Abstract—This paper provides a direct derivation of the prime-factor-decomposed computation algorithm of an N-point discrete cosine transform for the number N decomposable into two relative prime numbers. It also presents input and output index mappings in the form of tables—namely, \( n_0 \), \( n_r \), \( n_c \), \( n_{cr} \), and \( k \)-tables. The index mapping tables are useful for practical use of the prime-factor-decomposed computation of arbitrarily sized discrete cosine transforms.

I. INTRODUCTION

Since its first introduction in 1974 [1], the discrete cosine transform (DCT) has found applications in speech and image signal processing [1]-[8] as well as in telecommunication signal processing [9], [10]. The DCT has been applied for speech and image compression because its performance was nearly optimal, yet not being signal dependant. On the other hand, the DCT has been utilized for realizing filter banks in FDM-TDM transmultiplexers because its real computation was simpler and faster than the complex computation of the discrete Fourier transform (DFT).

Along with the expanded applications of the DCT, a number of fast computation techniques have been also introduced [11]-[19]. Depending on the number of points \( N \), the computation techniques can be divided into two categories: one on the general composite number cases, and the other one on the prime-factor cases. For the former case, \( N \) of special interest is of \( 2^n \) type; for the latter case, \( N \) is factorizable into two mutually relative prime numbers \( N_1 \) and \( N_2 \). A recent work reports that the number of real multiplications for the power-of-two case can reduce to \( (N/2) \log N \), and its structure resembles that of the fast Fourier transform (FFT) [14]. For the prime-factor case, the number of multiplications reduces to \( N(N_1 + N_2) \) in its most primitive form, and its structure is similar to that of the prime-factor algorithm of the DFT [19].

The prime-factor-decomposed computation of the DCT was proven to be powerful in reducing the computational work and time, still providing a simple and nice structure. However, this technique has not yet been widely utilized mainly because its input and output index mappings are seemingly too involved. In fact, the mappings are the only barrier to overcome in applying the prime-factor algorithm. This paper is therefore intended to provide a simple and organized method to perform the index mappings.

In this paper, a formal direct derivation of the prime-factor-decomposed computation algorithm will be presented first. The derivation is a direct one in the sense that it is based on the real cosine function without resorting to the DFT expressions or the complex functions. Then, based on the equations obtained during the derivation, input and output index mappings will be introduced in the form of tables. This tabulation will enable us to implement any prime-factor-decomposable DCT in a straightforward manner. Finally, the index mapping tables will be demonstrated through the 12-point DCT.

II. DIRECT DERIVATION OF PRIME FACTOR DECOMPOSITION

Let \( x(k), k = 0, 1, \cdots, N - 1 \), be a time-domain sequence and \( X(n), n = 0, 1, \cdots, N - 1 \), be its transform-domain data sequence. Then, by definition, the DCT and the inverse DCT (IDCT), respectively, have the expressions

\[
X(n) = \frac{2}{N} e(n) \sum_{k=0}^{N-1} x(k) \cos \left( \frac{\pi}{N} (2k + 1) n \right),
\]

\[
n = 0, 1, \cdots, N - 1,
\]

\[
x(k) = \sum_{n=0}^{N-1} e(n) X(n) \cos \left( \frac{\pi}{N} (2k + 1) n \right),
\]

\[
k = 0, 1, \cdots, N - 1,
\]

where

\[
e(n) = \begin{cases} 1/\sqrt{2}, & \text{if } n = 0, \\ 1, & \text{otherwise.} \end{cases}
\]

Since (1) can be realized simply by transposing the flowgraph for (2), and since the term \( e(n) \) means nothing but a slight modification of the data \( X(n) \), it is sufficient...
Throughout this paper we will assume that our discussion to consider the IDCT-like equation

\[ x(k) = \sum_{n=0}^{N-1} X(n) \cos \left( \frac{\pi (2k+1) n}{2N} \right) \]

for all \( n_1 \in \mathcal{A}_1 \) and all \( n_2 \in \mathcal{A}_2 \). We denote by \( \mathcal{A} \) and \( \mathcal{A} \) sets of \( N \) integers such that

\[ \mathcal{A} = \{ n | n = f(n_1, n_2), \, n_1 \in \mathcal{A}_1, \, n_2 \in \mathcal{A}_2 \} \]

\[ \mathcal{B} = \{ n | n = f(n_1, n_2), \, n_1 \in \mathcal{A}_1, \, n_2 \in \mathcal{A}_2 \} \]

Then it can be shown that the \( 2N \) integers in the collection of \( \mathcal{A} \) and \( \mathcal{B} \) are identical to the \( 2N \) integers in the collection of \( \mathcal{A} \) and \( \mathcal{B} \). This implies that a summation over \( N \) indexes in \( \mathcal{A} \) can split into two terms—a summation over the \( N \) indexes in \( \mathcal{A} \) and a summation over \( N \) indexes in \( \mathcal{B} \). Therefore, we can rewrite (4) as follows:

\[ x(k) = \frac{1}{2} \sum_{n=0}^{N-1} X(n) \cos \left( \frac{\pi (2k+1) n}{2N} \right) \]

\[ + \frac{1}{2} \sum_{n=0}^{N-1} X(n) \cos \left( \frac{\pi (2k+1) n}{2N} \right). \]

We denote, for all \((n_1, n_2)\) in \( \mathcal{A}_1 \times \mathcal{A}_2 \),

\[ \hat{X}(n_1, n_2) = s(n) X(n) |_{n-f(n_1, n_2)}, \]

\[ \tilde{X}(n_1, n_2) = X(n) |_{n-f(n_1, n_2)}, \]

where

\[ s(n) = \begin{cases} 1, & \text{if } n_1N_1 + n_2N_2 < N, \\ -1, & \text{otherwise.} \end{cases} \]

Then (10) can be rewritten as

\[ x(k) = \frac{1}{2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \{ \hat{X}(n_1, n_2) \]

\[ \cdot \cos \left( \frac{\pi (2k+1)(n_1N_2 + n_2N_1)}{2N} \right) \]

\[ + \tilde{X}(n_1, n_2) \cos \left( \frac{\pi (2k+1)}{2N} \right) \]

\[ \cdot (n_1N_2 - n_2N_1)/2N]. \]

The term \( s(n) \) reflects the negative sign appearing in the relation

\[ \cos \left( \frac{\pi (2k+1)(2N-n)}{2N} \right) = -\cos \left( \frac{\pi (2k+1) n}{2N} \right). \]

We do not need such a term for the second part of (10), since \( n_1N_1 + n_2N_2 < N \) for all \((n_1, n_2)\) in \( \mathcal{A}_1 \times \mathcal{A}_2 \). We now define \( \hat{X}(n_1, n_2) \) such that

\[ X(n_1, n_2) = \begin{cases} \hat{X}(n_1, n_2) = \hat{X}(n_1, n_2), & \text{if } n_1 = 0 \text{ or } n_2 = 0, \\ \hat{X}(n_1, n_2) + \tilde{X}(n_1, n_2), & \text{otherwise.} \end{cases} \]

A proof of this is given in Appendix A. The term collection here indicates the set obtained by listing all the elements in two sets. See footnote 11.
Then, since
\[ \sum_{n_2=1}^{N_2-1} \sum_{n_1=1}^{N_1-1} \hat{X}(n_1, n_2) \cos \left[ \pi (2k + 1) \right] = 0, \] (16a)
\[ \sum_{n_2=1}^{N_2-1} \sum_{n_1=1}^{N_1-1} \hat{X}(n_1, n_2) \cos \left[ \pi (2k + 1) \right] = 0, \] (16b)
equation (13) can be written in the form
\[ x(k) = 1/2 \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} X(n_1, n_2) \cos \left[ \pi (2k + 1) n_1/2N_1 \right] \]
\[ \cdot \left\{ \cos \left[ \pi (2k + 1) n_2/2N_2 \right] + \cos \left[ \pi (2k + 1) (n_1 N_2 - n_2 N_1)/2N \right] \right\} \]. \quad (17)

Recalling the relation
\[ 1/2 \left\{ \cos \left[ \pi (2k + 1) (n_1 N_2 + n_2 N_1)/2N \right] + \cos \left[ \pi (2k + 1) (n_1 N_2 - n_2 N_1)/2N \right] \right\} = \cos \left[ \pi (2k + 1) n_1 N_2/2N \right] \]
\[ \cdot \cos \left[ \pi (2k + 1) n_2/2N_2 \right]. \quad (18) \]
we can rewrite it again as
\[ x(k) = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} X(n_1, n_2) \cos \left[ \pi (2k + 1) n_1/2N_1 \right] \]
\[ \cdot \cos \left[ \pi (2k + 1) n_2/2N_2 \right]. \quad (19) \]

Now, we consider the output mapping which connects \( x(k) \) to \( x(k_1, k_2) \). We denote by \( k_1 \) and \( k_2 \), respectively, \( \hat{k}_1 = k \mod 2N_1, \hat{k}_2 = k \mod 2N_2 \). We define \( g \) to be a mapping from \( \mathfrak{X} \) to \( \mathfrak{X}_1 \times \mathfrak{X}_2 \) such that \( (k_1, k_2) = g(k) \) with
\[ k_1 = \begin{cases} \hat{k}_1, & \text{if } \hat{k}_1 < N_1, \\ 2N_1 - 1 - \hat{k}_1, & \text{otherwise}, \end{cases} \quad (20a) \]
\[ k_2 = \begin{cases} \hat{k}_2, & \text{if } \hat{k}_2 < N_2, \\ 2N_2 - 1 - \hat{k}_2, & \text{otherwise}, \end{cases} \quad (20b) \]
for each \( (k_1, k_2) \) in \( \mathfrak{X}_1 \times \mathfrak{X}_2 \), and a \( k \) in \( \mathfrak{X} \). Then, it can be shown that \( g \) is a one-to-one mapping, so there exists an inverse mapping \( g \) from \( \mathfrak{X}_1 \times \mathfrak{X}_2 \) to \( \mathfrak{X} \). Thus, we can define
\[ x(k_1, k_2) = x(k) \big|_{k=g^{-1}(k_1,k_2)}. \quad (21) \]

If we apply this to (19), we finally obtain the expression in (6), since
\[ \cos \left[ \pi (2k + 1) n_1/2N_1 \right] = \cos \left[ \pi (2k + 1) n_1/2N_1 \right]. \quad (22a) \]
\[ \cos \left[ \pi (2k + 1) n_2/2N_2 \right] = \cos \left[ \pi (2k + 1) n_2/2N_2 \right]. \quad (22b) \]

Therefore, we have shown that (4) and (6) are identical if \( X(n) \) and \( X(n_1, n_2) \) are connected through (7), (11), (15), and (17), and if \( x(k) \) and \( x(k_1, k_2) \) are connected through (20) and (21). The former three equations form the input mapping; and the latter two equations form the output mapping. Thus, we can now perform an N-point IDCT by cascading \( N_2 N_1 \)-point IDCT's and \( N_1 N_2 \)-point IDCT's, as is demonstrated in Fig. 1 for the numbers \( N = 12, N_1 = 3, N_2 = 4 \).

### III. TABULATION OF INDEX MAPPINGS

For any given \( N \), the main body performing \( N_2 N_1 \)-point IDCT's and \( N_1 N_2 \)-point IDCT's is quite straightforward to implement, as is illustrated in Fig. 1. But more care should be taken on the input mapping which converts \( X(n) \), \( n = 0, 1, \ldots, N - 1 \), to \( X(n_1, n_2), n_1 = 0, 1, \ldots, N_1 - 1, n_2 = 0, 1, \ldots, N_2 - 1 \), and the output mapping which converts \( x(k_1, k_2), k_1 = 0, 1, \ldots, N_1 - 1, k_2 = 0, 1, \ldots, N_2 - 1 \), to \( x(k), k = 0, 1, \ldots, N - 1 \). In this section, we will tabulate the index mappings relating \( n \) to \( (n_1, n_2) \) and relating \( (k_1, k_2) \) to \( k \), and will discuss how to utilize the resulting tables to realize the above input and output mappings.

We first consider the input mapping, which is represented by (7), (11), and (15). We set up a table with \( N_1 \) rows and \( N_2 \) columns, naming each row 0 through \( N_1 - 1 \) and each column 0 through \( N_2 - 1 \). Then we fill location \( (n_1, n_2) \), which is at row \( n_1 \) and column \( n_2 \), with \( f(n_1, n_2) \) for \( n_1 = 0, 1, \ldots, N_1 - 1, n_2 = 0, 1, \ldots, N_2 - 1 \), putting a negative sign to every \( f(n_1, n_2) \) meeting \( n_1 N_2 + n_2 N_1 \geq N \). The negative sign is meant to reflect the \( s(n) \) term in (12), designating that the input data \( X(f(n_1, n_2)) \) corresponding to the index should accompany a negative sign as in (11a) indicates. We name the resulting table \( \hat{n} \)-table. Then the \( \hat{n} \)-table represents (7a) and (11a). We define \( \hat{n} \)-table in a similar manner by listing \( f(n_1, n_2) \) to location \( (n_1, n_2), n_1 = 0, 1, \ldots, N_1 - 1, n_2 = 0, 1, \ldots, N_2 - 1 \). Then the \( \hat{n} \)-table will represent (7b) and (11b). Note that there is no negative sign in the \( \hat{n} \)-table.

If we denote by \( \hat{n}(n_1, n_2) \) and \( \hat{n}(n_1, n_2) \) the entries at \( (n_1, n_2) \) of the \( \hat{n} \)- and \( \hat{n} \)-tables, respectively, then we have \( \hat{n}(n_1, n_2) = \hat{n}(n_1, n_2) \) for \( n_1 = 0 \) or \( n_2 = 0 \), and \( \hat{n}(n_1, n_2) = \hat{n}(N_1 - n_1, N_2 - n_2) \) for the other \( n_1 \) and \( n_2 \). This reflects (8a) and (8b), implying that neither the entries in the \( \hat{n} \)-table nor the entries in the \( \hat{n} \)-table cover the set \( \mathfrak{X} \). So we introduce new tables which can cover the set \( \mathfrak{X} \). We define \( n_2 \)-table to be the table obtained from the \( \hat{n} \)- and \( \hat{n} \)-tables in the following manner. If \( n_1 = 0 \) or \( n_2 = 0 \), we take \( n_2(n_1, n_2) = \hat{n}(n_1, n_2) \), where \( n_2(n_1, n_2) \) denotes the entry at \( (n_1, n_2) \) of the \( n_2 \)-table. Otherwise, we take \( n_2(n_1, n_2) = \hat{n}(n_1, n_2) \) for the first \( (N_1 - 1) (N_2 - 1) \)/2

A proof of this is given in Appendix B.
entries taken columnwise, and \( n_c(n_1, n_2) = \hat{n}(n_1, n_2) \) for the others but without negative signs. Also, we define \( n_R \)-table in a similar manner, except that we take entries rowwise. Then, clearly, the entries in the \( n_R \)-table, as well as the entries in the \( n_c \)-table, cover the set rowwise. Then, clearly, the entries in the \( n_R \)-table and read out the input data \( X(n) \) columnwise, mapping by taking either \( X(1, 0), X(0, 1), \) \( X(1, 1), \) \( X(0, N_2 - 1), X(1, N_2 - 1), \) \( X(N_1 - 1, 0), X(N_1 - 1, 1), \) \( \cdots, X(N_1 - 1, N_2 - 1) \). Since this implies taking data rowwise, we now take the \( n_R \)-table and read out data \( X(n) \) rowwise. The other processes are the same as the previous case.

We need not consider both \( n_c \)- and \( n_R \)-tables for a given \( N \). If we want to perform the \( N_c \)-point IDCT operation first, followed by the \( N_R \)-point IDCT operation, then the \( n_c \)-table together with the \( \hat{n} \)- and \( \hat{n_R} \)-tables will suffice. But if we want to do \( N_R \)-point IDCT first, then we need the \( n_R \)-table instead of the \( n_c \)-table. Depending on the table we take for the input mapping, the output mapping also changes, as will be discussed below.

We now consider the output mapping, which is represented by (20) and (21). This case is rather simple compared to the previous case. Notice that this tabulation is possible due to the one-to-oneness of the mapping \( g \).

We make a table with \( N_1 \) rows and \( N_2 \) columns, naming each row 0 through \( N_1 - 1 \) and each column 0 through \( N_2 - 1 \) as before. We write \( k, k = 0, 1, \ldots, N - 1 \), to location \((k_1, k_2)\), where \( k_1 \) and \( k_2 \) follow (20a) and (20b), respectively. We name the resulting table \( k \)-table.

Using this table, we now consider realizing (21). We first consider the case when the \( N_1 \)-point IDCT's were performed first, followed by the \( N_2 \)-point IDCT's in the main body. For this case, the input data \( x(k_1, k_2) \) (to the output mapping box), which correspond to the output data of the \( N_2 \)-point IDCT's of the main body, are aligned in the order \( x(0, 0), x(0, 1), \ldots, x(0, N_2 - 1), x(1, 0), x(1, 1), \ldots, x(1, N_2 - 1), \ldots, x(N_1 - 1, 0), x(N_1 - 1, 1), \ldots, x(N_1 - 1, N_2 - 1) \). This implies that we have to take data rowwise. Thus, we read out the output data \( x(k) \) rowwise from the \( k \)-table, juxtaposing them to
the right-hand side of the corresponding $x(k_1, k_2)$. On these alignments, we now perform the operation corresponding to (21), which is nothing but a unity-gain line drawing between each $x(k_1, k_2)$ and the corresponding counterpart $x(k)$. For the other case, where we perform the $N_2$-point IDCT's first and the $N_1$-point IDCT's last, we can repeat a similar procedure. Since we now have the $N_1$-point IDCT's first and the $N_2$-point IDCT's last, we have to take the $nR$-table columnwise in the input mapping, taking the $k$-tuple rowwise in the output mapping. But if we want to have the reversed order, we have to take the $nC$-table columnwise in the input mapping, taking the $k$-tuple rowwise in the output mapping. For either case, the $n$ and $nR$-tables are used in common to provide butterfly connections.

IV. Example

We consider the 12-point IDCT/DCT to illustrate the decomposed computation and the corresponding $n$- and $k$-tables. We let $N_1 = 3$, $N_2 = 4$, and perform four 3-point IDCT’s first followed by three 4-point IDCT’s. Then we obtain the structure shown in Fig. 1. One can also show a similar structure for the case when three 4-point IDCT’s are taken first. We want to figure out the black boxes named input and output mappings in the figure.

We first draw up the $n R$, $n C$, and $n K$-tables for the input mapping box. By evaluating (7a) with $N_1 = 3$, $N_2 = 4$, $n_1 = 0, 1, 2$, and $n_2 = 0, 1, 2, 3$, we obtain the $n R$-table in Table I(a). Similarly, using (7b), we obtain the $n C$-table in Table I(b). We now derive the $n C$- and $n K$-tables from the above two tables. For the locations with $n_1 = 0$, or $n_2 = 0$ of both tables, we copy the corresponding entries in the $n R$- or $n C$-table. But for the other locations, we copy down the first three entries from the $n R$-table and the other three from the $n C$-table, since $(N_1 - 1) (N_2 - 1) / 2 = 3$. If we take the entries columnwise, then we obtain the $n C$-table; and if we take the entries rowwise, we obtain the $n K$-table. These two tables are shown in Table I(c) and I(d).

Now, we consider aligning the input data $X(n)$, $n = 0, 1, \ldots , 11$. Since we are taking four 3-point IDCT’s first in the main body, the output data of the input mapping box are aligned in the order of $X(0, 0)$, $X(1, 0)$, $X(2, 0)$; $X(0, 1)$, $X(1, 1)$, $X(2, 1)$; $X(0, 2)$, $X(1, 2)$, $X(2, 2)$; $X(0, 3)$, $X(1, 3)$, $X(2, 3)$. We can have the $n K$-table in aligning the input data $X(n)$ and read out the data columnwise. As a result, we have the order $X(0)$, $X(4)$, $X(8)$; $X(3)$, $X(1)$, $X(5)$; $x(6)$, $X(2)$, $X(10)$; $X(9)$, $X(11)$, $X(7)$.

The next step is to draw flowgraphs connecting those input and output data. For each $X(n_1, n_2)$ having $n_1 = 0$ or $n_2 = 0$, we draw a horizontal unity-gain line joining it with the corresponding $X(n)$. This achieves the first part of (15). The second part of it turns out to be butterfly operations which are realized by joining $X(n_1, n_2)$ and $X(3 - n_1, 4 - n_2)$ with the corresponding two $X(n)$’s identified through the $n R$ and $n K$-tables. For example, when $n_1 = 1$, $n_2 = 1$, we have 1 and 7 at location (1, 1) of the $n R$- and $n K$-table, respectively; and we have 1 and -7 at (3-1, 4-1) of the tables. Thus, we draw a butterfly joining $X(1, 1)$ and $X(2, 3)$ with $X(1)$ and $X(7)$. Each line in this butterfly has unity-gain except the line joining $X(7)$ with $X(2, 3)$ which has the gain -1, as is designated by the negative sign in -7. In a similar way, we can draw the second and the third butterfly operations. Due to the symmetric properties of the $n$- and $n R$-tables, the butterflies are dwindling with the crossing point of the first butterfly as their axes. All these are depicted in Fig. 2(a).

We now consider the output mapping. By evaluating (20a) and (20b) with $k = 0, 1, \ldots , 11$, we obtain the table in Table I(e). Drawing the flowgraph for the output mapping is quite straightforward. Since we took the $n K$-table for the output mapping, we have to take the $k$-table rowwise. Thus, we have the order $x(0, 0)$, $x(0, 1)$, $x(0, 2)$, $x(0, 3); x(1, 0)$, $x(1, 1)$, $x(1, 2)$, $x(1, 3); x(2, 0)$,
Fig. 3. (a) Input mapping and (b) output mapping when 4-point IDCT's come first, followed by 3-point IDCT's.

\[ x(2, 1), x(2, 2), x(2, 3) \]

for the input data of the output mapping. And by reading out the \(k\)-table rowwise, we obtain the sequence \(x(0), x(6), x(5), x(11), x(7), x(1), x(10), x(4), x(8), x(9), x(2), x(3)\) for the output data.

The remaining process is to realize (21) on these alignments, which is nothing but to join each \(x(k_1, k_2)\) and the corresponding \(x(k)\) with a unity-gain line. This is depicted in Fig. 2(b).

For the case where three 4-point IDCT's come first, followed by four 3-point IDCT's, one can show, by taking a similar procedure, that the resulting five tables, \(n_T-\) and \(n_R\)-tables are used for input signal alignment, \(n_T-\) and \(n_R\)-tables are used for output signal alignment and mapping. By employing these tables, we can find the necessary mappings in a straightforward manner. Furthermore, once the tables are generated, we can freely choose the DCT's to put first—either \(N_T\)-point or \(N_R\)-point. The index mapping tables are therefore expected to play a valuable role in practical applications of the prime-factor-decomposed computation of DCT.

APPENDIX A

Let the operation \(\mathcal{O}\)denote the collection of two sets, preserving the total number of elements, such that

\[ \mathcal{R} \cap \mathcal{O} = \{ f(n_1, n_2) \mid n_1 \in \mathcal{R}_1, n_1 \in \mathcal{R}_2 \}. \]

We want to prove in this appendix that

\[ \mathcal{R} \cap \mathcal{O} = \mathcal{R} \cap \mathcal{M}. \]
We define set $\mathcal{R}'$ to be

$$\mathcal{R}' = \{ n | n = n_1 N_1 + n_2 N_2, n_1 \in \mathcal{R}_1, n_2 \in \mathcal{R}_2 \},$$

and we first show that

$$\mathcal{R} \subset \mathcal{R}' \cup \tilde{\mathcal{R}}.$$  \hspace{1cm} (A4)

Since, by the SIR [23],

$\{ n | n = n_1 N_1 + n_2 N_1 \text{ modulo } N, n_1 \in \mathcal{R}_1, n_2 \in \mathcal{R}_2 \} = \mathcal{R}$,  \hspace{1cm} (A5)

it is sufficient to show that for each $n$ in $\mathcal{R}'$ with $n \geq N$, we can find $n \in \mathcal{R}$ with $n' \geq N$. We want to show that we can identify the element $n' = N \in \mathcal{R} \cap \mathcal{R}$. Let $n''$ denote the element of $\mathcal{R}$ which is at $(n_1, n_2) = (n_1, N_2 - n_2)$. Then

$$n'' = \left| n_1 N_2 - (N_2 - n_2) N_1 \right| = \left| n_1 N_2 + n_2 N_1 - N_1 N_2 \right| = \left| n' - N \right| = n' - N.$$  \hspace{1cm} (A6)

Thus, (A4) is verified.

Now, since $\mathcal{R}' \cap \mathcal{R} \subset \mathcal{R}$ and $\mathcal{R} \subset \mathcal{R}' \cup \tilde{\mathcal{R}}$, we have $\mathcal{R} \subset \mathcal{R}' \cup \tilde{\mathcal{R}}$. Therefore, to prove the relation in (22), it suffices to show that for each element in $\mathcal{R} \cup \tilde{\mathcal{R}}$, we can identify another identical element in the set. But if $n_1 = 0$ or $n_2 = 0$, $f(n_1, n_2) = f(n_1, n_2)$; otherwise, $f(n_1, n_2) = f(N_1 - n_1, n_2 - n_2)$ and $f(n_1, n_2) = f(n_1 - n_1, N_2 - n_2)$, by (8a) and (8b). Thus, the proof is complete.

**Appendix B**

In this appendix, we prove that $g$ is a one-to-one mapping from $\mathcal{R}$ to $\mathcal{R}_4 \times \mathcal{R}_4$. Let $\mathcal{R}_4$, $\mathcal{R}_B$, $\mathcal{R}_C$, and $\mathcal{R}_D$ denote subsets of $\mathcal{R}$ such that

$$\mathcal{R}_4 = \{ k | k \in \mathcal{R}, \tilde{k}_1 < N, \text{ and } \tilde{k}_2 < N \},$$  \hspace{1cm} (A7a)

$$\mathcal{R}_B = \{ k | k \in \mathcal{R}, \tilde{k}_1 < N, \text{ and } \tilde{k}_2 \geq N \},$$  \hspace{1cm} (A7b)

$$\mathcal{R}_C = \{ k | k \in \mathcal{R}, \tilde{k}_1 \geq N, \text{ and } \tilde{k}_2 < N \},$$  \hspace{1cm} (A7c)

$$\mathcal{R}_D = \{ k | k \in \mathcal{R}, \tilde{k}_1 \geq N, \text{ and } \tilde{k}_2 \geq N \}. $$  \hspace{1cm} (A7d)

Then $(k_1, k_2)$ of (20) is obtained by taking $(\tilde{k}_1, \tilde{k}_2)$ for all $k$ in $\mathcal{R}_A; (k_1, 2N_1 - 1 - k_2)$ for all $k$ in $\mathcal{R}_B; (2N_1 - 1 - \tilde{k}_1, \tilde{k}_2)$ for all $k$ in $\mathcal{R}_C; (2N_2 - 1 - k_1, 2N_2 - 1 - k_2)$ for all $k$ in $\mathcal{R}_D$. Also, we have the relation

$$\mathcal{R}_A \cup \mathcal{R}_B \cup \mathcal{R}_C \cup \mathcal{R}_D = \mathcal{R}.$$  \hspace{1cm} (A8)

for the operation $\tilde{\mathcal{R}}$ defined in Appendix A. Therefore, to prove the one-to-oneness of $g$, it suffices to show that no two $(k_1, k_2)$ taken from the above four sets can be identical. We prove this by contradiction.

To begin with, we suppose that a $(k_1, k_2)$ obtained from $\tilde{k}$ in $\mathcal{R}_4$ is identically obtained from $\tilde{k}$ in $\mathcal{R}_4$. Then, by (20) we have

$$
\begin{align*}
\tilde{k} &= 2a_1 N_1 + k_1, \hspace{1cm} (A9a) \\
\tilde{k} &= 2a_2 N_2 + k_2, \hspace{1cm} (A9b)
\end{align*}
$$

and

$$\begin{align*}
\tilde{k} &= 2b_1 N_1 + k_1, \hspace{1cm} (A10a) \\
\tilde{k} &= 2b_2 N_2 + (2N_2 - 1 - k_2), \hspace{1cm} (A10b)
\end{align*}
$$

where $a_1, a_2, b_1, b_2$ are integers. By (A9a) and (A10a), we have

$$\tilde{k} = 2(a_1 + a_2) N_1,$$

and, by (A9b) and (A10b), we obtain

$$\tilde{k} + \tilde{k} = 2(a_2 + b_2 + 1) N_2 - 1.$$  \hspace{1cm} (A11b)

But it is a contradiction because by (A10a) $\tilde{k}$ and $\tilde{k}$ are both even or both odd, while by (A10b) only one is even and the other one is odd.

Second, we suppose a $(k_1, k_2)$ obtained from $\tilde{k}$ in $\mathcal{R}_4$ is identically obtained from $\tilde{k}$ in $\mathcal{R}_4$. Then, by (20), we obtain

$$\begin{align*}
\tilde{k} &= c_1 N + (2N_1 - 1 - k_1), \hspace{1cm} (A12a) \\
\tilde{k} &= c_2 N + (2N_2 - 1 - k_2), \hspace{1cm} (A12b)
\end{align*}
$$

where $c_1, c_2$ are integers. By (A9a) and (A12a), we have

$$\tilde{k} + \tilde{k} = 2(a_1 + c_1 + 1) N_1 - 1.$$  \hspace{1cm} (A13a)

and, by (A9b) and (A12b), we obtain

$$\tilde{k} + \tilde{k} = 2(a_2 + c_2 + 1) N_2 - 1.$$  \hspace{1cm} (A13b)

Since the smallest integer $\tilde{k} + \tilde{k}$ that satisfies both (A13a) and (A13b) is $2N_1 N_2 - 1$, the fact that $\tilde{k}$ is in $\mathcal{R}$ implies that $\tilde{k}$ is not in $\mathcal{R}$, and vice versa. Therefore, the assumption leads to a contradiction.

We can show the other four cases in a similar manner, by applying the above reasons. This proves the one-to-oneness of $g$.

**References**


Byeong Gi Lee (S’80-M’82) was born in Dae-chon, Korea, on May 12, 1951. He received the B.S. and M.E. degrees in 1974 and 1978, respectively, from Seoul National University, Seoul, and Kyungpook National University, Taegu, Korea, both in electronics engineering; and the Ph.D. degree in 1982 from the University of California, Los Angeles, in electrical engineering.

From 1974 to 1979 he was with the Department of Electronics Engineering of ROK Naval Academy, Chinhae, Korea, as an Instructor and Naval Officer in active service. From 1982 to 1984 he worked for Granger Associates, Santa Clara, CA, as a Senior Engineer doing research and development on applications of digital signal processing to digital transmission. During the period 1984 to 1986 he worked for AT&T Bell Laboratories, North Andover, MA, as a member of the Technical Staff participating in lightwave transmission system development along with related standard works. Since September 1986 he has been with the Department of Electronics Engineering, Seoul National University, Seoul, Korea. His current fields of interest include theory and applications of digital signal processing, digital transmission and lightwave transmission systems, and circuit theory. He is the author of Electronics Engineering Experiment Series (5 volumes, all in Korean) and holds four U.S. patents.

Dr. Lee received the 1984 Myrl B. Reed Best Paper Award from the Midwest Symposium on Circuits and Systems, and exceptional contributions awards from AT&T Bell Laboratories. He is a member of Sigma Xi.