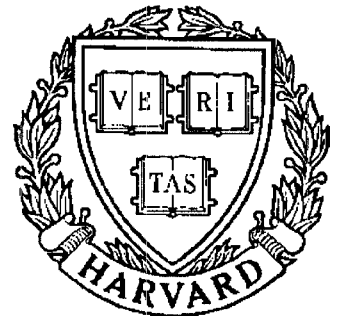


# TECHNICAL RESEARCH REPORT



S Y S T E M S  
R E S E A R C H  
C E N T E R



*Supported by the  
National Science Foundation  
Engineering Research Center  
Program (NSFD CD 8803012),  
the University of Maryland,  
Harvard University,  
and Industry*

## **Real-Time Parallel and Fully-Pipelined Two-Dimensional DCT Lattice Structures with Application to HDTV Systems**

*by C.T. Chiu and K.J.R. Liu*

# Real-Time Parallel and Fully-Pipelined Two-Dimensional DCT Lattice Structures with Application to HDTV Systems

**C.T. Chiu and K.J. Ray Liu**

Electrical Engineering Department  
Systems Research Center  
University of Maryland  
College Park, MD 20742  
Ph: (301) 405-6619

## ABSTRACT

The two-dimensional discrete cosine transform (2-D DCT) has been widely recognized as the most effective technique in image data compression. In this paper, we propose a new algorithm to compute the 2-D DCT from a frame-recursive point of view. Based on this approach, two real-time parallel lattice structures for successive frame and block 2-D DCT are developed. The systems is fully-pipelined with throughput rate  $N$  clock cycles for  $N \times N$  successive input data frame. This is the fastest pipelined structure for the 2-D DCT known so far. Moreover, the 2-D DCT architecture is module, regular, and locally-connected and requires only two 1-D DCT blocks which can be extended directly from the 1-D DCT structure without transposition. Therefore, it is very suitable for VLSI implementation for the high speed HDTV systems. We also propose a parallel 2-D DCT architecture and a new scanning pattern for the HDTV system to achieve higher performance. The VLSI implementation of the 2-D DCT using distributed arithmetics to increase computational efficiency and reduce round off error is also discussed.

---

This work is partially supported by the NSF grant ECD-8803012.



# 1 Introduction

In recent years, much research has been involved in image data compression which plays a significant role in image signal processing and transmission, especially for next generation TV, the “HDTV”. Image data compression can be classified into three catalogues, the predictive coding, transform coding, and hybrid coding [8]. In high speed image processing, transform coding is extensively used to reduce the bit rate because of its high compressional capability. The Karhunen-Loeve transform (KLT), which minimizes the mean square error of the system, is the optimal transform among various transform codings but is rarely employed due to its computational complexity. The suboptimal transform, the discrete cosine transform (DCT), which possesses superior energy compaction properties and near optimal performance but much simpler computations, is the most popular transform coding in image processing.

It is well known that block coding based on the two dimensional (2-D) DCT produces a highly compact 2-D transformed coefficients on the spatial domain [3]. By applying appropriate bit allocations and entropy coding schemes, *i.e.*, variable length coding and run length coding [34, 28], the bit rate of the HDTV systems can be greatly reduced [23, 28]. In practical applications, only small transform block size (typically 8 by 8 or 16 by 16) is utilized due to the hardware limitations. Many 2-D DCT algorithms are proposed to decrease the computational complexity and increase operational speed. They can be briefly divided into two groups, the row-column decomposition method and the direct 2-D method. The row-column method decomposes the 2-D DCT into two one-dimensional (1-D) DCT. That is, the 2-D DCT can be obtained by applying the 1-D DCT on the rows (or columns) of the input data frame, putting the transformed results in an intermediate matrix, transposing the matrix, and doing the 1-D DCT again on the columns (or rows) of the transposed matrix. Since there are lots of 1-D DCT algorithms, so are the row-column decomposition methods.

The performance of different row-column decomposition methods depends on that of the 1-D DCT algorithms. A classification of the 1-D DCT algorithms and their comparisons are given in [6]. Up to now, most of the 2-D DCT systolic array methods, such as proposed by Lee-Yasuda, and Ma, *et al* [5, 7], generate the 2-D DCT systolic array from the 1-D DCT systolic array using the row-column method. The direct 2-D method, in contrast to the row-column decomposition method, is a complete 2-D approach. Duhamel and Guillemot in [15] list two recent developed direct 2-D methods, the indirect and direct polynomial transform (PT) for the 2-D DCT. It is shown in [14] that these kinds of direct 2-D methods are more efficient than the row-column methods. The indirect PT for 2-D DCT in [14] maps the  $N$  by  $N$  DCT into a real  $N$  by  $N$  DFT followed by a number of rotations, and the real  $N$  by  $N$  DFT can be realized using PT. The direct PT for the 2-D DCT is shown to be more effective than the indirect approach [9], but the procedure for the general implementation is complicated. Up to now, most of the 2-D DCT chips are implemented by using row-column decomposition method, so are the DCT-based HDTV systems. It is due to its simplicity in hardware realizations. For HDTV systems, parallel DCT coding schemes are employed to satisfy the very high speed sampling rate requirements [36, 23].

The above mentioned approaches do have the same fundamental assumption : all the data in the processing 2-D block are available. This may not be true for the real-time data transmission systems such as HDTV systems. To eliminate the waiting time for the data to arrive, time-recursive processing concept can be exploited, *i.e.*, the result is adaptively updated when a new data arrives. Once all the data arrive, the result is completely available. The most important question here is that are we able to design a real-time VLSI system that is compatible to the data transmission speed? One of the major contribution of this paper is to answer such question for the 2-D DCT. In [6], Liu and Chiu proposed new unified parallel lattice structures for time-recursive 1-D orthogonal sinusoidal transforms. This algorithm decouples the transformed data components into  $N$  independent lattice modules,

hence, there is no global communication in the structure. Besides, every lattice module has the same architecture and is regular and modular. The number of multipliers of the lattice structure for the best case in [6] is a function of  $4N$ . Therefore, it is very suitable for VLSI implementation. In this paper, we proposed a new architecture for the 2-D DCT by employing the frame recursive concept on the successive input frame. Since this technique derives the transform from the 2-D input signal, it is a direct 2-D method. Therefore, the transposition in the row-column method is unnecessary. Moreover, the hardware complexity of this system equals that of the row-column method for series rows (or columns) input, that is, only two 1-D lattice DCT module arrays. For a  $N$  by  $N$  2-D DCT, the total number of multipliers requires is  $8N$ . The resulting architecture contains only independent lattice modules and shift registers. All the architecture of the lattice modules are the same and there are no global communications. Besides, there is no limitations on the transform size  $N$  and the structure can be easily extended into different number of  $N$ , even the prime number. Therefore, either from speed or hardware point of view, this is a rather promising architectures. Since the system is modular, regular, and fully-pipelined, it is very suitable for high speed video signal transmission. We will show that by employing distributed arithmetic technique in hardware implementation, the system performance can be further improved. For the HDTV applications, very high operating speed can be achieved by appropriate scanning arrangement and parallel processing.

We organize the rest of the paper as follows. In Section 2, the algorithm to achieve the 2-D DCT from the time-recursive manner is proposed. It will be shown that we can dually generate a 2-D discrete sine cosine transform (DSCT) simultaneously. The architectures for calculating moving frames 2-D DCT and block 2-D DCT are discussed in Section 3. A comparison between different 2-D DCT algorithms are given in Section 4. In Section 5, the VLSI implementation of the block 2-D DCT by employing the distributed arithmetic is discussed. In Section 6, the structure for applying our 2-D DCT block architecture to the

high speed HDTV system is considered. Finally, the conclusion is given in Section 7.

## 2 Dual Generation of 2-D DCT and DSCT

In this section, we will describe a new algorithm for 2-D DCT that requires only two 1-D DCT arrays and the system is fully pipelined. Focusing directly on the 2-D transformed signal and applying the frame recursive approach, we can not only derive the frame recursive relation of two successive frames of the 2-D DCT, but also the dual generation properties between the 2-D DCT and 2-D discrete sine-cosine transform (DSCT). Here the DSCT serves as the auxiliary transform which supports the time-recursive computations of the 2-D DCT.

### 2.1 Frame-Recursive 2-D Discrete Cosine Transform

The  $N \times N$  2-D DCT  $\{X_c(k, l) : k, l = 0, 1, \dots, N-1\}$  of a  $N \times N$  2-D data sequence  $\{x(m, n) : m = t, t+1, \dots, t+N-1; n = 0, 1, \dots, N-1\}$  is defined as

$$X_c(k, l, t) = \frac{4}{N^2} C(k) C(l) \sum_{m=t}^{N+t-1} \sum_{n=0}^{N-1} x(m, n) \cos \left[ \frac{\pi[2(m-t)+1]k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right], \quad (1)$$

where

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0, \\ 1 & \text{otherwise.} \end{cases}$$

Here the time index  $t$  in  $X_c(k, l, t)$  denotes that the transform starting from the  $t$ 'th row of the 2-D data sequence  $\{x(m, n) : m = t, t+1, \dots, t+N-1; n = 0, 1, \dots, N-1\}$  as shown in Fig. 1. In the following, we call  $X_c(k, l, t)$  the 2-D DCT of the  $t$ 'th frame of the 2-D data sequence  $x(m, n)$ . To derive the time-recursive relations between the successive data frame, let us start by considering the 2-D DCT of the 0'th frame data sequence,

$$X_c(k, l, 0) = \frac{4}{N^2} C(k) C(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \cos \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \quad (2)$$

Instead of focusing on  $X_c(k, l, 0)$  and utilizing various techniques to reduce the computational complexity. We will consider the 2-D DCT sequence of the first frame, which is

$$X_c(k, l, 1) = \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \cos \left[ \frac{\pi[2(m-1)+1]k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \quad (3)$$

By using trigonometric function expansions on  $\cos \left[ \frac{\pi[2(m-1)+1]k}{2N} \right]$ , (3) can be rewritten as

$$\begin{aligned} X_c(k, l, 1) &= \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \cos \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left( \frac{\pi k}{N} \right) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \\ &\quad + \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \sin \left[ \frac{\pi(2m+1)k}{2N} \right] \sin \left( \frac{\pi k}{N} \right) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \\ &= \bar{X}_c \cos \left( \frac{\pi k}{N} \right) + \bar{X}_{sc} \sin \left( \frac{\pi k}{N} \right), \end{aligned} \quad (4)$$

where

$$\bar{X}_c = \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \cos \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right], \quad (5)$$

and

$$\bar{X}_{sc} = \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \sin \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \quad (6)$$

We can view the term  $\sin \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]$  in (6) as a transform kernel, and define the  $N \times N$  2-D discrete sine-cosine transform (DSCT) sequence  $\{X_{sc}(k, l) : k, l = 0, 1, \dots, N-1\}$  of a 2-D data sequence  $\{x(m, n) : m = t, t+1, \dots, N+t-1; n = 0, 1, \dots, N-1\}$  as

$$X_{sc}(k, l, t) = \frac{4}{N^2} C(k) C(l) \sum_{m=t}^{N+t-1} \sum_{n=0}^{N-1} x(m, n) \sin \left[ \frac{\pi[2(m-t)+1]k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \quad (7)$$

Here  $X_{sc}(k, l, t)$  denotes the 2-D DSCT sequence of the  $t$ 'th frame of the 2-D data sequence  $x(m, n)$ . The 2-D DSCT of the 0'th plane data sequence is

$$X_{sc}(k, l, 0) = \frac{4}{N^2} C(k) C(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \sin \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \quad (8)$$

Similarly, we are interested in the 2-D DSCT of the first frame. According to the definition, it is

$$X_{sc}(k, l, 1) = \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \sin \left[ \frac{\pi[2(m-1)+1]k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]$$



$$\begin{aligned}
&= \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \sin \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \cos \left( \frac{\pi k}{N} \right) \\
&\quad - \frac{4}{N^2} C(k) C(l) \sum_{m=1}^N \sum_{n=0}^{N-1} x(m, n) \cos \left[ \frac{\pi(2m+1)k}{2N} \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \sin \left( \frac{\pi k}{N} \right) \\
&= \overline{X}_{sc} \cos \left( \frac{\pi k}{N} \right) - \overline{X}_c \sin \left( \frac{\pi k}{N} \right). \tag{9}
\end{aligned}$$

Both (4) and (9) suggest that the 2-D DCT  $X_c(k, l, 1)$  and 2-D DSCT  $X_{sc}(k, l, 1)$  can be dually generated from the intermediate values  $\overline{X}_c$  and  $\overline{X}_{sc}$  in a lattice form as shown in Fig. 2. A similar relation also exists in the dual generation of the 1-D DCT and 1-D DST [6]. The intermediate data  $\overline{X}_c$  and  $\overline{X}_{sc}$  differ from the original signal  $X_c(k, l, 0)$  and  $X_{sc}(k, l, 0)$  only in the 0'th row and the  $N$ 'th row of the input data frames. Therefore, the intermediate data  $\overline{X}_c$  and  $\overline{X}_{sc}$  can be obtained from  $X_c(k, l, 0)$  and  $X_{sc}(k, l, 0)$  by downdating the 0'th row transformed data and updating the  $N$ th row transformed data. That is,

$$\begin{aligned}
\overline{X}_c &= X_c(k, l, 0) - \frac{4}{N^2} C(k) C(l) \sum_{n=0}^{N-1} x(0, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \cos \left( \frac{\pi k}{2N} \right) \\
&\quad + \frac{4}{N^2} C(k) C(l) \sum_{n=0}^{N-1} x(N, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \cos \left[ \frac{\pi(2N+1)k}{2N} \right], \tag{10}
\end{aligned}$$

and

$$\begin{aligned}
\overline{X}_{sc} &= X_{sc}(k, l, 0) - \frac{4}{N^2} C(k) C(l) \sum_{n=0}^{N-1} x(0, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \sin \left( \frac{\pi k}{2N} \right) \\
&\quad + \frac{4}{N^2} C(k) C(l) \sum_{n=0}^{N-1} x(N, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \sin \left[ \frac{\pi(2N+1)k}{2N} \right]. \tag{11}
\end{aligned}$$

These two equations can be further simplified as

$$\overline{X}_c = X_c(k, l, 0) + \delta_c(k, l) \frac{2}{N} C(k) \cos \left( \frac{\pi k}{2N} \right), \tag{12}$$

and

$$\overline{X}_{sc} = X_{sc}(k, l, 0) + \delta_c(k, l) \frac{2}{N} C(k) \sin \left( \frac{\pi k}{2N} \right), \tag{13}$$

where

$$\delta_c(k, l) = \frac{2}{N} C(l) \sum_{n=0}^{N-1} \left[ (-1)^k x(N, n) - x(0, n) \right] \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \tag{14}$$

Substituting  $\overline{X}_c$  and  $\overline{X}_{sc}$  in (12) and (13) into the updated transformed signal  $X_c(k, l, 1)$  and  $X_{sc}(k, l, 1)$  in (4) and (9), the relation between the updated transformed signal and previous transformed signal can be represented in a lattice form as shown in Fig. 2. This lattice structure is similar to the lattice module obtained in the dual generation of the 1-D DCT and DST [6]. That is, the updated transformed signal can be generated from  $\delta_c(k, l)$  in a form which is similar to the 1-D DCT lattice module. What is worth noticing is that  $\delta_c(k, l)$  in (14) is the 1-D DCT of the data vector which is the difference between the parity of the  $N$ 'th row and 0'th row of the 2-D input sequence. Therefore, similar to [6], another 1-D DCT lattice module as shown in Fig. 3 is required to generate  $\delta_c(k, l)$ , which is obtained from  $X_c(l, 1)$ . We call this as lattice array  $I$  ( $LAI$ ) and that in Fig. 2 as lattice array  $II$  ( $LAI$ ). The difference between these two structures is that lattice array  $I$  feedbacks the outputs to add with the inputs by a delay stage. After obtaining the  $\delta_c$ , the updated transformed signal can be obtained recursively by feeding  $\delta_c$  into the lattice array  $II$ . Here we can see that the 2-D DCT can be obtained by using two 1-D DCT lattice arrays. It will be shown in the next section that the 2-D DCT obtained by this time-recursive approach is fully-pipelined and no transposition is needed. This is because that by using the frame-recursive approach, we start from the transformed 2-D DCT directly and avoid calculating the 2-D DCT indirectly from the 1-D DCT. Those 2-D transforms, which apply the row-column decomposition methods, have to transpose the intermediate matrix before applying the second 1-D transform. By the frame-recursive approach, we can apply two 1-D transform structure to attain the 2-D transformed signal, while the transposed procedure can be omitted. In this way, the operational speed can be greatly increased since it is unnecessary to wait until finishing the first 1-D transform. In addition, this method can obtain the 2-D DCT and 2-D DSCT simultaneously. In addition to processing the input 2-D data sequence by rows, the frame-recursive approach can also be updated by columns. In this case, 2-D DCT and 2-D discrete cosine-sine transform (DCST) are dually generated,

and all other results are similar and can be easily derived.

### 3 Architecture of Frame Recursive Lattice 2D-DCT and 2-D DSCT

Most of the 2-D DCT architectures are implemented by the row-column decomposition method [19, 18, 20]. It is due to the fact that the architectures of most fast direct 2-D algorithms are usually irregular and globally connected, therefore it is not practical for circuit realizations. Another reason is that it is better to generate the 2-D DCT system from existing 1-D DCT circuit rather than to build a new multiplier-saved 2-D DCT architectures which is not compatible with the 1-D DCT system. So the designing time can be greatly reduced. As shown in [6], the lattice structure of the 1-D DCT, which is obtained from the time-recursive approach, is regular, modular, and without global connection. By using the frame-recursive method, the 2-D DCT can be implemented by 1-D DCT lattice arrays, which are regular, simple and suitable for VLSI implementation. We will discuss two architectures, the moving frame 2-D DCT architecture and the block 2-D DCT architecture. The moving frame 2-D DCT architecture is used for calculating the 2-D DCT of sequential frames. For example, the 2-D DCT of the 0'th frame, first frame, second frame and so on. The block 2-D DCT architecture computes the 2-D DCT of a  $N \times N$  input data matrix for every  $N$  frame, *i.e.*, the 0'th frame, the  $N$ 'th frame, the  $2N$ 'th frame and so on.

#### 3.1 The Moving Frame 2-D DCT Architecture

The 2-D DCT recursive lattice structure can be constructed according to (4), (9), and (14). The intermediate values  $\delta_c$  in (14) are functions of both transformed domain components  $k$  and  $l$ . It is noted that the  $k$  components only affect the sign of the input data sequence. Using this property, we will show two approaches, the pre-matrix method and post-matrix

method, to obtain the moving frame 2-D DCT architectures. The pre-matrix method will be discussed first.

### 3.1.1 The Pre-Matrix Method

In this method, the intermediate values of  $\delta_c$  are realized directly from (14). As we can see, the only difference between input frames at time  $t=0$  and at  $t=1$  are the 0'th row and the  $N$ 'th row. Besides, the  $k$  components only affect the sign of the new input data sequence. Thus there are only two possible input sequence combinations,  $x(N, n) - x(0, n)$  and  $-x(N, n) - x(0, n)$ . The resulting architecture is shown in Fig. 4 which consists of lattice arrays and registers only. We will describe the functions of every blocks first, then demonstrate how the system works.

The circular shift matrix  $I$  ( $CSMI$ ) is a  $(N + 1) \times N$  shift registers connected as shown in the upper part of Fig. 5. When a new input data  $x(m, n)$  arrives every clock cycle, all the data are shifted in the way as indicated in Fig. 5. The first element in the 0'th row and  $N$ 'th row are sent out for summation and subtraction as shown in Fig. 4. The upper and lower  $LAI$ s execute the 1-D DCT on the rows of the 2-D input data for the even and odd transformed components  $k$  respectively. Because the length of the input vector is  $N$  and only the discrete cosine transformed data are needed, the  $LAI$ s send out the 1-D DCT transformed data  $\delta_c$ , (obtained from  $X_c(l, 1)$ ), every  $N$  clock cycles [6]. Due to the time recursive approach used, the initial values  $X_c(l, 0)$  and  $X_s(l, 0)$  in the  $LAI$ s (see Fig. 3) are reset to zeros every  $N$  clock cycles. The circular shift array in the middle of the system is a  $N \times 1$  shift registers as shown in Fig. 6. This special shift register accepts a data vector from the  $LAI$  every  $N$  clock cycles, and, after loading the data vector, it will shift the data circularly and send the data to the  $LIII$  every clock cycle. In  $LIII$ , the  $\delta_c$  comes from the circular shift array, and  $X_c(k, l)$  and  $X_s(k, l)$  from the shift register arrays located behind the  $LIII$ . We divide the  $LIII$  into two groups, the  $LIII_{even}$  and  $LIII_{odd}$ . The  $LIII_{even}$

contains only those lattice modules for even transformed components  $k$ , while  $LAII_{odd}$  contain only the odd lattice modules. It should be noticed that this system contains two lattice array  $I$  and only one lattice array  $II$ . The shift register array is a  $2N \times N$  registers. Their operations are shown in Fig. 7.

Following is to illustrate how this parallel lattice structure works to obtain the 2-D DCT and DSCT of the 2-D input successive frames. All the initial values of the circular shift matrix  $I$  ( $CSMI$ ), circular shift array, and shift register array are set to zeros. The input data sequence  $x(m, n)$  sequentially shifts row by row into the  $(N + 1) \times N$   $CSMI$ . First we calculate the difference between the 0'th row and the  $N$ 'th row data vector of the  $CSMI$ . The two resulting combinations of the input sequence,  $x(N, n) - x(0, n)$  and  $-x(N, n) - x(0, n)$  for  $n = 0, 1, 2, \dots, N - 1$ , are used as the input sequence for the lattice array  $Is$ , which consists of  $2N$  lattice modules to calculate the 1-D DCT for  $\{x(N, n) - x(0, n)\}$  and  $\{-x(N, n) - x(0, n)\}$ . The upper lattice array  $I$  is for the even transformed components  $k$  and the lower one for odd  $k$ . Suppose the data of the input vectors arrive serially per clock cycle, it takes  $N$  clock cycles to obtain the  $\delta_c(k, l)$  for both of the input sequence. At the  $N$ 'th cycle, the  $N$  transformed data  $\delta_c(k, l)$  are loaded into the circular shift arrays,  $CSA$ , which will shift circularly and send the data out of the register into the lattice array  $II$  for different  $k$  components every clock cycle, where  $X_c(k, l, 1)$  and  $X_{sc}(k, l, 1)$  are evaluated according to (9) and (14). Since both of the lattice array  $II$  has only  $N/2$  modules, every  $\delta_c$  is floating for  $N/2$  clock cycles. It is noted that a specific 2-D transform data  $X_c(k, l, t)$  and  $X_{sc}(k, l, t)$  are updated recursively every  $N$  clock cycles from  $X_c(k, l, t - 1)$  and  $X_{sc}(k, l, t - 1)$ . Therefore the outputs of the lattice array  $II$  are sent into the shift register array ( $SRA$ ) where data are delayed by  $N$  clock cycles. Each  $SRA$  contains  $N/2$  shift registers each with length  $N$ . The data in the rightest registers are sent back as the  $X_c(k, l, t - 1)$  and  $X_{sc}(k, l, t - 1)$  of lattice array  $II$ . At the  $N^2$  clock cycle, the 2-D DCT and DSCT of the 0'th frame are available. After this, the 2-D transformed data of successive frame can be

obtained every  $N$  clock cycles.

We observe two interesting results in the pre-matrix method. First, the lattice array can be viewed as a filter bank. It is because every lattice module itself is an independent digital filter with different frequency components  $l = 0, 1, \dots, N-1$ . Besides all the lattice modules in this architecture have the same structure. Second, the system requires 3 1-D DCT array and is fully pipelined with throughput rate  $N$  clock cycles. From the above discussion, transposition for the row-column decomposition method is unnecessary in this realization. According to the 1-D DCT architecture proposed in [6] (Liu-Chiu2 architecture), the total multipliers required in the 2-D DCT is  $12N$  and total number of adders is  $15N$ . Due to the goal to pipe out the results every  $N$  clock cycles, it requires three 1-D DCT structures in the system. We will show how to use only two 1-D DCT structures to attain the results at the same throughput rate in the post-matrix method.

### 3.1.2 The Post-Matrix Method

The intermediate value  $\delta_c$  in (14) can be rewritten as

$$\begin{aligned}\delta_c(k, l) &= (-1)^k \frac{2}{N} C(l) \sum_{n=0}^{N-1} x(N, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \\ &\quad - \frac{2}{N} C(l) \sum_{n=0}^{N-1} x(0, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right] \\ &= \acute{X}_c(N, l) - \acute{X}_c(0, l).\end{aligned}\tag{15}$$

That is, we can calculate the 1-D DCT of the 0'th row and  $N$ 'th row of the input frame individually, then do the summation later on. Our approach is to send the input sequence  $x(m, n)$  row by row directly into the lattice array  $I$ . It takes  $N$  clock cycles for the lattice array  $I$  to complete the 1-D DCT of one row input vector, then the array sends the 1-D DCT data parallelly to the  $CSMII$  as shown in Fig. 8 and the lower part of Fig. 5. At the output of the  $CSMII$ , the 1-D transformed data of the  $N$ 'th row and 0'th row are added together according to (15) depending on the sign of the  $k$  components (see Fig.8). Then

the results are sent to *CSA*, lattice array *II*, and *SRA*, whose operations remain intact as in the pre-matrix method. The whole structure is demonstrated in Fig. 8. Therefore, by transforming the input data first, we can implement the 2-D DCT by using only two 1-D DCTs and remain the same pipeline rate. The total numbers of multipliers and adders needed for the post-matrix method are  $8N$  and  $10N$  respectively.

### 3.2 The Block 2-D DCT Architecture

In most image processing applications, the 2-D DCT are executed block by block instead of successive frames [27, 29]. Corresponding to our frame recursive algorithm, we obtain another block of input data every  $N^2$  clock cycles. Note that this is also the total time required for all the  $N^2$  data to arrive in a transmission system. Recall that in the moving frame 2-D DCT architecture, the calculation of the 0'th frame 2-D DCT is equal to the block 2-D DCT of the 0'th frame. In that case, we assume that all the initial values of *CSMI* and *CSMII* are zeros. Corresponding to (15), the second terms  $\dot{X}_c(0, l)$  are zeros. The intermediate value of  $\delta_c(k, l)$  becomes

$$\delta_c(k, l) = (-1)^k \frac{2}{N} C(l) \sum_{n=0}^{N-1} x(N, n) \cos \left[ \frac{\pi(2n+1)l}{2N} \right]. \quad (16)$$

Here (16) suggests that the *CSMII* in Fig. 8 is redundant. The architecture for the block 2-D DCT is shown in Fig. 9. Following is an example to calculate block 2-D DCT for the 0'th frame. When the row data vector arrive, the lattice array *I* performs the 1-D DCT on them. Every  $N$  clock cycles, after the last datum of each row  $x(m, N-1)$  is available, the lattice array *I* complete the 1-D DCT for every row and send the  $N$  1-D DCT transformed data to the two length- $N$  *CSAs*. The upper *CSA* translates the intermediate value  $\delta_c(k, l)$  to the lattice array *II<sub>even</sub>*, so do the lower *CSA* except the signs of the output of the *CSA* are changed before sending to the lattice array *II<sub>odd</sub>*. The operations of the lattice array *II* and the *SRA* are the same as those in the previous methods. What is marvelous for our

frame-recursive approach is that by putting the *SRAs* in the final part of the system, the transposed operations are totally not necessary. Therefore the system can be operated in a fully pipelined manner and requires only two 1-D DCT lattice structures.

## 4 Comparisons

Since most of the 2-D DCT algorithms proposed are based on manipulating  $N \times N$  block signals. The 2-D block method we discussed in section 3.2 falls into this category. We will make a comparison with other algorithms based on block signals. The block 2-D DCT architecture is a fully-pipelined serial input parallel output (*SIPO*) system with throughput rate every  $N$  clock cycles and in terms of hardware complexity, it requires only two 1-D DCT architectures without the need of transposition operations. So it is attractive not only for its efficiency in term of system throughput but also for hardware simplicity and regularity.

In the following, the comparisons between our 2-D DCT block structure and those of others are based on the number of multipliers, adders and speed. For the sake for clearness, we divide the algorithms into two groups, one is parallel input parallel output (*PIPO*), the other is serial input parallel output (*SIPO*). The fast algorithms presented by Vetterli and Nussbaumer in [13, 14], Duhamel and Guillemot in [15], and Cho and Lee in [10] belong to the former class. Vetterli's algorithm in [14] mapped the 2-D DCT into a 2-D cosine DFT and sine DFT through a number of rotations and the 2-D DFT are computed by Polynomial Transform (PT) methods [14]. This method can reduce the number of multipliers more than 50% in comparison to the row-column method based on Chen's algorithms [2] and have a comparable amount of additions. Duhamel *et al* presented a PT-based algorithm which uses direct DCT approach in [15]. This direct PT 2-D DCT method provides a 10% improvement in both number of additions and multiplications compared to Vetterli's result in [14] but



it requires complex computations. Cho and Lees' algorithm is a direct 2-D method by employing the properties of trigonometric functions. The number of multipliers are the same as that of Duhamel, but the structure is more regular, and only real arithmetic is involved. Up to now, the best results for the first *PIPO* system in terms of the number of multipliers are  $(N^2)+2N+2$ , which are obtained by Duhamel and Guillemot, as well as Cho and Lee. But assuming that all the  $N^2$  input data arrive at the same time is not practical in the communication systems. The data waiting time is  $N^2$  which is always neglected in these approaches.

The systolic array approaches proposed by Lee and Yasuda in [5], Ma in [7], and ours are in the latter class. Lee-Yasuda presented a 2-D systolic DCT/DST array algorithm based on an IDFT version of the Goertzel algorithm via Horner's rule in [5]. The latest systolic array algorithm for 2-D DCT was proposed by Ma in [7], where he showed two systolic architectures of 1-D DCT arrays based on indirect approach proposed by Vetterli-Nussbaumer [14], then exploited the 2-D DCT systolic array by using the features of the two 1-D DCT systolic arrays. This method requires  $N+1$  1-D DCT structures and the total number of time steps is  $(N^2 + 2N + 2)$  [7]. We call the block 2-D DCT structure shown in Fig. 9 based on the Liu-Chiu2 module in [6] as Liu-Chiu2D which needs only two 1-D DCT and the total time steps is  $N^2$ . The comparisons regarding their inherent characteristics are given in Table 1. Besides, the quantities comparisons in terms of the number of multipliers and adders are given in Table 2 and Table 3. As we know, the *SIPO* method is more workable in hardware implementations. Our structure requires much fewer multipliers than Ma's structure and is highly regular, systematic, and with only local communications. In addition, this lattice 2-D DCT architecture can be generated from the 1-D DCT lattice array without modifications.

## 5 VLSI Implementation Using High-Speed Distributed Arithmetics

In this section, we will discuss the circuit realization of our block 2-D DCT structure shown in Fig. 9. Since this structure contains only shift registers and  $2N$  lattice modules, which are exactly the same except the multipliers' coefficients, we can foresee the VLSI implementation of this system is rather labor saving. Every lattice module is a modified normal form digital filter [21], which has least roundoff noise and more insensitive to the coefficient inaccuracy. Due to the fact that the block operation will reset all the outputs of lattice array  $I$  and lattice array  $II$  every  $N$  and  $N^2$  clock cycles respectively, the roundoff errors will be further minimized.

In the following we will focus our discussion on the  $8 \times 8$  block DCT with 12 bit 2's complement implementation. Suppose the lattice module is based on the Liu-chiu2 module with 4 multipliers [6], then for the 2-D DCT the total number of multipliers needed is 64, which require enormous area under 2 or 1.2  $\mu\text{m}$  CMOS technology. Besides, the system throughput is also limited by the operational speed of multipliers.

In [19, 22], Sun *et. al.* proposed the first working  $16 \times 16$  DCT Chip which incorporates distributed arithmetics method. Based on this memory-oriented structure, high speed, high accuracy, and efficient hardware implementation of the 2-D DCT can be achieved. Here we adopt the distributed arithmetic in our implementations. By employing this scheme, the system will have higher accuracy because given the same number of word length the result will undergo less roundoff operations than direct implementation in terms of multipliers. The lattice module in Fig. 2 and Fig. 3 can be redrawn as shown in Fig. 10. The dashed box in Fig. 10 can be implemented by a single ROM with three inputs and four outputs. Under this realization, the roundoff errors due to the multiplication are minimized since the distributed arithmetic convert the explicit multiplication into the implicit multiplication.

Therefore, the errors of the systems are all due to the quantization errors under finite precision implementations and adder operations. Under the 12 bits 2's complement realization, the RMS error values are about 60dB [19], which is satisfactory for most applications. Assume that every input of the ROM is 2-bit long, then the lattice module can be implemented by six ROMs and 22 adders as shown in Fig. 10. The ROM size for each lattice arrays is 18432 bits. By reducing the number of bits of every input of the ROM to one, the ROM size becomes 4608 bits, which is one-fourth of the previous case, but the the number of adder is doubled.

The other way to implement the lattice module by ROM is shown in Fig. 11. Every dashed box is realized by a ROM with one input and two outputs. If the number of bit of the input signal is 4 bits, then the realization of each ROM is given in Fig. 11. Using this method, the ROM size of each lattice array is 3456 bits and the number of adders needed is 16. When the number of bit of the input signal is reduced, the ROM size is shrunk but the number of adders is increased. We will implement the system based on the schematic diagram for each lattice module as shown in Fig. 11. The adder is a 12-bits carry-look-ahead full adder/subtractor which is combined from three 4-bit carry-look-ahead adders. Since the area of ROM is much less than that of multipliers and the speed is higher, circuit implementations under this approach can be fabricated for very high speed video signal processing. The VLSI circuit design of the 2-D DCT system is in progress and will be reported in the future.

## 6 Application to the HDTV systems

In recent years, focus of video signal processing research has been done on the high-definition television (HDTV) which will become future standard for the next generation television [32]. According to the CCIR Recommendation 601, the bit rate for transmitting an uncompressed

digital HDTV is about 1Gbps. This bit rate is too high even for broadband ISDN (BISDN) [32]. Besides, video signals contain a great deal of redundancy when psychological and visual effects are considered. To make HDTV systems practical, bit rate reduction and data compression are indispensable. In the past, lots of studies have been involved in differential pulse code modulation (DPCM), subband coding, and transform coding (especially DCT) to achieve bit rate reduction [35, 30, 36]. The DCT has obtained most attention due to its diverse attractive features. As it is known, the DCT approaches the statistical optimal transform, Karhunen-Loeve Transform (KLT), which minimize the mean square errors, for highly correlated signals [1]. Besides, it has superior energy compaction properties for transform coding. Many HDTV systems based on the DCT coding schemes show satisfactory speed and promising performance [23, 27, 28, 33, 34, 35, 26]. A common used encoder configuration of the DCT based source coding is shown in Fig. 12 [34, 26]. The DCT is performed on the 2-D video signals with block size  $8 \times 8$ , which is widely used due to its acceptable SNR and implementation complexity [34].

Although there is no uniform standard for HDTV, the interlaced mode with 1080 active line per frames, 30Hz frame frequency, and 2:1 interlaced ratio is presently under widely investigations due to its reasonable data rate [24]. With an assumption of coding each pel (luminance and chrominance ) with 2 bits, the bit rate required for transmission of the video signal under interlaced modes are 119.232 Mbits/s which satisfied the requirement of 140Mbits/s H4 hierarchy level [23], and allow sufficient margin for error protection, and auxiliary data. The 4:2:2 YUV signal is obtained from RGB signal by A/D converter and coordinate rearrangement. The intrafield 2-D DCT are used for data compression. The transformed signal are processed by entropy encoder, which is usually combined by the run-length coder and variable length encoder. Run length coder can reduce the bit-rate by coding every sequence of zeros with a single codeword. The variable length coder encodes the DCT coefficients with a variable length code adapted to their probabilistic density function

(pdf) distribution.

Most of the 2-D DCT implementations are based on the row-column decompositions methods. Although fast algorithms exist for the 1-D DCT, the second 1-D DCT cannot start until all the first 1-D DCTs are completed. To speed up the operations, one method is to parallelly execute the first 1-D DCT. For the  $8 \times 8$  case, there are 8 1-D DCT blocks to perform the first transform simultaneously. Suppose each signal is 10 bits long to have satisfied precision, then the total number of bits required in the input is 640 bits, which is not practical in the circuit realizations. From this point of view, our serial input 2-D DCT system is more practical in hardware implementations. Moreover, if the speed of the circuit components, such as ROM and adder, is high enough, our 2-D DCT system can be executed as fast as the sample clocking rate.

Although our 2-D DCT implementations are effective, transforming a Video frame of  $1080 \times 1920$  still requires intensive computations. This motivates us to design a 2-D DCT architecture suitable for the HDTV system to achieve higher performance. The block diagram of the 2-D DCT encoder is shown in Fig. 13, where five 2-D DCT chips are included. It is because that the ratio of pixel numbers per line for luminance signal Y and color difference signals U, and V is 4:2:2. As the sampling frequency of HDTV is very high, the pixels of Y are divided into four groups, in order to carry out DCT in parallel. Besides, the color difference signal Y and U are switched alternatively to another DCT coder. The scanning processor in Fig. 13 is used to divide the signal into four luminance components and one color difference components. The output of the 2-D DCT transformed data are sent to the entropy encoder in parallel or through multiplexers. Since the transform block size is  $8 \times 8$ , we divide the frame to  $135 \times 240$  blocks and 240 channels as shown in Fig. 14. The 2-D DCT are executed on each channel whose scanning pattern is shown in Fig. 14. It is all due to our system is based on row by row scanning order and is fully pipelined. Thus, such a scanning method would maximize the system throughput.

## 7 Conclusions

In this paper, we propose a new 2-D DCT algorithm using frame-recursive approach since in data transmission systems the data arrive serially. Based on this method, the 2-D DCT can be obtained by using only two 1-D DCT arrays, while the transposive procedure can be omitted. Therefore, it does not have the drawback of the row-column decomposition method in which the second 1-D DCT can not start until all the first 1-D DCTs are completed. In addition, this method can obtain the 2-D DCT and 2-D DSCT simultaneously. There are two methods, the pre-matrix and post-matrix method, for the moving frame 2-D DCT architecture. From the post-matrix method, the block 2-D DCT architecture can be developed. The systems are fully-pipelined and contain only two 1-D DCT lattice array. The lattice array contains  $N$  lattice modules which are modified normal form digital filter with different multiplying coefficients. The total number of multipliers required in the system is  $8N$  and system pipelined throughput rate is  $N$  clock cycles for  $N \times N$  input signal. The structure is very regular and efficient such that it is very suitable for VLSI implementation for the high speed HDTV systems. Many HDTV systems based on the DCT coding shows satisfactory speed and promising performance since the DCT has superior energy compaction property. Most of the 2-D DCT portion of the HDTV systems are implemented by row-column decompositions methods which needs two 1-D DCTs. To speed up the throughput, the first DCT operation has to be performed in parallel, while it is not practical in the circuit realizations due to the hardware complexity. From this point of view, our serial input 2-D DCT system is more workable in hardware implementations. The parallel 2-D DCT architecture and the scanning pattern proposed in Section 6 can process the video data in real time and eliminate the waiting time in the DCT coding so that the system throughput can be maximized. Therefore, such fully-pipelined system is very attractive for high speed transmission system where every arrived data can be processed

immediately.

## References

- [1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, pp. 90-93, Jan. 1974.
- [2] W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Communication, vol. COM-25, pp. 1004-1009, Sept. 1977.
- [3] N. S. Jayant, and P. Noll, **Digital Coding of Waveform**, Prentice Hall, 1984.
- [4] D. E. Dudgeon, and R. M. Mersereau, **Multidimensional Digital Signal Processing**, Prentice Hall, 1984.
- [5] N. H. Lee and Y. Yasuda, "New 2-D systolic array algorithm for DCT/DST," Electron. Lett., 1989, 25, pp. 1702-1704.
- [6] K. J. R. Liu, and C. T. Chiu, "Unified Parallel Lattice Structures for Time-Recursive Discrete Cosine/Sine/Hartley Transforms," submitted to IEEE Trans. Signal processing.
- [7] W. Ma, "2-D DCT systolic array implementation," Electronics Letters, Vol. 27, No. 3, pp. 201-202, 31st Jan. 1991.
- [8] H. G. Musmann, P. Pirsch, and H. G. Grallert, "Advances in picture coding," Proc. IEEE, vol. 73, pp. 523-548, Apr. 1985.
- [9] H. J. Nussbaumer, and P. Quandale, "Fast Polynomial Transform Computation of 2-d DCT," Proc. Int. Conf. on Digital signal Processing, Florence, Italy, pp. 276-283, 1981.
- [10] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-D Discrete Cosine Transform," IEEE Trans. on Circuits and Systems, vol. 38, No. 3, pp. 297-305, March. 1991.
- [11] A. Rosenfeld, and A. C. Kak, "Digital Picture Processing," 2nd edition, Academic Press, 1982.
- [12] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," IEEE Trans. Acoust., Speech, Signal processing, vol. ASSP-32, Aug. 1984.
- [13] M. Vetterli and H. Nussbaumer, "Simple FFT and DCT algorithm with reduced number of operations," Signal Processing, vol. 6, no. 4, pp. 267-278, Aug. 1984.
- [14] M. Vetterli, "Fast 2-D discrete Cosine Transform," IEEE ICASSP Proc., pp. 1538-1541, March. 1985.
- [15] P. Duhamel and C. Guillemot, "Polynomial Transform computation of the 2-D DCT," IEEE ICASSP Proc., pp. 1515-1518, March. 1990.

- [16] G. Peceli, "A Common Structure for Recursive Discrete Transforms," IEEE Trans. on Circuits and Systems, vol. 33, No. 10, pp. 1035-10386, Oct. 1986.
- [17] K. Rose, A. Heiman, and I. Dinstein, "DCT/DST Alternate-Transform Image Coding," IEEE Trans. on Communication, vol.38, No. 1, pp. 94-101, Jan. 1990.
- [18] B. Silikstrom, *et al.*, "A high speed 2-D Discrete Cosine-Transform," Integration, VLSI journal 5, pp. 159-163, 1987.
- [19] M. T. Sun, T. C. Chen, and A. M. Gottlieb, "VLSI implementation of a  $16 \times 16$  Discrete Cosine Transform," IEEE Trans. on Circuits and Systems, vol. 36, No. 4, pp. 610-617, Apr. 1989.
- [20] U. Totzek, F. Matthiesen, S. Wohlleben, and T. G. Noll, "CMOS VLSI Implementation of the 2D-DCT with Linear Processor Array," IEEE ICASSP Proc., pp. 937-940, May, 1990.
- [21] S. A. White, "High-Speed Distributed-Arithmetic Realization of a Second-Order Normal-Form Digital Filter," IEEE Trans. on Circuits and Systems, vol. 33, No. 10, pp. 1036-1038, Oct. 1986.
- [22] S. A. White, "Applications of Distributed-Arithmetic to Digital Signal Processing: A Tutorial Review," IEEE ASSP Magazine, pp. 4-19, July. 1989.
- [23] M. Barbero, S. Cucchi, H. Bailon, "A Flexible Architecture for a HDTV Codec based on DCT", Signal Processing of HDTV, II, L. Chiariglione, ed., pp. 587-594, Elsevier Science Publishers B.V., 1990.
- [24] M. Barbero, S. Cucchi, M. Stroppiana, "Coding Strategies based on DCT for the Transmission of HDTV", Signal Processing of HDTV, L. Chiariglione, ed., pp. 503-508, Elsevier Science Publishers B.V., 1988.
- [25] J. A. Bellisio and S. Chu, "Television coding for broadband ISDN," IEEE Globecom'86, vol. 2, pp. 894-890, Dec. 1-4, 1986, Houston Texas.
- [26] W. H. Chen, and W. K. Pratt, "Scene Adaptive Coder," IEEE Trans. on Communications, Vol. Com-32, No.3, pp. 225-232, Mar. 1984.
- [27] S. Cucchi, and F. Molo, "DCT-based Television Codec for DS3 digital Transmission," SMPTE Journal, pp. 640-646, Sep. 1989.
- [28] S. S. Dixit and J. B. Nardone, "A Variable Bit Rate Layered DCT Video Coder for Packet Switched (ATM) Networks," IEEE ICASSP Proc., pp. 2253-2256, May, 1990.
- [29] D. L. Gall, H. Gaggioni, and C. T. Chen, "Transmission of HDTV signals under 140Mbps/s using a subband decomposition and discrete cosine transform coding," Proc. 2nd Int. Workshop on signal processing on HDTV, 29 Feb-2 Mar. 1988, L'Aquila Italy.
- [30] R. K. Jorgen, "The Challenges of digital HDTV," IEEE Spectrum, pp. 28-30, 71-73, Apr. 1991.



- [31] K. Kinoshita, T. Nakahashi, and . Eto, "130 M bit/s (H4 rate) HDTV Codec based on the DCT algorithms," *Electronics Letters*, Vol. 26, No. 16, pp. 1245-1246, 2nd Aug. 1990.
- [32] R. L. Nickelson, "The evolution of HDTV in the work of the CCIR," *IEEE Trans. Broadcasting*, vol. 35, No. 3, pp. 250-258, Sep. 1989.
- [33] W. Paik, "Digicipher-All digital, Channel, Compatible, HDTV Broadcast System," *IEEE Trans. on Broadcasting*, Vol. 36, No., 4, pp. 245-254, Dec., 1990.
- [34] J. Suzuki, M. Nomura, and S. Ono, "Comparative Study of transform coding for Super High Definition Images," *IEEE ICASSP Proc.*, pp. 2257-2260, May, 1990.
- [35] K. H. Tzou, T. C. Chen, P. E. Fleischer, and M. L. Liou, "Compatible HDTV Coding for Broadband ISDN," *Proc. Globecom '88*, pp. 743-749, NOV. 1988.
- [36] Y. Yashima, and K. Sawada, "100 Mbit/s HDTV Transmission using a High Efficiency Codec," *Signal Processing of HDTV, II*, L. Chiariglione, ed., pp. 587-594, Elsevier Science Publishers B.V., 1990.

**Table 1** Comparisons of different 2-D DCT algorithms.

**Table 2** Comparisons of the number of multipliers.

**Table 3** Comparisons of the number of adders.

**Fig.1** The 2-D successive data frame.

**Fig.2** The lattice module of lattice array II.

**Fig.3** The lattice module of lattice array I.

**Fig.4** The pre-matrix moving frame 2-D DCT architecture.

**Fig.5** The circular shift matrix ( $CSM$ ) I and II.

**Fig.6** The circular shift array (CSA).

**Fig.7** The Shift Register Array .

**Fig.8** The post-matrix moving frame 2-D DCT architecture.

**Fig.9** The block 2-D DCT architecture.

**Fig.10** The realization of the lattice module under distributed arithmetic using one ROM.

**Fig.11** The realization of the lattice module under distributed arithmetic using three ROM.

**Fig.12** System diagram of the DCT based HDTV coder.

**Fig.13** The block diagram of the DCT encoder.

**Fig.14** Block construction of a Video frame and proposed scanning pattern.

	row-column method based on Chen in[2]	Duhamel[15] <i>et. al.</i>	Cho-Lee [10]	Ma [7]	Liu-Chiu2D
No. of multipliers	$2N^2 \ln(N)$ $-6N^2/2 + 8N$	$N^2$ $+2N + 2$	$N^2$ $+2N + 2$	$4N(N + 1)$	$8N$
Throughput	$N+$ transposition	$2N$	$N$	$2N + 1$	$N$
Limitation on transform size $N$	power of 2	power of 2	power of 2	no	no
Communication	global	global	global	local	local
I/O operation	<i>PIPO</i>	<i>PIPO</i>	<i>PIPO</i>	<i>SIPO</i>	<i>SIPO</i>
Approach of of algorithm	indirect	direct	direct	indirect	direct

Table 1: Comparisons of different 2-D DCT algorithms.

NO	row-column method based on Chen in[2]	Duhamel[15] <i>et. al.</i>	Cho-Lee [10]	Ma [7]	Liu-Chiu2D
8	256	96	96	288	64
16	1408	512	512	1088	128
32	7424	2560	2560	4224	256
64	37376	12288	12288	16640	512

Table 2: Comparisons of the number of multipliers.

No	row-column method based on Chen in[2]	Duhamel[15] <i>et. al.</i>	Cho-Lee [10]	Ma [7]	Liu-Chiu2D
8	416	484	466	432	78
16	2368	2531	2530	1632	158
32	12416	12578	12738	6336	318
64	61696	60578	42461	24960	638

Table 3: Comparisons of the number of adders.

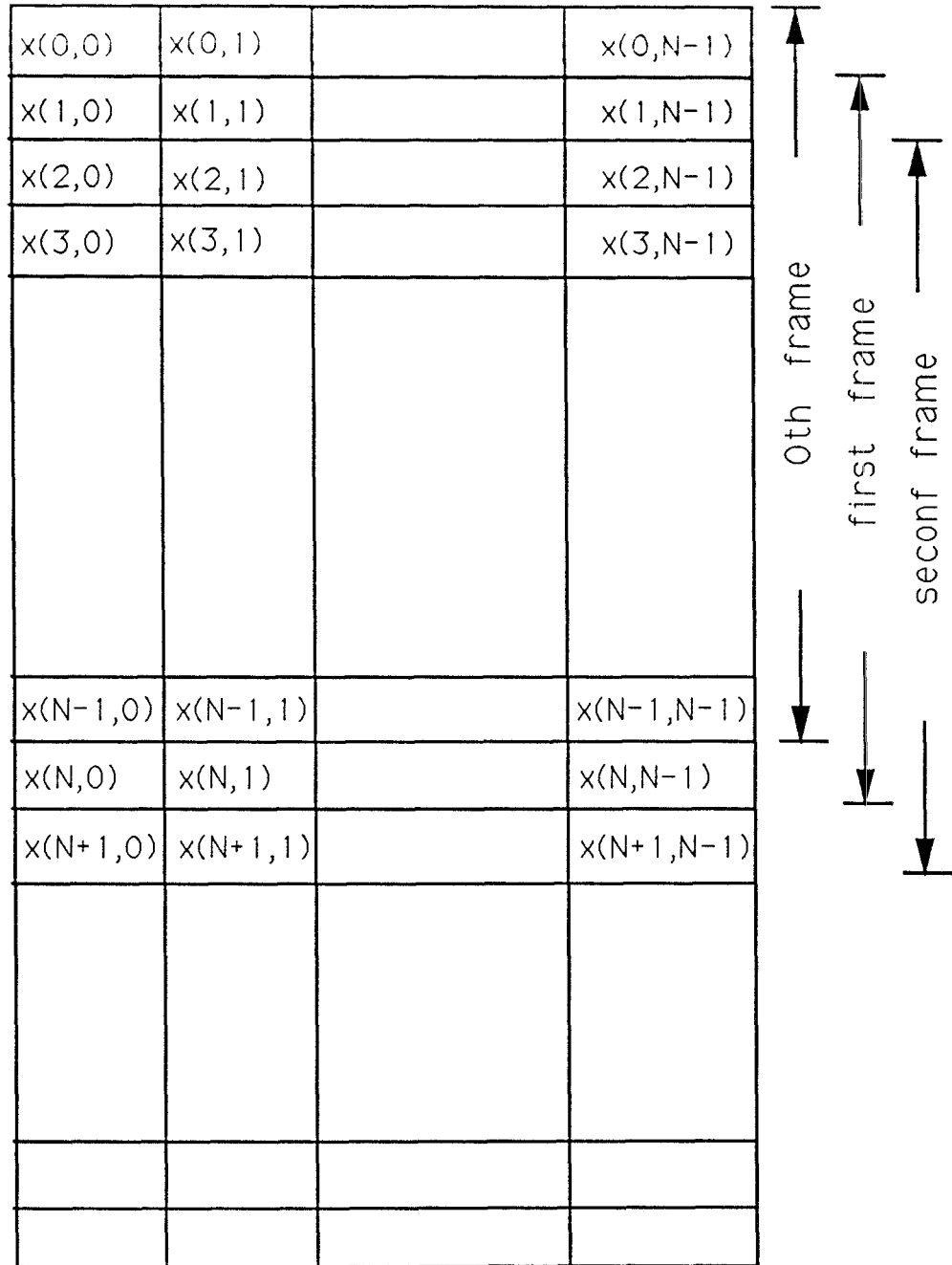


Figure 1: The 2-D successive data frame.

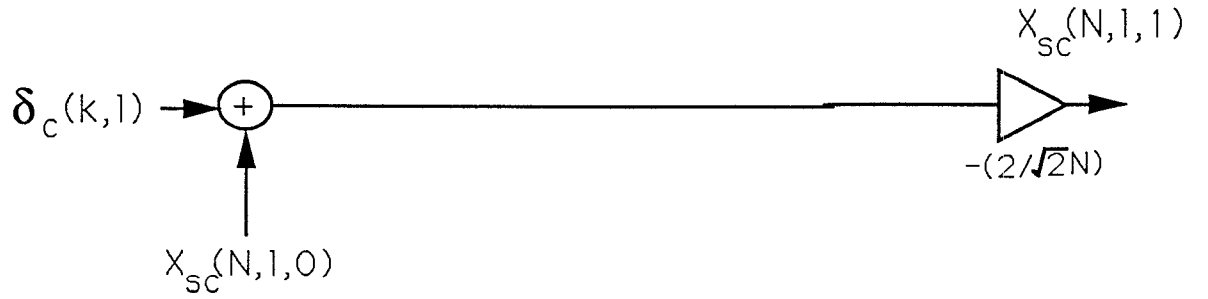
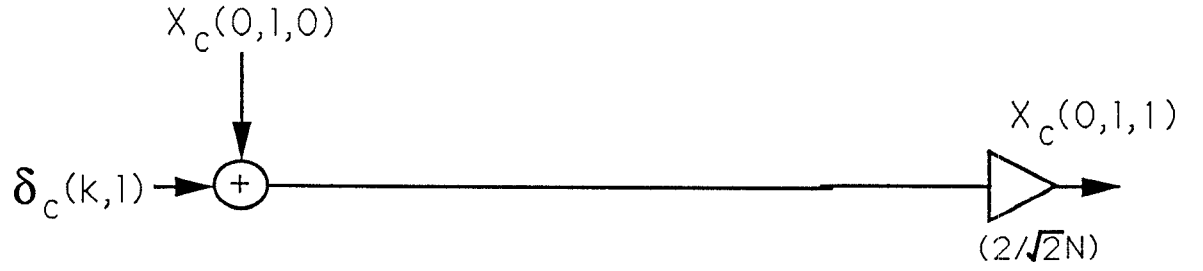
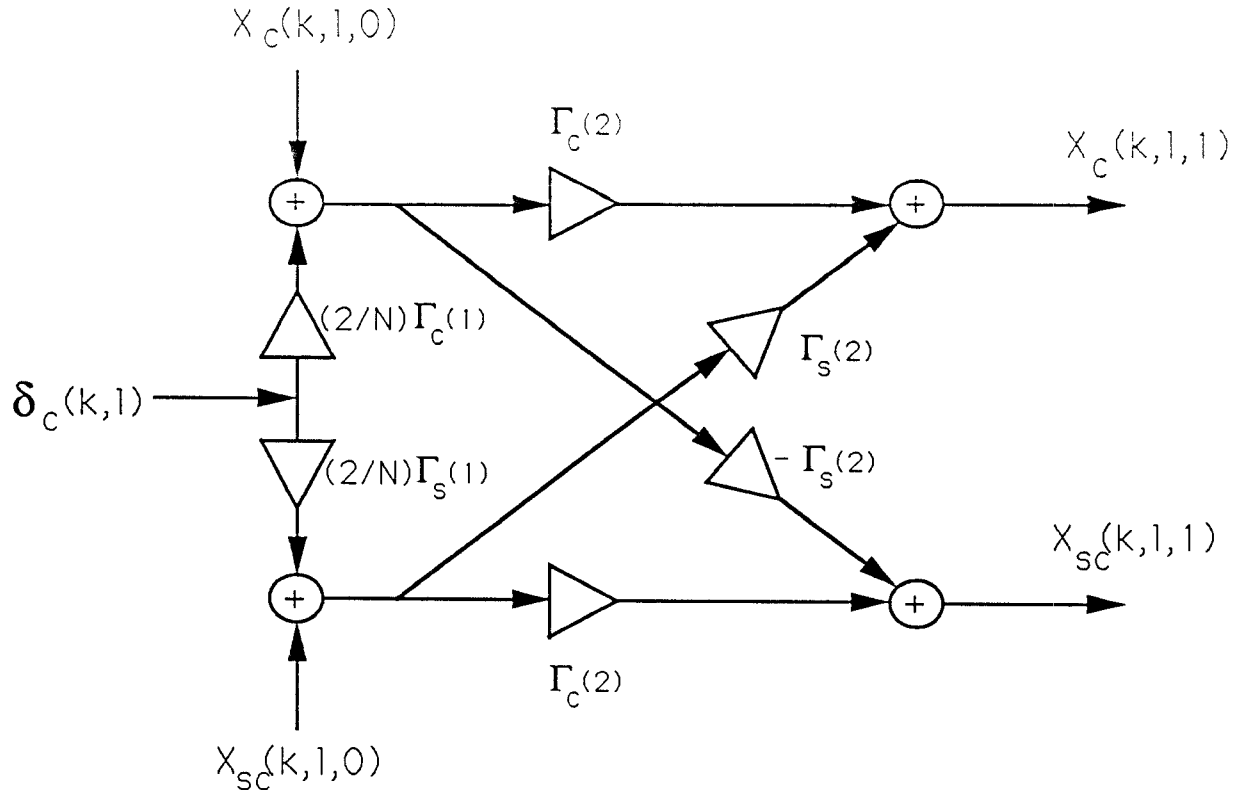


Figure 2: The lattice module of lattice array II.

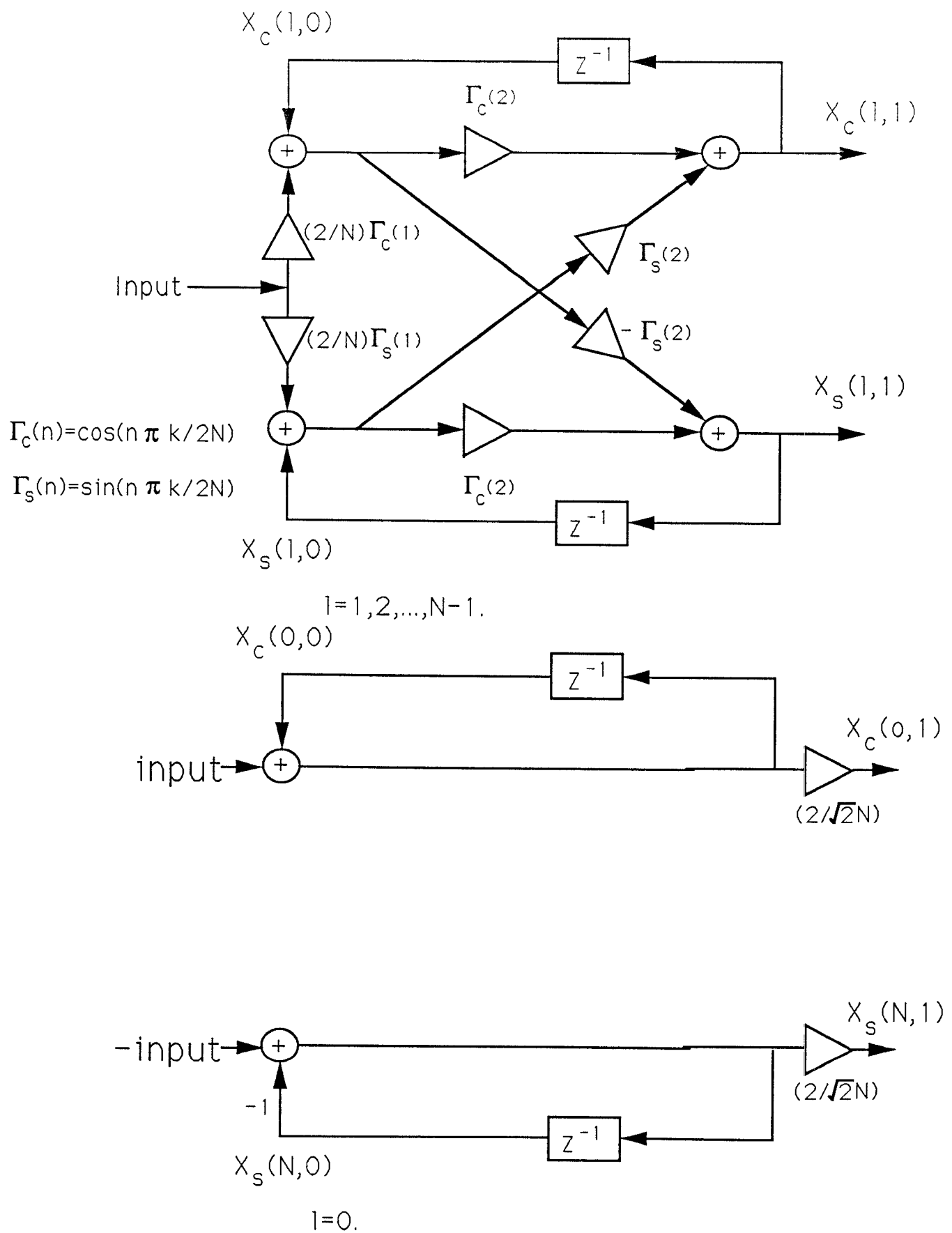


Figure 3: The lattice module of lattice array I.

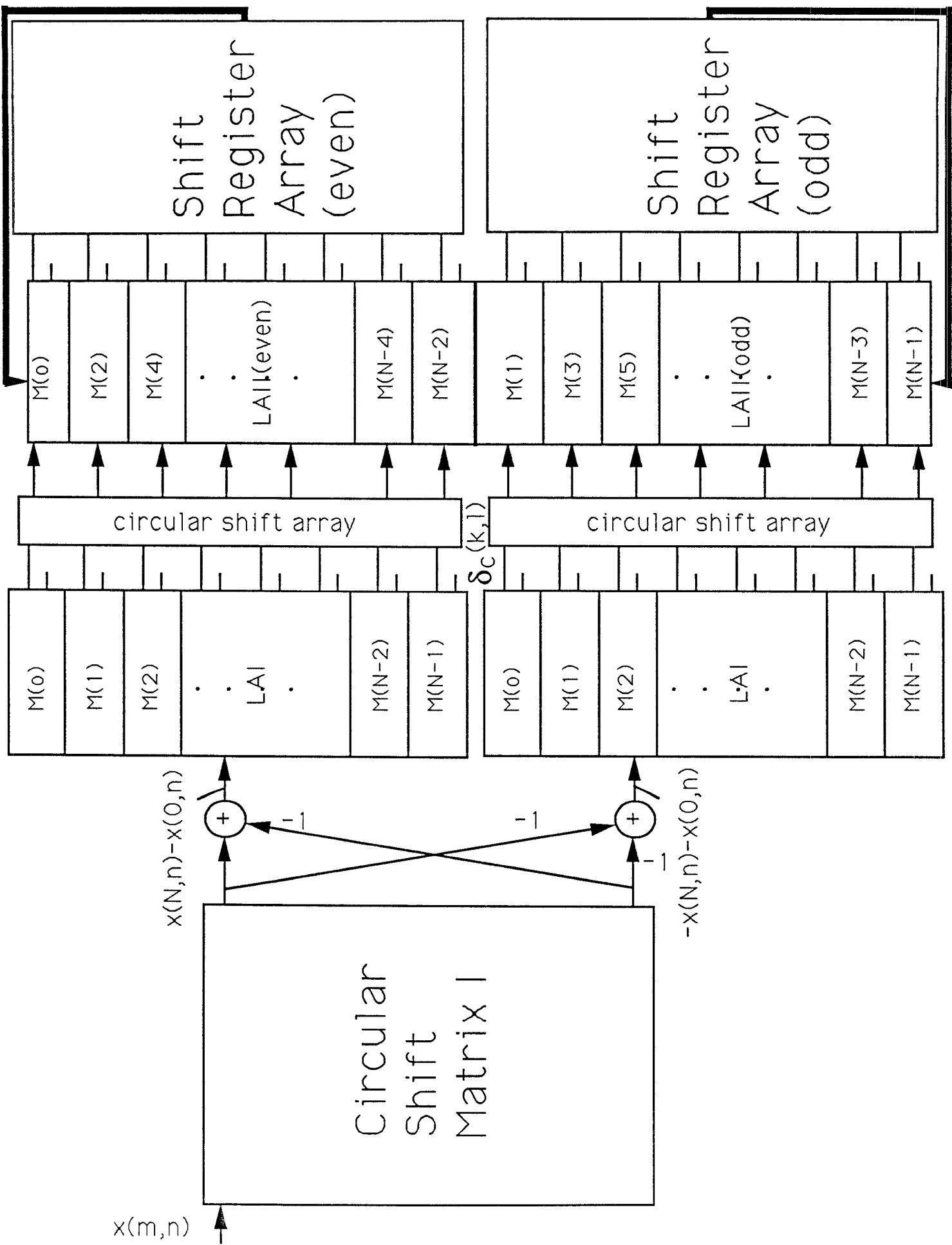
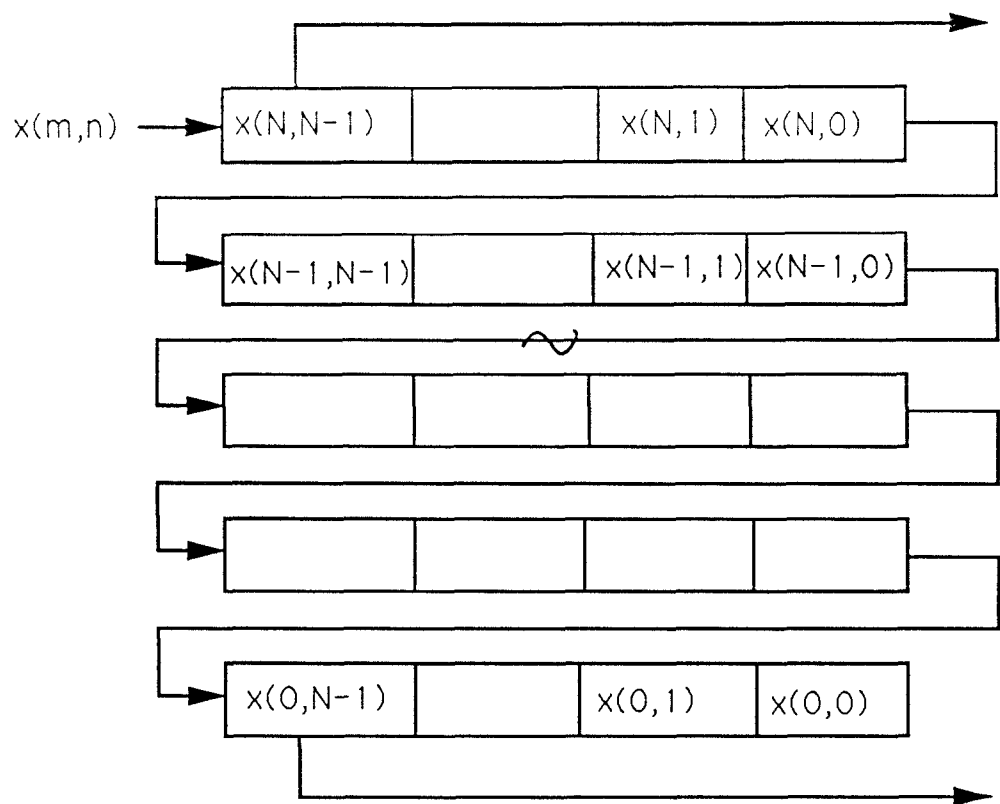
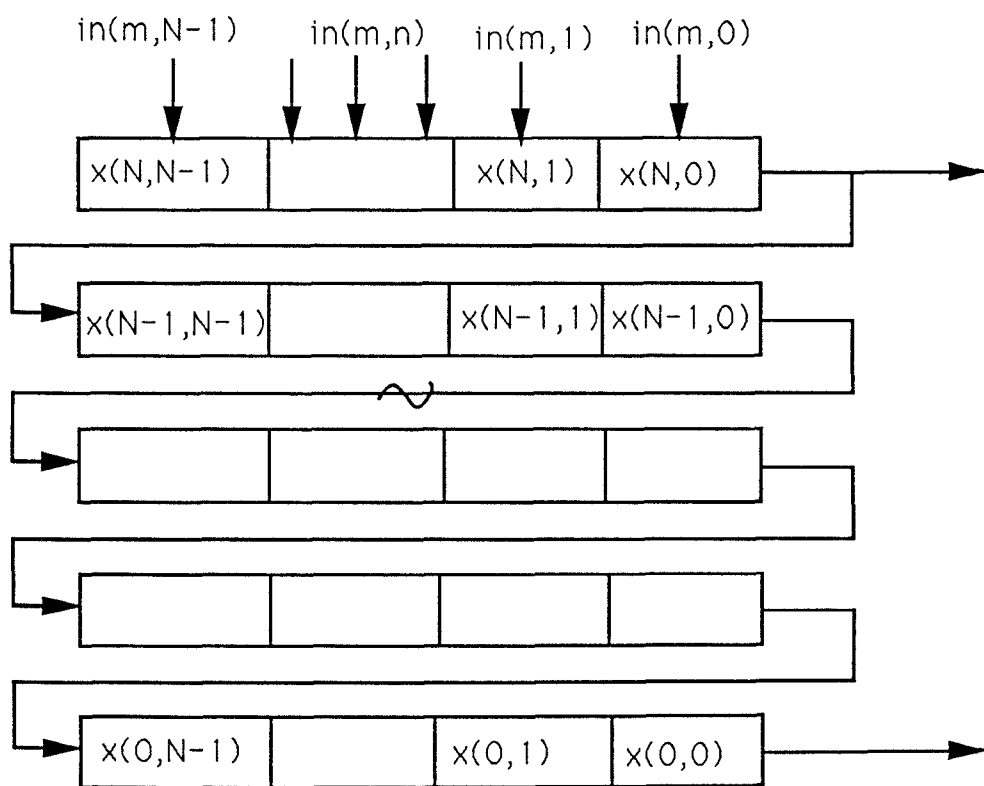


Figure 4: The pre-matrix moving frame 2-D DCT architecture.



Circular Shift Matrix I



Circular Shift Matrix II

Figure 5: The circular shift matrix (*CSM*) I and II.



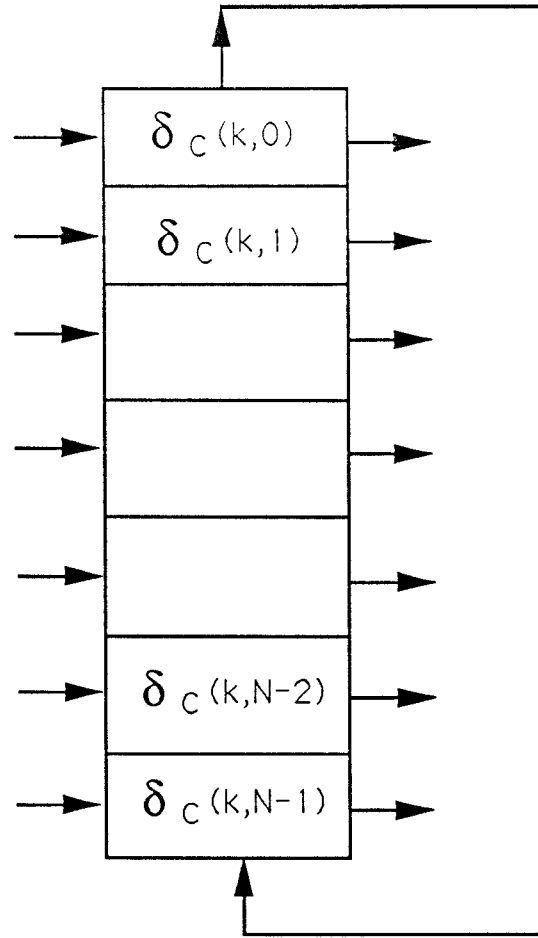


Figure 6: The circular shift array (CSA).

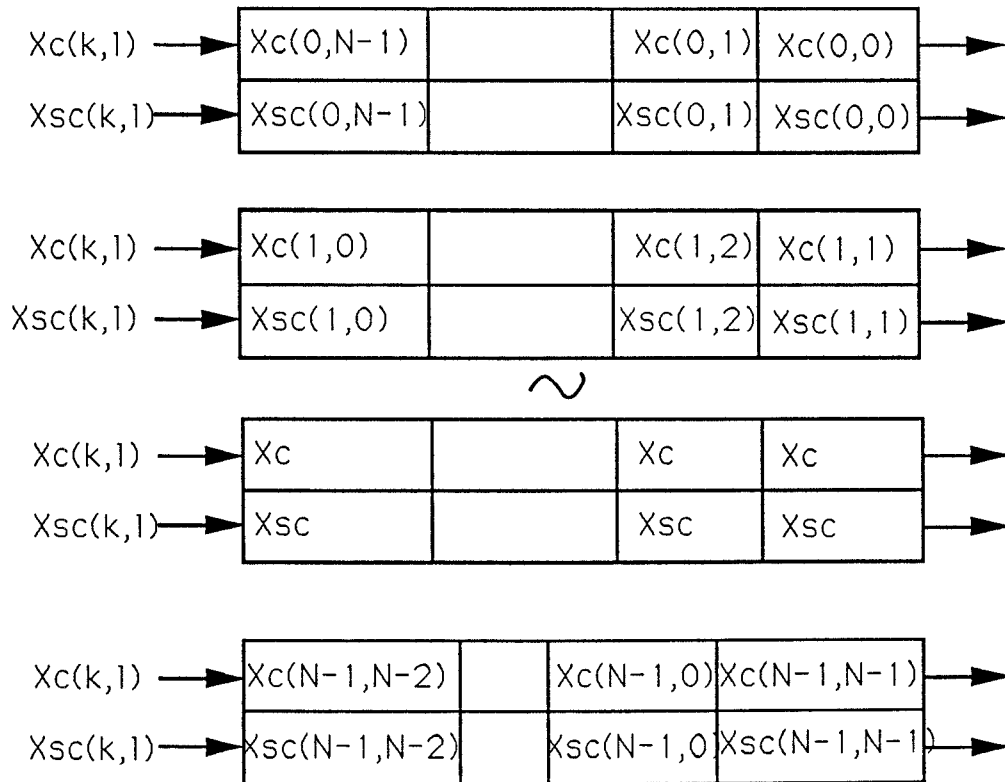


Figure 7: The Shift Register Array .

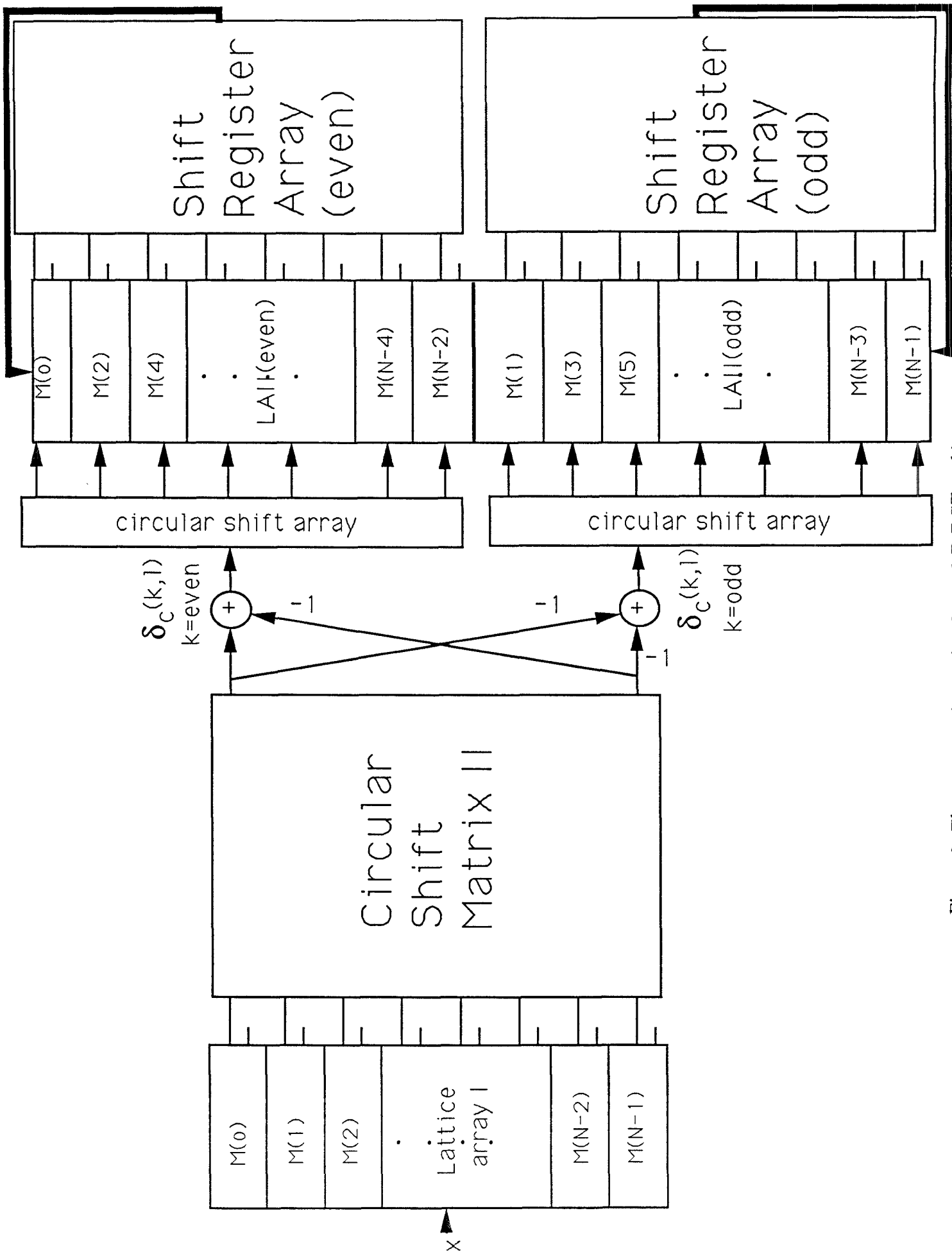


Figure 8: The post-matrix moving frame 2-D DCT architecture.

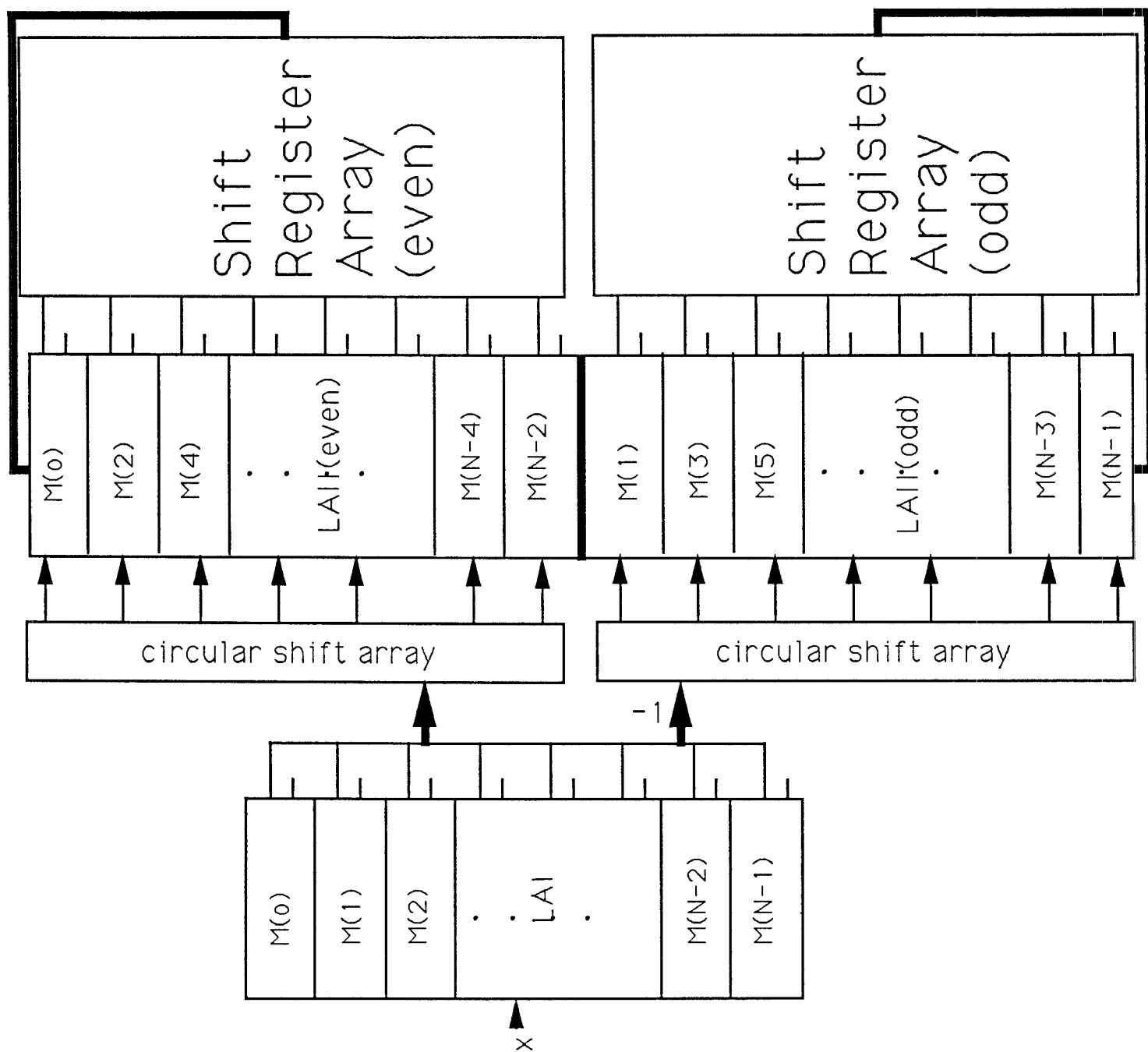


Figure 9: The block 2-D DCT architecture.

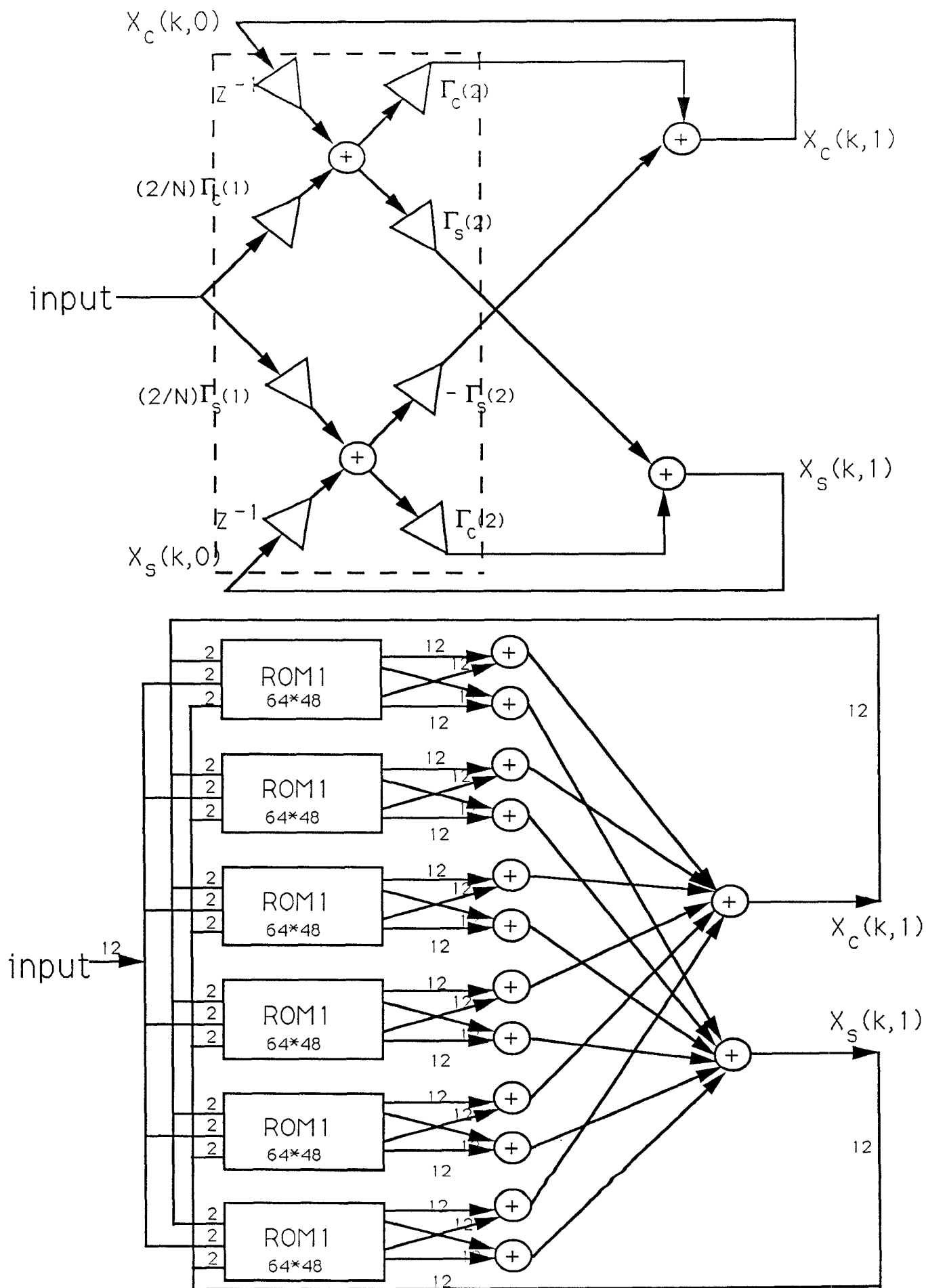


Figure 10: The realization of the lattice module under distributed arithmetic using one ROM.

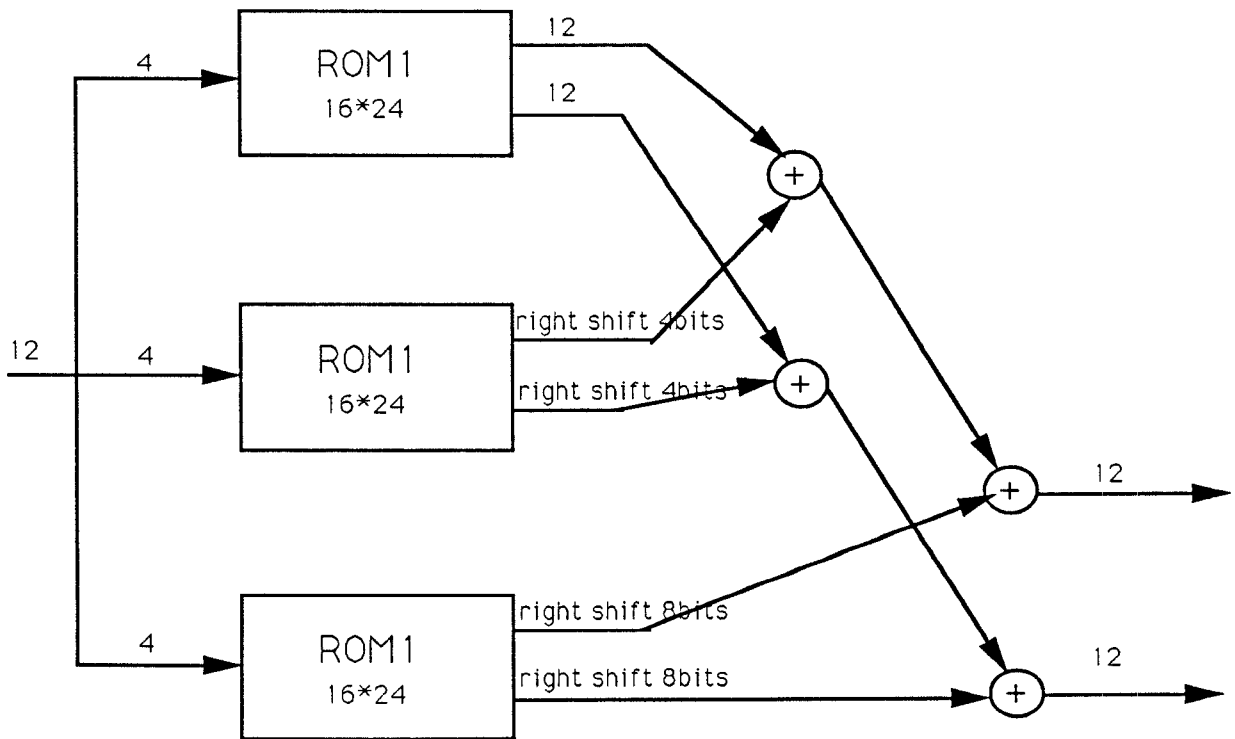
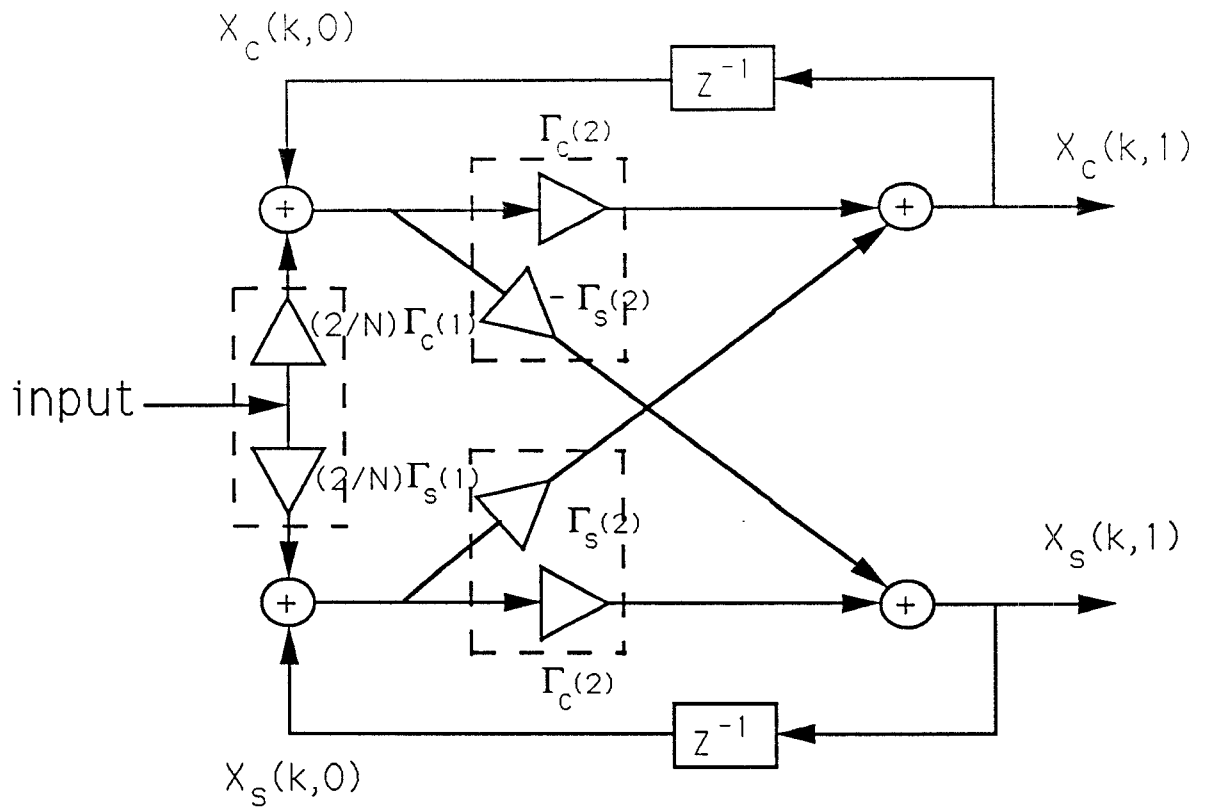


Figure 11: The realization of the lattice module under distributed arithmetic using three ROM.

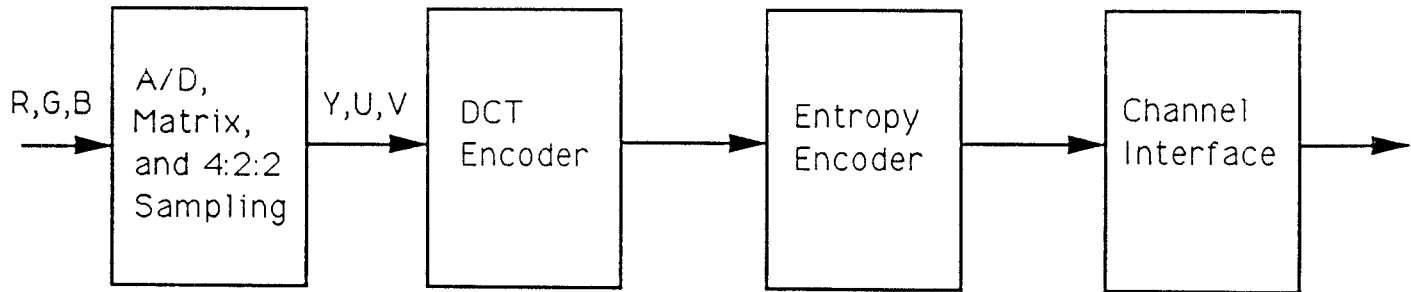


Figure 12: System diagram of the DCT based HDTV coder.

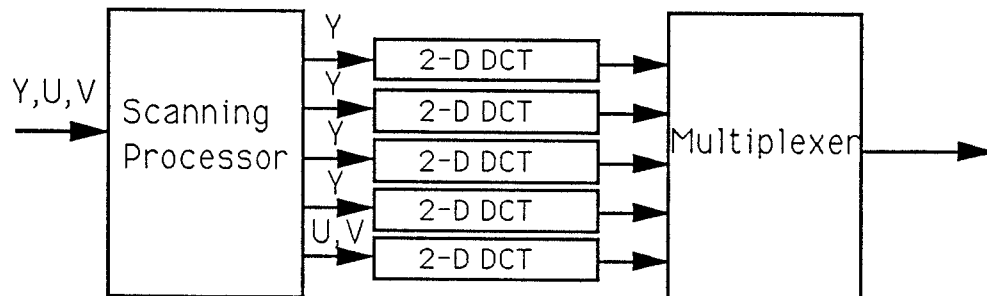


Figure 13: The block diagram of the DCT encoder.

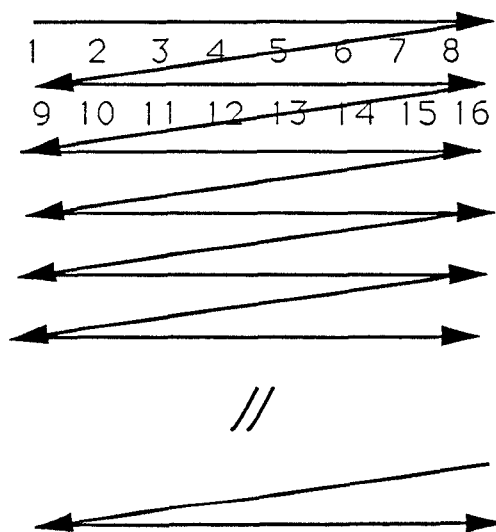
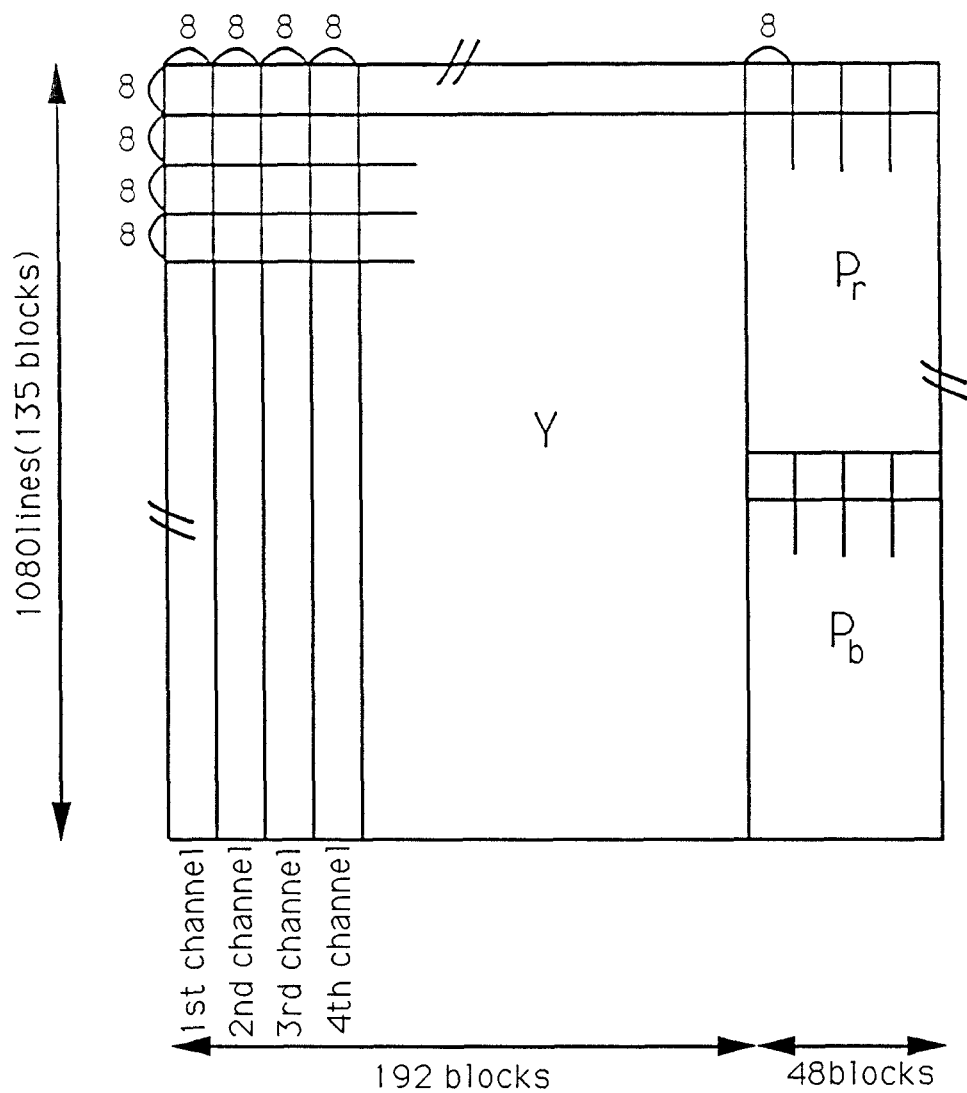


Figure 14: Block construction of a Video frame and proposed scanning pattern.