Low Computation Cycle and High Speed Recursive DFT/IDFT: VLSI Algorithm and Architecture

Lan-Da Van, Yuan-Chu Yu*, Chun-Ming Huang, and Chin-Teng Lin*
National Chip Implementation Center (CIC),
1F, No. 1, Prosperity Road 1, Science Park,
Hsinchu 300, Taiwan, R.O.C.
e-mail: ldvan@cic.org.tw
*Dept. of Electrical and Control Engineering, National Chiao-Tung University,
Hsinchu, 300, Taiwan, R.O.C.
e-mail: ctlin@mail.nctu.edu.tw

Abstract— In this paper, we propose two low-computation cycle and high-speed recursive discrete Fourier transform (DFT)/inverse DFT (IDFT) architectures adopting the hybrid of Chebyshev polynomial and register-splitting scheme. The proposed core-type recursive architecture achieves half computation-cycle reduction as well as less critical period compared with the conventional second-order DFT/IDFT architecture. So as to further reduce the number of computation cycles, based on the new core-type design, we develop the folded-type recursive DFT/IDFT architecture with the same operating frequency. Moreover, from the derivation results, the operation of DFT and IDFT can be performed with the same structure under different configurations.

I. INTRODUCTION

The discrete Fourier transform (DFT) has been widely applied in the analysis and implementation of communication systems such as OFDM-based wireless local area network (WLAN) [1, 2] and dual tone multi-frequency (DTMF) standards [3]. In many applications, the complex sequences in the time domain are expected to be analyzed in the frequency domain via 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/IDFT computations: 1) recursive-algorithm based architecture [3-8], 2) butterfly-based architecture [9-10], 3) ROM operation based structure [11], 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 [3-7], C-S's algorithm [13], Chebyshev polynomials [8, 14-15], and Clenshaw's recurrence formula (CRF) [16]. In [12-16], recursive expressions for the computation of the DCT that are suitable for VLSI implementation are presented. Note that in [12-16], recursive algorithms are used to design recursive DCT architectures rather than recursive DFT architecture. In [7], compared with the conventional second-order recursive DFT/IDFT architecture, Van *et al*. utilized computation-sharing and register-splitting schemes to reduce two multipliers and speedup the operation, respectively. Nevertheless, Van *et al*. did not improve the computation cycle. In [8], Fan *et al.* applied the previous proposed method to reduce computation cycles but the effect is limited. On the other hand, they only proposed recursive DFT algorithm but IDFT algorithm is not yet ready in [8]. Therefore, we are motivated to propose performance-oriented VLSI algorithm and architecture that possesses the following features: low computation cycle and high speed at the expense of slightly increased area overhead compared with the second-order recursive DFT/IDFT structure. Regarding the new lower computation cycle recursive DFT/IDFT architecture as a core, we can develop a folded architecture to achieve less computation cycles for OFDM-based WLAN applications. The paper is organized as follows. A new core-type recursive DFT algorithm and architecture by the hybrid of Chebyshev polynomial and register-splitting schemes is given in Section II. In Section III, we propose the corresponding novel recursive IDFT algorithm and architecture. In Section IV, a folded architecture that features lower computation cycle is constructed for DFT/IDFT. Complexity comparison results are tabulated in terms of the amount of computation cycles for each output as well as *N*-point DFT/IDFT, the critical period, and the number of real multipliers in Section V. At last, the concise statements remark this paper.

II. NEW RECURSIVE FORMULA FOR DFT

The DFT of the *N*-point input $x[n]$ is defined as

$$y[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn}, \qquad (1)$$

where $W_N = e^{-j2\pi/N}$. To develop recursive DFT algorithm, Eq. (1) can be recast as

$$y[k] = \sum_{n=0}^{N/2-1} x[n]\cdot W_N^{kn} + \sum_{n=0}^{N/2-1} x[N-1-n]\cdot W_N^{k(N-1-n)}$$
$$= \sum_{n=0}^{N/2-1} \left(x[n] + W_N^{-k}\cdot x[N-1-n]\right)\cdot \cos(\frac{2\pi kn}{N})$$
$$+ j\sum_{n=0}^{N/2-1} \left(-x[n] + W_N^{-k}\cdot x[N-1-n]\right)\cdot \sin(\frac{2\pi kn}{N}). \quad (2)$$

By folding two input data, only half summation terms are demanded to express $y[k]$. Eq. (2) can be treated as DCT and DST parts, $y_{DCT}(k)$ and $y_{DST}(k)$, respectively, as

$$y_{DCT}[k] = \sum_{n=0}^{N/2-1} \left(x[n] + W_N^{-k}\cdot x[N-1-n]\right)\cdot \cos(\frac{2\pi kn}{N}), \quad (3)$$

and

$$y_{DST}[k] = -\sum_{n=0}^{N/2-1} \left(x[n] - W_N^{-k}\cdot x[N-1-n]\right)\cdot \sin(\frac{2\pi kn}{N}). \quad (4)$$

In Eq. (3), we can define $r_k[n] = x[n] + W_N^{-k}\cdot x[N-1-n]$. Replacing n by $N/2\text{-}1\text{-}n$, Eq. (3) can be rewritten as

$$y_{DCT}[k] = \sum_{n=0}^{N/2-1} r_k[n]\cdot \cos(\frac{2\pi kn}{N})$$
$$= \sum_{n=0}^{N/2-1} r_k[N/2-1-n]\cdot \cos(\frac{2\pi k(N/2-1-n)}{N})$$
$$= (-1)^k \sum_{n=0}^{N/2-1} r_k[N/2-1-n]\cdot \cos(\frac{2\pi k(n+1)}{N})$$
$$= (-1)^k \cdot g_{N/2-1}(k), \quad (5)$$

where $g_{N/2-1}(k) = \sum_{n=0}^{N/2-1} r_k[N/2-1-n]\cdot \cos(\frac{2\pi k(n+1)}{N})$.

Let $\theta_k = \frac{2\pi k}{N}$, and $g_{N/2-1}(k)$ can be generalized as

$$g_i(k) = \sum_{n=0}^{i} r_k(i-n)\cdot \cos((n+1)\theta_k). \quad (6)$$

It is known that Chebyshev polynomials are well defined as

$$\cos(r\theta) = 2\cos((r-1)\theta)\cdot\cos\theta - \cos((r-2)\theta), \quad (7)$$
$$\sin(r\theta) = 2\sin((r-1)\theta)\cdot\cos\theta - \sin((r-2)\theta). \quad (8)$$

Using the recursive identity stated in (7), Eq. (6) can be deduced as

$$g_i(k) = \sum_{n=0}^{i} r_k[i-n].\cos((n+1)\theta_k)$$
$$= \sum_{n=0}^{i} r_k[i-n]\cdot\{2\cos(n\theta_k)\cdot\cos\theta_k - \cos((n-1)\theta_k)\}$$
$$= 2\sum_{n=0}^{i} r_k[i-n]\cdot\cos(n\theta_k)\cdot\cos\theta_k - \sum_{n=0}^{i} r_k[i-n]\cdot\cos((n-1)\theta_k)$$
$$= 2r_k[i]\cdot\cos\theta_k + 2\sum_{n=0}^{i-1} r_k[i-1-n]\cdot\cos((n+1)\theta_k)\cdot\cos\theta_k$$
$$- r_k[i]\cdot\cos\theta_k - r_k[i-1] - \sum_{n=0}^{i-2} r_k[i-2-n]\cdot\cos((n+1)\theta_k)$$
$$= r_k[i]\cdot\cos\theta_k - r_k[i-1] + 2\cos\theta_k\cdot g_{i-1}(k) - g_{i-2}(k). \quad (9)$$

The z-transform of Eq. (9) can be denoted as

$$\frac{g(k,z)}{r_k(z)} = \frac{\cos\theta_k - z^{-1}}{1 - 2\cos\theta_k z^{-1} + z^{-2}}. \quad (10)$$

For DST part in Eq. (4), by letting $s_k[n] = x[n] - W_N^{-k}\cdot x[N-1-n]$ and replacing n by $N/2\text{-}1\text{-}n$, $y_{DST}[k]$ can be derived in similar behavior as

$$y_{DST}[k] = -\sum_{n=0}^{N/2-1} s_k[n]\cdot\sin(\frac{2\pi kn}{N})$$
$$= -\sum_{n=0}^{N/2-1} s_k[N/2-1-n]\cdot\sin(\frac{2\pi k(N/2-1-n)}{N})$$
$$= (-1)^k \sum_{n=0}^{N/2-1} s_k[N/2-1-n]\cdot\sin(\frac{2\pi k(n+1)}{N})$$
$$= (-1)^k \cdot \sum_{n=0}^{N/2-1} s_k[N/2-1-n]\cdot\sin((n+1)\theta_k)$$
$$= (-1)^k \cdot h_{N/2-1}(k), \quad (11)$$

where $h_{N/2-1}(k) = \sum_{n=0}^{N/2-1} s_k[N/2-1-n]\cdot\sin((n+1)\theta_k)$.

Applying Eq. (8), $h_{N/2-1}(k)$ can be generalized as

$$h_j(k) = \sum_{n=0}^{j} s_k[j-n]\cdot\sin((n+1)\theta_k)$$
$$= \sum_{n=0}^{j} s_k[j-n]\cdot\{2\sin(n\theta_k)\cdot\cos\theta_k - \sin((n-1)\theta_k)\}$$
$$= 2\sum_{n=0}^{j} s_k[j-n]\cdot\sin(n\theta_k)\cdot\cos\theta_k - \sum_{n=0}^{j} s_k(j-n)\cdot\sin((n-1)\theta_k)$$
$$= 2\sum_{n=0}^{j-1} s_k[j-1-n]\cdot\sin((n+1)\theta_k)\cdot\cos\theta_k + s_k[j]\cdot\sin\theta_k$$
$$- \sum_{n=0}^{j-2} s_k[j-2-n]\cdot\sin((n+1)\theta_k)$$
$$= s_k[j]\cdot\sin\theta_k + 2\cos\theta_k\cdot h_{j-1}(k) - h_{j-2}(k). \quad (12)$$

The z-transform of Eq. (12) can be denoted as

$$\frac{h(k,z)}{s_k(z)} = \frac{\sin\theta_k}{1-2\cos\theta_k z^{-1}+z^{-2}}. \quad (13)$$

Eqs. (10) and (13) can be easily mapped into the structures as shown in Fig. 1(a) and (b), respectively. Compared with the conventional architectures [4, 7], it is obviously found that the computation cycles can be achieved to the reduction of 50% via the proposed algorithm and architecture.

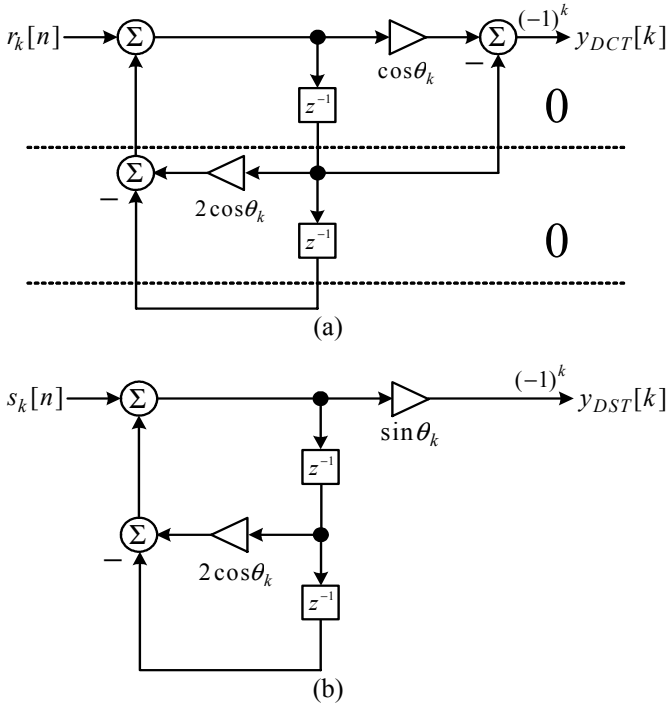

Fig. 1. Block diagram of low-computation cycle for (a) DCT part and (b) DST part of DFT computation.

For high-speed issue, we adopt the register-splitting scheme (i.e., one kind of retiming schemes) to reduce the critical path. Herein, we define two useful notations 0 and 1, where 0 and 1 indicate that the delay elements as shown in Fig. 1(a) 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 DCT part of the core type design in Fig. 1(a) can be represented as 00. In this case, there are four combinations as listed in Table 1, where T_m and T_a denote the operation time required for one real multiplication and one real addition, respectively. With minimum critical period and fewest registers in mind, we select the 10 register-splitting structure for DCT part. It is worthy of emphasizing that 10 and 11 as listed in Table 1 result in the same DCT part as depicted in the upper diagram of Fig. 2, where 1<= denotes a hardwired shifter with one-bit left shift. Similarly, DST part can be modified as the lower diagram of Fig. 2. In order to remain the minimum critical period for the recursive DFT computation, the forward pipeline register, ▨, is exploited for the final sum output. Later combining these two new parts into one, a recursive DFT architecture that possesses low computation cycle and higher speed than the conventional DFT structures can be obtained.

Table 1: Combinations of Register-Splitting for DCT Part

Com.	00	01	10	11
Period	$2T_m + 3T_a$	$2T_m + 3T_a$	$T_m + 2T_a$	$T_m + 2T_a$
# of Reg.	2	2	2	2
Opt.	No	No	Yes	Yes

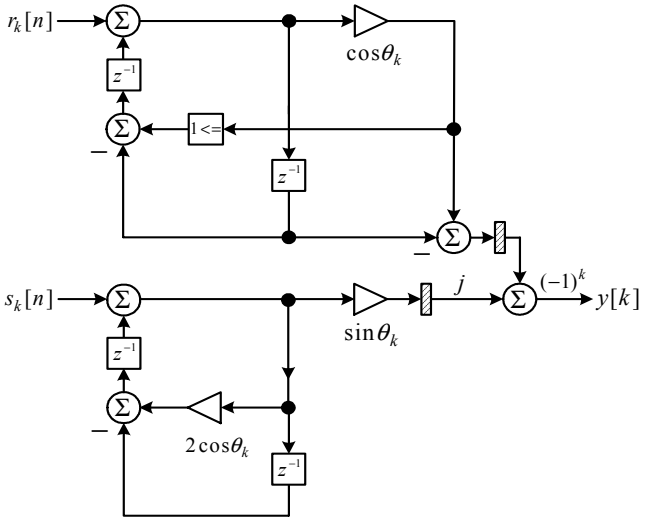

Fig. 2. Block diagram of the proposed low-computation cycle and high-speed recursive DFT architecture.

III. NEW RECURSIVE FORMULA FOR IDFT

The IDFT of the N-point input $y[k]$ is defined as

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} y[k]\cdot W_N^{-kn}, \quad (14)$$

To develop recursive IDFT algorithm, Eq. (14) can be recast as

$$\begin{aligned} x[n] &= \frac{1}{N}\sum_{k=0}^{N/2-1} y[k]\cdot W_N^{-kn} + \frac{1}{N}\sum_{k=0}^{N/2-1} y[N-1-k]\cdot W_N^{(k+1)n} \\ &= \frac{1}{N}\sum_{k=0}^{N/2-1}\left(y[k]+W_N^n\cdot y[N-1-k]\right)\cdot\cos\left(\frac{2\pi kn}{N}\right) \\ &\quad + j\cdot\frac{1}{N}\sum_{k=0}^{N/2-1}\left(y[k]-W_N^n\cdot y[N-1-k]\right)\cdot\sin\left(\frac{2\pi kn}{N}\right). \quad (15) \end{aligned}$$

Similarly, Eq. (15) can be treated as IDCT and IDST parts, $x_{IDCT}(n)$ and $x_{IDST}(n)$, respectively, as

$$x_{IDCT}[n] = \frac{1}{N}\sum_{k=0}^{N/2-1}\left(y[k]+W_N^n\cdot y[N-1-k]\right)\cdot\cos\left(\frac{2\pi kn}{N}\right), \quad (16)$$

and

$$x_{IDST}[n] = \frac{1}{N}\sum_{k=0}^{N/2-1}\left(y[k]-W_N^n\cdot y[N-1-k]\right)\cdot\sin\left(\frac{2\pi kn}{N}\right). \quad (17)$$

In Eq. (16), we can define $r_n[k] = y[k] + W_N^n \cdot y[N-1-k]$. Replacing k by $N/2\text{-}1\text{-}k$, Eq. (16) can be rewritten as

$$x_{IDCT}[n] = \frac{1}{N}\sum_{k=0}^{N/2-1} r_n[k]\cdot\cos(\frac{2\pi kn}{N})$$

$$= \frac{1}{N}\sum_{k=0}^{N/2-1} r_n[N/2-1-k]\cdot\cos(\frac{2\pi n(N/2-1-k)}{N})$$

$$= \frac{(-1)^n}{N}\sum_{k=0}^{N/2-1} r_n[N/2-1-k]\cdot\cos(\frac{2\pi n(k+1)}{N})$$

$$= \frac{(-1)^n}{N}\cdot g_{N/2-1}(n), \quad (18)$$

where $g_{N/2-1}(n) = \sum_{k=0}^{N/2-1} r_n[N/2-1-k]\cdot\cos(\frac{2\pi n(k+1)}{N})$.

Let $\theta_n = \frac{2\pi n}{N}$, and $g_{N/2-1}(n)$ can be generalized as

$$g_i(n) = \sum_{k=0}^{i} r_n(i-k)\cdot\cos((k+1)\theta_n). \quad (19)$$

Using the recursive identity stated in (7), Eq. (19) can be deduced as

$$g_i(n) = \sum_{k=0}^{i} r_n[i-k]\cdot\cos((k+1)\theta_n)$$

$$= \sum_{k=0}^{i} r_n[i-k]\cdot\{2\cos(k\theta_n)\cdot\cos\theta_n - \cos((k-1)\theta_n)\}$$

$$= r_n[i]\cdot\cos\theta_n - r_n[i-1] + 2\cos\theta_n\cdot g_{i-1}(n) - g_{i-2}(n). \quad (20)$$

The z-transform of Eq. (20) can be denoted as

$$\frac{g(n,z)}{r_n(z)} = \frac{\cos\theta_n - z^{-1}}{1-2\cos\theta_n z^{-1}+z^{-2}}. \quad (21)$$

For IDST part in Eq. (17), by letting $s_n[k] = y[k] - W_N^n \cdot y[N-1-k]$ and replacing k by $N/2\text{-}1\text{-}k$, $x_{IDST}[n]$ can be derived in similar behavior as

$$x_{IDST}[n] = \frac{1}{N}\sum_{k=0}^{N/2-1} s_n[k]\cdot\sin(\frac{2\pi kn}{N})$$

$$= \frac{1}{N}\sum_{k=0}^{N/2-1} s_n[N/2-1-k]\cdot\sin(\frac{2\pi n(N/2-1-k)}{N})$$

$$= \frac{-(-1)^n}{N}\sum_{k=0}^{N/2-1} s_n[N/2-1-k]\cdot\sin((k+1)\theta_n)$$

$$= \frac{-(-1)^n}{N}\cdot h_{N/2-1}(n), \quad (22)$$

where $h_{N/2-1}(n) = \sum_{k=0}^{N/2-1} s_n[N/2-1-k]\cdot\sin((k+1)\theta_n)$.

Applying Eq. (8), $h_{N/2-1}(n)$ can be generalized as

$$h_j(n) = \sum_{k=0}^{j} s_n[j-k]\cdot\sin((k+1)\theta_n)$$

$$= \sum_{k=0}^{j} s_n[j-k]\cdot\{2\sin(k\theta_n)\cdot\cos\theta_n - \sin((k-1)\theta_n)\}$$

$$= s_n[j]\cdot\sin\theta_n + 2\cos\theta_n\cdot h_{j-1}(n) - h_{j-2}(n). \quad (23)$$

The z-transform of Eq. (23) can be denoted as

$$\frac{h(n,z)}{s_n(z)} = \frac{\sin\theta_n}{1-2\cos\theta_n z^{-1}+z^{-2}}. \quad (24)$$

After using the register-splitting scheme, Eqs. (21) and (24) can be easily mapped into the modified structures as shown in Fig. 3. Again, from the proposed algorithm and architecture, it is obviously found that the computation cycles can be achieved to the reduction of 50%.

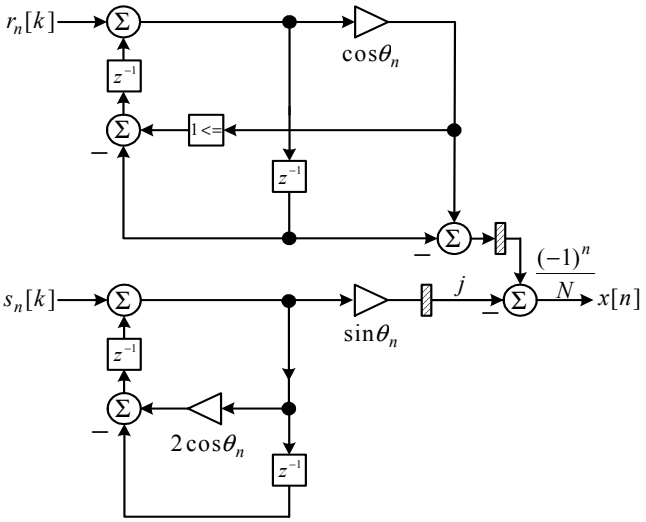

Fig. 3. Block diagram of the proposed low-computation cycle and high-speed recursive IDFT architecture.

IV. FOLDED ARCHITECTURE FOR DFT/IDFT

Recently, there are many modern applications, which need larger frame size to achieve the higher frequency resolution [1, 2]. However, the strict specification would make the implementation more difficult to meet the requirement. In order to further reduce the number of computation cycles for N-point DFT/IDFT, regarding the recursive processing kernel of the core-type design in Fig. 2 as a processing element (PE), we can regularly construct the folded recursive DFT (RDFT) structure as shown in Fig. 4. The whole architecture consists of a data buffer, a control unit and the number of $N/2$ RDFT units. The control unit not only plays the role of a sequence controller but also a parameter controller, which feed the proper coefficients for the RDFT units. The RDFT unit consists of one pre-processing unit and one recursive PE, where the pre-processing unit provides the intermediary data s_k and r_k to

the following recursive PE. The final output comes out from the $N/2$ recursive PEs in parallel. Based on the proper scheduling, the data streams can be processed continually every clock cycle without extra computation latency. Consequently, we can keep the maximum throughput rate for N-point DFT/IDFT in N computation cycles.

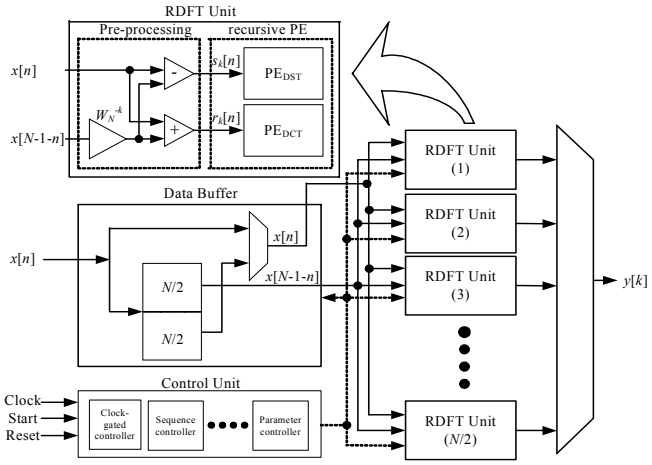

Fig. 4. Block diagram of the proposed folded-type recursive DFT architecture.

V. COMPARISONS AND DISCUSSIONS

In this section, we give a comprehensive comparison result as listed in Table 2 in terms of the number of computation cycles for each DFT/IDFT output as well as N-point DFT/IDFT calculation, the critical period, and the number of real multipliers. Note that the operation time of the complex multiplication requires $T_m + T_a$. Our proposed work 1 (i.e., core-type design) based on Chebyshev polynomial can save half computation cycles for each DFT/IDFT output compared with the existing works [4, 7] at the expense of slightly increased area cost. Comparing with the results of the recursive algorithm in [8] which, for example, requires 2794 computational cycles to obtain all 64-point DFT outputs, the proposed core-type architecture requires 2048 computational cycles. In other words, our proposed work 1 has the lowest computation cycles among existing structures [4, 7, 8]. Due to applying register-splitting scheme, the proposed one has the higher speed than the recursive structures of [4, 8] and possesses the same operation frequency as that of our previous work [7]. However, considering the hardware complexity, the proposed core type DFT/IDFT architecture requires two more multipliers than the previously proposed one [7]. Consequently, the proposed methods have many advantages over the conventional proposed algorithms. Furthermore, based on the proposed work 1, we can construct a folded recursive DFT/IDFT architecture. The folded architecture can significantly reduce the number of computation cycles for N-point DFT/IDFT from $N^2/2$ to N. Thus, more real-time operation can be achieved. Therefore, in Table 2, it reveals that our proposed architectures have characteristics of the lowest computation cycle and high speed.

Concerning the chip implementation, it is worth noticing that the recursive PE mainly dominates the performance of the whole architecture. Hence, we are encouraged to design one efficient recursive PE that takes into account of speed, area, and easy mutual interconnection. For the purpose of further reducing the critical path, the complex multiplier operations exploit the shift-and-add arithmetic. The active chip layout area of the proposed recursive PE as shown in Fig. 5 is 515 um x 515 um in TSMC 0.13 um CMOS process. The critical delay time obtained from the static timing analysis (STA) of Synopsys is 11.32 ns under the worst-case condition. It is expected that the folded type architecture only needs 0.72 μs to complete 64-point DFT/IDFT operation, i.e., 64 cycles. That means that we can meet the timing specification of the IEEE 802.11a standard [2]. Table 3 summarizes the chip characteristics of proposed recursive PE for DFT/IDFT structure.

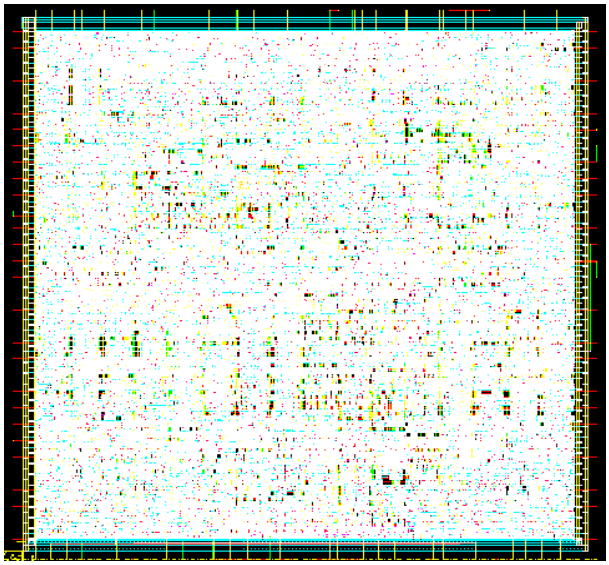

Fig. 5. Recursive PE layout for DFT/IDFT architecture.

Table 3: Chip Characteristics of the Proposed Recursive PE

DFT Length (N)	64 points
Input Word Length (w)	16 bits
Critical Delay Time	11.32 ns
Active Chip Area	515 um x 515 um
Process Technology	TSMC 0.13 um CMOS

VI. CONCLUSIONS

Two new recursive DFT/IDFT architectures based on the hybrid of Chebyshev polynomial and register-splitting scheme are devised in this paper. The analyzed results

expose that the proposed VLSI algorithm leads to the fewest computation cycle and higher speed than others. In addition, the proposed folding recursive architecture with regular organization is certainly amenable to VLSI implementation.

References

[1] R. D. J. van Nee and R. Prasad, *OFDM for wireless multimedia communications.* Norwood, MA: Artch House, 2000.

[2] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM," *IEEE Trans. of Solid-State Circuits,* vol. 39, no. 3, pp. 484-493, Mar. 2004.

[3] 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.

[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] L. D. Van and C. C. Yang, "High-speed area-efficient recursive DFT/IDFT architectures," in *Proc. IEEE Int. Symp. Circuits Syst.,* May 2004, vol. 3, pp. 357-360, Vancuover, Canada.

[8] C. P. Fan and G. A. Su, "Novel recursive discrete Fourier transform with compact architecture," *IEEE Asia-Pacific Conf. Circuits Syst.*, Dec. 2004, pp. 1081-1084, Tainan, Taiwan.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] C. H. Chen, B. D. Liu, J. F. Yang, and J. L. Wang, "Efficient Recursive Structures for forward and inverse discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 52, pp. 2665-2669, Sep. 2004.

[16] 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.

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

Parameters	Second Order DFT/IDFT	Core Type DFT/IDFT of ISCAS2004 [7]	Proposed Work 1 (Core Type)	Proposed Work 2 (Folded Type)
# of Computation Cycles for Each $y[k]$ or $x[n]$	N	N	$N/2$	$N/2$
# of Computation Cycles for N-Point DFT/IDFT	N^2	N^2	$N^2/2$	N
Critical Period	$T_m + 3T_a$	$T_m + 2T_a$	$T_m + 2T_a$	$T_m + 2T_a$
# of Real Multipliers	6	4	6 (Pre-processing Excluded)	$3N$ (Pre-processing Excluded)