A PC/AT-BASED ICT IMAGE ARCHIVING SYSTEM

A Thesis

Submitted to

The Department of Electronic Engineering

of

The Chinese University of Hong Kong

In

Partial Fulfilment of the Requirements

for the Degree of

Master of Philosophy

By

Ringo Wai-kit Lam

June, 1991

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Dr. C.S. Choy and Dr. W.K. Cham, for their patient guidance and valuable comment on my research, and for their patient reading of the manuscript of this thesis. Special thanks also to Mrs. Yum for preparing the photographs.

This thesis would not be finished without the help of all my colleagues in the Application Specific Integrated Circuit Laboratory of Chinese University with whom I have enjoyed every moment of my past three years research.

Finally, I would like to express my deepest gratitude to my father and sister for their moral support throughout the whole course of Master of Philosophy.

ABSTRACT

This thesis suggests the specifications, architecture and data storage format for digital image archiving system. The compression chip set, namely ICT (Integer Cosine Transform) chip set, of the image archiving system will first be discussed. It was known that some of the ICT transform matrices have higher decorrelating ability than the DCT but the hardware implementation of these ICT matrices is easier because of involving integer number only. These chips are now fabricated by 3 μm CMOS single layer metal gate array technology. They may be used to handle 1D or 2D transform.

With the ICT chip set, an image archiving system (IAS) employing the quantization method proposed by Chen and Smith in 1976 has been built. This system is based on the most common AT computer and it may zoom images or use less transform coefficients for reconstruction of images. We suggest a special hardware and user-friendly software for this system. As data storage formats may affect image searching time and display time, two possible formats are discussed and one of them is found more suitable for the IAS.

Experimental results show that the reconstructed images of the system are acceptable. The computational time of the system is fast enough for applications such as photo album display. Recently, the CCITT/ISO Joint Photographic Experts Group proposed an international standard for image compression and decompression. Our system may be modified to compatible with this standard.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Introduction

The demand for transmission and storage of image data will increase rapidly in this last decade of the twentieth century. Many countries have already planned or built their high-speed public transmission network which can provide a wide range of services including images to the subsrcibers. However, the bandwidth of the proposed 2B + D ISDN considered by many countries is only limited to 144 Kbits/s. For a picture with a spatial resolution of 512 x 512 pels and at 8-bit/pel intensity resolution, 2 Mbits are required to represent it. If it is transmitted over the above ISDN, it needs 13.9 seconds. For the huge memory and transmission bandwidth required by a digitized picture, the application of compression technique on the images is necessary.

There are two basic approaches to image compression, predictive coding and transform coding. The first approach is carried out in the spatial domain such as differential pulse code modulation (DPCM) and other predictive methods [1-3]. The principle behind these methods is that the image is usually highly correlated, on average the neighbour picture elements have similar intensity. Therefore, the values of one or more earlier elements may be used to predict the present element. Since its implementation is simple, many commercial products have adopted the DPCM, e.g. the still camera. However, the DPCM is not so efficient as the second approach, transform coding which can yield a better compression ratio but requires much more calculation [4]. It divides the original image into subpictures of a particular block size such that the highly correlated spatial data are transformed into weakly correlated coefficients. It results in significant and insignificant coefficients of which the insignificant ones can be discarded in the process. Therefore, the overall amount of memory is reduced. The main problem of transform coding is on the implementation but recently it has been eased from the advances in high-speed digital hardware. Some VLSI chips have been designed to perform the transformation. Some of them can operate at rates up to 20 MHz. Using these chips, an adaptive transform system will give good image quality at rates between 0.5 and 1.0 bit/pel. The use of transform coding is seen to be prevalent in the future.

At present one of the most important problems of using transform coding is the hardware implementation. In this thesis, we will concentrate on the implementation of a two-pass transform coding system. The first pass computes the transform and transform energy. The second pass determines the quantization levels and code words. The following pages will review the transform coding theory. The organisation of this thesis will be given in the final section of this chapter.

1.2 Transform Coding Theory

Both predictive and transform coding techniques are based on the inter-element correlation of the image. The higher the correlation of the image data is, the more the power spectral distribution is close to the low frequency components, thus requiring less channel capacity for transmission. The extent to which images may be compressed whilst still keeping satisfactory reproduction of the image is crucially dependent upon their correlation properties. Fortunately, most of the images have high values of correlation coefficient. Compared between predictive and transform coding, the first one is not so effective as the second one in decorrelating the image data, that means the compression ability of transform coding is higher. For active images, i.e. those containing a reasonable amount of spatial detail, they may be successfully coded by transform methods at a compression ratio as high as 8 : 1.

The following subsections will examine the basic requirements of a transform coding system. These requirements are the transform, the quantization and the bit allocation.

1.2.1 Image Transform Coder and Decoder

Based on the discoveries by Chen in 1976 [5] and 1984 [6], the image transform coder and decoder can be built in two ways : single-pass and two-pass system. In the two-pass scheme, data compression by transform coding consists of three processes which is done in the transmitter.

(a) It transforms highly correlated image elements into a set of weakly correlated coefficients.

(b) Bits are allocated to these coefficients so that more bits are allocated to those coefficients with higher variances.
(c) Coefficients are quantized before transmission of the coefficients.

Where (a) belongs to the first pass, and (b) and (c) belong to the second pass.

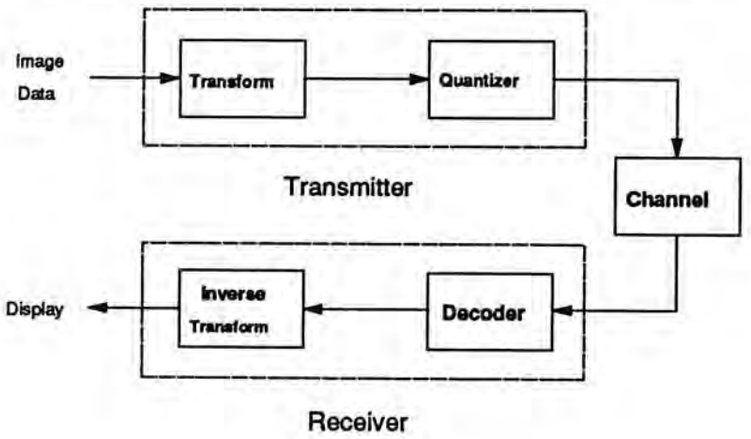


Fig 1.1 Block Diagram of a Transform Coding System

To illustrate the process, the block diagram of a transform coding system is shown in Fig.1.1. In this system, the original image is divided into subpictures of size n x n, where n is usually 8 or 16. The subpictures are transformed into an array of independent coefficients of which maximum information is packed into a minimum number of coefficients. More bits are assigned to the coefficients having larger variances, and fewer bits for coefficients having smaller variances. The final process is to quantize and code the coefficients for transmission. At the receiver, the received data with overhead information are decoded to the quantized transform coefficients, and an inverse transform is applied to the coefficients to recover the picture. The problem of this scheme is the memory required for the second pass and the complexity of the quantization algorithm. This thesis will propose an architecture for implementating this scheme.

There is another well known data compression scheme proposed by Chen and Pratt in 1984 [6]. This scheme is single-pass and is based on run-length coding algorithm. This method is well suited for intraframe coding of moving images.

1.2.2 Transformation

In transform coding technique, the most important part is to find the best transform function. Optimum transforms can convert the statistically dependent picture elements into an array of uncorrelated coefficients. The total energy of the spatial image data is preserved in the transform domain. The criterion of optimum transform is based on whether it can completely decorrelate the image data. The sole optimum transform which satisfies the criterion is the Karhunen-Loeve transform (KLT) [7]. However, its use in practice presents many problems. Firstly, the covariance function of an image is not stationary, and therefore one must either choose a different covariance matrix matched to the subpicture or use an average. Secondly, the covariance matrix sometimes turns out to be singular. Finally, the hardware implementation of the transformation is difficult. In other words, since the basis vectors of the KLT are the eigenvectors of the covariance matrices, each picture has its own KLT's basis vectors, the fast computational algorithm is difficult to find.

The practical consideration of the transform has prompted research into other suboptimum transforms. These transforms can be usually executed by a fast algorithm and approximate well the KLT. The increase in computational speed by fast algorithm allows processing of data blocks of larger size at higher sampling rates and may result in an overall performance better than that obtained with statistically optimum but computationally inefficient schemes. The first suboptimum transform to be investigated for image coding was the two-dimensional Fourier transform [8]. Then, the Walsh transform was discovered and it has a considerable decrease in computational requirement than the Fourier transform. In 1971, some interest was on the application of Haar transform [9]. Because the Haar transform takes differences of the samples or differences of local averages of the samples of the input vector, it is a very fast transform, but results in larger coding error. At about the same time, Enomoto and Shibata [10] introduced the Slant transform which matches basis vectors. Their work was subsequent extended by Pratt et Al. [11] who generalized this 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. One of the most important transforms is the Discrete Cosine Transform (DCT) which was introduced by Ahmed et al. in 1974. The DCT is a real transform close to the KLT of a first-order stationary Markov sequence and is superior in decorrelation of the coefficients. In 1976, Jain suggested that a sine transform (even sine-1 transform) is a KLT of a Markov Process under the condition that the boundary of the process is known [13]. 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 (even sine-1) [13]. Similar decomposition were proposed by Meiri and Yudelevich [14] who suggested using another sine transform (even sine-2) for their residue process in their pinned sine transform image coder. In 1981, Clarke showed that the DCT becomes the KLT of the source data in the limit as the adjacent element correlation coefficient tends to unity for the first-order Markov source [15].

Compared between the DCT and the Walsh-Hadamard Transform, the former has a high transform efficiency while the latter is easier to implement. Cham and Clarke made a compromise between these two transforms, they derived a new pair of transform called the High Correlation Transform (HCT) and Low Correlation Transform (LCT) from Walsh Transform using the theory of Dyadic Symmetry in 1983 [16]. The HCT has a better performance at higher values of inter-element correlation and the LCT is better for low correlation image data. Computations using these two transforms need additions, subtractions and binary shifts. Cham further investigated his work and generated the Integer Cosine Transform (ICT) from the DCT [17-18]. The main advantages of the ICT is ease of implementation and has efficiency as high as the DCT for some of the order-8 ICTs. In this thesis, a review of the ICT and the implementation of the ICT using ASIC technology are given in chapter 2.

1.2.3 Bit Allocation

The next procedure in transform coding is to assign bits to the coefficients. More bits are assigned to coefficients with higher variances according to either method:

(a) Method 1:

The rate distortion theory [19] from which the optimum bit allocation may be achieved states that :

the output of a source can be transmitted with average distortion D when the minimum transmission rate of the communication channel is R(D). For a source with Gaussian probability distribution and mean square error of D, R(D) is found to be

$$R(D) = \frac{1}{2}\log\left(\frac{\sigma^2}{D}\right) \qquad for \quad \sigma > \sqrt{D}$$

$$= 0 \qquad otherwise \qquad eqn(1.1)$$

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

Although equation (1.1) is only applicable to Gaussian data, it can provide an upper bound for the performance of any non-Gaussian source.

(b) Method 2:

This approach is to equalize the quantization distortion from each coefficient. In general, a total number of bits in a block, B, is divided among the coefficients in the following steps:

(1) The quantization distortion of the AC coefficients are determined. The number of bits assigned to each coefficient is set to zero so that the variance of the coefficient is equal to the quantization distortion.
(2) One bit is assigned to the coefficient with the largest quantization distortion such that it is reduced to one-fourth of the previous value. The number of bits assigned to this coefficient b_{ij} is increased by 1.
(3) If the number of bits assigned to the coefficients Σb_{ij} is larger than the predetermined value B , stop the process; otherwise, repeat (2).

The bit allocation scheme can be made adaptive [5]. One of the popular adaptive transform coding of both monochrome and colour image data is that of Chen and Smith who proposed the scheme in 1976. The picture is divided into sub-blocks of four classes on the basis of sub-block AC energy and the variances of the coefficients for each class are computed. The bit assignment of the coefficients is done in the same way as method 1 or 2 while generating four bit allocation maps. Since the coefficients are classified into different groups, it is more efficient in reducing the bit rate of transmission.

1.2.4 Quantization

The quantizer can be uniform or non-uniform. In uniform quantizers, the decision levels are equidistant, and the reconstruction levels are in the middle of the decision intervals. Uniform quantizer is not the most effective conversion, however, it is possible to quantize a random variable with a smaller error variance, thus reducing the average quantization error. The non-uniform quantization can be used in this sense. The best optimum quantizer is therefore a non-uniform quantizer.

In the process stated in section 1.2.3, bits are assigned to coefficients which must be quantized before being transmitted. The choice of an optimum quantizer must be based on a statistical basis. It was found that the optimum quantizer for DC coefficient is of Rayleigh density distribution whereas the quantizer for AC coefficient may be modelled by a Gaussian density distribution or Laplacian density distribution. However, some may argue that the choice of quantizer should be achieved by minimising the mean-square error of the quantizer. Max showed that when the distortion of a quantizer having a fixed number of output levels for a signal of known probability distribution is minimized, it is in fact an optimum quantizer [20]. The procedure of obtaining the Max quantizer is as follows:

The decision and reconstruction levels, d_i and r_i, are determined by

$$d_i = \frac{r_i + r_{i-1}}{2} \qquad eqn(1.2)$$

and

$$r_i = \frac{\int_{d_i}^{d_{i+1}} xp(x)dx}{\int_{d_i}^{d_{i+1}} p(x)dx} \qquad eqn(1.3)$$

where p(x) is the probability density function of the input signal.

These values of r_i and d_i can be calculated by recursive use of equation (1.2) and (1.3) when an initial value r_0 is set. This initial value should be verified after the process has stopped. For a fixed number of quantization levels L, if r_{L-1} is the centroid of d_{L-1} and $d_{L=}\infty$, then the initial choice of r_0 is correct; otherwise a new value of r_0 should be used for repeating the process.

It is worth noting that the DC coefficient may sometimes be quantized by uniform pdf quantizer because of its wide spread of distribution. The argument to this point is that the coefficient is merely a scaled blockwise average of the image luminance levels, there is no a priori reason why its distribution should exhibit any well-defined statistical characteristics. Therefore a uniform pdf is well to model the dc coefficient.

1.2.5 Entropy Coding

After level quantization of the coefficient data, there are some quantization levels more likely to occur than other levels. Entropy coding makes use of a variable length coding procedure which assigns shorter codeword to the highly probable levels, longer codeword to less probable levels. It was found that the bit rate reduction resulted by pdf-optimized quantization is not the best achievable by entropy coding. The best result may be obtained by using a uniform threshold quantization and then entropy coding the quantization levels. Recently, a coding scheme named as the JPEG (Joint Photographic Experts Group) has been proposed. This scheme, which may become the standard in the future, employs Huffman codes for coefficients.

1.2.6 Error of Transform Coding

This section discusses about the error introduced by the transform coding. The source of error is mainly due to the quantization and bit allocation. The reconstruction error variance σ_r at the receiver is found to be equal to the quantization error variance σ_q at the transmitter [21]. Their relationship is given by

$$\sigma_r^2 = \frac{1}{NxN}\sum_{i,j=0}^{N-1}\sigma_{r,ij}^2 = \frac{1}{NxN}\sum_{i,j=0}^{N-1}\sigma_{q,ij}^2 = \sigma_q^2 \qquad eqn(1.4)$$

Where $\sigma_{r,ij}^2 = E((X_{ij} - \hat{X}_{ij})^2), i,j = 0,1,\ldots,N-1$ is the reconstruction error variance of the (i,j) th input element and $\sigma_{q,ij}^2 = E((Q_{ij} - \hat{Q}_{ij})^2), i,j = 0,1,\ldots,N-1$is the quantization error variance of the (i,j) the transform coefficient.

Where ε_q^2 is the quantizer performance factor which depends on the quantizer characteristic and the pdf of the input signal. A variable correction factor ε^2 in the equation reflects the performance of a practical quantizer.

B	Quantizer Performance Factor $\varepsilon^2 s^{-2B}$				Variable Correction Factor ε^2			
(bits)	U	G	L	Γ	U	G	L	Γ
1	0.250035	0.363078	0.500350	0.666807	1.0001	1.4523	2.0001	2.6672
2	0.062517	0.117490	0.176198	0.231739	1.0003	1.8798	2.8192	3.7078
3	0.015631	0.034514	0.054450	0.070469	1.0004	2.2089	3.4848	4.5100
4	0.003908	0.009506	0.015382	0.019634	1.0006	2.4335	3.9377	5.0262
5	0.000977	0.002506	0.014102	0.005188	1.0007	2.5663	4.2005	5.3125
6	0.000244	0.000647	0.001062	0.001340	1.0008	2.6507	4.3487	5.4878
7	0.000061	0.000166	0.000270	0.000341	1.0010	2.7128	4.4200	5.5901

Table 1.1 Quantizer performance factor of four common pdfs at different bit rates

Here the symbols U, G, L and Γ have the meanings of uniform, Gaussian, Laplacian and Gamma density distribution respectively.

From the result of equation (1.4), it is not difficult to find that the optimum bit allocation cannot be exactly found by the rate distortion theory. The reconstruction error depends not only on the number of bits assigned to a block but also on the quantizer characteristic, the modified optimum bit allocation should be given by

$$B_{ij} = B + \frac{1}{2}\log_2 \frac{\varepsilon_{ij}^2 \sigma_{ij}^2}{\left[\prod_{k,l=0}^{N-1} \varepsilon_{kl}^2 \sigma_{kl}^2\right]^{1/(NxN)}} \quad , \quad i,j = 0,1,\ldots,N-1 \qquad eqn(1.5)$$

Average reconstruction error variance can be found from equation (1.5). The average reconstruction error variance is now given by

$$\min(\sigma_r^2) = \varepsilon^2 2^{-2B}\left(\prod_{k,l=0}^{N-1} \sigma_{kl}^2\right)^{1/(NxN)} \qquad eqn(1.6)$$

1.3 Organisation of the Thesis

Following this introductory chapter, chapter 2 will discuss the background theory of the Integer Cosine Transform (ICT) and a chip set based on this transform. A 2D Integer Cosine Transform system is also suggested in chapter 2. Using the chip set, a PC/AT-based image archiving system can be built. We propose an architecture of the system that efficiently performs the operations of forward and inverse transform. Coefficient quantization is based on the adaptive method proposed by Chen and Smith in 1976. The system accesses the secondary memory, e.g. Hard disk, to get part of the coefficients for display. When several images are shown on the screen, the user may zoom their most favourable image for details. The architecture of this system are given in details in chapter 3. In chapter 4, the performance of the image archiving system is evaluated using the criteria of time requirement, pictures taken from the monitor are also given. This chapter will also give an estimate of the system performance if advanced fabrication technique is used. Finally, the thesis concludes with chapter 5 which collates the work that has been performed in the course of the research and makes suggestions for future work.

CHAPTER 2 2D INTEGER COSINE TRANSFORM CHIP SET

2.1 Introduction

Transform coding, which may reduce the memory requirement of an image to 1/16 with reasonable quality, is one of the best techniques for image coding. The most widely used transform for image transform coding is the DCT and various DCT chips have been implemented, which use about 14 bits to represent a DCT kernel component [27-31]. Since the basis vector's components of the DCT are mainly real numbers, it was found that the implementation of the DCT in finite length arithmetic was more complicated than those transforms whose basis vector's components are integers only, i.e. the Walsh trasnform, Slant transform and HCT. Unfortunately, none of them have the same data compression ability as the DCT. Cham generated another transform called Integer Cosine Transform (ICT) as an alternative to the DCT. The ICT which has shown to be functionally compatible to the DCT and with performance close to the DCT [18]. The ICT (10,9,6,2,3,1) requiring only 4 bits for exact representation of its components is implemented on a LSI chip and it was found that the implementation was easier than that of the DCT.

This chapter discribes a LSI implementation of the ICT (10,9,6,2,3,1) and its associated intermediate storage, Data Sequencer, for 2D transform. Implementation of LSI chips has to consider the constraints such as die size, pin number and wiring complexity. Realization of ICT (10,9,6,2,3,1) on a chip can only be done by optimizing those limitations. To reduce the complexity of the ICT chip, modular architecture is employed to allow data pipelining. Moreover, integers of the transform kernel are generated by a decoding process which can reduce the internal wiring. The ICT chip, which can perform 1-D transform by itself, can perform 2D transform when it is used with another LSI gate array chip Data Sequencer. The Data Sequencer has been carefully designed to speed up inverse transform operations of low-pass filtering and subsampling. Low-pass filtering is a process which requires less transform coefficients to reconstruct the image and subsampling is a process which uses the few lowest frequency coefficients to make a smaller image. For subsampling operation, the reduction of time may be up to 1/12 of time required by

a normal 2D transform. The subsampling operation is useful for image archiving system. It allows a set of size-reduced images to be displayed on the screen and hence the user can easily locate the desired picture.

The background theory of the ICT (10,9,6,2,3,1) will be described in section 2. Section 3 will briefly explain the implementation of chips. The design requirement is considered in section 4. Section 5 will highlight the architecture of the chips. A 2D system will be given in section 6.

2.2 The Integer Cosine Transform (ICT)

The Integer Cosine Transform was derived from the DCT by the concept of Dyadic Symmetry. The (i,j) th kernel component of the order-8 DCT is

$$t_{ij} = \begin{cases} 1/\sqrt{8} & i = 0 \quad j \in [0,7] \\ (2/\sqrt{8})\cos[(j+1/2)i\pi/8] & i \in [1,7] \quad j \in [0,7] \end{cases} \qquad eqn(2.1)$$

By representing kernel components of the same magnitude using the same variable, the DCT kernel can be expressed as

$$[T] = \begin{bmatrix} k_0 & (1 & 1 & 1 & 1 & 1 & 1 & 1 & 1) \\ k_1 & (a & b & c & d & -d & -c & -b & -a) \\ k_2 & (e & f & -f & -e & -e & -f & f & e) \\ k_3 & (b & -d & -a & -c & c & a & d & -b) \\ k_4 & (1 & -1 & -1 & 1 & 1 & -1 & -1 & 1) \\ k_5 & (c & -a & d & b & -b & -d & a & -c) \\ k_6 & (f & -e & e & -f & -f & e & -e & f) \\ k_7 & (d & -c & b & -a & a & -b & c & -d) \end{bmatrix} = \begin{bmatrix} k_0 & J_0^t \\ k_1 & J_1^t \\ k_2 & J_2^t \\ k_3 & J_3^t \\ k_4 & J_4^t \\ k_5 & J_5^t \\ k_6 & J_6^t \\ k_7 & J_7^t \end{bmatrix} = [K][J] \qquad \text{eqn (2.2)}$$

Where k_i is a scaling constant such that $|k_i J_i| = 1$, J_i is the column vector of matrix [J] and [K] is a matrix of diag { $k_0, k_1, \ldots, k_7$ }. It can be shown that [T] is orthogonal if

$$ab = ac + bd + cd \qquad \text{eqn (2.3)}$$

[T] resembles the DCT if

$$a > b > c > d \text{ and } e > f \qquad \text{eqn (2.4)}$$

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 2.1 Transform efficiency for various transforms with the adjacent correlation coefficient ρ equal to 0.9

The [T] whose a, b, c, d, e and f are all integers and satisfy eqn (2.3) and eqn (2.4) is called an Integer Cosine Transform (ICT). There are many possible ICTs. One ICT with a=10, b=9, c=6, d=2, e=3 and f=1 has been shown to be a promising alternative to the DCT. This ICT, denoted ICT (10,9,6,2,3,1), is functionally compatible and has performance close to the DCT [18].

2.3 LSI Implementation [25] [32]

The chip set consists of two kinds of ICs, the ICT chip and the data sequencer chip. They can be wired into different configurations. For instance, a simple ICT chip is sufficient to perform the 1D transform, however multiple of ICT chips can be wired in parallel to do the same job with many folds of improvement in speed. A 1D transform system built with 8 ICT chips connected in parallel is shown below. It can complete the transform in 9 T_p μsec; T_p being the time required for the ICT chip to process one pixel.

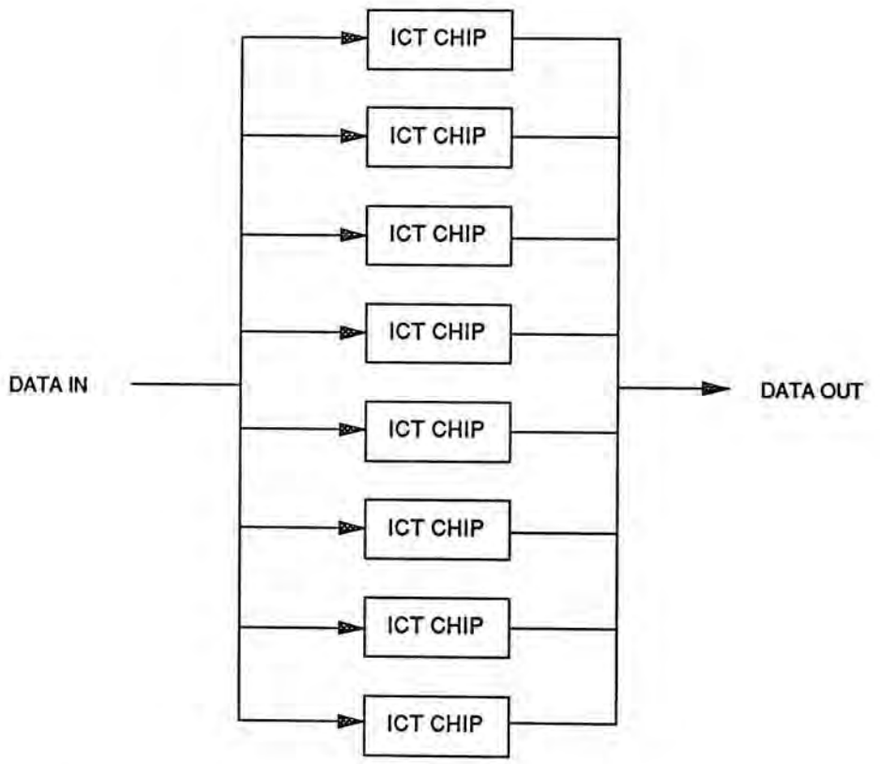


Fig.2.1 A 1D transform system built with 8 ICT chips

2.3.1 ICT chip

TYPE	PIN NO.	IN/OUT	FUNCTION
VDD	1,33	I	Power supply
GND	17,49	I	Grounding
ROW	61	I	Input clock
COL	54	O	Output clock for Data Sequencer or output strobe
LAT	55	O	Output strobe for updating S3, S2 and S1 of Data Sequencer
S3 - S1	2,3,4	I	Choose the transform vector in the ICT chip
CY3 - CY1	57,59,60	I	Determine the number of data input in one group
MODE 1	63	I	High : inverse transform Low : forward transform
MODE 2	24	I	High : 2's complement numbers Low : signed numbers
MODE 3	26	I	High : subtract 128 from the input Low : bypass the subtraction
MODE 4	27	I	High : add 128 to the output Low : bypass the addition
XSIGN	23	I	The input sign bit
X12 - X0	22,21,19, 18,16,15, 14,12,11, 10,8,7,5	I	Data input pins, X12 - MSB, X0 - LSB.
CSIGN	30	O	The output sign bit
C12 - C0	32,35,36, 37,39,41, 43,44,45, 47,50,51, 53	O	Data output pins C12 - MSB, C0 - LSB.
OEN	29	I	High : tri-state output Low : TTL compatible output
RESET	62	I	Reset the ICT chip

Table 2.2 Pin assignments of ICT chip

This chip is concerned about the transform stated in section 2.2. Owing to the limited facilities, it is fabricated by a low cost 3 μm CMOS single layer metal gate array technology and using the DIL package. The chip area is 62 mm^2. The input and output data length are equal to 14. The pin assignment of the chip is listed above.

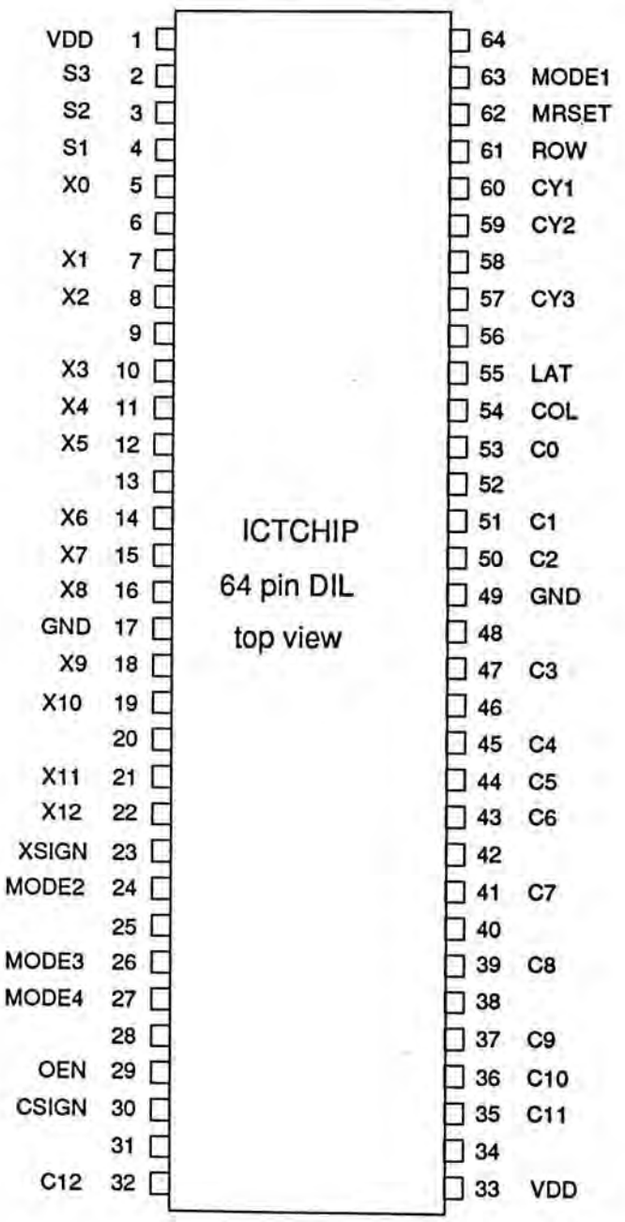


Fig.2.2 64-pin DIL package pinout of ICT chip

2.3.2 Data Sequencer

This chip is designed for storing the data output from the first stage of ICT chips in 2D transform. The LSI implementation of this chip is done by the same technology used for the ICT chip.

Symbol	Pin No.	Type	Name and Function
ROW	5	I	Clock out the data
COL	6	I	Latch the data into the Data Sequencer
I13 - I0	30,28,27, 25,23,22, 20,19,18, 14,13,12, 10,8	I	14-bit I/P data I13 - MSB, I0 - LSB.
CT1 - CT2	31,34	I	Determine the number of data fetched before sending out
RESET	35	I	High : reset S3, S2, S1 and BL
LAT	37	I	Generate S3, S2, S1 and BL
MODE	38	I	High : low-pass filtering Low : subsampling
S3 - S1	40,42,43	O	Choose the appropriate transform vector in the ICT chip S3 - MSB
BL	44	O	Notify the data is outputting from the Data Sequencer
Q13 - Q0	45,47,50, 51,53,54, 56,58,59, 61,62,63, 2,3	O	14-bit O/P data Q13 - MSB, Q0 - LSB.
GND	17,49	I	grounding
VDD	1,33	I	power supply

Table 2.3 Pin assignments of Data Sequencer

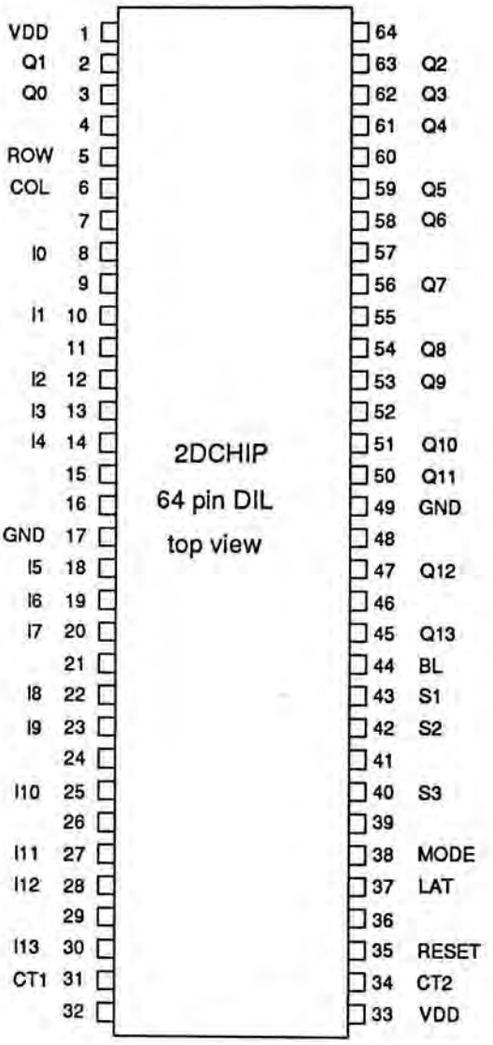


Fig.2.3 64-pin DIL package pinout of Data Sequencer

2.4 Design Considerations

To design the chip set, we have to consider both application and hardware constraint. In the following, we will discuss flexibility of using different number of the transform chips, the selection of transform matrix as given in Table 2.1, and the hardware constraints such as the gate size and the number of I/O pins provided by the package. Since the ICT chip can be used singly or together with the Data Sequencer in building a transform system, the interface between these two chips is carefully designed. For the Data Sequencer, design consideration has taken into account of the low-pass filtering and subsampling operations.

2.4.1 ICT chip

2.4.1.1 Specifications

In mathematics, when X is an input vector and C is the coefficient vector, the 1D forward transform is given by

$$\begin{aligned} C &= [T]\ X \\ &= [K]\ [J]\ X \\ &= [K]\ C_I \end{aligned} \qquad \text{eqn (2.5)}$$

and the inverse transform is expressed as

$$\begin{aligned} X &= [T]^t\ C \\ &= [J]^t\ [K]\ C \\ &= [J]^t\ C_K \end{aligned} \qquad \text{eqn (2.6)}$$

Because of limitation of the gate size, the ICT chip can only handle one transform vector of the matrix [T]. The 1D transform is done by 8 ICT chips in parallel or by feeding the same set of data eight times to one ICT chip. The function performed by one ICT chip is as follows:

Forward transform

$$Y_{ij} = \sum_{j=0}^{7} J_{ij} X_j \qquad eqn(2.7)$$

If vector Y is 1D transform of X, the full transform operation can be expressed

$$Y = \left[\sum_{j=0}^{7} t_{0j} X_j \quad \sum_{j=0}^{7} t_{1j} X_j \quad \cdots \quad \sum_{j=0}^{7} t_{7j} X_j \right]^t \qquad eqn(2.8)$$

Inverse transform

$$X_i = \sum_{j=0}^{7} J_{ji} Y_j \qquad eqn(2.9)$$

Where i = 0, 1, ..., 7.

In mathematics, when [X] is an input matrix and [C] is the coefficient matrix, the 2D forward transform is given by

$$\begin{aligned} [C] &= [T]\,[X]\,[T]^t \\ &= [K]\,[J]\,[X]\,[J]^t\,[K] \\ &= [K]\,[C_I]\,[K] \end{aligned} \qquad \text{eqn (2.10)}$$

and the inverse transform is expressed as

$$\begin{aligned} [X] &= [T]^t\,[X]\,[T] \\ &= [J]^t\,[K]\,[C]\,[K]\,[J] \\ &= [J]^t\,[C_K]\,[J] \end{aligned} \qquad \text{eqn (2.11)}$$

To perform 2D transform, 16 ICT chips are required. The normalization matrix [K] is calculated outside the chip. It may be included in the quantization process.

2.4.1.2 I/O Bit Length Consideration

For the sake of flexibility of using the chip to perform the 1D and 2D transforms, the ICT chip is desired to have the same bit length in the input and output. Therefore, the input of the second stage of ICT chips can accept all the data from the output of the first stage of ICT chips in a 2D transform system. Since the internal operation of the chip adds a number of data, the range of the output will be extended by 6 bits. To equalize the numbers of input and output pins, a truncation process is carried out before outputting the data as shown in Fig.2.3. In our design,

format of the input and output are 13 magnitude bits plus 1 sign bit. The 6 least significant bits of the output are truncated.

Considering the interface between the ICT chip and external circuit as well as the reduction of truncation error, the chip can be set to invert the sign bit of the input data in forward transform and that of the output data in inverse transform. The input data of the forward transform is in the range of $0 \leq X_i \leq 255$. If the data is applied to the chip, the sign bit of the chip should always be set to zero. On the other hand, the output of the chip in inverse transform must also be positive number, therefore the sign bit of the output must be equal to zero.

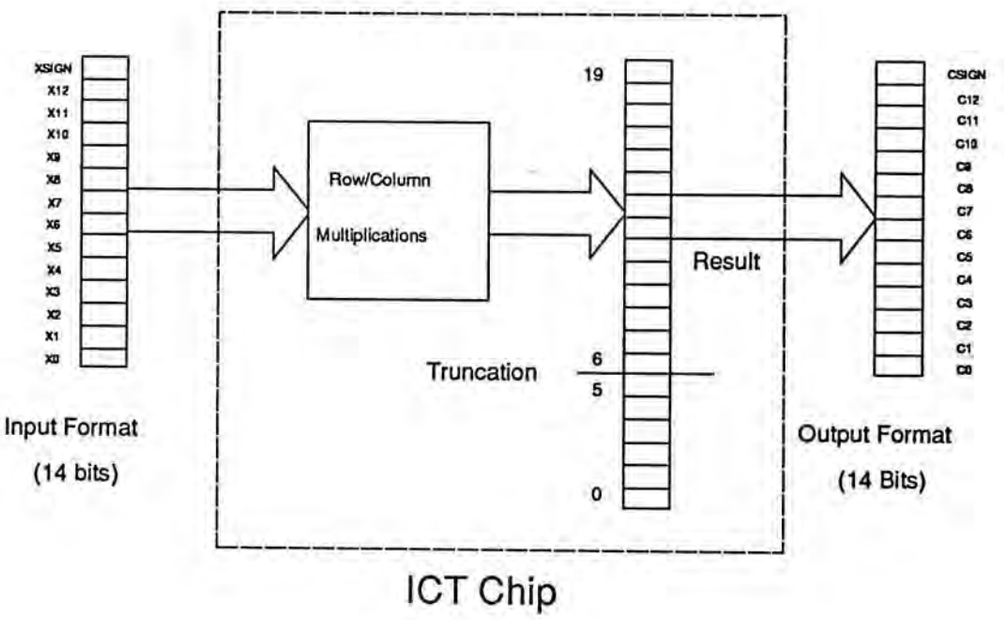


Fig.2.4 Truncation process of the ICT chip

When considering this characteristic of the transform, it is better to utilize the sign bit as a magnitude bit in the process. In one way, the truncation error of the operation may be reduced since the binary point is moved one position towards the MSB, the calculation accuracy can be increased by 1 bit. On the other hand, this setting may yield a simpler interface between the chip and the external circuit. To explain the interface problem, we may look into the following diagram which shows the input problem of the chip.

In the figure 2.5, the sign bit and 13 magnitude bits of the ICT chip are

hard-wired to the memory. For forward transform, the image data must be mapped to the 8 MSB of the magnitude bits. As the image data stored in the computer memory are usually represented by 8 bits, it is required to shift the data by one bit with its sign bit set to zero. For inverse transform, each transform coefficient is 16 bits, its 2 LSBs are not connected to the ICT chip. The output data in inverse transform faces similar problem as the input data in forward transform. This interface means more circuitry and thus time delay. However, by setting MODE 3 in forward transform and MODE 4 in inverse transform to high, we may directly apply the data to the chip without complicated circuitry. Their functions are equivalent to dc offset the image data by a magnitude of 128 in forward transform and restore its original dc level by adding the same number in inverse transform.

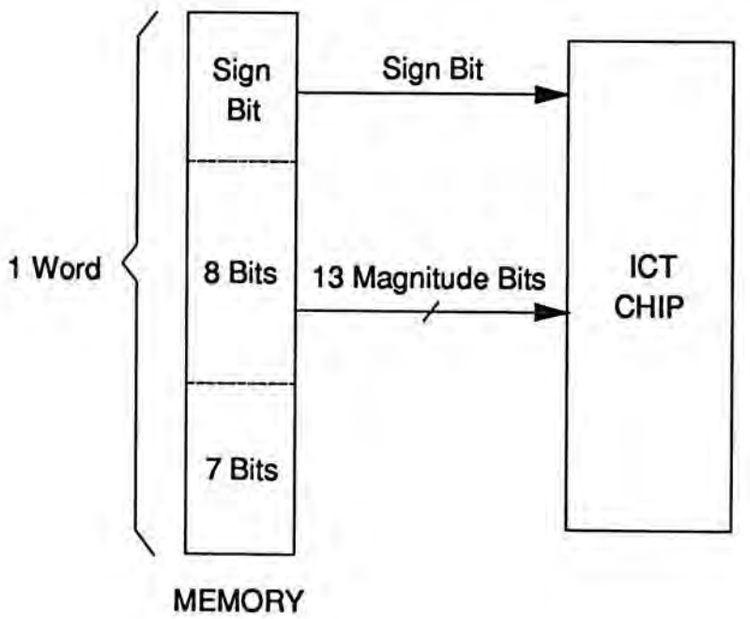


Fig.2.5 Interface problem between ICT chip and external circuit

2.4.1.3 Selection of the Transform Matrix [40]

The ICT (10,9,6,2,3,1) is chosen for designing the ICT chip because of its high transform efficiency and ease of implementation. The integer values of the ICT's basis vector components are so small that they can be implemented by at most four

binary shifts and additions. If an 8-bit image data column vector is input to the ICT chip, the internal operation of the ICT chip will at most generate a 14-bit integer output which is equal to the number of output bits. Therefore, there is no truncation error for 1D forward transform. Truncation error will in any case occur if the input is not an 8-bit integer, e.g. the transform coefficients input of the inverse transform. These coefficients are real number in nature but usually represented by several bytes in the computer. Because the ICT chip can only accept a maximum of 14 bits, the data needs to be truncated. This error will propagate to the output and affect the accuracy of the output data. To reduce the truncation error, some modification of the values of the ICT (10,9,6,2,3,1) is required. With investigation into the output format of the ICT chip, we may find that the matrix product of the 0th and the 4th row vector by the image data column vector are 11 bits in length. If the output data of the chip are normalized by square of the values given by the following normalization matrix [K],

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

$$= diag\left\{\frac{1}{|J_0|},\frac{1}{|J_1|},\frac{1}{|J_2|},\frac{1}{|J_3|},\frac{1}{|J_4|},\frac{1}{|J_5|},\frac{1}{|J_6|},\frac{1}{|J_7|}\right\} \qquad \text{eqn (2.12)}$$

the 0th and 4th data will at most be equal to 255 and 127.5 but the others will be much smaller. As a result, the binary point of the data is nevertheless determined by these two data and it must be located between the 8th and 9th bit counted from the MSB. As the truncation error is directly related to position of the binary point, it is desired to locate the binary near the MSB. Integer scaling factors R and M are included in the normalization matrix in order to advance the position of the binary point. The modified matrix [K] is

$$[\hat{K}] = diag\left\{\frac{1}{R\sqrt{8}},\frac{1}{\sqrt{442}},\frac{1}{M\sqrt{40}},\frac{1}{\sqrt{442}},\frac{1}{R\sqrt{8}},\frac{1}{\sqrt{442}},\frac{1}{M\sqrt{40}},\frac{1}{\sqrt{442}}\right\} \qquad eqn(2.13)$$

In accordance with the change in the normalization matrix, the ICT (10,9,6,2,3,1) is modified to the following format.

$$[\hat{J}] = \begin{bmatrix} R & R & R & R & R & R & R & R \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 3M & M & -M & -3M & -3M & -M & M & 3M \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ R & -R & -R & R & R & -R & -R & R \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ M & -3M & 3M & -M & -M & 3M & -3M & M \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix} \qquad \text{eqn (2.14)}$$

For clear understanding of the position of the binary point, a diagram is shown below to illustrate the finite length representation of a number.

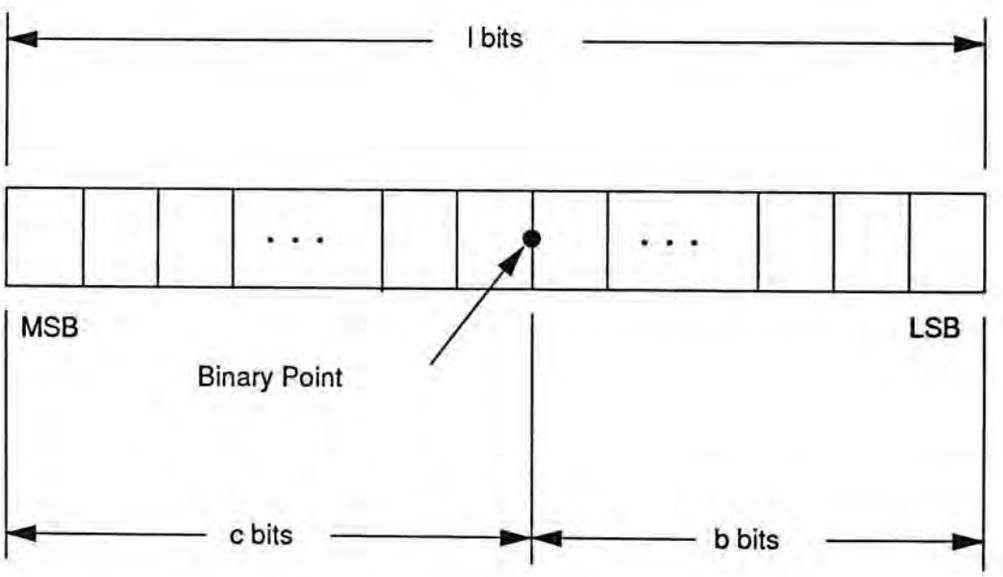


Fig.2.6 Illustration for position of the binary point

In this diagram, the truncation error e_T is in the range $\left(-\frac{1}{2}.2^{-b}, +\frac{1}{2}.2^{-b}\right)$. The maximum possible values of coefficients C_{Ki} are shown in the following table with the assumption that the image data are 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 2.4 Maximum Possible Values of C_{Ki} for Different Transform Matrices (MAX $[C_{Ki}]$)

The number of bits required to represent the integer value is

$$c = \text{Int}\,[\,\log_2\,(\,\text{MAX}\,[\,C_{Ki}\,]\,)] \qquad \text{for } i \in [0,7] \qquad \text{eqn (2.15)}$$

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

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

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 becomes

$$[\bar{J}] = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 9 & 3 & -3 & -9 & -9 & -3 & 3 & 9 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 3 & -9 & 9 & -3 & -3 & 9 & -9 & 3 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix} \qquad \text{eqn (2.16)}$$

2.4.2 Data Sequencer

The data sequencer acts as temporary storage for matrix multiplication's results and schedules these results to shift out for subsequent matrix multiplication. In addition, it is optimized to allow special operations, namely, low-pass filtering and subsampling, to execute in minimum time.

2.4.2.1 Normal Operation

Normal operation means that the number of input data is 8 x 8 and the output of the chip is also 8 x 8. In this operation, the chip latches 8 data from the output of the ICT chip by the rising edge of the COL signal. At the end of the COL signal it begins to supply 8 data to the ICT chip of the next stage which latches the data by the rising edge of the ROW signal. The Data Sequencer repeats the output of data set 8 times before switching to output another set of data which is just completely latched by the Data Sequencer.

2.4.2.2 Low-pass filtering operation

Filtering operation can be easily achieved in the transform domain by setting a number of transform coefficients to zero. Since it is obviously unnecessary to find product in multiplying zero cases. The calculation time can be reduced if these cases are skipped. The chip set has been optimized accordingly and the following table shows the calculation time for different degree of filtering.

Filtering size	Calculation time relative to normal operation
2 x 2	33.33 %
4 x 4	55.56 %
6 x 6	77.78 %

Table 2.5 Filtering size and its calculation time compared with normal operation

To achieve reduction in calculation time for the 2D transform, it is necessary to insert nil data cycles to the first stage matrix multiplication because the number of data generated for the second stage is more than that presented to the first stage.

2.4.2.3 Subsampling operation

In many applications such as image archiving, it is often necessary to display image in reduced size. This can be achieved by subsampling the image.

$$\begin{bmatrix} (\underline{X}_{11}) & X_{12} & \underline{X}_{13} & X_{14} & (\underline{X}_{15}) & X_{16} & \underline{X}_{17} & X_{18} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} & X_{27} & X_{28} \\ \underline{X}_{31} & X_{32} & \underline{X}_{33} & X_{34} & \underline{X}_{35} & X_{36} & \underline{X}_{37} & X_{38} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} & X_{47} & X_{48} \\ (\underline{X}_{51}) & X_{52} & \underline{X}_{53} & X_{54} & (\underline{X}_{55}) & X_{56} & \underline{X}_{57} & X_{58} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} & X_{67} & X_{68} \\ \underline{X}_{71} & X_{72} & \underline{X}_{73} & X_{74} & \underline{X}_{75} & X_{76} & \underline{X}_{77} & X_{78} \\ X_{81} & X_{82} & X_{83} & X_{84} & X_{85} & X_{86} & X_{87} & X_{88} \end{bmatrix}$$

Table 2.6 Subsampling data

Table 2.6 shows an 8 x 8 image block. The underscored elements are data required for 4 x 4 subsampling whereas those with parenthesis for 2 x 2 subsampling. The Data Sequencer is designed to reduce the time required for performing 2 x 2 and 4 x 4 subsampling operations. With these two degrees of subsampling, the dimension of the original image is reduced by 15/16 and 3/4 respectively. The time required to perform the subsampling operations in relative to normal operation is shown in Table 2.7.

Sampling size	Time required
2 x 2	8.33 %
4 x 4	27.78 %

Table 2.7 Subsampling size and its time required relative to normal operation

2.5 Architecture [32]

2.5.1 ICT chip

The ICT chip can be divided into several functional blocks, namely, the control, the multiplier, the ALU, the input-subtract and output-add circuits. Inside the control block, there are the R/C select, the sequence control, the counter, the decoder and the feedback control.

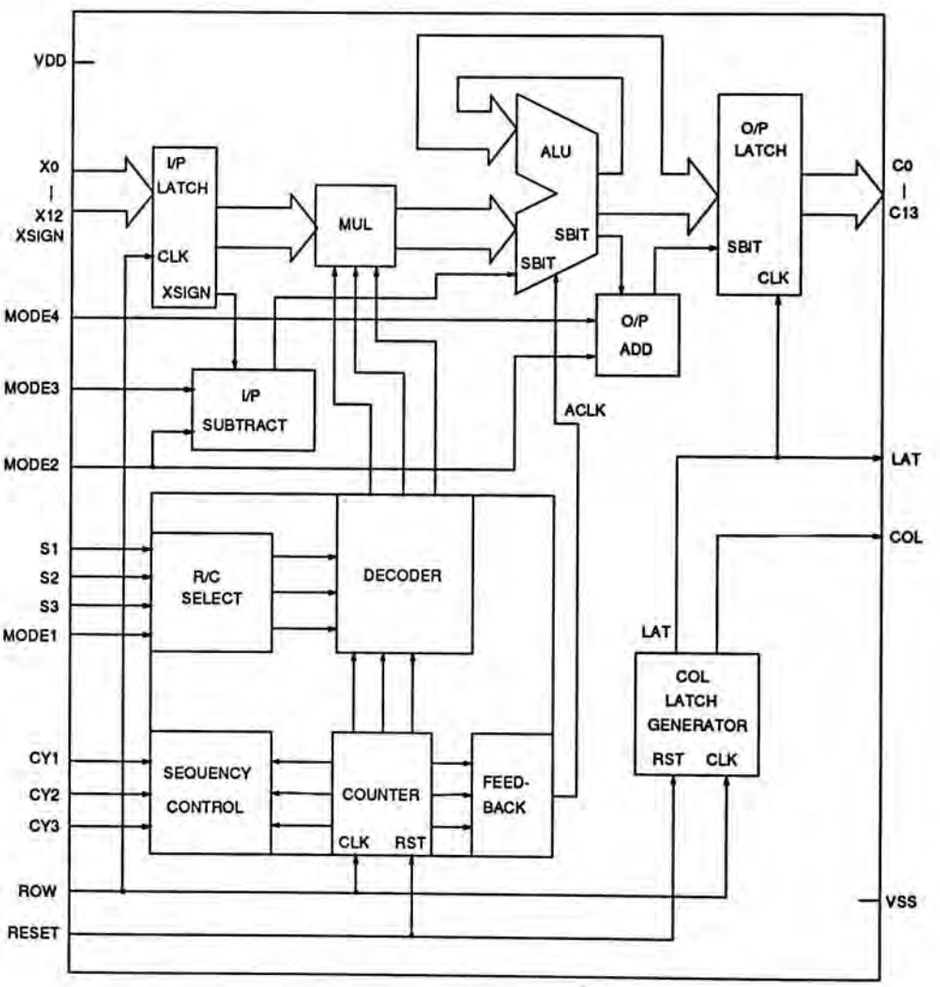


Fig. 2.7 Architecture of the ICT chip

2.5.1.1 Input Stage

The input data is buffered in this stage so they are stable during a cycle of operation.

2.5.1.2 Control Block

This block consists of five units which generate the internal timing signals and control signals for the operations of the ICT chip. A feedback signal generated determines at which period of time the feedback path of the ALU is to be updated. The ALU performs forward transform when MODE1 = 0 and inverse transform when MODE1 = 1. The counter is increased each time when the rising edge of the ROW comes. It is reset after the completion of latching a block of eight data. The countering sequence can be stopped prematurely by the sequency control to further speed up the processing time.

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 decoder divides the decoding into two operations. In the first operation, the decoder interprets the three select inputs to select a row or a column depending on the mode of operation. At the same time, it decodes the 3-bit counter outputs to select the right element in the corresponding row or column. Inside the decoder, it generates six decoded signals plus one for sign bit. Those six signals may select one of the integers in the ICT (10,9,6,2,9,3) for multiplication. To minimize the number of interconnects in the multiplier, the integer outputs are further compressed into three bits by the second operation of the decoder.

2.5.1.3 Multiplier

The ICT chip performs multiplication based on the shift-add algorithm. The multiplier circuitry is further optimized by exploiting the fact that the transform matrix has only six possible absolute values, i.e. the set (10,9,6,2,9,3). In other words, each product bit can be generated by manipulating the input data in six possible ways. These are shown in table 2.8. For example, the fourth bit of the product, P3, can be either X2, X2+X3+C2, X1+X2+C2, X0, X0+X3 or X0+X2 depending on the absolute value concerned in the transform matrix. Therefore, once an element in the transform matrix is selected for multiplication, the decoder in the control block in turn outputs a 3-bit value, which represents one of the values in the set (10,9,6,2,9,3), to the multiplier. The decoder also has separate output to take

care of signs in the transform matrix. In each transform cycle, the ALU accumulates the products from eight multiplications and the accumulated result, which is truncated to 14 bits, is latched out.

Multiplicand	x 2	x 3	x 6	x 8	x 9	x 10	Product
X0	0	X0	0		X0	0	P0
X1	X0	X0+X1	X0		X1	X0	P1
X2	X1	X1+X2+C1	X0+X1		X2	X1	P2
X3	X2	X2+X3+C2	X1+X2+C2	X0	X0+X3	X0+X2	P3
X4	X3	X3+X4+C3	X2+X3+C3	X1	X1+X4+C3	X1+X3+C3	P4
X5	X4	X4+X5+C4	X3+X4+C4	X2	X2+X5+C4	X2+X4+C4	P5
X6	X5	X5+X6+C5	X4+X5+C5	X3	X3+X6+C5	X3+X5+C5	P6
X7	X6	X6+X7+C6	X5+X6+C6	X4	X4+X7+C6	X4+X6+C6	P7
X8	X7	X7+X8+C7	X6+X7+C7	X5	X5+X8+C7	X5+X7+C7	P8
X9	X8	X8+X9+C8	X7+X8+C8	X6	X6+X9+C8	X6+X8+C8	P9
X10	X9	X9+X10+C9	X8+X9+C9	X7	X7+X10+C9	X7+X9+C9	P10
X11	X10	X10+X11+C10	X9+X10+C10	X8	X8+X11+C10	X8+X10+C10	P11
X12	X11	X11+X12+C11	X10+X11+C11	X9	X9+X12+C11	X9+X11+C11	P12
	X12	X12+C12	X11+X12+C12	X10	X10+C12	X10+X12+C12	P13
	0	C13	X12+C13	X11	X11+C13	X11+C13	P14
	0	0	C14	X12	X12+C14	X12+C14	P15
	0	0	0	0	C15	C15	P16

Table 2.8 Multiplication product and its inputs

The number of data points in one transform cycle is determined by the CY signals. Normally, the number of data points is eight. Other numbers will mean that filtering or subsampling is effected.

2.5.1.4 Accumulator

The accumulator is implemented by carry ripple adder design which can performs 20-bit 2's complement addition. One of the accumulator's operand is the product from the multiplier in section 2.5.1.3, and it changes as the clock comes. The other is the latched output feedback of the sum of previous products. The addition process of the transform is smoothly done in the accumulator without any intermediate product being stored.

2.5.1.5 Output Stage

The output stage is a set of tri-state latches. The 2's complement result of the accumulator may be converted to the sign-bit format in this stage. In inverse transform, the output stage may invert the sign bit of the 2's compliment output data by setting MODE 4 to high. If MODE 4 is set to high in inverse transform, MODE 3 should be set to high in forward transform correspondingly in order to return a correct recover of the image.

The output of the chip also has two clock signals LAT and COL. LAT is useful for signalling the external circuit that the output data is valid and the COL signal may be used to latch the output data. These signals can be connected to Data Sequencer directly.

2.5.2 Data Sequencer

The major functional blocks are two identical 8 x 14 first-in-first-out register blocks as shown in Fig.2.8. They operate alternatively as input storage and output source. This arrangement allows the first stage of ICT chips to continuously send their results for the next stage matrix multiplication. As a result, there is no speed degradation in 2D transform.

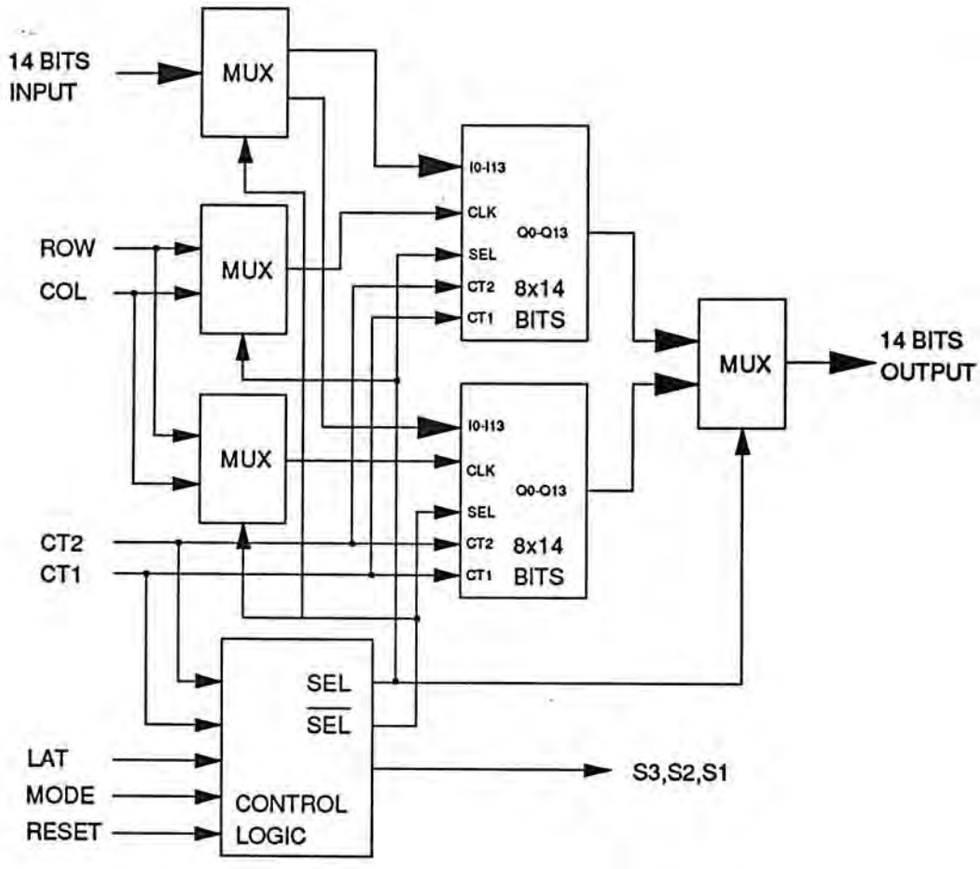


Fig.2.8 Architecture of the Data Sequencer

2.5.2.1 Input Stage

The input of the Data Sequencer is connected to a demultiplexer for directing the data to either storage inside the chip.

2.5.2.2 Control Logic

Part of the Data Sequencer's control is derived from the ICT chip, they are the LAT and COL signals. The LAT signal is used to update the values of S3, S2 and S1 which are in turn used to select the appropriate row or column vector of the second stage ICT chips. In subsampling operation, the generated sequence of S3, S2 and S1 are given in the following table.

Subsampling Size	S3 S2 S1
2 x 2	000,010,100,110,000,010,100,110, ...
4 x 4	000,100,000,100, ...

Table 2.9 The output patterns of S3, S2 and S1 for different subsampling sizes

The two control signals, CT2 and CT1 are used to set the number of data input to the Data Sequencer before being output its data, they can be connected to CY3 and CY2 of the ICT chip repectively. MODE is used to control the filtering operation and BL is an output signal for indicating that the output of the Data Sequencer is valid.

2.5.2.3 Internal Storage

The structure of an 1 x 8 bits storage is illustrated in Fig.2.9. It is built by an eight-bit shift register and an associated multiplexer circuit. The figure clearly shows that the subsampling and filtering operations are performed by selecting one of the output of the shift register.

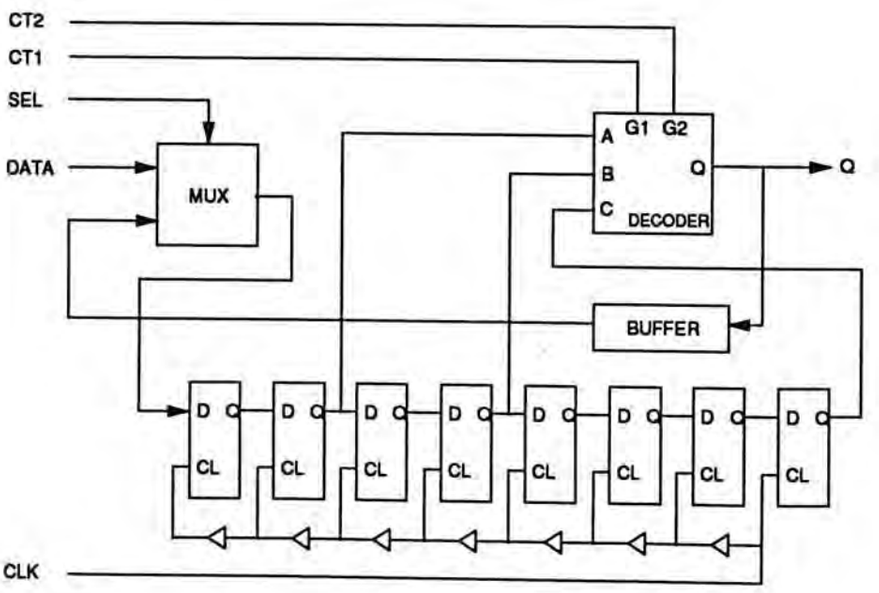


Fig.2.9 Structure of an 1 x 8 bits storage

The storage of the Data Sequencer is integrated by two blocks of 14 x 8 bits storage. After one block is filled up with the desired number of data, the other will begin to fill in data and the original block will output data at this time.

2.5.2.4 Output Stage

The output stage is multiplexed, the output data is either in high state or low state.

2.6 2D Integer Cosine Transform System [32]

2.6.1 Hardware Architecture

To perform 2-D transform, one simple configuration will have two ICT chips and one data sequencer connected in the way as depicted in Fig.2.10.

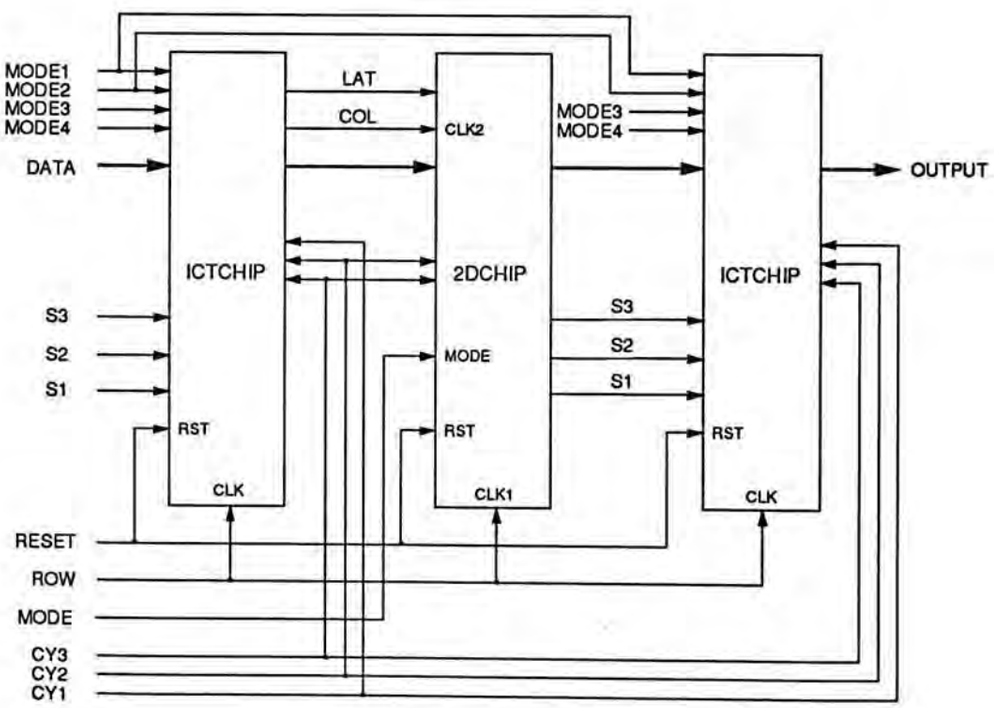


Fig. 2.10 Connection of two ICT chips and one Data Sequencer

This configuration can transform a 256 x 256 picture in 196.6 msec with a 3 MHz clock rate. If a faster response is desired, 8 of those can be connected in parallel to reduce the time required by 8 times. The configuration is shown in Fig.2.11.

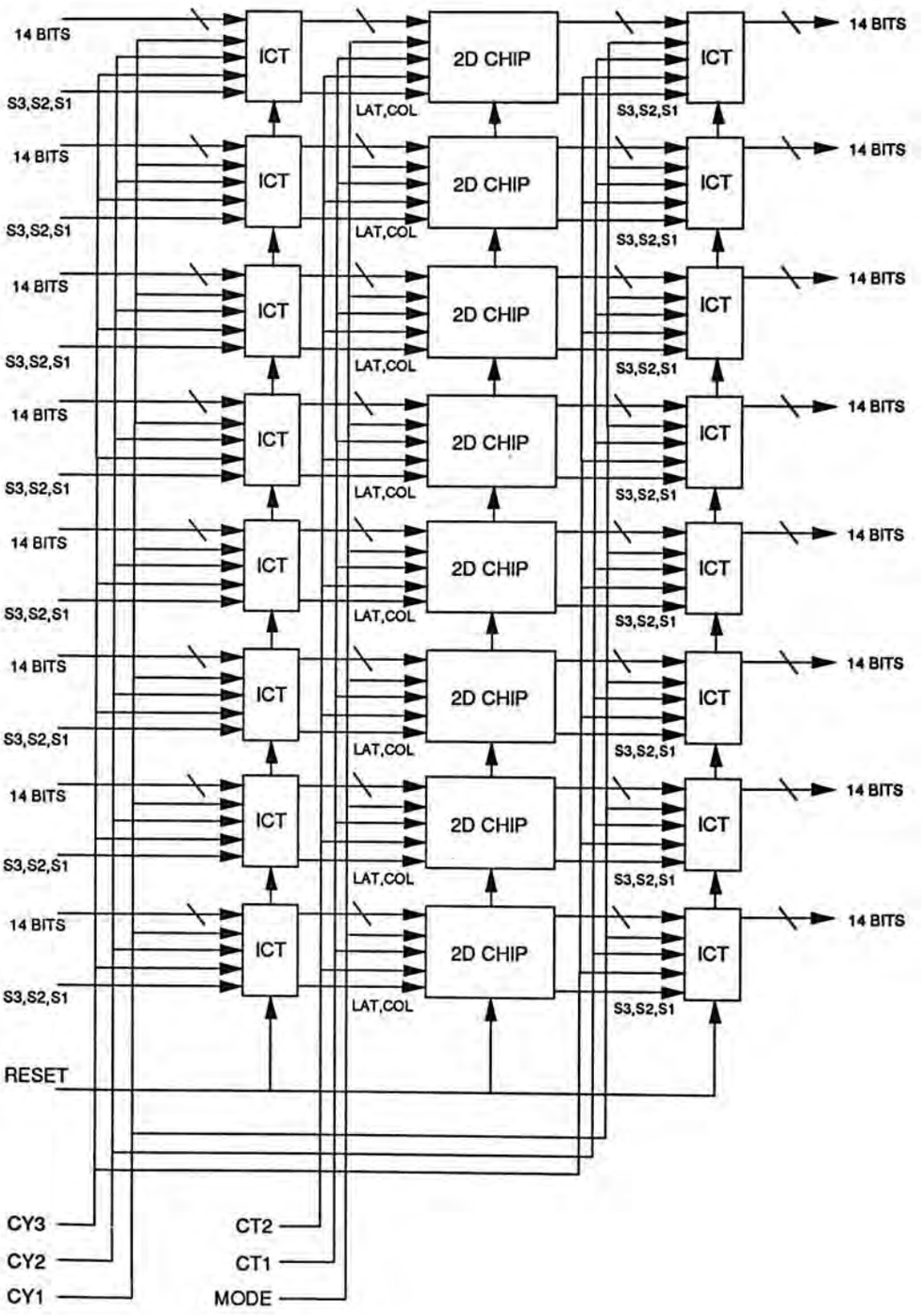


Fig.2.11 ICT unit built by 16 ICT chips and 8 Data Sequencer

2.6.2 Timing

The operation of the ICT chip is controlled by the ROW pulse signal. Data are latched in at the rising edge of the ROW pulse. After every 8 ROW pulses, the ICT chip generates a COL and LAT signal for Data Sequencer in sync with the 9th ROW pulses, data latched in during this period is invalid and the ICT chip will start another cylce at the next ROW pulse.

The Data Sequencer's operation is synchronized with the ROW and COL signals which control the shift-in and shift-out operations of the register blocks respectively. The timing sequence of the 2D transform system is shown in Fig.2.12.

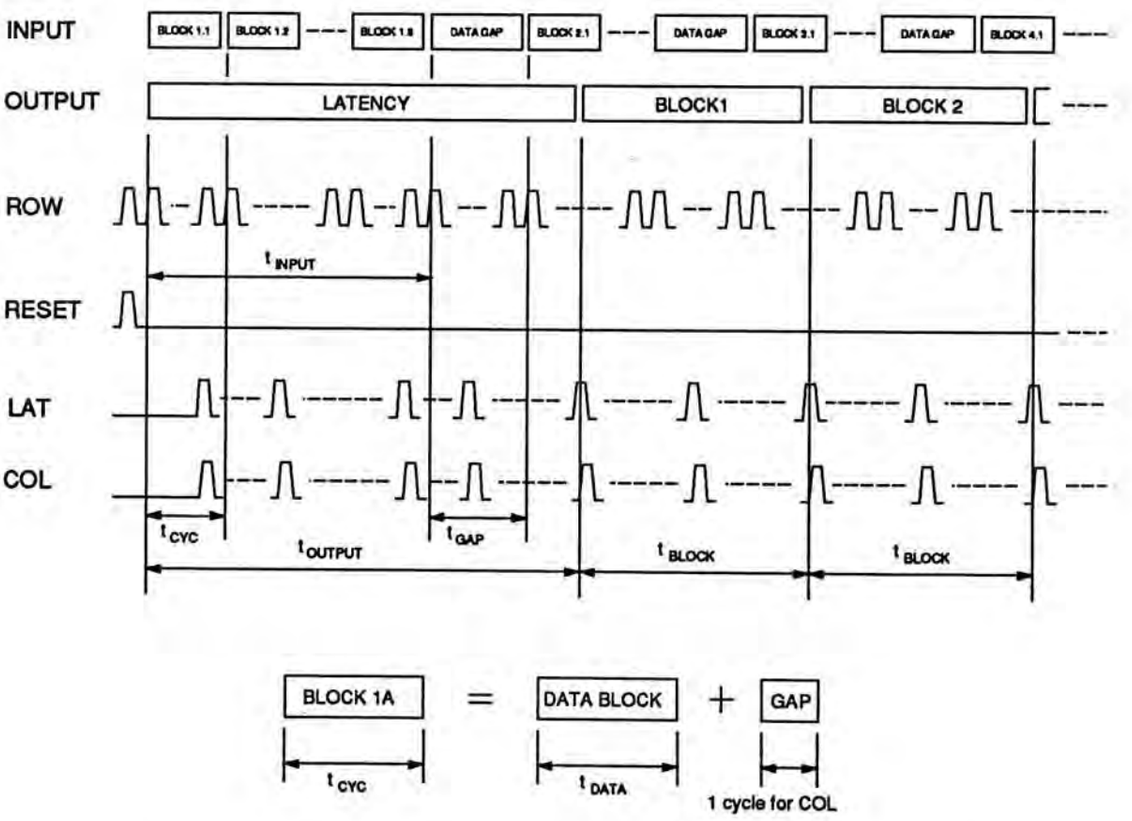


Fig.2.12 Synchronization of the 2D transform system

Where t_{CYC} equals to the number of cycles required to calculate one image vector. This includes one cycle to generate the COL and LAT pulses.

t_{INPUT} is the number of cycles required to input a sub-block of data to find the vector dot product.

t_{OUTPUT} is the number of cycles lapsed from the start of a 2-D transform operation to the first valid output.

t_{BLOCK} essentially equals to t_{INPUT} plus the time for the output to be valid.

t_{GAP} is only relevant in filtering operation. The following table shows the amount of t_{GAP} required for each degree of filtering.

Degree of filtering	t_{GAP} (Number of cycles)
2	18
4	20
6	14
8	0

Table 2.10 Degree of filtering and number of idle cycles

2.7 Conclusion

The chip set described can perform image transformation within reasonable time albeit the gate array used is not the most advanced. In addition, it is probably the first of its kind to be optimized for filtering and subsampling operations. It is expected that the complete 2-D transform system will be integrated into one chip as technology in future will allow.

CHAPTER 3 A PC/AT-BASED IMAGE ARCHIVING SYSTEM

3.1 Introduction

The surge of image compression brings interest on new applications such as picture data bases, facsimile and video teleconferencing. Over the twenties years, techniques for image compression have steadily improved. The amount of computation required by compression is reduced significantly. One of the common techniques implemented is the Differential Pulse Code Modulation (DPCM) but it is not adequate at high compression ratios. Moreover, it has two major shortcomings. They are the slope overload and the unpleasant streaks caused by bit errors. However, its simpler implementation is very favourable to be used in some compression products whose storage is not a constraint, e.g. Still-video camera and compact disc. For those applications which display high -quality images and transmit data in a low-bit rate channel, the DPCM method is not adequate.

Recently, many products [27-36] have used transform coding to condense image data at compression ratios of up to 16 to 1. Because it demands a great deal of computation, its real time implementation is only possible by using high-speed chip fabricated by the advanced VLSI technology. New applications such as real time image archiving system (IAS) can be built by the transform coding chip.

In this chapter, we propose architecture of an IAS based on the IBM-AT computer that uses the integer cosine transform. The specifications and the images's storage format will first be discussed. In section 3, the architecture of the IAS and how it performs data compression, low pass filtering and convenient image searching will be described. The software of the IAS will be analyzed in section 4.

3.2 Design Consideration

A photograph having resolution of 400 dpi (dot per inch) may be defined as good-quality photograph. To code such a good-quality colour photograph of 10 inch x 10 inch, a total of 4K x 4K x 3 bytes will be required. At present, the capacity of a 5 1/4 inches optical disk is about 1.2G-byte, and so can only store 25.6 images. In order to store more images on an optical disk, we can increase the optical disk

capacity or reduce the size of an image by means of data compression technique. Image Archiving System (IAS) employs this technique to display pictures from a digital photo album. In general, this IAS should have the following features :

(1) It should have data compression ability.
(2) As a very high resolution monitor is scarce and expensive, an IAS should be able to display a high resolution image on a low resolution monitor. In other words, the IAS should be capable of performing low pass filtering of images.
(3) As the data base of the system stores a large number of images, the IAS should provide a mean to search an image promptly and conveniently.

In this chapter, we propose an image archiving system that can provide the above three functions, namely (1) data compression, (2) low pass filtering and (3) convenient image searching.

To facilitate the above three functions, images in the IAS are forwardly transformed, the result is stored in the secondary storage in the form of transform domain. When images are converted back to spatial domain, they can be zoomed or low-pass filtered. Therefore, transformation is a very important step in the system operation. Among the many transforms, the Discrete Cosine Transform (DCT) is regarded as the industrial standard for its nearly optimal performance. The DCT together with an adaptive scheme can compress the memory requirement of an image, say a head and shoulder picture of a person, to 1/16 of its original size. The DCT however requires real number arithmetic and so is difficult to implement. A similar transform, the order-8 Integer Cosine Transform, that can be implemented using simple integer arithmetic and has a performance close to that of the DCT has thus been proposed as a substitute of the DCT.

3.2.1 Specifications

3.2.1.1 Operations supported

The system is versatile. It can

(1) accept a 256/512/1024/2048 x 256/512/1024/2048 image
(2) display the image on a 256 x 256 pel frame buffer with zooming capability
(3) code and transmit the image
(4) receive and decode the image

3.2.1.2 Image Format

The image format can be classified into three classes:

(1) Pel Domain -- binary file *.pel

The image data is stored in binary form of eight bits representing one pixel. The byte assignment of this kind of file is as follows :

byte $1..n^2$: x(1,1), x(1,2), ..., x(1,n), x(2,1), ..., x(n,n).

(2) Coefficient Domain -- binary file *.cod

After the image transformation, the image data is transformed into transform coefficients which is highly decorrelated. The image can be reconstructed from these coefficients. The byte assignment of this file is as follows :

byte 1 : number of horizontal block of the image,
byte 2 : number of vertical block of the image,
byte 3 : colour,
byte 4 : storage format,

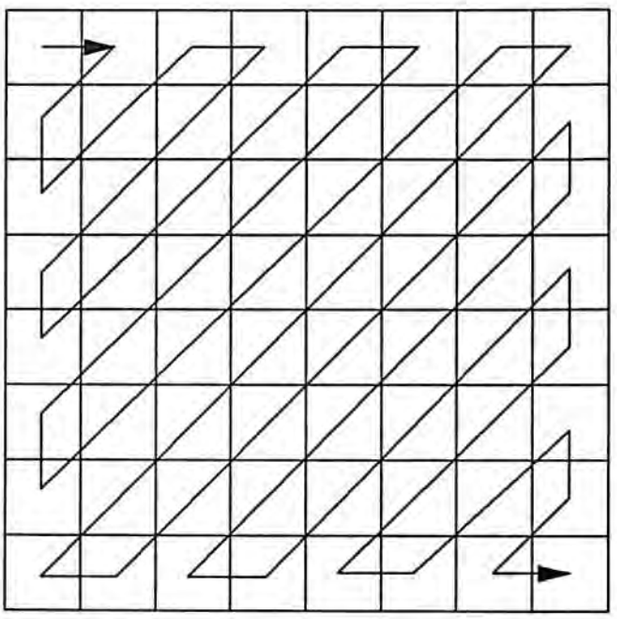


Fig.3.1 Zigzag storage path of coefficients

The coefficients are stored in the zigzag path. The two formats differ by whether to group coefficients according to their frequencies, i.e. DC, AC(0,1), AC(1,0) groups, etc.

byte 5 : (1) **Storage format 1 :**

$c(1,1)_{0,0}$, $c(1,2)_{0,0}$, ..., $c(2,1)_{0,0}$, ...
$c(1,1)_{0,1}$, $c(1,2)_{0,1}$, ..., $c(2,1)_{0,1}$, ...
$c(1,1)_{1,0}$, $c(1,2)_{1,0}$, ..., $c(2,1)_{1,0}$, ...
$c(1,1)_{2,0}$, $c(1,2)_{2,0}$, ..., $c(2,1)_{2,0}$, ...
.....
$c(1,1)_{n-1,0}$, $c(1,2)_{n-1,0}$, ..., $c(2,1)_{n-1,0}$, ...
$c(1,1)_{n,n-1}$, $c(1,2)_{n,n-1}$, ..., $c(2,1)_{n,n-1}$, ...
$c(1,1)_{n,n}$, $c(1,2)_{n,n}$, ..., $c(2,1)_{n,n}$, ...

Format 1 divides coefficients into groups of DC, AC(0,1), AC(1,0), and so on.

(2) **Storage format 2 :**

$c(1,1)_{0,0}$, $c(1,1)_{0,1}$, $c(1,1)_{1,0}$, $c(1,1)_{2,0}$, ..., $c(1,1)_{n-1,0}$, $c(1,1)_{n,n-1}$,

$c(1,1)_{n,n}$,
$c(1,2)_{0,0}$, $c(1,2)_{0,1}$, $c(1,2)_{1,0}$, $c(1,2)_{2,0}$, ..., $c(1,2)_{n-1,0}$, $c(1,2)_{n,n-1}$, $c(1,2)_{n,n}$, ...,
$c(2,1)_{0,0}$, $c(2,1)_{0,1}$, $c(2,1)_{1,0}$, $c(2,1)_{2,0}$, ..., $c(2,1)_{n-1,0}$, $c(2,1)_{n,n-1}$, $c(2,1)_{n,n}$, ...,
.....

Format 2 arranges coefficients of 8 x 8 image block in the zigzag path, coefficients in different blocks are stored in raster scanning order.

Where the subscript 0,0 represents the dc value of the 8 x 8 image block, other subscripts represent ac components of the image block, (1,1) is the image's block (1,1), (1,2) is the image block (1,2).

(3) Transmission (i.e.coded) Domain -- binary file *.txd

In this domain, transform coefficients are coded for the sake of reducing transmission or storage capacity. At this stage, our system uses the coding method proposed by Chen and Smith (C&S) in 1976. As more development is done, our system may use other coding scheme such as the JPEG which will be discussed in Chapter 5.

byte 1 : number of horizontal block of the image,
byte 2 : number of vertical block of the image,
byte 3 : colour,
byte 4 : storage format and compression code i, e.g. MSB = 0 for C&S's quantization method. The consecutive bytes will then be as follows :
byte 5... : Classification map, each data is 2 bits.
Variance map, each data is 4 bytes.

Bit map, each bit allocation data is 2 bits.
Coefficient codes stored in zigzag path as data in coefficient domain.

The sizes of class map, variance map and bit map depend on the image size.

3.2.1.3 Software

The software should be capable of controlling the display of any domain of files, *.pel and *.txd. The *.txd files can be zoomed in or out. The software can perform the file conversion between the pel domain, coefficient domain as well as the merging of the *.txd files into a photo album. In our proposed system, only files in spatial and transmission domains are saved. To perform the above operations, a user-friendly menu of the software is written for easy access.

3.2.2 Storage Format of the Coded Image

There are two storage formats, (1) grouping of same frequency coefficients in different 8 x 8 block and (2) grouping of coefficients in the same 8 x 8 block.

Grouping of same frequency coefficients may reduce the time required to fetch portions of data from hard disk or transmit them through the channel in subsampling and low-pass filtering operations. However, if all coefficients are required to reconstruct the image, the system may only begin to calculate the transformation after all these data are fetched. As a result, the time required for the transform may be longer than that by grouping coefficients of the same 8 x 8 block. The criteria of choosing these two formats thus depend on applications. For fast moving pictures, it is better to use format 2 because users are not concerned about zooming of the image. Therefore, subsampling operation is not interested. These applications include television broadcasting and teleconferencing. Other applications such as photo album may have a better performance if using format 1.

As the purpose of our system is to display still images, we thus suggest to use format 1.

		0th	1st	2nd	3rd	4th	5th	6th	7th	ROW
	DC Group	AC Group								
0th		1	2	6	7	15	16	28	29	
1st		3	5	8	14	17	27	30	43	
2nd		4	9	13	18	26	31	42	44	
3rd	AC	10	12	19	25	32	41	45	54	
4th	Group	11	20	24	33	40	46	53	55	
5th		21	23	34	39	47	52	56	61	
6th		22	35	38	48	51	57	60	62	
7th		36	37	49	50	58	59	63	64	
COLUMN										

Fig.3.2 Coefficients for subsampling and filtering operations

The subsampling operation divides into two degrees, i.e. 2 x 2 and 4 x 4. They accept those coefficient groups lying within

(1) 1 to 3 in Fig.3.2 for 2 x 2 subsampling operation and

(2) 1 to 10 in Fig.3.2 for 4 x 4 subsampling operation.

The low-pass filtering operation comprises three levels, the system accepts those coefficients lying within

(1) 1 to 3 in Fig.3.2 for 2 x 2 filtering operation,

(2) 1 to 10 in Fig.3.2 for 4 x 4 filtering operation and

(3) 1 to 21 in FIg.3.2 for 6 x 6 filtering operation.

The proposed scheme sequences the coefficients in such a way that the time of loading the low frequencies coefficients to the system is minimized. On the other hand, the number of coefficients read from the memory does not necessarily match

the above filtering options. The system may prematurely stop fetching coefficients. For example, if the user wants 3 x 3 filtering operation, the filtering operation may be done by selecting the 4 x 4 filtering option and correspondingly setting the bit allocation map of the unwanted coefficients to zero.

3.3 Hardware Architecture [41]

The system consists of seven PCBs connected by edge connectors. One of the PCBs is used to interface between the system and the AT computer, and the other six boards are composed of modules of input stage, input memory, ICT processor, output memory, output stage, address generator and control register. A frame buffer which can accept images of resolution of 256 x 256 may be connected to the system via an extended connector. All circuits are implemented in the wire-wrapping method, the highest processing clock rate is 0.75 MHz. Circuit diagram of the system hardware may be referred to a separate report "Circuit diagram of a PC/AT-Based Image Archiving System" [41]. Fig.3.3 is a photograph of the six boards.

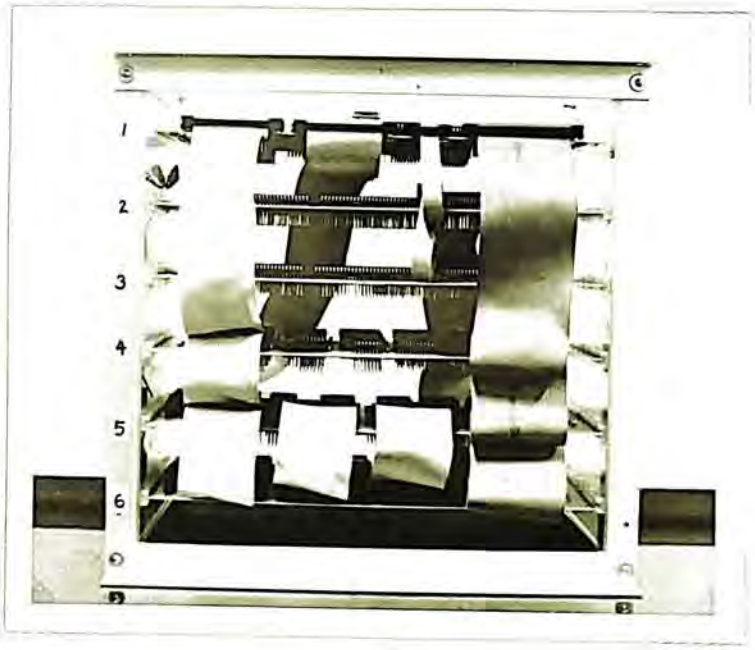


Fig.3.3 Photograph of six system boards

The function of each board is:

Board 1 - control, clock and address generator
Board 2 & 3 - ICT processor
Board 4 - bit and class maps memory module (part of the address generator)
Board 5 - input memory 1 & 2
Board 6 - output memory 1 & 2

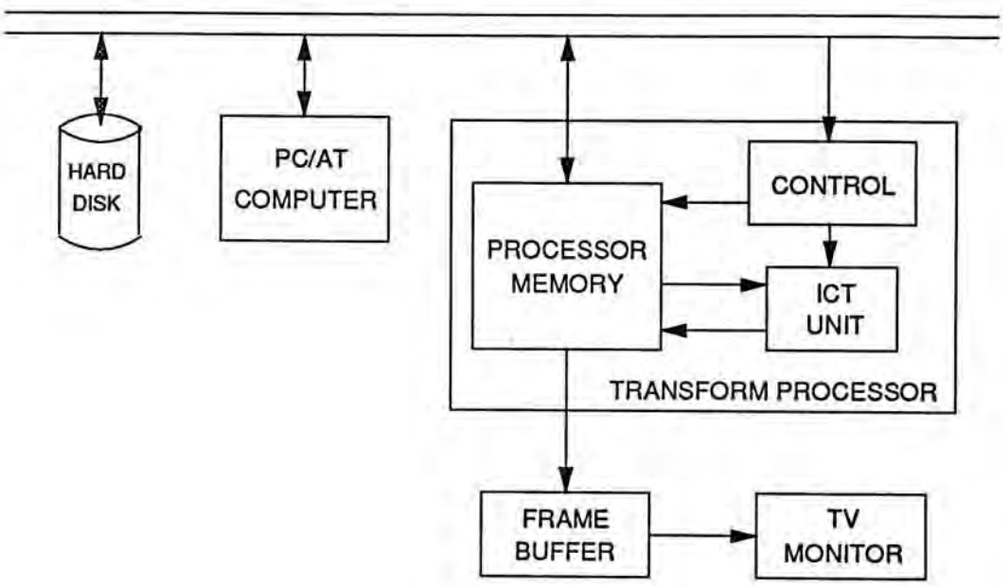


Fig.3.4 Conceptual block diagram of the IAS and the computer

A conceptual block diagram of the image system with the PC/AT computer is shown in Fig.3.4. Image taken by digital scanner are loaded from the hard disk through the disk controller to the image buffer of the AT computer. The computer uses 16-bit DMA to transfer data to the image system. Its transform result may be transferred back to the computer by 16-bit DMA. Coding of these coefficients are done by the computer, variance, bit allocation and quantization maps are then saved on the hard disk.

To retrieve the image, the computer loads the variance map, the bit allocation map, the class map and the coefficient codes of the image from the disk. An address map extra to these maps is internally required by the image system for expanding

the coefficient code to 8-bit format. This map is generated by the computer and transferred to the image system by the same 16-bit DMA as coefficient codes. Class and bit allocation maps are transferred by 8-bit I/O method. Finally, the recomposed image is returned to the frame buffer or the computer. During the operation of the image system, the AT machine is free to process other operations, therefore, the system performance is maximized. Architecture of the IAS is illustrated below:

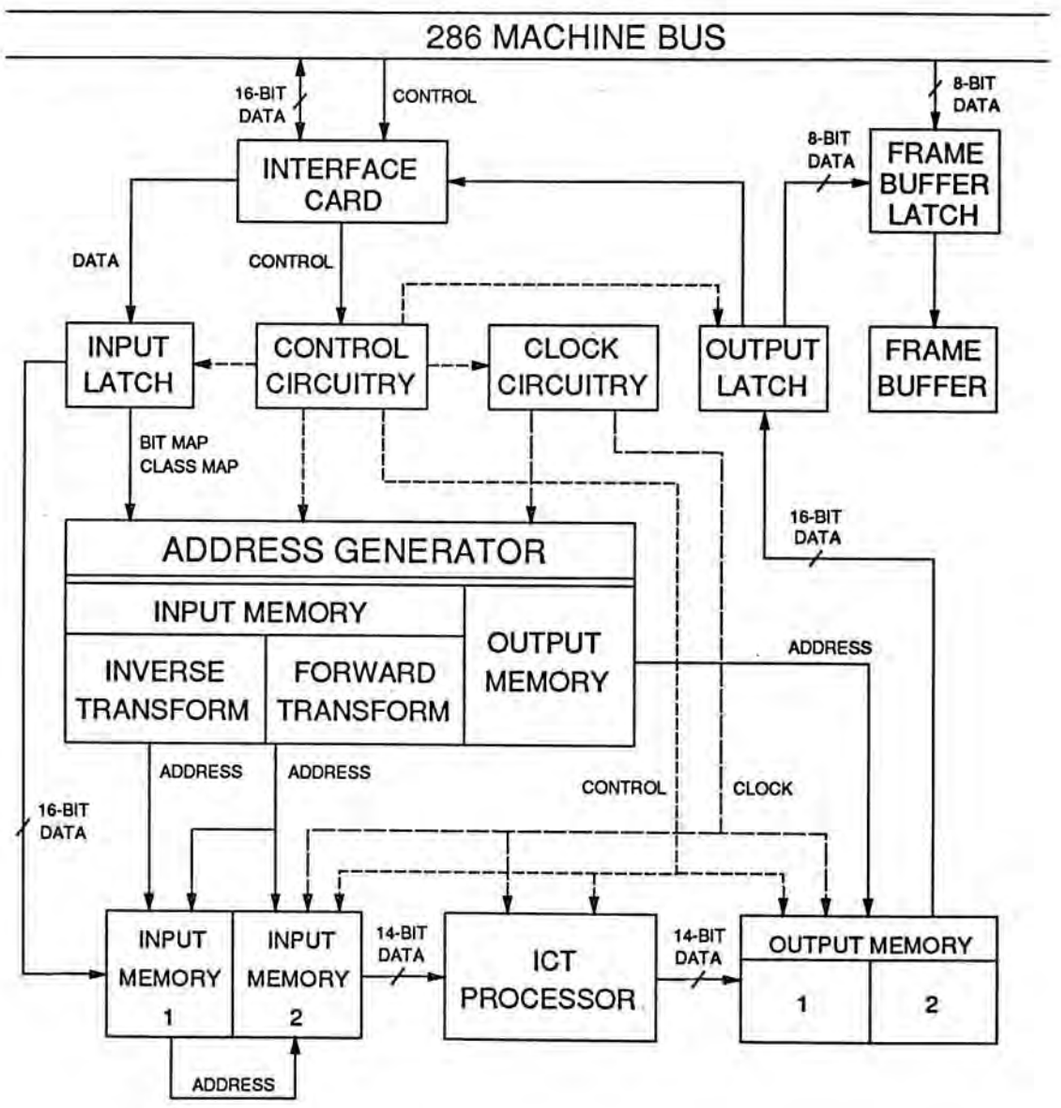


Fig.3.5 Architecture of the IAS

3.3.1 Input Stage

This stage which accepts the data transferred from the computer consists of two parts. The first part of the input stage latches the data and passes them to the second stage for storing in different memories, they are the input memory 1, the input memory 2, the address map, the bit map and the class map. The input is 16 bits in length for image data and 8 bits in length for bit allocation and class maps. A block diagram in Fig.3.6 shows the structure of the input stage.

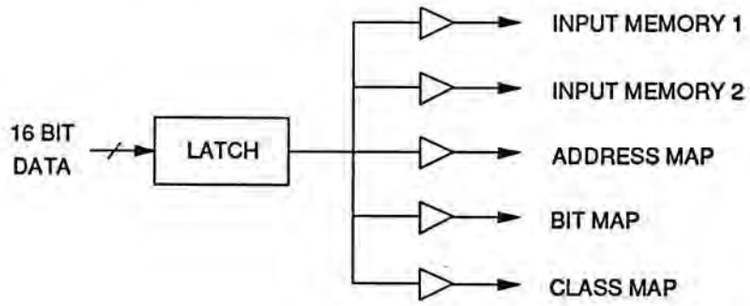


Fig.3.6 The input stage

3.3.2 Inverse Transform Address Generator

Since look-up table method is used in inverse transform, address generator is necessary for accessing the table and mapping to the memory to find the desired data. Fig.3.7 illustrates how the look-up table method can be used with the inverse transform address generator. It is worth noting that memory blocks in the diagram in fact belong to the input memory module but they are used for address mapping so it is better to include them in the diagram.

Before the inverse transform begins, the table maps should be loaded to the memory. Among those memory modules, the bit map must be loaded before the quantized table because the data loading address of the quantization table is multiplexed under the control of the bit map and other circuitry. The sequence of loading the other tables is not crucial.

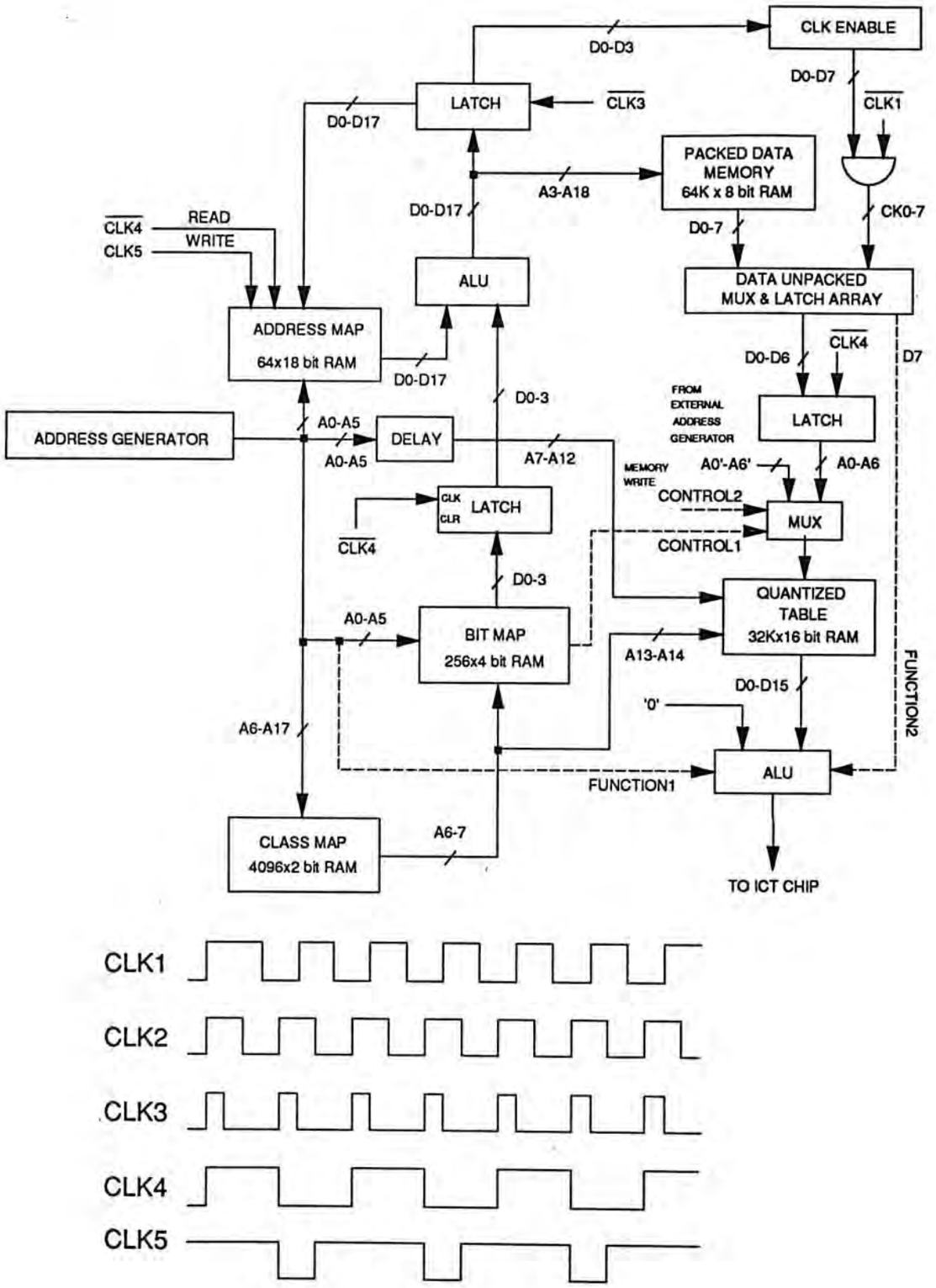


Fig.3.7 Architecture for accessing the look-up table with inverse transform address generator.

As the diagram shown, the address map, bit map and class map are directly addressed by the inverse transform address generator. The address of the packed data, the coefficient codes, is derived from an ALU which adds the values in the address map and the bit map. This address is updated each time when the rising edge of CLK3 comes. Since the input of the ICT chips is changed at the same frequency as the clock CLK4, the packed data memory is accessed twice as fast as the data input to the ICT chips. Considering a packed data stored in the memory, part of it may be stored in one byte and the remains may be stored in the succeeding byte. To unpack the data, firstly, the bit-map operand of the ALU is set to zero, the result is used to address the MSBs of the packed data. In the second memory access, the value from the bit map is added to the value from the address map, their result may address the LSBs of the packed data.

The unpacked data are in fact codes for coefficients, they are decoded by mapping to the quantization table. The highest address of this table is supplied by a delayed address from both inverse transform address generator and class map while the lowest address of this table is supplied by the data content of the unpacked data.

3.3.3 Input Memory

The input memory is built with two input memories, address generator, bit map and class map. The two memories are also used as the packed data memory as well as quantization table in inverse transform.

The two input memories are formed by four 32K x 8 bit SRAM in which two SRAMs are grouped into one module. In forward transform, image data are quantized to 8 bits. To maximize the data transfer efficiency between computer and IAS, image data are transferred in 16-bit format to the input memory. The fill-in operation of the input memories starts from memory one. When it is completed, the operation will switch to the other and the previous memory module will output its data at the same time. Since two SRAMs forms one module, they may be alternatively enabled to output the 8-bit data to the ICT processor.

In inverse transform, the two input memories store different types of data, module one for the packed data and module two for the quantization table. Their input formats are the same as in forward transform but their output formats are

different. The output of input memory 1 is still 8 bits but that of input memory 2 becomes 16 bits. Since the number of data input pins of the ICT chip is 14 only, the two LSBs of input memory 2 are disconnected to the ICT chip.

The quantization table is composed of a group of quantization tables. Each one is used for one coefficient type, i.e. DC type and 63 AC types for no class division of the coefficients, and 4 DC types and 252 AC types for four coefficients classes. Each type of quantization table is addressed by A7 to A14 from the address generator and the class map. The table content may be accessed through a multiplexer.

Other memory blocks of the input memory module are only useful for inverse transform. The minimum required sizes of these maps are 64 x 18 bits for address map, 256 x 4 bits for bit map and 4096 x 2 bits for class map when the maximum allowed picture size is 512 x 512. In implementing these memory blocks, 2K x 8 bit SRAMs are used.

3.3.3.1 Address Map

This map gives the address of the desired packed data. When the address map is first loaded, it points to the starting address of the first packed data. There are 64 starting addresses for 64 types of coefficients. The values of the address map is read out in the high state of CLK4 and updated in the low state of write signal CLK5. This updated value is equal to the summation of the value from bit map and the previous address value. In the read cycle of the address map, the MSBs of the packed coefficient are addressed. In the write cycle of the address map, the LSBs of the packed coefficient are addressed. It is worth noting that only the 16 MSBs of the ALU's result are used to address the packed data memory, but the 19-bit sum of the ALU will be written into the address map through a latch.

3.3.3.2 Bit Map

The packed data is at most 8 bits so the largest value represented by the bit map will be equal to 8 only. Thus, four bits suffice to represent the bit length of the code. The bit map can support the classification method proposed by Chen and Smith, it can at most accept 256 2-bit data. On the other hand, when the quantization

table is loaded, the value of the bit map may be used to control the address of this table. For example, if the value is 8, the quantization levels should be 256. Since half of them are negative numbers and they have the same magnitude as the positive counterparts, the quantization table only needs to store 128 levels. As a result, the time needed to load the quantization table is halved.

3.3.3.3 Class Map

Since the size of the class map is 4096 x 2 bits, it can accommodate the coefficient classification map of a 512 x 512 image. When no class is defined, the classification map does not require to be loaded. There is a control bit, namely CLASS, which may be defined by the user, to set A6 and A7 from the class map to zero all the time. The output of the class map also controls the highest address of the quantization table because different classes of coefficients may have different variances and thus different quantization levels.

3.3.4 ICT Processor

ICT processor is built with 16 ICT chips and 8 Data Sequencers. The architecture of the ICT processor has already been drawn in Fig.2.11. The input of the first stage ICT chips and the output of the second stage ICT chips are connected in parallel. The ICT processor can perform the forward transform and inverse transform. The inverse transform includes low-pass filtering, subsampling and normal operations. The ICT processor can be selected with tri-state output by setting the OEN pin of the second stage ICT chip to high. During the operation of the ICT chip, the first stage ICT chips latch their data and the second stage ICT chips output their data at the same time. By enabling the OEN pin of the second stage ICT chips sequentially, the output of each second stage ICT chip may be read by the output memory without timing conflict.

In filtering and subsampling operations, only part of the 8 x 8 block data is used, the computational time of the ICT processor is possibly reduced. However, the number of output data for filtering operation is the same as that of the normal 8 x 8 inverse transform, each second stage ICT chip will generate an output data, a mismatch of the input data number and output data number may produce some

technical problems. Therefore, from the implementation point of view, the ICT processor is designed to perform the filtering operation in the same way as the normal operation, i.e. regarding the input block of 8 x 8 data by forcing some unwanted coefficients to zero. The time reduction of filtering operation is mainly on the loading time of the data form the disk.

On the other hand, the reduction of computational time in subsampling operation is achievable since its output data is equal to its input data number. When the ICT processor latch 2 data, it will generate two output data. There is no timing mismatch between them.

3.3.5 Output Memory

The output memory also consists of two output memory modules, namely output memory 1 and output memory 2, which are built with four 32K x 8 bit SRAMs. After one of the output memory modules is filled up by the output of the ICT processor, the status of the memory modules will exchange and the filled memory module will output its data to the computer or the frame buffer. Because the input data to the output memory is 8 bits in forward transform and 14 bits in inverse transform, the memory modules can accept both 8 bit and 16 bit data input. When the memory module has 8 bit data input, its two SRAM load data alternatively; therefore, two adjacent image data will be transferred to the computer in 16-bit format.

The data stored in the output memory may be directed to the frame buffer. But the I/O size of frame buffer used is 8 bits, therefore, the two SRAMs of the memory module are accessed alternatively.

3.3.6 Address Generator

The address generator is a module comprising three address generators. The purpose of each address generator is described as follows:

(a) Address generator 1 is designed for

  (1) the data output from the input memory to the ICT processor in forward transform and

(2) the data output from the ICT processor to the output memory in inverse transform.

(b) Address generator 2 is used for

(1) the output of the ICT processor to the output memory in forward transform and

(2) the input of the quantization table to input memory 2 in inverse transform.

(c) Address generator 3 addresses

(1) the input memory for reading data from the computer in forward transform,

(2) the output memory for writing data to the computer and the frame buffer and

(3) the input memory 1 for packed data input in inverse transform.

The reasons of requiring three address generators are twofold. Firstly, the two memory modules are accessed separately, they do not have the same address at the same time, therefore, two address generators are necessary. Secondly, whenever one input or output memory module is filled up, the other module will read data from or write data to the computer, these data transfers require a third address generator.

3.3.6.1 Address Generator 1 (AG1)

This address generator is built of a 16-bit synchronous counter and a multiplexer. The purpose of the multiplexer is to arrange the output order of the counter such that different addressing sequences are provided. The architecture of the AG1 is drawn in Fig.3.8.

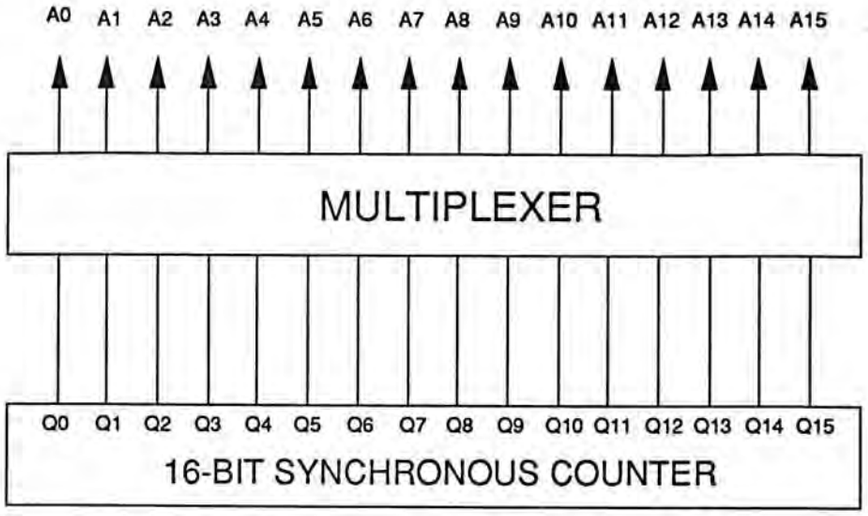


Fig.3.8 Address Generator 1

The output address sequence of the AG1 depends on the image size and the operation of the system. Conceptually, the addressing sequences of the AG1 can be classified into

(a) Forward transform

When the AG1 is used for the input memory in forward transform, the data are forwarded to the ICT processor in 8 x 8 block format. If the corresponding address of each element in a 256 x 256 image is shown in Fig.3.9, the relationship between the multiplexer's output, A0-A15, and the counter's output, Q0-Q15, will be

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
=	Q0	Q1	Q2	Q6	Q7	Q8	Q9	Q10	Q3	Q4	Q5	Q11	Q12	Q13	Q14	Q15

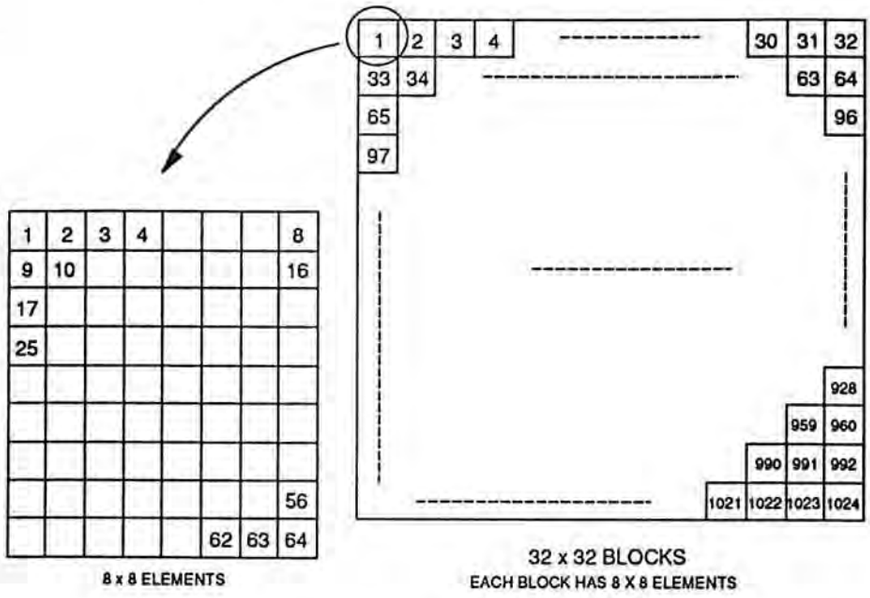


Fig.3.9 Corresponding address of each element in a 256 x 256 image

If the image is 512 x 512, the relationship between the multiplexer's output and the counter's output will become

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
=	Q0	Q1	Q2	Q6	Q7	Q8	Q9	Q10	Q11	Q3	Q4	Q5	Q12	Q13	Q14	Q15

(b) Inverse transform

In inverse transform, the output data of the ICT processor may be sent to the computer and the frame buffer; moreover, the number of output data depends upon the operation, different addressing sequences are provided here to minimize the location rearrangement time of the data. Their addressing sequences are

(1) To the computer

The output memory stores the image in a format similar to the data stored in the input memory in forward transform. The major difference is the number of data in the memory. For example, if the system performs the subsampling operation of accepting the 3 lowest sequency coefficients (2 x 2), 4K bytes and 16K bytes are required for 256 x 256 and 512 x 512

images respectively. If the 10 lowest sequency coefficients (4 x 4) are used to perform the subsampling operation, the numbers of output data will be 16K and 64K respectively. Table 3.1 shows how the multiplexer's output may be matched to the counter's output for outputting data to the computer in inverse transform.

256 x 256 image	2 x 2	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		Q0	Q2	Q3	Q4	Q5	Q6	Q1	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
	4 x 4	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		Q0	Q1	Q4	Q5	Q6	Q7	Q8	Q2	Q3	Q9	Q10	Q11	Q12	Q13	Q14	Q15
512 x 512 image	2 x 2	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		Q0	Q2	Q3	Q4	Q5	Q6	Q7	Q1	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
	4 x 4	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		Q0	Q1	Q4	Q5	Q6	Q7	Q8	Q9	Q2	Q3	Q10	Q11	Q12	Q13	Q14	Q15

Table 3.1 Multiplexer's output scheme for data transfer to computer in inverse transform

(2) To the frame buffer

The objective of this addressing format is to minimize the data transfer time between them. One possible way is to make the address sequence of the image data equivalent to that of the frame buffer. Before the data are transferred to the frame buffer, Both the address generator of the frame buffer and that of the ICT system are set to zero, these address generator will then have the same output value when they are updated each time.

The address sequence for the subsampling operations is worthy to pay special attention. The frame buffer used by the IAS has resolution of 256 x 256, it can display sixteen 64 x 64 images or four 128 x 128 images.

The addressing scheme proposed here allocates the first reduced-size image to the top left corner of the monitor, the second image to the right of the first one as shown in Fig.3.10.

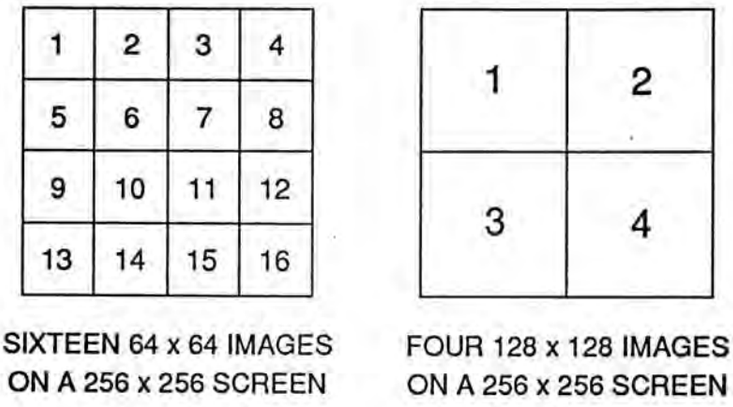


Fig.3.10 Image display format on the monitor

The addressing sequence of the subsampling operations are also given in Table 3.2.

256 x 256 image	2 x 2	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		Q0	Q2	Q3	Q4	Q5	Q6	Q12	Q13	Q1	Q7	Q8	Q9	Q10	Q11	Q14	Q15
	4 x 4	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		Q0	Q1	Q4	Q5	Q6	Q7	Q8	Q14	Q2	Q3	Q9	Q10	Q11	Q12	Q13	Q15

Table 3.2 Addressing sequence of the subsampling operations

3.3.6.2 Address Generator 2 (AG2)

This address generator is a 15-bit synchronous counter. The address tables of the counter is illustrated in Fig.3.11.

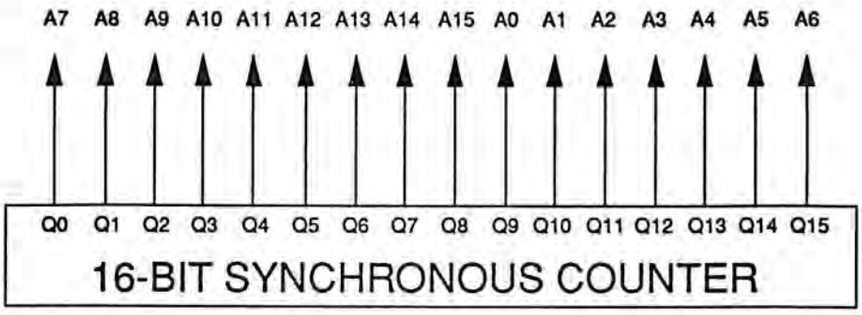


Fig.3.11 Architecture of Address Generator 2

The above address sequence drives the output memory directly. This addressing scheme results in 512 of coefficients of the same frequency stored in consecutive addresses of one memory module. If the image is 256 x 256, the other half of the same-frequency coefficients will be stored in the other memory module. Since these coefficients are necessary to find the variance and number of bits required to quantize them, the computational time will be shorter if they locate in consecutive addresses. When the AG2 is used for inverse transform, the six LSBs of the address generator are compared to the decoded value of the bit map,

3.3.6.3 Address Generator 3 (AG3)

Address Generator 3 is a 15-bit synchronous counter. It is used for loading data from the computer to the input memory and from the output memory back to the computer in forward transform. In inverse transform, it is used for loading the quantization table to input memory 2 and reading data from the output memory to the computer. The address provided by the AG3 is the value of the counter.

3.3.7 Control Register

The control register includes a status register and four flag bits which indicate the operations being performed.

The status includes

(1) Forward/Inverse transform

(2) Album display
(3) CY3 and CY2 for subsampling and filtering degrees
(4) Subsampling/Filtering operation

The flag bits are
(1) G1 and G2 for Green, Blue and Red
(2) To frame buffer/ computer
(3) 1 Class / 4 Classes

3.3.8 Interface Consideration

The system communicates with the computer through the 16-bit I/O data bus. Since the I/O data bus of the frame buffer is 8-bit only, the image output from the system to the frame buffer is reorganized to fit the format of the frame buffer. To eliminate the time required to download the data to the computer first and then transfer to the frame buffer for display, the frame buffer can obtain the image data directly from the system memory. Because the memory of the frame buffer is written through the external I/O write signal, the system thus needs to provide such a signal to the frame buffer.

3.3.9 Frame Buffer

A frame buffer which can display RGB colour image of resolution of 256 x 256 is employed. The I/O data bus of the frame buffer is 8-bit and it can give image output to RGB or monochrome monitor. The image loaded to the frame buffer can be scrolled up or down.

3.4 Software Structure

A user-friendly pull-down menu software in "C" language is written for the system. This software uses the expanded memory of the AT computer for storing the image data. Therefore, the computer can handle images of very high resolution.

Owing to the time constraint, the program is now written to accept image of resolution up to 512 x 512 only. Modification of the program to accept more detailed image is not trivial. The program structure is discussed below.

3.4.1 Main Menu

The main menu of the "ICTEXEC" software divides functions into several parts. They are

(a) Forward
(b) Inverse
(c) Album
(d) Display
(e) System
(f) Quit

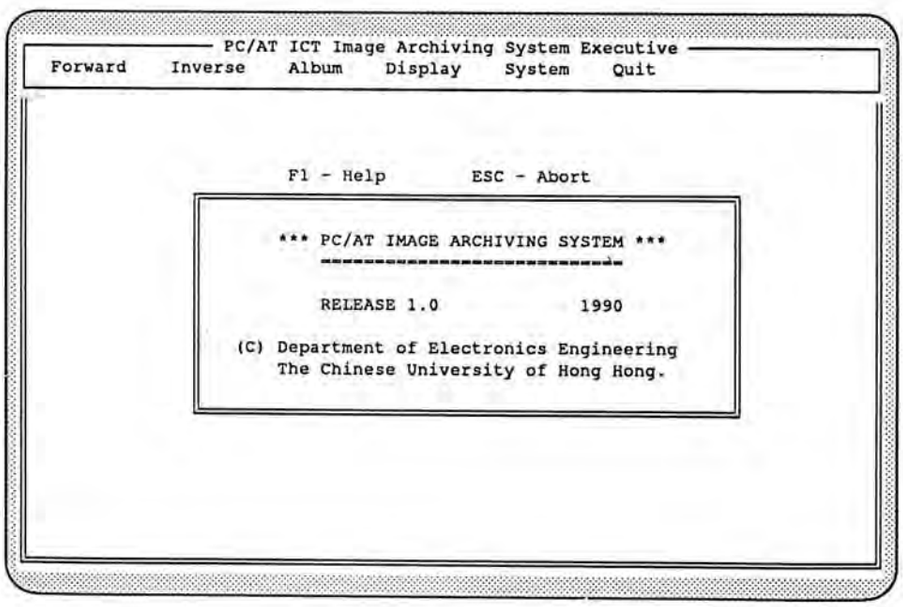


Fig.3.12 Main menu of the software ICTEXEC

3.4.2 Forward Transform

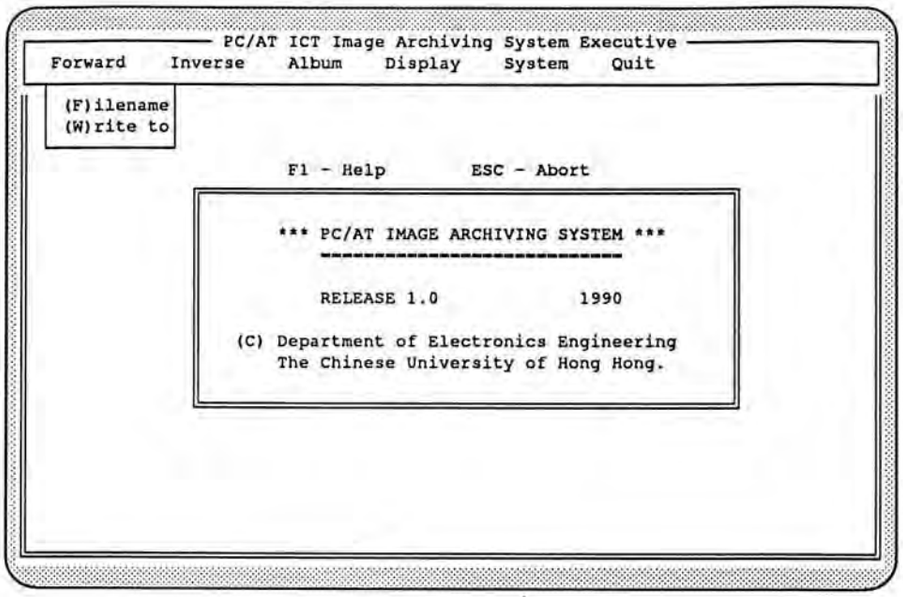


Fig.3.13 Menu of the Forward Transform option

When the user chooses the "Forward" option, image files with extension of "PEL" are listed on the screen for user's reference. The user may select one of the files for forward transform. Since the system employs the quantization and classification method proposed by Chen and Smith in 1976, the software will request the user to choose either the "1 Class" or the "4 classes" option. After the result of forward transform is sent back to the computer, the menu will ask the user to enter the number of bits for quantizing an 8 x 8 image block. The default number is 64.

3.4.3 Inverse Transform

In the Inverse Transform menu, the system may also list the file names with extension of "TXD". The Inverse Transform menu can perform the following functions.

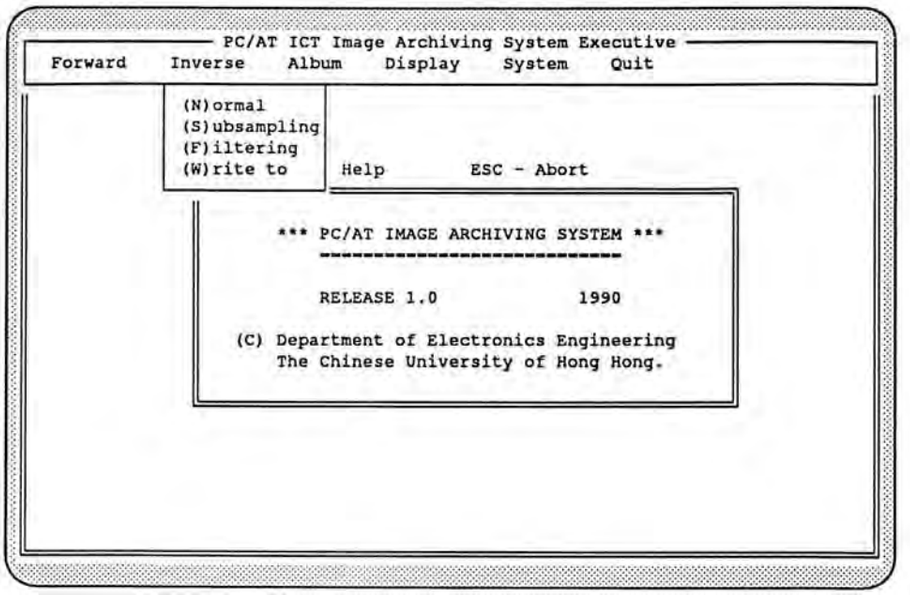


Fig.3.14 Menu of the Inverse Transform option

3.4.3.1 Normal

In this mode, the system performs the inverse transform of an image. The required information, i.e. the bit map, the class map, the coefficient lookup table, is loaded to the IAS. The computed result may be downloaded to the computer or to the frame buffer.

3.4.3.2 Subsampling

After the user enter the subsampling menu, the user may select either the 2 x 2 or 4 x 4 option for inverse transform.

3.4.3.3 Filtering

In the Filtering menu, the menu displays (a) 2 x 2, (b) 4 x 4 and (c) 6 x 6 options for the user's selection.

3.4.3.4 Album

This menu may allow the user to create, edit or display a photo album. The maximum number of pictures in each album is limited to 64. In the "Creat Album" and "Edit Album" options, the user may add or delete pictures with the help of a table which lists all the ".TXD" files. If the user enter the "Display Album" mode, a photo album will be displayed on the monitor. In the next chapter, a photograph showing the display format of the photo album on the monitor is given. The resolution of each picture of the photo album is 32 x 32. The user may choose one of the pictures for a detailed display.

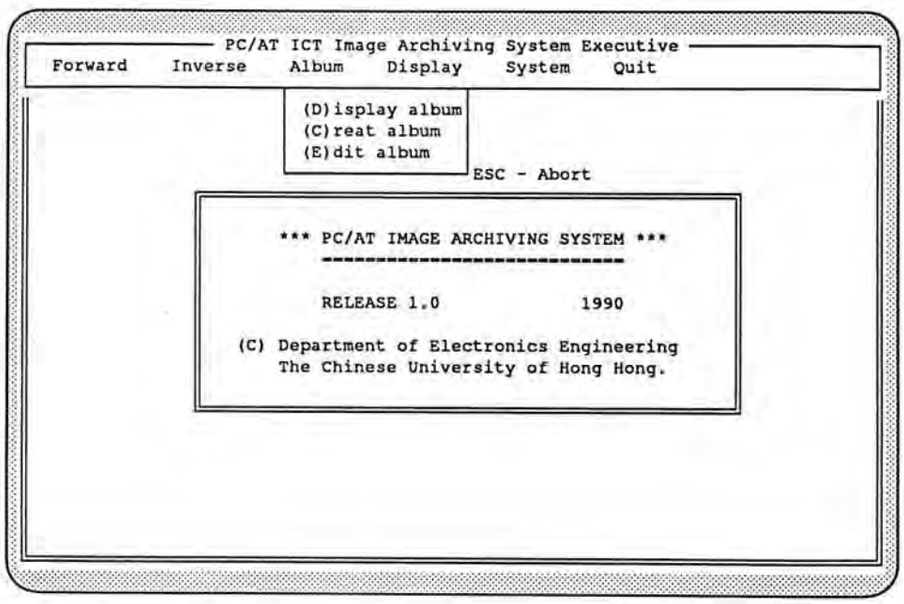


Fig.3.15 Menu of Photo Album option

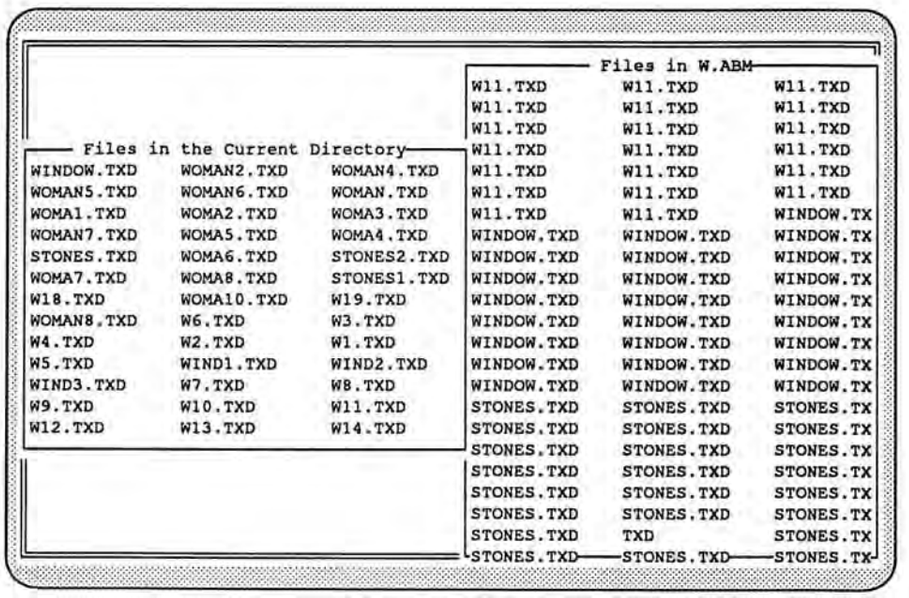


Fig.3.16 Sub-menu of Photo Album option - Edit Album

3.4.3.5 Display and System

In the Display mode, images stored in the computer can be loaded to the frame buffer for display. The user may define the position of the picture if its resolution is less than 256 x 256.

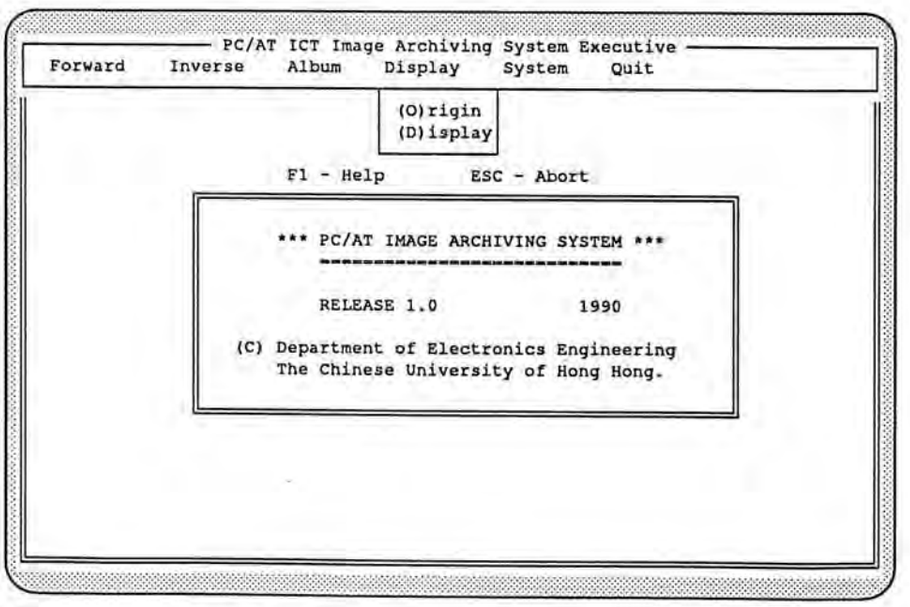


Fig.3.17 Menu of Display option

In the System mode, the user may change directory, drive, filename as well as delete file and temporarily exit to the DOS system.

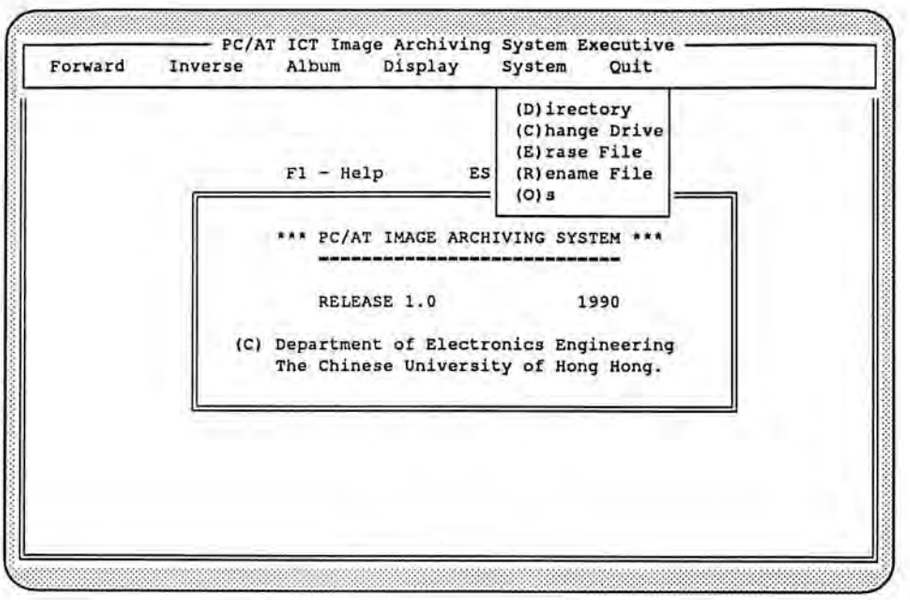


Fig.3.18 Menu of System option

3.5 Conclusion

In this chapter, we have discussed about the hardware and software of Image Archiving System (IAS). This system can perform the forward, inverse transforms. In inverse transform, normal, subsampling and low-pass filtering operations can be done. With the help of subsampling operation, this system may be used to creat, edit or display a photo album. The searching time of pictures from the photo album is minimized by a special coefficient storage format. At present, the system is built with the 74 LS series ICs by wire-wrapping technique and it is operated at 0.75 MHz data processing rate.

CHAPTER 4 SYSTEM PERFORMANCE EVALUATION

4.1 Introduction

A set of photographs taken from the system monitor is given in section 4.2. The system involves several parts which have different characteristics. These parts are the AT computer, the ICT chip set, the frame buffer, and the system associated hardware. To evaluate the performance of the system, we will look into the computation time of each part and their constraints in section 4.3.

Although image transform was well developed over the last twenty years, there are at present a few systems with which our system can be compared. Most of the present image transform products exclusively deal with transform process, e.g. DCT., our system may be the first one to display a photo album by using transform coding technique. By comparing the present transform chips, section 4.4 will discuss about the performance of the ICT chip set.

4.2 Result of Image Display

The following are the photographs taken from the system screen output after the inverse transform, original images are included for comparison.

(a) Original

(b) Normal Inverse Transform, 1 bit/pel

(c) 2 x 2 Subsampling

(d) 4 x 4 Subsampling

(e) 2 x 2 Filtering

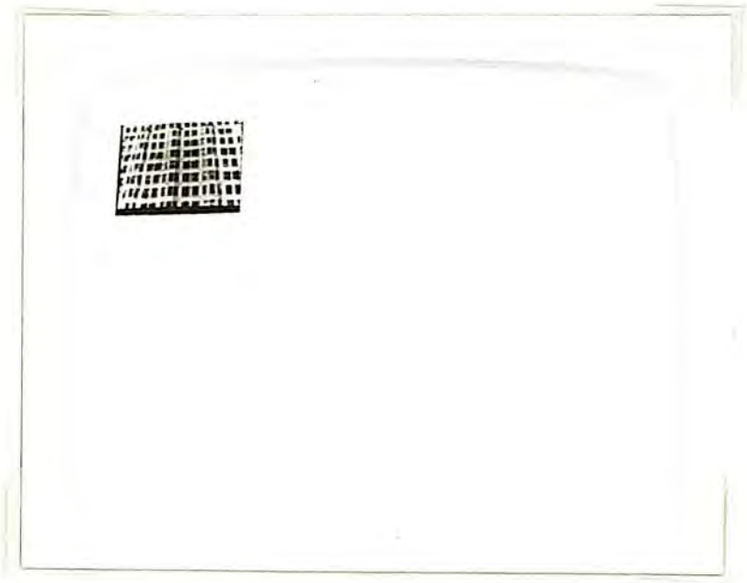
(f) 4 x 4 Filtering

(g) 6 x 6 Filtering

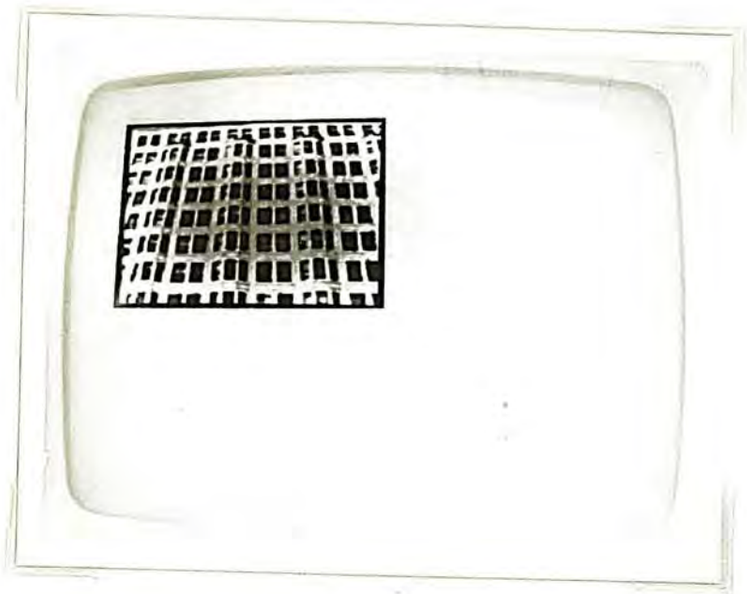
Fig.4.1 Woman

(a) Original

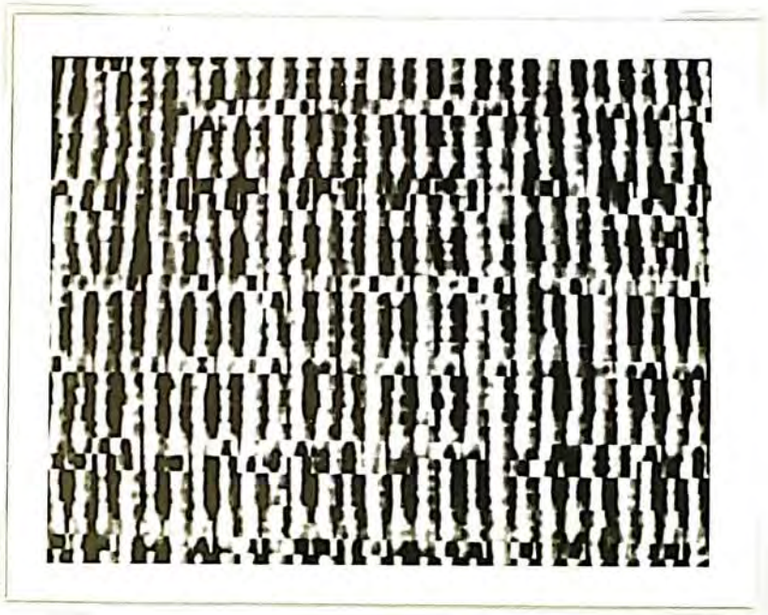
(b) Normal Inverse Transform, 1 bit/pel



(c) 2 x 2 Subsampling

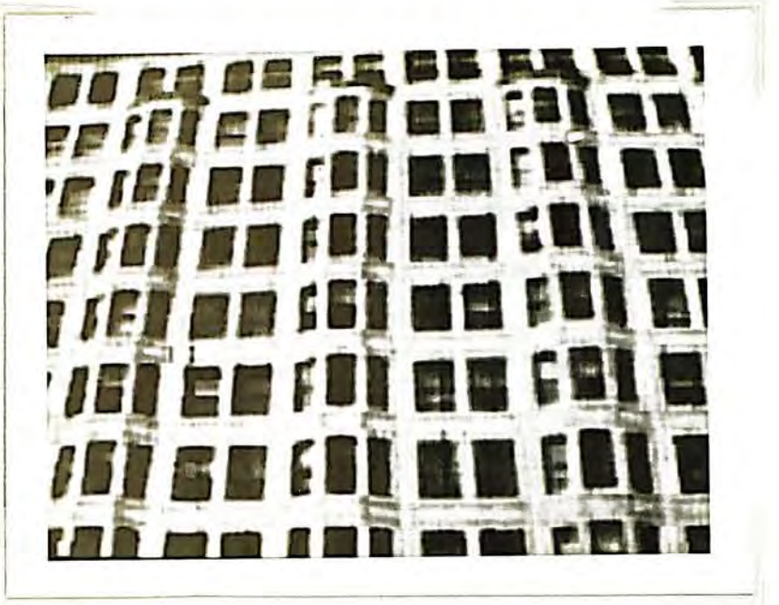


(d) 4 x 4 Subsampling



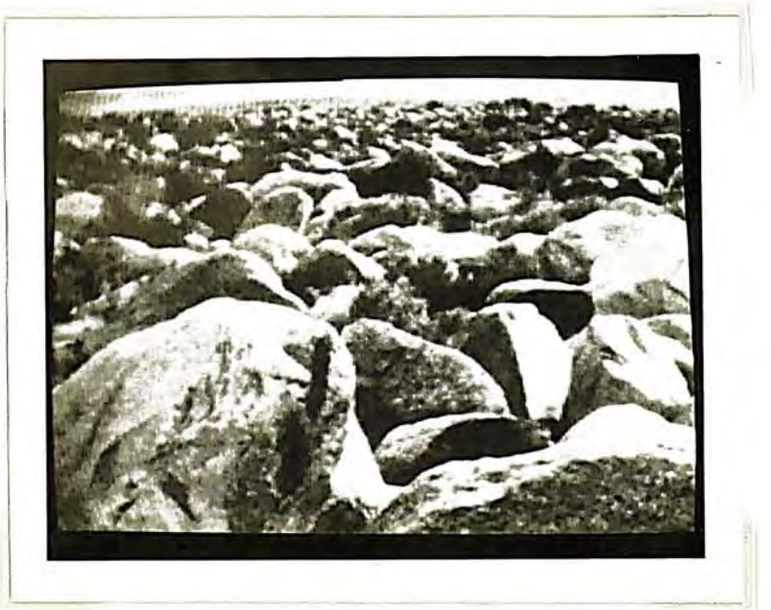
(e) 2 x 2 Filtering

(f) 4 x 4 Filtering



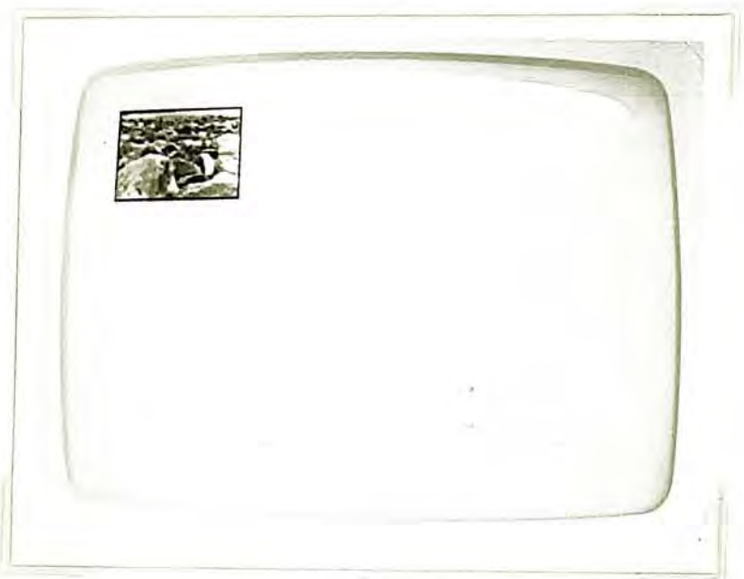
(g) 6 x 6 Filtering

Fig.4.2 Window

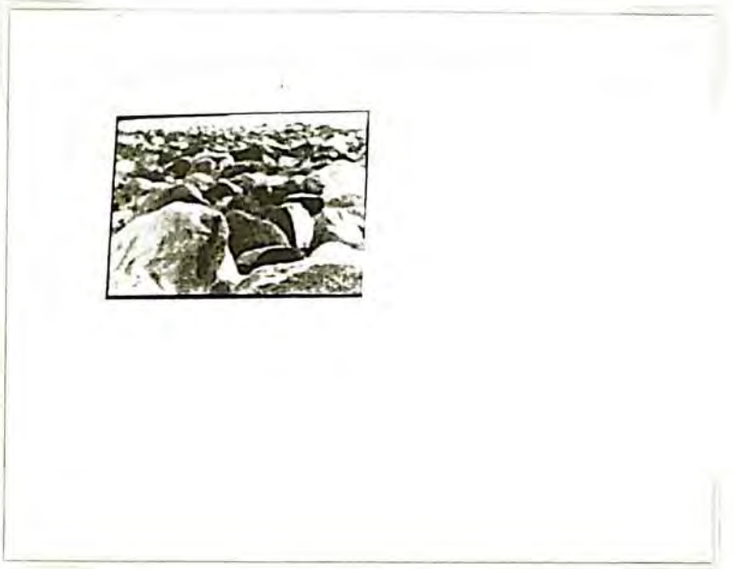


(a) Original

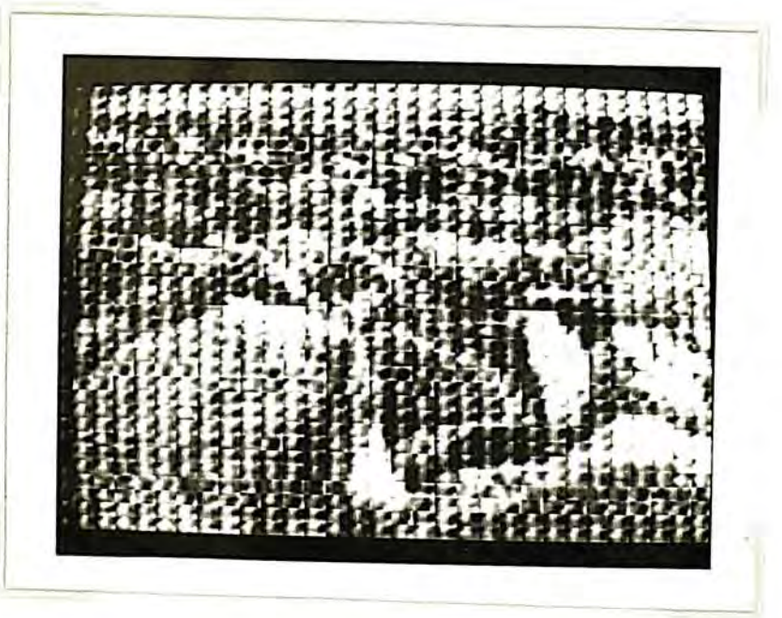
(b) Normal Inverse Transform, 1 bit/pel



(c) 2 x 2 Subsampling



(d) 4 x 4 Subsampling



(e) 2 x 2 Filtering

(f) 4 x 4 Filtering

(g) 6 x 6 Filtering

Fig.4.3 Stones

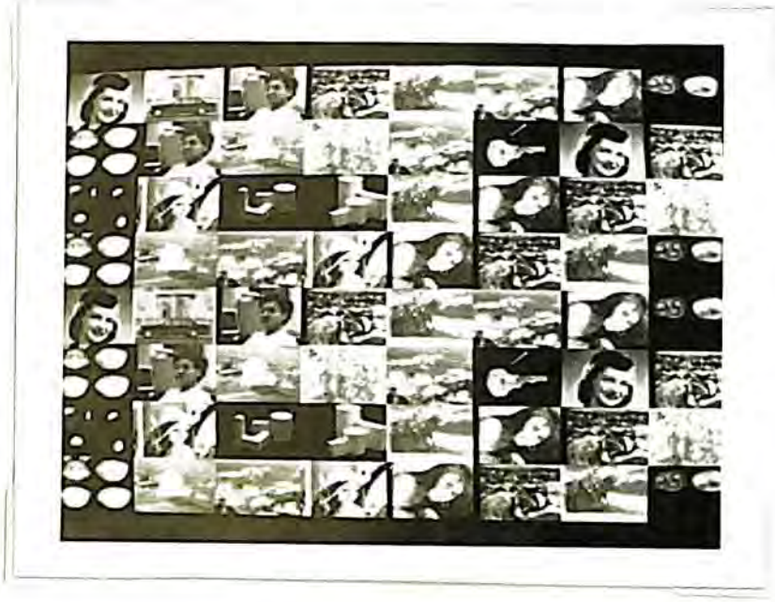


Fig.4.4 Photo Album

Fig.4.1 exhibits a small amount of detail. Fig.4.2 contains a lot of sharp edges and Fig.4.3 shows a large amount of detail.

Fig.(a) is the photoshot of the original image which is stored in spatial domain. Fig.(b) is the reconstructed image by using all the transform coefficients. Fig.(c) and (d) are the reconstructed images by using the lowest 2 x 2 and 4 x 4 sequency coefficients respectively. Since Fig.(c) and (d) subsample the reconstructed images, dimensions are reduced to 1/16 and 1/4 of Fig.(b) respectively.

Fig.(e), (f) and (g) are the reconstructed images by using the lowest 2 x 2, 4 x 4 and 6 x 6 sequency coefficients respectively but their dimensions are the same as Fig.(b).

The above results are obtained by the Chen and Smith's quantization method. Coefficients are quantized with an average bit rate of 1. bit/pel with no coefficient classification.

If four classes are used to categorize coefficients, the result will be much better. As the photographs shown, the subsampling operation gives rise to a reduction of image size, and the filtering operation gives an acceptable result as the filtering

degree is 4 x 4 or above. In the last photograph, Fig.4.4, a photo album is shown to have maximum number of pictures equal to 64, the user may select any picture for normal inverse transform, subsampling or filtering operation.

4.3 Computation Time Requirement

The system computation time includes the disk data loading time, the computer operating time, the I/O transfer time, the IAS hardware processing time and the image display time. In fact, this time depends on what kind of operations the system performs, the following assumptions are made,

(a) the image is 256 x 256,
(b) the disk accessing time is 23 msec,
(c) the disk data transfer rate is 5 MBit /sec.
(d) the CPU of the computer operates at 16 MHz.
(e) the I/O bus operates at 8 MHz, the 16-bit DMA bus cycle is 0.625 μsec.
(f) the ICT system clock is at 6 MHz, the ICT chip set operates at one-eighth of the system clock.
(g) the human response time is not considered here.
(h) the computer's calculation time of coding the transform coefficient is approximated to 10 sec in practical running. In this calculation, the computer has to find the variance of the coefficient, the bit map, the class map, the coefficient lookup table and finally the compressed codes of the image coefficients. If a predefined variance table of coefficients is used, the computation time can be reduced to one-third, further time reduction can be done by using hardware implementation.

According to different operations, the system performance is as follows.

(1) Forward Transform :

The IAS hardware may begin to calculate the transform once one of the input memory is completed in latching the 8-bit data. It is equal to 16384 16-bit DMA bus cycles. On the other hand, the computer may download the computed result

once one of the output memory is completed in latching the data from the ICT chip sets. However, the data output from the ICT chips is 14-bit, it thus requires 32768 16-bit DMA bus cycles after the ICT chip sets have finished the computation.

The time required for the computer to transfer an spatial domain image to the IAS hardware, T1

= disk accessing time + 256 x 256 bytes / disk data transfer rate
+ 16384 x DMA bus cycle time
= 23 msec + 256 x 256 x 8 / 5MHz + 16384 x 0.625 μsec
= 138 msec.

The ICT chip set calculation time, T2

= 256 x 256 x 9/8 / ICT chip set clock rate
= 256 x 256 x 9/8 / 0.75 MHz
= 98 msec.

The time required for the computer to download coefficients and to find codes.

= data download time + computer calculation time for coding the coefficients + disk accessing time + data transfer time.
= 32768 x 0.625 μsec + 10 sec.
= 10 sec

The bottle-neck of the computation in Forward Transform is therefore the coefficient coding time.

(2) Inverse Transform :

The system has to load all the maps and coefficient codes before the computation of the ICT chip. Similar to the Forward Transform, as one of the output memory is completed in latching the data from the ICT chips, the data may be downloaded to the computer or transferred to the frame buffer. In the following calculation, we assume that 64 bits are used to represent a 8 x 8 image subblock, a total of

8484 bytes including the various tables is required to transfer from the disk to the computer. From the practical running of the system, it is found that those tables require 39 msec to transfer from the computer to the IAS hardware.

The time required for data transfer from the disk to the computer

= Disk accessing time + Data size x 8 / 5 MHz + Table transfer time

= 23 msec + 14 msec + 39 msec

= 76 msec

Where Data size for

(a) Normal operation = 8484 bytes

(b) Subsampling operation and Filtering operation :

1316 bytes < data size < 8484 bytes.

Data size depends on the subsampling and filtering degrees, as well as the bit allocation of coefficient codes. However, the maximum data size is 8484 bytes. This maximum value is used in the calculation.

The ICT chip set calculation time :

(a) Normal operation = 98 msec.

(b) Subsampling operation :

(i) 2 x 2 = 64 x 64 x 3/2 / 0.75 MHz = 8 msec

(ii) 4 x 4 = 128 x 128 x 5/4 / 0.75 MHz = 27 msec

(c) Filtering operation = 98 msec.

The time for the data transferred to the frame buffer

= 32768 / 0.75 MHz

= 44 msec

The total time for the inverse transform

= 76 msec + 98 msec + 44 msec

= 218 msec

(3) Album Display

In the Album Display mode, the monitor may display the dc coefficients of 64 images as the original image resolution is 256 x 256. To load these data, the computer requires

= 64 x (Disk accessing time + data transfer time

= 64 x (23 msec + 1024 x 8 / 5 MHz)

= 64 x (23 msec + 1.7 msec)

= 1.5 sec

If one of the images is desired to zoom in for full resolution, the time required is equal to that for normal operation of the inverse transform. If it is zoomed for resolution of 64 x 64 or 128 x 128, the time required will be equal to that for the corresponding subsampling operation of the inverse transform.

4.4 Comparison to Other Transform Chips and Image Transform Systems

The 3 MHz maximum operating frequency of the ICT chip is not so promising compared with other DCT chips fabricated by advanced techniques, Table 4.1. lists some examples of the DCT chips and their maximum operating frequency. In general these chips can implement 1D DCT of different lengths and / or 2D DCT of different block sizes. Some of them integrate other operations such as quantization , coding, etc, with the DCT. The current development is to increase the throughput rate and process various block sizes. However, our ICT chip set cannot process various block sizes and at a fast throughput rate. There are several reasons for this unsatisfactory result. Apart from the old technology used, which induces problems such as long wiring length and limited number of gates, the design is also not optimized. Therefore, the performance may be greatly improved by stringent design.

Name	Technology	Function	Max. Data Rate
University of Linkoping: DCT chip [31]	4 μm CMOS SLM[1], in design and architecture status	1D 8-point, 1D 16-point	1 MHz
ENST and SGS-Thomson: DCT chip [28]	2.0 μm CMOS DLM[3] custom-made	DCT and IDCT[2] (4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16)	13.5 MHz
Universite de Neuchatel, Switzerland: DCT chip [30]	2.0 μm CMOS DLM	2D (16x16)	16 MHz
CCETT: TCAD [29]	1.2 μm CMOS DLM custom-made	2D (8x8)	27 MHz
Zoran: Digital Image Processing chip [33]	custom-made	17 high-level DSP-oriented instructions, 2D FCT[4] (8x8, 16x16) 1D FCT (8-point, 16-point)	20 MHz
Inmos: 2-D DCT Image Processor [35]	CMOS custom-made	2D (8x8), 2D linear filter or matrix transposition	27 MHz
C-Cube Microsystems: CL550A JPEG Image Compression Processor [34]	CMOS custom-made	compression/decompression scheme conforms to the CCITT/ISO JPEG standard *(refer to Chapter 5)*	28.5 MHz

Table 4.1 Performance of other DCT chips

1 SLM : Single Layer Metal

2 IDCT : Inverse Discrete Cosine Transform

3 DLM : Double Layer Metal

4 FCT : Fast Cosine Transform

When the ICT chip set was tested by the HP 16500A Logic Analysis System, it was found that the operating frequency of the ICT chip set was limited by the ICT chip rather than the Data Sequencer. The data latching clock "ROW" of the ICT chip must fulfil the requirements as shown in Fig.4.5.

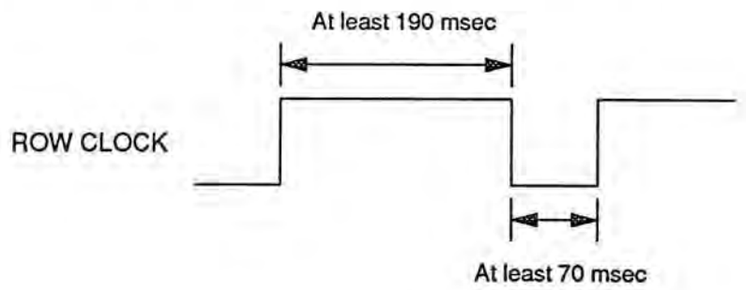


Fig.4.5 ROW clock of the ICT chip

Because of the propagation delay in the series adder structure of the ALU of the ICT chip, the duty cycle of the ROW clock in maximum frequency is larger than 0.5. The post-layout simulation of the ICT chip shows that it can operate at a clock cycle of 170 nsec and the adder delay is 90 nsec. Owing to the large difference in the practical result and the post-layout simulation result, we assume that the actual adder delay is the post-layout simulation clock cycle times 1.5 (260nsec / 170 nsec), the propagation delay of the adder will then be about 130 nsec. This delay may be greatly reduced by replacing the series adder by the look ahead carry adder. If an adder circuitry similar to the 74 series IC is implemented on the MCE's CMOS gate array LSI chip, a 4-bit look ahead carry adder may require about 60 2-input NAND gates. By series connecting 5 such adders, we may build a 20-bit adder. As one adder may comprise four levels of NAND gates, the propagation delay will be 10 nsec if 2.5 nsec is for each NAND gate. Since each 4-bit adder is serially connected, the propagation delay of a 20-bit adder will be equal to 50 nsec. Given by the MCE array design manual, a total of 140 gates is needed for the series carry adder, the additional number of gates required for the look ahead carry design is equal to 160. For a 70% utilization of the 2800 gates chip, the number of these additional gates is still within the acceptable level. With this new design, the ROW cycle will be

reduced to 190 - (130 - 50) + 70 nsec = 180 nsec which is approximated to 5.5 MHz in terms of frequency.

Owing to limited channels in each track and single layer metal technology, the shift registers of the Data Sequencer has the clock problem which can only be solved by adding buffers to the clock signal so as to direct its path as shown in Fig.2.9.

With the standard cell manual provided by Toshiba as well as reference from Motorola and Hitachi [37-39], we may estimate that the maximum frequency of the ICT chip set fabricated by different techniques as given by Table 4.2. If circuit modification is made, the frequency of the ICT chip set may be boosted by a factor of 1.44, that is equivalent to 5.5 MHz / 3.8 MHz. The estimated frequencies of this table will go into more detail in the appendix.

Technology	Delay time of 2-input NAND	Estimated frequency without modification	Estimated frequency with modification
3 μm DLM	2.5 nsec	5.7 MHz	8.2 MHz
2 μm DLM	1.5 nsec	9.5 MHz	13.7 MHz
1.5 μm DLM	1.0 nsec	14.3 MHz	20.5 MHz
1 μm DLM	0.65 nsec	21.9 MHz	31.5 MHz

Table 4.2 Estimated maximum operating frequencies of the ICT chip sets fabricated by different techniques

With the above data, the performance of the ICT chip is shown not as good as the other DCT chips. However, if the ALU of the ICT chip is replaced by the look-ahead carry adder, the performance will be improved by a factor of 1.5. As the ICT chip is fully-customed made, a comparable performance with the others is possible.

Since the hardware of the ICT system is only built by the wire-wrapping technique of the 74 LS series IC, the speed of the system can be improved a great

deal by using the ASIC design. By using the similar 3 μm CMOS single layer metal technique, the speed may be improved by a factor of about 3. Because it is possibly the first kind of system to handle digital photo album using PC/AT computer, we cannot compare its performance to other system. However, it can be seen that even low level technology is used in building this system, the system performance is acceptable.

4.5 Conclusion

This chapter has discussed about the output result of the system and its performance. It was found that even though the system is now operated at 0.75 data rate, its performance is still acceptable by most image archiving applications. The system may be improved a great deal if more advanced fabrication techniques are used. Meanwhile, the ICT chip sets are compared with other image compression chips, it is found that the ICT chip sets have room for improvement.

CHAPTER 5 CONCLUSION

5.1 Further Development

The future work of the research may use new technology for fabricating the ICT chip sets as well as the image archiving system. The system also needs to be modified for the JPEG scheme.

5.1.1 Employment of JPEG Scheme

The Joint Photographic Experts Group (JPEG) is a sub-working group of both IEC/ISO and the CCITT which has been developing a new scheme JPEG [36] for continuous grayscale or colour still-image compression since 1986. Its proposed standard gives good compression performance on images and requires less processing time than the two-pass scheme proposed by Chen and Smith in 1976.

The further work of the research may be extended to using the JPEG scheme rather than the two-pass scheme. The JPEG divides the image into non-overlapping 8 x 8 blocks. Each transform coefficient is linearly quantized with an assigned quantization value, Q(u,v). Two default quantization matrix, one for luminance components and the other for chrominance components, are given in Equation.5.1. and 5.2. After quantization, DC and AC coefficients undergo different processes.

Q_1 = Quantization matrix for luminance component

$$= \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \qquad \text{eqn (5.1)}$$

Q_2 = Quantization matrix for chrominance component

$$= \begin{bmatrix} 17 & 18 & 24 & 47 & 66 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \quad \text{eqn (5.2)}$$

(1) Coding Model for DC Coefficients

DC coefficients are coded with DPCM technique in the following way:

Code 1	Code 2
SIZE	AMPLITUDE

Where SIZE indicates the number of bits for coding the AMPLITUDE of the predicted difference, Delta, of the DPCM code. SIZE ranges from 0 to 15 as follows:

SIZE	AMPLITUDE
0	0
1	-1, 1
2	-3, -2, 2, 3
3	-7, ..., -4, 4, ..., 7
4	-15, ..., -8, ..., 8, ..., 15
5	-31, ..., -16, ..., 16, ..., 31
6	-63, ..., -32, ..., 32, ..., 63
7	-127, ..., -64, ..., 64, ..., 127
8	-255, ..., 128, ..., 128, ..., 255
9	-511, ..., -256, ..., 256, ..., 511
10	-1023, ..., -512, ..., 512, ..., 1023

11	-2047, ..., -1024, ..., 1024, ..., 2047
12	-4095, ..., -2048, ..., 2048, ..., 4095
13	-8191, ..., -4096, ..., 4096, ..., 8192
14	-16383, ..., -8192, ..., 8192, ..., 16383
15	-32767, ..., -16384, ..., 16384, ..., 32767

The SIZE of DC coefficient code is then followed by the Huffman coding (maximum length 16 bits). A default table is given below :

SIZE	Luminance code	Chrominance code
0	010	00
1	011	01
2	100	100
3	00	101
4	101	110
5	110	1110
6	1110	11110
7	11110	111110
8	111110	1111110
9	1111110	111111100
10	11111110	111111101
11	111111110	111111110

The predicted difference Code 2 is appended to the Huffman code of SIZE.

(2) Coding Model for AC Coefficients

Each non-zero AC coefficients are coded in combination with the following runlength of zero-valued AC coefficients as shown in Fig.3.1.

The AC coefficients are coded in the following way:

Code 1	Code 2
Runlength, Size	Amplitude

Where Runlength is the number of consecutive zero coefficients between non-zero coefficients,
Size is the number of bits to represent Amplitude.

Code 1 is then Huffman coded to which Code 2 specifying the sign and exact magnitude of the non-zero coefficient is appended. The End-of-Block (EOB) of Code 1 is coded by a special value 00H. An example of AC codeword for luminance is illustrated below:

Runlength / Size	Codeword	
0/0	1010	(EOB)
0/1	00	
0/2	01	
.	.	
.	.	
.	.	
0/A	1111111110000011	
1/1	1100	
1/2	111001	
.	.	
.	.	
.	.	
E/A	1111111111110100	(16 consecutive zero)
F/0	111111110111	
F/1	1111111111110101	
.	.	
.	.	
.	.	
F/A	1111111111111110	

The chrominance components have a similar code table as the luminance components.

In our proposed disk storage scheme, all the DC coefficients are grouped together, then the AC_{01}, AC_{10}, AC_{20} coefficients, and so on. These coefficient blocks are stored on the disk in the zigzag path. In the future study, Huffman coding of the coefficients will be considered. To replace the quantization method of Chen and Smith by the JPEG scheme, Huffman codes of coefficients must be stored on the disk in the same sequency as proposed for fast response to subsampling and filtering operations. Since the Huffman code includes information about other AC coefficients, i.e. the runlength of zero coefficients. As a result, the system does not know how much data should be loaded to the computer, a possible solution is to load the same number of codes as the degree of filtering or subsampling. For example, if the subsampling degree is 2 x 2, the system requires all the DC, AC_{01} and AC_{10} coefficients, it may fetch the first three blocks of codes from the disk.

5.1.2 ICT Chip Set

The ICT chip set is now only fabricated by old technology. The future work may extend to integrate the ICT chip and the Data Sequencer into one chip and use the look-ahead carry ALU rather than the series adder in the ICT chip. On the other hand, the Data Sequencer is now constructed by a set of shift registers. Because its function is to transpose the data, it may be replaced by the Dynamic RAM without any refresh circuitry as long as the period of storing and loading the data is short enough. This condition is satisfied as the operating frequency is above 100 kHz. If 1.2 μm DLM gate array technology is used, a 40,000 gates chip is enough to integrate all the 24 chips into one chip.

At present, there is an extra ROW signal for generating the LAT signal as shown in Fig.5.1 when the ICT chip latches a data vector. The result is the addition of one dummy cycle to the operation, the calculation time is thus increased. A future ICT chip may eliminate this redundant cycle as well as includes some new functions, such as selection of different ICT transform matrix (Table 2.1) for calculation, addition and subtraction of a user-defined value of the input and output data, a full range selection of subsampling and filtering degrees, etc.

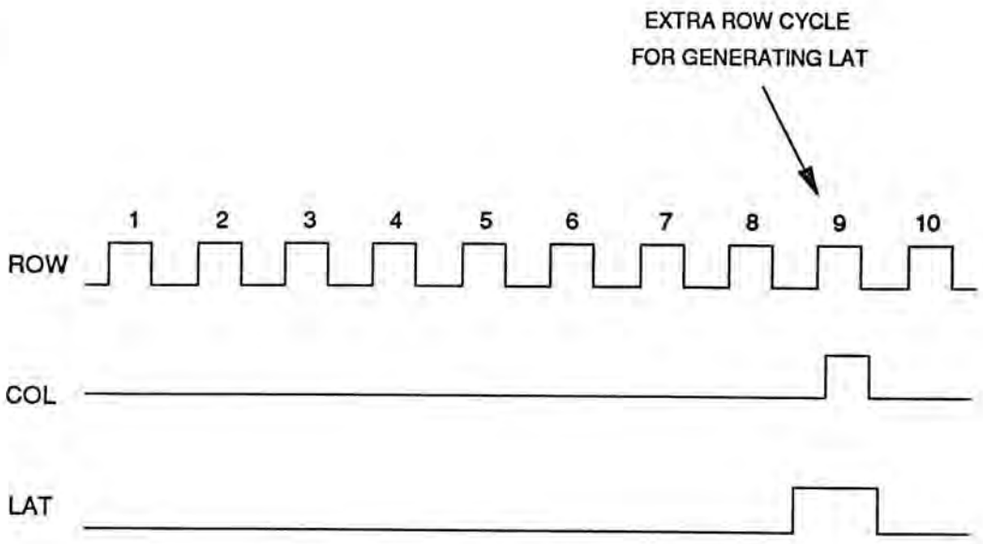


Fig.5.1 Extra ROW signal for generating the LAT signal

5.2 Summary of the Image Archiving System

This system may be used to find the ICT forward transform and inverse transform of image, as well as to display pictures of digital photo album. The image data are divided into groups of 8 x 8 blocks, each block is sent to the system for calculating the transform. The result is coded and stored on the hard disk in the sequence of DC coefficients block first, then AC_{01} coefficients block, AC_{10} coefficients block and so on following a zigzag path. Each coefficient block has 1024 elements if the resolution of image is 256 x 256.

The system employs 16 ICT chips and 8 Data Sequencer for building the ICT unit for transform operation, these chips are fabricated by 3 μm CMOS single layer metal technology. Each chip has a maximum of 2800 gates but only 70 % of the gates are used at present. The minimum clock cycle of these chips is 260 nsec.

The Image Archiving System operates in a pipe-line fashion. As one of the input memory is completed in latching data, the system begins the calculation. When one of the output memory is completed in latching its data from the system, these data may be sent back to the computer and the system continue to process its data. Therefore, the processing time is minimized.

The maximum data processing rate of the system is 0.75 MHz at present. It

can be increased as the address generators for accessing the system memory are built with fast ICs rather than the 74LS ICs. The system may complete the forward transform including coding of the transform coefficients in 10 sec. In inverse transform, it only need no more than 220 msec for a 256 x 256 image.

CHAPTER 6 REFERENCES

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

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

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

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

[5] W.H. Chen, C.H. Smith, 'Adaptive Coding of Monochrome and Color Images', *IEEE Trans. Comm.,* Vol. Com-25, pp.1285-1292, No.11, Nov 1977.

[6] W.H. Chen, W.K. Pratt, 'Scene Adaptive Coder', *IEEE Trans. Comm.,* Vol. Com-32, pp.225-232, No.3, March 1984.

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

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

[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] A.K. Jain, 'A Fast KL Transform for A Class of Random Processes', *IEEE Trans. Commun.* Vol.COM-24, pp.1023-1029, Sept. 1976.

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

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

[16] 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.

[17] 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.

[18] W.K. Cham, 'Development of Integer Discrete Cosine Transforms by The Principle of Dyadic Symmetry', *IEE Proceedings,* Vol.136, Pt.I, No.4, pp.276-282, Aug., 1989.

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

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

[21] N.S. Jayant and P.Noll, 'Digital Coding of Waveform', *Englewood Cliffs,* NJ: Prentice-Hall, 1984.

[22] K.N. Ngan, 'Image Display Techniques Using the Cosine Transform', *IEEE TRANS.*, vol.ASSP-32, No.1, Feb 1984.

[23] K.N. Ngan, 'Adaptive Transform Coding of Video Signals', *IEE PROC.*, Vol. 129, Pt. F, pp.28-39, No.1, Feb 1982.

[24] S. Matsumura, B.Sikström, U. Sjöström, L. Wanhammar, 'LSI Implementation of An 8 Point Discrete Cosine Transform', *International Conference on Computers, Systems and Signal Processing, Bangalore, India,* Dec. 10 - 12, 1984.

[25] C.S. Choy, W.K. Cham, L. Lee, 'A LSI Implementation of Integer Cosine Transform', *IEEE Singapore ICCS '88,* pp. 17.5.1 - 17.5.5, 1988.

[26] N. Ahmed, T. Natarajan, K.R. Rao, 'Discrete Cosine Transform', *IEEE Trans.*, vol. C-23, no.1, pp. 90-93, Jan. 74.

[27] F. Jutand, N.Demassieux, G. Concordel, J. Guichard, E. Cassimatis, 'A Single Chip Video Rate 16 x 16 Discrete Cosine Transform', *ICASSP 86,* vol.2, pp. 805-808, 7-11 april, 1986.

[28] N.Demassieux F.Jutand, 'A Real-Time Discrete Cosine Transform Chip', *Digital Signal Processing 1,* pp.6-14, 1991.

[29] JC. Carlach, P.Penard, JL. Sicre, 'TCAD : A 27 MHz 8 x 8 Discrete Cosine Transform Chip', *CCETT,* private publication, Sept. 1988.

[30] I. Defilippis, U. Sjöström, M. Ansorge, F. Pellandini, 'A 2-Dimensional 16 Point Discrete Cosine Transform Chip for Real Time Video Applications', *Douzieme Colloque Gretsi, Juan-Les-Pins,* 12-16, June 1989.

[31] B. Sikström, L. Wanhammar, M. Afghahi and J. Pencz, 'A High Speed 2-D Discrete Cosine Transform Chip', *Integration, the VLSI journal 5,* pp. 159-169, 1987.

[32] C.S. Choy, W.K. Lam, W.K. Cham, 'An Integer Cosine Transform Processor Unit and Application', *ITEC'91.*

[33] 'ZR36010 Image Compression Processor, Preliminary Data Sheet', *Zoran,* 1987.

[34] 'CL550A JPEG Image Compression Processor, Preliminary Data Book', *C-Cube Microsystems,* Feb, 1990.

[35] 'IMSA 121 Data Sheet : 2-D Discrete Cosine Transform Image Processor', *Inmos,* April, 1989.

[36] Joint Photographic Experts Group, 'JPEG Technical Specification, Revision 5', *ISO/IEC, JTC1/SC2/WG8, CCITT, SGVIII,* Dec. 15, 1989.

[37] 'TC24SC Series Cell Library', *Toshiba,* vol.1 & vol1.1, 1990

[38] 'CMOS Standard Cell Data', *Motorola.*

[39] 'CMOS Gate Array', *Hitachi.*

[40] K.T. Lo, 'Orthogonal Transforms in Digital Image Coding', *Master thesis, Chinese University of Hong Kong,* p.5.5, May, 1989.

[41] W.K. Lam, 'Circuit Diagram of a PC/AT-Based Image Archiving System', *Chinese University of Hong Kong,* 1991.

CHAPTER 7 APPENDIX

The estimated frequency of the ICT chip set is based on the following results:

We have designed several ASIC chips by using the same technology of MCE (Micro Circuit Engineering) Limited but no one can operate as fast as the speed estimated by its post-layout simulation. The maximum operating frequencies of all circuits in nominal situation are tested about two-third of those given by post-layout simulation. This nominal situation is 5 V supply voltage and 25°C ambient temperature.

In addition, as compared with 3 μm CMOS ASIC chips of other IC manufacturers, e.g. Motorola, a 2800 gates chip of MCE has a die size of 0.096 in^2 but a 10000 gate standard cell chip of Motorola requires only 0.123 in^2. The technology employed by MCE should thus not be as advanced as those of other manufacturers and its given gate delay time is not reliable.

The post-layout simulation of the ICT chip set shows that the chip set can operate with a minimum clock cycle of 170 nsec compared with the practical result of 260 nsec. We therefore estimate that the operating frequency of the ICT chip set will be boosted by a factor of 260/170 = 1.5 if it is fabricated by other manufacturers. The first row of Table 4.1 is calculated by multiplying the present maximum frequency 3.8 MHz by this factor. The estimated frequencies of using other technologies are computed by multiplying the estimated frequency of the 3 μm DLM technology by the inverse ratio of the gate delay time to that of the 3 μm DLM CMOS, i.e. 2.5 nsec / delay time x 5.7 MHz.