High-Speed Area-Efficient Recursive DFT/IDFT Architectures

Lan-Da Van and Chih-Chyau Yang

Chip Implementation Center (CIC), National Applied Research Laboratories,
No. 1, Prosperity Rd. 1, Science-Based Industrial Park, Hsinchu, Taiwan, R.O.C.

E-mail: {ldvan, ccyang} @cic.org.tw

ABSTRACT

In this paper, we propose several high-speed area-efficient recursive discrete Fourier transform (DFT)/inverse DFT (IDFT) designs adopting the module-sharing and register-splitting schemes. The proposed core architecture achieves one multiplier reduction as well as less critical period and a saving of nearly half multiplications compared with the second-order and first-order recursive DFT structures, respectively. So as to reduce the number of computation cycles, based on the new core design, we develop the area-efficient parallel and folded recursive DFT/IDFT architectures. Moreover, due to the advantages of regular and modular structure, the resulting high-speed area-efficient recursive DFT/IDFT architectures are amenable to application-specific integrated circuit (ASIC) design.

1. Introduction

The discrete Fourier transform (DFT) has been widely applied in the analysis and implementation of discrete-time signal processing [1] and communication systems such as dual tone multi-frequency (DTMF) application [2-3]. In many applications, the complex sequences in time-domain are expected to be frequency-domain signals via the DFT computation. Without loss of generality, the input data is assumed as complex-valued data. From existing research, there are possible four categories for the structures of DFT computations: 1) recursive-algorithm based architecture [1-6], 2) butterfly-based architecture [1, 7], 3) ROM operation based structure [8], and 4) multiplier-accumulator based structure. It is well known that the DFT architectures based on the recursive algorithm are more area-efficient than those realized by other approaches. Until now, the existing recursive algorithms for the orthogonal transform in the scope of DFT/DCT/DST (discrete Fourier/cosine/sine transform) involve the following: Goertzel algorithm [1-6, 9], C-S's algorithm [10], Chebyshev polynomials [11], and Clenshaw's recurrence formula (CRF) [12-13]. In [10-12], recursive expressions for the computation of the DCT-II that are suitable for VLSI implementation are presented. The recursive DCT-II architecture [11] is based on Chebyshev polynomials of the third kind while those in [12] are based on CRF. Recently, Kidambi [13] furnished recursive DCT-IV and DST-IV architectures, where this approach can be possible to develop recursive DFT architecture. Note that in [10-13], recursive algorithms are used to design recursive DCT/DST architectures rather than recursive DFT architecture. In [1, 6], the original second-order recursive DFT architecture derived from Goertzel algorithm has one redundant multiplier and thus we can reuse the same multiplier to save the redundant one. This area-reduction strategy is referred to as the module-sharing scheme. Thus, the modified recursive DFT architecture has lower area than the preceding one [1, 6]. Next, we apply register-splitting scheme [14] to speedup the area-efficient architecture without affecting the system transfer function. Therefore, the proposed architecture possesses the following features: high speed, reduction of one multiplier compared with the second-order recursive DFT structure, and a saving of nearly half multiplications for each DFT output compared with the first-order recursive DFT structure. Regarding the new area-efficient recursive DFT/IDFT architecture as a core, we can develop parallel- and folded-type architectures to achieve less computation cycles for real-time media applications. The paper is organized as follows. A review of the first- and second-order recursive DFT structures is given in Section 2. In Section 3, we propose three new recursive DFT/IDFT architectures by module-sharing and register-splitting schemes: core-, parallel-, and folded-type architectures. In Section 4, comparison results are tabulated in terms of the critical period, the number of real multipliers, the amount of real multiplications as well as real additions for each DFT/IDFT output sequence, and the number of computation cycles for N-point DFT/IDFT. At last, the concise statements conclude this paper.

2. A Review of First- and Second-Order Recursive DFT Structures

Given input sequence and DFT output sequence denoted as $x[n]$ and $X[K]$, respectively, the N-point DFT can be defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} , \tag{1}$$

where $W_N = e^{-j2\pi/N}$. The Goertzel algorithm [4] making use of the periodicity of the sequence W_N^{kn} can be used to reduce computation. For convenience of deriving a new architecture, we begin a review of the recursive DFT expression based on Goertzel algorithm by noting that

$$W_N^{-kN} = e^{j(2\pi/N)Nk} = e^{j2\pi k} = 1 . \tag{2}$$

Because of Eq. (2), we may multiply the right side of Eq. (1) by W_N^{-kN} without affecting the equation. Thus,

$$X[k] = W_N^{-kN} \sum_{r=0}^{N-1} x[r] W_N^{kr} = \sum_{r=0}^{N-1} x[r] W_N^{-k(N-r)} . \tag{3}$$

In order to simplify the final expression, let us define the sequence

$$y_k(n) = \sum_{r=-\infty}^{\infty} x[r] W_N^{-k(n-r)} u[n-r] . \tag{4}$$

From Eqs. (3) and (4) and the fact that $x[n] = 0$ for $n < 0$ and $n \geq N$, it follows that

$$X[k] = y_k[n] \big|_{n=N} . \tag{5}$$

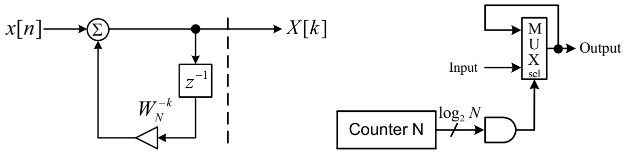
Eq. (4) can be interpreted as a discrete convolution of the finite-duration sequence $x[n]$, $0 \leq n \leq N-1$, with the sequence $W_N^{-kn} u[n]$. As a consequence, $y_k[n]$ can be

0-7803-8251-X/04/$17.00 ©2004 IEEE

regarded as the response of a system with impulse response $W_N^{-kn}u[n]$ to a finite-length input $x[n]$. In particular, $X[k]$ is the value of the output when $n = N$. Taking the z-transform of Eq. (4), we can obtain the first-order transfer function as

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} . \qquad (6)$$

Eq. (6) can be mapped into the first-order recursive DFT structure as shown in Fig. 1(a), where initial rest conditions are assumed and the vertical dash-line denotes the down-sample operation with N for each crossing signal path. Note that the dash-line as shown in Fig. 1(a) can be possibly implemented by multiplexer-type or register-type down-sampling realization. Here, we adopt the multiplexer-type down-sampling realization as shown in Fig. 1(b) due to the advantages of less area and exact mapping from the equation to the architecture. In Fig. 1(b), if $sel = 1$, the lower-side signal is passed to the output; otherwise, the upper-side signal is selected as the output signal for the multiplexer. In this correspondence, since the input $x[n]$ and the coefficient W_N^{-k} are in complex domain, the computation of each new value of $y_k[n]$ through the first-order recursive DFT structure as shown in Fig. 1(a) requires four real multiplications and four real additions. All the intervening values $y_k[1]$, $y_k[2], \ldots,$ $y_k[N-1]$ must be computed in order to compute $y_k[N] = X[k]$, so the use of the first-order recursive DFT structure as a computational algorithm requires $4N$ real multiplications and $4N$ real additions to compute $X[k]$ for a particular value of k. However, a large number of multiplications are required for the first-order recursive DFT architecture, even if the one avoids the computation or storage of the coefficients W_N^{kn} in Eq. (1) at each n th time index.



(a) (b)

Fig. 1. (a) Block diagram of the first-order recursive DFT structure and (b) a multiplexer-type dash-line implementation with down-sampling value of N.

It is possible to retain this simplification while reducing the number of multiplications by a factor of 2. To see how this may be treated, the transfer function of the first-order recursive DFT structure in Fig 1(a) can be noted. Multiplying both the numerator and the denominator of $H_k(z)$ by the factor $(1 - W_N^k z^{-1})$, we obtain

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{(1 - W_N^{-k} z^{-1})(1 - W_N^k z^{-1})}$$

$$= \frac{1 - W_N^k z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} . \qquad (7)$$

Eq. (7) can be mapped into the second-order recursive DFT structure as shown in Fig. 2.

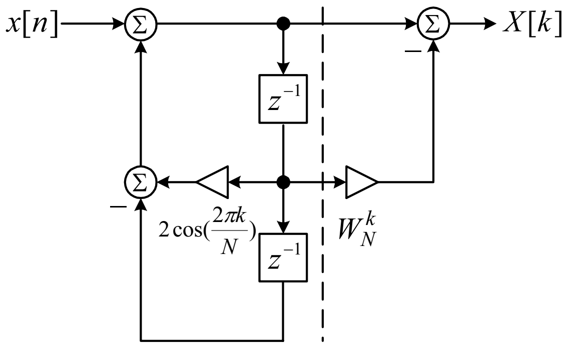


Fig. 2. Block diagram of the second-order recursive DFT structure.

In Fig. 2, only two real multiplications per sample are required to implement the poles of this system as shown in Fig. 2. Note that, in the denominator of Eq. (7), the coefficients are real and the factor –1 need not be counted as a multiplication. It is worthy of emphasizing that the complex multiplication by $-W_N^k$ required to implement the zero of the transfer function need not be performed at every iteration of the difference equation, but only after the N th iteration. Thus, the total computation is $2N$ real multiplications and $4N$ real additions for the poles plus four real multiplications and four real additions for the zero. The coefficients W_N^{kn} are again computed implicitly in the iteration of the recursion formula implied in Fig. 2. The second-order recursive DFT structure can decrease the number of multiplications by Goertzel algorithm; however, the amount of multipliers and the value of the critical period are sacrificed. Hence, the structures in Figs. 1(a) and 2 are not efficient.

3. New Recursive DFT/IDFT Architectures

Keeping in mind that we are encouraged to design an efficient architecture that satisfies the features of the lower critical period (i.e., high speed), less number of multipliers (i.e., low area), and less number of multiplications. Substituting the definition $W_N = e^{-j2\pi/N}$ into Eq. (7), Eq. (7) can be recast as

$$H_k(z) = \frac{1 - \cos(2\pi k/N)z^{-1} + j\sin(2\pi k/N)z^{-1}}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} . \qquad (8)$$

From Eq. (8), we find that there are two the same multiplicands $\cos(2\pi k/N)$ of the multiplier appeared in the first-order of the numerator and denominator. Let the feedforward and feedback signal paths of the first-order go through the same multiplier and then the feedback signal path is adjusted by shift register to obtain the two times result. Based on the above description, we can easily modify the second-order structure as a new area-efficient architecture as shown in Fig. 3, where is a hardwired shifter with one-bit left shift. The above reducing area method is referred to as the module-sharing scheme.

For the speed issue, we adopt the register-splitting scheme (i.e., one kind of retiming schemes), to reduce the critical period and this scheme has been successfully used in 2-D IIR/FIR digital filter [14]. Herein, we define two useful notations 0 and 1, where 0 and 1 indicate that the delay elements as shown in Fig. 3 are at top-to-down and bottom-to-up signal paths, respectively. Thus, we can easily use the digital number sequence to represent different register-splitting structures. For example, the proposed core design in Fig. 3 can be represented as 00. In this case, there are four combinations as listed in Table 1, for register-splitting structures of Fig. 3. With minimum critical period in mind, we select the 10 register-splitting structure for our design. Note that 10 and 11 as listed in Table 1 result in the same DFT design as depicted in Fig. 4. The new DFT architecture owns higher speed and smaller area than the second-order DFT structures. As to the number of operations of the high-speed area-efficient recursive DFT architecture in Fig. 4, only two real multiplications per sample and two real multiplications by $\sin(2\pi k/N)$ are required to implement the poles and the imaginary part of the DFT output, respectively. Thus, the total computation is $2N$ real multiplications and $4N$ real additions for the poles plus two real multiplications and four real additions for the zero.

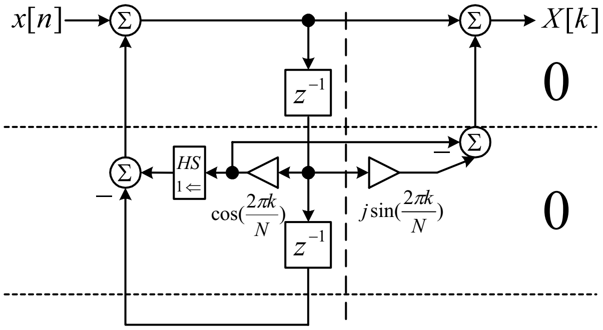


Fig. 3. Block diagram of the proposed area-efficient recursive DFT architecture.

Table 1: Combinations of Register-Splitting Structures

Combinations	00	01	10	11
Critical Period	$T_m + 4T_a$	$T_m + 4T_a$	$T_m + 2T_a$	$T_m + 2T_a$

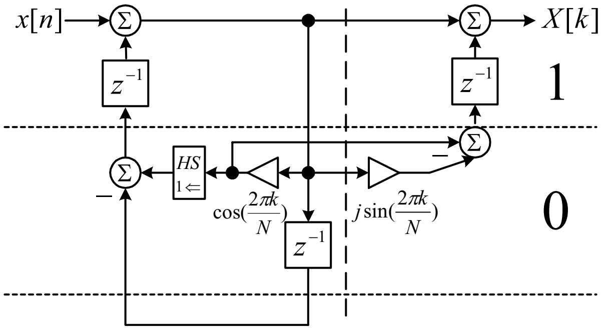


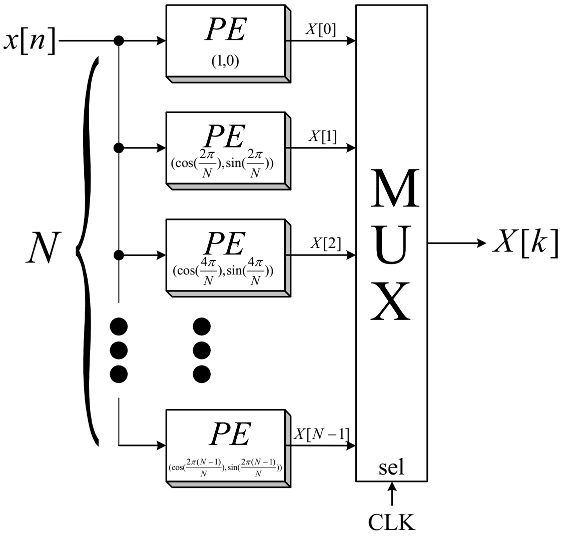
Fig. 4. Block diagram of the proposed high-speed area-efficient recursive DFT architecture.

In similar behavior, the resulting transfer function of the recursive IDFT can be obtained as

$$H_n(z) = \frac{1-\cos(2\pi n/N)z^{-1} - j\sin(2\pi n/N)z^{-1}}{1-2\cos(2\pi n/N)z^{-1} + z^{-2}}. \qquad (9)$$

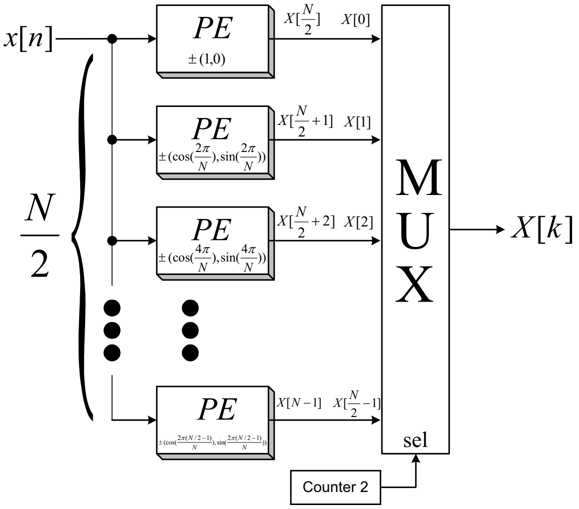
Via the module-sharing and register-splitting schemes, Eq. (9) can be realized as a new recursive IDFT structure.

In order to reduce the number of computation cycles for N-point DFT/IDFT, utilizing this powerful core design as shown in Fig. 4 as a processing element (PE), we can construct the parallel recursive DFT structure as shown in Fig. 5. From comparison results as listed in Table 2, it can be seen that the parallel recursive structure significantly reduces the number of computation cycles from N^2 to $2N$. Importantly, the parallel recursive structure is more area-efficient than that based on the conventional first- and second-order DFT designs. For sake of area saving, the parallel recursive structure can be improved as a folded recursive DFT architecture in Fig. 6 with sacrificing the number of N computation cycles.



$x[n]$
N
PE
(1,0)
$X[0]$
PE
$(\cos(\frac{2\pi}{N}),\sin(\frac{2\pi}{N}))$
$X[1]$
PE
$(\cos(\frac{4\pi}{N}),\sin(\frac{4\pi}{N}))$
$X[2]$
PE
$(\cos(\frac{2\pi(N-1)}{N}),\sin(\frac{2\pi(N-1)}{N}))$
$X[N-1]$
M
U
X
sel
$X[k]$
CLK

Fig. 5. Block diagram of the parallel high-speed area-efficient recursive DFT structure.



$x[n]$
$\frac{N}{2}$
PE
$\pm(1,0)$
$X[\frac{N}{2}]$ $X[0]$
PE
$\pm(\cos(\frac{2\pi}{N}),\sin(\frac{2\pi}{N}))$
$X[\frac{N}{2}+1]$ $X[1]$
PE
$\pm(\cos(\frac{4\pi}{N}),\sin(\frac{4\pi}{N}))$
$X[\frac{N}{2}+2]$ $X[2]$
PE
$\pm(\cos(\frac{2\pi(N/2-1)}{N}),\sin(\frac{2\pi(N/2-1)}{N}))$
$X[N-1]$ $X[\frac{N}{2}-1]$
M
U
X
sel
$X[k]$
Counter 2

Fig. 6. Block diagram of the folded high-speed area-efficient recursive DFT structure.

4. Comparison Results

In this section, we give a comprehensive comparison results as listed in Table 2 in terms of the critical period, the number of real multipliers, the total real multiplications as well as real additions for each DFT/IDFT output, and the number of computation cycles. Let T_m and T_a denote the operation time required for one real multiplication and one real addition, respectively. Note that the operation time of the complex multiplication requires $T_m + T_a$ and the operation time of the multiplexer in Fig. 1(b) compared to T_m and T_a can be ignored here. Our proposed work 1 (i.e., core-type design) has the same highest speed and lowest number of the multipliers as the first-order recursive DFT/IDFT architecture due to applying register-splitting and module-sharing schemes, respectively. Although the first-order recursive DFT/IDFT structure owns the above advantages as listed in the first- and second rows of Table 2, the one cannot overcome the large operations for each DFT/IDFT output. That is, the one needs large power consumption. Our proposed work 1 and the second-order DFT/IDFT architecture based on Goertzel algorithm can save nearly half multiplications for each DFT/IDFT output compared with the first-order recursive DFT/IDFT structure. Furthermore, based on the proposed work 1, we can construct parallel and folded recursive DFT/IDFT architectures. These two architectures can significantly reduce the number of computation cycles for N-point DFT/IDFT from N^2 to $2N$ and $3N$, respectively. Thus, more real-time operation can be achieved. Note that, although these two architectures extra require multipliers, these two ones are still area-efficient compared with those based on conventional core designs. Therefore, in Table 2, it reveals that our proposed architectures have characteristics of high speed, area-efficient, and fewer computing operations.

5. Conclusion

We have devised three new recursive DFT/IDFT architectures based on Goertzel algorithm by the hybrid of module-sharing and register-splitting schemes. The module-sharing scheme can highly reduce the number of multipliers. On the other hand, register-splitting scheme results in a high-speed architecture. Based on Goertzel algorithm, we retain the characteristic of low operations for DFT/IDFT designs.

6. References

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989, ch 9.

[2] M. D. Felder, J. C. Mason, and B. L. Evans, "Efficient dual-tone multifrequency detection using the nonuniform discrete Fourier transform," *IEEE Signal Processing Lett.*, vol. 5, pp. 160-163, Jul. 1998.

[3] S. L. Gay, J. Hartung, and G. L. Smith, "Algorithms for multi-channel DTMF detection for the WE DSP32 family," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1989, pp. 1134-1137.

[4] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *American Math. Monthly*, vol. 65, pp. 34-35, Jan. 1958.

[5] V. V. Cizek, "Recursive calculation of Fourier transform of discrete signal," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1982, pp. 28-31.

[6] T. E. Curtis and M. J. Curtis, "Recursive implementation of prime radix and composite radix Fourier transforms," *IEE Colloquium on Signal Processing Applications of Finite Field Mathematics*, Jun. 1989, pp. 2/1-2/9.

[7] H. V. Sorensen, D. L. Jones, M. T. Heideman, C S. Burrus, "Real-valued fast Fourier transform algorithms," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 35, pp. 849-863, June 1987.

[8] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Acoust., Speech, Signal Processing Mag.,* pp. 4-19, Jul. 1989.

[9] J. hunter and J. V. McCanny, Discrete cosine transform generator for VLSI synthesis," *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1998, pp. 2997-3000.

[10] L. P. Chau and W. C. Sin, "Recursive algorithm for the discrete cosine transform with general lengths," *Elect. Lett.*, vol. 30, no. 3, pp. 197-198, Feb. 1994.

[11] Z. Wang, G. A. Jullien and W. C. Miller, "Recursive algorithms for the forward and inverse discrete cosine transform with arbitrary length," *IEEE Signal Processing Lett.*, vol. 1, no. 7, pp. 101-102, Jul. 1994.

[12] M. F. Aburdene, J. Zheng and R. J. Kozick, "Computation of discrete cosine transform using Clenshaw's recurrence formula," *IEEE Signal Processing Lett.*, vol. 2, no. 8, pp. 155-156, Aug. 1995.

[13] S. S. Kidambi, "Recursive implementation of the DCT-IV and DST-IV," *IEEE Symposium on Advances in Digital Filtering and Signal Processing*, 1998, pp. 106-110.

[14] L. D. Van, "A new 2-D systolic digital filter architecture without global broadcast," *IEEE Trans. VLSI Systs.,* vol. 10, pp. 477-486, Aug. 2002.

Table 2: Comparison Results among the Recursive DFT/IDFT Architectures

Parameters	First-Order DFT/IDFT	Second-Order DFT/IDFT	Proposed Work 1 (Core Type)	Proposed Work 2 (Parallel Type)	Proposed Work 3 (Folded Type)
Critical Period	$T_m + 2T_a$	$T_m + 3T_a$	$T_m + 2T_a$	$T_m + 2T_a$	$T_m + 2T_a$
# of Real Multipliers	4	5	4	$4N$	$2N$
# of Real Multiplications for Each Output $X[K]$ or $x[n]$	$4N$	$2N+4$	$2N+2$	$2N+2$	$2N+2$
# of Real Additions for Each Output $X[K]$ or $x[n]$	$4N$	$4N+4$	$4N+4$	$4N+4$	$4N+4$
# of Computation Cycles for N-Point DFT/IDFT	N^2	N^2	N^2	$2N$	$3N$