ORTHOGONAL TRANSFORMS IN DIGITAL IMAGE CODING

BY

LO KWOK TUNG

（盧 國 棟）

A Master Thesis

Submitted in partial fulfilment of the requirements

for the degree of

Master of Philosophy

in

Department of Electronics

The Chinese University of Hong Kong

May 1989

To the innocent civilians and students

who lost their lives in Beijing

- Democracy is our future -

ABSTRACT

In the past two decades, with the continuing growth of the modern communication and computer technologies, the need for transmission and storage of image data is rapidly increasing. The amount of data required to represent a digitized picture is very large. However, the bandwidth of a communication channel is usually limited, the need for compression of the image data is therefore obvious. Various methods have been proposed to compress image data to as low bit-rate as possible, and transform coding is found to be one of the most efficient techniques. By using transform coding, a picture can be compressed to as low as 0.5 bit per pixel.

In practice, a transform is chosen subject to two opposing criteria, the compression ability and the computation effectiveness. In this thesis, a method to generate new orthogonal transforms based on the principle of dyadic symmetry is presented. Several transforms which have simpler structure but whose compression ability for image data is close to that of the Discrete Cosine Transform (DCT) have been proposed. On the other hand, an enhacement to the DCT, called Odd Weighted Cosine Transform (OWCT), is developed. However, the transformation process is time consuming since it requires a lot of computation efforts. Therefore, the development of fast computational algorithms and dedicated hardware structures or integrated circuits for those orthogonal transforms is necessary. In this thesis, a new fast recursive algorithm is developed for computation of all types of the radix-2 Discrete W Transform (DWT). A modified structure for implementation of the Integer Cosine Transform (ICT) is also described and a new ICT processor is developed by using the ASIC (Application Specific Integrated Circuit) technology.

ACKNOWLEDGEMENTS

It is my pleasure to express my gratitude to my supervisor, Dr. W.K. Cham, for his most stimulating guidance, valuable advice and encouragement throughout the course of this research work, and his patient reading and useful comments in the preparation of this thesis and other manuscripts. Thanks are also due to Dr. C.S. Choy for his helpful discussions and criticisms.

Finally, I wish to express my deepest gratitude to my parent for their moral and financial support throughout the course of my whole education.

LIST OF PRINCIPLE SYMBOLS AND ABBREVIATIONS

σ^2 : Variance

$\oplus$: Exclusive OR

$| \ |$: Norm of a Vector

ρ : Adjacent Element Correlation

$[C_x]$: Covariance Matrix of Vector X

D : Mean Square Distortion

E[ ] : Expected Value of the variable in [ ]

F : Number Field

R(D) : Rate Distortion Function

[T] : Transform Kernal

T_i : The i^{th} column vector of [T]

$t_{i,j}$: The $(i,j)^{th}$ element of [T]

$[T]^t$: Transpose of [T]

ASIC : Application Specific Integrated Circuit

CDS : Common Dyadic Symmetry

DCT : Discrete Cosine Transform

DS : Dyadic Symmetry

DHT : Discrete Hartley Transform

DST : Discrete Sine Transform

DWT : Discrete W Transform

DPCM : Differential Pulse Code Modulation

HCT : High Correlation Transform

ICT : Integer Cosine Transform

KLT : Karhunen-Loeve Transform

LCT : Low Correlation Transform

OWCT : Odd Weighted Cosine Transform

PSCT : Phase Shift Cosine Transform

TABLE OF CONTENTS

ABSTRACT ..... (i)

ACKNOWLEDGEMENT ..... (ii)

LIST OF PRINCIPLE SYMBOLS AND ABBREVIATIONS ..... (iii)

TABLE OF CONTENTS ..... (iv)

CHAPTER 1

INTRODUCTION

1.1 Introduction 1-1
1.2 Transform Coding Theory 1-1
1.2.1 Transformation - A Review 1-2
1.2.2 Bit Allocation 1-4
1.2.3 Quantization 1-5
1.3 Organisation of the Thesis 1-5

CHAPTER 2

GENERATION OF NEW ORTHOGONAL TRANSFORMS

2.1 Introduction 2-1
2.2 Theory of Dyadic Symmetry 2-1
2.3 Generation of Order-8 Orthogonal Transforms 2-4
2.3.1 Basic Principle 2-4
2.3.2 Generation Method 2-5
2.3.2.1 Scheme A 2-5
2.3.2.2 Scheme B 2-12
2.3.2.3 Scheme C 2-14
2.3.3 Efficiency of the New Orthogonal Transforms 2-16
2.4 Generalization of Large Block Size 2-17
2.5 Conclusion 2-18

CHAPTER 3

ODD WEIGHTED COSINE TRANSFORM

3.1 Introduction 3-1
3.2 Development of the Odd Weighted Cosine Transform 3-2
3.3 Fast Computational Algorithm for the OWCT 3-6
3.4 Performance Evaluation 3-9
3.4.1 One Dimensional Covariance Function 3-9

3.4.1.1 Residue Correlation 3-9
3.4.1.2 Maximum Reducible Bits 3-10
3.4.1.3 Transform Efficiency 3-11
3.4.1.4 Mean Square Error 3-12
3.4.2 Two Dimensional Covariance Function 3-13
3.4.2.1 Basis Restriction Error 3-13
3.5 Conclusion 3-14
3.6 Note on Publication 3-15

CHAPTER 4

A FAST RECURSIVE ALGORITHM FOR THE DISCRETE W TRANSFORM

4.1 Introduction 4-1
4.2 Symmetry Properties of the Discrete W Transform 4-2
4.3 The Fast Recursive Algorithm for the DWT 4-3
4.4 Computational Requirement 4-8
4.5 Conclusion 4-9
4.6 Note on Publication 4-10

CHAPTER 5

LSI IMPLEMENTATION OF THE INTEGER COSINE TRANSFORM

5.1 Introduction 5-1
5.2 The Integer Cosine Transform (ICT) 5-2
5.3 Design Considerations 5-3
5.3.1 Specifications 5-3
5.3.2 Selection of the Transform Matrix 5-4
5.3.3 I/O Pins Considerations 5-7
5.3.3.1 2-D Forward ICT 5-8
5.3.3.2 2-D Inverse ICT 5-9
5.4 Architecture 5-13
5.4.1 Input Stage 5-13
5.4.2 Control Block 5-13
5.4.3 Multiplier 5-14
5.4.4 Accumulator 5-15
5.4.5 Output Stage 5-15
5.4.6 Synchronization 5-16
5.5 Applications 5-17
5.5.1 One-Dimensional Transform 5-17

5.5.2 Two-Dimensional Transform 5-18
5.6 Conclusion 5-18

CHAPTER 6

CONCLUSION

6.1 Summary of Discoveries 6-1
6.2 Suggestions for Further Research 6-2

REFERENCES R-1

Chapter 1 INTRODUCTION

1.1 Introduction

Over the past ten years, with the increasing growth of the modern communication technologies, the demand for transmission and storage of image data is increasing rapidly. For a picture with a spatial resolution of 512 x 512 pels and at 8-bits/pel intensity resolution, it requires 2 Mbits to represent. For those large memory and/or channel capacity requirements for digital image storage and transmission, the compression technique for the image data is necessary.

Image data compression techniques can be classified into two basic categories : (a) predictive coding, and (b) transform coding. Techniques such as differential pulse code modulation (DPCM) and other predictive methods [1] [2] fall in the first category. Those methods exploit redundancy in the data and related to the casual model representation of image signals [2]. The principle of the method is simply stated as follows. Since image data source is highly correlated, on average the picture elements lying in the neighbour will tend to have the same gray levels. Therefore the value of one or more earlier pixels (previously coded) in the same line can be used to predict the present element. With the nature of the image given in a statistical context, on the average the prediction will be quite good with small coding error. In the second category, compression is achieved by an energy preserving (orthogonal) transformation of the given image data into another array such that maximum energy information is packed into minimum number of coefficients in the transform domain [1]-[4]. This technique employs the non-causal model representations of signals [2]. It is found that the transform coding method has a higher compression ability than predictive coding [2]. In this thesis, we will concentrate on transform coding system with emphasis on the transformation process. In the following pages, methods of generating new orthogonal transforms and implementation techniques of the transforms will be discussed. In the next section, some aspects of transform coding will be examined and a review on the transformations will be given. The organisation of this thesis will be given in the final section of this chapter.

1.2 Transform Coding Theory

Compression of data by transform coding method is achieved by three basic processes. The first process is to transform highly correlated image data into weakly correlated coefficients. The second process is to allocate bits to these

transform coefficients such that more bits are allocated to those coefficients with higher variance. The final process is to quantize the coefficients before the transmission. In order to achieve significant reduction in bit-rate whilst keeping the implementation simple, we must taking into consideration of all three processes.

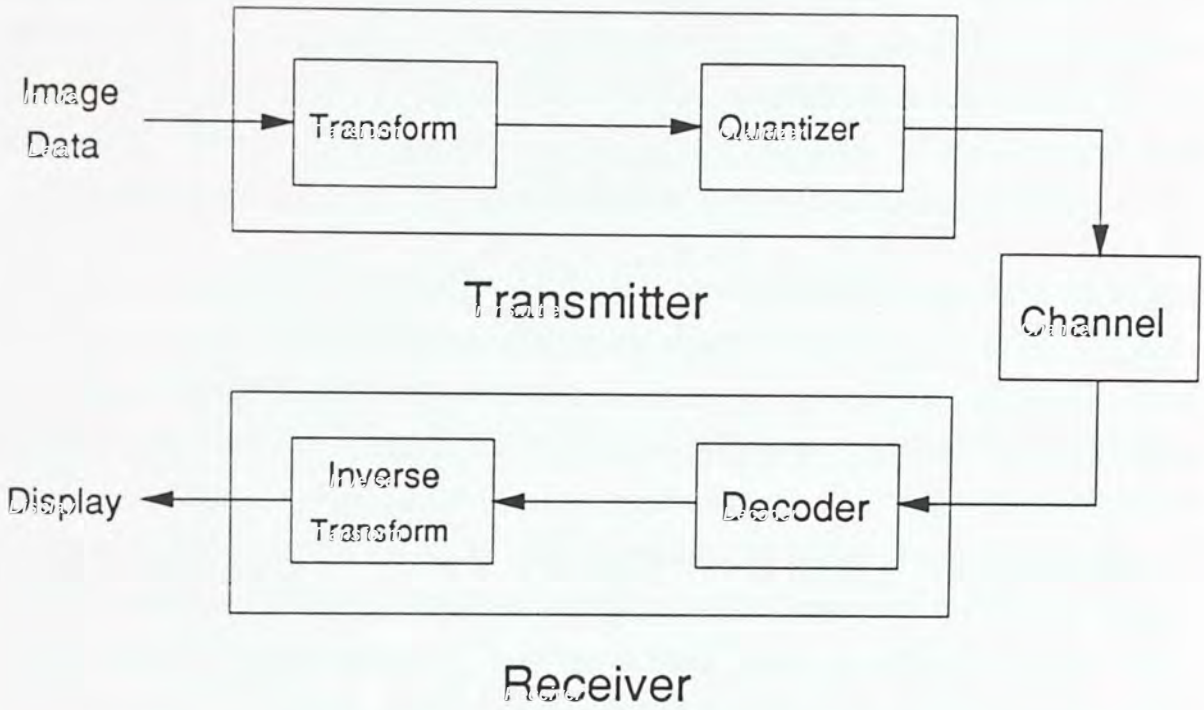


Figure 1.1 Block Diagram of a Transform Coding System

Figure 1.1 shows the block diagram of a basic transform coding system. The original picture image is divided into many sub-pictures of size n x n, where n is usually 8 or 16. Each sub-picture is transformed by an orthogonal transform into a set of weakly correlated coefficients which are then quantized and coded for transmission. At the receiver, the received data are decoded to the quantized transform coefficients, and an inverse transformation is applied to these coefficients to recover the original picture.

1.2.1 Transformation - A Review

The primary objective of the transformation is to convert statistically dependent picture elements into an array of uncorrelated coefficients such that most of the energy is packed into a minimum number of coefficients and the total energy of the spatial image data is preserved in the transform domain.

An orthogonal transform is said to be optimal when it completely decorrelate the picture image data. The Karhunen-Loeve Transform (KLT) [2] [5] is found to be the optimal one since it's basis vectors are in fact the eigenvectors of the covariance matrix of the incoming image data. The KLT is picture dependent since different pictures have different covariance matrices and so different KLTs. Although the explicit form of the KLT defined based on the first-order Markov model which is a widely accepted model for the image data has been found [6], the fast computational algorithm is still not known. Because of the computational difficulty of the KLT, it is not used for a practical transform coding system.

The problems of KLT mentioned before can be solved by the application of a sub-optimal transform which has fast computational algorithms. The first sub-optimal transform to be investigated for transform coding was the two-dimensional Fourier transform [7]. This was followed by the discovery that the Walsh transform could be utilized in place of the Fourier transform with a considerable decrease in computational requirement [8]. In 1971, investigation began into the application of Haar Transform (HT) [9]. The Haar transform has an extremely efficient computational algorithm, but results in a larger coding error. At about the same time, Enomoto and Shibata designed a new order-8 transform to match typical image vectors [10]. Pratt [11] generalized this transform which is now known as the Slant transform, and later applied it to digital image coding with a fast computational algorithm [12] resulting in a lower coding error for moderate block sizes in comparison to other unitary transforms. In the mean time, several different transforms have also been proposed for digital image coding.

The Discrete Cosine Transform (DCT), proposed by Ahmed et al. [13] in 1974, has drawn much interest for the researhers for it is asymmpotically equivalent to the KLT for the first-order stationary Markov process and its efficient fast computational algorithms exist [14] [15]. Comparisons between the DCT and other sub-optimal transforms using the first-order Markov model have shown that the DCT results in the least mean square error [16]. Clarke [17] has shown that when the adjacent correlation coefficient of the Markov-I model close to one, the KLT is in fact a discrete cosine transform. In 1976, Jain suggested that a sine transform (DST-I) is a KLT of a Markov process under the condition that the boundary of the process is known [18]. In other words, when the first-order stationary Markov process is decomposed into a boundary process and a residue process, the KLT of the residue process is a sine transform. Similar decomposition were proposed by Meiri and Yufelevich [19] and another sine transform was

generated for their residue process in their pinned sine transform coder. In the following years, several sinusoidal transforms were proposed and investigated for digital image coding [20] [21]. Wang [22] generalized the family of sinusoidal transforms and classified the even type cosine and sine transforms into four different versions. In this thesis, a new sinusiodal transform called Odd Weighted Cosine Transform (OWCT) is defined in chapter 3. It is shown that the OWCT has a better performance than the DCT in transform coding of images.

Apart from the sinusoidal transforms, the C-Matrix Transform (CMT) defined by Kwak et al. [23] has also found application in image coding for its simpler structure than the DCT since only integers are contained in the transform matrix. Cham and Clarke [24] generated two new transforms, the High Correlation Transform (HCT) and the Low correlation Transform (LCT) which are derived from the Walsh transform by the principle of Dyadic Symmetry. Computations of these two transform require only additions, subtractions and binary shifts. The performance of the two transforms in term of transform efficiency [24] both lie between that of the Walsh transform and that of the DCT for order-8 and 16. The HCT and LCT can be used to replace the Walsh transform. Cham further extended his work and generated the Integer Cosine Transform (ICT) from the DCT [25]. It has been shown that some order-8 ICTs have higher transform efficiency than the order-8 DCT. Implementation of the ICT is simpler than the DCT. In this thesis, we extend the work of Cham and generate a set of new orthogonal transforms which will be discussed in chapter 2. The implementation of the ICT using ASIC technology is given in chapter 5.

In 1984, a new transform called Discrete W Transform (DWT) is defined by Wang [22]. There are four types of the DWT and the type I DWT is also called Discrete Hartley Transform (DHT) by Bracewell [26]. In the past few years, the DWT has found useful in digital image coding [27] and other applications [28]. This is because it can replace the Discrete Fourier Transform (DFT) in some cases and it has a FFT like fast computational algorithm and only requires real operations. In chapter 4, a new fast recursive algorithm is developed for the DWT.

1.2.2 Bit Allocation

The bit-allocation process in the transform coding theory is base on the criterion that more bits are allocated to coefficients with larger variance. The optimal bit allocation can be derived from the Rate Distortion Theory [29], which states that the output of a source can be transmitted with average distortion D if the

transmission rate is larger than R(D). If D is the mean square error and the source has Gaussian probability distribution, then the minimum transmission rate R(D) is found to be

$$R(D) = \begin{cases} \frac{1}{2}\log_2\left(\frac{\sigma^2}{D}\right) & \text{for } \sigma > \sqrt{D} \\ 0 & \text{otherwise} \end{cases} \qquad (1.1)$$

where σ^2 is variance of the source. Equation (1.1) can be used to determine the number of bits required for each transform coefficient for minimum distortion [29].

The bit allocation scheme can be made adaptive. In an adaptive bit allocation scheme [30], the sub-pictures of the image are divided into different classes, and the variance of the coefficients for each class are computed. Equation (1.1) again is used to find the number of bits assigned to the coefficients for each class.

1.2.3 Quantization

As shown in the block diagram of a transform coding system in Figure 1.1, transform coefficients must be quantized before transmission. This process causes an unavoidable error called the quantization error. this error is the most significant factor causing the degradation in the reconstructed image when the communication channel and storage are assumed error-free.

In order to reduce the overall quantization error, quantizer should be designed according to the statistic of the incoming data being quantized. Max [31] showed how to minimize the distortion of a quantizer having a fixed number of output levels for a signal of known probability distribution. Such quantizer is called an optimal quantizer.

1.3 Organisation of the Thesis

Generation of new orthogonal transforms based on the two opposing criteria, the computation effectiveness and the compression ability, will be discussed in the next two chapters. In chapter 2, generation of new orthogonal transforms by the principle of dyadic symmetry will be discussed. Sereral definitions and theorems are derived and different schemes of generation of new orthogonal transforms are developed. An enhacement to the DCT called the Odd Weighted Cosine Transform (OWCT) is then defined in chapter 3. However, the transformation process requires a lot of computation efforts. In the following chapters, methods to speed up the

transformation process including the development of a fast computational algorithm and a dedicated integrated circuit for the transforms will be described. In chapter 4, a new fast recursive algorithm for all types of the radix-2 Discrete W Transform (DWT) is derived. In chapter 5, a LSI implementation of the Integer Cosine Transform using ASIC technology will be discussed. Finally, the overall conclusion and the summary of discoveries are given in chapter 6. Suggestions for future works will also be given in this chapter.

Chapter 2 GENERATION OF NEW ORTHOGONAL TRANSFORMS

2.1 Introduction

Over the past twenty years, many orthogonal transforms have been proposed and investigated for digital image coding. The Discrete Cosine Transform (DCT) [13] is widely accepted as the industrial standard for the transform being used in the practical transform coding system. Apart from the sinusoidal transforms, several non-sinusoidal transforms such as the C-Matrix Transform (CMT) [23], High Correlation Transform (HCT) and Low Correlation Transform (LCT) [24] which have simpler structure have been proposed for image coding.

The HCT and LCT defined by Cham [24] are derived from the Walsh transform by the principle of dyadic symmetry. Computations of these two transform require only additions, subtractions and binary shifts. It has been shown that the transform efficiencies [24] of these two transforms lie between those of the Walsh transform and the DCT. In this chapter, we extend the work of Cham and generate several new orthogonal transforms. In section 2.2, some basic theories of the dyadic symmetry will first be introduced. Then some definitions and theorems related to generation of new order-8 orthogonal transforms will be defined and derived in section 2.3. Several schemes of generating new orthogonal transforms are also developed in this section. The transform efficiencies of the new orthogonal transforms will then be given in section 2.4. The generalization of these new orthogonal transforms to larger block sizes will be discussed in section 2.5 and finally the overall conclusion will be drawn in the last section.

2.2 Theory of Dyadic Symmetry

The concept of dyadic symmetry was first proposed by Cham [24] [32]. In this section, the basic definitions and theorems of dyadic symmetry are stated without proof and the detail discusssion will be found in the works of Cham [24] [32].

Let F be a number field. Unless specified otherwise, all vectors in F are column vectors.

Definition 2.1 : Vector A of 2^m elements $[a_0 \ a_1 \ \dots \ a_{2^m-1}]$ in F is said to have S^{th} *dyadic* symmetry (DS) iff

$$a_j = t \cdot a_{j \oplus s}$$

for $j \in [0, 2^m - 1]$ and $S \in [1, 2^m - 1]$

where t=1 when the type of dyadic symmetry is even and t=-1 when the type is odd.

The dyadic symmetry describes the symmetry properties of the vector components of a vector. For example, the order-8 vector A, having the even type S^{th} dyadic symmetry, are given in Table 2.1.

Dyadic Symmetry S	Vector A								
	[a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7]
1	[a	a	b	b	c	c	d	d]
2	[a	b	a	b	c	d	c	d]
3	[a	b	b	a	c	d	d	c]
4	[a	b	c	d	a	b	c	d]
5	[a	b	c	d	b	a	d	c]
6	[a	b	c	d	c	d	a	b]
7	[a	b	c	d	d	c	b	a]

Table 2.1 The Seven Vectors A having S^{th} Even Dyadic Symmetry

Definition 2.2 : Two order 2^m vectors U and V in F are said to have a common dyadic symmetry (CDS) S, where $S \in [1, 2^m - 1]$, if both U and V have even or odd type S^{th} dyadic symmetry.

Theorem 2.1 : Two order 2^m vectors U and V in F are orthogonal if U and V have a CDS S and the type of the S^{th} dyadic symmetry of the two vectors are different.

Let B be a binary field with "exclusive or" as addition and "logical and" as multiplication. Unless specified otherwise all vectors in B are row vectors. Dyadic symmetries can be regarded as vectors in B.

Theorem 2.2 : If an order 2^m vector U in F have dyadic symmetry $S_1, S_2, ..., S_r$, then this vector also has dyadic symmetry S_k, where

$$S_k = S_1 \oplus S_2 \oplus ... \oplus S_r$$

Example : an order-8 vector A has dyadic symmetry (001) and (010) has also dyadic symmetry (011).

Definition 2.4 : The dyadic symmetry S_1, S_2, ..., S_m are said to be dependent if there exist m element k_1, k_2, ..., k_m in F, not all zero, such that

$$k_1 \cdot S_1 \oplus k_2 \cdot S_2 \oplus \ ... \ \oplus k_m \cdot S_m = 0$$

otherwise, the m dyadic symmetry are said to be linearly independent.

Theorem 2.3 : Consider order 2^m vectors V_0, V1, V_2, ..., V_{N-1} in F which have m independent CDSs, says S_1, S_2, ..., S_m . Let vector E_i = [e_{i1}, e_{i2}, ..., e_{im}] in B represents the type of the m independent CDSs of vector V_i, where $e_{ij} = 0$ if the type of S_i is even and $e_{ij} = 1$ if the type of S_i is odd. If E_i for i $\in$ [0, N-1] are all different, then the N vectors are independent.

It is interesting to note that if e_{ij} is the binary representation of the type of the $(2^j - 1)^{th}$ dyadic symmetry ('0' is even type and '1' is odd type) of the sequency-ordered Walsh basis vector W_i, where i $\in$ [0, N-1] and j $\in$ [1, m], E_i can also be regarded as a binary value which indicated the number of zero crossing (sequency) of the basis vector. This property is illustrated by an order-8 sequency-ordered Walsh matrix as shown in Table 2.2.

Sequency i	Independent CDS 001 011 111 S_1 S_2 S_3	Walsh Basis Vector W_i
0	0 0 0	[1 1 1 1 1 1 1 1]
1	0 0 1	[1 1 1 1 -1 -1 -1 -1]
2	0 1 0	[1 1 -1 -1 -1 -1 1 1]
3	0 1 1	[1 1 -1 -1 1 1 -1 -1]
4	1 0 0	[1 -1 -1 1 1 -1 -1 1]
5	1 0 1	[1 -1 -1 1 -1 1 1 -1]
6	1 1 0	[1 -1 1 -1 -1 1 -1 1]
7	1 1 1	[1 -1 1 -1 1 -1 1 -1]

Table 2.2 An Order-8 Sequency-Ordered Walsh Matrix

Theorem 2.4 : Every basis vector of an order 2^m Walsh matrix has all 2^m -1 dyadic symmetry

Corollary 2.1 : All basis vector of an order 2^m Walsh matrix have 2^m -1 CDS or m independent CDSs.

Theorem 2.5 : For any pair of basis vectors of an order 2^m Walsh matrix, if m - 1 of the m independent CDSs are of the same type, then the one left will be of the different type.

2.3 Generation of Order-8 Orthogonal Transforms

2.3.1 Basic Principle

Consider the order 2^m sequency-ordered Walsh matrix. It has 2^m basis vectors and each of them has m independent CDS. We can pick up (m-r) DS, S_1, S_2, ..., S_{m-r} which are also independent. There will be a set of 2^r basis vectors whose dyadic symmetries S_1, S_2, ..., S_{m-r} are of the same type.

Definition 2.5 : If the 2^r basis vectors are replaced by another set of 2^r orthogonal vectors which are without the remaining r independent CDS. The r independent CDS of the set of 2^r vectors are said to be destroyed.

Example : Consider the order-8 (m=3) Walsh matrix shown in Table 2.2, we pick up one independent CDS, say S_1. There are 2^2 basis vectors W_0, W_1, W_2 and W_3 whose S_1 Dyadic symmetry are all type 0. S_2 and S_3, the remaining independent CDSs of W_0, W_1, W_2 and W_3, are said to be destroyed if W_0, W_1, W_2 and W_3 are replaced by another 4 orthogonal vectors U_0, U_1, U_2 and U_3 which do not have dyadic symmetry S_2 and S_3. There are many possible U_0, U_1, U_2 and U_3. The following is a possible set.

$$
\begin{array}{lcccccccccc}
U_0 & [& a & a & b & b & c & c & d & d &] \\
U_1 & [& d & d & c & c & -b & -b & -a & -a &] \\
U_2 & [& c & c & -d & -d & -a & -a & b & b &] \\
U_3 & [& b & b & -a & -a & d & d & -c & -c &]
\end{array}
$$

where a, b, c and d are arbitrary constants.

Theorem 2.6 : The 2^r basis vectors of an order 2^m matrix generated by destroying r independent CDSs from an order 2^m sequency ordered Walsh matrix are orthogonal.

[Proof] By definition 2.5, the set of 2^r new basis vectors have the same type (m-r) independent CDS and have at least one different type in those (m-r) Dyadic symmetry when comparing with any one of the remain $2^{(m-r)}$ basis vectors. Then by theorem 2.1, the set of 2^m new basis vectors is orthogonal to any one of the remaining basis vectors.

By theorem 2.6, we can generate order 2^m orthogonal transform matrices by destroying independent CDSs from the basis vectors of the order 2^m Walsh matrix. There are many schemes to destroy the dyadic symmetry. The schemes will become more complicated as the order of the matrix increases. In this thesis, we shall show how to generate order-8 transform matrices from the Walsh matrix.

2.3.2 Generation Method

Cham and Clarke [24] generated the HCT and LCT by destroying at most one independent CDS from the Walsh basis vectors by a definite scheme. The performance of HCT and LCT in term of transform efficiency lies between that of the DCT and Walsh transform. In the following pages, we extend the work of Cham to generate a set of new order-8 orthogonal transforms by destroying at most two independent CDSs from the Walsh basis vectors. The method is based on two basic criteria. The first one is to make the low sequency basis vectors assemble like a cosine wave; and the second is to remain DC (zero sequency) vector constant in the transform matrix. We propose three different schemes for generating new order-8 orthogonal transform matrices.

2.3.2.1 Scheme A - Reserve the 111th Dyadic Symmetry

The scheme is illustrated in Figure 2.1. CDSs S_1 and S_2 of Walsh basis vectors W_1, W_3, W_5 and W_7 are destroyed. CDSs S_1 of W_2 and W_6 are destroyed.

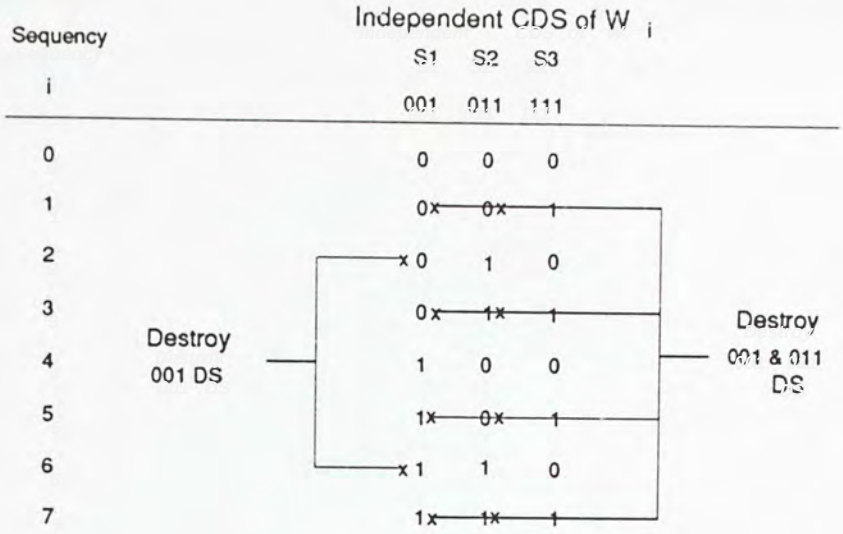


Figure 2.1 Scheme A

By the remaining $(111)^{th}$ Dyadic symmetry, the Walsh matrix can be decomposed into two order-4 Walsh matrices as follows:

$$[W_8] = [P_{A8}] \begin{bmatrix} W_4 & 0 \\ 0 & \bar{I}_4 W_4 \bar{I}_4 \end{bmatrix} [B_{A8}] \qquad (2.1)$$

where

i)
$$[P_{A8}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

ii)
$$[B_{A8}] = \frac{1}{\sqrt{2}} \begin{bmatrix} I_4 & \bar{I}_4 \\ \bar{I}_4 & -I_4 \end{bmatrix}$$

iii) I is the identity matrix

iv) $\bar{I}$ is the anti-diagonal identity matrix.

v) $[P_A]$ is a permutation matrix which puts the even row vectors into upper half and the odd row vectors into lower half.

Similarly, by the $(011)^{th}$ Dyadic symmetry, the order-4 Walsh matrix can further be decomposed as follows

$$[W_4] = [P_{A4}] \begin{bmatrix} W_2 & 0 \\ 0 & \bar{I}_2 W_2 \bar{I}_2 \end{bmatrix} [B_{A4}] \tag{2.2}$$

where

$$[P_4] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[B_{A4}] = \frac{1}{\sqrt{2}} \begin{bmatrix} I_2 & \bar{I}_2 \\ \bar{I}_2 & -I_2 \end{bmatrix}$$

According to the scheme shown in Figure 2.1, the lower corner order-4 Walsh matrix in equation (2.1) and the lower corner order-2 Walsh matrix in equation (2.2) should be replaced by another order-4 and order-2 orthogonal matrices which don't have any dyadic symmetry. After replacing the corresponding matrices, a new order-8 orthogonal transform matrix $[TX_{A8}]$ can be generated as follows

$$[TX_{A8}] = [P_{A8}] \begin{bmatrix} D_4 & 0 \\ 0 & \bar{I}_4 E_4 \bar{I}_4 \end{bmatrix} [B_{A8}]$$

and

$$[D_4] = [P_{A4}] \begin{bmatrix} W_2 & 0 \\ 0 & \bar{I}_2 E_2 \bar{I}_2 \end{bmatrix} [B_{A4}] \tag{2.3}$$

where $[E_4]$ and $[E_2]$ are sequency-ordered orthogonal matrices which don't have any dyadic symmetry in their basis vectors. The order-2 orthogonal matrix $[E_2]$ can simply be the following form

$$\begin{bmatrix} e & f \\ f & -e \end{bmatrix}$$

In the case of order-4 orthogonal matrix $[E_4]$, the sign of the basis vector components should have the following distribution to maintain the sequency of the matrix.

Sequency	Sign Distribution
0	+ + + +
1	+ + - - / + - - - / + + + -
2	+ - - + / + - + + / + + - +
3	+ - + -

There are many possible order-4 orthogonal matrices with sign distribution as shown above. The following seven matrices have been studied.

Matrix 1

$$\begin{bmatrix} a & b & c & d \\ d & c & -b & -a \\ c & -d & -a & b \\ b & -a & d & -c \end{bmatrix}$$

Orthogonal for all a, b, c and d

Matrix 2

$$\begin{bmatrix} a & b & c & d \\ c & d & -a & -b \\ b & -a & -d & c \\ d & -c & b & -a \end{bmatrix}$$

Orthogonal for all a, b, c and d

Matrix 3

$$\begin{bmatrix} a & b & c & d \\ b & -d & -a & -c \\ c & -a & d & b \\ d & -c & b & -a \end{bmatrix}$$

Orthogonal iff $ab = cd + bd + ac$

Matrix 4

$$\begin{bmatrix} a & b & c & d \\ b & d & a & -c \\ c & a & -d & b \\ d & -c & b & -a \end{bmatrix}$$

Orthogonal iff $cd = ac + bd + ab$

Matrix 5

$$\begin{bmatrix} a & b & c & d \\ b & d & -c & -a \\ c & -c & 0 & c \\ d & -a & c & -b \end{bmatrix}$$

Orthogonal iff

$b = a + d$ and $c^2 = a^2 + d^2 + ad$

Matrix 6

$$\begin{bmatrix} a & b & c & d \\ c & c & 0 & -c \\ d & -a & -c & b \\ b & -d & c & -a \end{bmatrix}$$

Orthogonal iff

$d = a + b$ and $c^2 = a^2 + b^2 + ab$

Matrix 7

$$\begin{bmatrix} a & b & c & d \\ b & 0 & -b & -b \\ c & -b & -d & a \\ d & -b & a & -c \end{bmatrix}$$

Orthogonal iff

$a = c + d$ and $b^2 = c^2 + d^2 + cd$

where a, b, c, d, e and f are arbitrary constants.

Using the above matrices as matrices $[E_2]$ and $[E_4]$, we can find seven new orthogonal transforms by scheme A.

Transform 1 (TX1)

The transform matrix of TX1 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (d & c & -b & -a & a & b & -c & -d) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & -d & -a & b & -b & a & d & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (b & -a & d & -c & c & -d & a & -b) \end{bmatrix}$$

Orthogonal for all a, b, c, d, e and f

Transform 2 (TX2)

The transform matrix of TX2 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (c & d & -a & -b & b & a & -d & -c) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (b & -a & -d & c & -c & d & a & -b) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -c & b & -a & a & -b & c & -d) \end{bmatrix}$$

Orthogonal for all a, b, c, d, e and f

Transform 3 (TX3)

The transform matrix of TX7 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (b & -d & -a & -c & c & a & d & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & -a & d & b & -b & -d & a & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -c & b & -a & a & -b & c & -d) \end{bmatrix}$$

Orthogonal iff $ab = bd + cd + ac$

Transform 4 (TX4)

The transform matrix of TX8 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (b & d & a & -c & c & -a & -d & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & a & -d & b & -b & d & -a & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -c & b & -a & a & -b & c & -d) \end{bmatrix}$$

Orthogonal iff $cd = ac + bd + ab$

Transform 5 (TX5)

The transform matrix of TX9 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (b & d & -c & -a & a & c & -d & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & -c & 0 & c & -c & 0 & c & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -a & c & -b & b & -c & a & -d) \end{bmatrix}$$

Orthogonal iff $b = a + d$ and $c^2 = a^2 + d^2 + ad$

Transform 6 (TX6)

The transform matrix of TX10 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (c & c & 0 & -c & c & 0 & -c & -c) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (d & -a & -c & b & -b & c & a & -d) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (b & -d & c & -a & a & -c & d & -b) \end{bmatrix}$$

Orthogonal iff $d = a + b$ and $c^2 = a^2 + b^2 + ab$

Transform 7 (TX7)

The transform matrix of TX11 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (b & 0 & -b & -b & b & b & 0 & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & -b & -d & a & -a & d & b & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -b & a & -c & c & -a & b & -d) \end{bmatrix}$$

Orthogonal iff $a = c + d$ and $b^2 = c^2 + d^2 + cd$

2.3.2.2 Scheme B - Reserve the 011th Dyadic Symmetry

The scheme is illustrated as follows

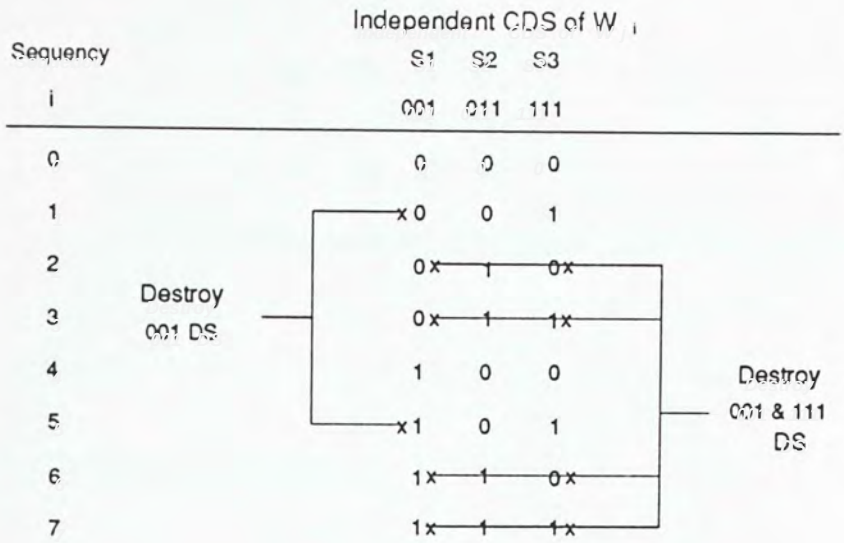


Figure 2.2 Scheme B

The method of generating new orthogonal matrices is similar to scheme A. The new orthogonal transform matrix $[TX_{B8}]$ generated in this scheme is as follows

$$[TX_{B8}] = [P_{B8}]\begin{bmatrix} D_4 & 0 \\ 0 & E_4 \end{bmatrix}[B_{B8}]$$

and

$$[D_4] = [P_{A4}]\begin{bmatrix} W_2 & 0 \\ 0 & \bar{I}_2 E_2 \bar{I}_2 \end{bmatrix}[B_{A4}] \qquad (2.4)$$

where

$$[P_{B8}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[B_{B\,8}] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

There are two new sequency ordered orthogonal transform matrices generated in this scheme.

Transform 8 (TX8)

The transform matrix of TX3 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (e & f & f & e & -e & -f & -f & -e) \\ k_2 & (a & b & -b & -a & -d & -c & c & d) \\ k_3 & (d & c & -c & -d & a & b & -b & -a) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (f & -e & -e & f & -f & e & e & -f) \\ k_6 & (c & -d & d & -c & -b & a & -a & b) \\ k_7 & (b & -a & a & -b & c & -d & d & -c) \end{bmatrix}$$

Orthogonal for all a, b, c, d, e and f

Transform 9 (TX9)

The transform matrix of TX4 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (e & f & f & e & -e & -f & -f & -e) \\ k_2 & (a & b & -b & -a & -d & -c & c & d) \\ k_3 & (c & d & -d & -c & b & a & -a & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (f & -e & -e & f & -f & e & e & -f) \\ k_6 & (b & -a & a & -b & -c & d & -d & c) \\ k_7 & (d & -c & c & -d & a & -b & b & -a) \end{bmatrix}$$

Orthogonal for all a, b, c, d, e and f

2.3.2.3 Scheme C - Reserve the 001th Dyadic Symmetry

The scheme is illustrated as follows

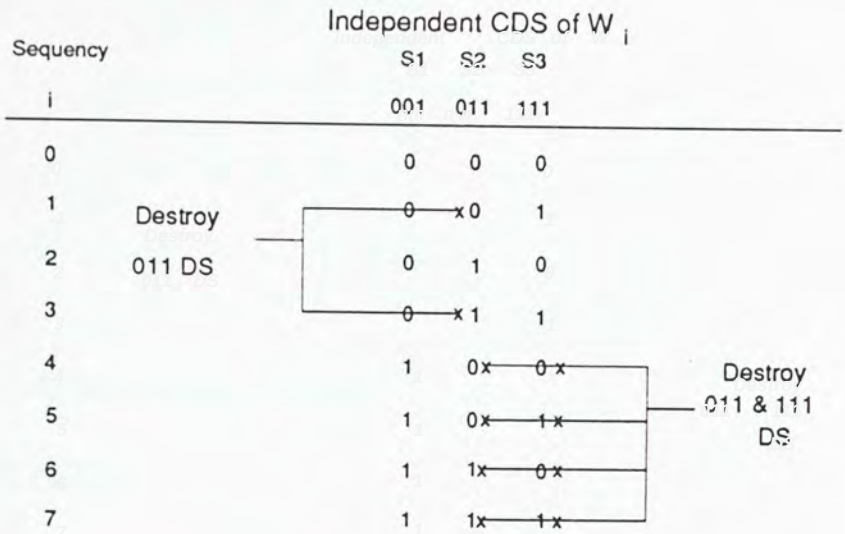


Figure 2.3 Scheme C

The method of generating new orthogonal matrices is similar to scheme A. The new orthogonal transform matrix $[TX_{C8}]$ generated in this scheme is as follows

$$[TX_{C8}] = \begin{bmatrix} D_4 & 0 \\ 0 & E_4 \end{bmatrix} [B_{C8}]$$

and

$$[D_4] = [P_{A4}] \begin{bmatrix} W_2 & 0 \\ 0 & \bar{I}_2 E_2 \bar{I}_2 \end{bmatrix} [B_{A4}] \qquad (2.4)$$

where

$$[B_{C8}] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

There are four new order-8 sequency ordered orthogonal transform matrices generated.

Transform 10 (TX10)

The transform matrix of TX5 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (e & e & f & f & -f & -f & -e & -e) \\ k_2 & (1 & 1 & -1 & -1 & -1 & -1 & 1 & 1) \\ k_3 & (f & f & -e & -e & e & e & -f & -f) \\ k_4 & (a & -a & -b & b & c & -c & -d & d) \\ k_5 & (d & -d & -c & c & -b & b & a & -a) \\ k_6 & (c & -c & d & -d & -a & a & -b & b) \\ k_7 & (b & -b & a & -a & d & -d & c & -c) \end{bmatrix}$$

Orthogonal for all a, b, c, d, e and f

Transform 11 (TX11)

The transform matrix of TX6 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (e & e & f & f & -f & -f & -e & -e) \\ k_2 & (1 & 1 & -1 & -1 & -1 & -1 & 1 & 1) \\ k_3 & (f & f & -e & -e & e & e & -f & -f) \\ k_4 & (a & -a & -b & b & c & -c & -d & d) \\ k_5 & (c & -c & -d & d & -a & a & b & -b) \\ k_6 & (b & -b & a & -a & -d & d & -c & c) \\ k_7 & (d & -d & c & -c & b & -b & a & -a) \end{bmatrix}$$

Orthogonal for all a, b, c, d, e and f

Transform 12 (TX12)

The transform matrix of TX12 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (e & e & f & f & -f & -f & -e & -e) \\ k_2 & (1 & 1 & -1 & -1 & -1 & -1 & 1 & 1) \\ k_3 & (f & f & -e & -e & e & e & -f & -f) \\ k_4 & (a & -a & -b & b & c & -c & -d & d) \\ k_5 & (b & -b & d & -d & -a & a & c & -c) \\ k_6 & (c & -c & a & -a & d & -d & -b & b) \\ k_7 & (d & -d & c & -c & b & -b & a & -a) \end{bmatrix}$$

Orthogonal iff $ab = bd + cd + ac$

Transform 13 (TX13)

The transform matrix of TX13 is

$$\begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (e & e & f & f & -f & -f & -e & -e) \\ k_2 & (1 & 1 & -1 & -1 & -1 & -1 & 1 & 1) \\ k_3 & (f & f & -e & -e & e & e & -f & -f) \\ k_4 & (a & -a & -b & b & c & -c & -d & d) \\ k_5 & (b & -b & -d & d & a & -a & c & -c) \\ k_6 & (c & -c & -a & a & -d & d & -b & b) \\ k_7 & (d & -d & c & -c & b & -b & a & -a) \end{bmatrix}$$

Orthogonal iff $cd = ac + bd + ab$

2.3.3 Efficiency of the New Orthogonal Transform

Using the proposed three schemes, we generate thirteen new orthogonal transform matrices. The transform matrices of all new transforms have arbitrary constant a, b, c, d, e and f whose values needed to be determined. The optimal values of these constants are searched in the integer field using on the criterion of transform efficiency [24] which measures the decorrelation ability of an orthogonal transform on a first-order Markov process of adjacent element correlation coefficient ρ. Detailed descriptions of transform efficiency will be given in the next chapter. The range of searching is limited to 10, and the time required for searching a optimal point is about 5 minutes under the IBM PC-AT environment. From the searching results, we found that most of the new transforms have the largest

efficiency when a=10, b=9, c=6, d=2, e=3 and f=1. Table 2.3 shows the corresponding transform efficiency of each new transforms with different adjacent element correlation coefficients ρ.

ρ	0.80	0.90
DCT	84.9664	89.8357
TX1	81.8920	88.4449
TX2	84.3783	89.6576
TX3	85.7462	90.0289
TX4	66.7639	75.0167
TX5	77.0503	82.1533
TX6	72.2185	78.4059
TX7	82.5350	87.2167
TX8	61.2947	69.5347
TX9	60.3603	68.5029
TX10	68.1778	77.1991
TX11	68.7879	77.7797
TX12	67.5725	76.9479
TX13	67.5724	76.9479

Table 2.3 Transform Efficiency of New Orthogonal Transforms with a=10, b=9, c=6, d=2, e=3 and f=1

It is noted that transforms TX1, TX2 and TX3, which generated by scheme A, have the efficiency close to the DCT since their low sequency basis vectors are resembled more likely to a cosine wave than the other transforms.

2.4 Generalized to Large Block Size

The generalization of the transforms generated by scheme to any block size N is shown below.

$$[TX_{A\,N}] = [P_{A\,N}]\begin{bmatrix} TX_{A\,N/2} & 0 \\ 0 & \bar{I}_{N/2}\, E_{N/2}\, \bar{I}_{N/2} \end{bmatrix}[B_{A\,N}] \qquad (2.5.a)$$

and

$$[E_{N/2}] = \begin{bmatrix} E_{A\,N/4} & E_{B\,N/4} \\ R\{E_{B\,N/4}\} & -R\{E_{A\,N/4}\} \end{bmatrix} \qquad (2.5.b)$$

where

$$[B_{AN}] = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{N/2} & \bar{I}_{N/2} \\ \bar{I}_{N/2} & -I_{N/2} \end{bmatrix}$$

$$[P_{AN}]^t = \begin{bmatrix} 1 & 0 & 0 & . & . & . & . & . & . \\ 0 & 0 & 1 & 0 & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & 0 & 1 & 0 \\ 0 & 1 & 0 & . & . & . & . & . & . \\ 0 & 0 & 0 & 1 & 0 & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . & 0 & 1 \end{bmatrix}$$

where the R{X} is an operation on a N x N matrix X, for Y=R{X} then the (i,j) element of Y is given by

$$\text{sign } \{Y(i,j)\} = \text{sign } \{X(i,j)\}$$

$$\text{mag } \{Y(i,j)\} = \text{mag } \{X(i,N-1-j)\}$$

$$\text{for } i, j = 0, 1, 2, \ldots, N-1$$

where sign{x} is the sign of x and mag {x} represents the magnitude of x

By the equations (2.5.a) and (2.5.b), the order N new transform matrix $[TX_{AN}]$ can be recursively generated by the order-2 and order-4 matrices shown in the above section. It is noted that the equation (2.5.b) is only valid for the $[E_A]$ matrix have the structure of matrix 1 shown in section 2.3.2.1.

2.5 Conclusion

In this chapter, the method of generation of new orthogonal transforms by destroying dyadic symmetry is extended. Several definition and theorems are defined and derived. Three different schemes are proposed for generation of new transforms and thirteen new transforms are generated. It it found that the transforms generated from scheme A have better transform efficiency than the other schemes. It may be due to the reason that the lower sequency basis vectors of those transforms resembled more likely to a cosine shape. The transform efficiency of the transforms TX1, TX2 and TX3 are virtually the same as the DCT. Since the basis vector components of those transforms are integers instead of real numbers as those in the DCT, the implementation of such transforms is simpler than the DCT.

However, it is noted that the performance of the transforms can be affected by the magnitude of the basis vector components. It is believed that if we modify the basis vector components of the DCT, a better transform will be found. In the next chapter, an enhacement to the DCT called Odd weighted Cosine Transform (OWCT) is developed by modifying the magnitude of the odd basis vector components.

Chapter 3 ODD WEIGHTED COSINE TRANSFORM

3.1 Introduction

Transform coding is an efficient technique in achieving high compression rate for image data. The Karhunen-Loeve Transform (KLT) is statistically optimal in energy compaction since it diagonalizes the covariance matrix of the input random sequence [5]. However, the implementation of the KLT is very complicated as it requires a large number of multiplications to generate the transform matrix for each incoming signal vector and there is no fast computational algorithm. The Discrete Cosine Transform (DCT), proposed by Ahmed et al. [13] (which is also known as DCT-II by Wang [22]), has been widely accepted as the best substitute for the KLT in the transform coding system for its asymptotically equivalent to the KLT [16] and its fast algorithms [14] [15] [22] [37] exist.

During past ten years, many researchers proposed many different types of sinusoidal transforms to approximate the KLT [18]-[21] and tested via various criteria. Wang [33] developed a modified version for the DCT, called the Phase Shift Cosine Transform (PSCT), by putting a little phase change to the even part of the transform matrix of the DCT. The PSCT in comparison with the DCT has better performance for adjacent element coefficient close to unity, however, the non-constant zero order (sequency) basis vector of the PSCT causes an undesirable distribution of the AC energy within what should be purely DC coefficient. These effect may degrade the performance of the transform in some applications such as filtering, data compression etc.[34]. In this chapter, we propose a new transform, called the Odd Weighted Cosine Transform (OWCT), which modifies only the odd part of the transform matrix of the DCT and keeps the zero-order basis vector constant. In Section 3.2, the OWCT will be defined and the fast computational algorithm will be discussed in Section 3.3. In Section 3.4, we compare the OWCT with the DCT via various criteria based on both the one and two dimensional covariance function of the Markov-I model.

3.2 Development of the Odd Weighted Cosine Transform (OWCT)

There are many different types of cosine transform and only the even type is found applications in digital image coding. According to the classification of Wang [22], there are four types for the even DCT.

$$[C^{I}_{N+1}] = \sqrt{\frac{2}{N}}\left[k_i k_j \cos\left(ij\frac{\pi}{N}\right)\right]$$

$$i, j = 0, 1, 2, ..., N$$

$$[C^{II}_{N}] = \sqrt{\frac{2}{N}}\left[k_i \cos\left(i\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)\right]$$

$$[C^{III}_{N}] = \sqrt{\frac{2}{N}}\left[k_j \cos\left(\left(i+\frac{1}{2}\right)j\frac{\pi}{N}\right)\right]$$

$$[C^{IV}_{N}] = \sqrt{\frac{2}{N}}\left[\cos\left(\left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)\right]$$

$$i, j = 0, 1, 2, ..., N-1$$

where $C^{\wedge}_{N}$ represents the order-N version A discrete cosine transform and

$$k_i = \begin{cases} 1 & \text{if } i \neq 0 \text{ and } i \neq N \\ \frac{1}{\sqrt{2}} & \text{if } i=0 \text{ or } i=N \end{cases}$$

The version II discrete cosine transform (DCT-II) is the most commonly used version [13]. An order N DCT-II can be decomposed into two order N/2 transforms, the odd part is the DCT-IV, and the even part is the DCT-II [22] as shown in equation (3.1).

$$[C^{II}_{N}] = [P_N]\begin{bmatrix} C^{II}_{N/2} & 0 \\ 0 & \bar{I}\, C^{IV}_{N/2}\, \bar{I} \end{bmatrix}[B_N] \qquad (3.1)$$

where

(i) $$[B_N] = \frac{1}{\sqrt{2}}\begin{bmatrix} I_{N/2} & \bar{I}_{N/2} \\ \bar{I}_{N/2} & -I_{N/2} \end{bmatrix}$$

(ii) I is the identity matrix,

(iii) $\bar{I}$ is the anti-diagonal identity matrix,

(iv) P is a permutation matrix which puts the even row vectors into upper half and the odd row vectors into lower half [22].

Theorem 3.1 : If matrix U is orthonormal then matrix $T = \bar{I} U \bar{I}$ is orthonormal.

Theorem 3.2 : If matrices U and V are orthonormal then matrix T is orthonormal where

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} I & \bar{I} \\ \bar{I} & I \end{bmatrix}$$

Theorem 3.1 and 3.2 imply that the matrix $C^{IV}_{N/2}$ in equation (3.1) can be replaced by another orthonormal matrix and the matrix product remains orthonormal. Therefore, we can modify the odd part of the order N DCT-II matrix by substituting an order N/2 orthonormal matrix U with the DCT-IV matrix. A new orthogonal transform, called Odd Weighted Cosine Transform (OWCT), is defined in the following matrix form

$$[OWCT_N] = [P_N] \begin{bmatrix} C^{II}_{N/2} & 0 \\ 0 & \bar{I} U_{N/2} \bar{I} \end{bmatrix} [B_N]$$

where

$$U_M = \frac{2}{\sqrt{M}} \cos\frac{(2i+1)(2j+1)\pi}{4M} \sin\left[\frac{\pi}{4} - (-1)^i f_M(i) f_M(j) \left(\frac{M}{4} - \frac{1}{2} - \langle j \rangle_{M/2}\right)\beta\right]$$

for i, j = 0, 1, 2, ..., M-1 and M=N/2

$\langle m \rangle_n$ means the residue of m mod n and

$$f_M(X) = \begin{cases} 1, & X \le \frac{M}{2} - 1 \\ -1, & X \ge \frac{M}{2} \end{cases}$$

e.g. for M=4

$$[U_4] = \begin{bmatrix} a\cos\left(\frac{1}{16}\pi\right) & b\cos\left(\frac{3}{16}\pi\right) & b\cos\left(\frac{5}{16}\pi\right) & a\cos\left(\frac{7}{16}\pi\right) \\ b\cos\left(\frac{3}{16}\pi\right) & -a\cos\left(\frac{7}{16}\pi\right) & -a\cos\left(\frac{1}{16}\pi\right) & -b\cos\left(\frac{5}{16}\pi\right) \\ b\cos\left(\frac{5}{16}\pi\right) & -a\cos\left(\frac{1}{16}\pi\right) & a\cos\left(\frac{7}{16}\pi\right) & b\cos\left(\frac{3}{16}\pi\right) \\ a\cos\left(\frac{7}{16}\pi\right) & -b\cos\left(\frac{5}{16}\pi\right) & b\cos\left(\frac{3}{16}\pi\right) & -a\cos\left(\frac{1}{16}\pi\right) \end{bmatrix}$$

where

$$a = \sin\left(\frac{\pi}{4} - \frac{1}{2}\beta\right)$$

$$b = \sin\left(\frac{\pi}{4} + \frac{1}{2}\beta\right)$$

e.g. for M=8

$$[U_8] = \begin{bmatrix} a\cos A & b\cos 3A & c\cos 5A & d\cos 7A & d\cos 9A & c\cos 11A & b\cos 13A & a\cos 15A \\ d\cos 3A & c\cos 9A & b\cos 15A & -a\cos 11A & -a\cos 5A & -b\cos A & -c\cos 7A & -d\cos 13A \\ d\cos 5A & c\cos 15A & -b\cos 7A & -a\cos 3A & -a\cos 13A & b\cos 9A & c\cos A & d\cos 11A \\ a\cos 7A & -b\cos 11A & -c\cos 3A & d\cos 15A & d\cos A & c\cos 13A & -b\cos 5A & -a\cos 9A \\ a\cos 9A & -b\cos 5A & -c\cos 13A & d\cos A & -d\cos 15A & -c\cos 3A & b\cos 11A & a\cos 7A \\ d\cos 11A & -c\cos A & b\cos 9A & a\cos 13A & -a\cos 3A & b\cos 7A & c\cos 15A & -d\cos 5A \\ d\cos 13A & -c\cos 7A & b\cos A & -a\cos 5A & a\cos 11A & b\cos 15A & -c\cos 9A & d\cos 3A \\ a\cos 15A & -b\cos 13A & c\cos 11A & -d\cos 9A & d\cos 7A & -c\cos 5A & b\cos 3A & -a\cos A \end{bmatrix}$$

where

$$A = \frac{\pi}{32}$$

$$a = \sin\left(\frac{\pi}{4} - \frac{3}{2}\beta\right)$$

$$b = \sin\left(\frac{\pi}{4} - \frac{1}{2}\beta\right)$$

$$c = \sin\left(\frac{\pi}{4} + \frac{1}{2}\beta\right)$$

$$d = \sin\left(\frac{\pi}{4} + \frac{3}{2}\beta\right)$$

Determination of β

In the transform matrix of the OWCT, there is a parameter β dependent on the order N and the adjacent correlation coefficient ρ. It is based on the assumption that the input data are sampled from an one-dimensional, zero-mean, unit-variance first-order Markov process which will be discussed in section 3.4. β is in the range of $0<\beta<\frac{\pi}{N-1}$. Computer searching technique is used to find the optimal value of β for each value of N and ρ under the criteria of Maximum Reducible Bits [33]. The results are listed in Table 3.1 and the basis vectors for the OWCT and DCT-II with N=8 are shown in Figure 3.1. However, the optimal values β under other criteria, such as Residue Correlation [16] and Transform Efficiency [24] etc., are very close to those values shown in Table 3.1. It is also noted that β is not the critical parameter to determine the performance of the OWCT for various criteria.

ρ	0.95	0.90	0.85	0.80
N=8	0.0231	0.0453	0.0663	0.0860
N=16	0.0184	0.0329	0.0435	0.0502

Table 3.1 Best Value of β for the OWCT

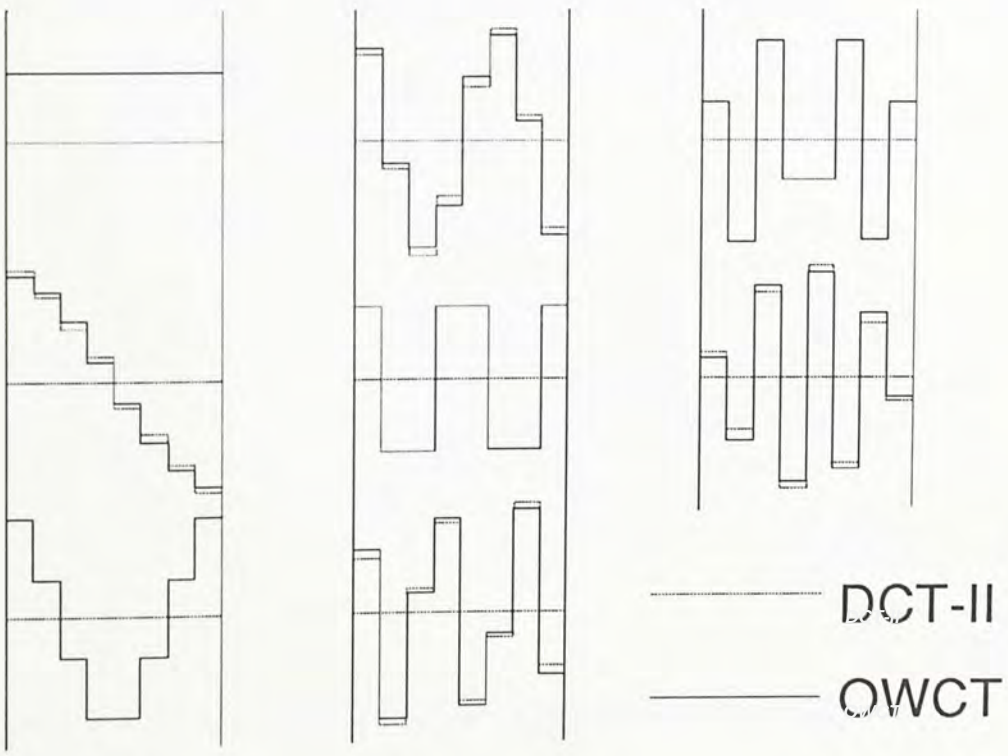


Figure 3.1 Basis Vectors for the OWCT and DCT-II with N=8

3.3 Fast Computational Algorithm for the OWCT

The fast computational algorithm for the OWCT is a little modification of the recursive computational algorithm for the DCT-II defined by Hou [36]. The matrix U can be decomposed as the following

$$[U_M] = [C_M^{IV}]\,[R_M]$$

where M=N/2 and

$$[R_M] = \begin{bmatrix} Q_{M/2} & W_{M/2} \\ -W_{M/2} & Q_{M/2} \end{bmatrix}$$

$$[Q_{M/2}] = \begin{bmatrix} \cos\left(\frac{M-1}{2}\beta\right) & & & & & \frac{1}{\sqrt{2}}\sin\left(\frac{M-1}{2}\beta\right) \\ & \cos\left(\frac{M-2}{2}\beta\right) & & & \frac{1}{\sqrt{2}}\sin\left(\frac{M-2}{2}\beta\right) & \\ & & \ddots & \iddots & & \\ & & \cos\left(\frac{\beta}{2}\right) & \frac{1}{\sqrt{2}}\sin\left(\frac{\beta}{2}\right) & & \\ & & \frac{-1}{\sqrt{2}}\sin\left(\frac{\beta}{2}\right) & \cos\left(\frac{\beta}{2}\right) & & \\ & & \iddots & \ddots & & \\ & \frac{-1}{\sqrt{2}}\sin\left(\frac{M-2}{2}\beta\right) & & & \cos\left(\frac{M-2}{2}\beta\right) & \\ \frac{-1}{\sqrt{2}}\sin\left(\frac{M-1}{2}\beta\right) & & & & & \cos\left(\frac{M-1}{2}\beta\right) \end{bmatrix}$$

$$[W_{M/2}] = \frac{1}{\sqrt{2}} \begin{bmatrix} \sin\left(\frac{M-1}{2}\beta\right) & & & & & \\ & \sin\left(\frac{M-2}{2}\beta\right) & & & & \\ & & \ddots & & & \\ & & \sin\left(\frac{\beta}{2}\right) & & & \\ & & & -\sin\left(\frac{\beta}{2}\right) & & \\ & & & & \ddots & \\ & & & & -\sin\left(\frac{M-2}{2}\beta\right) & \\ & & & & & -\sin\left(\frac{M-1}{2}\beta\right) \end{bmatrix}$$

The C_M^{IV} in equation (2) can be expressed in terms of C_M^{II} as follows :-

$$[U_M] = [C_M^{IV}][R_M]$$

$$= [L_M][C_M^{II}]\ \mathrm{diag}[\cos(\phi_m)][R_M]$$

$$= [L_M][C_M^{II}][\hat{R}_M]$$

where m=0, 1, 2,, ..., M-1

$$\phi_m = \frac{(2m+1)}{4M}$$

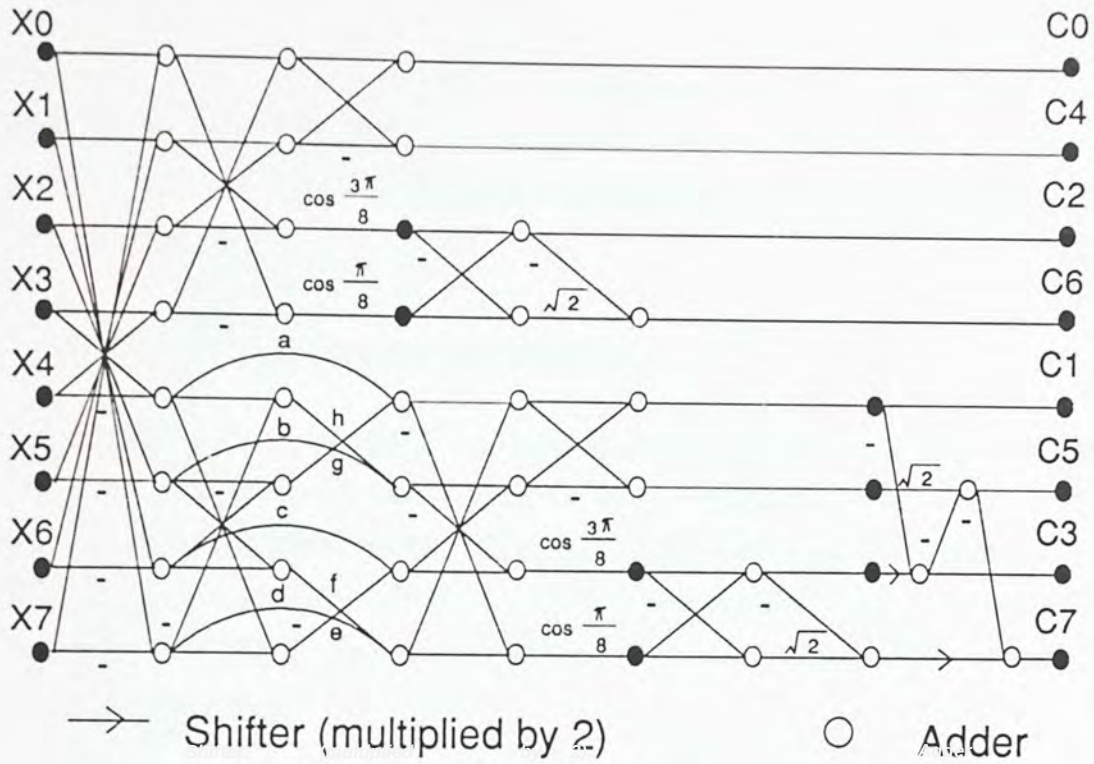
$$[L_M] = \begin{bmatrix} 1 & 0 & 0 & 0 & . & . & . & 0 \\ -1 & 2 & 0 & 0 & . & . & . & 0 \\ 1 & -2 & 2 & 0 & . & . & . & 0 \\ -1 & 2 & -2 & 2 & . & . & . & 0 \\ . & & & & & & & . \\ . & & & & & & & . \\ . & & & & & & & . \\ -1 & 2 & -2 & 2 & . & . & . & 2 \end{bmatrix}$$

and

$$[\hat{R}_M] = \begin{bmatrix} \cos(\phi_0) & R_0 \\ \cos(\phi_1) & R_1 \\ \vdots & \\ \cos(\phi_{M-1}) & R_{M-1} \end{bmatrix}$$

where R_i represents the ith row vector of $[R_M]$

Therefore the order-N OWCT can be decomposed into two order N/2 DCT-II which can be computed using a fast computational algorithm. The signal flow graph for the fast computational algorithm of the OWCT with N=8 is shown in Figure 2. For order N OWCT, it needs N/2 multiplications and N additions more than the DCT-II by using this algorithm. Table 2 and 3 compare the actual number of multiplications and additions being used in the fast computation of the DCT-II and OWCT.



$$a=\cos\frac{7\pi}{16}\cos\frac{\beta}{2}\quad b=\cos\frac{5\pi}{16}\cos\frac{\beta}{2}\quad c=\cos\frac{3\pi}{16}\cos\frac{\beta}{2}\quad d=\cos\frac{\pi}{16}\cos\frac{\beta}{2}$$

$$e=\frac{-1}{\sqrt{2}}\cos\frac{3\pi}{16}\sin\frac{\beta}{2}\quad f=\frac{1}{\sqrt{2}}\cos\frac{\pi}{16}\sin\frac{\beta}{2}\quad g=\frac{1}{\sqrt{2}}\cos\frac{7\pi}{16}\sin\frac{\beta}{2}\quad h=\frac{1}{\sqrt{2}}\cos\frac{5\pi}{16}\sin\frac{\beta}{2}$$

Figure 3.2 Signal Flow Graph of the Fast OWCT for N=8

N	8	16	32	64
DCT-II	12	32	80	192
OWCT	16	40	96	224

Table 3.2 Number of Multiplications

N	8	16	32	64
DCT-II	29	81	209	513
OWCT	37	97	241	577

Table 3.3 Number of Additions

3.4 Performance Evaluation

The performance of the OWCT is evaluated under the assumption that the statistics of the input sequence is a zero-mean, unit-variance first order Markov-I process. The OWCT is tested via various criteria for both one and two dimensional covariance functions.

3.4.1 One Dimensional Covariance Function

Consider an input vector X sampled from an one-dimensional, zero-mean, unit-variance first-order Markov-I process with covariance matrix $[C_x]$. The (i,j) element of the covariance matrix $[C_x]$, says c_{ij}, is equal to $\rho^{|i-j|}$, where ρ is the adjacent element correlation. The vector X is transformed into Y by an unitary transform T, Y = [T] X. The covariance matrix of the vector Y in the transform domain $[C_Y]$ is given as

$$[C_Y] = E[Y\,Y^t]$$

$$= [T]\,[C_X]\,[T]^t$$

$$= \begin{bmatrix} S_{0,0} & \cdots & \cdots & S_{0,N-1} \\ . & \cdots & \cdots & . \\ . & \cdots & \cdots & . \\ S_{N-1,0} & \cdots & \cdots & S_{N-1,N-1} \end{bmatrix}$$

where the superscript t means transpose of the matrix.

3.4.1.1 Residue Correlation

The residue correlation (RC) [16] measures the proportional correlation left in by a sub-optimal transform T and it is given as the ratio of two Hilbert-Schmidt norms

$$RC = \frac{|\,C_X - T\,C_X\,T^{-1}\,|^2}{|\,C_X - I\,|^2}$$

$$= \frac{\left[N\,|\,C_X\,|^2 - \sum_{i=0}^{N-1} |\,s_{ii}\,|^2\right]}{N\,|\,C_X\,|^2 - N}$$

As shown in Figure 3.3, the residue correlations of the OWCT are smaller than those of the DCT-II for N=8 and 16 and for various adjacent correlation coefficients.

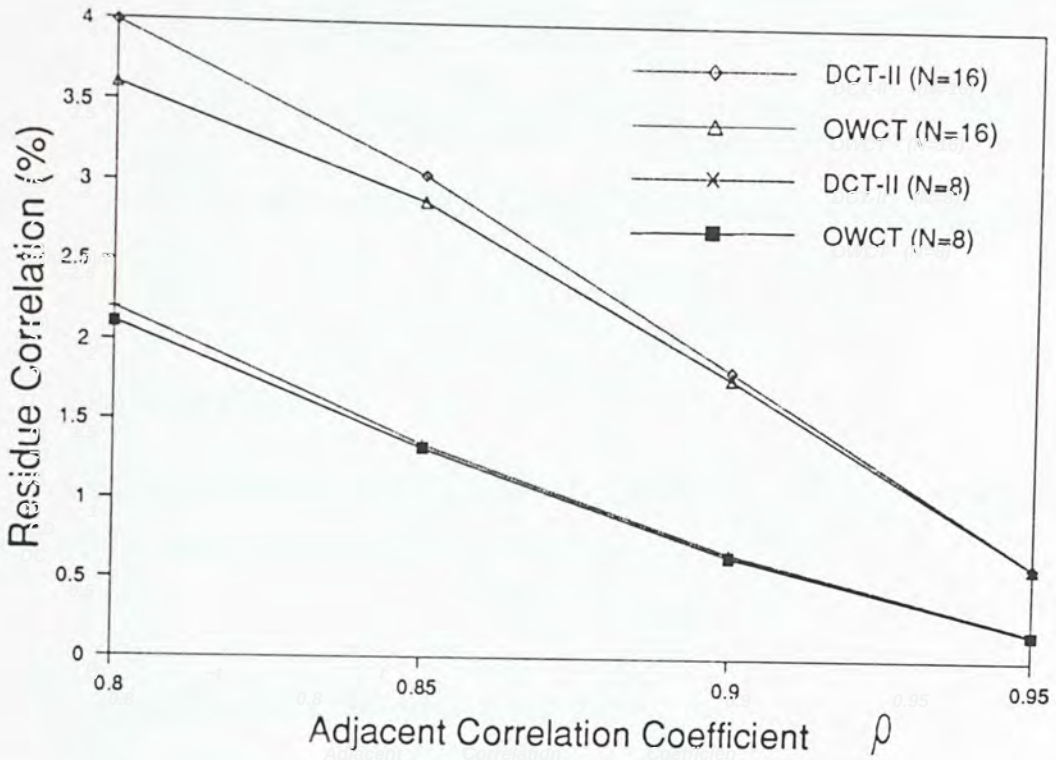


Figure 3.3 Residue Correlation for the OWCT and DCT-II with N=8 and N=16

3.4.1.2 Maximum Reducible Bits

The maximum reducible bits (MRB) [33] of an unitary transform T is defined as

$$\mathrm{MRB} = -\frac{1}{2N} \sum_{i=0}^{N-1} \log_2 s_{ii}$$

This criterion measures the maximum bits which can be reduced from each transform component. The greater the MRB is, the higher compression ability the transform has. Table 3.4 shows the MRB of the DCT-II and OWCT for N=8 and N=16. It can be seen that values of MRB of the OWCT are greater than those of the DCT-II.

ρ		0.80	0.85	0.90	0.95
N=8	OWCT	0.63738	0.80242	1.04284	1.46607
	DCT-II	0.63638	0.80170	1.04244	1.46595
N=16	OWCT	0.68428	0.86073	1.11781	1.57076
	DCT-II	0.68348	0.85996	1.11723	1.57052

Table 3.4 Maximum Reducible Bits for the OWCT and DCT-II with N=8 and N=16

3.4.1.3 Transform Efficiency

The transform efficiency [24] measures the ability of an unitary transform T to decorrelate the input vector X and it is given as

$$\text{Transform Efficiency } (\%) = \frac{\sum_{i=0}^{N-1} |s_{ii}|}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |s_{ij}|} \times 100$$

The transform efficiency versus different correlation coefficients ρ for the DCT-II and OWCT with N=8 and N=16 is shown in Figure 3.4. It can be seen that the transform efficiency of the OWCT is higher than the DCT-II and implies that the OWCT has higher decorrelation ability.

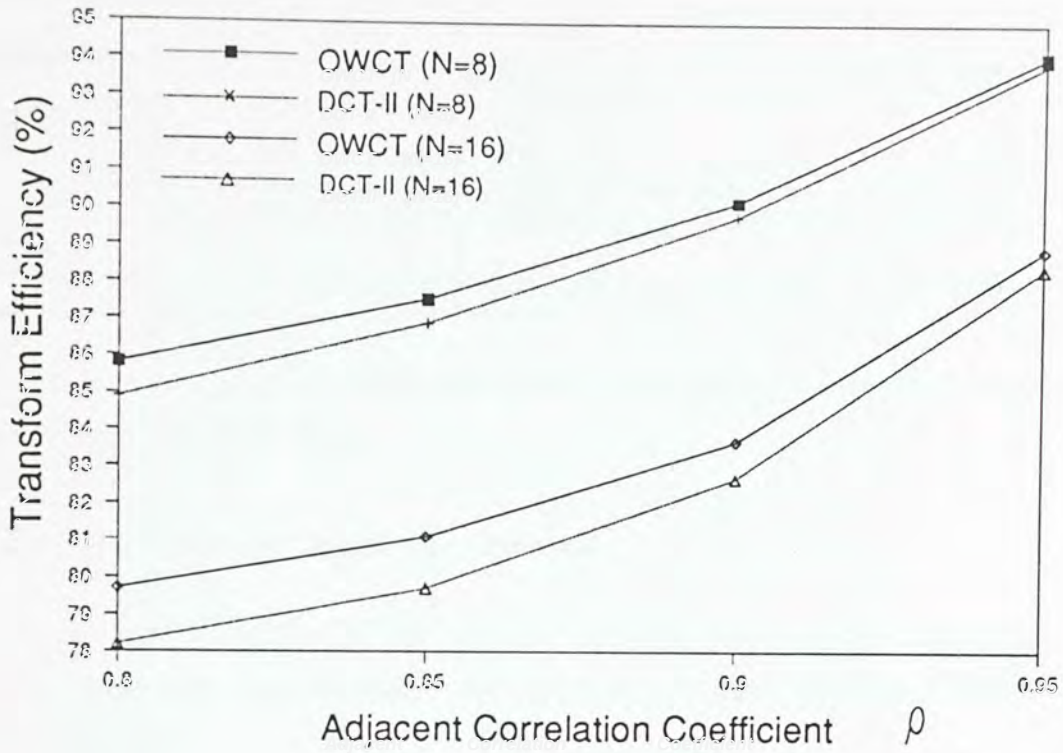


Figure 3.4 Transform Efficiency for the OWCT and DCT-II with N=8 and N=16

3.4.1.4 Mean Square Error (Wiener Filtering)

This criterion measures the performance of an unitary transform T to filter an additive noise (zero mean) from the signal in the transform domain [35]. The mean square error of a scaler Wiener filter is defined as

$$\text{Mean Square Error} = 1 - \frac{1}{N}\sum_{i=0}^{N-1} \frac{s_{ii}^2}{s_{ii}+\phi_{ii}}$$

where ϕ_{ii} represent the i^{th} diagonal element of the transform domain covariance matrix of the noise. Table 3.4 lists the mean square error for the OWCT and DCT-II with N=8 and N=16 based on the signal to noise ratio (SNR), k_o=1. It is shown that the OWCT marginally performs better than the DCT-II in the scaler Wiener filtering application.

ρ		0.80	0.85	0.90	0.95
N=8	OWCT	0.327244	0.294826	0.254529	0.200416
	DCT-II	0.327492	0.294967	0.254583	0.200424
N=16	OWCT	0.314591	0.280082	0.237148	0.178555
	DCT-II	0.314960	0.280411	0.237358	0.178612

Table 3.5 Mean Square Error for the OWCT and DCT-II with N=8 and N=16 based on SNR, K_o=1

3.4.2 Two Dimensional Covariance Function

Let the n by n matrix [X] be a sample of a two-dimensional zero mean, unit variance, stationary, non-seperable and isotropic Markov random process with covariance function

$$f_{ijpq} = E[X_{i,j}, X_{p,q}] = \rho^{\sqrt{(i-p)^2+(j-q)^2}}$$

where ρ is the adjacent element correlation coefficient in vertical and horizontal directions. The matrix [X] is transformed into [C] by a transform T. Variance σ_{uv} of C_{uv} can be derived from f_{ijpq} and T.

3.4.2.1 Basis Restriction Error

Let Ω be the set containing K index pairs (u,v) corresponding to the largest K σ_{uv}. The basis restriction error is defined as

$$e(K) = 1 - \left\{ \frac{\sum\limits_{u,v \in \Omega} \sum \sigma_{u,v}^2}{\sum\limits_{u=0}^{N-1} \sum\limits_{v=0}^{N-1} \sigma_{u,v}^2} \right\}$$

The basis restriction errors for the DCT-II and OWCT for N=8 are very close and the results are listed in Table 3.6.

K	1	2	3	4	5	6	7	8
OWCT	0.18705	0.13813	0.08920	0.07643	0.06681	0.05719	0.05218	0.04717
DCT-II	0.18705	0.13810	0.08915	0.07641	0.06679	0.05717	0.05218	0.04718

9	10	11	12	13	14	15	16	17
0.04402	0.04088	0.03805	0.03588	0.03371	0.03214	0.03057	0.02908	0.02759
0.04405	0.04092	0.03809	0.03591	0.03373	0.03216	0.03058	0.02909	0.02759

Table 3.6 Basis Restriction Error versus k, no. of Coefficients Retained, for the OWCT and DCT-II with N=8 and ρ=0.95

In the above paragraph, numerical results obtained from computer simulation have shown that the OWCT performs better than the DCT-II for N=8 and N=16 under different criteria. In the transform matrix of the OWCT, there is a parameter β dependent on the order N and the adjacent correlation coefficient ρ. In general cases, the adjacent correlation of a typical picture is close to unity. In practical applications, we may fix β at a suitable value obtained from larger correlation coefficients and will get a satisfactory result for a wide range of ρ. For example, if we fix $\beta = 0.0453$, the OWCT will have higher transform efficiency than those of the DCT-II over the range of $0.7 < \rho < 0.95$ as shown in Figure 3.5.

3.5 Conclusion

A new transform, the Odd Weighted Cosine Transform (OWCT), is proposed. The OWCT is obtained by weighting some kernel components in the odd part of the transform matrix of the DCT-II. It is shown that the performance of the OWCT is slightly better than the DCT-II under different well known criteria. A fast computational algorithm for the OWCT has been derived and it was found that hardware implementation of the transform is similar to that of the DCT-II with more multipliers and adders required. To speed up the transformation process, fast computational algorithms are necessary for the orthogonal transforms. The DCT which involves only real numbers can be regarded as a particular case of the Discrete Fourier Transform (DFT). The Discrete W Transform (DWT) which also involves only real numbers can be used to compute DFT with computational advantages. In the next chapter, a new fast recursive algorithm is derived for all types of the radix-2 DWT.

3.6 Note on Publication

A paper based on all the results described in this chapter and entitled 'Odd Weighted Cosine Transform' is accepted for presentation at the International Conference on Image Processing, Singapore, Oct. 1989. This paper is jointly authored with Dr. W.K. Cham.

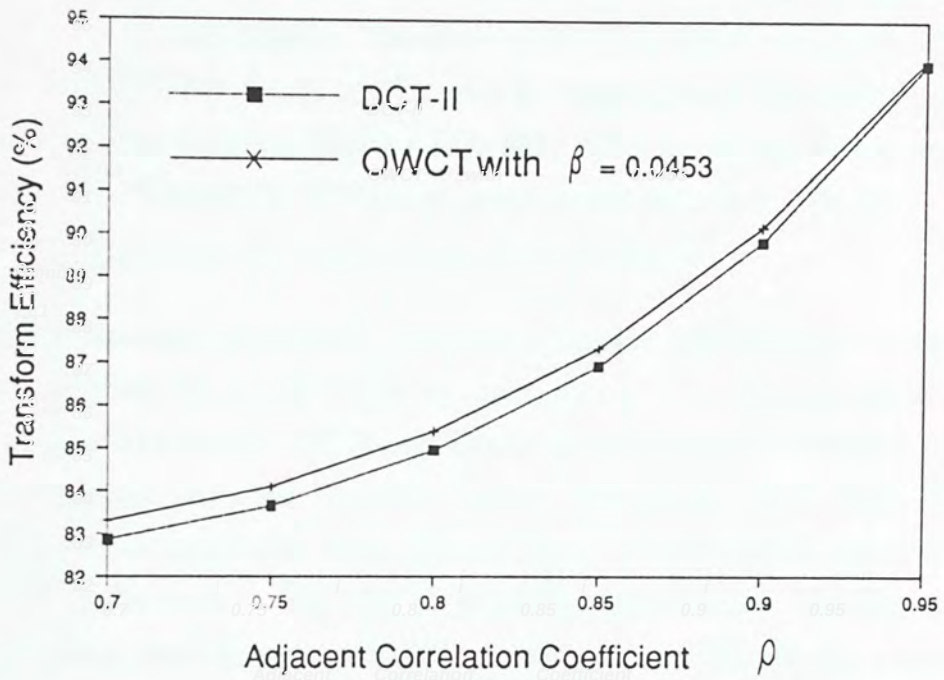


Figure 3.5 Transform Efficiency for the DCT-II and OWCT with β fixed at 0.0453 and N=8

CHAPTER 4 A FAST RECURSIVE ALGORITHM FOR THE DISCRETE W TRANSFORM

4.1 Introduction

The Discrete W Transform (DWT) defined by Wang [22] is a real approach for the spectral analysis. There are four types of the DWT and the type I DWT is also called the Discrete Hartley Transform (DHT) by Bracewell [26]. In the past few years, the DWT has found applications in digital image and signal processing, data compression and digital filtering [27] [28]. This is because it can replace the Discrete Fourier Transform (DFT) in some cases and it has a FFT like fast computational algorithm and only requires real operations.

There are several algorithms developed for the computation of the type I DWT and they can be grouped into two categories : (a) decomposition into the Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST) [22] [40], and (b) in relation with the Discrete Fourier Transform (DFT) [38] [39]. Fast algorithms of these categories were derived from the fast algorithms of the DCT, DST and DFT. Recently, Wang [42] proposed an algorithm to calculate the DWT based on its own internal properties. This algorithm requires the least numbers of multiplications and additions but, unfortunately, it requires secant multipliers and so is numerically unstable when the order is large. In this chapter, fast and numerically stable recursive algorithms are developed for all four types of the radix-2 DWT based on their intrinsic characteristics. In section 4.2 of this chapter, the symmetry properties of the DWT are analysed using the principle of Dyadic Symmetry [24] and the fast algorithms are developed in section 4.3. The computational requirements of the fast type I DWT algorithm are compared with other existing algorithms in section 4.4 and the final conclusion will be drawn in last section.

4.2 Symmetry Properties of the Discrete W Transform

The four types of the DWT matrices are defined as follows

$$[W_N^I] = \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + ij\frac{2\pi}{N}\right)\right]$$

$$[W_N^{II}] = \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + i\left(j+\frac{1}{2}\right)\frac{2\pi}{N}\right)\right]$$

$$[W_N^{III}] = \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + \left(i+\frac{1}{2}\right)j\frac{2\pi}{N}\right)\right]$$

$$[W_N^{IV}] = \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + \left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{2\pi}{N}\right)\right]$$

for i,j = 0, 1, 2, ..., N-1

where the superscript represents the type and the subscript indicates the order. The type I (DWT-I) is also called the Discrete Hartley Transform (DHT).

Theorem 4.1 : The order N DWT-I has the (N/2)th dyadic symmetry. The even part of the basis vectors has the even type and the odd part has the odd type.

[Proof] Let $W_N^I(i,j)$ represents the j^{th} element of the i^{th} basis vector of the DWT-I matrix.

$$j \oplus N/2 = N/2 + j \qquad \text{for } j = 0, 1, 2, ..., N/2 - 1$$

When i is even

$$W_N^I(2i, j\oplus N) = \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + 2i\left(j+\frac{N}{2}\right)\frac{2\pi}{N}\right)\right]$$

$$= \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + (iN+2ij)\frac{2\pi}{N}\right)\right]$$

$$= \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4} + 2ij\frac{2\pi}{N}\right)\right]$$

$$= W_N^I(2i, j)$$

When i is odd

$$W_N^I(2i+1, j \oplus N) = \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4}+(2i+1)\left(j+\frac{N}{2}\right)\frac{2\pi}{N}\right)\right]$$

$$= \sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4}+(2i+1)j\frac{2\pi}{N}+2i\pi+\pi\right)\right]$$

$$= -\sqrt{\frac{2}{N}}\left[\sin\left(\frac{\pi}{4}+(2i+1)j\frac{2\pi}{N}\right)\right]$$

$$= -W_N^I(2i+1, j)$$

Theorem 4.2 : The order N DWT-II has the $(N/2)^{th}$ dyadic symmetry. The even part of the basis vectors has the even type and the odd part has the odd type.

[Proof] similar to that of theorem 4.1.

4.3 The Fast Recursive Algorithm for the DWT

Let { x(j) : j=0, 1, ..., N-1 } be the input sequence and { y(i) : i=0, 1, ..., N-1 } be the DWT-I transform coefficient sequence. We have

$$y(i) = \sqrt{\frac{2}{N}}\sum_{j=0}^{N-1} x(j)\sin\left(\frac{\pi}{4}+ij\frac{2\pi}{N}\right)$$

Divide y(i) into even and odd parts, then

$$y(2i) = \sqrt{\frac{2}{N}}\sum_{j=0}^{N-1} x(j)\sin\left(\frac{\pi}{4}+2ij\frac{2\pi}{N}\right) \qquad (4.1.a)$$

$$y(2i+1) = \sqrt{\frac{2}{N}}\sum_{j=0}^{N-1} x(j)\sin\left(\frac{\pi}{4}+(2i+1)j\frac{2\pi}{N}\right) \qquad (4.1.b)$$

for i = 0, 1, 2, ..., N/2 - 1

By theorems 1 and 2, equations (4.1.a) and (4.1.b) can also be written in following form :

$$y(2i) = \sqrt{\frac{2}{N}}\sum_{j=0}^{N/2-1} r(j)\sin\left(\frac{\pi}{4}+ij\frac{2\pi}{N/2}\right) \qquad (4.2.a)$$

$$y(2i+1) = \sqrt{\frac{2}{N}} \sum_{j=0}^{N/2-1} s(j) \sin\left(\frac{\pi}{4} + \left(i+\frac{1}{2}\right) j \frac{2\pi}{N/2}\right) \tag{4.2.b}$$

$$\text{for } i = 0, 1, 2, \ldots, N/2 - 1$$

where

$$r(j) = x(j) + x\left(j+\frac{N}{2}\right)$$

$$s(j) = x(j) - x\left(j+\frac{N}{2}\right)$$

Equation (4.2.a) is in fact the order N/2 DWT-I of sequence r(j) and equation (4.2.b) is the order N/2 DWT-III of the sequence r(j). Equation (4.2) can be written in matrix form as in equation (4.3.a).

$$[W_N^I] = [P_N] \begin{bmatrix} W_{N/2}^I & 0 \\ 0 & W_{N/2}^{III} \end{bmatrix} [B_N] \tag{4.3.a}$$

where

$$[B_N] = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

$$[P_N]^t = \begin{bmatrix} 1 & 0 & 0 & . & . & . & . & . & . \\ 0 & 0 & 1 & 0 & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & 0 & 1 & 0 \\ 0 & 1 & 0 & . & . & . & . & . & . \\ 0 & 0 & 0 & 1 & 0 & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . & 0 & 1 \end{bmatrix}$$

I_N is the order N identity matrix.

By the same approach, we have the following expression for the DWT-II and DWT-III

$$[W_N^{II}] = [P_N] \begin{bmatrix} W_{N/2}^{II} & 0 \\ 0 & W_{N/2}^{IV} \end{bmatrix} [B_N] \tag{4.3.b}$$

and

$$[W_N^{III}] = [W_N^{II}]^t$$

$$= [B_N]\begin{bmatrix} W_{N/2}^{III} & 0 \\ 0 & W_{N/2}^{IV} \end{bmatrix}[P_N]^t \qquad (4.3.c)$$

Equations (4.3.a), (4.3.b) and (4.3.c) imply that fast computational algorithms of $[W_N^I]$, $[W_N^{II}]$ and $[W_N^{III}]$ can be derived from the fast computational algorithm of $[W_N^{IV}]$. The i^{th} DWT-IV transform coefficient of input sequence { x(j) } is

$$y(i) = \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} x(j) \sin\left(\frac{\pi}{4} + (2i+1)\phi_j\right) \qquad (4.4)$$

$$\text{for } i,j = 0, 1, 2, \ldots, N-1$$

where

$$\phi_j = \frac{(2j+1)\pi}{2N}$$

By the following identity

$$\sin\left[\frac{\pi}{4} + (2i+1)\phi_j\right] = 2\sin\left[\frac{\pi}{4} + (2i)\phi_j\right]\cos\phi_j - \sin\left[\frac{\pi}{4} + (2i-1)\phi_j\right]$$

equation (4.4) can be written as

$$y(i) = \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} x(j)\left\{2\sin\left[\frac{\pi}{4} + (2i)\phi_j\right]\cos\phi_j - \sin\left[\frac{\pi}{4} + (2i-1)\phi_j\right]\right\} \qquad (4.5)$$

equation (4.5) can also be expressed in the following matrix form

$$[W_N^{IV}] = [L_N]\,[W_N^{II}]\,[R_N] \qquad (4.6)$$

where

$$[L_N] = \begin{bmatrix} 1 & 0 & 0 & 0 & . & . & . & 0 \\ -1 & 2 & 0 & 0 & . & . & . & 0 \\ 1 & -2 & 2 & 0 & . & . & . & 0 \\ -1 & 2 & -2 & 2 & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ -1 & 2 & -2 & 2 & . & . & . & 2 \end{bmatrix}$$

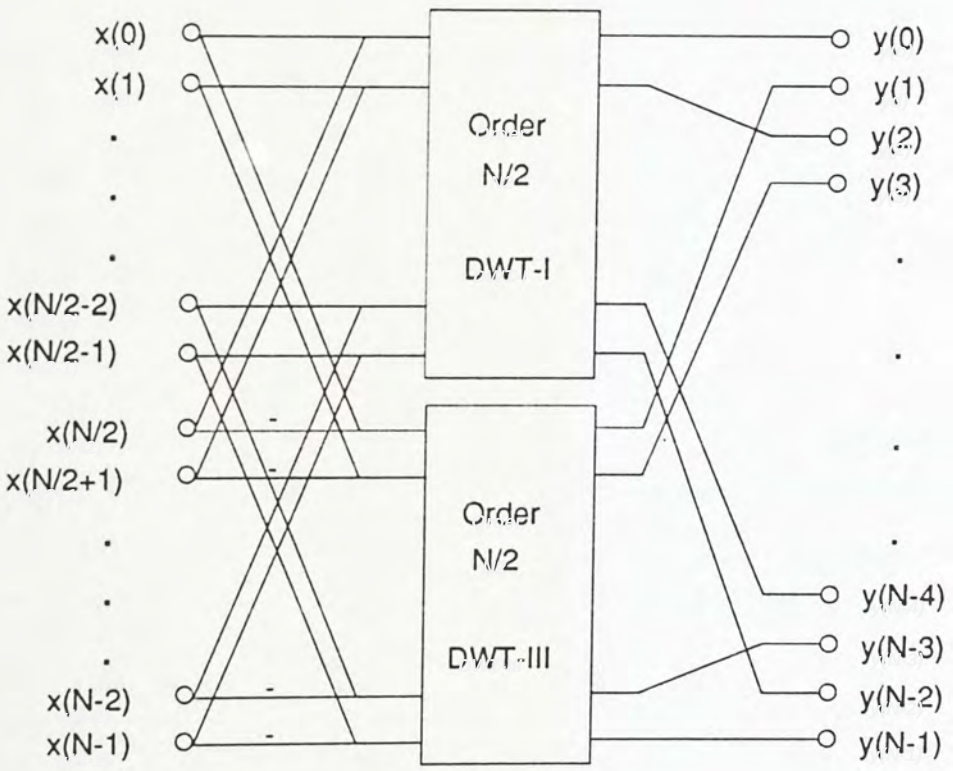
$$[R_N] = diag\{\cos\phi_j\}$$

Equation (4.6) shows the relationship between the DWT-IV and DWT-II. It is noted that elements in $[L_N]$ are only 0, ±1, ±2, so it can be implemented by using adders and shifters only. By utilizing equations (4.3) and (4.6), all four types order 2^m DWTs can be generated from the order 2 DWTs, which are :

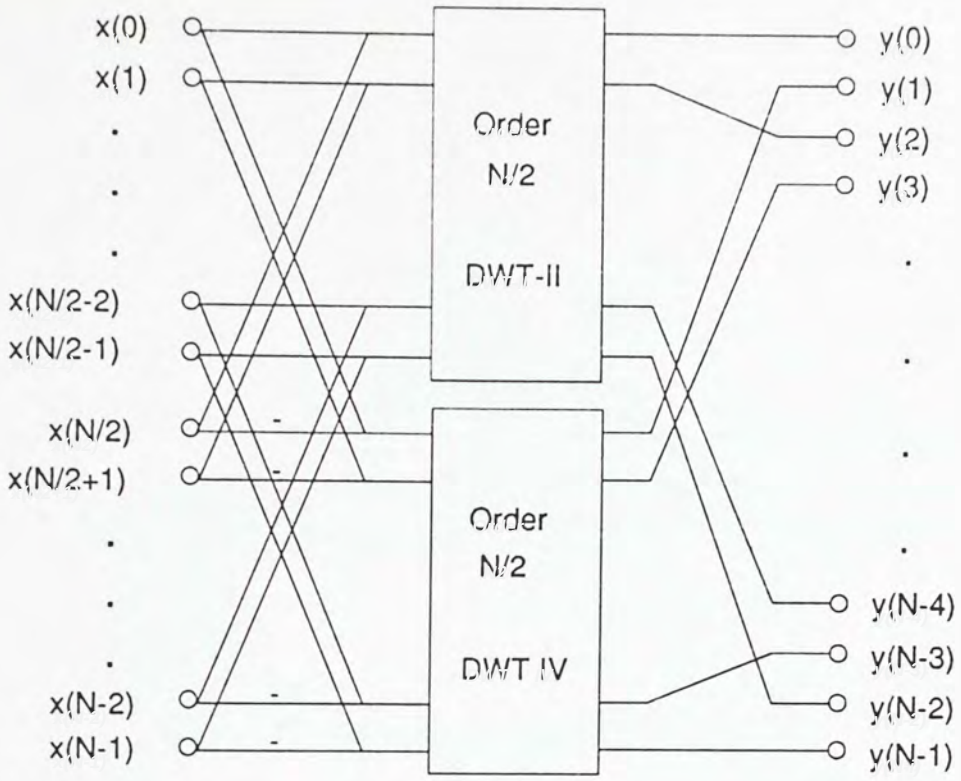
$$[W_2^I] = [W_2^{II}] = [W_2^{III}] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$[W_2^{IV}] = \frac{1}{\sqrt{2}}\begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

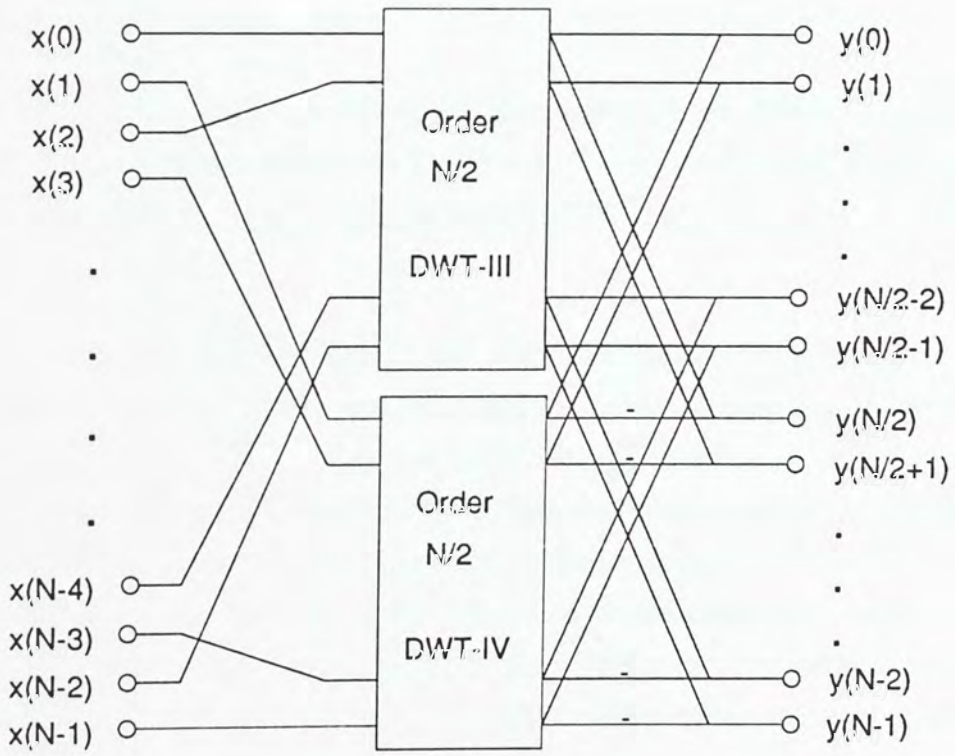
The following figures show the signal flow diagrams of the recursive algorithms for the order N DWT matrices. It should be noted that each y(i) of all signal flow diagrams should be multiplied by a normalizing constant $1/\sqrt{N}$.



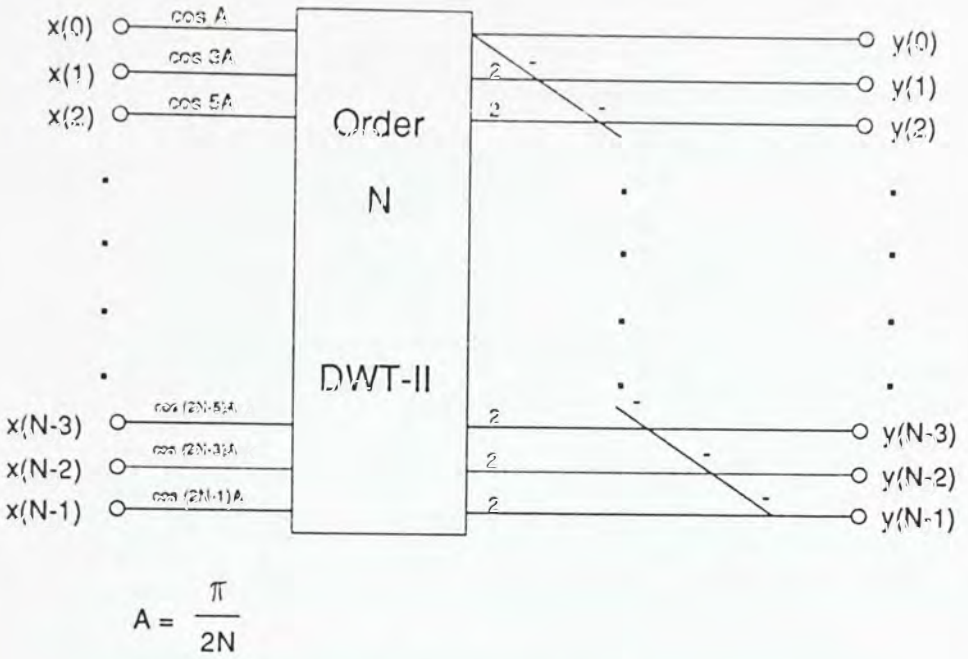
(a) Order N DWT-I



(b) Order N DWT-II



(c) Order N DWT-III



(d) Order N DWT-IV

Figure 4.1 The Flow Diagram for All Four Type Radix-2 DWT Matrices

4.4 Computational Requirement

Computational requirements of all four types of the DWT using the above recursive algorithms are shown in Table 4.1. Table 4.2 compares the number of multiplications and additions used in the fast DWT-I algorithm to the other known fast DWT-I algorithms.

The numbers of multiplications and additions required in this algorithm is the same as the Wang's algorithm and less than the other existing algorithms. Wang's algorithm, however, consists of secant multipliers. The round off error of the finite length representation of the secant multipliers increases with the transform order since the range of a secant multiplier is from one to infinity. Therefore, longer bit length should be provided for larger transform order, otherwise very large error may occur. Wang's algorithm is suitable for small transform orders only. When compared with the Wang's method, the present algorithm requires a few more binary shifts but it avoids numerically unstable since it consists of only the cosine multipliers. As the present algorithm has simple structure, it is conceived that the algorithm is also suitable for VLSI implementation.

	Multiplications	Additions
DWT-I	$\frac{N}{2}(M-3)+2$	$\frac{N}{2}(3M-5)+6$ *for* $N \geq 4$
DWT-II	$\frac{N}{2}(M-1)$	$\frac{3}{2}N(M-1)$ *for* $N \geq 4$
DWT-III	$\frac{N}{2}(M-1)$	$\frac{3}{2}N(M-1)$ *for* $N \geq 4$
DWT-IV	$\frac{N}{2}(M+1)$	$\frac{N}{2}(3M+1)$ *for* $N \geq 4$

Table 4.1 Computational Requirements for the Recursive Algorithm for the DWT (where $M = \log_2 N$)

Order N	This Method		Wang [42]		Kwong [41]		Suehiro [40]		Sorensern [39]	
	N_M	N_A	N_M	N_A	N_M	N_A	N_M	N_A	N_M	N_A
4	0	8	0	8	0	8	0	8	0	8
8	2	22	2	22	4	26	2	26	2	22
16	10	62	10	62	20	74	10	72	12	64
32	34	166	34	166	68	194	34	194	42	166
64	98	422	98	422	196	482	98	482	124	416
128	258	1030	258	1030	516	1154	258	1154	330	998

Table 4.2 Computation Comparison of Different Algorithm for Calculating DWT-I (DHT) (where N_M is the no. of multiplications and N_A is the no. of additions)

4.5 Conclusion

The symmetry properties of the DWT matrices are analysed using the dyadic symmetry. A new recursive algorithm for calculation all four types radix-2 DWT's is developed based on these internal properties. The new algorithm with a few more shift registers needed requires less multiplications and additions than other existing algorithms. This algorithm avoids the problem of numerical unstable from Wang's method with finite length arithmetic and is suitable for computation of DWT of

large orders. Apart from fast computational algorithms, a dedicated hardware or integrated circuit is also an efficient method to implement the transformation process. In the next chapter, a LSI implementation of the Integer Cosine Transform (ICT) by using the ASIC technology will be described.

4.6 Note on Publication

A paper based on all the results reported in this chapter and entitled 'A New Fast Recursive Algorithm for the Discrete W Transform' was submitted to the Proceeding of International Symposium on Computer Architecture and Digital Signal Processing, Hong Kong, Oct. 1989. This paper is jointly authored with Dr. W.K. Cham.

CHAPTER 5 LSI IMPLEMENTATION OF THE INTEGER COSINE TRANSFORM

5.1 Introduction

The adaptive transform coding system proposed by Chen & Smith [30] was found to be one of the most effective system in the image data compression. The Discrete Cosine Transform (DCT) [13] has been most widely used in those systems for it is asymptotically equivalent to the Karhunen-Loeve Transform (KLT) which is statistically optimal in data compression ability.

Recently, due to the fast development of the VLSI technologies and the continuous reduction of hardware cost, many research works are extended to the implementation of the DCT in areas of hardware realization and processing architecture [43]. Since the basis vector components of the DCT are mainly real numbers, it is found that the implementation of the DCT in finite length arithmetic is more complicated than those transforms whose basis vector components are integers only, i.e. the Walsh transform, Slant transform, CMT and HCT. Unfortunately, none of them have the same data compression ability as the DCT. Cham [2] generated a transform called Integer Cosine Transform (ICT) by replacing the DCT kernel with integers based on the principle of dyadic symmetry. The ICT is found to have virtually the same compression ability as the DCT and has a simpler implementation than the DCT in finite length arithmetic. Choy & Cham [44] implemented the ICT in a LSI chip by using ASIC technology. However, the speed of this ICT chip is slow and the coding error is large. In this chapter, we introduce a modified structure for implementation of the ICT. The major difference is the selection of the transform matrix in the ICT chip and the input and output pins consideration. Due to the limited facilities, the new ICT processor is developed by using a low cost 3 μm gate array with 2800 gates from Micro Circuit Engineering.

The Integer Cosine Transform will first be introduced in section 5.2. The design considerations and the detailed description of the architecture of the ICT will be given in section 5.3 and 5.4. Section 5.5 will describe some typical applications of the ICT chip and the overall conclusion will be drawn in the last section.

5.2 The Integer Cosine Transform (ICT)

For order N DCT, the (i,j) element of the transform matrix, $T_N(i,j)$, is given by

$$T_N(i,j) = \begin{cases} \dfrac{1}{\sqrt{N}} & \text{for } i=0 \\ \dfrac{2}{\sqrt{N}} \cos\left\{ \dfrac{(2i+1)j}{N} \right\} & \text{for } i \in [0, N-1] \end{cases}$$

The kernel of the order-8 Integer Cosine Transform (ICT) is generated by replacing the basis vector components of the order-8 DCT with variables as follows

$$[T] = \begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (b & -d & -a & -c & c & a & d & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & -a & d & b & -b & -d & a & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -c & b & -a & a & -b & c & -d) \end{bmatrix} = \begin{bmatrix} k_0 & J_0^t \\ k_1 & J_1^t \\ k_2 & J_2^t \\ k_3 & J_3^t \\ k_4 & J_4^t \\ k_5 & J_5^t \\ k_6 & J_6^t \\ k_7 & J_7^t \end{bmatrix}$$

$$= [K][J] \qquad (5.1)$$

where $[K] = \text{diag}\{k_0, k_1, ..., k_7\}$ is a scaling matrix, J_i is the column vector of matrix [J] and $k_i J_i = 1$ for $i \in [0, 7]$.

There are infinite set of ICTs since there are infinite sets of integers (a,b,c,d,e,f) satisfy the following condition for the ICT :

i) $a\,b = a\,c + b\,d + c\,d$

ii) $a > b > c > d$ and $e > f$

The following table shows the performance of several order 8 ICTs based on the criterion of transform efficiency [24] which measures the decorrelation ability of the transform.

Transform or ICT(a,b,c,d,e,f)	Transform Efficiency (%)
ICT (230,201,134,46,3,1)	90.221
ICT (120,105,70, 24,3,1)	90.219
ICT (55, 48, 32, 11,3,1)	90.213
ICT (10, 9, 6, 2, 3, 1)	90.176
ICT (5, 3, 2, 1, 3, 1)	81.051
DCT	89.836
CMT	86.785
Slant Transform	85.842
HCT	84.097
Walsh Transform	77.140

Table 5.1 Transform Efficiency for various transforms with the adjacent correlation coefficient ρ equal to 0.9

5.3 Design Considerations

5.3.1 Specification

With an input vector X, the one-dimensional transform coefficient vector C of the ICT is given by

$$\begin{aligned} C &= [T]X \\ &= [K][J]X \\ &= [K]C_I \end{aligned} \qquad (5.2.a)$$

and the inverse

$$\begin{aligned} X &= [T]^t C \\ &= [J]^t [K] C \\ &= [J]_t C_K \end{aligned} \qquad (5.2.b)$$

With an input matrix [X], the two dimensional ICT transform coefficient matrix [C] is then given by

$$\begin{aligned} [C] &= [T][X][T]^t \\ &= [K][J][X][J]^t[K] \\ &= [K][C_I][K] \end{aligned} \qquad (5.3.a)$$

and the inverse

$$\begin{aligned}[X] &= [T]^t [X] [T] \\ &= [J]^t [K] [C] [K] [J] \\ &= [J]^t [C_K] [J]\end{aligned} \tag{5.3.b}$$

The objective of the ICT chip is to perform the following one-dimensional forward and inverse ICT transform : $C_I = [J]\ X$ and $X = [J]^t\ C_K$. Multiplication of the normalization matrix [K] which has not been implemented in the ICT chip is expected to be carried out by an external processor. In a transform coding system, the normalization process can be absorbed into the quantization process. Due to the limitation of the I/O pins and gate size of one chip, we only implement one of the row multiplication (or column in inverse) of the whole transform matrix in one single chip. Therefore, the functions of one ICT chip are as follows :

forward transform

$$Y_{Ii} = \sum_{j=0}^{7} J_{ij} X_j \tag{5.4.a}$$

inverse transform

$$X_i = \sum_{j=0}^{7} J_{ji} Y_{Kj} \tag{5.4.b}$$

where $i \in [0,7]$

5.3.2 Selection of the Transform Matrix

For the transform matrix of the ICT shown in equation (5.1), we select a=10, b=9, c=6, d=2, e=3, f=1 as the transform matrix to be implemented in our ICT chip for two reasons. The first one is that the ICT(10,9,6,2,3,1) has a better transform efficiency than the DCT as shown in Table 5.1. The other reason is that the integer values of the basis vector components (multipliers) are quite small and can be easily implemented by a few number of binary shifts (at most 4) and additions.

There still have some modifications for the transform matrix before realization due to the finite length considerations. For the 1-D forward transform, there is no error occurred when finite length representation of the input data since all of them are integers. But for the inverse transformation, truncation error will occur because the elements of the input vector C_{Ki} are real numbers instead of integers. This

truncation error will also propagate to the output and affect the accuracy of the output data and it is affected by the value of the normalization matrix [K] and the component values of the basis vectors. For the ICT (10,9,6,2,3,1), the normalization matrix [K] is

$$[K] = diag\{\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{442}}, \frac{1}{\sqrt{40}}, \frac{1}{\sqrt{442}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{442}}, \frac{1}{\sqrt{40}}, \frac{1}{\sqrt{442}}$$

$$= diag\left\{\frac{1}{|J_0|}, \frac{1}{|J_1|}, \frac{1}{|J_2|}, \frac{1}{|J_3|}, \frac{1}{|J_4|}, \frac{1}{|J_5|}, \frac{1}{|J_6|}, \frac{1}{|J_7|}\right\}$$

Some integer constants should be included in the matrix [J] so that all $|J_i|$ are of similar magnitudes.

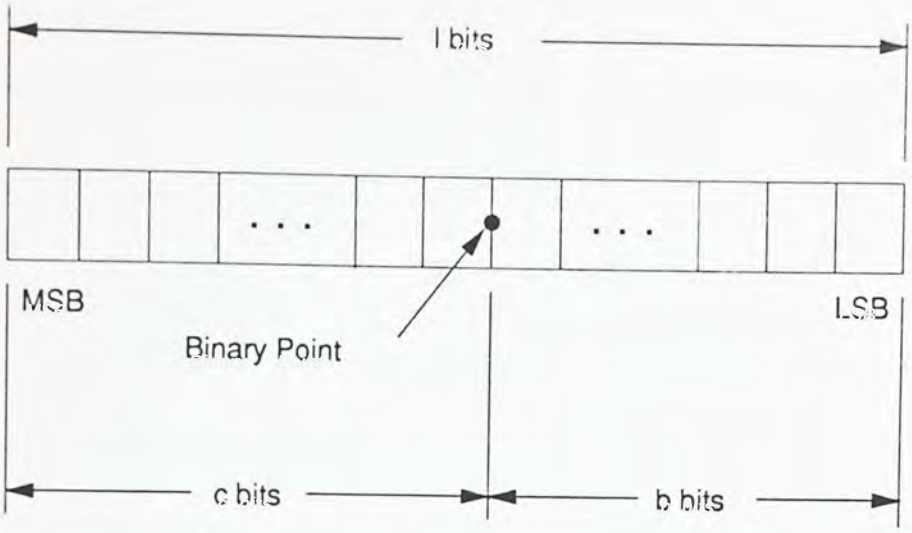
$$[\hat{J}] = \begin{bmatrix} R & R & R & R & R & R & R & R \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 3M & M & -M & -3M & -3M & -M & M & 3M \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ R & -R & -R & R & R & -R & -R & R \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ M & -3M & 3M & -M & -M & 3M & -3M & M \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix}$$

and

$$[\hat{K}] = diag\left\{\frac{1}{R\sqrt{8}}, \frac{1}{\sqrt{442}}, \frac{1}{M\sqrt{40}}, \frac{1}{\sqrt{442}}, \frac{1}{R\sqrt{8}}, \frac{1}{\sqrt{442}}, \frac{1}{M\sqrt{40}}, \frac{1}{\sqrt{442}}\right\}$$

where R and M are integers.

The finite length representation of a number is as shown as follows :



The truncation error e_T is in the range $(-\frac{1}{2}.2^b, +\frac{1}{2}.2^b)$. The maximum possible values of C_{Ki} of the two transform matrix are as shown in the following table with the assumption that the image data is in the range of $0 \le X_i \le 255$.

i	0	1	2	3	4	5	6	7
[J]	255	15.58	51	15.58	127.5	15.58	51	15.58
[Ĵ]	255/R	15.58	51/M	15.58	127.5/R	15.58	51/M	15.58

Table 5.2 Maximum Possible Value of C_{Ki} for Different Transform Matrices (MAX $[C_{Ki}]$)

The number of bits required to represent the magnitude before the binary point is

$$c = \text{Int}\,[\,\log_2(\,\text{MAX}\,[\,C_{Ki}\,]\,)] \qquad \text{for } i \in [0,7]$$

where Int[X] represents the smallest integer greater than X.

For the modified transform matrix, if R=8, M=4 then c=4, and if R=8, M=3 then c=5. Compared with c=8 for the original one, the modified transform has smaller truncation error since b=l-c and e_T is in the range $(-\frac{1}{2}.2^b, +\frac{1}{2}.2^b)$.

Finally we select the modified transform matrix with R=8 and M=3 to be implemented in the ICT chip. We choose M=3 because the multiplier is simpler than that of M=4 although the truncation error is slightly larger. Then the transform matrix implemented in the ICT processor is shown below

$$[\bar{J}] = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 9 & 3 & -3 & -9 & -9 & -3 & 3 & 9 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 3 & -9 & 9 & -3 & -3 & 9 & -9 & 3 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix}$$

5.3.3 I/O Pins Considerations

Although the ICT chip is designed to perform the 1-D transform, it is also desired to perform the 2-D transform by using the same chip. Therefore the bit length of the input and output data should be equal since the output of one chip will be connected to the input of another chip when performing 2-D transform according to equations (5.3.a) and (5.3.b). On the other hand, there should have 6 more bits for the output data than the input due to the internal multiplication and addition operations. Therefore, a truncation process is carried out before latching out the output data as shown in Figure 5.1. In our design, the data format of the input and output are also 14 bits length including one sign bit and thirteen magnitude bits. The 6 least significant bits of the resulted data after the row (or inverse) multiplication are truncated, so that the output keep in 14 bits length. With this arrangement, there would not have any error when performing the 1-D forward transform since the input image data is 8 bits length (assume in the range of [-128, 127]). The truncation error for the 2-D forward and inverse transform are evaluated based on several MSE criteria by computer simulation method for several pictures with size of 256 x 256 pixels as shown in Figure 5.7.

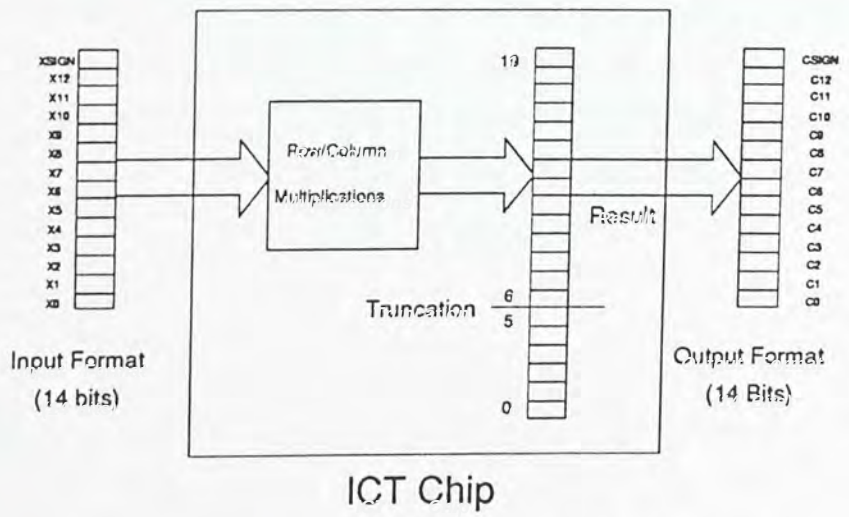


Figure 5.1 Truncation Process of the ICT Chip

5.3.3.1 2-D Forward ICT

The 2-D forward ICT given in Equation (5.3.a) can be implemented by the ICT chips and the block diagram is shown in Figure 5.2.

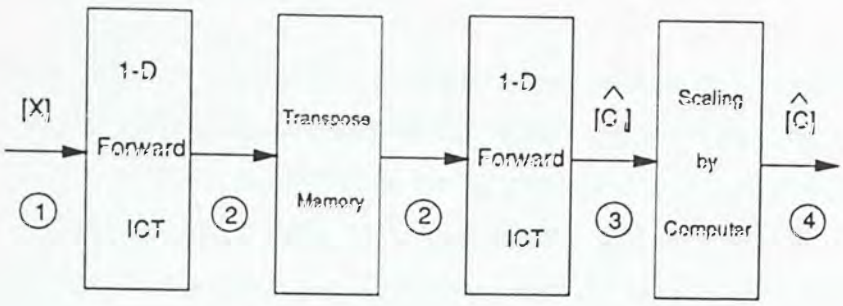


Figure 5.2 2-D Forward ICT

In the above diagram, the elements of matrix [X] and $[\hat{C}]$ are both in 14-bit format. The scaling process is done by computer software so that the elements of matrix in $[\hat{C}]$ are real numbers. If the image data are in the range of [-128, 127], then the input image data in stage 1 of Figure 5.2 should be represented in the following 14-bit format :

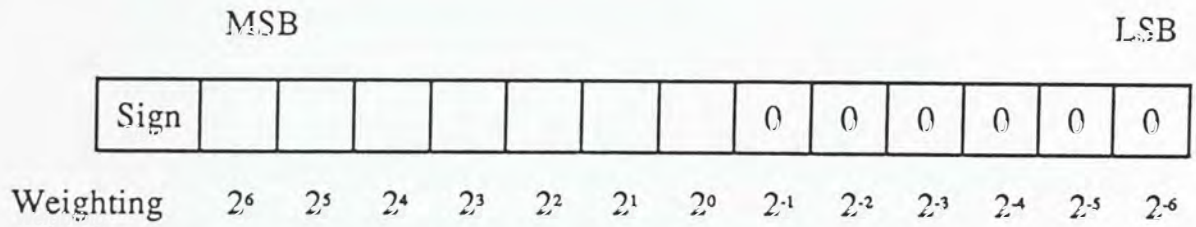


Figure 5.3 Format of Elements in Stage 1

When the input data pass through a 1-D forward ICT, the weight of the MSB of the output data (in stage 2 of Figure 5.2) will be increased by 6. Finally, the data format of the non-scaling 2-D transform coefficients in stage 3 will have the weighting of 2^{18} at MSB and 2^6 at LSB as shown in Figure 5.4. The truncation error introduced in this 2-D forward process in the ICT chip will be evaluated by the quantity of Error 1.

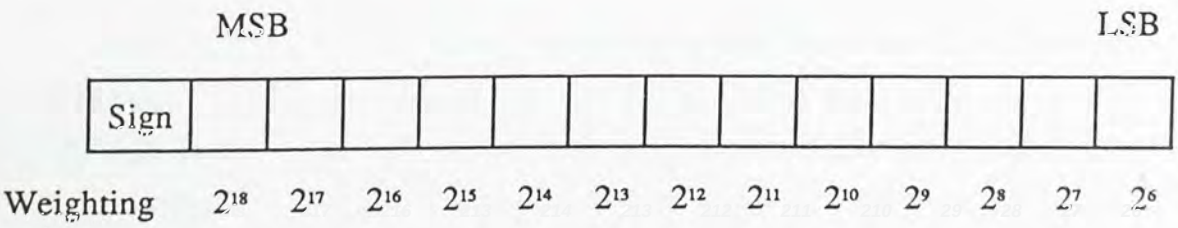


Figure 5.4 Format of Elements in Stage 3

Error 1 : MSE of the 2-D forward transform coefficients

$$\text{Error 1} = \frac{\sum_{i=0}^{M-1}\left[\sum_{p=0}^{N-1}\sum_{q=0}^{N-1}\left|C_{pq}-\hat{C}_{pq}\right|\right]_i}{256 \times 256}$$

C_{pq} which are the elements of matrix [C] represent the 2-D transform coefficients without truncation errors and $\hat{C}_{pq}$ which are the elements of matrix $[\hat{C}]$ represent the 2-D transform coefficients which contain truncation errors introduced in the two 1-D forward transforms. M is the numbers of the sub-block in a whole picture. For a picture with size 256 x 256 pixels and the size of sub-block N is 8, M is equal to 32 x 32.

Simulation Results

Picture	Error 1
WOMAN	5.39134756535776E-003
STONE	6.20556041738619E-003
HOUSE	6.22151545193286E-003

5.3.3.2 2-D Inverse ICT

The 2-D Inverse ICT given in Equation (5.3.b) can be implemented by the ICT chips and the block diagram is shown in Figure 5.5.

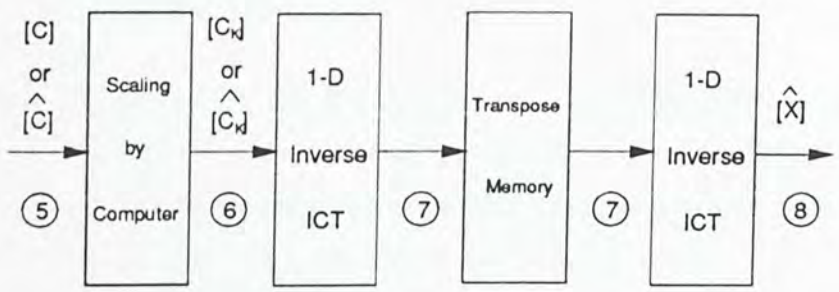


Figure 5.5 2-D Inverse ICT

Before entering the ICT chip, the transform coefficient matrix [C] in stage 5 of Figure 5.5 which are in floating point format should be scaled by computer software to form $[C_K]$ in state 6. Elements of $[C_K]$ should then be represented in a 14-bit format instead of real number representation. The finite-length format of the elements of matrix $[C_K]$ is determined by the magnitude of the largest

coefficients which are the DC coefficients. If the image data are in the range of [-128, 127], then the maximum magnitude of the DC coefficients of matrix $[C_K]$ will be equal to 2. Elements of $[C_K]$ can be represented by the following 14-bit format :

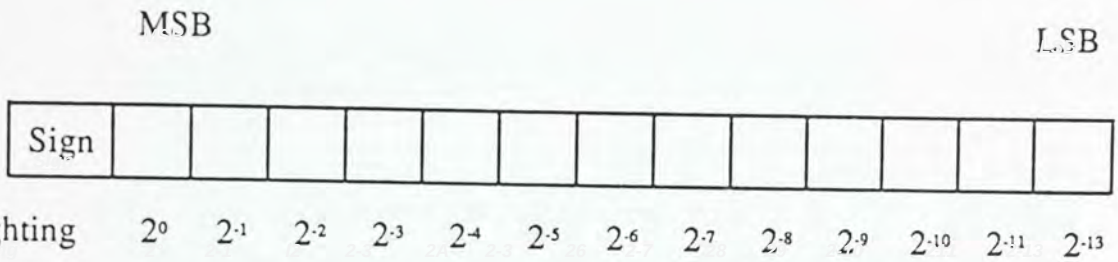


Figure 5.6 Format of Elements in Stage 6

When the input data pass through a 1-D Inverse ICT, the weight of the MSB of the output data (in stage 7 of Figure 5.5) will be increased by 6. Finally, the data format of the recovered picture elements in stage 8 will have the weighting of 2^{12} at MSB and 2^0 at LSB. The truncation error introduced in this 2-D inverse process in the ICT chip will be evaluated by the quantity of Error 2 and Error 3.

Error 2 : MSE of the pixels formed by 2-D inverse transformation of [C] whose elements contain no truncation errors

$$[\hat{X}] = [J]^t [C_K] [J]$$

$$\text{Error 2} = \frac{\sum_{i=0}^{M-1} \left[\sum_{p=0}^{N-1} \sum_{q=0}^{N-1} | X_{pq} - \hat{X}_{pq} | \right]_i}{256 \times 256}$$

Elements of matrix $[C_K]$ are the normalized transform coefficients which are obtained by multiplying elements of [C] using real number multiplication. In this case, the matrix $[C_K]$ contains only the truncation error introduced by the process of finite length representation. X_{pq} represent the original picture elements and $\hat{X}_{pq}$ represent the recovered picture elements.

Simulation Results

Picture	Error 2
WOMAN	5.66970825195313E-001
STONE	5.52658081054688E-001
HOUSE	6.82154931640625E-001

Error 3 : MSE of the pixels formed by 2-D inverse transformation of $[\hat{C}]$ whose elements contain truncation errors due to the 2-D forward transform

$$[\hat{X}] = [J]^t [\hat{C}_K] [J]$$

$$\text{Error 3} = \frac{\sum_{i=0}^{M-1} \left[\sum_{p=0}^{N-1} \sum_{q=0}^{N-1} | X_{pq} - \hat{X}_{pq} | \right]_i}{256 \times 256}$$

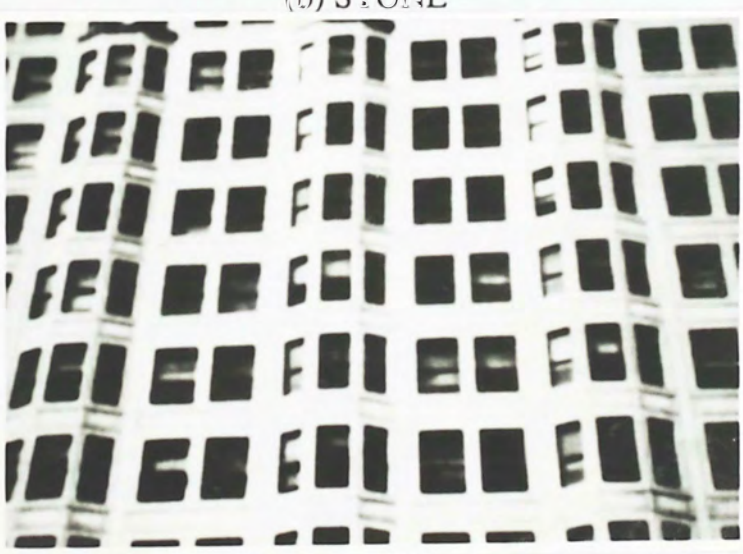
Elements of matrix $[\hat{C}_K]$ are the normalized transform coefficients which are obtained by multiplying elements of $[\hat{C}_K]$ using real number multiplication The elements of matrix $[\hat{C}_K]$ contain truncation errors introduced by the process of finite length representation introduced in stage 6 and also the error generated by the two forward 1-D transforms of the ICT chips. X_{pq} represent the original picture elements and $\hat{X}_{pq}$ represent the recovered picture elements.

Simulation Results

Picture	Error 3
WOMAN	5.79666137695313E-001
STONE	5.66879272460938E-001
HOUSE	6.85839843750000E-001

(a) WOMAN

(b) STONE



(c) HOUSE

Figure 5.7 Test Pictures

5.4 Architecture

The block diagram of the ICT chip is shown in Figure 5.8. It has five major parts which are the input stage, the control block, the multiplier, the accumulator and the output stage.

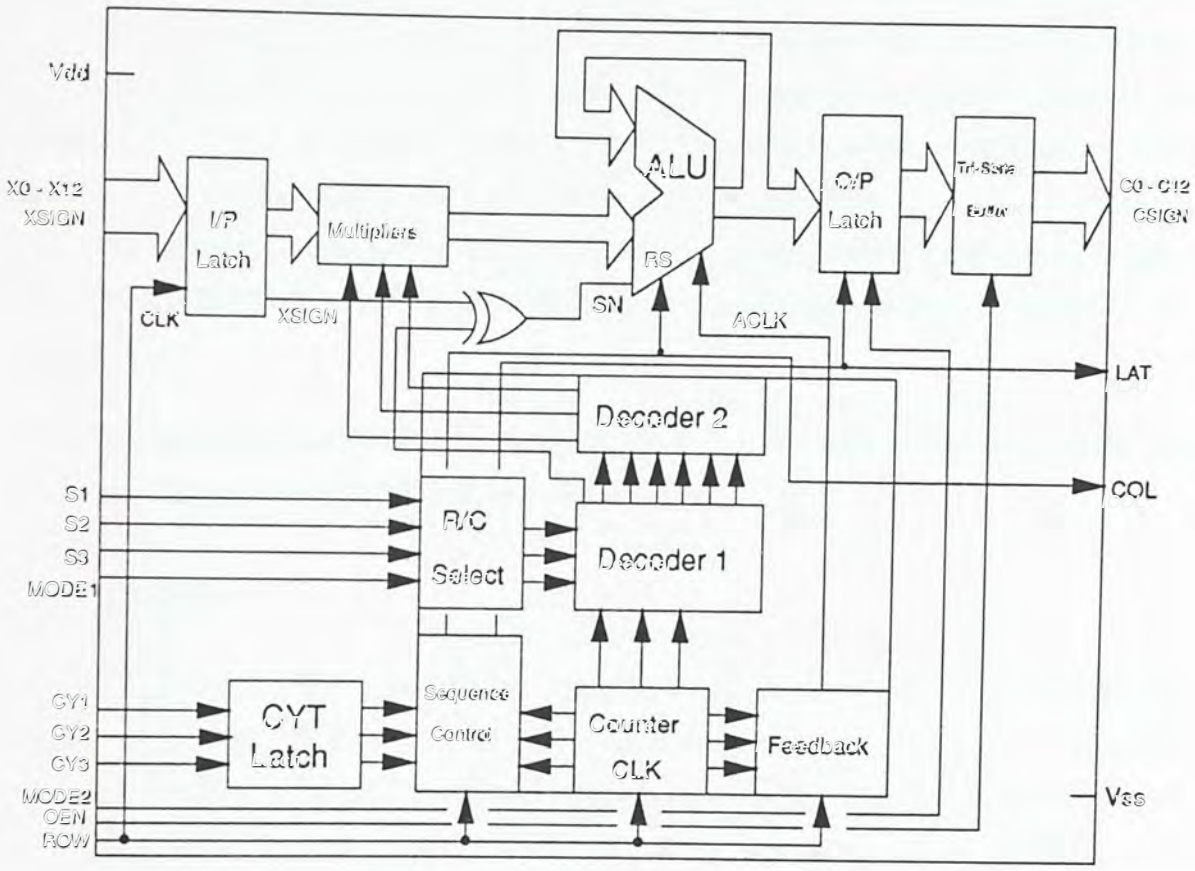


Figure 5.8 The Block Diagram of the ICT Chip

5.4.1 Input Stage

This stage acts as a buffer for the input data so that the input data are stable during a cycle of operation.

5.4.2 Control Block

This block generates the internal timing signals and control signals for the operations of the ICT chip. This block also controls the ALU to perform forward transform when MODE1=0 and inverse transform when MODE1=1. The R/C (Row/Column) select block decodes the three signals S1, S2 and S3 so that the correct Row/Column is selected in the multiplication.

The counter is increased every time a data is latched in. It is reset after one cycle. The counting sequence can be stopped prematurely by the sequence control and this is useful in inverse transformation to speed up the processing time.

The two decoders together select the desired matrix element as the multiplier for the calculation. Conceptually, the decoder interprets the three select inputs S1, S2 and S3 to select a specific row or column depending on the mode of operation. At the same time, it decodes the counter outputs to select the right element in the corresponding row or column. Since there are six different values in the transform matrix, the decoder represents each integer by asserting one of its seven outputs. The extra integer output is the sign bit. The integer outputs are further compressed into three bits by the decoder 2 to minimize the number of interconnects in the multiplier.

The accumulator feedback control determine at which period of time the feedback path of the accumulator is to be updated.

5.4.3 Multiplier

In the multiplier, a decoding algorithm rather than the normal shift-add algorithm is used. This is achieved by examining how each product is generated with each corresponding multiplier. As shown in table 5.3, one can see that each bit in the product is the result of one of the six possible operations indicated on each row. For example, p_2 can be generated from x_1, $x_1+x_2+c_1$, x_0+x_1, 0, x_2 or x_1 depending on the multiplier which is in turn determined by the outputs from the decoder 2. The sign bit of the multiplier is implemented by a simple EXOR gate.

Multiplicand	Operations						Product
X	x 2	x 3	x 6	x 8	x 9	x 10	P
x_0	0	x_0	0	0	x_0	0	p_0
x_1	x_0	x_0+x_1	x_0	0	x_1	x_0	p_1
x_2	x_1	$x_1+x_2+c_1$	x_0+x_1	0	x_2	x_1	p_2
x_3	x_2	$x_2+x_3+c_2$	$x_1+x_2+c_2$	x_0	x_0+x_3	x_0+x_2	p_3
x_4	x_3	$x_3+x_4+c_3$	$x_2+x_3+c_3$	x_1	$x_1+x_4+c_3$	$x_1+x_3+c_3$	p_4
x_5	x_4	$x_4+x_5+c_4$	$x_3+x_4+c_4$	x_2	$x_2+x_5+c_4$	$x_2+x_4+c_4$	p_5
x_6	x_5	$x_5+x_6+c_5$	$x_4+x_5+c_5$	x_3	$x_3+x_6+c_5$	$x_3+x_5+c_5$	p_6
x_7	x_6	$x_6+x_7+c_6$	$x_5+x_6+c_6$	x_4	$x_4+x_7+c_6$	$x_4+x_6+c_6$	p_7
x_8	x_7	$x_7+x_8+c_7$	$x_6+x_7+c_7$	x_5	$x_5+x_8+c_7$	$x_5+x_7+c_7$	p_8
x_9	x_8	$x_8+x_9+c_8$	$x_7+x_8+c_8$	x_6	$x_6+x_9+c_8$	$x_6+x_8+c_8$	p_9
x_{10}	x_9	$x_9+x_{10}+c_9$	$x_8+x_9+c_9$	x_7	$x_7+x_{10}+c_9$	$x_7+x_9+c_9$	p_{10}
x_{11}	x_{10}	$x_{10}+x_{11}+c_{10}$	$x_9+x_{10}+c_{10}$	x_8	$x_8+x_{11}+c_{10}$	$x_8+x_{10}+c_{10}$	p_{11}
x_{12}	x_{11}	$x_{11}+x_{12}+c_{11}$	$x_{10}+x_{11}+c_{11}$	x_9	$x_9+x_{12}+c_{11}$	$x_9+x_{11}+c_{11}$	p_{12}
x_{13}	x_{12}	$x_{12}+x_{13}+c_{12}$	$x_{11}+x_{12}+c_{12}$	x_{10}	$x_{10}+x_{13}+c_{12}$	$x_{10}+x_{12}+c_{12}$	p_{13}
	x_{13}	$x_{13}+c_{13}$	$x_{12}+x_{13}+c_{13}$	x_{11}	$x_{11}+x_{13}+c_{13}$	$x_{11}+x_{13}+c_{13}$	p_{14}
	x_{13}	$x_{13}+c_{14}$	$x_{13}+c_{14}$	x_{12}	$x_{12}+x_{13}+c_{14}$	$x_{12}+x_{13}+c_{14}$	p_{15}
	x_{13}	$x_{13}+c_{15}$	$x_{13}+c_{15}$	x_{13}	$x_{13}+c_{15}$	$x_{13}+c_{15}$	p_{16}
	x_{13}	$x_{13}+c_{16}$	$x_{13}+c_{16}$	x_{13}	$x_{13}+c_{16}$	$x_{13}+c_{16}$	p_{17}

Table 5.3 Multiplier Design Table (+ means OR)

5.4.4 Accumulator

The accumulator is a 20-bit carry ripple adder and performs 2's complement addition. Its outputs are feedback as one of the operand through a set of latches. In operation, these latches update the feedback operand every time a multiplication is completed. Thus no intermediate product has to be stored in this implementation.

5.4.5 Output Stage

The output stage consists of a set of latches and a set of tri-state buffers. In addition, it will convert the 2's complement results from the accumulator to the sign-magnitude format if required before the data is latched out at the end of a transformation. An external signal is used to control the output data of the ICT chip.

5.4.6 Synchronization

As the numbers of cycle of the operation is controlled by the CY_1, CY_2 and CY_3 signals, the synchronization of the ICT chip is achieved by the ROW signal. As shown in Figure 5.9, a MRSET signal is used to reset the chip before operations. At each rising edge of the ROW signal, the input data is latched in to the ICT chip and internal counter is increased by 1. When the counter value is equal to eight (or the binary value of [CY_3 CY_2 CY_1] plus one for inverse transformation), the computation cycle is completed and a LAT signal is activated after the falling edge of the ROW signal in order to latch out the resulted data from the accumulator to the output latch. One more ROW signal is required for the ICT chip to generate an internal COL signal. The COL signal is used to clear the accumulator value after one cycle of calculation and make it ready for the next cycle.

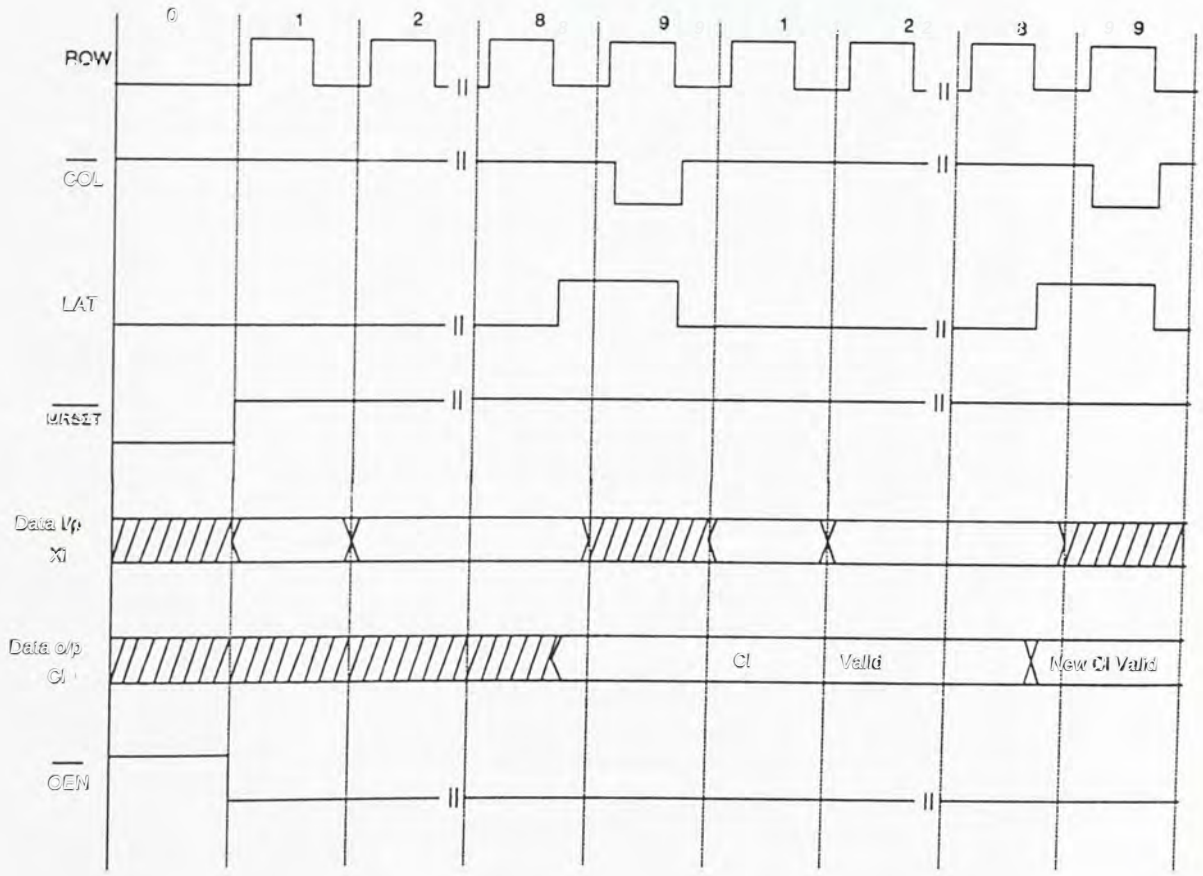


Figure 5.9 The Timing Diagram for the ICT Chip

5.5 Applications

5.5.1 One-dimensional Transform

As each ICT chip can perform one of the row (or column) multiplication of the ICT transform matrix, the 1-D transform can be achieved by connecting the eight ICT chips together as shown in Figure 5.10.

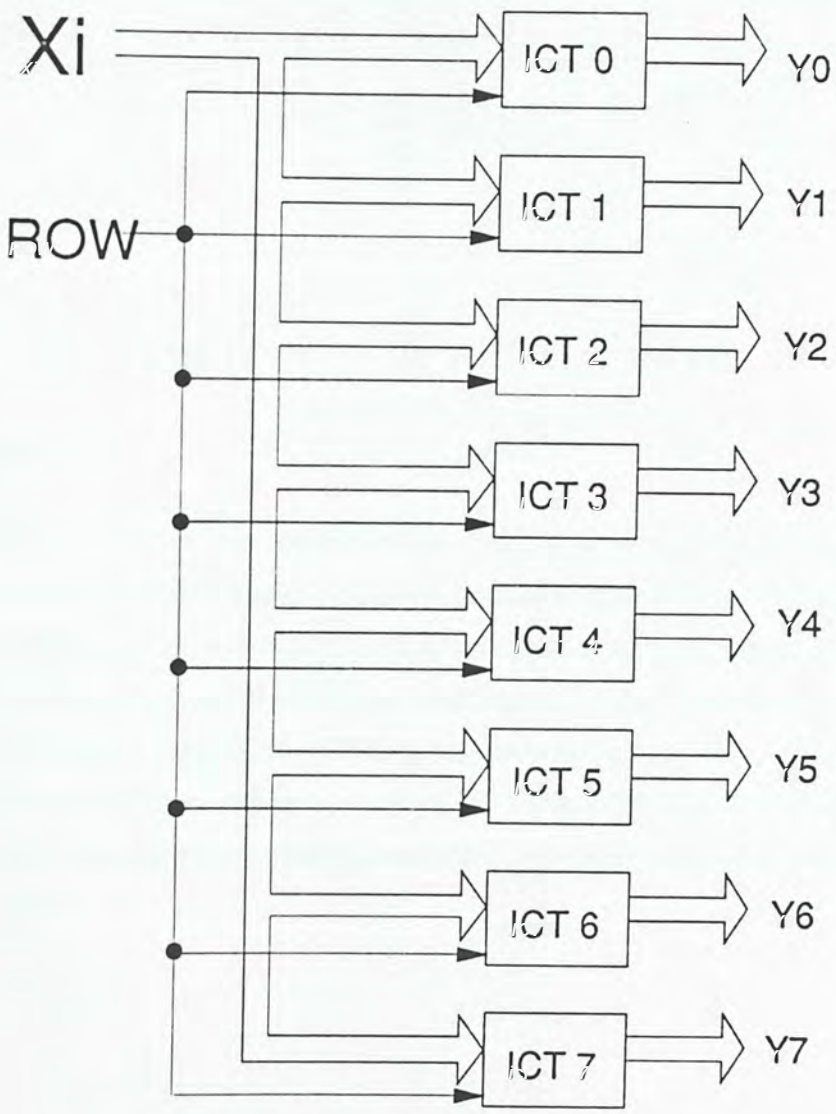


Figure 5.10 Circuit for 1-D ICT Transform

5.5.2 Two-dimensional Transform

The modular approach adopted allows the ICT chips to be connected in a pipeline to perform the 2-D transform. The proposed circuit is shown in Figure 5.11. The data sequence controller is used to schedule the results obtained from the

first matrix multiplication in the 2-D transform and to make them available to the subsequent matrix multiplication in the right format and sequence (transpose operation).

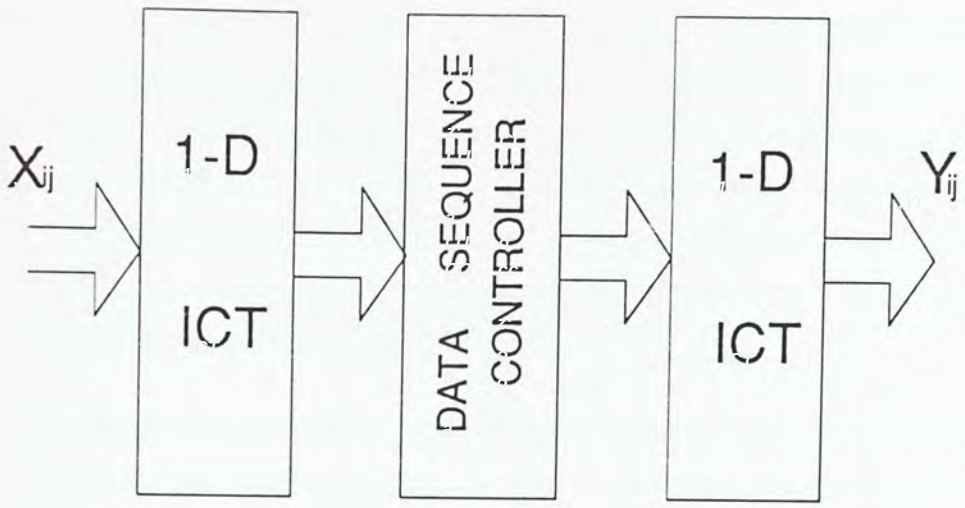


Figure 5.11 Circuit for 2-D ICT Transform

5.6 Conclusion

A modified structure for implementation of the Integer Cosine Transform was described. A new ICT chip has been developed in a lost cost gate array with 2800 gates. The transform matrix and the input and output considerations of the new ICT chips are improved. Compared with the existing ICT chip, this new ICT chip has advantages of higher transform efficiency, smaller truncation error and faster speed. With the aid of data sequence controller, several new ICT chips can perform the two-dimensional transformation in about 0.5 seconds for a picture with a size of 512 x 512 pixels.

Chapter 6 CONCLUSION

6.1 Summary of the Discoveries

The method of generating new orthogonal transforms by destroying dyadic symmetry, which was first proposed by Cham, was extended in chapter 2. Several definitions and theorems related to the generation of new transforms were defined and derived. Thirteen new orthogonal transforms have been generated by this method. It was found that some of them have transform efficiency very close to the DCT. It is believed that these transforms are most useful in a practical image coding system for its ease of implementation.

However, if performance is of paramount importance, one should use the best transform in an image coding system. A new orthogonal transform called Odd Weighted Cosine Transform (OWCT) was proposed in chapter 3. Tests using various criteria have shown that the OWCT has better performance than the "industrial standard" DCT. A fast computational algorithm for the OWCT has also been derived. It was found that computation of the OWCT requires only a few more additions and multiplications than the DCT.

In chapter 4 of this thesis, the symmetry properties of all the four types DWT matrices were analysed by using the principle of dyadic symmetry. A new recursive algorithm based on those symmetry properties was developed for all four types radix-2 DWT matrices. The new algorithm has been shown to have less computational requirements than other existing algorithms and it is also numerically stable in finite length arithmetic and suitable for calculation DWT of large orders.

Finally, a modified structure for implementation of the Integer Cosine Transform (ICT) was proposed in chapter 5. A new ICT processor has been developed by using a low cost 3 μm gate array with 2800 gates. Compared with the existing ICT chip, this new ICT chip has the advantages of higher transform efficiency, smaller truncation error and faster speed of operation. It is believed the transform coding system implemented by those ICT chips can be used in slow-scan TV systems and other image processing applications.

6.2 Suggestions for Further Research

For the new order-8 orthogonal transforms generated from chapter 2, there are arbitrary constants a, b, c, d, e and f in their transform matrices. It is believed that there should have some set of constants existed for each transform such that the transform has maximum transform efficiency. If we find an efficient method to optimize those arbitrary constants, a transform with better performance than the DCT may be found.

The OWCT only modifies the odd part of the transform matrix of the DCT. It is believed that if we find a weighted version of whole transform matrix of the DCT, a better transform may be found. One suggestion is to combine the OWCT with the Phase Shift Cosine Transform (PSCT) proposed by Wang to form such a new transform.

By using the designed ICT chip, we can construct a low cost image processing system for operations such as image compression, image transmission, low-pass filtering as well as image zooming and subsampling.

REFERENCES

[1] R.C. Gonzalez and P. Wintz, 'Digital Image Processing', Addison-Wesley Publishing Company, 1987.

[2] A.K. Jain, 'Image Data Compression : A Review', *Proc. IEEE*, vol.69, no.3, pp.349-389, March 1981.

[3] P.A. Wintz, 'Transform Picture Coding', *Proc. IEEE*, vol.60, pp.809-820, July 1972.

[4] R.J. Clarke, 'Transform Coding of Images', Academic Press, 1985.

[5] A. Rosenfeld and A.C. Kak, 'Digital Picture Processing', vol.1, Academic Press, 1982.

[6] W.D. Ray and R.M. Drive, 'Further Decomposition of the Karhunen-Loeve Series Representation of a Stationary Random Process', *IEEE Tran. Information Theory*, vol.IT-16, pp.845-850, Nov. 1970.

[7] H.C. Andrew and W.K. Pratt, 'Fourier Transform Coding of Images', *Proc. International Conf. on System Sciences*, Hawaii, pp.677-679, Jan. 1968.

[8] W.K. Pratt and J. Kane, 'Hadamard Transform Image Coding', *Proc. IEEE*, vol.57, no.1, pp.58-68, Jan. 1969.

[9] H. Andrews, 'Computer Techniques in Image Processing', Academic Press, 1970.

[10] H. Enomot and K. Shibata, 'Orthogonal Transform Coding System for Television Signals', *IEEE Trans. Electromagn. Comp.*, vol.EMC-13, pp.11-17, Aug. 1971.

[11] W.K. Pratt, W.H. Chen and L.R. Welch, 'Slant Transform for Image Coding,' *Proc. Symp. Appl. of Walsh Functions*, March 1972.

[12] W.K. Pratt, W.H. Chen and L.R. Welch, 'Slant Transform Image Coding', *IEEE Trans. Commun.*, vol.COM-22, no.8, pp.1075-1093, Aug. 1974.

[13] N. Ahmed, T. Natarajan and K.R. Rao, 'Discrete Cosine Transform', *IEEE Trans. Comput.*, vol. C-23, pp.90-93, Jan. 1974.

[14] W.H. Chen, C.H. Smith and S.C. Fralick, 'A Fast Computational Algorithm for the Discrete Cosine Transform', *IEEE Trans. Commun.*, vol.COM-25, pp.1004-1009, Sep. 1977.

[15] Z. Wang, 'Reconsideration of "A Fast Computational Algorithm for the Discrete Cosine Transform"', *IEEE Trans. Commun.*, vol.COM-31, pp.121-123, Jan. 1983.

[16] M. Hamidi and J. Peral, 'Comparison of the Cosine and Fourier Transforms of Markov-1 Signals,' *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp.428-429, Oct. 1976.

[17] R.J. Clarke, 'Relation Between the Karhunen-Loeve and Cosine Transforms', *Proc. IEE*, vol.128, pt.F, no.6, pp.359-360, Nov. 1981.

[18] A.K. Jain, 'A Fast Karhunen-loeve Transform for a Class of Random Process', *IEEE Trans. Commun.*, vol.COM-24, pp.1023-1029, Sep. 1976.

[19] A.Z. Meiri, E. Yudilerich, 'A Pinned Sine Transform Image Coder', *IEEE Trans. Commun.*, vol.COM-29, no.12, pp.1729-1735, Dec. 1981.

[20] A.K. Jain, 'A Sinusoidal Family of Unitary Transforms', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, pp.356-365, Oct. 1979.

[21] H. Kitajima, 'A Symmetric Cosine Transform', *IEEE Trans. Comput.*, vol. C-29, pp.317-323, April 1980.

[22] Z. Wang, ' Fast Algorithm for the Discrete W Transform and for the Discrete Fourier Transform', *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp.803-816, August 1984.

[23] H.S. Kwak, R. Srinivasan and K.R. Rao, 'C-Matrix Transform', *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp.1304-1307, Oct. 1983.

[24] W.K. Cham and R.J. Clarke, 'Application of the Principle of Dyadic Symmetry to the Generation of Orthogonal Transforms', *Proc. IEE*, vol.133, pt.F, no.3, pp.264-270, June 1986.

[25] W.K. Cham and Y.T. Chan, 'Integer Discrete Cosine Transform', *Proceedings of 1987 IASTED International Symposium on signal Processing and its Applications*, Australia, pp.674-676, 24-28 Aug. 1987.

[26] R.N. Bracewell, 'The Hartley Transform,' Oxford University Press, 1986.

[27] P.S. Yen, 'Data Compression Properties of the Hartley Transform', *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, no.3, pp.450-451, March 1989.

[28] Z. Baoyu, 'A Circular Convolution Algorithm for Discrete W Transform', *Proceedings of IEEE Singapore ICCS'88*, pp.1027-1030, Oct. 1988.

[29] L.D. Davisson, 'Rate-Distortion Theory and Application', *Proc. IEEE*, vol.60, pp.800-808, July 1972.

[30] W.H. Chen and C.H. Smith, 'Adaptive Coding of Monochrome and Color Images,' *IEEE Trans. Commun.*, vol. COM-25, pp.1285-1292, Nov. 1977.

[31] J. Max, 'Quantizing for Minimum Distortion', *IRE Trans. Information Theory*, vol.6, pp.7-12, March 1960.

[32] W.K. Cham, 'Transform Coding of Pictorial Data', *PhD Thesis*, Dept. Electronic & Electrical Eng., Loughborough University of Technology, 1983.

[33] Z. Wang, 'The Phase Shift Cosine Transform', *ACTA Electronica Sinica*, vol.14, no.6, pp.11-19, Nov. 1986.

[34] R.J. Clarke, 'Application of Sine Transform in Image Processing', *Electronics Letter*, vol.19, no.13, pp.490-491, June 1983.

[35] N. Ahmed and K.R. Rao, 'Orthogonal Transform in Digital Signal Processing', New York, springer-Verlag, 1975.

[36] A.K.Jain, 'Advances in Mathematical Models for Image Processing', *Proc. IEEE*, vol.69, pp.502-528, May 1981.

[37] H.S. Hou, 'A Fast Recursive Algorithm for Computing the Discrete Cosine Transform', *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp.1455-1461, Oct. 1987.

[38] R.N. Bracewell, 'The Fast Hartley Transform', *Proc. IEEE*, vol. 72, pp.1010-1018, Aug. 1984.

[39] H.V. Sorensen, D.L. Jones, C.S. Burrus and M.T. Heideman, 'On Computing the Discrete Hartley Transform', *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-33, pp.1231-1238, Oct. 1985.

[40] N. Suehiro and M. Hatori, 'Fast Algorithms for the DFT and Other Sinusoidal Transforms', *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-34, pp.642-644, June 1986.

[41] C.P. Kwong and K.P. Shiu, 'Structured Fast Hartley Transform Algorithms', *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-34, pp.1000-1002, Aug. 1986.

[42] Z. Wang, 'The Fast W Transform - Algorithm and Programme', *Scientia Sinica*, series A, vol.5, pp.549-560, May 1988.

[43] M.T. Sun, L. Wu and M.L. Liou, 'A Concurrent Architecture for VLSI Implementation of Discrete Cosine Transform', *IEEE Trans. Circuits and Systems*, vol. CAS-34, No.8, pp.992-994, Aug. 1987.

[44] C.S. Choy, W.K. Cham and L. Leung, 'A LSI Implementation of Integer Cosine Transform,' *Proceedings of IEEE Singapore ICCS'88*, pp.551-555, Oct. 1988.