transform according to Eq. (3.35) is analogous to rearranging the elements of $A(I)=A(I_2M+I_1)$. This rearrangement changes the value of index I_2 to I_2' where I_2' is a function of I_1, I_2 and the parameter q of the inverse modular permutation operator P(q,N). Thus a permutation of index i_2 in $a(i_1N+i_2)$ to $((pi_2))_N$ results in the permutation of index I_2 in $A(I_2M+I_1)$ to $((q(I_1+I_2)-I_1))_N$. This has been illustrated in flow chart form in Fig. 3.2. The permutation law derived above has been verified with M = 4, N = 5 and M = 8, N = 3 and the results are listed below. The results tally with the permutation law derived theoretically.

M N p	$a(i_1N+((pi_2))_N)$					q	$A(M((q(I_1+I_2)-I_1))_N+I_1)$				
	a(0)	(al)	a(2)	a(3)	a(4)		A(0)	A(4)	A8)	A(12)	A(16)
4 5 2	a(5)	a(6)	a(7)	a(8)	a(9)	1	A(1)	A(5)	A(9)	A(13)	A(17)
	a(10)	a(11)	a(12)	a(13)	a(14)		A(2)	A(6)	A(10)	A(14)	A(18)
	a(15)	a(16)	a(17)	a(18)	a(19)		A(3)	A(7)	A(11)	A(15)	A(19)
	a(0)	a(2)	a(4)	a(1)	a(3)		A(0)	A(12)	A(4)	A(16)	A(8)
4 5 2	a(5)	a(7)	a(9)	a(6)	a(8)	3	A(9)	A(1)	A(13)	A(5)	A(17)
	a(10)	a(12)	a(14)	a(11)	a(13)		A(18)	A(10)	A(2)	A(14)	A(6)
	a(15)	a(17)	a(19)	a(16)	a(18)		A(7)	A(19)	A(11)	A(3)	A(15)
	a(0)	a(3)	a(1)	a(4)	a(2)		A(0)	A(8)	A(16)	A(4)	A(12)
4 5 3	a(5)	a(8)	a(6)	a(9)	a(7)	2	A(5)	A(13)	A(1)	A(9)	A(17)
	a(10)	a(13)	a(11)	a(14)	a(12)		A(10)	A(18)	A(6)	A(14)	A(2)
	a(15)	a(18)	a(16)	a(19)	a(17)		A(15)	A(3)	A(11)	A(19)	A(7)
	a(0)	a(4)	a(3)	a(2)	a(1)		A(0)	A(16)	A(12)	A(8)	A(4)
4 5 4	a(5)	a(9)	a(8)	a(7)	a(6)	4	A(13)	A(9)	A(5)	A(1)	A(17)
	a(10)	a(14)	a(13)	a(12)	a(11)		A(6)	A(2)	A(18)	A(14)	A(10)
	a(15)	a(19)	a(18)	a(17)	a(16)		A(19)	A(15)	A(11)	A(7)	A(3)

8 3 1	a(0)	a(1)	a(2)	1	A(0)	A(8)	A(16)
	a(3)	a(4)	a(5)		A(1)	A(9)	A(17)
	a(6)	a(7)	a(8)		A(2)	A(10)	A(18)
	a(9)	a(10)	a(11)		A(3)	A(11)	A(19)
	a(12)	a(13)	a(14)		A(4)	A(12)	A(20)
	a(15)	a(16)	a(17)		A(5)	A(13)	A(21)
	a(18)	a(19)	a(20)		A(6)	A(14)	A(22)
	a(21)	a(22)	a(23)		A(7)	A(15)	A(23)

8 3 2	a(0)	a(2)	a(1)	2	A(0)	A(16)	A(8)
	a(3)	a(5)	a(4)		A(9)	A(1)	A(17)
	a(6)	a(8)	a(7)		A(18)	A(10)	A(2)
	a(9)	a(11)	a(10)		A(3)	A(19)	A(11)
	a(12)	a(14)	a(13)		A(12)	A(4)	A(20)
	a(15)	a(17)	a(16)		A(21)	A(13)	A(5)
	a(18)	a(20)	a(19)		A(6)	A(22)	A(14)
	a(21)	a(23)	a(22)		A(15)	A(7)	A(23)

3.43 BIT-PLANE PERMUTATION OF ROWS

It is known that Hadamard transform of order $M=2^s$, s being an integer, exhibits bit-plane permutation property. In the transform defined in section (3.41) M-term columnwise H T is taken. Let the M rows of a(i), i=0,1,2,...,MN-1 written as a two-dimensional array of M rows and N columns and read row by row be subjected to bit-plane permutation operator P(a b ... k) treating each

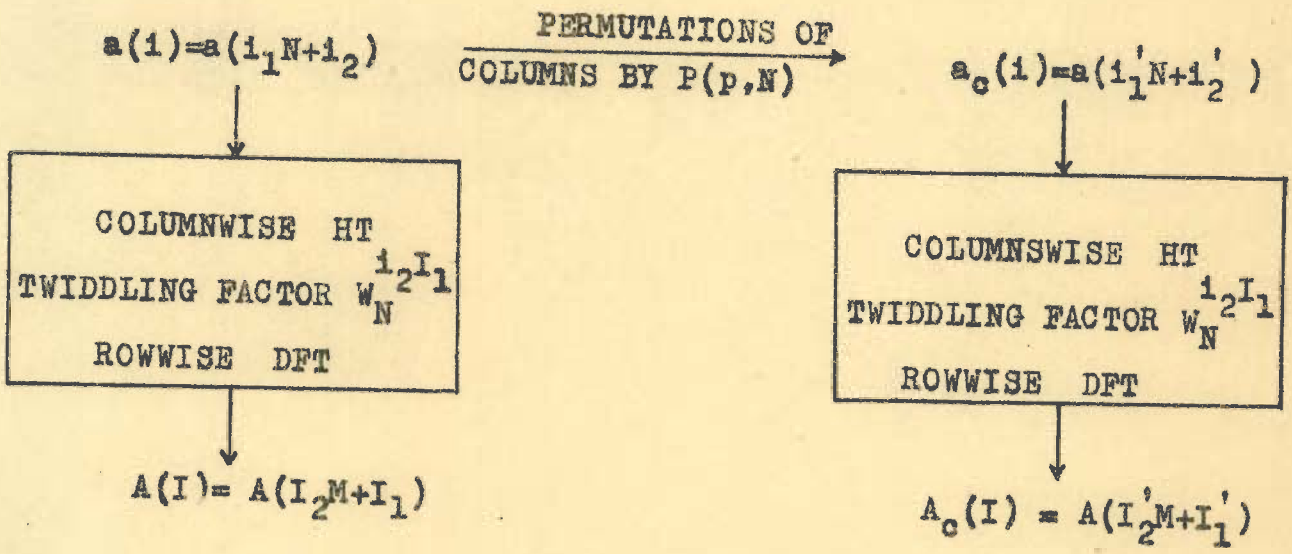


FIG: 3.2 : FLOW CHART OF TRANSFORM WITH COLUMNS PERMUTED of the M rows as an element. Let

$$(i_1)_{decimal} = (i_1^{s-1}\ i_1^{s-2}\ \ldots\ i_1^{t}\ \ldots\ i_1^{1}\ i_1^{0})_{binary}$$

and

$$P(a\ b\ \ldots k)(i_1^{s-1}\ i_1^{s-2}\ \ldots\ i_1^{t}\ \ldots\ i_1^{1}\ i_1^{0})_{binary}$$
$$= (i_1^{a}\ i_1^{b}\ \ldots .\ i_1^{k})_{binary} = (i_{1,P})_{decimal}, \text{ say}$$

Then the resulting transform samples, $A_r(I)$, would be given by

$$A_r(I) = A(I_2''M+I_1'')$$
$$= \sum_{i_2=0}^{N-1} W_N^{i_2I_2}\ W_N^{i_2I_1} \sum_{i_{1,P}=0}^{M-1} a(i_1PN+i_2)(-1)^{\sum_{t=0}^{s-1} i_{1,P}^{t} I_1^{t} \text{ modulo } 2} \quad \ldots (3.41)$$
$$I=0,1,2,\ldots,MN-1$$

where I_1'' and I_2'' are the new values of indices I_1 and I_2 respectively.

and the permuted input signal sample sequence, $a_r(i)$ would be given by

$$a_r(i) = a(i_{1,P}N+i_2)\ , \quad i = 0,1,2,\ldots, MN-1 \quad \ldots (3.42)$$

$$A_r(I) = A(I_2''M+I_1'')$$
$$= \sum_{i_2=0}^{N-1} W_N^{((i_2I_2))_N}\ W_N^{((i_2I_1))_N} \sum_{i_1=0}^{M-1} a(i_1N+i_2)(-1)^{\sum_{t=0}^{s-1} i_1^{t} I_{1,P}^{t} \text{ modulo } 2}$$

where $(I_{1,P})_{decimal} = P(a\ b \ldots k)(I_1^{s-1}\ I_1^{s-2} \ldots I_1^{t} \ldots I_1^{1}\ I_1^{0})_{binary}$

$$= (I_1^a\ I_1^b\ \ldots\ I_1^k)_{binary} \qquad \ldots (3.43)$$

and $(I_1)_{decimal} = (I_1^{s-1}\ I_1^{s-2}\ \ldots I_1^t\ \ldots\ I_1^1\ I_1^0)_{binary}$

$$A_r(I) = A(I_2''M+I_1'')$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2 I_2))_N}\ W_N^{((i_2 I_1))_N}\ h_{i_2}^{(I_{1,P})}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2 I_2))_N}\ W_N^{((i_2 I_1 - i_2 I_{1,P}))_N}\ W_N^{((i_2 I_{1,P}))_N} h_{i_2}^{(I_{1,P})}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2((I_2+I_1-I_{1,P}))_N))_N}\ H_{i_2}^{(I_{1,P})}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2 I_2''))_N}\ H_{i_2}^{(I_1'')}\ , \text{ say} \qquad \ldots (3.44)$$

This would give expressions for I_1'' and I_2'' as

$$I_1'' = I_{1,P}''$$

and $\qquad I_2 = ((I_1 + I_2 - I_{1,P}))_N$

So if the rows of $a(i) = a(i_1N+i_2)$ are permuted so that the input signal sample sequence is given by $a_r(i) = a(i_{1,P}N+i_2)$ then the transform samples $A(I)=A(I_2M+I_1)$ are given by $A_r(I)=A(M((I_1+I_2-I_{1,P}))_N+I_{1,P})$

This result shows that a rowwise bit-plane permutation on $a(i) = a(i_1N+i_2)$ first and then taking transform according to Eq. (3.35) is analogous to rearranging the elements of $A(I) = A(I_2M+I_1)$. This rearrangement changes the values of indices I_1 and I_2 to I_1'' and I_2'' respectively. While I_1 depends upon I_1 and bit-plane permutation operator P(a b ... k) the other new index I_2 is a function of I_1, I_2 and bit-plane permutation operator P(a b...k) which is known to be self-reciprocal. Thus a permutation of index i_1 in $a(i_1N+i_2)$ to $i_{1,P}$ results in the permutation of both the indices I_1 and I_2 in $A(I_2M+I_1)$. This has been illustrated in flow chart form in Fig. 3.3. The permutation law derived above has been verified with M = 4, N = 5 and M = 8, N=3 and the results are listed below. These results tally with the permutation law derived theoretically. It may be pointed out that the elements in such rows of A(I) for which $I_1 = I_{1,P}$ are unchanged in position.

M N P(ab..k)	$a(i_{1,P}N+i_2)$					$A(M((I_1+I_2-I_{1,P}))_N+I_{1,P})$				
	a(0)	a(1)	a(2)	a(3)	a(4)	A(0)	A(4)	A(8)	A(12)	A(16)
4 5 P(0 1)	a(10)	a(11)	a(12)	a(13)	a(14)	A(18)	A(2)	A(6)	A(10)	A(14)
	a(5)	a(6)	a(7)	a(8)	a(9)	A(5)	A(9)	A(13)	A(17)	A(1)
	a(15)	a(16)	a(17)	a(18)	a(19)	A(3)	A(7)	A(11)	A(15)	A(19)

8 3 P(2 0 1)

a(0)	a(1)	a(2)	A(0)	A(8)	A(16)
a(6)	a(7)	a(8)	A(18)	A(2)	A(10)
a(3)	a(4)	a(5)	A(9)	A(17)	A(1)
a(9)	a(10)	a(11)	A(3)	A(11)	A(19)
a(12)	a(13)	a(14)	A(4)	A(12)	A(20)
a(18)	a(19)	a(20)	A(22)	A(6)	A(14)
a(15)	a(16)	a(17)	A(13)	A(21)	A(5)
a(21)	a(22)	a(23)	A(7)	A(15)	A(23)

8 3 P(1 2 0)

a(0)	a(1)	a(2)	A(0)	A(8)	A(16)
a(3)	a(4)	a(5)	A(1)	A(9)	A(17)
a(12)	a(13)	a(14)	A(12)	A(20)	A(4)
a(15)	a(16)	a(17)	A(13)	A(21)	A(5)
a(6)	a(7)	a(8)	A(18)	A(2)	A(10)
a(9)	a(10)	a(11)	A(19)	A(3)	A(11)
a(18)	a(19)	a(20)	A(6)	A(14)	A(22)
a(21)	a(22)	a(23)	A(7)	A(15)	A(23)

8 3 P(1 0 2)

a(0)	a(1)	a(2)	A(0)	A(8)	A(16)
a(6)	a(7)	a(8)	A(18)	A(2)	A(10)
a(12)	a(13)	a(14)	A(12)	A(20)	A(4)
a(18)	a(19)	a(20)	A(6)	A(14)	A(22)
a(3)	a(4)	a(5)	A(1)	A(9)	A(17)
a(9)	a(10)	a(11)	A(19)	A(3)	A(11)
a(15)	a(16)	a(17)	A(13)	A(21)	A(5)
a(21)	a(22)	a(23)	A(7)	A(15)	A(23)

8 3 (P(0 2 1)	a(0)	a(1)	a(2)	A(0)	A(8)	A(16)
	a(12)	a(13)	a(14)	A(4)	A(12)	A(20)
	a(3)	a(4)	a(5)	A(9)	A(17)	A(1)
	a(15)	a(16)	a(17)	A(13)	A(21)	A(5)
	a(6)	a(7)	a(8)	A(18)	A(2)	A(10)
	a(18)	a(19)	a(20)	A(22)	A(6)	A(14)
	a(9)	a(10)	a(11)	A(3)	A(11)	A(19)
	a(21)	a(22)	a(23)	A(7)	A(15)	A(23)

The values of $i_{1,p}$ and $I_{1,p}$ for given i_1, I_1 and bit-plane permutation operators P(a b .. k) are listed in Table 3.3 for M = 4 and M = 8.

3.44 MODULAR PERMUTATION OF COLUMNS AND BIT- PLANE PERMUTATION OF ROWS

In section (3.42) and (3.43) the effect of modular permutation on columns and bit-permutation on rows of a(i) written as a two-dimensional array of M rows and N columns and read row by row have been investigated. Let the N columns of a(i) be subjected to modular permutation operator P(p,N) treating each of the N columns as an element , and M rows be subjected to bit-plane permutation operator P(a b.... k) treating each of the M rows as an element. Then the resulting transform samples, $A_{rc}(I)$, would be given by

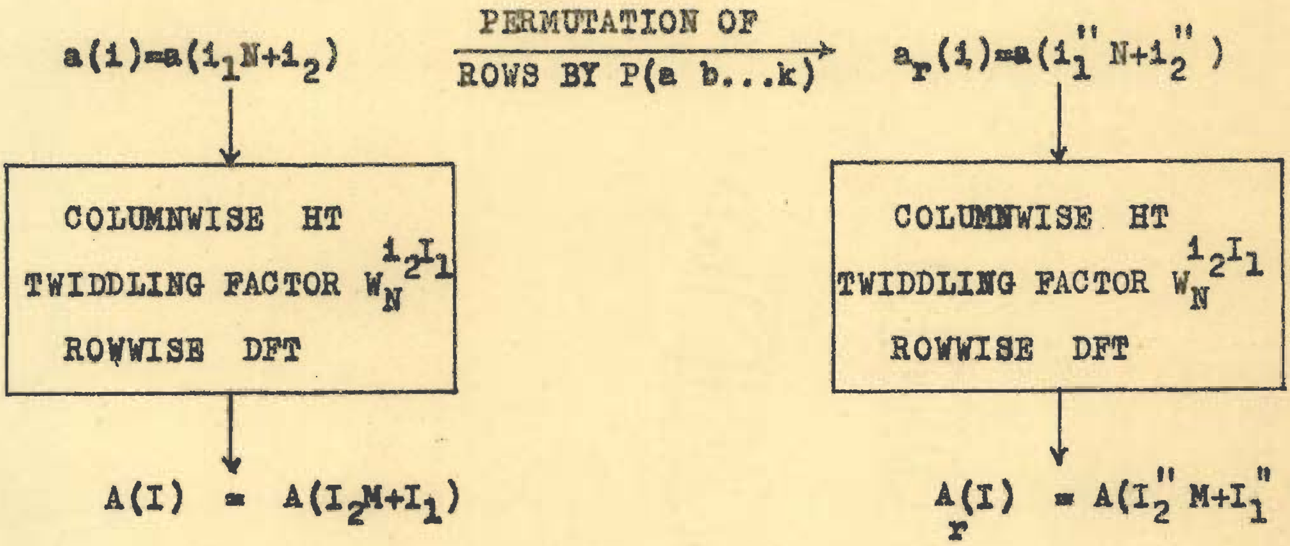


FIG. 3.3 : FLOW CHART OF TRANSFORM WITH ROWS PERMUTED

Table-3.3 : Computation of $i_{1,P}$ and $I_{1,P}$

M=4 $i_{1,P}=P(a\ b\ \ldots k)\ i_1$ and $I_{1,P}=P(a\ b \ldots k) I_1$

i_1, I_1 dec.	binary		P(1 0) $i_{1,P}$, $I_{1,P}$ binary		dec.	P(0 1) $i_{1,P}$, $I_{1,P}$ binary		dec.
0	0	0	0	0	0	0	0	0
1	0	1	0	1	1	1	0	2
2	1	0	1	0	2	0	1	1
3	1	1	1	1	3	1	1	3

M=8 $i_{1,P}=P(a\ b \ldots k)\ i_1$ and $I_{1,P}=P(a\ b \ldots k)\ I_1$

i_1, I_1 dec.	binary	P(2 1 0) $i_{1,P}$, $I_{1,P}$ binary	dec.	P(2 0 1) $i_{1,P}$, $I_{1,P}$ bin.	dec.	P(1 2 0) $i_{1,P}$, $I_{1,P}$ bin.	dec.	P(1 0 2) $i_{1,P}$, $I_{1,P}$ bin.	dec.	P(0 2 1) $i_{1,P}$, $I_{1,P}$ bin.	dec.	P(0 1 2) $i_{1,P}$, $I_{1,P}$ bin.	dec.
0	0 0 0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0
1	0 0 1	0 0 1	1	0 1 0	2	0 0 1	1	0 1 0	2	1 0 0	4	1 0 0	4
2	0 1 0	0 1 0	2	0 0 1	1	1 0 0	4	1 0 0	4	0 0 1	1	0 1 0	2
3	0 1 1	0 1 1	3	0 1 1	3	1 0 1	5	1 1 0	6	1 0 1	5	1 1 0	6
4	1 0 0	1 0 0	4	1 0 0	4	0 1 0	2	0 0 1	1	0 1 0	2	0 0 1	1
5	1 0 1	1 0 1	5	1 1 0	6	0 1 1	3	0 1 1	3	1 1 0	6	1 0 1	5
6	1 1 0	1 1 0	6	1 0 1	5	1 1 0	6	1 0 1	5	0 1 1	3	0 1 1	3
7	1 1 1	1 1 1	7	1 1 1	7	1 1 1	7	1 1 1	7	1 1 1	7	1 1 1	7

$$A_{rc}(I) = A(I_2'''M + I_1''')$$

$$= \sum_{i_2=0}^{N-1} W_N^{i_2 I_2} W_N^{i_2 I_1} \sum_{i_{1,P}}^{M-1} a(i_{1,P}N + ((pi_2))_N) \, (-1)^{\sum_{t=0}^{s-1} i_{1,P}^{t} I_1^{t} \text{ modulo } 2} \quad \ldots (3.45)$$

$$I = 0,1,2,\ldots,MN-1$$

where I_1''' and I_2''' are the new values of indices I_1 and I_2 respectively.

and the permuted input signal sample sequence $a_{rc}(i)$ would be given by

$$a_{rc}(i) = a(i_{1,P}N + ((pi_2))_N) \;,\; i = 0,1,2,\ldots,MN-1 \quad \ldots (3.46)$$

$$A_{rc}(I) = A(I_2'''M + I_1''')$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2 I_2))_N} W_N^{((i_2 I_1))_N} \sum_{i_1=0}^{M-1} a(i_1 N + ((pi_2))_N) \, (-1)^{\sum_{t=0}^{s-1} i_1^t I_{1,P}^t \text{ modulo } 2}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2 I_2))_N} W_N^{((i_2 I_1))_N} h_{((pi_2))_N}^{(I_{1,P})}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2 I_2))_N} W_N^{((i_2 I_1))_N - ((pi_2 I_{1,P}))_N} W_N^{((pi_2 I_{1,P}))_N} h_{((P_2))_N}^{(I_{1,P})}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((i_2I_2))_N+((i_2I_1))_N-((pi_2I_{1,P}))_N} H_{((pi_2))_N}^{((I_{1,P})}$$

$$= \sum_{i_2=0}^{N-1} W_N^{((I_2''' pi_2))_N} H_{((pi_2))_N}^{(I_1''')} \text{, say} \qquad \ldots (3.47)$$

So $I_1''' = I_{1,P}$ and for obtaining expressions for I_2''' use

$$((I_2''' pi_2))_N = ((i_2I_2))_N+((i_2I_1))_N-((pi_2 I_{1,P}))_N$$

to get

$$I_2''' = ((q(I_1+I_2) - I_{1,P}))_N$$

So if rows and columns of $a(i) = a(i_1N+i_2)$ are permuted so that the input signal sample sequence is given by $a_{rc}(i) = a(i_{1,P}N+((pi_2))_N)$ then the transform samples $A(I) = A(I_2M+I_1)$ are given by $A_{rc}(I) = A(M((q(I_1+I_2) -I_{1,P}))_N + I_{1,P})$.

This result shows that rowwise bit-plane permutation and columnwise modular permutation on $a(i)=a(i_1N+i_2)$ first and then taking transform according to Eq. (3.35) is equivalent to rearranging the elements of $A(I) = A(MI_2+I_1)$. This rearrangement changes the values of indices I_1 and I_2 to I_1''' and I_2''' respectively. While I_1''' depends upon I_1 and bit-plane permutation operator

P(a b...k) the other new index I_2''' is a function of I_1, I_2, bit-plane permutation operator P(a b...k) and inverse modular permutation operator P(q,N). This has been illustrated in flow chart form in Fig. 3.4. The permutation law derived above has been verified with M=4, N=5 and M=8, N=3 and the results are listed below. These results tally with the permutation law derived theoretically.

M N p P(a b..k)	$a(i_{1,p}N+((pi_2))_N)$	q	$A(M((q(I_1+I_2)-I_{1,2}))_N+I_{1,p})$
4 5 2 P(0 1)	a(0) a(2) a(4) a(1) a(3) a(10)a(12)a(14)a(11)a(13) a(5) a(7) a(9) a(6) a(8) a(15)a(17)a(19)a(16)a(18)	3	A(0) A(12) A(4) A(16) A(8) A(6) A(18) A(10)A(2) A(14) A(1) A(13) A(5) A(17)A(9) A(7) A(19) A(11)A(3) A(15)
4 5 3 P(0 1)	a(0) a(3) a(1) a(4) a(2) a(10)a(13)a(11)a(14)a(12) a(5) a(8) a(6) a(9) a(7) a(15)a(18)a(16)a(19)a(17)	2	A(0) A(8) A(16)A(4) A(12) A(2) A(10) A(18)A(6) A(14) A(13)A(1) A(9) A(17)A(5) A(15)A(3) A(11)A(19)A(7)
4 5 4 P(0 1)	a(0) a(4) a(3) a(2) a(1) a(10)a(14)a(13)a(12)a(11) a(5) a(9) a(8) a(7) a(6) a(15)a(19)a(18)a(17)a(16)	4	A(0) A(16) A(12)A(8) A(4) A(10)A(6) A(2) A(18)A(14) A(9) A(5) A(1) A(17)A(13) A(19)A(15) A(11)A(7) A(3)

	a(0)	a(2)	a(1)		A(0)	A(16)	A(8)
	a(6)	a(8)	a(7)		A(2)	A(18)	A(10)
	a(3)	a(5)	a(4)		A(1)	A(17)	A(9)
8 3 2 P(2 0 1)	a(9)	a(11)	a(10)	2	A(3)	A(19)	A(11)
	a(12)	a(14)	a(13)		A(12)	A(4)	A(20)
	a(18)	a(20)	a(19)		A(14)	A(6)	A(22)
	a(15)	a(17)	a(16)		A(13)	A(5)	A(21)
	a(21)	a(23)	a(22)		A(15)	A(7)	A(23)
	a(0)	a(2)	a(1)		A(0)	A(16)	A(8)
	a(3)	a(5)	a(4)		A(9)	A(1)	A(17)
	a(12)	a(14)	a(13)		A(4)	A(20)	A(12)
8 3 2 P(1 2 0)	a(15)	a(17)	a(16)	2	A(13)	A(5)	A(21)
	a(6)	a(8)	a(7)		A(2)	A(18)	A(10)
	a(9)	a(11)	a(10)		A(11)	A(3)	A(19)
	a(18)	a(20)	a(19)		A(6)	A(22)	A(14)
	a(21)	a(23)	a(22)		A(15)	A(7)	A(23)
	a(0)	a(2)	a(1)		A(0)	A(16)	A(8)
	a(6)	a(8)	a(7)		A(2)	A(18)	A(10)
	a(12)	a(14)	a(13)		A(4)	A(20)	A(12)
8 3 2 P(1 0 2)	a(18)	a(20)	a(19)	2	A(6)	A(22)	A(14)
	a(3)	a(5)	a(4)		A(9)	A(1)	A(17)
	a(9)	a(11)	a(10)		A(11)	A(3)	A(19)
	a(15)	a(17)	a(16)		A(13)	A(5)	A(21)
	a(21)	a(23)	a(22)		A(15)	A(7)	A(23)

8 3 2 P(0 2 1)	a(0)	a(2)	a(1)	2	A(0)	A(16)	A(8)
	a(12)	a(14)	a(13)		A(12)	A(4)	A(20)
	a(3)	a(5)	a(4)		A(1)	A(17)	A(9)
	a(15)	a(17)	a(16)		A(13)	A(5)	A(21)
	a(6)	a(8)	a(7)		A(2)	A(18)	A(10)
	a(18)	a(20)	a(19)		A(14)	A(6)	A(22)
	a(9)	a(11)	a(10)		A(3)	A(19)	A(11)
	a(21)	a(23)	a(22)		A(15)	A(7)	A(23)

8 3 2 P(0 1 2)	a(0)	a(2)	a(1)	2	A(0)	A(16)	A(8)
	a(12)	a(14)	a(13)		A(12)	A(4)	A(20)
	a(6)	a(8)	a(7)		A(18)	A(10)	A(2)
	a(18)	a(20)	a(19)		A(6)	A(22)	A(14)
	a(3)	a(5)	a(4)		A(9)	A(1)	A(17)
	a(15)	a(17)	a(16)		A(21)	A(13)	A(5)
	a(9)	a(11)	a(10)		A(3)	A(19)	A(11)
	a(21)	a(23)	a(22)		A(15)	A(7)	A(23)

The technique developed in this chapter can be used to obtain a class of transforms with prescribed permutation properties by choosing suitable transforms kernels in place of HT and DFT. The twiddling factors in each such resulting transform would have to be defined keeping in view the properties of the component transforms used.

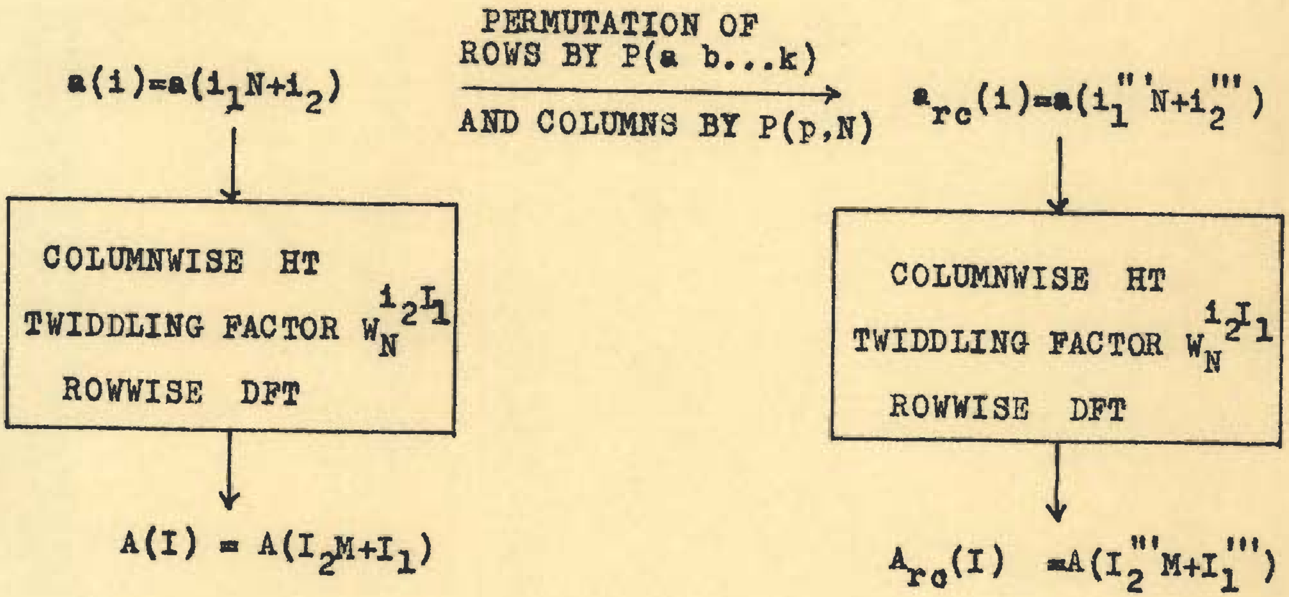


FIG. 3.4 : FLOW CHART OF TRANSFORM WITH ROWS AND COLUMNS PERMUTED

CHAPTER - 4

INTER RELATIONS AMONG VARIOUS TRANSFORMS

The ordinary multiplication of two matrices is defined only when the two component matrices are conformable for multiplication. This matrix multiplication commutes only under special conditions. It is well known that Kronecker product of two matrices A and B is always defined irrespective of the dimensions of the component matrices. It is also known that Kronecker product of A and B can be obtained from Kronecker product of B and A by pre and post multiplication by sparse permutation matrices of suitable dimensions and vice-versa. Another matrix product known as Chinese product can be defined when the numbers of rows (columns) of the two component matrices are coprimes. This matrix product always commutes. Analytical expressions have been developed for obtaining Chinese product from Kronecker product and vice-versa. A combination of Kronecker product and Chinese product concepts has been proposed to define Chinese-Kronecker product and Kronecker-Chinese product. Analytical expressions have also been developed to obtain one from another by pre and post multiplication with suitable sparse permutation matrices. The concept of the special matrix products has been applied to linear systems. The advantage of the work reported is that the output corresponding to any particular input can be deduced from outputs to simpler inputs in terms of which that particular input can be synthesized.

4.1 SPECIAL MATRIX PRODUCTS

If A and B are matrices of dimensions $M_1 \times N_1$ and $M_2 \times N_2$ given as

$$A = \begin{bmatrix} a(0,0) & a(0,1) & a(0,2) & \cdots & a(0,N_1-2) & a(0,N_1-1) \\ a(1,0) & a(1,1) & a(1,2) & \cdots & a(1,N_1-2) & a(1,N_1-1) \\ a(2,0) & a(2,1) & a(2,2) & \cdots & a(2,N_1-2) & a(2,N_1-1) \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ a(M_1-2,0) & a(M_1-2,1) & a(M_1-2,2) & \cdots & a(M_1-2,N_1-2) & a(M_1-2,N_1-1) \\ a(M_1-1,0) & a(M_1-1,1) & a(M_1-1,2) & \cdots & a(M_1-1,N_1-2) & a(M_1-1,N_1-1) \end{bmatrix} \quad \ldots (4.1)$$

Alternatively

$$A = [a(i_1,j_1)] \qquad \begin{matrix} i_1 = 0,1,2,\ldots,M_1-1 \\ j_1 = 0,1,2,\ldots,N_1-1 \end{matrix} \quad \ldots (4.2)$$

$$B = \begin{bmatrix} b(0,0) & b(0,1) & b(0,2) & \cdots & b(0,N_2-2) & b(0,N_2-1) \\ b(1,0) & b(1,1) & b(1,2) & \cdots & b(1,N_2-2) & b(1,N_2-1) \\ b(2,0) & b(2,1) & b(2,2) & \cdots & b(2,N_2-2) & b(2,N_2-1) \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ b(M_2-2,0) & b(M_2-2,1) & b(M_2-2,2) & \cdots & b(M_2-2,N_2-2) & b(M_2-2,N_2-1) \\ b(M_2-1,0) & b(M_2-1,1) & b(M_2-1,2) & \cdots & b(M_2-1,N_2-2) & b(M_2-1,N_2-1) \end{bmatrix} \quad \ldots (4.3)$$

Alternatively

$$B = [b(i_2,j_2)] \quad \begin{matrix} i_2 = 0,1,2,\ldots,M_2-1 \\ j_2 = 0,1,2,\ldots,N_2-1 \end{matrix} \quad \ldots (4.4)$$

Then C_k the Kronecker product or direct product of matrices A and B, denoted by $A \otimes_k B$ is a matrix of dimension $M_1M_2 \times N_1N_2$ and is given as

$$C_k = A\otimes_k B = \begin{bmatrix} a(0,0)B & a(0,1)B & a(0,2)B & \ldots & a(0,N_1-2)B & a(0,N_1-1)B \\ a(1,0)B & a(1,1)B & a(1,2)B & \ldots & a(1,N_1-2)B & a(1,N_1-1)B \\ a(2,0)B & a(2,1)B & a(2,2)B & \ldots & a(2,N_1-2)B & a(2,N_1-1)B \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a(M_1-2,0)B & a(M_1-2,1)B & a(M_1-2,2)B & \ldots & a(M_1-2,N_1-2)B & a(M_1-1,N_1-1)B \\ a(M_1-1,0)B & a(M_1-1,1)B & a(M_1-1,2)B & \ldots & a(M_1-1,N_1-2)B & a(M_1-1,N_1-1)B \end{bmatrix} \quad \ldots (4.5)$$

where $C_k = [c_k(i,j)]$ $\quad \begin{matrix} i=0,1,2,\ldots,M_1M_2-1 \\ j=0,1,2,\ldots,N_1N_2-1 \end{matrix} \quad \ldots (4.6)$

$$c_k(i,j)=a(i_1,j_1)b(i_2,j_2)$$

and

$$\begin{aligned} i &= i_2 + M_2\, i_1 \\ j &= j_2 + N_2\, j_i \end{aligned} \quad \ldots (4.7)$$

Some of the important properties of Kronecker product of are :

1) $(A \otimes_k B) \otimes_k C = A \otimes_k (B \otimes_k C)$

2) $(A \otimes_k B)^T \quad = A^T \otimes_k B^T$ $\quad \ldots (4.8)$

3) $(A \otimes_k B)^{-1} \quad = A^{-1} \otimes_k B^{-1}$

4) $(A+C) \otimes_k (B+D) = A \otimes_k B + A \otimes_k D + C \otimes_k B + C \otimes_k D$

5) $(A \otimes_k B)(C \otimes_k D) = A C \otimes_k B D$

6) $\alpha A \otimes_k \beta B \quad = \alpha \beta (A \otimes_k B)$

7) $(A_o \otimes_k A_1 \otimes_k \ldots \otimes_k A_{N-1})(B_o \otimes_k B_1 \otimes_k \ldots \otimes_k B_{N-1})$

$$= A_oB_o \otimes_k A_1B_1 \otimes_k \ldots \otimes_k A_{N-1} B_{N-1}$$

Let $\underset{(q)}{e_k}$ be a q-dimensional vector which is one in the k^{th} and zero elsewhere. This is termed as unit vector. Also let

$$E_{ik}^{(pxq)} \doteq \underset{(p)}{e_i} \; \underset{(q)}{e_k^T} \qquad \ldots (4.9)$$

be termed as some elementary matrix of dimension p x q with one in the location (i,k) and zero elsewhere. Brewer [12] has defined a permutation matrix U_{pxq} as

$$U_{pxq} = \sum_{i=0}^{p-1} \sum_{k=0}^{q-1} E_{ik}^{(pxq)} \otimes_k E_{ki}^{(qxp)} \qquad \ldots (4.10)$$

and which is of dimension pq x pq with precisely a single one in each row and each column, rest of the elements being zero. He has defined another matrix as

$$\bar{U}_{pxq} \doteq \sum_{i=0}^{p-1} \sum_{k=0}^{q-1} E_{ik}^{(pxq)} \otimes_k E_{ik}^{(pxq)} \qquad \ldots (4.11)$$

and which is of dimension p^2xq^2 with precisely a single one in each row and each column, rest of the elements being zero. Some of the relationships which hold for these permutation matrices are given by Brewer [12] and are listed below:

1. $e_i^T \; e_k = \delta_{ik}$ where δ_{ik} is the Kronecker Delta
   (p) (p)

2) $E_{ik}^{(pxq)} \; E_{mn}^{(qxr)} = \delta_{km} \; E_{in}^{(pxr)}$

3) $A = \sum_{i=0}^{p-1} \sum_{k=0}^{q-1} A_{ik} \; E_{ik}^{(pxq)}$ ... (4.12)

4) $E_{ik}^{(sxp)} \; A \; E_{mn}^{(qxr)} = A_{km} \; E_{in}^{(sxr)}$

5) $(U_{pxq})^T = U_{qxp}$

6) $U_{pxq}^{-1} = U_{qxp}$

7) $(E_{ik}^{(pxq)})^T = E_{ki}^{(qxp)}$

8) $U_{px1} = U_{1xp} = I_p$

9) $U_{nxn} = U_{nxn}^T = U_{nxn}^{-1}$

10) $U_{nxn} \; \bar{U}_{nxn} = \bar{U}_{nxn}$

The concepts of these permutation matrices can be used to obtain relationship between $A \otimes_k B$ and $B \otimes_k A$.

Based on Chinese remainder theorem [11] one can define multiplications of one-dimensional and two-dimensional arrays provided the corresponding dimensions of the arrays are coprimes. Moharir [40] has defined Chinese product of sequences.

If A and B are one-dimensional sequences of length M_1 and M_2 respectively, M_1 and M_2 being coprimes, and given as

$$A = [a(0)\ a(1)\ a(2)\ \ldots\ a(M_1-2)\ a(M_1-1)] \quad \ldots (4.13)$$

and $$B = [b(0)\ b(1)\ b(2)\ \ldots\ b(M_2-2)\ b(M_2-1)]$$

then the sequence

$$C = [c(0)\ c(1)\ c(2)\ \ldots\ c(M-2)\ \ c(M-1)\ ] \quad \ldots (4.14)$$

is defined as the Chinese product of sequences A and B if

$$M = M_1\, M_2 \quad \ldots (4.15)$$

and $$c(i) = a(i_1)\, b(i_2) \quad \ldots (4.16)$$

where

$$\begin{aligned} i &\equiv i_1 \text{ modulo } M_1 \\ &\equiv i_2 \text{ modulo } M_2 \end{aligned} \quad \ldots (4.17)$$

i.e., i is congruent to i_1 modulo M_1 and also congruent to i_2 modulo M_2. For any given i, i_1 and i_2 can be uniquely determined and vice-versa. If A and B be two-dimensional arrays of Eqs. (4.2) and (4.4) and where M_1, M_2 are coprimes and N_1, N_2 are coprimes then the two-dimensional array

$$C_c = [c_c\ (i,j)] \qquad \begin{matrix} i = 0,1,2,\ldots,\ M-1 \\ j = 0,1,2,\ldots,\ N-1 \end{matrix} \quad \ldots (4.18)$$

is defined as the Chinese product of matrices A and B, $C_c = A \otimes_c B$, if

$$M = M_1 \, M_2$$
$$N = N_1 \, N_2 \qquad \ldots (4.19)$$

and

$$c_c(i,j) = a(i_1,j_1) \, b(i_2,j_2) \qquad \ldots (4.20)$$

$$\begin{aligned} i &\equiv i_1 \text{ modulo } M_1 \\ &\equiv i_2 \text{ modulo } M_2 \\ j &\equiv j_1 \text{ modulo } N_1 \\ &\equiv j_2 \text{ modulo } N_2 \end{aligned} \qquad \ldots (4.21)$$

Thus the dimension of the matrix resulting from Chienese product, if defined, of two matrices is the same as of one obtained by Kronecker product of the same matrices. It is clear from the definitions of Chinese product and Kronecker product that the array obtained is case of former is a rowwise and columnwise permutation of the array obtained in case of latter. If A and B are orthonormal matrices then $C_k = A \otimes_k B$ is known to be an orthonormal matrix. Further rowwise and/or columnwise permutation of an orthonormal matrix results in an orthonormal matrix. Hence the matrix obtained by Chinese product of orthonormal matrices, $C_c = A \otimes_c B$, would be an orthonormal matrix.

The concepts of Kronecker product and Chinese product can be exploited to define two more matrix products, viz. Kronecker-Chinese product and Chinese-Kronecker product of matrices.

If A and B be two-dimensional arrays of Eqs. (4.2) and (4.4) and where N_1, N_2 are coprimes then the two-dimensional array

$$C_{kc} = [c_{kc}(i,j)] \qquad \begin{array}{l} i = 0,1,2,\ldots,M-1 \\ j = 0,1,2,\ldots,N-1 \end{array} \quad \ldots (4.22)$$

is defined as the Kronecker-Chinese product of matrices A and B, $C_{kc} = A \otimes_{kc} B$, if

$$\begin{aligned} M &= M_1 M_2 \\ N &= N_1 N_2 \end{aligned} \quad \ldots (4.23)$$

and

$$c_{kc}(i,j) = a(i_1,j_1)\, b(i_2,j_2) \quad \ldots (4.24)$$

where

$$\begin{aligned} i &= i_2 + M_2 i_1 \\ j &\equiv j_1 \text{ modulo } N_1 \\ &\equiv j_2 \text{ modulo } N_2 \end{aligned} \quad \ldots (4.25)$$

If A and B be two-dimensional arrays of Eqs. (4.2) and (4.4) and where N_1, M_2 and coprimes then the two-dimensional array

$$C_{ck} = [c_{ck}(i,j)] \qquad \begin{array}{l} i = 0,1,2,\ldots,M-1 \\ j = 0,1,2,\ldots,N-1 \end{array} \quad \ldots (4.26)$$

is defined as the Chinese-Kronecker product of matrices

A and B, $C_{ck} = A \otimes_{ck} B$, if

$$M = M_1 M_2$$
$$N = N_1 N_2 \quad \ldots (4.27)$$

and

$$c_{ck}(i,j) = a(i_1,j_1)\; b\,(i_2,j_2) \quad \ldots (4.28)$$

where

$$\begin{aligned} i &\equiv i_1 \text{ modulo } M_1 \\ &\equiv i_2 \text{ modulo } M_2 \\ j &= j_2 + N_2\, j_1 \end{aligned} \quad \ldots (4.29)$$

The matrix obtained by Kronecker-Chinese product of two matrices A and B can be thought of as a columnwise permuted version of the matrix obtained by Kronecker product of the matrices A and B. Similarly the matrix obtained by Chinese-Kronecker product of two matrices A and B can be thought of as a rowwise permuted version of the matrix obtained by Kronecker product of the matrices A and B. It is known that if the two matrices A and B are orthonormal then the matrix obtained by Kronecker product of these orthonormal matrices is itself an orthonormal matrix. Since matrices obtained by Kronecker-Chinese and Chinese-Kronecker product of component matrices are the columnwise and rowwise permuted versions of the matrix obtained by Kronecker product of the same component matrices and that rowwise and/or columnwise permutations on an orthonormal matrix do not change its orthonormality hence the matrices obtained by Kronecker-Chinese product and Chinese-Kronecker product of orthonormal matrices A and B would always be orthonormal matrices.

4.2 RELATION BETWEEN CHINESE AND KRONECKER PRODUCTS

Let C_c and C_k be the matrices obtained by Chinese product and Kronecker product respectively of two matrices A and B of dimensions $M_1 \times N_1$ and $M_2 \times N_2$, M_1 , M_2 being coprimes and N_1, N_2 being coprimes. It has been stated that the matrices C_c and C_k are rowwise and columnwise permuted version of each other. In what follows expressions would be derived **for** permutation matrices P_1 and P_2 of suitable dimensions which are defined by

$$C_c = P_1 C_k P_2 \qquad \ldots (4.30)$$

$$C_k = P_1^{-1} C_c P_2^{-1} \qquad \ldots (4.31)$$

Since the matrices P_1 and P_2 are to effect rowwise and columnwise permutation respectively these matrices would have only a single one in each row and each column, rest of the elements being zero. Further, P_1 and P_2 would be square matrices of orders $M_1 M_2$ and $N_1 N_2$ respectively. A change in dummy variables in Eqs. (4.6) and (4.7) would give

$$c_k(i,j) = a(i_1', j_1') \, b(i_2', j_2') \qquad \ldots (4.32)$$

and

$$i = i_2' + M_2 i_1'$$
$$j = j_2' + N_2 j_1' \quad \ldots (4.33)$$

From Eqs (4.21) one can write

$$i_1 = i - M_1 [i/M_1]$$
$$i_2 = i - M_2 [i/M_2]$$
$$j_1 = j - N_1 [j/N_1] \quad \ldots (4.34)$$
$$j_2 = j - N_2 [j/N_2]$$

Similarly from Eq. (4.33)

$$i_1' = [i/M_2]$$
$$i_2' = i - M[i/M_2]$$
$$j_1' = [j/N_2] \quad \ldots (4.35)$$
$$j_2' = j - N_2 [j/N_2]$$

where [x/y] stands for integer part of (x/y). From Eqs. (4.34) and (4.35) it is clear that

$$i_2 = i_2' \text{ for all } i$$
$$j_2 = j_2' \text{ for all } j \quad \ldots (4.36)$$

and hence

$$b(i_2, j_2) = b(i_2', j_2') \text{ for all } i, j \quad \ldots (4.37)$$

This would imply that $b(i_2, j_2)$ part of $c_c(i,j)$ is element by element equal to $b(i_2', j_2')$ part of $c_k(i,j)$. But the corresponding $a(i_1, j_1)$ of $c_c(i,j)$ is not necessarily equal to the corresponding $a(i_1', j_1')$ of $c_k(i,j)$. So the permutation matrices P_1 and P_2 should be such that they permute the elements of the matrix $C_k = A \otimes_k B$ in such a way that the resulting permuted matrix is equal to the matrix $C_c = A \otimes_c B$. In Eq. (4.30) the matrix C_k is to be premultiplied with the matrix P_1 and postmultiplied with the matrix P_2. It is known that premultiplication of a matrix with a matrix having a single element one in each row and each column, rest of the elements being zero, results in rowwise permutation of the former matrix. Similarly a columnwise permuted version of a matrix can be obtained by post-multiplying it with a matrix having a single element one in each row and each column, rest of the elements being zero. So in Eq. (4.30) rowwise permutation on the matrix C_k is effected by the matrix P_1 and columnwise permutation by the matrix P_2. The effect of the premultiplication and postmultiplication of the matrix C_k is that its element can be shifted from any location in the two-dimensional array to any desired location by suitable choice of matrices P_1 and P_2.

It is evident from the definition of C_k that in its i^{th} row the $b(i_2', j_2')$ part of the element $c_k(i,j) =$

$a(i_1', j_1')\ b(i_2', j_2')$ would repeat itself at all $j \equiv j_2'$ modulo N_2 Similarly in j^{th} column of C_k the $b(i_2', j_2')$ part would repeat itself at all $i \equiv i_2'$ modulo M_2.

Let the permutation matrix P_2 of order $N_1 N_2$ be such that it permutes the $(j + j_s)$ modulo $N_1 N_2$ th column of the matrix C_k to j^{th} column. This would mean that in the j^{th} column of the matrix P_2 there should be a one in the $(j+j_s)$ modulo $N_1 N_2$ th row. Further let the permutation matrix P_1 of order $M_1 M_2$ be such that it permutes the resulting columnwise permuted matrix in such a way that its $(i+i_s)$ modulo $M_1 M_2$ th row goes to its i^{th} row. This would mean that in the i^{th} row of the matrix P_1 there should be a one in the $(i+i_s)$ modulo $M_1 M_2$ th column. These could be summarised as

1) The permutation matrix P_2 which is a square matrix of order $N_1 N_2$ with precisely one element a one in each row and each column, rest of the elements being zero, should have ones at locations $((j+j_s)$ modulo $N_1 N_2$ $, j)$.

2) The permutation matrix P_1 which is a square matrix of order $M_1 M_2$ with precisely one element a one in each row and each column, rest of the elements being zero, should have ones at locations,

$(i, (i+i_s)$ modulo $M_1 M_2)$.

3) The permutation matrices P_1 and P_2 are completely defined if expressions for i_s and j_s are obtained in terms of known parameters.

The combined effect of premultiplication by the matrix P_1 and post multiplication by the matrix P_2 is to obtain the condition

$$a(i_1, j_1) = a(i_1', j_1') \quad \text{for all } i, j. \qquad \ldots (4.38)$$

while maintaining the condition given by Eq. (4.37). Since the matrix P_2 is to effect columnwise permutation on the matrix C_K, hence it should result in

$$j_1 = j_1'$$

or

$$j - N_1 [j/N_1] = [j/N_2] \qquad \ldots (4.39)$$

This can be achieved by taking

$$j_s \propto (j_1 - j_1')$$

$$= j - N_1 [j/N_1] - [j/N_2] \qquad \ldots (4.40)$$

Since j_1 repeats itself with a period of N_1 hence j_s should be taken modulo N_1. This gives

$$j_s \propto (j - N_1[j/N_1] - [j/N_2]) \text{ modulo } N_1 \qquad \ldots (4.41)$$

This equation alongwith Eq. (4.37) suggests that the proportionality constant should be N_2 and hence

$$j_s = N_2(j- [j/N_2])\text{modulo } N_1 \qquad \ldots (4.42)$$

The locations of one's in the permutation matrix P_2 are $((j+N_2(j-[j/N_2])\text{modulo } N_1)\text{modulo } N_1 N_2, j)$ ... (4.43)

Since the matrix P_1 is to effect rowwise permutation hence it should result in

$$i_1 = i_1'$$

or

$$i - M_1 [i/M_1] = [i/M_2] \qquad \ldots (4.44)$$

This can be achieved by taking

$$i_s \,\alpha\, (i_1 - i_1')$$

$$= i- M_1[i/M_1] - [i/M_2] \qquad \ldots (4.45)$$

Since i_1 repeats itself with a period of M_1 hence i_s should be taken modulo M_1. This gives

$$i_s \,\alpha\, (i-M_1 [i/M_1]-[i/M_2]) \text{ modulo } M_1 \qquad \ldots (4.46)$$

This equation alongwith Eq. (4.37) suggests that the proportionality constant should be M_2 and hence

$$i_s = M_2 (i-[i/M_2]) \text{ modulo } M_1 \qquad \ldots (4.47)$$

The location of ones in the permutation matrix P_1 are $(i,(i+M_2(i-[i/M_2]) \text{ modulo } M_1) \text{ modulo } M_1 M_2)$

... (4.48)

Thus the permutation matrices P_1 and P_2 are completely defined with the help of Eqs. (4.48) and (4.43) respectively. The two matrices P_1 and P_2 would be equal if A and B are square matrices.

Example Let A and B be two square matrices of orders two and three respectively and given as

$$A = \begin{bmatrix} a(0,0) & a(0,1) \\ a(1,0) & a(1,1) \end{bmatrix}$$

$$B = \begin{bmatrix} b(0,0) & b(0,1) & b(0,2) \\ b(1,0) & b(1,1) & b(1,2) \\ b(2,0) & b(2,1) & b(2,2) \end{bmatrix}$$

Then

$$C_k = A \otimes_k B = \begin{bmatrix} a(0,0)b(0,0) & a(0,0)b(0,1) & a(0,0)b(0,2) & a(0,1)b(0,0) \\ & & a(0,1)b(0,1) & a(0,1)b(0,2) \\ a(0,0)b(1,0) & a(0,0)b(1,1) & a(0,0)b(1,2) & a(0,1)b(1,0) \\ & & a(0,1)b(1,1) & a(0,1)b(1,2) \\ a(0,0)b(2,0) & a(0,0)b(2,1) & a(0,0)b(2,2) & a(0,1)b(2,0) \\ & & a(0,1)b(2,1) & a(0,1)b(2,2) \\ a(1,0)b(0,0) & a(1,0)b(0,1) & a(1,0)b(0,2) & a(1,1)b(0,0) \\ & & a(1,1)b(0,1) & a(1,1)b(0,2) \\ a(1,0)b(1,0) & a(1,0)b(1,1) & a(1,1)b(1,2) & a(1,1)b(1,0) \\ & & a(1,1)b(1,1) & a(1,1)b(1,2) \\ a(1,0)b(2,0) & a(1,0)b(2,1) & a(1,0)b(2,2) & a(1,1)b(2,0) \\ & & a(1,1)b(2,1) & a(1,1)b(2,2) \end{bmatrix}$$

$$
C_c = A \circledast_c B = \begin{bmatrix}
a(0,0)b(0,0) & a(0,1)b(0,1) & a(0,0)b(0,2) & a(0,1)b(0,0) \\
 & & a(0,0)b(0,1) & a(0,1)b(0,2) \\
a(1,0)b(1,0) & a(1,1)b(1,1) & a(1,0)b(1,2) & a(1,1)b(1,0) \\
 & & a(1,0)b(1,1) & a(1,1)b(1,2) \\
a(0,0)b(2,0) & a(0,1)b(2,1) & a(0,0)b(2,2) & a(0,1)b(2,0) \\
 & & a(0,0)b(2,1) & a(0,1)b(2,2) \\
a(1,0)b(0,0) & a(1,1)b(0,1) & a(1,0)b(0,2) & a(1,1)b(0,0) \\
 & & a(1,0)b(0,1) & a(1,1)b(0,2) \\
a(0,0)b(1,0) & a(0,1)b(1,1) & a(0,0)b(1,2) & a(0,1)b(1,0) \\
 & & a(0,0)b(1,1) & a(0,1)b(1,2) \\
a(1,0)b(2,0) & a(1,1)b(2,1) & a(1,0)b(2,2) & a(1,1)b(2,0) \\
 & & a(1,0)b(2,1) & a(1,1)b(2,2)
\end{bmatrix}
$$

The locations of ones in the permutation matrix P_1 of order six are listed below:

Row of matrix P_1	i_s	Column of matrix P_1
0	0	0
1	3	4
2	0	2
3	0	3
4	3	1
5	0	5

i.e., the locations (0,0), (1,4), (2,2), (3,3), (4,1) and (5,5) in matrix P_1 would have entries 1's and rest of the locations would have entries 0's.

Similarly the locations of ones in the permutation matrix P_2 of order six are listed below:

Column of matrix P_2	j_s	row of matrix P_2
0	0	0
1	3	4
2	0	2
3	0	3
4	3	1
5	0	5

i.e., the locations (0,0), (1,4), (2,2), (3,3), (4,1) and (5,5) in matrix P_2 would have entries 1's and rest of the locations would have entries 0's. Thus

$$P_1 = P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It can be easily verified that

$$C_c = P_1 \quad C_k \quad P_2$$
$$C_k = P_1^{-1} \quad C_c \quad P_2^{-1}$$

4.3 RELATION BETWEEN CHINESE-KRONECKER AND KRONECKER-CHINESE PRODUCTS

It has been stated in section (4.1) that matrices resulting from Kronecker-Chinese product and Chinese-Kronecker product of two matrices, if defined, are the permuted versions of the matrix obtained by Kronecker product of the same two matrices. It implies that the matrices obtained by Kronecker-Chinese product and Chinese-Kronecker product of two matrices, if defined, are the rowwise and columnwise permuted versions of each other. It should thus be possible to obtain one from the other by premultiplication and postmultiplication with suitable permutation matrices P_1 and P_2. In what follows expressions would be derived for permutation matrices P_1 and P_2 of suitable dimensions which are defined by

$$C_{kc} = P_1 C_{ck} P_2 \quad \text{... (4.49)}$$

and

$$C_{ck} = P_1^{-1} C_{kc} P_2^{-1} \quad \text{... (4.50)}$$

Since the matrices P_1 and P_2 are to effect rowwise and columnwise permutation respectively these matrices would have only a single one in each row and each column, rest of the elements being zero. Further, P_1 and P_2 would be square matrices of order $M_1 M_2$ and $N_1 N_2$ respectively. A change in dummy variables in Eqs. (4.28) and (4.29) would give

$$c_{ck}(i,j) = a(i_1', j_1')b(i_2', j_2') \quad \text{... (4.51)}$$

and

$$\begin{aligned} i &\equiv i_1' \text{ modulo } M_1 \\ &\equiv i_2' \text{ modulo } M_2 \\ j &\equiv j_2' + N_2 j_1' \end{aligned} \quad \text{... (4.52)}$$

From Eq. (4.25) one can write

$$\begin{aligned} i_1 &= [i/M_2] \\ i_2 &= i - M_2 [i/M_2] \\ j_1 &= j - N_1 [j/N_1] \\ j_2 &= j - N_2 [j/N_2] \end{aligned} \quad \text{... (4.53)}$$

Similarly from Eq. (4.52)

$$\begin{aligned} i_1' &= i - M_1\,[i/M_1] \\ i_2' &= i - M_2\,[i/M_2] \\ j_1' &= [j/N_2] \\ j_2' &= j - N_2\,[j/N_2] \end{aligned} \qquad \ldots (4.54)$$

where [x/y] stands for integer part of (x/y). From Eqs. (4.53) and (4.54) it is clear that

$$\begin{aligned} i_2 &= i_2' \quad \text{for all } i \\ j_2 &= j_2' \quad \text{for all } j \end{aligned} \qquad \ldots (4.55)$$

and hence

$$b(i_2, j_2) = b(i_2', j_2') \text{ for all } i,j \qquad \ldots (4.56)$$

This would imply that $b(i_2,j_2)$ part of $c_{kc}(i,j)$ is element by element equal to $b(i_2', j_2')$ part of $c_{ck}(i,j)$. But the corresponding $a(i_1,j_1)$ of $c_{kc}(i,j)$ is not necessarily equal to the corresponding $a(i_1',j_1')$ of $c_{ck}(i,j)$. So the permutation matrices P_1 and P_2 should be such that they permute the elements of the matrix $C_{ck} = A \otimes_{ck} B$ in such a way that the resulting permuted matrix is equal to the matrix $C_{kc} = A \otimes_{kc} B$.

The permutation matrix P_2 which is a square matrix of order N_1N_2 with precisely one element a one in each row and each column, rest of the elements being zero, should have ones at locations $((j+j_s) \text{ modulo } N_1N_2 \, , \, j)$.

The permutation matrix P_1 which is a square matrix of order M_1M_2 with precisely one element a one in each row and each column, rest of the elements being zero, should have ones at locations

$$(i, \, (i+i_s) \text{ modulo } M_1 \, M_2).$$

The permutation matrices P_1 and P_2 are completely defined if expressions for i_s and j_s are obtained in terms of known parameters.

Proceeding as in Section (4.2) one gets

$$i_s = M_2 \, ([i/M_2] - i) \text{ modulo } M_1 \qquad \ldots (4.57)$$

and

$$j_s = N_2 \, (j - [j/N_2]) \text{ modulo } N_1 \qquad \ldots (4.58)$$

The locations of ones in the permutation matrix P_1 are

$$(i, \, (i+M_2([i/M_2] - i) \text{ modulo } M_1) \text{modulo } M_1 \, M_2) \qquad \ldots (4.59)$$

The locations of ones in the permutation matrix P_2 are

$$((j+N_2(j-[j/N_2]) \text{ modulo } N_1) \text{modulo } N_1N_2, \, j) \qquad \ldots (4.60)$$

Thus the permutation matrices P_1 and P_2 are completely defined with the help of Eqs. (4.59) and (4.60) respectively.

Example Let A and B be two matrices of dimensions 3×5 and 4×2 respectively and given as

$$A = \begin{bmatrix} a(0,0) & a(0,1) & a(0,2) & a(0,3) & a(0,4) \\ a(1,0) & a(1,1) & a(1,2) & a(1,3) & a(1,4) \\ a(2,0) & a(2,1) & a(2,2) & a(2,3) & a(2,4) \end{bmatrix}$$

$$B = \begin{bmatrix} b(0,0) & b(0,1) \\ b(1,0) & b(1,1) \\ b(2,0) & b(2,1) \\ b(3,0) & b(3,1) \end{bmatrix}$$

Then

The locations of ones in the permutation matrix P_1 of order 12 are listed below:

Row of matrix P_1	i_s	Column of matrix P_1
0	0	0
1	8	9
2	4	6
3	0	3
4	0	4
5	8	1
6	4	10
7	0	7
8	0	8
9	8	5
10	4	2
11	0	11

$$C_{ck} = Q_{ck}B = \left[\begin{array}{cccccc}
a(0,0)b(0,0) & a(0,0)b(0,1) & \begin{array}{c}a(0,1)b(0,0)\\a(0,3)b(0,0)\end{array} & \begin{array}{c}a(0,1)b(0,1)\\a(0,3)b(0,1)\end{array} & \begin{array}{c}a(0,2)b(0,0)\\a(0,4)b(0,0)\end{array} & \begin{array}{c}a(0,2)b(0,1)\\a(0,4)b(0,1)\end{array} \\
a(1,0)b(1,0) & a(1,0)b(1,1) & \begin{array}{c}a(1,1)b(1,0)\\a(1,3)b(1,0)\end{array} & \begin{array}{c}a(1,1)b(1,1)\\a(1,3)b(1,1)\end{array} & \begin{array}{c}a(1,2)b(1,0)\\a(1,4)b(1,0)\end{array} & \begin{array}{c}a(1,2)b(1,1)\\a(1,4)b(1,1)\end{array} \\
a(2,0)b(2,0) & a(2,0)b(2,1) & \begin{array}{c}a(2,1)b(2,0)\\a(2,3)b(2,0)\end{array} & \begin{array}{c}a(2,1)b(2,1)\\a(2,3)b(2,1)\end{array} & \begin{array}{c}a(2,2)b(2,0)\\a(2,4)b(2,0)\end{array} & \begin{array}{c}a(2,2)b(2,1)\\a(2,4)b(2,1)\end{array} \\
a(0,0)b(3,0) & a(0,0)b(3,1) & \begin{array}{c}a(0,1)b(3,0)\\a(0,3)b(3,0)\end{array} & \begin{array}{c}a(0,1)b(3,1)\\a(0,3)b(3,1)\end{array} & \begin{array}{c}a(0,2)b(3,0)\\a(0,4)b(3,0)\end{array} & \begin{array}{c}a(0,2)b(3,1)\\a(0,4)b(3,1)\end{array} \\
a(1,0)b(0,0) & a(1,0)b(0,1) & \begin{array}{c}a(1,1)b(0,0)\\a(1,3)b(0,0)\end{array} & \begin{array}{c}a(1,1)b(0,1)\\a(1,3)b(0,1)\end{array} & \begin{array}{c}a(1,2)b(0,0)\\a(1,4)b(0,0)\end{array} & \begin{array}{c}a(1,2)b(0,1)\\a(1,4)b(0,1)\end{array} \\
a(2,0)b(1,0) & a(2,0)b(1,1) & \begin{array}{c}a(2,1)b(1,0)\\a(2,3)b(1,0)\end{array} & \begin{array}{c}a(2,1)b(1,1)\\a(2,3)b(1,1)\end{array} & \begin{array}{c}a(2,2)b(1,0)\\a(2,4)b(1,0)\end{array} & \begin{array}{c}a(2,2)b(1,1)\\a(2,4)b(1,1)\end{array} \\
a(0,0)b(2,0) & a(0,0)b(2,1) & \begin{array}{c}a(0,1)b(2,0)\\a(0,3)b(2,0)\end{array} & \begin{array}{c}a(0,1)b(2,1)\\a(0,3)b(2,1)\end{array} & \begin{array}{c}a(0,2)b(2,0)\\a(0,4)b(2,0)\end{array} & \begin{array}{c}a(0,2)b(2,1)\\a(0,4)b(2,1)\end{array} \\
a(1,0)b(3,0) & a(1,0)b(3,1) & \begin{array}{c}a(1,1)b(3,0)\\a(1,3)b(3,0)\end{array} & \begin{array}{c}a(1,1)b(3,1)\\a(1,3)b(3,1)\end{array} & \begin{array}{c}a(1,2)b(3,0)\\a(1,4)b(3,0)\end{array} & \begin{array}{c}a(1,2)b(3,1)\\a(1,4)b(3,1)\end{array} \\
a(2,0)b(0,0) & a(2,0)b(0,1) & \begin{array}{c}a(2,1)b(0,0)\\a(2,3)b(0,0)\end{array} & \begin{array}{c}a(2,1)b(0,1)\\a(2,3)b(0,1)\end{array} & \begin{array}{c}a(2,2)b(0,0)\\a(2,4)b(0,0)\end{array} & \begin{array}{c}a(2,2)b(0,1)\\a(2,4)b(0,1)\end{array} \\
a(0,0)b(1,0) & a(0,0)b(1,1) & \begin{array}{c}a(0,1)b(1,0)\\a(0,3)b(1,0)\end{array} & \begin{array}{c}a(0,1)b(1,1)\\a(0,3)b(1,1)\end{array} & \begin{array}{c}a(0,2)b(1,0)\\a(0,4)b(1,0)\end{array} & \begin{array}{c}a(0,2)b(1,1)\\a(0,4)b(1,1)\end{array} \\
a(1,0)b(2,0) & a(1,0)b(2,1) & \begin{array}{c}a(1,1)b(2,0)\\a(1,3)b(2,0)\end{array} & \begin{array}{c}a(1,1)b(2,1)\\a(1,3)b(2,1)\end{array} & \begin{array}{c}a(1,2)b(2,0)\\a(1,4)b(2,0)\end{array} & \begin{array}{c}a(1,2)b(2,1)\\a(1,4)b(2,1)\end{array} \\
a(2,0)b(3,0) & a(2,0)b(3,1) & \begin{array}{c}a(2,1)b(3,0)\\a(2,3)b(3,0)\end{array} & \begin{array}{c}a(2,1)b(3,1)\\a(2,3)b(3,1)\end{array} & \begin{array}{c}a(2,2)b(3,0)\\a(2,4)b(3,0)\end{array} & \begin{array}{c}a(2,2)b(3,1)\\a(2,4)b(3,1)\end{array}
\end{array}\right]$$

$$
C_{kc} = A \otimes_{kc} B =
\begin{bmatrix}
a(0,0)b(0,0) & a(0,1)b(0,1) & \begin{matrix} a(0,2)b(0,0) \\ a(0,1)b(0,0) \end{matrix} & \begin{matrix} a(0,3)b(0,1) \\ a(0,2)b(0,1) \end{matrix} & \begin{matrix} a(0,4)b(0,0) \\ a(0,3)b(0,0) \end{matrix} & \begin{matrix} a(0,0)b(0,1) \\ a(0,4)b(0,1) \end{matrix} \\
a(0,0)b(1,0) & a(0,1)b(1,1) & \begin{matrix} a(0,2)b(1,0) \\ a(0,1)b(1,0) \end{matrix} & \begin{matrix} a(0,3)b(1,1) \\ a(0,2)b(1,1) \end{matrix} & \begin{matrix} a(0,4)b(1,0) \\ a(0,3)b(1,0) \end{matrix} & \begin{matrix} a(0,0)b(1,1) \\ a(0,4)b(1,1) \end{matrix} \\
a(0,0)b(2,0) & a(0,1)b(2,1) & \begin{matrix} a(0,2)b(2,0) \\ a(0,1)b(2,0) \end{matrix} & \begin{matrix} a(0,3)b(2,1) \\ a(0,2)b(2,1) \end{matrix} & \begin{matrix} a(0,4)b(2,0) \\ a(0,3)b(2,0) \end{matrix} & \begin{matrix} a(0,0)b(2,1) \\ a(0,4)b(2,1) \end{matrix} \\
a(0,0)b(3,0) & a(0,1)b(3,1) & \begin{matrix} a(0,2)b(3,0) \\ a(0,1)b(3,0) \end{matrix} & \begin{matrix} a(0,3)b(3,1) \\ a(0,2)b(3,1) \end{matrix} & \begin{matrix} a(0,4)b(3,0) \\ a(0,3)b(3,0) \end{matrix} & \begin{matrix} a(0,0)b(3,1) \\ a(0,4)b(3,1) \end{matrix} \\
a(1,0)b(0,0) & a(1,1)b(0,1) & \begin{matrix} a(1,2)b(0,0) \\ a(1,1)b(0,0) \end{matrix} & \begin{matrix} a(1,3)b(0,1) \\ a(1,2)b(0,1) \end{matrix} & \begin{matrix} a(1,4)b(0,0) \\ a(1,3)b(0,0) \end{matrix} & \begin{matrix} a(1,0)b(0,1) \\ a(1,4)b(0,1) \end{matrix} \\
a(1,0)b(1,0) & a(1,1)b(1,1) & \begin{matrix} a(1,2)b(1,0) \\ a(1,1)b(1,0) \end{matrix} & \begin{matrix} a(1,3)b(1,1) \\ a(1,2)b(1,1) \end{matrix} & \begin{matrix} a(1,4)b(1,0) \\ a(1,3)b(1,0) \end{matrix} & \begin{matrix} a(1,0)b(1,1) \\ a(1,4)b(1,1) \end{matrix} \\
a(1,0)b(2,0) & a(1,1)b(2,1) & \begin{matrix} a(1,2)b(2,0) \\ a(1,1)b(2,0) \end{matrix} & \begin{matrix} a(1,3)b(2,1) \\ a(1,2)b(2,1) \end{matrix} & \begin{matrix} a(1,4)b(2,0) \\ a(1,3)b(2,0) \end{matrix} & \begin{matrix} a(1,0)b(2,1) \\ a(1,4)b(2,1) \end{matrix} \\
a(1,0)b(3,0) & a(1,1)b(3,0) & \begin{matrix} a(1,2)b(3,0) \\ a(1,1)b(3,0) \end{matrix} & \begin{matrix} a(1,3)b(3,1) \\ a(1,2)b(3,1) \end{matrix} & \begin{matrix} a(1,4)b(3,0) \\ a(1,3)b(3,0) \end{matrix} & \begin{matrix} a(1,0)b(3,1) \\ a(1,4)b(3,1) \end{matrix} \\
a(2,0)b(0,0) & a(2,1)b(0,1) & \begin{matrix} a(2,2)b(0,0) \\ a(2,1)b(0,0) \end{matrix} & \begin{matrix} a(2,3)b(0,1) \\ a(2,2)b(0,1) \end{matrix} & \begin{matrix} a(2,4)b(0,0) \\ a(2,3)b(0,0) \end{matrix} & \begin{matrix} a(2,0)b(0,1) \\ a(2,4)b(0,1) \end{matrix} \\
a(2,0)b(1,0) & a(2,1)b(1,1) & \begin{matrix} a(2,2)b(1,0) \\ a(2,1)b(1,0) \end{matrix} & \begin{matrix} a(2,3)b(1,1) \\ a(2,2)b(1,1) \end{matrix} & \begin{matrix} a(2,4)b(1,0) \\ a(2,3)b(1,0) \end{matrix} & \begin{matrix} a(2,0)b(1,1) \\ a(2,4)b(1,1) \end{matrix} \\
a(2,0)b(2,0) & a(2,1)b(2,1) & \begin{matrix} a(2,2)b(2,0) \\ a(2,1)b(2,0) \end{matrix} & \begin{matrix} a(2,3)b(2,1) \\ a(2,2)b(2,1) \end{matrix} & \begin{matrix} a(2,4)b(2,0) \\ a(2,3)b(2,0) \end{matrix} & \begin{matrix} a(2,0)b(2,1) \\ a(2,4)b(2,1) \end{matrix} \\
a(2,0)b(3,0) & a(2,1)b(3,1) & \begin{matrix} a(2,2)b(3,0) \\ a(2,1)b(3,0) \end{matrix} & \begin{matrix} a(2,3)b(3,1) \\ a(2,2)b(3,1) \end{matrix} & \begin{matrix} a(2,4)b(3,0) \\ a(2,3)b(3,0) \end{matrix} & \begin{matrix} a(2,0)b(3,1) \\ a(2,4)b(3,1) \end{matrix}
\end{bmatrix}
$$

i.e., the locations (0,0), (1,9), (2,6), (3,3), (4,4), (5,1), (6,10), (7,7), (8,8), (9,5), (10,2) and (11,11) in matrix P_1 would have entries 1's and rest of the locations would have entries 0's.

Similarly the locations of ones in the permutation matrix P_2 of order 10 are listed below:

Column of matrix P_2	j_s	row of matrix P_2
0	0	0
1	2	3
2	2	4
3	4	7
4	4	8
5	6	1
6	6	2
7	8	5
8	8	6
9	0	9

i.e., the locations (0,0), (1,5), (2,6), (3,1), (4,2) (5,7), (6,8), (7,3), (8,4) and (9,9) would have entries 1's and rest of the locations would have entries 0's. The permutation matrices P_1 and P_2 can now be written as

$$P_1 = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

and

$$P_2 = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

It can be easily verified that

$$C_{kc} = P_1 \, C_{ck} \, P_2$$

and $\qquad C_{ck} = P_1^{-1} \; C_{kc} \; P_2^{-1}$

4.4 PERMUTATION PROPERTIES OF CHINESE AND KRONECKER PRODUCTS OF DFT KERNELS

It has been pointed out in earlier sections that the matrices obtained by Kronecker product and Chinese product of two component matrices of proper dimensions can be obtained from each other by premultiplication and post-miltiplication with suitable permutation matrices. Moharir [41] has reported relation between the DFT of Chinese product of two one-dimensional sequences and the DFTs of the two individual one-dimensional arrays. In this section some more results would be obtained wherein the transform samples of higher order system would be related to transform samples of component lower order systems.

Let $\qquad x_1$ and x_2 be given as

$$x_1 = x_1(m_1) \; , \; m_1 = 0,1,2,\ldots,M_1-1 \qquad \ldots (4.61)$$

$$x_2 = x_2(m_2) \; , \; m_2 = 0,1,2,\ldots,M_2-1 \qquad \ldots (4.62)$$

be any two one-dimensional input signal sample sequences and that M_1, M_2 are coprimes. Further let y_1, y_2 be M_1-term and M_2-term DFTs of discrete sequences x_1 and x_2 respectively where

$$y_1 = y_1(n_1) \quad , \quad n_1 = 0,1,2,\ldots,M_1-1 \qquad \ldots (4.63)$$

$$y_2 = y_2(n_2) \quad , \quad n_2 = 0,1,2,\ldots,M_2-1 \qquad \ldots (4.64)$$

Moharir [41] has reported that

1) The M_1 M_2-term DFT of a sequence which is obtained by Chinese product of the sequences x_1 and x_2 is equivalent to the sequence which results from the Chinese product of y_1 modularly permuted by operator $P(\beta,M_1)$ and y_2 modularly permuted by operator $P(\alpha,M_2)$, where α and β are related to M_1, M_2 as

$$\alpha M_1 + \beta M_2 = 1 \qquad \ldots (4.65)$$

It has been shown that α and β would be coprimes.

2) The output obtained in the above scheme is also equivalent to the sequence obtained by Chinese product of y_1 and y_2 and modularly permuting this sequence by operator $P(\gamma, M_1,M_2)$, where γ is given as

$$\begin{aligned} \gamma &\equiv \beta \text{ modulo } M_1 \\ &\equiv \alpha \text{ modulo } M_2 \end{aligned} \qquad \ldots (4.66)$$

4) The sequence obtained by Chinese product of x_1 modularly permuted by operator $P(a,M_1)$ and x_2 modularly permuted by operator $P(e, M_2)$ is equivalent the sequence obtained by Chinese product of x_1 and x_2 and modularly permuting this sequence by

operator $P(\gamma, M_1M_2)$, where γ is given as

$$\begin{aligned} \gamma &\equiv a \text{ modulo } M_1 \\ &\equiv e \text{ modulo } M_2 \end{aligned} \qquad \ldots (4.67)$$

4) The M_1M_2-term DFT of the sequence obtained in (3) above is equivalent to the sequence obtained by Chinese product of y_1 and y_2 and modularly permuting this sequence by operator $P(\eta, M_1 . M_2)$, where η is given as

$$\eta = \lambda\gamma \qquad \ldots (4.68)$$

where

$$\begin{aligned} \lambda &\equiv b \text{ modulo } M_1 \\ &\equiv d \text{ modulo } M_1 \end{aligned} \qquad \ldots (4.69)$$

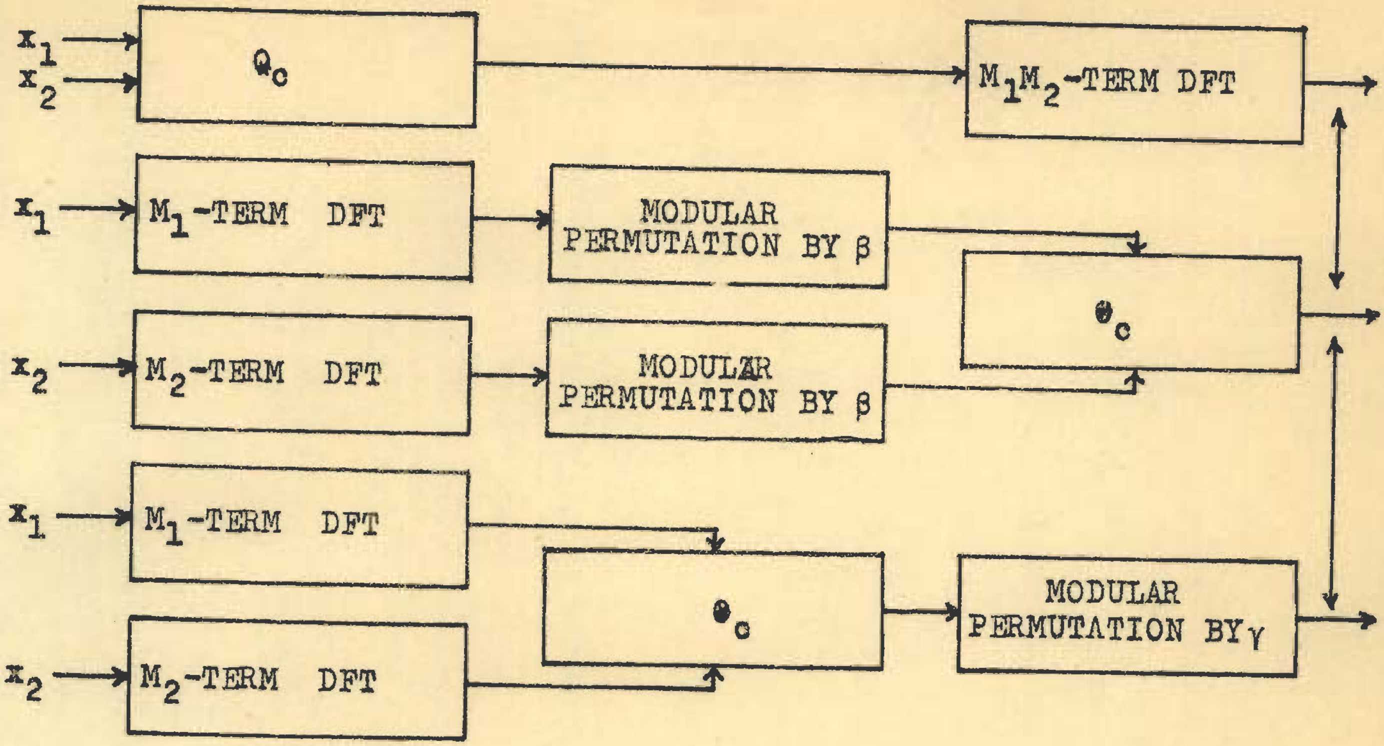
and

$$a\, b \equiv 1 \text{ modulo } M_1 \qquad \ldots (4.70)$$

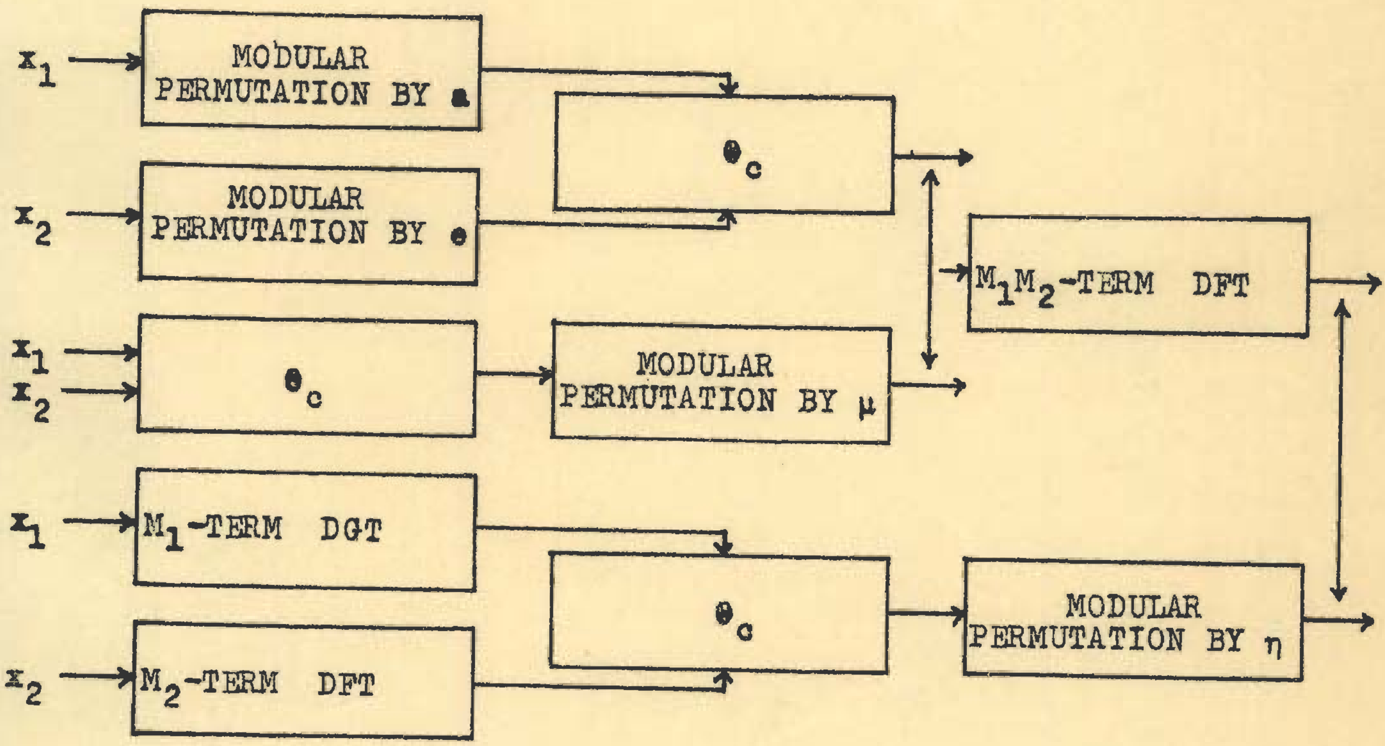
$$e\, d \equiv 1 \text{ modulo } M_2 \qquad \ldots (4.71)$$

It has been reported that α, M_2 and β, M_1 would also be coprimes. The results summarized above have been illustrated in Fig. 4.1. In what follows some of the results of Moharir [41] would be extended to get some new results.

Since y_1, y_2 are M_1-term and M_2-term DFTs of x_1 and x_2 respectively hence



(a)



(b)

FIG. 4.1 : EQUIVALENCE AMONG SYSTEMS WITH SMALLER AND LONGER INPUTS

$$y_1(n_1) = \sum_{m_1=0}^{M_1-1} x_1(m_1) \exp\left(\frac{j2\pi m_1 n_1}{M_1}\right) \quad m_1, n_1 = 0,1,2,..,M_1-1 \qquad \text{... (4.72)}$$

and

$$y_2(n_2) = \sum_{m_2=0}^{M_2-1} x_2(m_2) \exp\left(\frac{j2\pi m_2 n_2}{M_2}\right) \quad m_2, n_2 = 0,1,2,..,M_2-1 \qquad \text{... (4.73)}$$

Consider the scheme given in Fig. 4.2. If $y(n)$ be the Chinese product of sequences $y_1(n_1)$ and $y_2(n_2)$ then

$$y(n) = y_1(n_1)\, y_2(n_2) \qquad n = 0,1,2,...,M_1M_2-1 \qquad \text{... (4.74)}$$

where

$$\begin{aligned} n &\equiv n_1 \text{ modulo } M_1 \\ &\equiv n_2 \text{ modulo } M_2 \end{aligned} \qquad \text{... (4.75)}$$

Substituting for $y_1(n_1)$ and $y_2(n_2)$ in Eq. (4.74) gives

$$y(n) = \sum_{m_1=0}^{M_1-1} x_1(m_1) \exp\left(\frac{j2\pi m_1 n_1}{M_1}\right) \sum_{m_2=0}^{M_2-1} x_2(m_2) \exp\left(\frac{j2\pi m_2 n_2}{M_2}\right)$$

$$= \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} x_1(m_1) x_2(m_2) \exp\left[j2\pi\left(\frac{m_1 n_1}{M_1} + \frac{m_2 n_2}{M_2}\right)\right]$$

$$= \sum_{m=0}^{M_1M_2-1} x(m) \exp\left[j2\pi\left(\frac{m_1 n_1}{M_1} + \frac{m_2 n_2}{M_2}\right)\right] \qquad \text{... (4.76)}$$

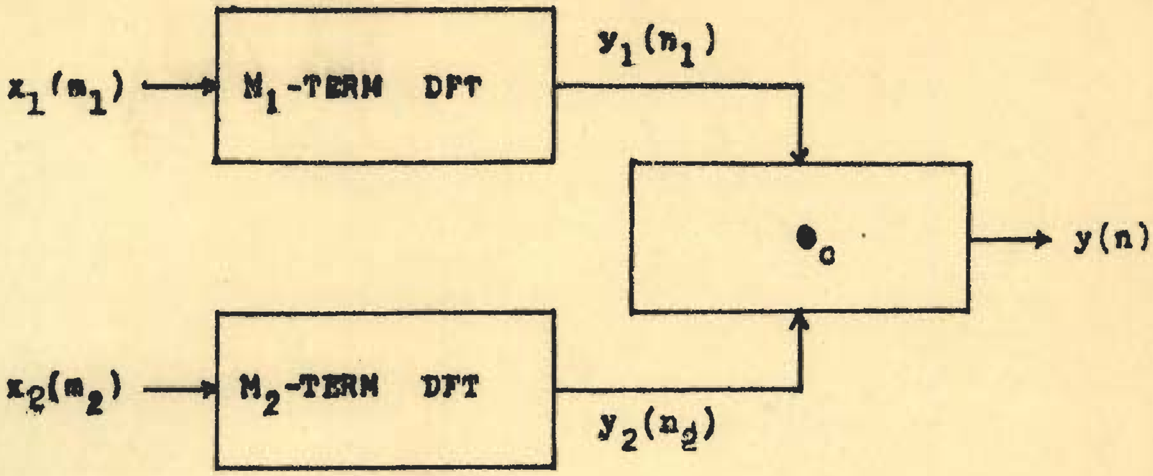


FIG. 4.2 : LINEAR SYSTEM WITH DFT AND CHINESE PRODUCT

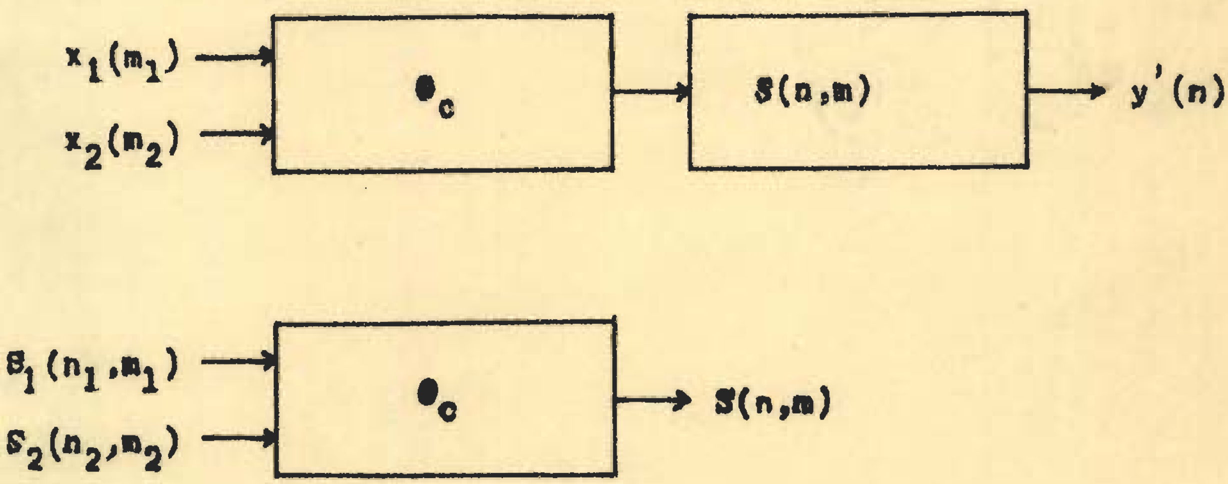


FIG. 4.3 : CHINESE PRODUCT OF SEQUENCE TRANSFORMED BY KERNEL OBTAINED BY CHINESE PRODUCT OF DFT KERNELS

where

$$m \equiv m_1 \text{ modulo } M_1$$
$$\equiv m_2 \text{ modulo } M_2 \qquad \ldots (4.77)$$

and

$$m = \beta\, m_1 M_2 + \alpha\, m_2 M_1 \qquad \ldots (4.78)$$

Next consider the scheme given in Fig. 4.3. If S_1 and S_2 are the M_1-term and M_2-term DFT kernels given as

$$S_1 = [S_1(n_1,m_1)] = [\exp (\frac{j2Xn_1m_1}{M_1})]$$
$$n_1,m_1=0,1,2,\ldots,M_1-1 \qquad \ldots (4.79)$$

and

$$S_2 = [S_2(n_2,m_2)] = [\exp(\frac{j2\pi n_2m_2}{M_2})]$$
$$n_2,m_2=0,1,2,\ldots,M_2-1 \qquad \ldots (4.80)$$

Then the Chinese product of S_1 and S_2 is given by

$$S = S_1 \otimes_c S_2 = [S(n,m)] \qquad \ldots (4.81)$$

$$\text{or, } S(n,m) = S_1(n_1,m_1)\, S_2\, (n_2,m_2) \qquad \ldots (4.82)$$

where

$$n \equiv n_1 \text{ modulo } M_1$$
$$\equiv n_2 \text{ modulo } M_2$$

and

$$m \equiv m_1 \text{ modulo } M_1$$
$$\equiv m_2 \text{ modulo } M_2$$

Substituting $S_1(n_1,m_1)$ and $S_2(n_2,m_2)$ in Eq. (4.82) gives

$$S(n,m) = \exp\left[j2\pi\left(\frac{n_1m_1}{M_1} + \frac{n_2m_2}{M_2}\right)\right] \qquad \ldots (4.83)$$

The two-dimensional array given by Eq. (4.81) represents M_1M_2-term kernel. If $x(m)$, a sequence resulting from Chinese product of sequences x_1 and x_2, be subjected to this M_1M_2-term kernel then the output $y'(n)$, $n = 0,1,2,\ldots, M_1M_2$ would be

$$y'(n) = \sum_{m=0}^{M_1M_2-1} x(m)\exp\left[j2\pi\left(\frac{n_1m_1}{M_1} + \frac{n_2m_2}{M_2}\right)\right] \qquad \ldots (4.84)$$

$$= y(n)$$

It can thus be stated that the sequence obtained by Chinese product of M_1-term DFT of x_1 and M_2-term DFT of x_2 is equivalent to the output of a system the input to which is the sequence resulting from the Chinese product of x_1 and x_2, and the M_1M_2-term transformation kernel is the two-dimensional array resulting from the Chinese product of M_1-term DFT kernel and M_2-term DFT kernel. In a similar way one can prove the following :

1) The sequence resulting from Kronecker product of M_1-term DFT of x_1 and M_2-term HT of x_2 is equivalent to the output of a system the input to which is the sequence resulting from the Kronecker product of x_1 and x_2, and the M_1M_2-term transformation kernel is the two-dimensional array resulting from the Kronecker prodict of M_1-term DFT and M_2-term HT. This has been illustrated in Fig.4.4.

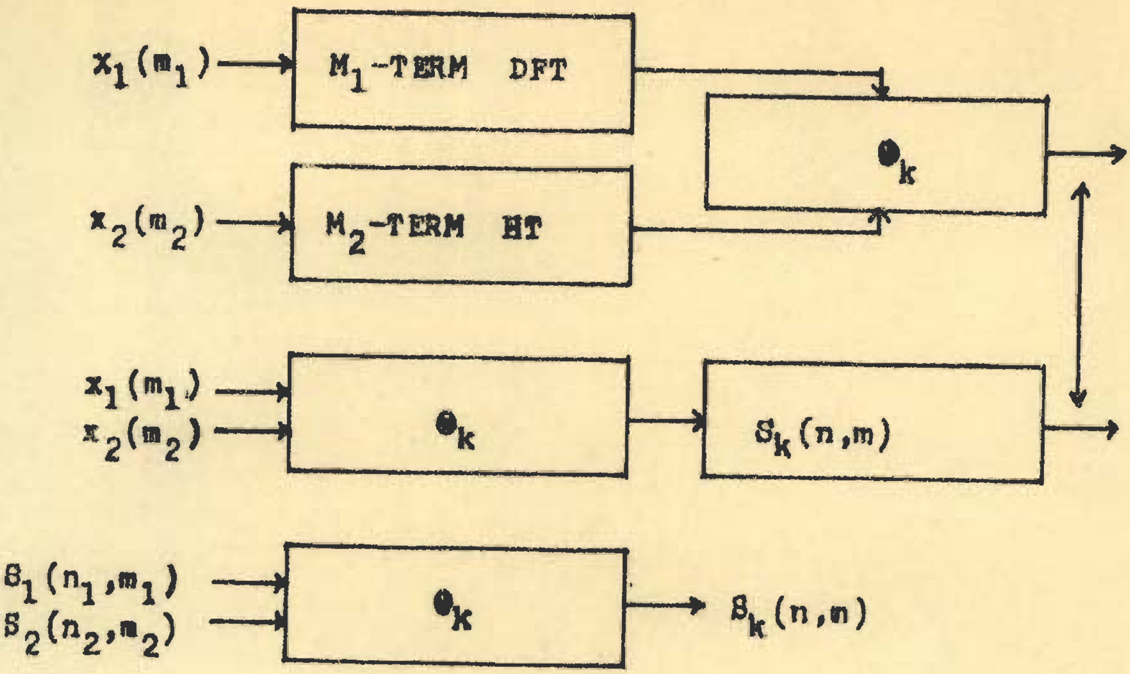


FIG. 4.4 : SOME EQUIVALENT SCHEMES WITH KRONECKER PRODUCTS

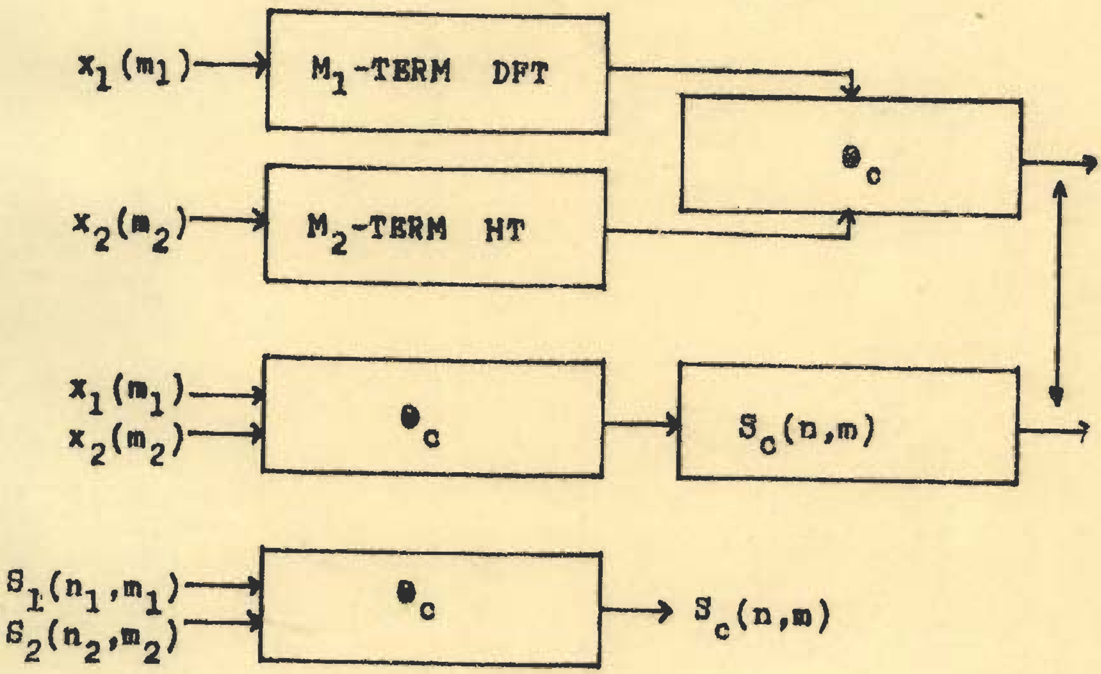


FIG.4.5 : SOME EQUIVALENT SCHEMES WITH CHINESE PRODUCTS

2) The sequence resulting from the Chinese product of M_1-term DFT of x_1 and M_2-term HT of x_2 is equivalent to the output of a system the input to which is the sequence resulting from the Chinese product of x_1 and x_2 , and the M_1M_2-term transformation kernel is the two-dimensional array resulting from the Chinese product of M_1-term DFT and M_2-term HT. This has been illustrated in Fig. 4.5.

C H A P T E R - 5

SYNTHESIS OF TRANSFORM KERNELS

5.1 HADAMARD ARRAYS

A square matrix is an orthogonal matrix if its transpose and inverse are equal except for a constant factor. In an orthogonal matrix:

a) The sum of the sequares of all the elements of any of its rows is equal to unity, i.e., the normalization is done to unity. The normalization can be done to any other number.

b) the sum of the products of the corresponding elements of any two distinct rows is zero.

c) the value of its determinant is equal to $\pm$ 1.

A matrix that is inverse to an orthogonal matrix will itself be orthogonal. The product of orthogonal matrices is an orthogonal matrix.

A Hadamard matrix is a matrix with entries $\pm$ 1 and whose row vectors are orthogonal.

Hadamard matrix of rank 1 is $H_1 = [1]$

Hadamard matrix of rank 2 is $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

For all practical purposes H_2 is considered as the basic Hadamard matrix. Hadamard matrices of ranks equal to integer powers of two can be obtained by Kronecker products of Hadamard matrices of proper lower ranks. The name Hadamard matrix comes from the fact that its determinant satisfies Hadamard's determinant theorem with equality. This theorem states that if

$X = [x_{ij}]$ is a matrix of order N then

$$\left|\det X\right|^2 \quad \prod_{i=1}^{N} \prod_{j=1}^{N} \left|x_{ij}\right|^2$$

If H is a Hadamard matrix of order h and normalized to h then

a) $H H^T = h I_h$ where I_h is identity matrix of order h.

b) $\det H = h^{h/2}$

c) $H H^T = H^T H$

d) it may be changed into other Hadamard matrices by rowwise permutation, columnwise permutation and multiplication of rows and columns by -1. The matrices thus obtained are termed as H-equivalents. It is known that not all the matrices of the same order are H-equivalemts.

Every Hadamard matrix is H-equivalent to a Hadamard matrix which has all the elements of its first row and first column as 1. Matrices of the latter form are called 'normalized'. If an Hadamard matrix exists, its order must be 1,2 or 4 N. But Hadamard matrices of order 4N are not known for all values of N. That an Hadamard matrix of order 4N must exist for every N is neither proved nor disproved yet. If H is a normalized Hadamard matrix of order 4N then its every row (column) except the first has 2N -1's and 2N 1's. Further N -1's in any row (column) overlap with N -1's in each other row (column). A good amount of literature dealing with construction of Hadamard matrices from Hadamard arrays is available. A brief review of such literature is conducted in Appendix A.

Hadamard matrices derived from Hadamard arrays can have good application in Hadamard spectrometry [34].

5.2 PARTITIONED MATRIX KRONECKER PRODUCT METHOD

Let A be a matrix of dimension $M_1m \times N_1n$. This is partitioned rowwise and columnwise to give M_1N_1 sub-matrices, $A(i_1,j_1)$ $i_1 = 0,1,2,\ldots,M_1-1$ and $j_1 = 0,1,2,\ldots,N_1-1$ of dimension $m \times n$ each.

$$A = [A(i_1,j_1)] \qquad \begin{array}{l} i_1 = 0,1,2,\ldots,M_1-1 \\ j_1 = 0,1,2,\ldots,N_1-1 \end{array} \qquad \ldots (5.1)$$

$$A = \begin{bmatrix} A(0,0) & A(0,1) & \cdots & A(0,j_1) & \cdots & A(0,N_1-1) \\ A(1,0) & A(1,1) & \cdots & A(1,j_1) & \cdots & A(1,N_1-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ A(i_1,0) & A(i_1,1) & \cdots & A(i_1,j_1) & \cdots & A(i_1,N_1-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ A(M_1-1,0) & A(M_1-1,1) & \cdots & A(M_1-1,j_1) & \cdots & A(M_1-1,N_1-1) \end{bmatrix} \quad \ldots (5.2)$$

$A(i_1,j_1)$ represents a submatrix of dimension m x n at the i_1^{th} partition along the row and j_1^{th} partition along the column.

$$A(i_1,j_1) = [A(mi_1+x,\ nj_1+ł)] \qquad \begin{aligned} x &= 0,1,2,\ldots,m-1 \\ ł &= 0,1,2,\ldots,n-1 \end{aligned} \quad \ldots (5.3)$$

Let B be a matrix of dimension n M_2 x pN_2. This is partitioned rowwise and columnwise to give M_2N_2 sub-matrices of dimension n x p each.

$$B = [B(i_2,j_2)] \qquad \begin{aligned} i_2 &= 0,1,2,\ldots,M_2-1 \\ j_2 &= 0,1,2,\ldots,N_2-1 \end{aligned} \quad \ldots (5.4)$$

$$B = \begin{bmatrix} B(0,0) & B(0,1) & \cdots & B(0,j_2) & \cdots & B(0,N_2-1) \\ B(1,0) & B(1,1) & \cdots & B(1,j_2) & \cdots & B(1,N_2-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ B(i_2,0) & B(i_2,1) & \cdots & B(i_2,j_2) & \cdots & B(i_2,N_2-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ B(M_2-1,0) & B(M_2-1,1) & \cdots & B(M_2-1,j_2) & \cdots & B(M_2-1,N_2 1) \end{bmatrix} \quad \ldots (5.5)$$

$B(i_2,j_2)$ represents the submatrix of dimension n x p at i_2^{th} partition along row and j_2^{th} partition along the column.

$$B(i_2,j_2) = [b(ni_2+\ell,\ pj_2+y)] \quad \begin{matrix} \ell = 0,1,2,\ldots,n-1 \\ y = 0,1,2,\ldots,p-1 \end{matrix} \quad \ldots (5.6)$$

If Kronecker products is taken of matrices A and B treating the submatrices $A(i_1,j_1)$ of dimension mxn and submatrices $B(i_2,j_2)$ of dimension nxp as elements and the resulting matric C is written

$$C = [C(i,j)] \quad \begin{matrix} i = 0,1,2,\ldots,M_1M_2-1 \\ j = 0,1,2,\ldots,N_1N_2-1 \end{matrix} \quad \ldots (5.7)$$

then C(i,j) would be submatrices of dimension mxp and given as

$$C(i,j) = A(i_1,j_1)\ B(i_2,j_2) \qquad \ldots (5.8)$$

where

$$\begin{aligned} i &= i_2 + M_2\, i_1 \\ j &= j_2 + N_2\, j_1 \end{aligned} \qquad \ldots (5.9)$$

The resulting matrix C can be written as

$$C = \begin{bmatrix} C(0,0) & C(0,1) & \ldots & C(0,j) & \ldots & C(0,N_1N_2-1) \\ C(1,0) & C(1,1) & \ldots & C(1,j) & \ldots & C(1,N_1N_2-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ C(i,0) & C(i,1) & \ldots & C(i,j) & \ldots & C(i,N_1N_2-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ C(M_1M_2-1,0) & C(M_1M_2-1,1) & \ldots & C(M_1M_2-1,j) & \ldots & C(M_1M_2-1,N_1N_2-1) \end{bmatrix} \qquad \ldots (5.10)$$

The dimension of the matrix C would be $mM_1M_2 \times pN_1N_2$. In what follows it would be shown that if A and B are orthonormal matrices then the matrix C would also be orthonormal. Thus a procedure has been proposed to obtain an orthonormal kernel of higher order starting with orthonormal kernels of lower orders.

$$C(i,j) = A(i_1,j_1)\ B(i_2,j_2)$$

$$= [a(mi_1+x, nj_1+\ell)]\ [b(ni_2+\ell, pj_2+y)]$$

$$= [\sum_{\ell=0}^{n-1} a(mi_1+x, nj_1+\ell)\ b(ni_2+\ell, pj_2+y)]$$

$$x = 0,1,2,\ldots,m-1$$

$$y = 0,1,2,\ldots,p-1 \qquad \ldots (5.11)$$

If the matrix A is orthonormal it satisfies

$$\sum_{t=0}^{N_1 n-1} a(mi_1+x,t)\ a^*(mi_1'+x',t)$$

$$= \delta mi_1+x, mi_1'+x'$$

$$= 0 \text{ if } i_1 \neq i_1', x \neq x'$$

$$= 1 \text{ if } i_1 = i_1', x = x'$$

$$\ldots (5.12)$$

Similarly if the matrix B is orthonormal it satisfies

$$\sum_{v=0}^{N_2 p-1} b(ni_2+\ell', v)\ b^*(ni_2'+\ell, v) \qquad = \delta_{ni_2+\ell', ni_2'+\ell}$$

$$= 0 \text{ if } i_2 \neq i_2', \ell' \neq \ell$$

$$= 1 \text{ if } i_2 = i_2', \ell' = \ell$$

$$\ldots (5.13)$$

The matrix C can be also be written as

$$C = [c(I,J)] \qquad \begin{array}{l} I = 0,1,2,\ldots,mM_1M_2-1 \\ J = 0,1,2,\ldots,pN_1N_2-1 \end{array} \qquad \ldots (5.14)$$

The indices I and J can alternatively be expressed as

$$\begin{aligned} I &= i_1 m\, M_2 + i_2 m + x \\ J &= j_1 p\, N_2 + j_2 p + y \end{aligned} \qquad \ldots (5.15)$$

The matrix C would be orthonormal if it satisfies

$$\sum_{J=0}^{pN_1N_2-1} c(I,J)\, c^*(I', J) = \delta_{I,I'} \begin{array}{l} = 0 \text{ if } I \neq I' \\ = 1 \text{ if } I = I' \end{array} \qquad \ldots (5.16)$$

where $I' = i_1' \, mM_2 + i_2' m + x'$

Now

$$\sum_{J=0}^{pN_1N_2-1} c(I,J)\, c^*(I', J)$$

$$= [\sum_{y=0}^{p-1} [\sum_{t=0}^{n-1} a(mi_1+x,t) b(ni_2+t,y)] [\sum_{t=0}^{n-1} a^*(mi_1'+x',t) b^*(ni_2'+t,y)]$$

$$+ \sum_{y=0}^{p-1} [\sum_{t=0}^{n-1} a(mi_1+x,t) b(ni_2+t,p+y)] [\sum_{t=0}^{n-1} a^*(mi_1'+x',t) b^*(ni_2'+t,p+y)]$$

$$+ \vdots$$

$$+ \sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,t)b(ni_2+t,(N_2-1)p+y)\right] \left[\sum_{t=0}^{n-1} a^*(mi_1'+x',t)b^*(ni_2'+t,(N_2-1)p+y)\right] \Bigg]$$

$$+ \Bigg[\sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,n+t)b(ni_2+t,y) \right] \left[\sum_{t=0}^{n-1} a^*(mi_1'+x',n+t)b^*(ni_2'+t,y)\right]$$

$$+ \sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,n+t)b(ni_2+t,p+y)\right] \left[\sum_{t=0}^{n-1} a^*(mi_1'+x',n+t)b^*(ni_2'+t,p+y)\right]$$

$$+ \vdots$$

$$+ \sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,n+t)b(ni_2+t,(N_2-1)p+y)\right] \left[\sum_{t=0}^{n-1} [a^*(mi_1'+x',n+t)b^*(ni_2'+t,(N_2-1)p+y)]\right] \Bigg]$$

$$+ \vdots$$

$$+ \Bigg[\sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,(N_1-1)n+t)b(ni_2+t,y)\right] \left[\sum_{t=0}^{n-1} a^*(mi_1'+x',(N_1-1)n+t)b^*(ni_2'+t,y) \right]$$

$$+ \sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,(N_1-1)n+t)b(ni_2+t,p+y)\right] \left[\sum_{t=0}^{n-1} a^*(mi_1'+x',(N_1-1)n+t)b^*(ni_2'+t,p+y)\right]$$

$$+ \vdots$$

$$\vdots$$

$$+ \sum_{y=0}^{p-1} \left[\sum_{t=0}^{n-1} a(mi_1+x,(N_1-1)n+t)\, b(ni_2+t,(N_2-1)p+y) \right]$$

$$\left[\sum_{t=0}^{n-1} a^*(mi_1'+x',(N_1-1)n+t)\, b^*(ni_2'+t,(N_2-1)p+y) \right]] \quad \ldots (5.17)$$

$$[s_0+s_1+\ldots s_{N_2-1}]+[s_{N_2} +\ldots+s_{2N_2-1}]+\ldots+[s_{N_1N_2-N_2}$$

$$+s_{N_1N_2-N_2+1}+\ldots+s_{N_1N_2-1}] \quad \ldots (5.18)$$

The right hand side consists of N_1N_2 terms and each terms represents the sum of products of p corresponding elements of rows I and I' of the matrix C.

$$[s_0+s_1+\ldots+s_{N_2-1}]$$

$$= [a(mi_1+x,0)a^*(mi_1'+x',0)[b(ni_2+0,0)b^*(ni_2'+0,0)$$

$$+b(ni_2+0,1)b^*(ni_2'+0,1)+\ldots+b(ni_2+0,N_2p-1)$$

$$b^*(ni_2'+0,N_2p-1)]$$

$$+a(mi_1+x,0)a^*(mi_1'+x',1)[b(ni_2+0,0)b^*(ni_2'+1,0)+b(ni_2+0,1)$$

$$b^*(ni_2'+1,1)+\ldots+b(ni_2+0,N_2p-1)b^*(ni_2'+1,N_2p-1)]$$

$$+ \vdots$$

$$+a(mi_1+x,0)a^*(mi_1'+x',n-1)[b(ni_2+0,0)b^*(ni_2'+n-1,0)+b(ni_2+0,1)$$

$$b^*(ni_2'+n-1,1)+\ldots+b(ni_2+0,N_2p-1)b^*(ni_2'+n-1,$$

$$N_2p-1)]$$

$$+[a(mi_1+x,1)a^*(mi_1'+x',0)[b(ni_2+1,0)b^*(ni_2'+0,0)+b(ni_2+1,1)$$

$$b^*(ni_2'+0,1)+\ldots+b(ni_2+1,N_2p-1)b^*(ni_2'+0,N_2p-1)]$$

$$+a(mi_1+x,1)a^*(mi_1'+x',1)[b(ni_2+1,0)b^*(ni_2'+1,0)+b(ni_2+1,1)$$

$$b^*(ni_2'+1,1)+\ldots+b(ni_2+1,N_2p-1)b^*(ni_2'+1,N_2p-1)]$$

$$+ \vdots$$

$$+a(mi_1+x,1)a^*(mi_1'+x',n-1)[b(ni_2+1,0)b^*(ni_2'+n-1,0)+b(ni_2+1,1)$$

$$b^*(ni_2'+n-1,1)+\ldots+b(ni_2+1,N_2p-1)b^*(ni_2'+n-1,N_2p-1)]$$

$$+ \vdots$$

$$+[a(mi_1+x,n-1)a^*(mi_1'+x',0)[b(ni_2+n-1,0)b^*(ni_2'+0,0)$$

$$+b(ni_2+n-1,1)b^*(ni_2'+0,1)+\ldots+b(ni_2+n-1,N_2p-1)b^*(ni_2'+0,N_2p-1)]$$

$$+a(mi_1+x,n-1)a^*(mi_1'+x',1)[b(ni_2+n-1,0)b^*(ni_2'+1,0)+b(ni_2+n-1,1)$$

$$b^*(ni_2'+1,1)+\ldots+b(ni_2+n-1,N_2p-1)b^*(ni_2'+1,N_2p-1)]$$

$$+ \vdots$$

$$+a(mi_1+x,n-1)a^*(mi_1'+x',n-1)[b(ni_2+n-1,0)b^*(ni_2'+n-1,0)$$

$$+b(ni_2+n-1,1)b^*(ni_2'+n-1,1)+\ldots+b(ni_2'+n-1,N_2p-1)b^*(ni_2'+n-1,N_2p-1)]$$

$$=a(mi_1+x,0)[a^*(mi_1'+x',0)\delta ni_2,ni_2'+a^*(mi_1'+x',1)\delta ni_2,ni_2'+1+\ldots$$
$$\ldots+a^*(mi_1'+x',n-1)\delta ni_2,ni_2'+n-1]$$
$$+a(mi_1+x,1)[a^*(mi_1'+x',0)\delta ni_2+1,ni_2'+a^*(mi_1'+x',1)\delta ni_2+1,ni_2'+1+\ldots$$
$$\ldots+a^*(mi_1'+x',n-1)\delta ni_2+1,ni_2'+n-1]$$

$$+ \vdots$$

$$+a(mi_1+x,n-1)[a^*(mi_1'+x',0)\delta ni_2+n-1,ni_2'+a^*(mi_1'+x',1)$$
$$\delta_{ni_2+n-1,ni_2'+1}+\ldots+a^*(mi_1'+x',n-1)\delta_{ni_2+n-1,\ ni_2'+n-1}]$$

$$= a(mi_1+x,0)\sum_{t=0}^{n-1}a^*(mi_1'+x',t)\delta_{ni_2,ni_2'+t}$$
$$+ a(mi_1+x,1)\sum_{t=0}^{n-1}a^*(mi_1'+x',t)\delta_{ni_2+1,ni_2'+t}$$
$$+ \vdots$$
$$+a(mi_1+x,n-1)\sum_{t=0}^{n-1}a^*(mi_1'+x',t)\delta_{ni_2+n-1,ni_2'+t}$$
$$= \sum_{t'=0}^{n-1}a(mi_1+x,t')\left[\sum_{t=0}^{n-1}a^*(mi_1'+x',t)\delta_{ni_2+t',ni_2'+t}\right] \quad \ldots (5.19)$$

$$\sum_{J=0}^{pN_1N_2-1} c(I,J)\ c^*(I',J)$$
$$= \sum_{q=0}^{N_1-1}\left[\sum_{t'=0}^{n-1}a(mi_1+x,qn+t')\left[\sum_{t=0}^{n-1}a^*(mi_1'+x',qn+t)\delta_{ni_2+t',ni_2'+t}\right]\right] \quad \ldots (5.20)$$

$$= \sum_{q=0}^{N_1-1} [\sum_{\ell'=0}^{n-1} a(mi_1+x, qn+\ell')[\sum_{\ell=0}^{n-1} a^*(mi_1'+x', qn+\ell)] \quad \ldots (5.21)$$

if B is orthonormal

i.e. $i_2 = i_2'$, $\ell = \ell'$

$= 0$ if B is not orthonormal i.e. $i_2 \neq i_2'$ or $\ell \neq \ell'$

$$\sum_{J=0}^{pN_1N_2-1} c(I,J)c^*(I',J) = \sum_{q=0}^{N_1-1} [\sum_{\ell'=0}^{n-1} a(mi_1+x, qn+\ell')$$

$$a^*(mi_1'+x', qn+\ell')] \text{ if B is orthonormal}$$

$$= \sum_{t=0}^{nN_1-1} a(mi_1+x,t)a^*(mi_1'+x',t)$$

where $t = qn+\ell'$

$= \delta_{mi_1+x, mi_1'+x'}$

$= 1$ if A is also orthonormal i.e.

$i_1 = i_1', x = x'$

$= 0$ if A is not orthonormal i.e.,

$i_1 \neq i_1'$ or $x \neq x'$

Thus

$$\sum_{J=0}^{pN_1N_2-1} c(I,J)\, c^*(I',J) = 1 \text{ if A and B both are orthonormal}$$

$= 0$ if A or B or both are not orthonormal

Thus it has been proved that if A and B are orthonormal matrices then the matrix C defined by Eqs. (5.7) and (5.8) would also be an orthonormal matrix. This method of obtaining higher order orthonormal matrix starting with lower order orthonormal matrices may be termed as Partitioned Matrix Kronecker Product Method. This method when used with techniques given in Appendix A could be used to generate still other orthonormal matrices. For some chosen values of N and t, $N = 4t$, the submatrices A_1, A_2, A_3, A_4 of dimension $t \times t$ each and which are real, symmetric and cyclic can be written with the help of techniques mentioned in Appendix A. These A_i's could be used to obtain a Hadamard matrix of Williamson type of order $N = 4t$. This matrix is known to be orthonormal. Let this matrix be called A and partitioned as in case of matrix A in this section with $M_1 = N_1 = 4$ and $m = n = t$. In a similar way another orthonormal matrix B can be obtained and partitioned with $M_2 = N_2 = 4$ and $n = p = t$. Then

$$C = [C(i,j)] \qquad i,j = 0,1,2,\ldots,15$$

where $C(i,j)$ is a submatrix of dimension $t \times t$.

$$C(i,j) = A(i_1,j_1)\, B(i_2,j_2) \qquad i_1,i_2,j_1,j_2 = 0,1,2,3$$

where

$$i = i_2 + 4\, i_1$$

$$j = j_2 + 4\, j_1$$

All the submatrices A (i_1, j_1) and $B(i_2, j_2)$ would be real, symmetric and cyclic.

The matrix C thus obtained would be an orthonormal matrix of order 16t. This can be used as an orthonormal kernel for processing sample sequences of length 16t.

Let the one-dimensional signal sample sequence X of length 16t be partitioned into 16 partitions and each partition contains t samples. So

$$X = [x_0 \mid x_1 \mid \ldots\ldots \mid x_{15}]^T$$

where x_i is a submatrix of dimension t x 1. With this partitioned matrix as input and the matrix C as orthonormal transform kernel the transform samples can be written as

$$\begin{bmatrix} X_0 \\ \cdots\cdots \\ X_1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ X_{15} \end{bmatrix} = \begin{bmatrix} C(0,0) & C(0,1) & \ldots & C(0,15) \\ C(1,0) & C(1,1) & \ldots & C(1,15) \\ \vdots & \vdots & & \vdots \\ C(15,0) & C(15,1) & \ldots & C(15,15) \end{bmatrix} \begin{bmatrix} x_0 \\ \cdots\cdots \\ x_1 \\ \cdots\cdots \\ \vdots \\ \cdots\cdots \\ x_{15} \end{bmatrix}$$

where the transform samples are also partitioned in submatrices of dimension t x 1 each. The transform samples in any partition are given as

$$X_j = \sum_{i=0}^{15} x_i \, C(i,j)$$

$$= \sum_{i_1=0}^{3} \sum_{i_2=0}^{3} x_{i_2+4i_1} \; A(i_1,j_1) \; B(i_2,j_2)$$

Since all $A(i_1,j_1)$ and $B(i_2,j_2)$ are cyclic sub-matrices the transform kernel C would have the property that some cyclic shift within all the partitions of the signal sample sequence would result in similar cyclic shifts within all the partitions of the transform samples.

With A_i's computed for certain N and t, $N = 4t$, higher order orthonormal kernels ($>4t$) can be obtained using following constructions

1) Baumert-Hall type construction, viz. H[12,4,3], would give orthonormal matrices A and B of order 12t. This would result in orthonormal matrix C of order 144t.

2) Baumert-Hall-Welch type construction, viz. H[20,4,5], would give orthonormal matrices A and B of order 20t. This would result in orthonormal matrix C of order 400t.

3) Quaternion orthonormal type constructions would give orthonormal matrices A and B of order $2^{p+2}t$, $p = 1,2,\ldots,$. This would result in orthonormal matrix C of order $(2^{2p+4}t)$.

5.3 PARTITIONED MATRIX CHINESE PRODUCT METHOD

If A and B are two matrices of dimension $mM_1 x nN_1$ and $nM_2 x pN_2$ respectively such that M_1, M_2 are coprimes and so are N_1, N_2, one can define a matrix

$$C = [C(i,j)] \quad \dots (5.23)$$

such that

$$C(i,j) = A(i_1,j_1)\, B(i_2,j_2) \quad \dots (5.24)$$

where

$$\begin{aligned} i &\equiv i_1 \text{ modulo } M_1 \\ &\equiv i_2 \text{ modulo } M_2 \\ j &\equiv j_1 \text{ modulo } N_1 \\ &\equiv j_2 \text{ modulo } N_2 \end{aligned}$$

The matrix C would be of dimension $mM_1M_2 x pN_1N_2$. Though this matrix C has the same dimension as the matrix C given by Eq. (5.7) and (5.8), the submatrices defined by Eqs. (5.8) and (5.24) are different because the constituent submatrices $A(i_1,j_1)$ and $B(i_2,j_2)$ are different in the two cases. It can be proved on similar lines as adopted in section (5.2) that if A and B are orthonormal matrices then the matrix C defined by Eqs. (5.23) and (5.24) would also be an orthonormal matrix. So this can be thought of as another method of obtaining higher order orthonormal transform kernels from lower order orthonormal transform kernels. This method of getting higher order orthonormal transform kernel starting with lower order orthonormal transform kernels may be termed as Partitioned Matrix Chinese Product Method. This method, however, cannot be used when matrices A and B are obtained by techniques given in Appendix-A as in that case Chinese product of matrices A and B, treating submatrices as elements, would not be defined.

CHAPTER - 6

TRANSLATION INVARIANT SYSTEMS

6.1 TRANSLATION INVARIANT TRANSFORMS

A translation invarient transform is one which is invariant to cyclic shifts in the input in the sense that the transform domain samples remain unchanged when the input samples have undergone cyclic shifts. This usage has to be distinguished from that in Chapter-2. In pattern recognition problems the position of the pattern being recognised is frequently irrelevant. Human eye could be thought of to possess the best pattern recognition ability. If an attempt is made to achieve the pattern recognition ability of the human eye then it is important to know whether the proposed scheme has the redundancy reduction ability similar to human eye. An algorithm may do well with pattern which have undergone an unknown amount of shift, but may not be satisfactory for, say, hand written characters.

Let $a(i)$, $i = 0,1,2,\ldots,N-1$ be a sequence of N samples and

$$A(I) = T\,a(i) \qquad I = 0,1,2,\ldots,N-1 \quad \ldots(6.1)$$

gives the transform samples where T is some transform operator.

Further, let $a_t(i)$ be the sequence a(i) after the sequence is shifted left cyclically by t locations and

$$A_t(I) = T\, a_t(i) \qquad \dots (6.2)$$

gives the transform samples corresponding to $a_t(i)$. Then A(I) is termed as the translation invariant feature of the sequence a(i) if

$$A(I) = A_t(I) \quad \text{for all} \quad 1 \leqslant t \leqslant N-1 \qquad \dots (6.3)$$

A translation invariant transform is defined as an ordered set of translation invariant features of the input sequence. The individual members of this set are known as components of the transform. It is not necessary that the number of components of the transform should be equal to the number of components in the input sequence. For a two-dimensional pattern the translation invariant transform may be defined as the doubly indexed set of translation invariant features of the two-dimensional pattern. Some of the well known translation invariant features are briefly discussed.

1. Magnitude of the Discrete Fourier Transform Components

Let a(i), i = 0,1,2,...,N-1 be the input sample sequence of length N and A(I), I = 0,1,2,...,N-1 the resulting DFT samples. Then $|A(I)|^2$ is invariant to cyclic shifts in a(i). The resulting translation invariant transform X(I) is defined as

$$X(I) = A(I)\,A^{*}(I) \quad , I = 0,1,2,\ldots,N-1 \ldots \quad (6.4)$$

where superscript * denotes complex conjugation.

The DFT of a sequence can be computed quite fast with the help of FFT algorithms given by Cooley and Tukey. This transform has superiority since DFT of a sequence of prime or composite length can also be calculated [48,58]. However, the DFT suffers from round-off errors due to finite word length.

2. Power Spectrum of Hadamard Transform Components

Let a(i), $i = 0,1,2,\ldots,2^n-1$, n being integer, be the input sample sequence of length 2^n and A(I), $I = 0,1,2,\ldots,2^n-1$ the resulting HT samples, then it has been shown by Ahmed et al [3] that its power spectrum X(s) is translation invariant where

$$X(0) = [A(0)]^2 \quad \ldots (6.5)$$

$$X(s) = \sum_{I=2^{s-1}}^{2^s-1} [A(I)]^2 \quad , \quad s = 1,2,\ldots,n \quad \ldots (6.6)$$

It has been shown by Arazi [9] that this HT power spectrum is invariant to many other permutations as well which is not a desirable feature. Since Hadamard transform can be computed by addition and substraction only its computation is quite fast. Fast algorithms for computation of HT and similar to FFT exist. The HT is free from round-off errors due to finite word length if fixed point arithmetic is implied. If the operations are performed in the floating point mode there can still be round-off errors. Studies on translation invariant spectra of the complex BIFORE transform [2], modified complex BIFORE transform [49] and complex Harr transform [50] have been reported by Ahmed et al.

3. Rapid Transform

This transform was given by Reitboeck and Brody[53]. They have given two algorithms, namely Algorithm A and Algorithm B, for the computation of Rapid transform (RT) of a one-dimensional sequence. The tree graphs for these two algorithms are identical with two (out of several possible) tree graphs for the execution of HT [24]. The RT is invariant under translation and reflection. The RT and HT differ, however, in the arithmetic operations at the nodal points and in their general properties. The similarities in the algorithms for RT and HT have been exploited by Ulman [61] to propose a third algorithm for RT. In case of two-dimensional patterns one can use either two one dimensional transforms in succession or the two-dimensional transform given by Reitboeck and Brody. This is named as Algorithm A A and has been derived on the basis of algorithm A for one-dimensional case. A two-dimensional algorithm can also be obtained on the basis of Algorithm B for one-dimensional case. The RT has been reported [53] to have succeeded in a recognition rate of 80% to 100% for letters having different positions, distortions, inclinations, rotation upto 15^{o} and size variation upto 1:3 relative to a reference set of 10 letters. When executed on a digital computer, this transform is much faster than the FFT. Detailed studies on RT have been reported by Wagh [62,63,64,65,66].

The RT is based on two functions - addition and substraction without sign. It has been shown by Wagh [65] that the translation invariant property of this transform is not due to the specific choice of these functions but due to their symmetry. Thus any pair of symmetric functions can be used to define a new translation invariant transform. Such transforms could be computed using algorithms similar to those for RT. A detailed study of some such transforms has been performed by Wagh [65]. The members of this class are transforms which are translation and reflection invariant but may differ from each other very widely with respect to their other properties. The basic feature of the members of the class of translation invariant transforms is that each member of this class is based on a pair of binary symmetric functions. The selection of the appropriate pair of binary symmetric functions would be decided by the application. Some typical binary symmetric functions of two variables x_0 and x_1 are

$$(x_0+x_1),\ |x_0-x_1|,\ x_0x_1,\ (x_0^2+x_1^2),\ (x_0+x_1)\,x_0x_1,$$
$$(x_0^3-x_0+x_1^3-x_1),\ \max(x_0,x_1),\ \min(x_0,x_1) \text{ etc.}$$

In particular if the variables x_0 and x_1 are logical variables then other possible binary symmetric functions are

$(x_0$ AND $x_1)$, $(x_0$ OR $x_1)$, $(x_0$ EOR $x_1)$, $(x_0$ NAND $x_1)$,
$(x_0$ NOR $x_1)$ etc.

4. Max.-min. Transform

If the pair of binary symmetric functions chosen are $\max(x_0,x_1)$ and $\min(x_0,x_1)$ then the resulting translation invariant transform is termed as Max.-min. transform. This transform has been studied in detail by Wagh [65].

A fairly good volume of literature is available which deal with techniques for computation of various translation invariant transforms from the point of view of hardware implementation. It has been observed that the hardware implementation of RT is cheaper and simpler.

It is because of the fact that the two binary symmetric functions used are addition and substraction without sign and identical set of operations at every stage in the evaluation of the transform. The complexity in the implementation of any member of the class of translation invariant transforms is directly dependent upon the complexity in the evaluation of the two binary symmetric functions. In a given situation if only high speed is required then one would go for RT but if there is the consideration of restricting the transform storage memory then the modification of RT as given by Moharir [32] would be preferred.

5. Parseval's Transform

Orthonormal kernels which are circulants, i.e., in which any row is obtained by left cyclic shift of the elements of the previous row, have a lot of structural redundancy. There is only one such kernel if the entries are restricted to ± 1. If the entries are allowed to be $[\pm 1, 0]$ or $[\pm 2, \pm 1, 0]$ many such kernels could be obtained. Listed below are the first rows of some of orthonormal circulant kernels [40].

Order	First Row												
4	1	-1	-1	-1									
6	-1	0	-1	-1	0	1							
6	-1	0	-1	1	0	-1							
6	-1	-1	-1	-1	2	-1							
6	-1	-2	-1	1	-1	1							
7	0	0	-1	0	-1	-1	1						
8	0	-1	0	-1	0	-1	0	1					
8	0	0	-1	-1	0	0	-1	1					
10	0	0	0	-1	-1	0	0	0	-1	1			
10	0	0	-1	0	-1	0	0	-1	0	1			
13	0	0	-1	0	-1	-1	-1	1	1	0	-1	1	-1
13	0	0	-1	0	1	-1	-1	-1	-1	0	1	-1	1
15	-1	-1	-1	-1	2	2	2	-1	2	2	-1	-1	2
	-1	2											
21	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	-1	-1	2
	2	-1	-1	-1	-1	2	-1	2					
28	0	0	-1	0	1	-1	1	0	0	-1	0	-1	1
	1	0	0	1	0	-1	-1	-1	0	0	-1	0	-1
	-1	1											
42	0	0	-1	0	2	-1	1	0	0	-1	0	-1	-1
	1	0	0	2	0	-1	-1	1	0	0	-1	0	-1
	-1	1	0	0	-1	0	-1	-1	-2	0	0	-1	0
	-1	2	1										

42	0	0	-1	0	-1	1	1	0	0	1	0	1	-1
	2	0	0	-1	0	-1	-2	1	0	0	1	0	-2
	-1	-1	0	0	-1	0	-1	1	1	0	0	-2	0
	1	-1	-1										
78	0	0	-1	0	2	-1	-1	1	1	0	2	1	-1
	0	0	-1	0	-1	-1	-1	1	1	0	-1	1	-1
	0	0	2	0	-1	-1	-1	1	-2	0	-1	1	-1
	0	0	-1	0	-1	-1	-1	-2	1	0	-1	1	-1
	0	0	-1	0	-1	-1	2	1	1	0	-1	1	2
	0	0	-1	0	-1	2	-1	1	1	0	-1	-2	-1
121	0	0	0	0	1	-1	1	-1	0	-1	-1	0	-1
	-1	1	1	0	0	1	-1	-1	1	1	1	-1	-1
	1	-1	0	0	1	1	1	-1	0	0	0	-1	0
	0	0	1	1	-1	1	1	-1	0	1	-1	0	-1
	0	-1	0	1	1	0	-1	0	0	1	0	-1	1
	1	0	1	0	0	1	1	0	0	-1	1	0	0
	1	0	1	-1	0	1	1	1	1	1	-1	1	-1
	1	1	0	-1	-1	1	0	1	-1	-1	-1	0	-1
	1	-1	0	1	0	-1	0	-1	-1	-1	0	1	-1
	1	1	1	1									

The Parseval's theorem states that

$$\sum_{\text{all } i} [a(i)][b^*(i)] = \frac{1}{N} \sum_{\text{all } j} [A(j)][B^*(j)]$$

where [a(i)], [A(j)] and [b(i)], [B(j)] are Fourier transform pairs. It implies that if [a(i)] and [b(i)] are orthonormal then [A(j)] and [B(j)] would also be orthonormal.

The product of a DFT kernel and an orthonormal circulant can be thought of as a column by column Fourier transform of the columns of the circulant. Because the circulant is orthonormal the resultant kernel is also orthonormal by Parseval's theorem. Using orthonormal circulant to take a transform of any column vector would provide permutation invariance to cyclic shifts in the sense of Chapter-2, i.e., if the column vector is cyclically shifted so is its transform. If this transform is further subjected to DFT the modulus of the resultant transform will be invariant in the sense of this chapter. This really means that a transform kernel which is obtained by a product of a DFT kernel and an orthonormal circulant has the property that the modulus of the transform is invariant to cyclic shifts in the sense of this chapter. Such orthonormal transforms may be termed as Parseval's transform.

The translation invariant property of these transforms is similar to that of DFT. Since the kernel of this transform is obtained by post multiplying the DFT kernel with an orthonormal circulant of the same order it is hoped that this new transform would exhibit some more properties. It might be of interest to study these transforms in detail.

6.2 CHARACTER RECOGNITION

In the last section some transforms have been mentioned which can be used for pattern recognition purposes. The RT had been shown to be superior to the then existing transforms. Wagh [65] has defined a class of transforms for such purposes and has shown that RT could be thought of as a **member** of that class. He has discussed in det_il a transform : max-min transform or MT. In this section a few more transforms would be defined and their applicability to pattern recognition application discussed. In what follows a pattern would be represented by a two-dimensional array of 0's and 1's. An element in the array would be represented by a 1 if half or more of it is shaded by the pattern, otherwise by a 0. The two-dimensional transform of the arrays would be obtained by two one-dimensional transforms in succession-one rowwise and the other columnwise. In all the transforms that would be discussed in this section the same functional block would be used everywhere in the algorithm. This reduces hardware requirements. The various schemes for obtaining the transforms would make use of the concept of block-multiplexing and strand-multiplexing [37].

1. OR-AND Transform

The transform of a two-dimensional array, representing a pattern by symbols drawn from [0,1], is obtained by taking the rowwise transform, followed by columnwise transform. The sequence of rowwise and columnwise transform could be changed, if desired. The principle of the OR-AND transform for a one-dimensional sequence of length $N = 2^n$, n being integer, is illustrated in Fig. 6.1.

This transform has been found to be invariant to translation, inversion and small rotation. The transform was applied to characters A,E,I,O and U individually and in pairs of two. The observed important properties of this transform are given below:

a) It is distinct for each character when the character is represented as arrays of dimensions 16 x 16 and 16 x 32.

b) It is distinct for all character pairs AE, AI, AO and AU when the character pairs are represented as arrays of dimensions 16 x 32. The property is invariant to commutation of characters in any pair.

c) Similar results hold when the character pairs are represented as arrays of dimension 32 x 16, i.e., characters are written one below the other.

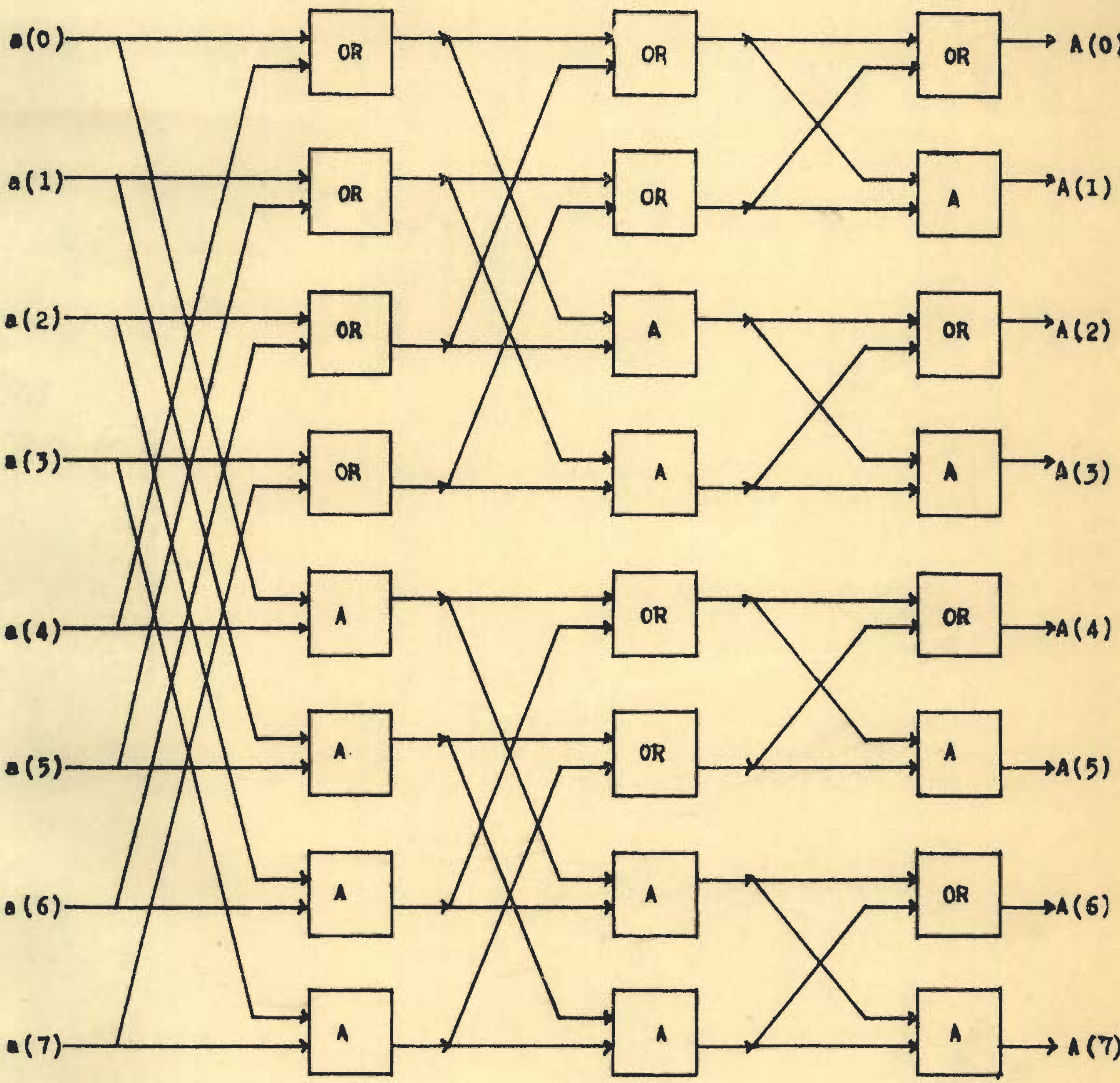


FIG.-6.1 : TREE-GRAPH OF 1-D OR-AND TRANSFORM (8 INPUTS)

d) Similar results are obtained when columnwise transform is taken first and followed by rowwise transform.

e) The number of 1's in the transform domain arrays is the same as that in the corresponding pattern domain arrays. It is due to truth tables for OR and AND gates.

Character A in pattern domain and transform domain is illustrated in Fig. 6.2 and Fig. 6.3 respectively.

2. EOR-AND Transform

This transform is a minor modification of the OR-AND transform. The tree graph for basic one-dimensional transform used is similar to that illustrated in Fig. 6.1 with the difference that all the OR gates are now replaced by EOR gates. The properties observed with this transform are the same as that for OR-AND transform with the difference that the number of 1's in the transform domain arrays are much smaller than those in the pattern domain arrays. It is due to the truth table for EOR gate. This reduction in number of 1's in the transform domain arrays makes this transform superior to OR-AND transform. Characcter A in transform domain is illustrated in Fig. 6.4.

3. Monogenic Function Transform

A monogenic polynomial in N variables is one in which given the leading term the complete polynomial can

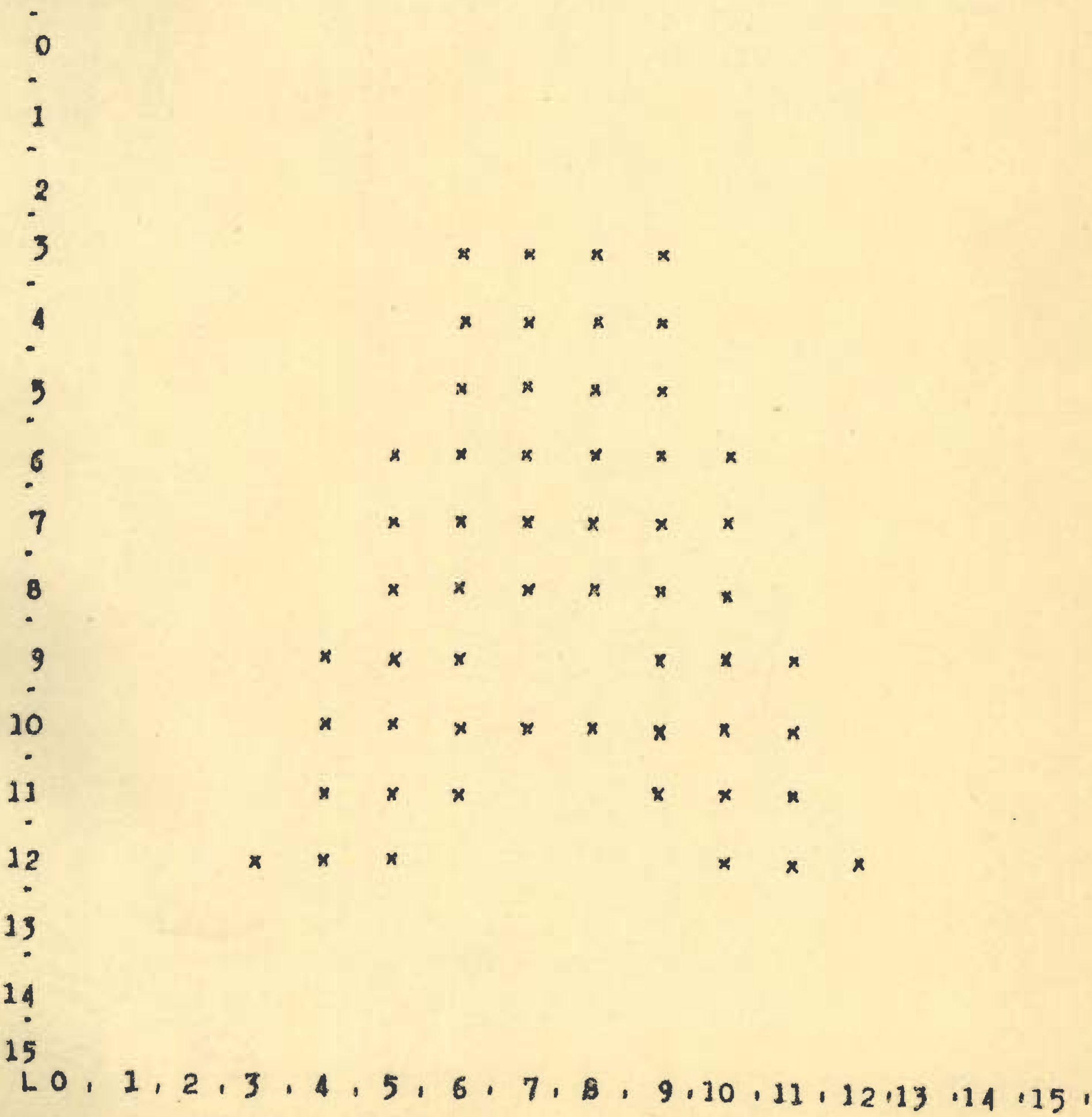


FIG.-6.2 : CHARACTER A IN PATTERN DOMAIN

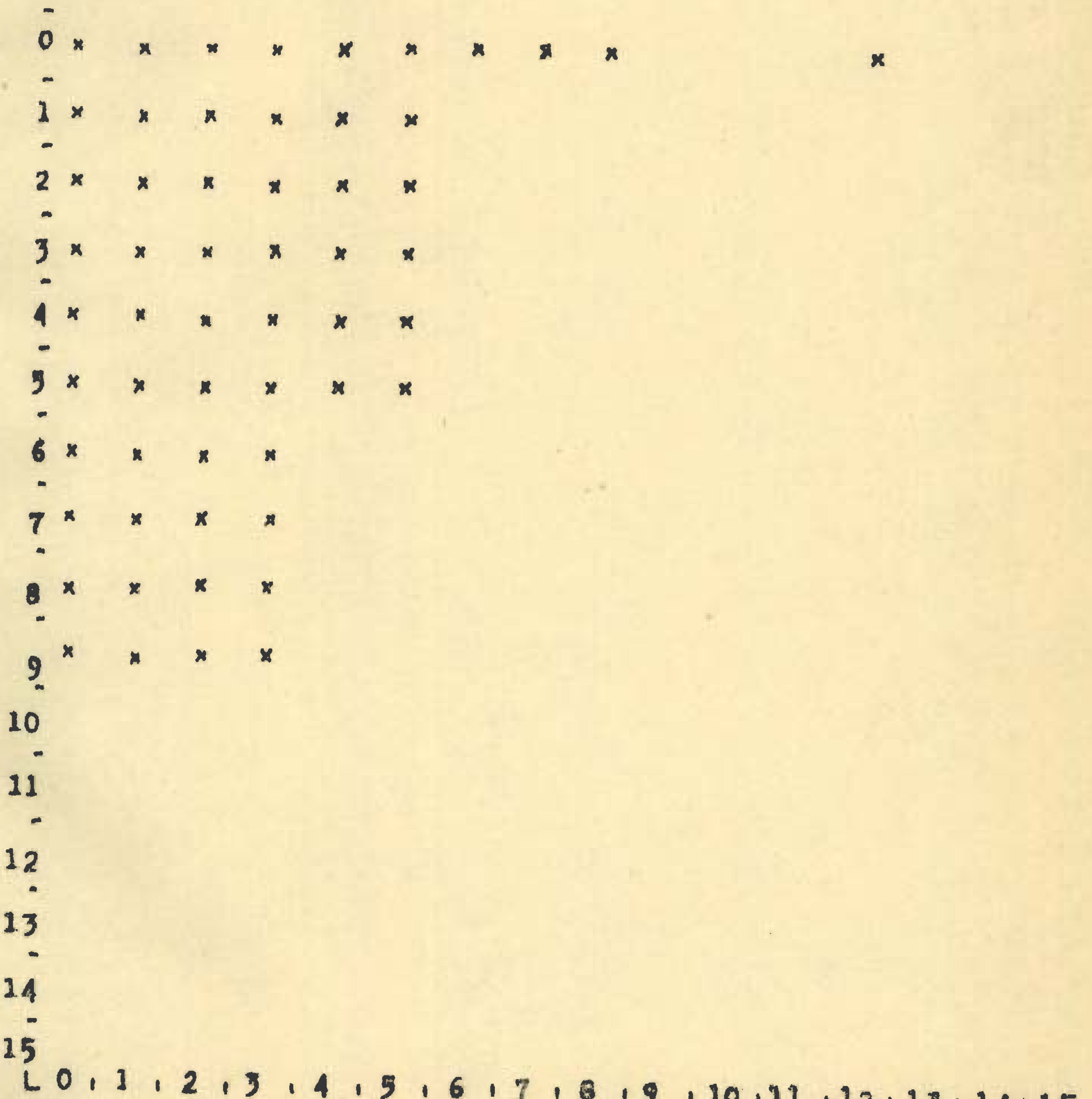


FIG.-6.3 : CHARACTER A TRANSFORMED BY OR-AND TRANSFORM

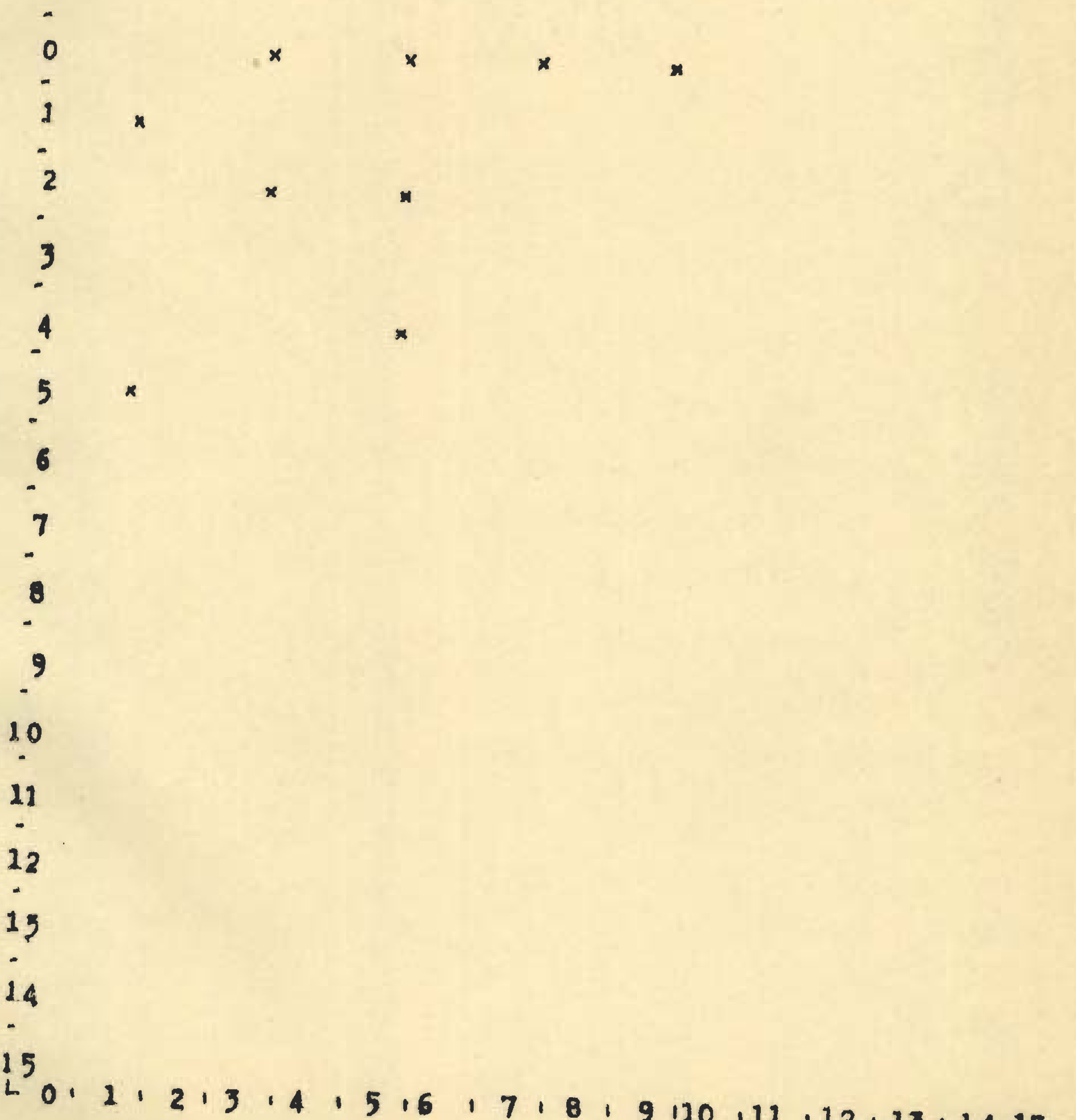


FIG.-6.4 : CHARACTER A TRANSFORMED BY EOR-AND TRANSFORM be written. Any term can be obtained by adding unity modulo N to the indices of variables in the preceeding term. More explicitly a monogenic polynomial of N variables $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{N-1}$ can be written as [18]

$$f(\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{N-1}) + f(\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_{N-1}, \alpha_0) + f(\alpha_2, \alpha_3, \alpha_4, \ldots, \alpha_{N-1}, \alpha_0, \alpha_1) + \ldots\ldots + f(\alpha_{N-1}, \alpha_0, \alpha_1, \ldots, \alpha_{N-2})$$

and $f(\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{N-1})$ is termed as the gene of the monogenic polynomial.

A transform can be defined in terms of monogenic polynomials. If the functional block has N inputs and N outputs then N monogenic polynomials are necessary to define the transformation caused by the functional block. This in turn means that N genes, each a function of N variables-inputs, are needed to define a transform of order N. Since an infinitely large number of genes can be defined in terms of N variables, by properly selecting the genes one can obtain a wide class of such transforms. The transforms thus obtained would be termed as Monogenic Function transforms.

If the inputs are a(i), i = 0,1,2,3, the transform samples are A(I), I = 0,1,2,3 and the genes are a(0), a(0). a(1), a(0).a(1).a(2) and a(0).a(1).a(2).a(3) then a class of monogenic function transform is defined as

$$A(0) = a(0) + a(1) + a(2) + a(3) \qquad \ldots (6.7)$$

$$A(1) = a(0).\ a(1) + a(1).\ a(2) + a(2).\ a(3) + a(3).\ a(0)$$

$$A(2) = a(0).\ a(1).\ a(2) + a(1)\ a(2).\ a(3) + a(2).\ a(3).\ a(0) + a(3).\ a(0).\ a(1)$$

$$A(3) = a(0).\ a(1)\ a(2).\ a(3)$$

The various members of this class of monogenic function transforms can be obtained by choosing various arithmetic, logical or other operations for . and + operations in Eq. (6.7) above.

Another class of monogenic function transform can be obtained by choosing different genes. A typical monogenic function transform was studied where . and + of Eq. (6.7) were chosen as logical AND and OR respectively, and (i), i = 0,1,2,3 was binary : 0 and 1. The resulting transform is defined as

$$A(0) = a(0) \text{ OR } a(1) \text{ OR } a(2) \text{ OR } a(3) \qquad \ldots (6.8)$$

$$A(1) = [a(0) \text{ AND } a(1)] \text{ OR } [a(1) \text{ AND } a(2)] \text{ OR } [a(2) \text{ AND } a(3)] \text{ OR } [a(3) \text{ AND } a(0)]$$

A(2) = [a(0) AND a(1) AND a(2)] OR [A(1) AND a(2) AND a(3)] OR [a(2) AND a(3) AND a(0)] OR [a(3) AND a(0) AND a(1)]

A(3) = a(0) AND a(1) AND a(2) AND a(3)

using this as the basic functional block (F) a one-dimensional monogenic function transform of a sequence of length N = 16 has been illustrated in Fig. 6.5. Monogenic function transform of two-dimensional arrays are obtained by obtaining the rowwise transform followed by columnwise transform.

When this transform was studied with inputs same as in case of OR-AND and EOR-AND transforms, it was observed that the results are similar to that of OR-AND transform. In this transform also the number of 1's in the transform domain array is the same as that in the corresponding pattern domain array . It is due to the definition of the monogenic function transform chosen. Character A in pattern domain and transform domain are illustrated in Fig. 6.2 and Fig. 6.6 respectively.

6.3 LABELLED SFT

The Rapid transform and some other binary symmetric function transforms have been mentioned in section 6.1. These transforms have so far been studied from pattern

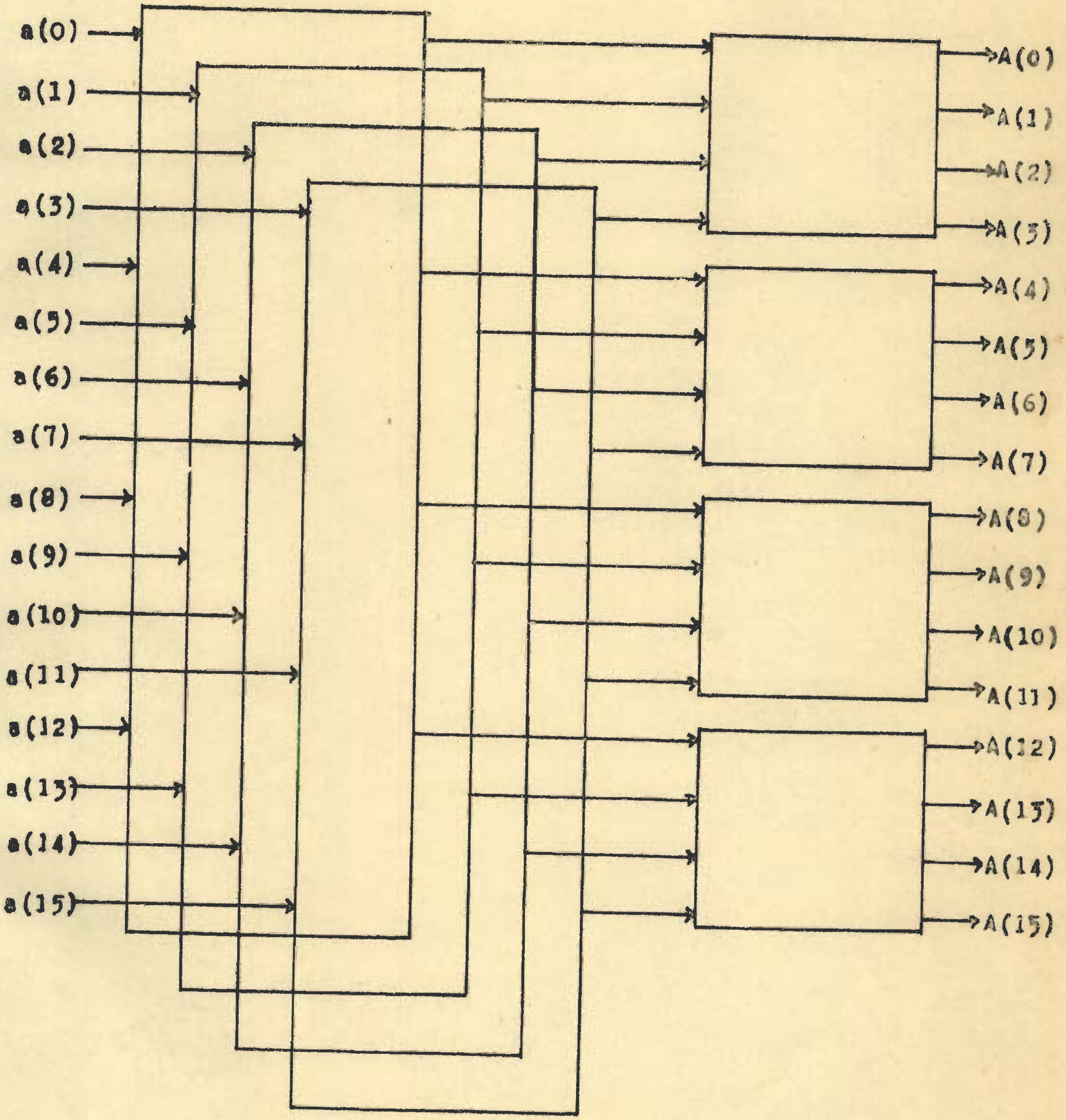


FIG.-6.5 : TREE-GRAPH OF 1-D MONOGENIC FUNCTION TRANSFORM (16 INPUTS)

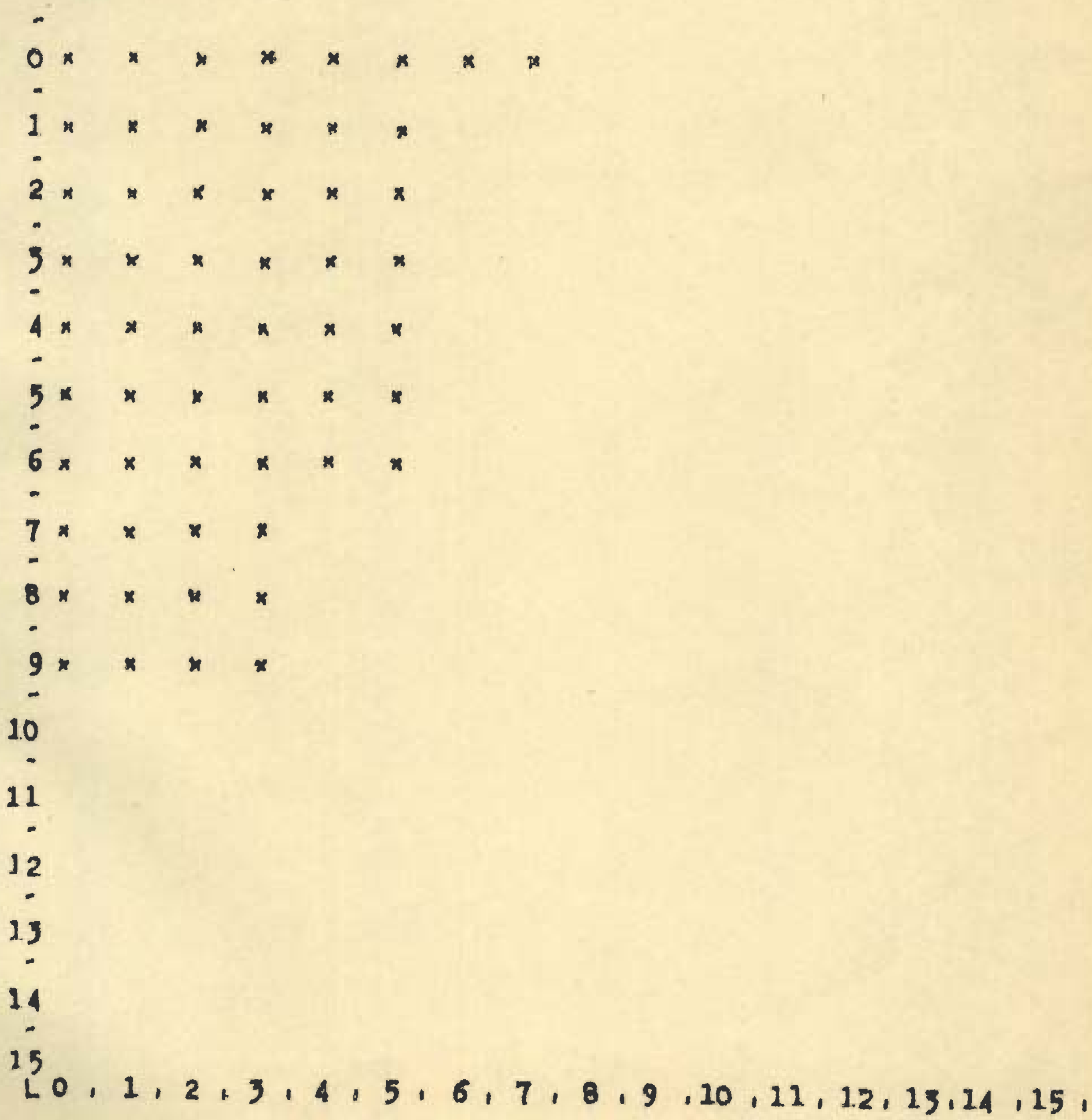


FIG.-6.6 : CHARACTER A TRANSFORMED BY MONOGENIC FUNCTION TRANSFORM recognition point of view where the received transform arrays are correlated with the stored transform arrays and decision given in favour of the character for which the correlation between stored transform array and received transform array is maximum. Some of these transforms like RT are non invertible in the sense that given the transform domain array the pattern domain array cannot be uniquely obtained. In this section an attempt would be made to explore the possibility that such non invertible transforms could be inverted using additional information, if necessary, so that the transformation could be used for coding the digitized message and picture signals.

In case of RT if the input samples are a(i), $i = 0,1,2,\ldots,N-1$, $N = 2^n$, n being integer, then Rapid transform samples A(I), $I = 0,1,2,\ldots,N-1$ are defined as

$$A(I) = Y_n(I) \quad , \quad I = 0,1,2,,\ldots,2^n-1 \qquad \ldots (6.9)$$

where

$$Y_o(I) = a(i) \qquad I = i = 0,1,2,\ldots,2^n-1 \qquad \ldots (6.10)$$

$$Y_r(2I) = Y_{r-1}(I) + Y_{r-1}(I+2^{n-1})$$

$$Y_r(2I+1) = \left| Y_{r-1}(I)-Y_{r-1}(I+2^{n-1}) \right| \qquad \ldots (6.11)$$

$$r = 1,2,\ldots,n$$
$$I = 0,1,2,\ldots,2^{n-1}-1$$

The tree graph for RT has been shown in Fig. 6.7. In taking RT of a sequence of length 2^n there are n intermediate columns and in each intermediate column there are 2^{n-1} substraction without sign operations and 2^{n-1} addition operations. If the input samples a(i) are bounded between 0 and 1 then the transform samples would be bounded between 0 and $2^n L$ where 2^n is the length of the input samples a(i). However, not all the A(I) have the same bounds [32]. The bounds on any A(I) is given by $2^{z(I)} L$ where z(I) is the number of zeroes in the n bit binary representation of I. In particular if the input is binary, i.e., all the input samples a(i) are drawn from [0,1] then the transform samples A(I) would be bounded between 0 and 2^n. Thus each transform sample would need n bits and the transmission of all the transform samples A(I), I = 0,1,2,..., N-1 would require n N bits. If it is desired that all the a(i), i = 0,1,2,...,N-1 be recovered from given A(I), I = 0,1,2,...,N-1 then information about signs at all the $(n2^{n-1})$ substraction points is necessary. If this is done then it would result in a scheme in which the information about labels at all the substraction points is transmitted alongwith the transform samples. This modified scheme may be termed as 'labelled Rapid transform'. Thus it has been possible to invert a non linear transform with the help of labels. In fact any non linear transform can be inverted if

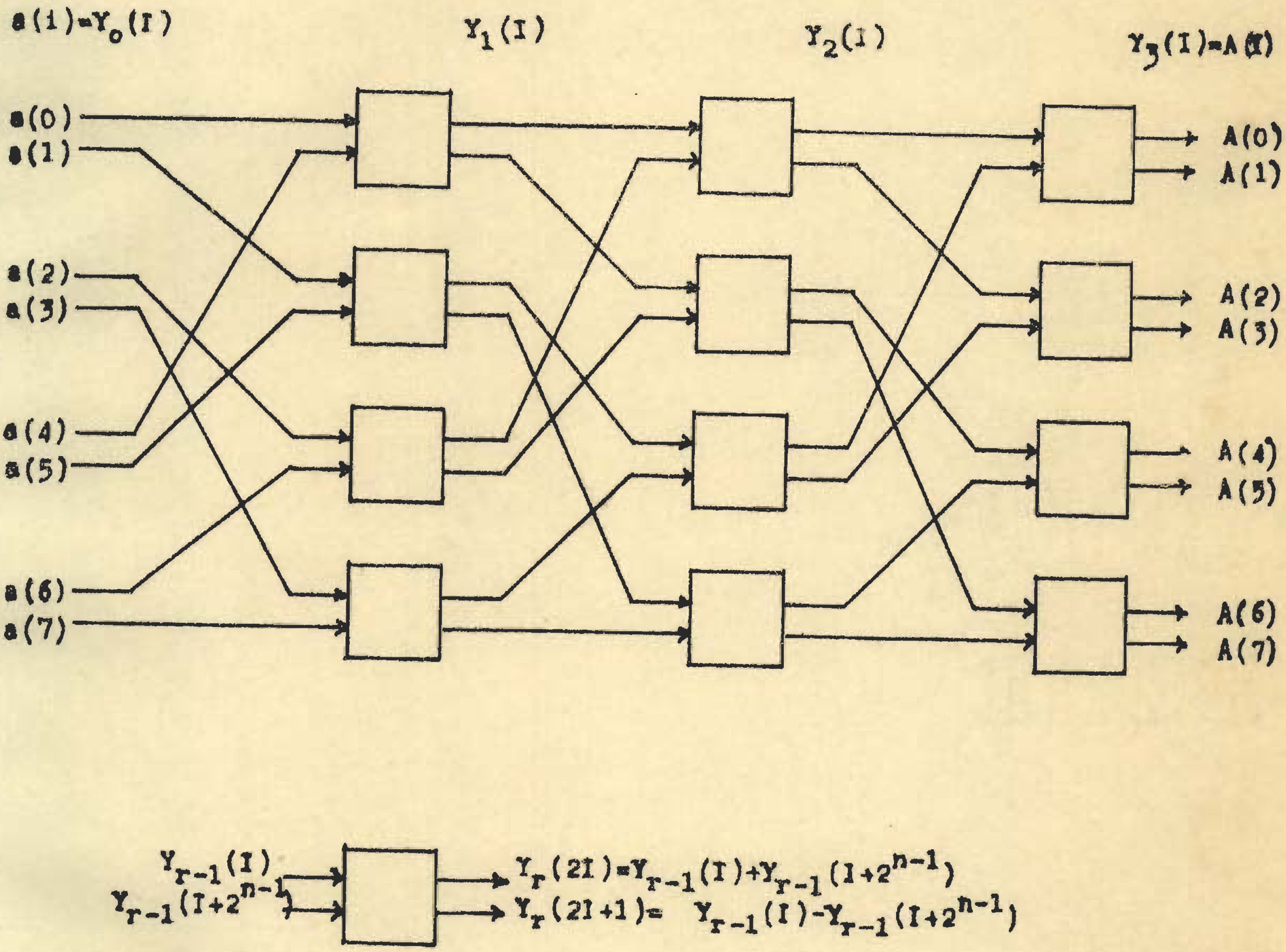


FIG.-6.7 : TREE-GRAPH FOR RT (8 INPUTS)

information regarding suitable labels is available.

The basic functional block for labelled RT has been shown in Fig. 6.8 and complete tree graph in Fig. 6.9. It is clear that this scheme can be used for coding digitized audio and video signals. This scheme would require $(n2^n + n2^{n-1})$ or $(nN + \frac{nN}{2})$ bits in contrast to nN bits required for pattern recognition purposes.

Moharir [32] has suggested modification in RT so that the transform samples A(I) have the same amplitude bounds as those on the input samples a(i). Thus if a(i) takes values zeroes and ones then A(I) would have values between zero and one. The defining equation of the modified RT as given by Moharir are

$$Y_r(2I) = \left| Y_{r-1}(I) + Y_{r-1}(I + 2^{n-1}) \right|_c$$

and

$$Y_r(2I+1) = \left| Y_{r-1}(I) - Y_{r-1}(I + 2^{n-1}) \right|_c$$

$$r = 1,2,\ldots,n \qquad I = 0,1,2,\ldots,2^{n-1}-1 \qquad \ldots (6.12)$$

where

$$\left| x \right|_c = \left| x - x_m \right| \qquad \ldots (6.13)$$

and x_m is the mean of the lower and upper bounds on x. If the input sequence is binary $x_m = 0.5$ and the transform samples A(I) would have values drawn from [0,0.5,1]. If this transform is applied to a binary sequence of

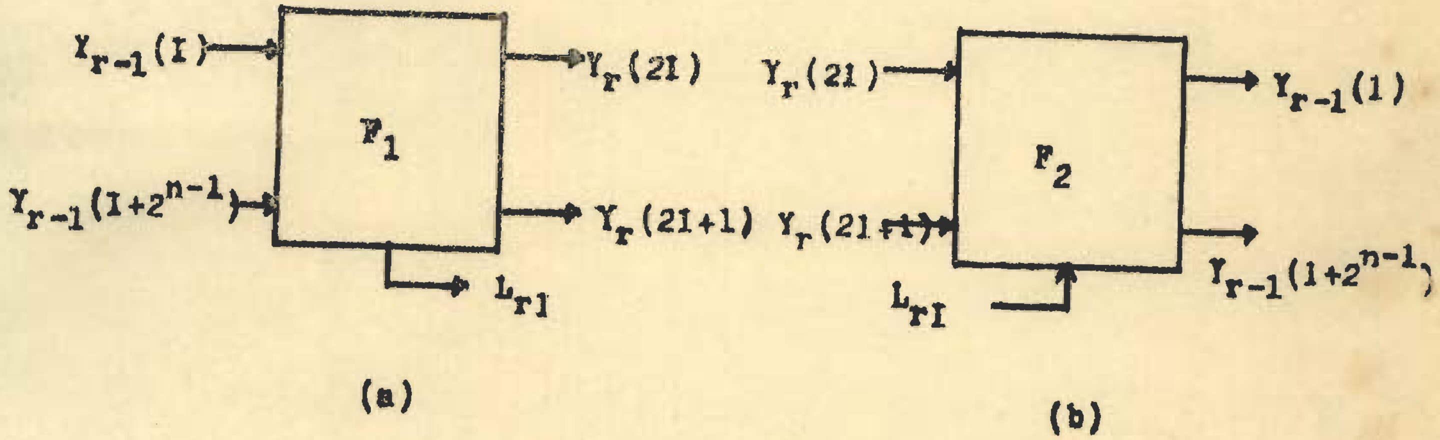


FIG.-6.8 : I^{TH} FUNCTIONAL BLOCK IN r^{TH} COLUMN OF (a) TRANSMITTER (b) RECEIVER IN LABELLED RT

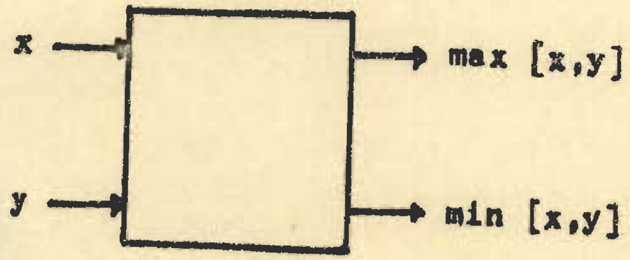


FIG.-6.10 : FUNCTIONAL BLOCK OF MT

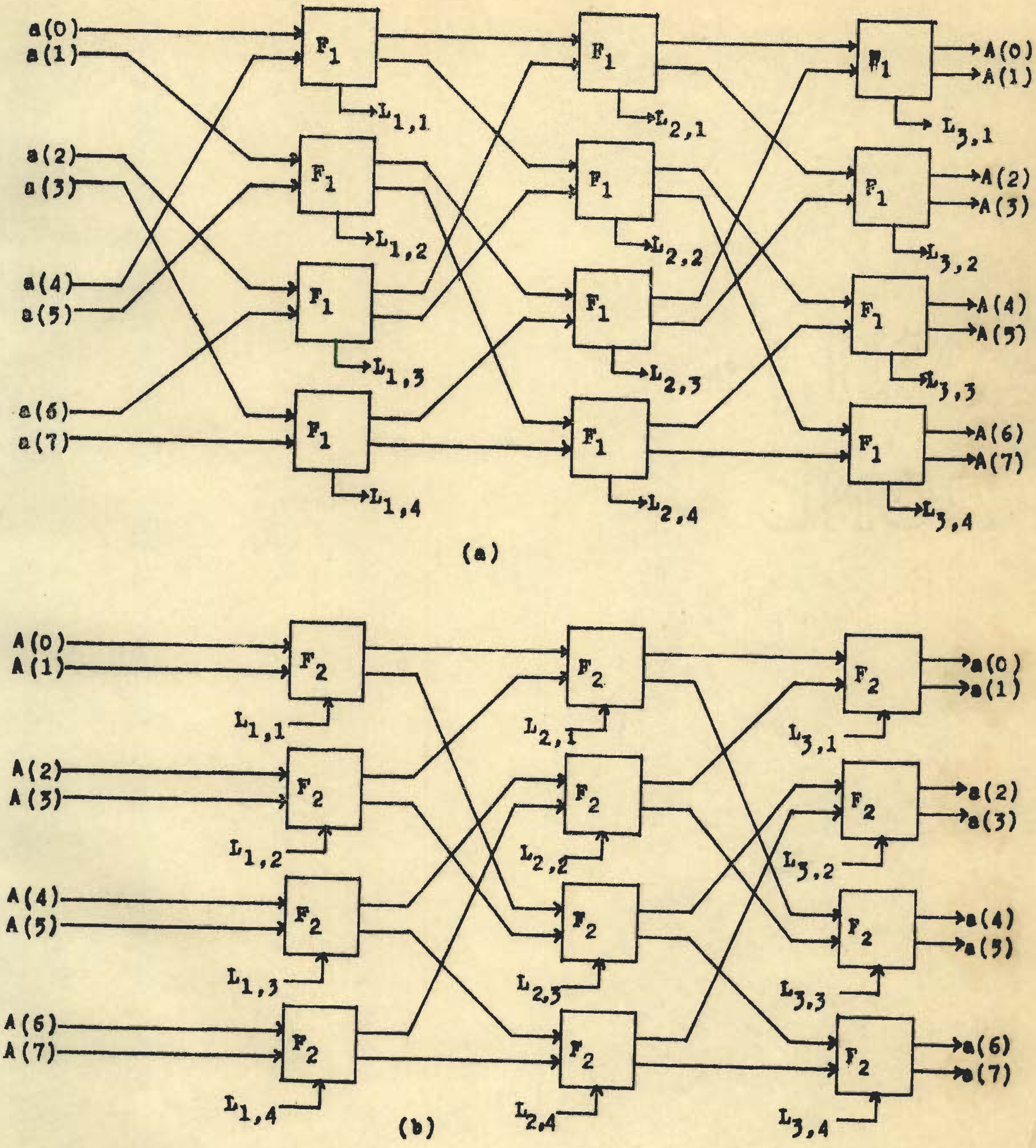


FIG.-6.9 : TREE-GRAPH FOR LABELLED RT (8 INPUTS)
(a) TRANSMITTER (b) RECEIVER

length 2^n then 2N bits would be needed for transmission of transform samples. If inversion is desired at the receiving end then an additional $\frac{nN}{2}$ bits would be needed for transmission of labels. Thus a total of $(2N + \frac{nN}{2})$ bits would be required as against $(nN + \frac{nN}{2})$ bits in case when RT scheme is labelled. Thus there would be a saving of (n-2)N bits for achieving the same objective if the modified RT proposed by Moharir is used in place of RT.

If the basic functional block is max-min transform instead of RT, as illustrated in Fig. 6.10, then both inputs x and y would appear at the output either in the same order or in the reversed order. This order of x and y can be labelled by one bit. The resulting scheme may be termed as 'labelled MT'. If the inputs x and y are from a binary alphabet [0,1] and the input sequence a(i) is of length $N(=2^n)$ then the transform samples A(I), $I = 0,1,2,\ldots,2^n-1$ would also consist of 0's and 1's. Thus N bits would be required for transmission of N transform samples. So if it is desired that a(i), $i = 0,1,2,\ldots,2^n-1$ be recovered from a knowledge of A(I), $I = 0,1,2,\ldots,2^n-1$ then a total of $(N + \frac{nN}{2})$ bits would have to be transmitted. This scheme is superior to labelled RT with Moharir's modification in the sense that with similar inputs and to achieve same objective of getting back the input samples from transform samples this scheme requires N bits less.

Wagh [65] has defined a class of symmetric function transforms (SFT). The scheme proposed in this section can be successfully applied to define a class of 'labelled SFT' by replacing the functional block of Fig. 6.10 with the functional block of any other SFT, and labels used to represent different parameters when different members of the class of SFT are used as basic functional blocks.

The purpose of transmitting labelled SFT instead of the signal could be that (i) SFT inverted with the help of labels is more immune to the deletarious effects of channel noise than the signal when transmitted directly, (ii) SFT may be amenable to severe quantization without having intolerable bad effects on inverted version. Both these properties may arise because of the redistribution of channel or quantization noise that results during inversion with the help of labels. Thus, nonlinear transforms would have been put to use rather than linear ones, with the hope that they may perform better for given purposes.

There is also a good possibility of secrecy coding. SFT as such is non invertible to recover the original signal but with the help of suitable labels it is invertible.

Thus labels serve as a key to unique inversion. If these labels are transmitted directly, they would be available even to unauthorized receivers. Therefore, labels could be subjected to modification with the help of a rule-book which is available only to authorized receiver, and then transmitted. The authorized receiver can recover the labels assuming that the rules are invertible. The rule-book can change frequently but synchronously by prior agreement between the transmitter and authorized receiver. The secrecy would further be enhanced if the SFT used is also changed frequently but synchronously. Use of transforms in secrecy coding has been suggested earlier [35,38].

CHAPTER - 7

CONCLUSIONS

7.1 SUMMARY AND CONCLUSION

The existing literature on finite group theory does not help one find out the number of possible transitive abelian permutation groups of order N and degree N which could be constructed out of a permutation group of degree N and order N! If this could be known then the second part would be to obtain all the elements of each such group. One important property of such groups which could be of some help is that all the elements of any group are the powers of any primitive element of the group. If an attempt is made to work it out by exploiting the definition of the transitive abelian permutation group then the task becomes really difficult for $N \geqslant 6$. It may be possible to write a computer programme for this purpose but that is likely to be quite complicated and may require enormous memory and time for higher values of N. It may be worthwhile trying to develope some fast algorithms for this purpose.

Siddiqui [57] has defined a class of one-dimensional permutation invariant systems (1-D P-I systems) relative to some transitive abelian permutation group. He has obtained the corresponding finite discrete system matrices for realizable cases and has done a comprehensive analysis. The class of 1-D P-I system as given by Siddiqui has been enlarged by extending the notion of permutation invariance. The resulting class of 1-D system which is invariant to some other prescribed permutation has been named as one-dimension reciprocal permutation system (1-D R-P system). It is relative to a pair of transitive abelian permutation groups. In case of 1-D P-I system all the possible system matrices had ranks equal to their respective orders. But in the proposed 1-D R-P system it has been observed that a system matrix with the rank equal to its order did not result in most of the cases. Though some conditions have been stipulated under which the rank of the resulting system matrix would be equal to its order, these conditions cannot be claimed to be very sound as they are based on results obtained with transitive abelian permutation groups of order and degree upto five only. It may be worth to investigate it in detail and propose theorems that would describe cases under which discrete finite system matrices for 1-D R-P system would have ranks equal to their respective orders.

Rao [52] has extended the work of Siddiqui [57] and has given some good results for two-dimensional permutation invariant systems (2-D P-I systems). The notion of proposed 1-D R-P system can be extended to define a class of 2-D R-P system. As in case of 1-D R-P system in this case also many resulting discrete finite system matrices would have ranks lesser than orders. The theorems obtained for defining conditions under which system matrices for 1-D R-P system would have ranks equal to order can be extended to two-dimensional cases.

The permutation property of DFT kernels with respect to 'modular permutation' is known. Also the permutation property of HT kernels, when the order of the kernel is integer power of two, is known. The equivalence between DFT of a one-dimensional sequence and when the sequence is written as a two-dimensional array read row by row is known. The equivalence is established by introducing the concept of twiddling factors. All the elements of the intermediate array obtained as a result of columnwise DFT of the sequence, written as two-dimensional array and read row by row are multiplied with twiddling factors and then rowwise DFT taken of the resulting array. The transform samples are read column by column. Instead of taking columnwise as well as rowwise DFT an attempt has been made to define transform, namely 'Fourier twiddled H-DF transform' where columnwise transform is HT and rowwise transform is DFT. This necessiates a modification in the twiddling factor if some permutation properties are to be retained. It was proposed to study the permutation properties of this transform. The one-dimensional input sample sequence was written as a two-dimensional array. Each row(column) was treated as an element for applying the known permutations for HT (DFT) which are defined for one-dimensional sequences. This resulted in rearrangement of transform samples. The cases studied include bit-plane permutation of rows, modular permutation of columns and both. Expressions have been derived for this rearrangement of transform samples for given permutation of two-dimensional input array. The results obtained have been supported with worked out examples.

The technique developed in respect of Fourier twiddled H-DF transform could be used to obtain a class of such transforms by choosing various pairs of orthonormal kernels for performing columnwise and rowwise transformation on the sequence of input samples written as a two-dimensional array. The twiddling factor would be decided by the nature of the two component transforms if some permutation properties have to be achieved. The permutation properties of the resulting transform would depend upon the permutation properties of the two component transforms.

The concept of Kronecker product of matrices is well known and many theorems related to it are available in the literature. The matrix obtained by Kronecker product of two orthonormal matrices is known to be orthonormal. Thus by taking Kronecker product of two orthonormal matrices of lower order one can generate an orthonormal matrix of higher order. A new matrix multiplication, namely Chinese product, of two matrices has been proposed. Further the matrix obtained by Chinese product of two matrices is the rowwise and columnwise permuted version of the matrix obtained by the Kronecker product of the same component matrices. Since rowwise and columnwise permutation of a matrix does not change its orthonormality hence the matrix obtained by Chinese product of two orthonormal matrices would itself be an orthonormal matrix. Since the matrix obtained by Chinese product of two matrices is a rowwise and columnwise permuted version of the matrix obtained by the Kronecker product of the same matrices it should be possible to obtain one from the other by premultiplication and postmultiplication with suitable permutation matrices. Expressions have been derived which define the two permutation matrices. The results obtained have been illustrated with examples. Now that the conversion of one from the other is possible the algebra available in respect of Kronecker products can be suitably modified so as to be useful in case of Chinese products. Unlike Kronecker product which does not commute this Chinese product commutes. But the Chinese product of any two matrices is not always defined as is the case with Kronecker product. For Chinese product to be defined the respective dimensions of the two arrays should be coprimes i.e., they do not have any factor in common. The notions of Kronecker product and Chinese product have been used to define two more matrix products namely Chinese-Kronecker product and Kronecker-Chinese product. As one would expect, the matrices resulting from such products of two matrices are rowwise or columnwise permuted versions of the matrix obtained by the Kronecker product of the same component matrices. Thus the matrices obtained by Chinese-Kronecker product and Kronecker-Chinese product of two orthonormal matrices would still be orthonormal matrices. These matrix products are defined only when the dimensions of the rows and columns respectively of the two component matrices are coprimes. Expressions have been derived for permutation matrices which would help obtain one from the other. The results obtained have been illustrated with example.

Much work has been reported regarding construction and equivalence of Hadamard matrices. An excellent survey of the existing literature has been compiled by Wallis [67]. There are various constructions which give Hadamard arrays of various orders. The constructions of interest are Williamson design, Baumert-Hall design and Baumert-Hall-Welch design. All these Hadamard arrays yield Hadamard matrices which are orthonormal with the basic components of all these constructions which are four square submatrices known as indeterminates, suitably chosen. In these constructions the indeterminates are real, symmetric and cyclic submatrices. These assumptions simplify the requirements for orthonormality and hence lead to simpler-search.

Two more matrix products have been proposed which give orthonormal matrices of higher orders if the component matrices are orthonormal matrices of lower order. The component matrices are partitioned rowwise and columnwise into submatrices having dimensions such that the ordinary matrix product of submatrix of one with the submatrix of the other is defined. The two original matrices are relabelled treating the submatrices as elements. Kronecker product is taken of the two matrices by treating submatrices as elements. The resulting matrix is a higher order matrix and the method has been named as 'partitioned matrix Kronecker product'. It has been shown that if the starting matrices are orthonormal then the matrix obtained by this matrix product of the component matrices is again an orthonormal matrix. In a similar way another matrix product namely 'partitioned matrix Chinese product' has been defined. In this matrix product Chinese product is taken of component matrices treating their submatrices as elements as in partitioned matrix Kronecker product method. Since the matrix obtained by this method would be a rowwise and columnwise permuted version of the matrix obtained by partitioned matrix Kronecker product method, this matrix would also be orthonormal.

The concept of Chinese product has been applied to linear systems. The advantage of the work reported is that the output of a linear system corresponding to any particular input can be deduced from outputs to simpler inputs in terms of which that particular input can be synthesized.

Many transforms are known which could be used for pattern recognition in general and character recognition in particular. Fast computational algorithms are known for such transforms. The received transform domain arrays are correlated with stored transform domain arrays for various characters and decision given in favour of the character for which the correlation is maximum. Most of these transforms are nonlinear and noninvertible. Since the location of the characters in the pattern domain is of little consequence, many of the transforms used are translation invariant in the sense that if the input sequence undergoes a cyclic shift the transform samples remain unchanged. Besides these transforms may exhibit some more permutation properties. Wagh [65] has proposed a class of such transforms and named them as 'symmetric function transforms'. Various members of this class can be obtained by defining a pair of binary symmetric functions. Some transforms have been defined and their performance evaluated in respect of character recognitions. In case of some of the transforms proposed the locations of ones and zeros make the transform domain arrays quite distinct for various characters but in some cases it is not. The transforms for which the transform domain arrays are distinct for various characters could be classified into two categories : first in which the number of ones in the transform domain are equal to number of ones in the pattern domain and the second in which the number of ones in the transform domain are less. The transforms falling in the latter category are significant.

If the non-linear transforms used for patter recognition could be inverted then they could be used for message or picture transmission. An attempt has been made to invert the nonlinear transforms and obtain the input signal sequence from its transform samples with the help of labels at various functional blocks in the scheme. The resulting system has been named as 'labelled symmetric function transform'. The notion of this transform has been illustrated with examples.

6.2 SCOPE FOR FUTURE WORK

1) The study of permutation invariant systems and the reciprocal permutation system is incomplete unless the listing of all transitive abelian permutation groups of order N and degree N is possible. Formulation of fast algorithms for finding out the total number of such groups and listing of all the elements of each such group is an open problem.

2) Once algorithms for writing all transitive abelian permutation groups of order N and degree N are known it would be worthshile to find out the conditions under which the discrete finite system matrices corresponding to the reciprocal permutation systems would have ranks equal to their orders.

3) A new transform namely Fourier twiddled H-DF transform has been defined and it has been indicated that proper selection of a pair of orthonormal kernels with known permutation properties along-with suitable twiddling factors could be used to define a class of such transforms. A study of relative merits of some such transforms can be made with respect to the complexities in the permutation properties of the members of the class of transforms.

4) The concept of Chinese remainder theorem has been used to define a matrix product known as Chinese product. Relationship between Chinese product and Kronecker product (when the former exists) has been established. Since the Chinese product is commutative its algebra would be simpler than the algebra of Kronecker products which do not commute.

5) The notion of Chinese products of one-dimensional and two-dimensional arrays could be applied in the study of linear systems. It could help in the analysis and design of systems with longer inputs by its equivalent representation in terms of systems with smaller inputs.

6) It has been mentioned that the class of symmetric function transforms which were proposed basically for applications in pattern recognition can be modified so as to be useful for transmission of message, picture and discrete data signals. The modified scheme has been termed as labelled symmetric function transforms. The study of such transforms from information transmission point of view is expected to yield some good results. The relative study of various transforms would give an idea about superiority of one over another with respect to computational complexies, hardware requirements and system performance.

7) As labelled symmetric function transforms provide equivalent alternative descriptions of the signals, it would be of interest to study relations between properties of signals and those of the symmetric function transforms. In particular, for some channels, labelled symmetric function transforms may provide better protection against channel noise than the signal when used directly for transmission, because effects of noise will get redistributed during inverse transformation. Similarly, these alternative equaivalent descriptions may be more amenable to severe quantization, as effects

of quantization noise also will get redistributed during inverse transformation.

8) The notion of Kronecker product of matrices has been extended to Kronecker product of signal flow graphs by Moharir [42]. A similar extension could be thought of for Chinese , Kronecker-Chinese, Chinese-Kronecker, partitioned matrix Kronecker and partitioned matrix Chinese products. This would lead to the definition of a class of new non linear transforms in terms of the known non linear transforms.

APPENDIX - A

HADAMARD ARRAYS

If A_1, A_2, A_3 and A_4 are square submatrices of order t then [39, 43, 67].

$$M = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ -A_2 & A_1 & -A_4 & A_3 \\ -A_3 & A_4 & A_1 & -A_2 \\ -A_4 & -A_3 & A_2 & A_1 \end{bmatrix} \qquad \ldots (A.1)$$

is an orthonormal kernel of order 4t provided

$$M\, M^{*T} = 4t\, I \qquad \ldots (A.2)$$

or equivalently

$$\sum_{p=1}^{4} A_p\, A_p^{*T} = 4t\, I \qquad \ldots (A.3)$$

and

$$A_p A_q^{*T} = A_q A_p^{*T} \quad p \neq q, p, q = 1,2,3,4 \qquad \ldots (A.4)$$

where I is an identity matrix of order 4t.

The matrix M of Eq. (A.1) is given by

$$M = e\, A_1 + i\, A_2 + j\, A_3 + k\, A_4 \qquad \ldots (A.5)$$

where e,i,j and k are square matrices of order four.

They have all the entries drawn from 0,1,-1 and further, the entries 1,-1 appear in them in mutually exclusive and collectively exhaustive locations, i.e., no two of these matrices have entries 1,-1 in the same location and not all of them have an entry zero in any location. The matrices e,i,j and k are isomorphic to the quaterions satisfying the usual conditions and given as

$$e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$j = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$$

$$k = ij = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

The multiplication table for e,i,j and k can be written as

mult.	e	i	j	k
e	e	i	j	k
i	i	e	k	-j
j	j	-k	e	i
k	k	j	-i	e

Since e is a unit matrix

$$e^2 = i^2 = j^2 = k^2 = I_4$$

where I_4 is a unit matrix of order four.

Generalized form of such arrays is an Hadamard array $H[h,k,\lambda]$. It is an h x h array whose entries are $\pm A_1, \pm A_2, \ldots, \pm A_k$, $k \leq h$. $A_1, A_2, \ldots, A_k$ are called indeterminates.

In any row or column of the array there are λ entries $\pm A_1$, λ entries $\pm A_2, \ldots, \lambda$ entries $\pm A_k$.

If $A_1, A_2, \ldots, A_k$ are treated as elements of any commutative ring, then rows as well as columns are pair-wise orthogonal.

Some special cases of Hadamard arrays are:

a) Williamson type

b) Baumert-Hall type

c) Baumert-Hall-Welch type

Williamson type : This is an Hadamard array with H $[h, k, \lambda]$ as $H[4,4,1]$, i.e., each indeterminate $\pm A_i$ is repeated only once in each row and each column, viz.

$$M = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ -A_2 & A_1 & -A_4 & A_3 \\ -A_3 & A_4 & A_1 & -A_2 \\ -A_4 & -A_3 & A_2 & A_1 \end{bmatrix} \qquad \ldots.(A.7)$$

Baumert-Hall type : This is an Hadamard array with $H[h,k,\lambda]$ as $H[12,4,3]$. This type of Hadamard array is also written as BH [12]. This is a generalization of Williamson type and is given in Eq. (A.8).

Baumert-Hall-Welch type : This is an Hadamard array with $H[h,k,\lambda]$ as $H[20,4,5]$ with the condition that the blocks are circulant. This type of Hadamard array is also written as BHW [20] and is given in Eq. (A.9).

$$M_1 = H[12,4,3] = BH[12] = \left[\begin{array}{cccc cccc}
A_1 & A_1 & A_1 & A_2 & -A_2 & A_3 & -A_3 & -A_4 \\
 & & & & A_2 & A_3 & -A_4 & -A_4 \\
A_1 & -A_1 & A_2 & -A_1 & -A_2 & -A_4 & A_4 & -A_3 \\
 & & & & -A_2 & -A_4 & -A_3 & -A_3 \\
A_1 & -A_2 & -A_1 & A_1 & -A_4 & A_4 & -A_2 & A_2 \\
 & & & & -A_3 & -A_4 & A_3 & -A_3 \\
A_2 & A_1 & -A_1 & -A_1 & A_4 & A_4 & A_4 & A_3 \\
 & & & & A_3 & -A_2 & -A_2 & -A_3 \\
A_2 & -A_4 & A_4 & A_4 & A_1 & A_1 & A_1 & A_3 \\
 & & & & -A_3 & A_2 & -A_3 & A_2 \\
A_2 & A_3 & -A_4 & A_4 & A_1 & -A_1 & A_3 & -A_1 \\
 & & & & -A_4 & A_3 & A_2 & -A_2 \\
A_4 & -A_3 & A_2 & -A_2 & A_1 & -A_3 & -A_1 & A_1 \\
 & & & & A_2 & A_3 & A_4 & -A_4 \\
-A_3 & -A_4 & -A_3 & -A_4 & A_3 & A_1 & -A_1 & -A_1 \\
 & & & & -A_4 & A_2 & -A_2 & -A_2 \\
A_4 & -A_3 & -A_2 & -A_2 & -A_2 & A_3 & A_3 & -A_4 \\
 & & & & A_1 & A_1 & A_1 & A_4 \\
-A_4 & -A_2 & A_3 & A_3 & A_3 & A_2 & A_2 & -A_4 \\
 & & & & A_1 & -A_1 & A_4 & -A_1 \\
A_3 & -A_2 & -A_3 & A_3 & A_4 & -A_2 & -A_4 & -A_2 \\
 & & & & A_1 & -A_4 & -A_1 & A_1 \\
-A_3 & -A_4 & -A_4 & A_3 & -A_3 & -A_2 & A_2 & A_2 \\
 & & & & A_4 & A_1 & -A_1 & -A_1
\end{array}\right] \quad \ldots (A.8)$$

$$
M_{2}
= H[20,4,5] = BHW[20] =
\left[\begin{array}{ccccc ccccc ccccc ccccc}
-A_4 & A_2 & -A_3 & -A_3 & -A_2 & A_3 & A_1 & -A_4 & -A_4 & -A_1 & -A_2 & -A_1 & A_3 & -A_3 & -A_1 & A_1 & -A_2 & -A_4 & A_4 & -A_2 \\
-A_2 & -A_4 & A_2 & -A_3 & -A_3 & -A_1 & A_3 & A_1 & -A_4 & -A_4 & -A_1 & -A_2 & -A_1 & A_3 & -A_3 & -A_2 & A_1 & -A_2 & -A_4 & A_4 \\
-A_3 & -A_2 & -A_4 & A_2 & -A_3 & -A_4 & -A_1 & A_3 & A_1 & -A_4 & -A_3 & -A_1 & -A_2 & -A_1 & A_3 & A_4 & -A_2 & A_1 & -A_2 & -A_4 \\
-A_3 & -A_3 & -A_2 & -A_4 & A_2 & -A_4 & -A_4 & -A_1 & A_3 & A_1 & A_3 & -A_3 & -A_1 & -A_2 & -A_1 & -A_4 & A_4 & -A_2 & A_1 & -A_2 \\
A_2 & -A_3 & -A_3 & -A_2 & -A_4 & A_1 & -A_4 & -A_4 & -A_1 & A_3 & -A_1 & A_3 & -A_3 & -A_1 & -A_2 & -A_2 & -A_4 & A_4 & -A_2 & A_1 \\
 & & & & & & & & & & & & & & & & & & & \\
-A_3 & A_1 & A_4 & A_4 & -A_1 & -A_4 & -A_2 & -A_3 & -A_3 & A_2 & -A_1 & A_2 & -A_4 & A_4 & A_2 & -A_2 & -A_1 & -A_3 & A_3 & -A_1 \\
-A_1 & -A_3 & A_1 & A_4 & A_4 & A_2 & -A_4 & -A_2 & -A_3 & -A_3 & A_2 & -A_1 & A_2 & -A_4 & A_4 & -A_1 & -A_2 & -A_1 & -A_3 & A_3 \\
A_4 & -A_1 & -A_3 & A_1 & A_4 & -A_3 & A_2 & -A_4 & -A_2 & -A_3 & A_4 & A_2 & -A_1 & A_2 & -A_4 & A_3 & -A_1 & -A_2 & -A_1 & -A_3 \\
A_4 & A_4 & -A_1 & -A_3 & A_1 & -A_3 & -A_3 & A_2 & -A_4 & -A_2 & -A_4 & A_4 & A_2 & -A_1 & A_2 & -A_3 & A_3 & -A_1 & -A_2 & -A_1 \\
A_1 & A_4 & A_4 & -A_1 & -A_3 & -A_2 & -A_3 & -A_3 & A_2 & -A_4 & A_2 & -A_4 & A_4 & A_2 & -A_1 & -A_1 & -A_3 & A_3 & -A_1 & -A_2
\end{array}\right]
$$

$$
\begin{bmatrix}
A_2 & -A_1 & -A_3 & A_3 & -A_1 & A_1 & A_2 & -A_4 & A_4 & A_2 & -A_4 & -A_2 & A_3 & A_3 & A_2 & -A_3 & A_1 & -A_4 & -A_4 & -A_1 \\
-A_1 & A_2 & -A_1 & -A_3 & A_3 & A_2 & A_1 & A_2 & -A_4 & A_4 & A_2 & -A_4 & -A_2 & A_3 & A_3 & -A_1 & -A_3 & A_1 & -A_4 & -A_4 \\
A_3 & -A_1 & A_2 & -A_1 & -A_3 & A_4 & A_2 & A_1 & A_2 & -A_4 & A_3 & A_2 & -A_4 & -A_2 & A_3 & -A_4 & -A_1 & -A_3 & A_1 & -A_4 \\
-A_3 & A_3 & -A_1 & A_2 & -A_1 & -A_4 & A_4 & A_2 & A_1 & A_2 & A_3 & A_3 & A_2 & -A_4 & -A_2 & -A_4 & -A_4 & -A_1 & -A_3 & A_1 \\
-A_1 & -A_3 & A_3 & -A_1 & A_2 & A_2 & -A_4 & A_4 & A_2 & A_1 & -A_2 & A_3 & A_3 & A_2 & -A_4 & A_1 & -A_4 & -A_4 & -A_1 & -A_3 \\
 & & & & & & & & & & & & & & & & & & & \\
-A_1 & -A_2 & -A_4 & A_4 & -A_2 & A_2 & -A_1 & A_3 & -A_3 & -A_1 & A_3 & A_1 & A_4 & A_4 & -A_1 & -A_4 & A_2 & A_3 & A_3 & -A_2 \\
-A_2 & -A_1 & -A_2 & -A_4 & A_4 & -A_1 & A_2 & -A_1 & A_3 & -A_3 & -A_1 & A_3 & A_1 & A_4 & A_4 & -A_2 & -A_4 & A_2 & A_3 & A_3 \\
A_4 & -A_2 & -A_1 & -A_2 & -A_4 & -A_3 & -A_1 & A_2 & -A_1 & A_3 & A_4 & -A_1 & A_3 & A_1 & A_4 & A_3 & -A_2 & -A_4 & A_2 & A_3 \\
-A_4 & A_4 & -A_2 & -A_1 & -A_2 & A_3 & -A_3 & -A_1 & A_2 & -A_1 & A_4 & A_4 & -A_1 & A_3 & A_1 & A_3 & A_3 & -A_2 & -A_4 & A_2 \\
-A_2 & -A_4 & A_4 & -A_2 & -A_1 & -A_1 & A_3 & -A_3 & -A_1 & A_2 & A_1 & A_4 & A_4 & -A_1 & A_3 & A_2 & A_3 & A_3 & -A_2 & -A_4
\end{bmatrix} \quad \ldots (A.9)
$$

In particular if A_1, A_2, A_3 and A_4 are real and symmetric, complex conjugation and transpose could be dropped from conditions given by Eqs. (A.3) and (A.4). Further if they are circulant, pairwise commutativity, i.e., Eq. (A.4) is automatically satisfied. So if A_1, A_2, A_3 and A_4 are real, symmetric and circulant submatrices of order t then the condition for orthonormality of Hadamard arrays given by Eqs. (A.3) and (A.4) becomes

$$A_1^2 + A_2^2 + A_3^2 + A_4^2 = 4t\ I \qquad \ldots \text{(A.10)}$$

If A_1, A_2, A_3 and A_4 have q^{th} roots of unity as entries then the resulting orthonormal quaternion kernel is represented as $F_q(4t)$, q always being even.

Some important properties of orthonormal kernels of quaternion type are:

1) If M be an orthonormal kernel of the quatirnion type of order $N = 4t$ then M^{-1} is also an orthonormal kernel of the quaternion type.
2) If M be an orthonormal kernel of the quaternion type of order $N = 4t$ then there exist quaternion orthonormal kernels of order $(2^i N)$, $i = 1,2,\ldots$

3) If M be an orthonormal kernel of the quaternion type of order N with A_1, A_2, A_3 and A_4 as submatrices then M' given below would be an orthonormal kernel of the quaternion type of order 2N with A_1', A_2', A_3' and A_4' as submatrices.

$$M' = \begin{bmatrix} A_1' & A_2' & A_3' & A_4' \\ -A_2' & A_1' & -A_4' & A_3' \\ -A_3' & A_4' & A_1' & -A_2' \\ -A_4' & -A_3' & A_2' & A_1' \end{bmatrix} \quad \ldots (A.11)$$

where

$$A_1' = \begin{bmatrix} A_1 & A_2 \\ A_2 & A_1 \end{bmatrix} \quad \ldots (A.12)$$

$$A_2' = \begin{bmatrix} A_1 & -A_2 \\ -A_2 & A_1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} A_3 & A_4 \\ A_4 & A_3 \end{bmatrix}$$

$$A_4' = \begin{bmatrix} A_3 & -A_4 \\ A_4 & A_3 \end{bmatrix}$$

4) If M, M^{-1} of order N be extended to obtain two kernels of order $2N$, they do not form an inverse pair.

5) If K be an orthonormal kernel of order k then submatrices given by

$$A_i' = K \otimes_k A_i \qquad i = 1,2,3,4 \quad \ldots \text{(A.13)}$$

and

$$A_i'' = A_i \otimes_k K \qquad i = 1,2,3,4 \quad \ldots \text{(A.14)}$$

can also be used to form orthonormal kernels.

Hadamard matrices of the Williamson type of order $N = 4t$ can be constructed for many values of t with the help of table listed by Wallis [67]. For the desired N and t, $N = 4t$, values of μ_1, μ_2, μ_3 and μ_4 are taken from the table. Write X_1, X_2, X_3 and X_4 as

$$\begin{aligned} X_1 &= -\mu_1 + \mu_2 + \mu_3 + \mu_4 \\ X_2 &= \mu_1 - \mu_2 + \mu_3 + \mu_4 \\ X_3 &= \mu_1 + \mu_2 - \mu_3 + \mu_4 \\ X_4 &= \mu_1 + \mu_2 + \mu_3 - \mu_4 \end{aligned} \qquad \ldots \text{(A.15)}$$

The quantities μ_1, μ_2, μ_3 and μ_4 are listed in terms of N^{th} roots of unity ω_i. Eq. (A.15) can be re-written as

$$X_1 = 2\,[I + \sum C_{i_1}\,\omega_i]$$
$$X_2 = 2\,[I + \sum C_{i_2}\,\omega_i]$$
$$X_3 = 2\,[I + \sum C_{i_3}\,\omega_i] \qquad \ldots \text{(A.16)}$$
$$X_4 = 2\,[I + \sum C_{i_4}\,\omega_i]$$

where C_{i_1}, C_{i_2}, C_{i_3} and C_{i_4} take values from 0,1,-1. The submatrices A_1, A_2, A_3 and A_4 are given by

$$A_1 = I + C_{i_1}\,S_i$$
$$A_2 = I + C_{i_2}\,S_i$$
$$A_3 = I + C_{i_3}\,S_i \qquad \ldots \text{(A.17)}$$
$$A_4 = I + C_{i_4}\,S_i$$

where

$$S_i = T^i + T^{t-i} \qquad \ldots \text{(A.18)}$$

and T is a square matrix of order t given as

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \qquad \ldots \text{(A.19)}$$

Once A_1, A_2, A_3, A_4 are known one can easily write Hadamard matrix of Williamson type of order $N = 4t$ by substituting them in Eq. (A.7). This procedure ensures A_i's to be real, symmetric and circulant and hence it is easier to verify the orthonormality the resulting Hadamard matrix.

Example Let $t = 5$ so that $N = 4t = 20$

From the table [67]

$$\mu_1 = 1$$

$$\mu_2 = 1$$

$$\mu_3 = 1-2\omega_1$$

$$\mu_4 = 1-2\omega_2$$

This gives

$$X_1 = 2(1-\omega_1-\omega_2)$$

$$X_2 = 2(1-\omega_1-\omega_2)$$

$$X_3 = 2(1+\omega_1-\omega_2)$$

$$X_4 = 2(1-\omega_1+\omega_2)$$

and hence

$$A_1 = I-S_1-S_2$$

$$A_2 = I-S_1-S_2$$

$$A_3 = I+S_1-S_2$$

$$A_4 = I-S_1+S_2$$

$$S_1 = T^1 + T^4 = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 1&0&0&0&0 \end{bmatrix} + \begin{bmatrix} 0&0&0&0&1 \\ 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \end{bmatrix} = \begin{bmatrix} 0&1&0&0&1 \\ 1&0&1&0&0 \\ 0&1&0&1&0 \\ 0&0&1&0&1 \\ 1&0&0&1&0 \end{bmatrix}$$

$$S_2 = T^2 + T^3 = \begin{bmatrix} 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 1&0&0&0&0 \\ 0&1&0&0&0 \end{bmatrix} + \begin{bmatrix} 0&0&0&1&0 \\ 0&0&0&0&1 \\ 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \end{bmatrix} = \begin{bmatrix} 0&0&1&1&0 \\ 0&0&0&1&1 \\ 1&0&0&0&1 \\ 1&1&0&0&0 \\ 0&1&1&0&0 \end{bmatrix}$$

Thus A_1, A_2, A_3 and A_4 can be written as

$$A_1 = A_2 = \begin{bmatrix} 1&-1&-1&-1&-1 \\ -1&1&-1&-1&-1 \\ -1&-1&1&-1&-1 \\ -1&-1&-1&1&-1 \\ -1&-1&-1&-1&1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1&1&-1&-1&1 \\ 1&1&1&-1&-1 \\ -1&1&1&1&-1 \\ -1&-1&1&1&1 \\ 1&-1&-1&1&1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

Substituting these values of A_1, A_2, A_3 and A_4 in Eq. (A.7) results in an Hadamard matrix of the Williamson type of order 20. It cam be easily verified that

$$A_1^2 + A_2^2 + A_3^2 + A_4^2 = 20I$$

and hence the resulting matrix of order 20 would be an orthonormal matrix.

REFERENCES CITED

1. Special Issue on FFT and its Applications, Trans. IEEE, AU-15, 2, June 1967.
2. Ahmed, N., and Rao, K.R., 'Complex BIFORE Transform', Electronic Letters, Vol.6, 256-258, 1970.
3. Ahmed, N., Rao, K.R., and Albussattar, A.L., 'BIFORE or Hadamard Transform', Trans.IEEE, AU-19, 225-234, 1971.
4. Ahmed, N., and Rao, K.R., Orthonormal Transforms for Digital Signal Processing , Springer-Verlag, 1975.
5. Andrews, H.C., Computer Techniques in Image Processing, Academic Press, New York, 1970.
6. Andrews, H.C., and Caspari, K.L., 'A Generalized Technique for Spectral Analysis', Trans. IEEE, C-19, 16-25, Jan. 1970.
7. Andrews, H.C., and Kane, J., 'Kronecker Matrices, Computer Implementation and Generalized Spectra', JACM, Vol-17, 260-268, 1970.
8. Andrews, H.C., and Caspari, K.L., 'Degrees of Freedom and Modular Structure in Matrix Multiplication', Trans. IEEE, C-20, 133-141, Feb. 1971.

9. Arazi, B., 'Hadamard Transforms of Some Specially Shuffled Signals', Trans.IEEE, ASSP-23, 580-583, 1975.

10. Baumert, L.D., Cyclic Difference Sets, Lecture Notes in Mathematics, 182, Springer-Verlag, 1971.

11. Berlekamp, E.R., Algebraic Coding Theory, McGraw-Hill Book Co., 1968.

12. Brewer, John W., 'Kronecker Products and Matrix Calculus in System Theory', Trans.IEEE, CAS-25, 772-781, Sept. 1978.

13. Burnside, W., Theory of Groups of Finite Order, Dover Publications, New York, 1955.

14. Carmichael, R.D., Introduction to the Theory of Groups of Finite Order, Dover Publications, New York, 1956.

15. Caspari, K.L., 'Generalized Spectral Analysis', Applications of Walsh Functions Proc., Editor Bass, C.A., Washington D.C., 195-207, 1970.

16. Cheng, D.K., and Liu, J.J., 'Walsh Transform Analysis of Discrete Dyadic-Invariant Systems', Trans. IEEE, EMC-16, 136-140, May 1974.

17. Cheng., D.K., and Liu, J.J., 'Time-Domain Analysis of Dyadic-Invariant Systems', Proc.IEEE,62, 1038-1040, July 1974.

18. Faddeev, D.,and Sominsky, I., Problems In Higher Algebra, Mir Publishers, 1968.

19. Fine, N.J., 'On the Walsh Functions', Trans.Amer. Math. Soc., 65, 372-414, 1949.

20. Fraleigh, John B., A First Course in Abstract Algebra, Addison-Wesley Publishing Co., 1967.

21. Gentleman, W.M., and Sande, G., 'Fast Fourier Transforms-for Fun and Profit', 1966 Fall Joint Computer Conf., AFIPS Conf. Proc., 29, 563-578, 1966.

22. Gethöffer, H., 'Mutual Mapping of Generalized Convolution Systems', Proc. 1972 Symp. Appl. of Walsh Functions, Washington, 310-317,1972.

23. Gold, B., and Rader, C.M., Digital Processing of Signals, McGraw-Hill Book Co., 1969.

24. Hadamard, J., 'Resolution d'une Question Relative aux Determinants', Bull. Sci, Math., Ser 2, 17, PtI, 240-246, 1893.

25. Hall, M., The Theory of Groups, Macmillan, New York, 1959.

26. Harmuth, H.F., Transmission of Information by Orthogonal Functions, Springer-Verlag, 1972.

27. Kochendörffer, R., Group Theory (English Translation), McGraw-Hill Book Co., 1970.

28. Kurosh, A., Higher Agebra (English Translation), Mir Publishers, 1972.

29. Ledermann, W., Introduction to Group Theory, Oliver and Boyd, Edinburgh, 1973.

30. Levy, P., 'On a Generalization of the Rademacher Orthogonal Functions', Commentarii Mathematica Helvetica (Swiss Mathematical Communications), 16, 146-152, 1944 (English Translation : Applied Physics Laboratory, John Hopkins University, Maryland, U.S.A., Translation Series No. T-2659, 20 July 1972).

31. Mirsky, L., An Introduction to Linear Algebra, Oxford University Press, 1955.

32. Moharir, P.S., 'Translation Invariant Transforms with Uniform Amplitude Bounds for Pattern Recognition', JIETE, 19, 163-164, 1973.

33. Moharir, P.S., 'Modular Permutation Properties of Discrete Fourier Transform Pair', JIETE, 20, 122-123, March-April 1974.

34. Moharir, P.S., 'Two-Dimensional Encoding Marks for Hadamard Spectrometric Imager', Trans. IEEE, EMC-16, 126-130, May 1974.

35. Moharir, P.S., 'Binary Toeplitz-Kronecker Transforms and their Applications', JIETE, 20, 482-487, 1974.

36. Moharir, P.S., 'Permutation Properties of Discrete Walsh Transform Pair', JIETE, 20, 593-597, 1974.

37. Moharir, P.S., 'Block-and Strand-Multiplexing for Fast Computation of a Class of Transforms', Int.J. Electronics, 38, 401-412, 1975.

38. Moharir, P.S., 'Secrecy Coding by Fourier-Kronecker Pruning of Certain Transforms', Int.J. Electronics, 39, 539-543, 1975.

39. Moharir, P.S., 'Generalized Hadamard-Fourier Transforms', JIETE, 22, 123-127, 1976.

40. Moharir, P.S. 'Chinese Product Theorem for Generalized PN Sequences', Electronics Letters, 13, 121-122, 1977.

41. Moharir, P.S., 'Wide-Sense Translinear Systems', JIETE, 24, 351-354, 1978.

42. Moharir, P.S., 'Butterfly, and Symmetric Function Transforms', JIETE, 24, 529-533, 1978.

43. Moharir, P.S., 'New Signal-Processing Transforms', Unpublished Manuscript.

44. Moharir, P.S., 'Self-Dual Codes Based on (v,k, λ) Difference Sets', Unpublished Manuscript.

44. Paley, R.E.A.C., 'A Remarkable Series of Orthogonal Functions', Proc. London Math. Soc., 34, 241-279, 1932.

46. Pichler, F.,'Walsh Function and Linear System Theory', Proc. 1970 Symp. Appl. of Walsh Functions, Washington, 175-182, 1970.

47. Pichler, F., 'On State-Space Description of Linear Dyadic-Invariant Systems', Proc. 1971, Symp. Appl. of Walsh Functions, Washington, 166-170, 1971.

48. Rader, C., 'Disccete Fourier Transform when the Number of Data Samples is Prime', Proc. IEEE, 56, 1107-1108, 1968.

49. Rao, K.R., and Ahmed, N., 'Modified Complex BIFORE Transform', Proc. IEEE,60, 1010-1012, 1972.

50. Rao, K.R. et. al., 'Complex Harr Transform', Trans. IEEE, ASSP-24, 102-104, 1976.

51. Rao, K.R., and Ahmed, N., 'Orthonormal Transforms for Digital Signal Processing', 1976 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Philadelphia, April 12-14, 1976.

52. Rao, P.R.K., Certain Generalizations of Permutation Invariant Systems, Ph.D.Thesis, Electrical Engineering Department, I.I.T. Kanpur, 1978.

53. Reitboeck, H., and Brody, T.P., 'A Transformation with Invariance under Cyclic Permutation for Applications in Pattern Recognition', Information and Control, 15, 130-154, 1969.

54. Rosenbloom, J.H., 'Physical Interpretation of Dyadic Groups', Proc. 1971 Symp. Appl. of Walsh Functions, Washington, 158-165, 1971.

55. Ryser, H.J., Combinatorial Mathematics, The Carus Mathematical Monographs, 14, John Wiley and Sons, Inc., 1963.

56. Schmidt, O.U., Abstract Theory of Groups (English Translation), W.H.Freeman Company, San Francisco, 1966.

57. Siddiqui, M.U., A Study of Permutation Invariant Linear Systems, Ph.D.Thesis, Electrical Engineering Department, I.I.T., Kanpur, 1977.

58. Singleton, R,C., 'An Algorithm for Computing the Mixed Radix Fast Fourier Transform', Trans. IEEE, AU-17, 99-103, 1963.

59. Stockham, T.G., Jr., 'High-Speed Convolution and Correlation', 1966 Spring Joint Computer Conf., AFIPS Conf. Proc. 28, 229-233, 1966.

60. Szabo, N.S., and Tanaka, R.I., Residue Arithmatic and its Application to Computer Technology, McGraw-Hill Book Co., 1967.

61. Ulman, L.J., 'Computation of the Hadamard Transform and the R-Transform in Ordered Form', Trans. IEEE, C-19, 359-360, 1970.

62. Wagh, M.D., 'R-Transform Amplitude Bounds and Transform Volume',JIETE, 21, 501-502, 1975.

63. Wagh, M.D., 'Periodicity in R-Transformation', JIETE, 21, 560-561, 1975.

64. Wagh, M.D., and Kanetkar, S.V., 'A Multiplexing Theorem and Generalization of R-Transform', Int.J. Comp. Maths.(Section A), 5, 163-171, 1975.

65. Wagh, M.D., Translation Invariant Transforms, Ph.D. Thesis, Electrical Engineering Department, I.I.T., Bombay, 1977.

66. Wagh, M.D., 'An Extension of R-Transforms to Patterns of Arbitrary Lengths', Int. J. Comp. Maths.

67. Walis, M.D., Street, A.P., and Wallis, J.S., Combinatorics: Room Squares, Sum-Free Sets, Hadamard Matrices, Lecture Notes in Mathematics, 292, Springer-Verlag, 1972.

68. Walsh, J.L., 'A Closed Set of Normal Orthogonal Functions', Amer. J. Math., 45, 5-24, 1923.

69. Zassenhaus, H., The Theory of Groups, Chelsa, 1958.