2

N74-26084

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-680

RM2: Transform Operations

Robert F. Rice

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

March 1, 1974

JUN 1974
RECEIVED
NASA STI FACILITY
INPUT BRANCH

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-680

RM2: Transform Operations

Robert F. Rice

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

March 1, 1974

Copyright © 1974
Jet Propulsion Laboratory
California Institute of Technology

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

REPRODUCTION RESTRICTIONS OVERRIDDEN
NASA Scientific and Technical Information Facility

REPRODUCTION RESTRICTIONS OVERRIDDEN
NASA Scientific and Technical Information Facility

PREFACE

The work described in this report was performed by the Astrionics Division of the Jet Propulsion Laboratory.

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

ILLUSTRATIONS

TABLES

PRECEDING PAGE BLANK NOT FILMED

ABSTRACT

This report introduces the two-dimensional transform used in the research TV source encoder, RM2. It is shown that both conceptually and in terms of the number of required computations, the RM2 transform is considerably simpler than the Fast Hadamard Transform. The latter can in fact be generated by extending the RM2 transform.

I. INTRODUCTION

This report provides details of the two-dimensional transform used in the research TV source encoder, RM2. For the uninitiated reader, adequate background is provided for the main topic which is carried as far as some implementation considerations.

It is shown that, both conceptually and in terms of the number of required computations, the RM2 transform is considerably simpler than the Fast Hadamard Transform (FHT). In fact, it is demonstrated in the Appendix that the FHT can be generated by extending the RM2 transform.

Previously, the FHT had generally been considered the simplest transform having practical applications to TV source encoding. The demonstration that the simpler RM2 transform can be made quite powerful when combined with other techniques will be the subject of subsequent reports. The reader may consult the extensive bibliography by Wintz[1] to see what others have done using transforms for TV source encoding.

II. BASIC TRANSFORM*

The standard representation for an arbitrary N by N matrix $\tilde{A}$ with elements a_{ij} is given as linear combinations of the N^2 basis vectors

$$\tilde{e}_{ij} = \left[\begin{array}{ccc} \mathbf{0} & \vdots & \mathbf{0} \\ & 0 & \\ \text{--------}0\ 0 & 1 & 0\ 0\text{---} \\ & 0 & \\ & 0 & \\ \mathbf{0} & \vdots & \mathbf{0} \\ & \cdot & \end{array}\right] \begin{array}{c} \\ \\ i \\ \\ \\ \\ \\ \end{array} \tag{1}$$

$$\hphantom{\tilde{e}_{ij} = \quad\quad\quad} j$$

That is

$$\tilde{A} = \sum_{i,j} a_{ij}\, \tilde{e}_{ij} \tag{2}$$

The $\tilde{e}_{ij}$ are said to span the N^2 dimensional space and are called basis vectors. The scalar product for two arbitrary vectors in this space is defined by

$$< \tilde{A},\ \tilde{B} > = \sum_{i,j} a_{ij}\, b_{ij} \tag{3}$$

The norm of a vector is given by

$$\|\tilde{A}\| = (<\tilde{A},\ \tilde{A}>)^{1/2} \tag{4}$$

*Only the most basic abstract algebra is involved here. The reader is referred to the introductory chapters of Ref. 2 for more details.

If $\langle \tilde{A}, \tilde{B} \rangle = 0$, $\tilde{A}$ and $\tilde{B}$ are said to be orthogonal. If in addition $\|\tilde{A}\| = \|\tilde{B}\| = 1$ they are orthonormal. One can easily verify that the $\tilde{e}_{ij}$ form an orthonormal set. Also $\langle \tilde{A}, \tilde{e}_{ij} \rangle = a_{ij}$ so that

$$\tilde{A} = \sum_{i,j} \langle \tilde{A}, \tilde{e}_{ij} \rangle \tilde{e}_{ij} \tag{5}$$

It is a consequence of vector space theory that any set of N^2 orthogonal vectors $\tilde{H}_0, \tilde{H}_1, \ldots \tilde{H}_{N^2-1}$ span the space so that we can write

$$\tilde{A} = \sum_{k} \langle \tilde{A}, \tilde{H}_k \rangle \frac{\tilde{H}_k}{\|\tilde{H}_k\|^2} \tag{6}$$

Thus, we see that (5) is just a special case of (6).

Now turning to the simplest non-trivial case of N = 2, consider the Hadamard set of orthogonal basis vectors[3]

$$\tilde{H}_0 = 1/4 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \tilde{H}_1 = 1/4 \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$\tilde{H}_2 = 1/4 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \tilde{H}_3 = 1/4 \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \tag{7}$$

The reader may verify, using (3) and (4) that $\|\tilde{H}_i\| = 1/2$ so that we can write

$$\tilde{A} = 4 \sum_{k} \langle \tilde{A}, \tilde{H}_k \rangle \tilde{H}_k \tag{8}$$

Now the first term, $\bar{A} = \langle \tilde{A}, \tilde{H}_0 \rangle$, is simply the average of the four elements of the array $\tilde{A}$. Thus, if the other terms were deleted or set to zero, we would have a new $\tilde{A}'$ which has all four elements equal to the original average, $\bar{A}$. Adding back in the deleted terms one by one, we see that these coefficients, $\langle \tilde{A}, \tilde{H}_i \rangle$, tell us how much the array tends towards $\tilde{H}_i$, from the all constant array. For example, suppose $\langle \tilde{A}, \tilde{H}_1 \rangle = 1$, $\langle \tilde{A} \quad \tilde{H}_2 \rangle = \langle \tilde{A}, \tilde{H}_3 \rangle = 0$. Using (7) and (8), we get

$$\tilde{A} = \begin{bmatrix} \bar{A} + 1 & \bar{A} - 1 \\ \bar{A} + 1 & \bar{A} - 1 \end{bmatrix}. \tag{9}$$

These operations can also be interpreted as a linear transformation of the vector $\tilde{A}$ into another vector $\tilde{C}$. First rewrite $\tilde{A}$ as

$$\begin{bmatrix} a_0 & a_1 \\ a_3 & a_2 \end{bmatrix}.$$

Letting T_{22} denote this transformation

$$\tilde{A} = \begin{bmatrix} a_0 & a_1 \\ a_3 & a_2 \end{bmatrix} \xrightarrow{T_{22}} \begin{bmatrix} \langle \tilde{A}, \tilde{H}_0 \rangle & \langle \tilde{A}, \tilde{H}_1 \rangle \\ \langle \tilde{A}, \tilde{H}_3 \rangle & \langle \tilde{A}, \tilde{H}_2 \rangle \end{bmatrix} = \tilde{C} = \begin{bmatrix} c_0 & c_1 \\ c_3 & c_2 \end{bmatrix} \tag{10}$$

In terms of matrix multiplication, we can write

$$\tilde{C} = 1/4 \ \tilde{W} \tilde{A} \tilde{W}. \tag{11}$$

Where $\tilde{W}$ is the 2 by 2 Hadamard matrix

$$\tilde{W} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Noting that $\tilde{W}\tilde{W} = 2\tilde{I}$ where $\tilde{I}$ is the identity matrix, we get the inverse T_{22}^{-1} defined by

$$\tilde{A} = \tilde{W}\tilde{C}\tilde{W}. \tag{12}$$

Thus, other than for a factor of 2^2 (essentially free in a digital system), the inverse requires the same operations.

Let us consider these steps in more detail. Assume we start with 5 registers $X_0, X_1, \ldots, X_4$ with the first four initially loaded with the corresponding coefficients, a_i if transform, c_i if inverse. These steps are enumerated in Table 1. At completion, requiring 8 additions, the four registers contain the desired result, transform or inverse.

The basic transform T_{22} described in Table 1 is really the Fast Hadamard Transform[3] for the simplest case of an array of size 2 by 2. We have discussed it in detail because it forms a fundamental part of what follows. More involved operations will be avoided by repeated usage of T_{22}.

Table 1. T_{22} Operations

Operation	Register Contents
Initialize X_0	α_0
Initialize X_1	α_1
Initialize X_2	α_2
Initialize X_3	α_3
$X_0 + X_3 \rightarrow X_0$	$\alpha_0 + \alpha_3$
$X_1 + X_2 \rightarrow X_4$	$\alpha_1 + \alpha_2$
$X_2 + X_3 \rightarrow X_2$	$\alpha_2 + \alpha_3$
$X_0 + X_4 \rightarrow X_0$	$\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3$
$X_1 + X_3 \rightarrow X_3$	$\alpha_1 + \alpha_3$
$X_0 - 2X_4 \rightarrow X_1$	$\alpha_0 + \alpha_3 - (\alpha_1 + \alpha_2)$
$X_0 - 2X_2 \rightarrow X_4$	$\alpha_0 + \alpha_1 - (\alpha_2 + \alpha_3)$
$X_0 - 2X_3 \rightarrow X_2$	$\alpha_0 + \alpha_2 - (\alpha_1 + \alpha_3)$
$X_4 \rightarrow X_3$	$\alpha_0 + \alpha_1 - (\alpha_2 + \alpha_3)$
Shift X_i right 2 places if transform.	

III. HIGHER DIMENSIONS

Assume that our source data to be coded has been loaded into an array of size 2^J by 2^J which we denote $\tilde{C}_0^J$. The most obvious way to use T_{22} in coding the source data is to apply it separately to each of the 2^{2J-2} two by two subarrays. In this case, we would end up with 2^{2J-2} coefficients of each type c_0, c_1, c_2, and c_3 (see (10)). For the moment, let us assume these are separately placed in four arrays of size 2^{J-1} by 2^{J-1} (in their same relative location) and denoted respectively by $\tilde{C}_i^{J-1}$, i = 0, 1, 2, 3. An example to clarify this is shown below for J = 2 (Fig. 1). The additional notation is self-explanatory.

The coefficient sets $\tilde{C}_1^{J-1}$, $\tilde{C}_2^{J-1}$, $\tilde{C}_3^{J-1}$ represent data sources which are quite similar statistically in terms of their effect on reconstruction. Statistically, they tend to be distributed about zero in a bell-shaped fashion. Their effect on reconstruction is to add back in detail to the constant (equal to the average) two by two arrays represented by the $\tilde{C}_0^{J-1}$ terms. On the other hand, the $\tilde{C}_0^{J-1}$ terms are quite different and so we direct our attention there, leaving $\tilde{C}_1^{J-1}$, $\tilde{C}_2^{J-1}$, and $\tilde{C}_3^{J-1}$ arrays as described.

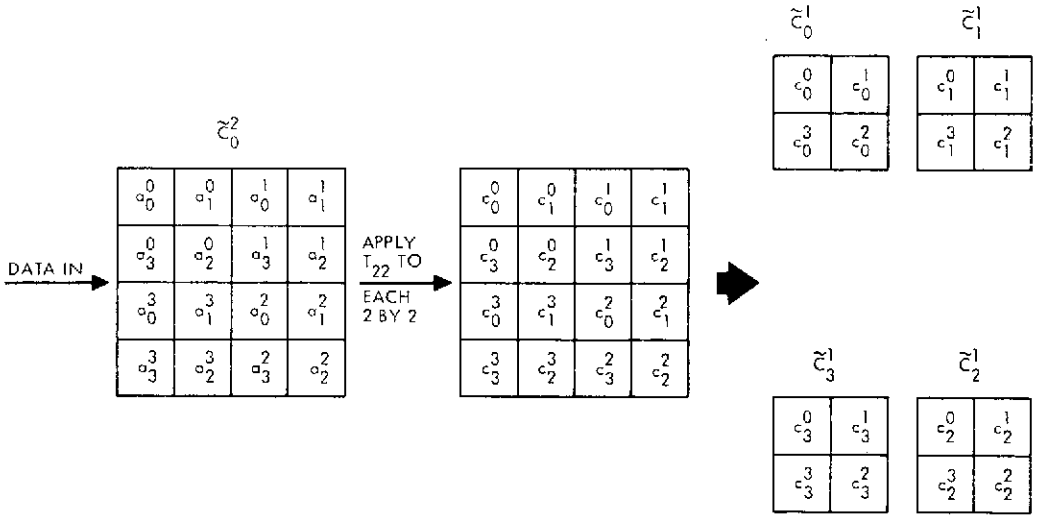

Fig. 1. Illustrative Example

The $\tilde{C}_0^{J-1}$ array contains two by two averages and thus statistically looks much like the original data except at one half the sampling rate in both dimensions. This leads us to apply T_{22} again to all the two by twos in $\tilde{C}_0^{J-1}$. We get four new arrays $\tilde{C}_0^{J-2}$, $\tilde{C}_1^{J-2}$, $\tilde{C}_2^{J-2}$, $\tilde{C}_3^{J-2}$. Again, we can draw the same conclusions, with $\tilde{C}_0^{J-2}$ containing four by four averages of the original data.

We can now generalize. Let T^{J-j} denote the application of T_{22} to all two by twos of $\tilde{C}_0^{J-j}$, $j = 0, 1, \ldots, J-1$. The collection of coefficients c_i, $i = 0, 1, 2, 3$ are placed respectively in arrays $\tilde{C}_0^{J-j-1}$, $\tilde{C}_1^{J-j-1}$, $\tilde{C}_2^{J-j-1}$, and $\tilde{C}_3^{J-j-1}$ all of dimension 2^{J-j-1} by 2^{J-j-1}. Applying T^{J-j} successively, $j = 0, 1, \ldots, J-1$ $\tilde{C}_0^{J-j-1}$ will contain the average values of each 2^{j+1} by 2^{j+1} subarray of the original data array $\tilde{C}_0^{J}$. At termination, $j = J-1$, $\tilde{C}_0^{0}$ is a single term equal to the average of all elements of the original array. This process is best described by the diagram in Fig. 2.

Number of Computations

We are interested in the total number of additions or subtractions required to generate the arrays of Fig. 2. In Table 1, we showed that each use of T_{22} requires eight and thus we need only determine the number of times T_{22} is used by T^J, T^{J-1}, ..., T^1 (i.e., the number of two by two's in each ... of the $\tilde{C}_0^k$, $k = 1, 2, \ldots, J$). Adding these terms, we have $2^{2J}\{1/4 + (1/4)^2 + \ldots (1/4)^J\}$ for a total number of calculations of

$$n_T = (8/3)\left\{2^{2J} - 1\right\} . \qquad (13)$$

This compares with 2^{4J} direct calculations for the Hadamard transformation or $(2J)2^{2J}$ using the Fast Hadamard approach. For 64 by 64 arrays this means the new transform requires approximately 1/4 the number of calculations as the Fast Hadamard.

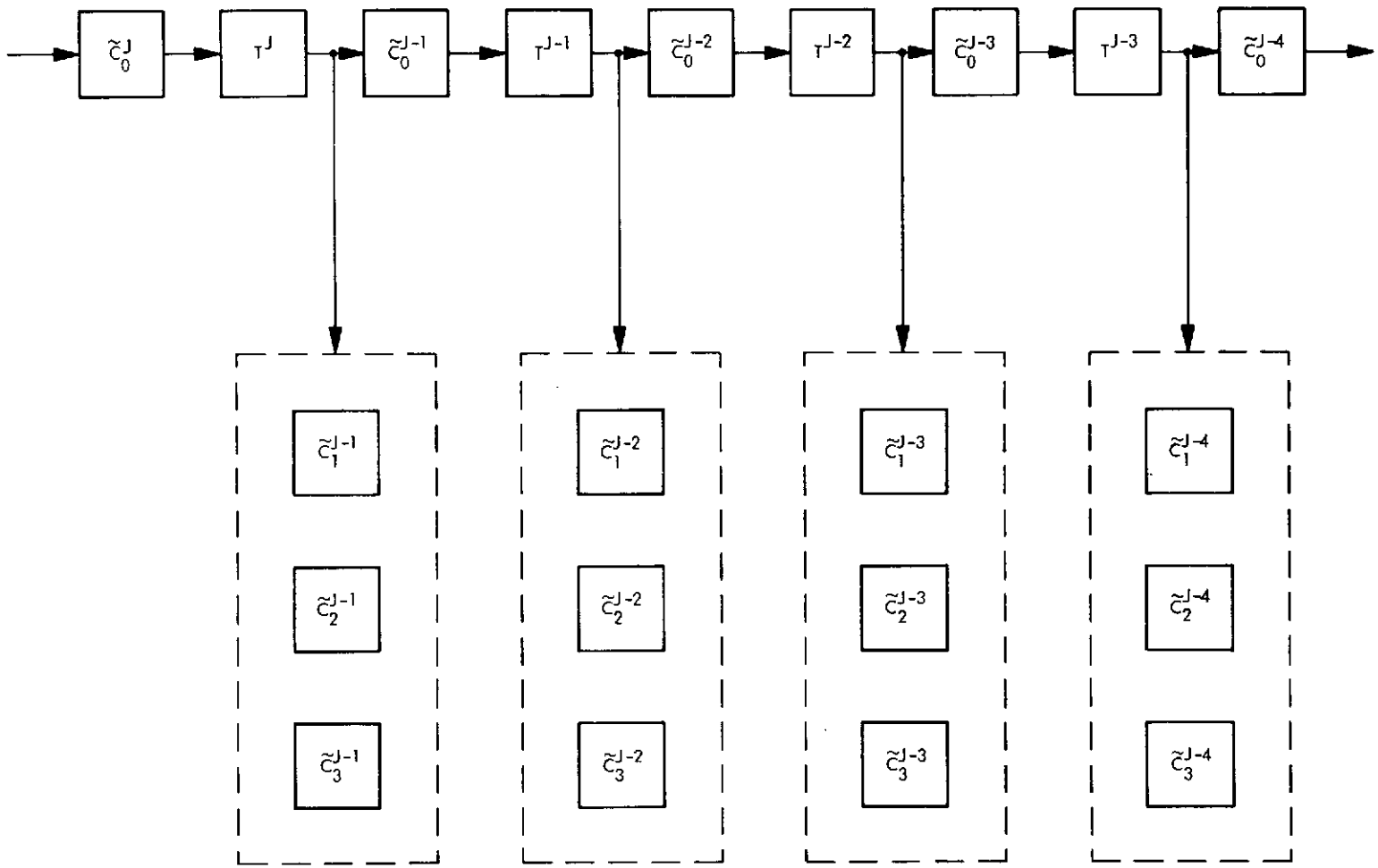

Fig. 2. Transform Block Diagram

Basis Vectors

The 2^J by 2^J input vector $\tilde{A}$ entered into $\tilde{C}_0^J$ in Fig. 2 can be represented in terms of an expansion of the form (6). In fact, the expansion of Fig. 2 actually generates the 2^{2J} coefficients, $<\tilde{A}, \tilde{H}_k>$, although we have not explicitly exhibited these basis vectors, $\tilde{H}_k$. This topic may prove to be of interest to some and leads to an interesting relationship with the Fast Hadamard transform. However, in order to avoid introducing unnecessary complications into the main thrust of this paper, we defer this subject to the Appendix.

IV. ADDRESSING

The following discussion should be particularly useful in software implementations of Fig. 2 using higher level languages such as Fortran (and the Fast Hadamard in the Appendix).

Assume that array $\tilde{C}_0^J$ is a 2^J by 2^J random access memory initially loaded with source data in the same pattern as it appears visually. The standard two dimensional address of any element a_{ij} is, of course, defined by the row number i and the column number j.

Now the arrays $\tilde{C}_i^k$ correspond to 2^{2k} memory locations. The operation T^k on $\tilde{C}_0^k$ produces four new arrays $\tilde{C}_0^{k-1}$, $\tilde{C}_1^{k-1}$, $\tilde{C}_2^{k-1}$, and $\tilde{C}_3^{k-1}$, each requiring $2^{2(k-1)}$ memory locations. Since $\tilde{C}_0^k$ is no longer required, no new memory is actually required. Since this is true for each k, we can, by suitable addressing, use the input memory $\tilde{C}_0^J$ for all coefficients generated.

Consistent with our labeling of coefficients, we label the quadrants of any square array by the arrangement:

$$\begin{bmatrix} 0 & 1 \\ 3 & 2 \end{bmatrix} \tag{14}$$

Clearly, the location of an element a_{ij} which lies in $\tilde{C}_0^J$ is uniquely specified by giving J quadrant numbers (e.g., which quarter of the 2^J by 2^J array, which quarter of the 2^{J-1} by 2^{J-1} subarray and so on). We need only to relate these quadrant numbers to row and column numbers i and j.

Let k_ℓ, $\ell = 1, 2, \ldots, J$ denote the quadrant of the subarray of size 2^ℓ by 2^ℓ in which a_{ij} resides. Define

$$U(X) = \begin{cases} 1 \text{ if } X \bmod 4 > 1 \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

Then we can write the row and column addresses

$$i = 1 + \sum_{\ell=1}^{J} 2^{\ell-1} U(k_\ell)$$

$$j = 1 + \sum_{\ell=1}^{J} 2^{\ell-1} U(k_\ell + 1) \tag{16}$$

The first application of T^J in Fig. 2 requires that for each value of k_2, ..., k_J we apply T_{22} to the four numbers obtained by setting k_1 equal to 0, 1, 2, and 3. The results are returned to the corresponding locations so that the $2^{2(J-1)}$ coefficients of $\tilde{C}^{J-1}_{k_1}$ reside in all i, j locations satisfying (16) with k_1 fixed at 0, 1, 2, or 3.

The application of T^{J-1} to $\tilde{C}^{J-1}_0$ requires that we fix $k_1 = 0$ and then for each value of k_3, k_4, ..., k_J we apply T_{22} to the four numbers obtained by setting $k_2 = 0$, 1, 2, and 3. At completion, the $2^{2(J-2)}$ coefficients of $\tilde{C}^{J-2}_{k_2}$ reside in all i, j locations satisfying (16) with k_2 fixed at 0, 1, 2, or 3 and $k_1 = 0$.

In general, the application of $T^{J-\ell}$ to $\tilde{C}^{J-\ell}_0$ requires that we fix $k_1 = k_2 = \ldots = k_\ell = 0$ and then for each value of $k_{\ell+2}$, ..., k_J we apply T_{22} to the four numbers obtained by setting $k_{\ell+1} = 0$, 1, 2, and 3. We get the $2^{2(J-\ell-1)}$ coefficients of $\tilde{C}^{J-\ell-1}_{k_{\ell+1}}$ located in all i, j locations satisfying (16) with $k_{\ell+1}$ fixed at 0, 1, 2, or 3 and $k_1 = k_2 = \ldots = k_\ell = 0$.

Clearly, we can at any time retrieve desired subsets of coefficients for coding purposes, simply by observing these constraints imposed on the $\{k_\ell\}$ and on the i, j through (16).

V. SHIFT REGISTER IMPLEMENTATION

Figure 3 exhibits another way of looking at the construction of Fig. 2 utilizing shift registers.

Starting at the leftmost part of Fig. 3, input samples of $\tilde{C}_0^J$ are clocked into the processor T^J line by line at some sample rate t_s.* The first line, and subsequent odd-numbered lines, enter the 2^J sample shift register SR_J. These samples are combined in pairs with the corresponding samples from the following even-numbered lines to form two by two input arrays which are then processed by T_{22}. Clearly, each pair of odd and even lines generates one line of the arrays $\tilde{C}_j^{J-1}$, j = 0, 1, 2, 3. In particular, the lines of $\tilde{C}_{0,}^{J-1}$ sequentially enter processor T^{J-1} which performs functionally in the same manner as T^J to produce the arrays $\tilde{C}_j^{J-2}$, j = 0, 1, 2, 3. However, the input sample rate to T^{J-1} has been quartered and the shift register storage reduced in half.

Generalizing, odd lines of $\tilde{C}_0^{J-k}$ enter T^{J-k} at a sample rate $2^{-2k} \cdot t_s$. These samples are temporarily stored in a 2^{J-k} sample shift register, SR_{J-k}. Samples from SR_{J-k} are combined in pairs with the corresponding samples from the following even lines to generate two by two arrays. T_{22} is applied to each two by two array to generate, line by line, the coefficients of arrays $\tilde{C}_j^{J-k-1}$, j = 0, 1, 2, 3.

Observe that the generation of all coefficients of all arrays is completed almost simultaneously with the entry of the last data sample into T^J. The delay is approximately the time required for J uses of T_{22}.

---

*The same arguments hold with a column by column structure.

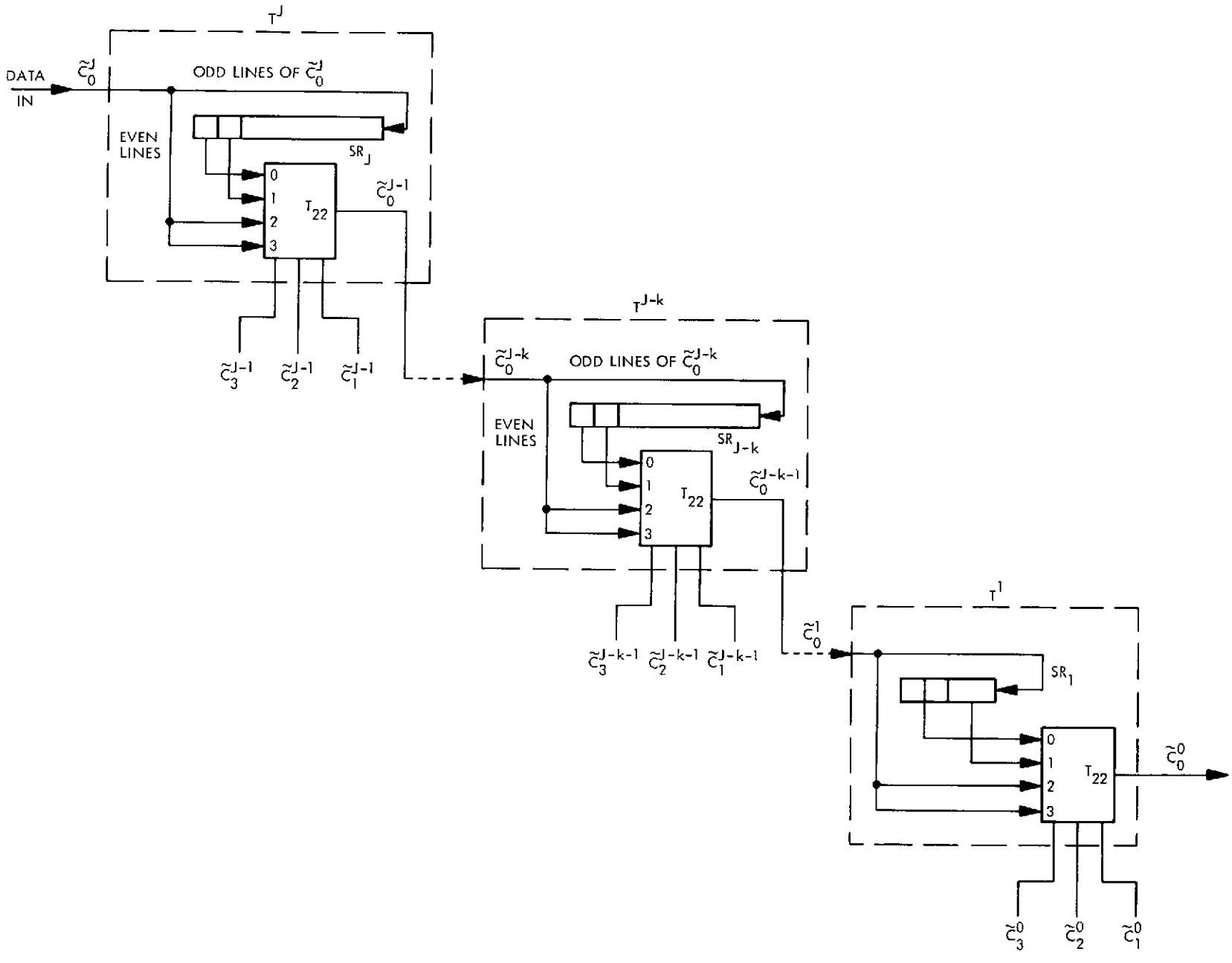

Fig. 3. A Shift Register Implementation

APPENDIX

MORE ON THE TRANSFORM

BASIS VECTORS

We start out by labeling the coefficients in the arrays of Fig. 2. In particular, for the $2^{2\ell}$ elements of $\tilde{C}_j^{\ell}$ number them row by row using the additional notation $c_j^{\ell}(k)$ to indicate the kth coefficient of $\tilde{C}_j^{\ell}$. Here k = 1, 2, ..., 2^{ℓ} in the first row, k = 2^{ℓ} + 1, 2^{ℓ} + 2, ..., $2^{\ell+1}$ in the second row and so on. Similarly, we label the $2^{2\ell}$ subarrays of size $2^{J-\ell}$ by $2^{J-\ell}$ residing in an array of size 2^J by 2^J (e.g., $\tilde{C}_0^J$). This labeling establishes a one to one relationship between any coefficient of $\tilde{C}_j^{\ell}$ and the area over which it has influence in the input array $\tilde{C}_0^J$. It is this one to one relationship which is important, the particular labeling was chosen for convenience. An example is shown below for an input array of size 2^6 by 2^6, and subarrays of size 2^3 by 2^3.

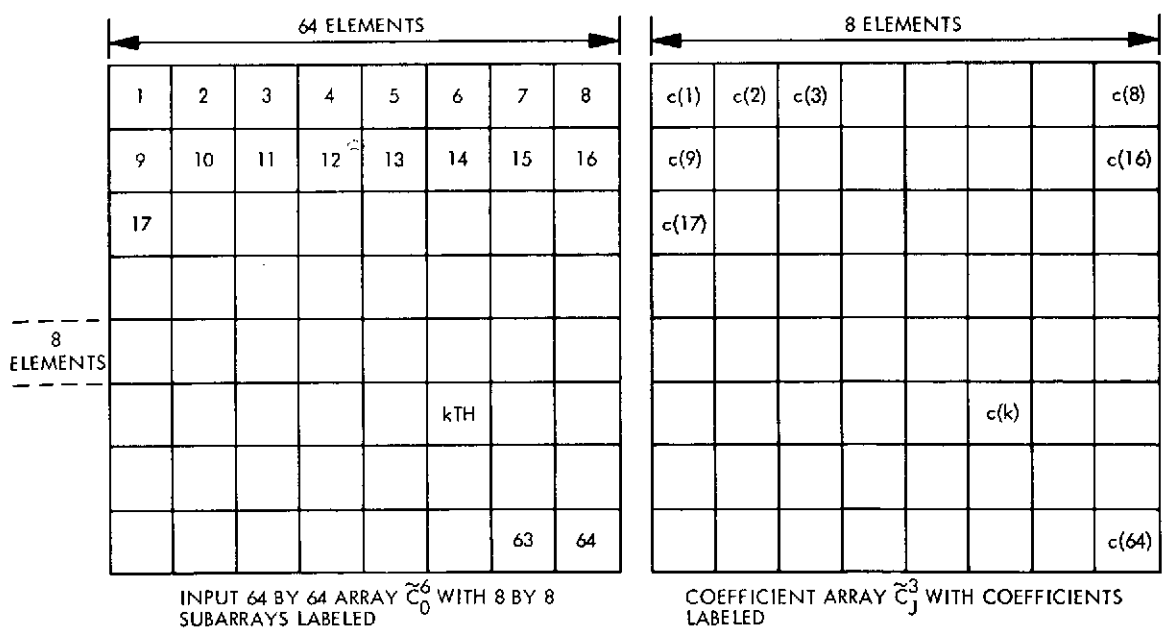

Fig. A-1. Coefficient and Subarray Labeling

Let S_0^J denote the 2^{2J} dimensional vector space spanned by the standard basis vectors $\tilde{e}_{ij}$ in (1). Now let S_0^{J-1} be the 2^{2J-2} dimensional subspace of S_0^J spanned by the orthogonal vectors

$$\tilde{V}_0^{J-1}(1),\ \tilde{V}_0^{J-1}(2),\ \ldots\ ,\ \tilde{V}_0^{J-1}(2^{2J-2}). \tag{A-1}$$

where $4 \cdot \tilde{V}_0^{J-1}(k)$ is a 2^J by 2^J array containing 1's in all four positions of the kth two by two and zeroes elsewhere. For example, $\tilde{V}_0^{J-1}(2)$ is given by

$$\tilde{V}_0^{J-1}(2) = 1/4 \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & \\ \vdots & \vdots & & & & & & \vdots & \vdots \\ 0 & 0 & \cdots & & & & & 0 & 0 \\ 0 & 0 & \cdots & & & & & 0 & 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \end{matrix} 2^J \qquad \underbrace{\qquad\qquad}_{2^J} \tag{A-2}$$

We can similarly define subspaces S_j^{J-1}, $j = 1, 2, 3$ spanned by the orthogonal vector sets $\left\{ \tilde{V}_j^{J-1}(k) \right\}$ where $4 \cdot \tilde{V}_1^{J-1}(k)$, $4 \cdot \tilde{V}_2^{J-1}(k)$ and $4 \cdot \tilde{V}_3^{J-1}(k)$ are 2^J by 2^J arrays with

$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

respectively in the kth two by two and zeroes elsewhere. We note that all vectors are orthogonal with a norm equal to 1/2.

If we now take the scalar product of these vectors with the input vector $\tilde{A}$ we find

$$\langle \tilde{A},\ \tilde{V}_j^{J-1}(k) \rangle = c_j^{J-1}(k) \tag{A-3}$$

Therefore, we can write

$$\tilde{A} = 4 \sum_{j=0}^{3} \left\{ \sum_{k} c_j^{J-1}(k)\, \tilde{V}_j^{J-1}(k) \right\} \tag{A-4}$$

Now specifically consider the subspace S_0^{J-1}. The projection of $\tilde{A}$ onto this subspace is simply the j = 0 terms in (A-4). We wish to represent this subspace in terms of a new set of basis vectors. Let S_0^{J-2} be the 2^{2J-4} dimensional subspace of S_0^{J-1} spanned by the orthogonal vectors

$$\tilde{V}_0^{J-2}(1),\ \tilde{V}_0^{J-2}(2),\ \ldots\ ,\ \tilde{V}_0^{J-2}(2^{2J-4}). \tag{A-5}$$

where $16 \cdot \tilde{V}_0^{J-2}(k)$ is a 2^J by 2^J array containing 1's in all 16 positions of the kth four by four and zeroes elsewhere. Similarly, we define subspaces S_j^{J-2}, j = 1, 2, 3 spanned by the orthogonal vector sets $\left\{ \tilde{V}_j^{J-2}(k) \right\}$ where $16 \cdot \tilde{V}_1^{J-2}(k)$, $16 \cdot \tilde{V}_2^{J-2}(k)$ and $16 \cdot \tilde{V}_3^{J-2}(k)$ are 2^J by 2^J arrays with

$$\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \tag{A-6}$$

respectively in the kth four by four and zeroes elsewhere. We note that $\left\| \tilde{V}_j^{J-2}(k) \right\| = 1/4$ and $\langle \tilde{A},\ \tilde{V}_j^{J-2}(k) \rangle = c_j^{J-2}(k)$. Therefore, the projection of $\tilde{A}$ on S_0^{J-1} in (A-4) can be replaced by

$$16 \sum_{j=0}^{3} \sum_{k} c_j^{J-2}(k)\, \tilde{V}_j^{J-2}(k) \tag{A-7}$$

The pattern should be established now. We continue to break up each orthogonal subspace $S_0^{J-\ell}$ into four new orthogonal subspaces $S_0^{J-\ell-1}$, $S_1^{J-\ell-1}$, $S_2^{J-\ell-1}$, and $S_3^{J-\ell-1}$ until $S_0^{J-\ell}$ contains only one vector, $\tilde{V}_0^0(1)$, where $2^{2J}\tilde{V}_0^0(1)$ is the all 1's array. Then our original input array $\tilde{A}$ can be written as

$$\tilde{A} = 2^{2J}\, c_0^0(1)\, \tilde{V}_0^0(1) + \sum_{j=1}^{3}\sum_{\ell=1}^{J} (4)^{\ell} \left\{ \sum_{k_\ell} c_j^{J-\ell}(k_\ell)\, \tilde{V}_j^{J-\ell}(k_\ell) \right\} \tag{A-8}$$

A RELATIONSHIP WITH HADAMARD

In this section we demonstrate that the RM2 transform, using T_{22}, can be extended to generate the FHT.

Hadamard Transform [3]

First consider the standard Hadamard matrix of size two by two, given earlier as

$$\tilde{H}^1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{A-9}$$

In general, given the 2^m by 2^m Hadamard matrix $\tilde{H}^m$, we can get $\tilde{H}^{m+1}$ by the construction

$$\tilde{H}^{m+1} = \left[\begin{array}{c|c} \tilde{H}^m & \tilde{H}^m \\ \hline \tilde{H}^m & -\tilde{H}^m \end{array}\right] \tag{A-10}$$

The 2^m rows (also the columns) of $\tilde{H}^m$ contain only +1's and -1's. The number of sign changes along a row (or column) is usually called "sequency". Each row has a different sequency number ranging from 0 to 2^{m-1}. We can, therefore, conveniently use this sequency number to label the 2^m distinct row (and column) vectors thus generated. Let $r_y^m(i_y)$ denote a column vector equal to the column of $\tilde{H}^m$ with sequency i_y. Similarly, let $r_x^m(i_x)$ denote a row vector equal to the row of $\tilde{H}^m$ with sequency i_x.

The 2^{2m} Hadamard basis vectors (2^m by 2^m arrays) are formed from the dyadic products

$$\tilde{h}^m(i_x, i_y) = r_y^m(i_y)\, r_x^m(i_x) = \left[r_y^x(i_y)\right] \left[r_x^m(i_x) \longrightarrow\right] \qquad (A\text{-}11)$$

(the row factor spans 2^m columns)

The reader may verify that

$$\begin{aligned} \tilde{h}^1(0,0) &\longleftrightarrow 4\,\tilde{H}_0 \\ \tilde{h}^1(1,0) &\longleftrightarrow 4\,\tilde{H}_1 \\ \tilde{h}^1(1,1) &\longleftrightarrow 4\,\tilde{H}_2 \\ \tilde{h}^1(0,1) &\longleftrightarrow 4\,\tilde{H}_3 \end{aligned} \qquad (A\text{-}12)$$

where the $\tilde{H}_k$ were given in (7). The $\tilde{h}^m(i_x, i_y)$ can be made orthonormal by dividing by 2^{2m-1}.

Returning to the construction in (A-10), we note that if $r_y^m(i_y)$ is a column of $\tilde{H}^m$, then both

$$\begin{matrix} 2^m \\ \\ 2^m \end{matrix} \left[\begin{array}{c} r_y^m(i_y) \\ \hline r_y^m(i_y) \end{array}\right] \quad \text{and} \quad \left[\begin{array}{c} r_y^m(i_y) \\ \hline -r_y^m(i_y) \end{array}\right] \qquad (A\text{-}13)$$

are column vectors of $\tilde{H}^{m+1}$. A similar conclusion holds for the rows. Thus, if $\tilde{X} = \tilde{h}^m(i_x, i_y)$ is a Hadamard basis vector for the space of 2^m by 2^m matrices, then

$$\begin{bmatrix} \tilde{X} & \tilde{X} \\ \tilde{X} & \tilde{X} \end{bmatrix}, \begin{bmatrix} \tilde{X} & -\tilde{X} \\ \tilde{X} & -\tilde{X} \end{bmatrix}, \begin{bmatrix} \tilde{X} & -\tilde{X} \\ -\tilde{X} & \tilde{X} \end{bmatrix} \text{ and } \begin{bmatrix} \tilde{X} & \tilde{X} \\ -\tilde{X} & -\tilde{X} \end{bmatrix} \qquad (A\text{-}14)$$

are Hadamard basis vectors in the space of 2^{m+1} by 2^{m+1} matrices.

The Fast Hadamard from T_{22}

Define $E^{J-\ell}(k_\ell)$, $k_\ell = 1, 2, \ldots, 2^{2(J-\ell)}$ as the subspace of all linear combinations of $\tilde{e}_{ij}$ which have components in the k_ℓ^{th} subarray of size 2^ℓ by 2^ℓ.

For the simplest case, divide S_0^J into $2^{2(J-1)}$ orthogonal subspaces, each of dimension 4,

$$\begin{aligned} &S_0^J \cap E^{J-1}(1) \\ &S_0^J \cap E^{J-1}(2) \\ &\vdots \\ &S_0^J \cap E^{J-1}(2^{2(J-1)}) \end{aligned} \qquad \text{(A-15)}$$

Without loss of generality, choose $S_0^J \cap E^{J-1}(1)$. This subspace consists of four 2^J by 2^J arrays $\tilde{u}_0$, $\tilde{u}_1$, $\tilde{u}_2$, and $\tilde{u}_3$ corresponding to $\tilde{e}_{11}$, $\tilde{e}_{12}$, $\tilde{e}_{22}$, $\tilde{e}_{21}$ respectively. We can represent these vectors by

$$\begin{aligned} \tilde{v}_0 &= 1/2\ (\tilde{u}_0 + \tilde{u}_1 + \tilde{u}_2 + \tilde{u}_3) \\ \tilde{v}_1 &= 1/2\ (\tilde{u}_0 + \tilde{u}_3 - \tilde{u}_1 - \tilde{u}_2) \\ \tilde{v}_2 &= 1/2\ (\tilde{u}_0 + \tilde{u}_2 - \tilde{u}_1 - \tilde{u}_3) \\ \tilde{v}_3 &= 1/2\ (\tilde{u}_0 + \tilde{u}_1 - \tilde{u}_2 - \tilde{u}_3) \end{aligned} \qquad \text{(A-16)}$$

If we wish any scalar product $\langle \tilde{A}, \tilde{v}_i \rangle$ we have from (A-16):

$$\langle \tilde{A}, \tilde{v}_k \rangle = \frac{1}{2} \cdot \left[\langle \tilde{A}, \tilde{u}_0 \rangle \pm \langle \tilde{A}, \tilde{u}_1 \rangle \pm \langle \tilde{A}, \tilde{u}_2 \rangle \pm \langle \tilde{A}, \tilde{u}_3 \rangle \right] \quad \text{(A-17)}$$

or more simply, we apply $2 \cdot T_{22}$ using Table 1.

From previous discussion, we see that the four non-zero positions of $\tilde{v}_0$, $\tilde{v}_1$, $\tilde{v}_2$, and $\tilde{v}_3$ are the normalized Hadamard basis vectors for two by twos. Clearly, this is true for all the subspaces in (A-15).

Now form four orthogonal subspaces:

$$\begin{matrix} S_0\, S_0^J \\ S_1\, S_0^J \\ S_2\, S_0^J \\ S_3\, S_0^J \end{matrix} \quad \text{(A-18)}$$

where $S_{j_1} S_0^J$ consists of all linear combinations of vectors of the form $\tilde{v}_{j_1}$ generated in (A-16). We recognize these as S_0^{J-1}, S_1^{J-1}, S_2^{J-1}, and S_3^{J-1} respectively by our earlier notation.

Now divide each such subspace into $2^{2(J-2)}$ orthogonal subspaces:

$$\left(S_{j_1} S_0^J \right) \cap E^{J-2}(k_2) \qquad \begin{matrix} k_2 = 1, 2, \ldots, 2^{2(J-2)} \\ j_1 = 0, 1, 2, 3 \end{matrix} \quad \text{(A-19)}$$

Each subspace contains four orthogonal vectors, each with the same two by two Hadamard basis vector in its non-zero positions (which are located respectively in the four quadrants of the k_2^{th} four by four).

Using $\tilde{u}_0$, $\tilde{u}_1$, $\tilde{u}_2$, and $\tilde{u}_3$ to represent these four vectors, we can again use (A-16) to generate four new vectors which span this subspace. In particular, let $\tilde{X}_{j_1}$ be the two by two Hadamard basis vector corresponding to each subspace

$$\left(S_{j_1} S_0^J\right) \cap E^{J-2}(k_2).$$

Applying (A-16) to each subspace, we get 4 new orthogonal vectors whose non-zero positions are (except for a normalizing factor) given in (A-14). Thus, we have generated four valid Hadamard basis vectors of size four by four. Now for a given k_2 (i.e., a given 4 by 4), the four Hadamard basis vectors generated must be distinct for each j_1 since otherwise the subspaces

$$\left(S_{j_1} S_0^J\right) \cap E^{J-2}(k_2), \quad \left(S_\ell S_0^J\right) \cap E^{J-2}(k_2), \ell \neq j_1$$

could not be orthogonal. Therefore, this process has generated all of the 2^4 Hadamard basis vectors for each k_2. The scalar product of the input vector, $\tilde{A}$, with each of the vectors generated for a fixed j_1 and k_2 is again obtained simply by a single application of $2 \cdot T_{22}$.

Now form the 2^4 subspaces

$$S_{j_2} S_{j_1} S_0^J \qquad j_1, j_2 = 0, 1, 2, 3. \tag{A-20}$$

where $S_{j_2} S_{j_1} S_0^J$ consists of all linear combinations of the $2^{2(J-2)}$ vectors formed using $\tilde{v}_{j_2}$ in (A-16) on the subspace $S_{j_1} S_0^J$. Let $\tilde{X}_{j_2 j_1}$ denote the particular normalized Hadamard basis vector corresponding to non-zero locations of these vectors.

The pattern should be clear now. After the mth step, we have the 2^{2m} orthogonal subspaces

$$S_{j_m} S_{j_{m-1}} \cdot \cdot \cdot \cdot S_{j_1} S_0^J . \tag{A-21}$$

Where $S_{j_m} \ldots S_0^J$ consists of all linear combinations of 2^J by 2^J arrays, which have Hadamard basis vector $\tilde{X}_{j_m j_{m-1} \cdots j_1}$ in each of the $k_m = 1, 2, .. 2^{2(J-m)}$ subarrays of size 2^m by 2^m, and zeroes elsewhere.

We form the $2^{2(J-m-1)}$ orthogonal subspaces of dimension 4

$$\left(S_{j_m} \cdot \cdot \cdot \cdot S_{j_1} S_0^J\right) \cap E^{J-m-1}(k_{m+1}) \tag{A-22}$$

$$k_{m+1} = 1, 2, \ldots, 2^{2(J-m-1)}.$$

We then represent each such subspace by four new vectors using (A-16). Again by (A-14) the non-zero positions of the new vectors are valid Hadamard basis vectors of dimension 2^{m+1} by 2^{m+1} and denoted by $\tilde{X}_{j_{m+1} j_m \cdots j_1}$. Fixing k_{m+1}, this process must produce 4 distinct vectors for each value of j_m, j_{m-1}, $\ldots$, j_1 since otherwise the subspaces $S_{j_m} S_{j_{m-1}} \ldots S_0^J$ could not be orthogonal. Thus, we have generated all the $2^{2(m+1)}$ orthonormal Hadamard basis vectors. The process continues until m = J.

To obtain the scalar product of the input vector $\tilde{A}$ with these 2^{2J} Hadamard basis vectors requires one more application of $2 \cdot T_{22}$ for each quadruple of vectors generated. We can now add up the total number of applications of T_{22}.

In the first expansion, there are $2^{2(J-1)}$ subspaces requiring a single application of $2 \cdot T_{22}$ each. On the second step, we have $2^{2(J-2)}$ subspaces

for each of 2^2 Hadamard basis vectors. In general, we have $2^{2(J-\ell)}$ subspaces for each of $2^{2\cdot\ell}$ Hadamard basis vectors. The process continues until $\ell = J-1$. Adding all these together and multiplying by the number of computations per T_{22} use (see Table 1), we get:

$$\left[\begin{matrix}\text{No. of}\\ \text{computations}\end{matrix}\right] = 2^3 \left[1 \cdot 2^{2(J-1)} + 2^{2\cdot 1} \cdot 2^{2(J-2)} + \ldots + 2^{2(J-1)} \cdot 1\right]$$

$$= (\log 2^{2J})\, 2^{2J} . \qquad \text{(A-23)}$$

We recognize the latter figure as the number of computations required to generate the Fast Hadamard Transform.

REFERENCES

1. P. A. Wintz, "Transform Picture Coding", Proceedings of the IEEE, Vol. 60, No. 7, pp. 809-820, July 1972.

2. Timothy and Bona, "State Space Analysis: An Introduction", McGraw-Hill, Inc., 1968.

3. Pratt, Kane and Andrews, "Hadamard Transform Image Coding", Proceedings of the IEEE, Vol. 57, No. 1, pp. 58-68, January 1969.