Efficient Prime Factor Algorithm and Address Generation Techniques for the Discrete Cosine Transform

Lap-Pui Chau, Daniel Pak-Kong Lun, and Wan-Chi Siu

***Abstract*—This brief proposes an efficient prime factor algorithm for the discrete cosine transform. In the proposal, we formulate the decomposition directly, by using the proposed input and output mapping, a novel in-place address generation scheme for input index mapping is derived, while the formulations in the literature require a table to store the index mapping. Besides, our approach requires one output index mapping only while the conventional algorithms require two index mapping. Hence, by using the proposed mappings and address generation techniques, less temporary storage is required, such that a reduction on memory requirement can be achieved during the implementation. A comparison of the address generation time between our approach and the conventional approach is also shown.**

***Index Terms*—Address generation technique, discrete cosine transform, fast algorithm, prime factor algorithm.**

I. Introduction

The discrete cosine transform (DCT) [1]–[8] has been widely used in the area of digital signal processing because of its energy packing capabilities and also approaching to the statistically optimal transform in decorrelating a signal governed by a Markov model. Recently, researchers have been prominent in developing algorithms for real-time implementation of DCT at high bit rates and which is now recommended as a standard transform for image coding and data compression by various standard committees. This coding scheme can be applied in teleconferencing, videophones and high-definition television system.

Since DCT had been appeared, many efficient algorithms were proposed to improve the computing speed and hardware complexity. Previously, Yang, Narasimha, and Lee [3], [4] developed a prime factor algorithm for the DCT but its index mapping is quite complicated. Lee [5] realized their input index mapping by constructing two index tables but it occupies extra memory. Chakrabarti and JaJa [6] implemented the DCT from DHT and found that the systolic architecture was suitable for its realization. Lee and Huang [7] modified Lee's [5] algorithms and based on the previous research efforts to divide the input mapping into five disjoint groups such that the complexity of the algorithm is decreased. Recently, Bi [8] proposed a straightforward approach to convert a one-dimensional (1-D) input sequence into a two-dimensional (2-D) array, which reduce the computational complexity and also avoid irregular computational structure.

Although there are several algorithms [3]–[8] for computing the DCT of an arbitrary length using a prime factor decomposition technique, these algorithms usually need to calculate two output index functions, namely $N_2k_1 + N_1k_2$ and $N_2k_1 - N_1k_2$, respectively.

Besides, some researchers concentrate on the realization of the algorithm expecting to reduce the complexity of the implementation, however in-place address generation technique is also important to reduce the storage requirement while implementation. Let us recall that conventional methods [3]–[8] require a table to store the input index mapping, all these methods required extra memory to store this index table. Many different efficient techniques [9]–[12] have been reported in the literature for an in-place computation and address generation of the DFT. The term "in-place" is defined as the computation where the memory requirement is restricted to be as large as the transform length. Actually, a very small amount of temporary buffers is usually allowed for the storage of the intermediate results while implementation.

In this brief, we propose an algorithm to directly implement the prime factor DCT. This algorithm involves one output index function $N_2k_1 + N_1k_2$ only, such that the memory space and the computational complexity are greatly reduced. Besides, we propose a novel in-place address generation scheme for input index addressing which generates the address column by column or row by row during the DCT processing, by using this address generation scheme, a table to store the mapping sequence is not required. Furthermore, the address generation time is improved by about 20%.

II. Direct Decomposition of Prime Factor DCT Algorithm

The DCT of a sequence $\{x(i'):\ i' = 0, 1, \ldots, N-1\}$ can be written as

$$Y(k) = \sum_{i'=0}^{N-1} x(i') \cos\left[\frac{\pi}{2N}(2i'+1)k\right], \quad \text{for } k = 0, 1, \ldots, N-1. \tag{1}$$

In order to obtain a simplified form of the 2-D DCT, the input sequence $x(i')$ should be permuted by (2) and (3)

$$i' = 2i, \qquad \text{for } i = 0, \ldots, \frac{N-1}{2} \tag{2}$$

$$= 2N - 2i - 1 \quad \text{for } i = \frac{N-1}{2} + 1, \ldots, N-1. \tag{3}$$

Now consider the transform length N which can be factorized into two mutually prime factors N_1 and N_2, respectively, then this 1-D DCT can be decomposed into a 2-D DCT via a suitable bijective mapping. Let $i = f(i_1, i_2)$ be the input index mapping and $k = g(k_1, k_2)$ be the output index mapping.

Now we define

$$i = \left\langle N_2N_2^{-1}i_1 + N_1N_1^{-1}i_2 \right\rangle_N \tag{4}$$

$$k = \langle N_1k_2 + N_2k_1 \rangle_N \tag{5}$$

where $\langle N_1N_1^{-1}\rangle_{N_2} = 1$, $\langle N_2N_2^{-1}\rangle_{N_1} = 1$ and $\langle x\rangle_N = x$ modulo N for $i_1, k_1 = 0, 1, \ldots, N_1 - 1$ and $i_2, k_2 = 0, 1, \ldots, N_2 - 1$. Because $f(i_1, i_2)$ and $g(k_1, k_2)$ are bijective mapping, (1) becomes

$$Y(k_2, k_1) = \sum_{i_2=0}^{N_2-1} \sum_{i_1=0}^{N_1-1} x(i_2, i_1) \cdot \cos\left[\frac{2\pi}{N}\langle N_1i_2k_2 + N_2i_1k_1\rangle_N + \frac{\pi}{2N}\langle N_1k_2 + N_2k_1\rangle_N\right]. \tag{6}$$

According to (6), we note that the first term of the cosine argument $\langle N_2i_1k_1 + N_1i_2k_2\rangle_N$ is multiplied by $(2\pi/N)$, which means that the value of the cosine function remains unchanged if $N_2i_1k_1 + N_1i_2k_2 > N$ after the decomposition. Note that the second term of cosine argument $\langle N_1k_2 + N_2k_1\rangle_N$ is multiplied by $(\pi/2N)$, hence the value of the cosine function will be changed if $N_1k_2 + N_2k_1 > N$ after the

Manuscript received September 6, 1999; revised September 15, 2001. This paper was recommended by Associate Editor A. Skodras.

L.-P. Chau is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: lpchau@ieee.org).

D. P.-K. Lun and W.-C. Siu are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong.

Publisher Item Identifier S 1057-7130(01)11057-8.

decomposition, so the equation should be compensated for the case of $N_1k_2 + N_2k_1 > N$.

If $N_1k_2 + N_2k_1 < N$, the value of the cosine function is unchanged after decomposition, (6) becomes

$$\begin{aligned} Y(k_2, k_1) = &\sum_{i_2=0}^{N_2-1}\sum_{i_1=0}^{N_1-1} x(i_2, i_1) \\ &\cdot\left\{\cos\left[\frac{\pi}{2N_2}(4i_2+1)k_2\right]\cos\left[\frac{\pi}{2N_1}(4i_1+1)k_1\right]\right\} \\ &-\sum_{i_2=0}^{N_2-1}\sum_{i_1=0}^{N_1-1} x(i_2, i_1) \\ &\cdot\left\{\cos\left[\frac{\pi}{2N_2}(4i_2+1)(N_2-k_2)\right]\right. \\ &\left.\cdot\cos\left[\frac{\pi}{2N_1}(4i_1+1)(N_1-k_1)\right]\right\}. \end{aligned} \tag{7}$$

If $N_1k_2 + N_2k_1 > N$, (6) should be compensated by $-(\pi/2)$, and (6) becomes

$$\begin{aligned} &Y(k_2, k_1) \\ &= \sum_{i_2=0}^{N_2-1}\sum_{i_1=0}^{N_1-1} x(i_2, i_1) \\ &\cdot\left\{\cos\left[\frac{\pi}{2N_2}(4i_2+1)(N_2-k_2)\right]\cos\left[\frac{\pi}{2N_1}(4i_1+1)k_1\right]\right\} \\ &+\sum_{i_2=0}^{N_2-1}\sum_{i_1=0}^{N_1-1} x(i_2, i_1) \\ &\cdot\left\{\cos\left[\frac{\pi}{2N_2}(4i_2+1)k_2\right]\cos\left[\frac{\pi}{2N_1}(4i_1+1)(N_1-k_1)\right]\right\}. \end{aligned} \tag{8}$$

In order to obtain the same type of kernel of the DCT as shown in (1), the same permuting algorithm as in (2) and (3) should be applied to (7) and (8), so we have the mappings as shown in (9)–(12).

Let

$$i_2' = 2i_2 \qquad \text{for } i_2 = 0, \ldots, \frac{N_2-1}{2} \tag{9}$$

$$= 2N_2 - 2i_2 - 1 \quad \text{for } i_2 = \frac{N_2-1}{2}+1, \ldots, N_2-1 \tag{10}$$

and

$$i_1' = 2i_1 \qquad \text{for } i_1 = 0, \ldots, \frac{N_1-1}{2} \tag{11}$$

$$= 2N_1 - 2i_1 - 1 \quad \text{for } i_1 = \frac{N_1-1}{2}+1, \ldots, N_1-1 \tag{12}$$

and let us define $Y'(k_2, k_1)$ to be the output of the 2-D DCT

$$\begin{aligned} Y'(k_2, k_1) = &\sum_{i_2'=0}^{N_2-1}\sum_{i_1'=0}^{N_1-1} x(i_2', i_1') \\ &\cdot\left\{\cos\left[\frac{\pi}{2N_2}\left(2i_2'+1\right)k_2\right]\cos\left[\frac{\pi}{2N_1}\left(2i_1'+1\right)k_1\right]\right\} \end{aligned} \tag{13}$$

we have

$$Y(k_2, k_1) = Y'(k_2, k_1) - Y'(N_2 - k_2, N_1 - k_1) \tag{14}$$

$$Y(k_2, k_1) = Y'(N_2 - k_2, k_1) + Y'(k_2, N_1 - k_1). \tag{15}$$

From (13) to (15), $Y(k_2, k_1)$ can be obtained by combining two 2-D DCT outputs. The major advantage of the 2-D DCT is that it can be implemented by short length DCT modules so that the number of operations will be greatly reduced.

TABLE I
INPUT ADDRESS MAPPING FOR 15-POINT DCT

$i_1 \backslash i_2$	0	1	2	3	4
0	x(0)	x(11)	x(12)	x(6)	x(5)
1	x(10)	x(1)	x(7)	x(13)	x(4)
2	x(9)	x(8)	x(2)	x(3)	x(14)

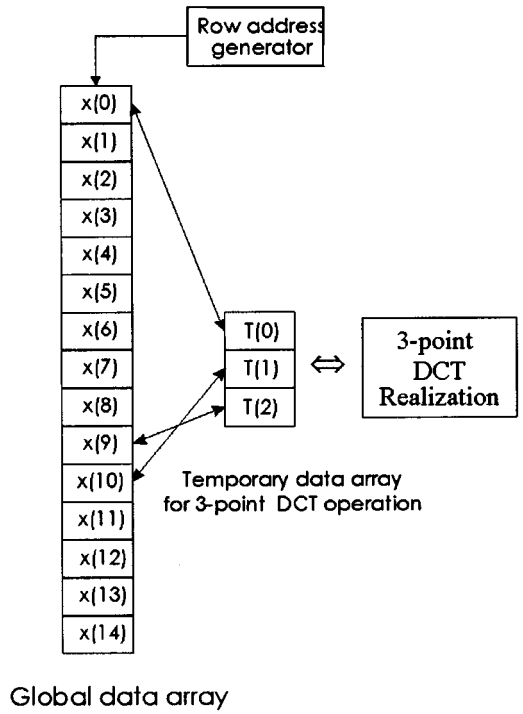

Fig. 1. Data loading and retrieval operation of 3-point DCT algorithm.

III. ADDRESS GENERATION TECHNIQUES FOR INPUT INDEX MAPPING

The address generation scheme can be derived by permuting the input sequence. By substituting (4) and (5) and (9)–(12) into (2) and (3), we can obtain two sets of equations.

Formula A:

$$A_C = \begin{cases} \langle k_C N_2 + C\rangle_{2N} \\ \langle k_C N_2 - C - 1\rangle_{2N} \end{cases}$$

for K_C is even from 0 to $N_1 - 1$ and $C = 0$ to $N_2 - 1$.

Formula B:

$$A_R = \begin{cases} \langle k_R N_2 + R\rangle_{2N} \\ \langle k_R N_2 - R - 1\rangle_{2N} \end{cases}$$

for K_R is even from 0 to $N_2 - 1$ and $R = 0$ to $N_1 - 1$ where A_C and A_R are the addresses in Column C and Row R, respectively.

Example: In this example a 15-point prime factor DCT is going to be evaluated. Let $N_1 = 3$ and $N_2 = 5$. According to Formula A, the addresses in column 0 are 0, $2N_2 - 1 = 9$, $2N_2 = 10$, respectively. According to Formula B, the address in row i_1' can be determined by $\langle address\rangle_{2N_1} = i_1'$ or $2N_1 - i_1' - 1$ i.e., $\langle 0\rangle_6 = 0$, $\langle 9\rangle_6 = 3 = 6 - 2 - 1$, $\langle 10\rangle_6 = 4 = 6 - 1 - 1$. The row numbers for addresses 0,

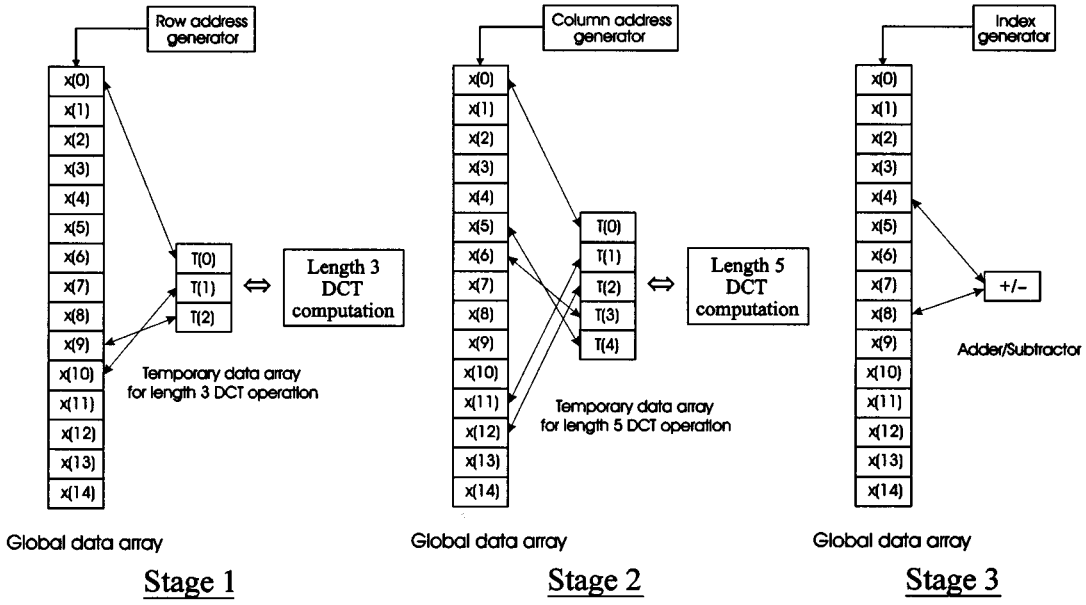

Fig. 2. Graphical representation for in-place implementation of 15-point prime factor DCT algorithm.

TABLE II
OUTPUT MAPPING FOR 15-POINT DCT

$k_1 \backslash k_2$	0	1	2	3	4
0	Y(0)	Y(3)	Y(6)	Y(9)	Y(12)
1	Y(5)	Y(8)	Y(11)	Y(14)	Y(2)
2	Y(10)	Y(13)	Y(1)	Y(4)	Y(7)

9, and 10 are 0, 2, and 1, respectively. Table I shows the input mapping for a 15-point prime factor DCT.

For the realization of the prime factor DCT algorithm, the computation in each dimension is independent. The global data array store the input data in sequence, then the address generation scheme for input index is used to select the global data to temporary data array, since the address index is generated in an on-demand basis that means the relevant row or column address is generated when the system request during the DCT processing. So no table is required to store the indexing map. The temporary data are processed using the short length DCT algorithm, e.g., [2]. After the short length DCT implementation, the temporary data will be passed back to the global data array. Fig. 1 demonstrates the operations of the in-place prime factor DCT algorithm.

IV. OUTPUT INDEX MAPPING

In this section, we arrange to obtain the output index mapping $k = \langle N_1k_2 + N_2k_1 \rangle_N$. From the previous direct decomposition, the transform output can be obtained after some add and subtract operations. In case of $N_1k_2 + N_2k_1 < N$, subtraction is required. Otherwise, addition is performed. It is obviously that the decomposition is a decimation-in-frequency algorithm, because the transformed sequence is grouped into different frequency index parts. That is, the input is in normal order and the output requires permutation. Actually, the decomposition is an in-place algorithm but the output is not in-order. In order to get the output in-order, an extra unscrambling is required. Table II shows the output mapping for a length-15 prime-factor DCT.

From the above mapping and the previous derivation, we know that if $N_1k_2 + N_2k_1 < N$, we have

$$
\begin{aligned}
Y(0) &= Y'(0,0) = Y(0,0), Y(3) = Y'(1,0) = Y(1,0),\\
Y(6) &= Y'(2,0) = Y(2,0), Y(9) = Y'(3,0) = Y(3,0)\\
Y(12) &= Y'(4,0) = Y(4,0), Y(5) = Y'(0,1) = Y(0,1),\\
Y(10) &= Y'(0,2) = Y(0,2),\\
Y(8) &= Y(1,1) = Y'(1,1) - Y'(5-1,3-1) = Y'(1,1) - Y'(4,2)\\
Y(11) &= Y(2,1) = Y'(2,1) - Y'(5-2,3-1) = Y'(2,1) - Y'(3,2)\\
Y(14) &= Y(3,1) = Y'(3,1) - Y'(5-3,3-1) = Y'(3,1) - Y'(2,2)\\
Y(13) &= Y(1,2) = Y'(1,2) - Y'(5-1,3-2) = Y'(1,2) - Y'(4,1)
\end{aligned}
$$

if $N_1k_2 + N_2k_1 > N$, we have

$$
\begin{aligned}
Y(2) &= Y(4,1) = Y'(5-4,1) + Y'(4,3-1) = Y'(1,1) + Y'(4,2)\\
Y(1) &= Y(2,2) = Y'(5-2,2) + Y'(2,3-2) = Y'(3,2) + Y'(2,1)\\
Y(4) &= Y(3,2) = Y'(5-3,2) + Y'(3,3-2) = Y'(2,2) + Y'(3,1)\\
Y(7) &= Y(4,2) = Y'(5-4,2) + Y'(4,3-2) = Y'(1,2) + Y'(4,1).
\end{aligned}
$$

Fig. 2 shows the steps for the computation of 15 point prime factor decomposition algorithm with unscrambling output in global data array. The set $\{x_i,\ i = 0,\ 1,\ \ldots,\ 14\}$ is the input sequence, which is stored in global data array and mapped according to our algorithm into the 2-D DCT array. After the two stages row–column evaluation, the output data can be obtained by some add and subtract operations according to the input and output index function.

V. SOFTWARE PERFORMANCE FOR INPUT INDEX MAPPING

The new address generation scheme has been tested using the Microsoft C. The aim of this section is to give the results of comparison of our proposed address generation scheme with the conventional scheme which was proposed by Lee [5]. Fig. 3 shows the percentage improvement of the new approach as compared with the conventional approach. As shown in the figure, the new approach improves an average of about 20% in speed over the address generation using the conventional approach.

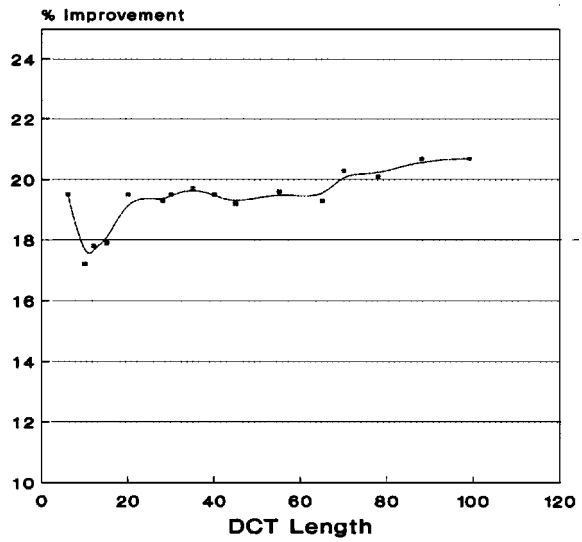

Fig. 3. Percentage improvement for the new approach to the conventional approach.

VI. Conclusion

In this brief, an efficient implementation of prime factor algorithm has been developed. Our objective is to design a prime factor algorithm which requires less memory consumption while implementation such that it can be implemented using built-in-memory processor or hardware implementation. Unlike other existing prime factor DCT algorithms, this address generation algorithm generate the input index mapping in on-demand basis which does not require a table to store the index mapping. For the long length prime factor DCT application, due to the memory restriction, it is not practical to generate an extra temporary 2-D address array. This shows the importance of our approach for an in-place address generation technique. On the other hand, we have also discussed the output index mapping, which is based on the equation $N_1k_2+N_2k_1$, the outputs can be obtained in a straightforward way after a fast unscrambling procedure. The memory space and implementation complexity is further reduced since the equation $N_1k_2 - N_2k_1$ appeared in reference [5] is eliminated. Finally, the address generation time is also improved by about 20% over a wide range of DCT lengths.

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. 23, pp. 90–93, Jan. 1974.
[2] L. P. Chau and W. C. Siu, "Direct formulation for the realization of discrete cosine transform using recursive structure," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 50–52, Jan. 1995.
[3] P. P. N. Yang, M. J. Narasimha, and B. G. Lee, "A prime factor decomposition algorithm for the discrete cosine transform," in *Proc., Int. Conf. Comput., Syst. Signal Processing*, India, 1984, pp. 132–135.
[4] P. P. N. Yang and M. J. Narasimha, "Prime factor decomposition of the discrete cosine transform and its hardware realization," in *Proc., IEEE Int. Conf. Acoust., Speech Signal Processing*, San Diego, CA, 1984, pp. 772–775.
[5] B. G. Lee, "Input and output index mappings for a prime-factor-decomposed computation of discrete cosine transform," *IEEE Trans. Acoust., Speech Signal Processing*, vol. 37, pp. 237–244, Feb. 1989.
[6] C. Chakrabarti and J. JaJa, "Systolic architectures for the computation of the discrete hartley and the discrete cosine transforms based on prime factor decomposition," *IEEE Trans. Comput.*, vol. 39, pp. 1359–1368, Nov. 1990.
[7] P. Z. Lee and F. Y. Huang, "An efficient prime-factor algorithm for the discrete cosine transform and its hardware implementations," *IEEE Trans. Signal Processing*, vol. 42, pp. 1996–2005, Aug. 1994.
[8] G. Bi, "Index mapping for prime factor algorithm of discrete cosine transform," *Electron. Lett.*, vol. 35, no. 3, pp. 198–200, Feb. 1999.
[9] C. S. Burrus and P. W. Eschenbacher, "An in-place, in-order prime factor FFT algorithm," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-29, pp. 806–817, Aug. 1981.
[10] C. Temperton, "Implementation of a self-sorting in-place prime factor FFT algorithm," *J. Computat. Phys.*, vol. 58, pp. 283–299, Oct. 1985.
[11] K. L. Wong, R. Chan, D. P. K. Lun, and W. C. Siu, "Efficient address generation for prime factor algorithms," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, pp. 1518–1528, Sept. 1990.
[12] D. P. K. Lun and W. C. Siu, "An analysis for the realization of an in-place and in-order prime factor algorithm," *IEEE Trans. Signal Processing*, vol. 41, pp. 2362–2370, July 1993.

On RNS-Based Enhancements for Direct Digital Frequency Synthesis

P. V. Ananda Mohan

***Abstract*—It is shown that Chren's techniques for reducing the ROM requirements for the implementation of reside number systems–based digital frequency synthesis are not correct.**

***Index Terms*—Digital frequency synthesis (DFS), residue number systems (RNS), VLSI architectures.**

I. Introduction

The use of residue number systems (RNS) for digital frequency synthesis (DFS) has been considered by Chren, [1], [2]. The classical digital frequency synthesizers facilitate generation of sinusoidal output by using the architecture of Fig. 1(a). Herein, the "sine" function is stored in ROM and is sequentially read. The L bit address of the ROM is generated by a phase accumulator of L bit resolution which repeatedly adds a phase increment of k corresponding to the frequency to be generated. Since, sine-wave is symmetric, only sine function for one quadrant (0 to 90 degrees) is stored. The L bit word, however, corresponds to a maximum of 360 degrees. The MSB of this word contains sign information and hence used to generate two's complement of the contents read from the ROM so as to obtain negative value of the sample stored in ROM. In addition, the bit next to the MSB is used to selectively invert the lower order bits of the address by using exclusive-OR gates controlled by the second MSB. This can be appreciated by noting that for the case of values in the second quadrant, complementing the bits yields an address corresponding to the correct value. Assuming a clock frequency of f_c and frequency setting word k, and an L bit accumulator, the frequency of the sinusoid generated is $f_c.k/2^L$. The phase increment is $2.\pi.k/2^L$.

The figures of merit of a DFS design are frequency, phase, and amplitude resolution. The frequency resolution is evidently $f_c/2^L$ corresponding to $k = 1$. Alternatively, the width of the accumulator is the frequency resolution. Phase resolution is the width W of the ROM address. Note that while phase accumulation can be done with resolution as desired, the restriction on the memory size may need truncation of L bits to W bits. Amplitude resolution is evidently the number of bits used to represent the stored samples in the ROM.

Manuscript received July 5, 2000; revised October 4, 2001. This paper was recommended by Associate Editor W. Sandham.

The author is with Transmission R&D, I.T.I. Ltd., Bangalore 560016, India.

Publisher Item Identifier S 1057-7130(01)11058-X.