# Novel Shared Multiplier Scheduling Scheme for Area-Efficient FFT/IFFT Processors

S.V.V.Sudhakar[1], R.Radha Kumari[2]
[1] PG Scholar, St. Ann's College of Engineering & Technology, Chirala, A.P, India
[2] Associate Professor, St. Ann's College of Engineering & Technology, Chirala, A.P, India

***Abstract*- This paper based on a new shared multiplier scheduling scheme (SMSS) for area-efficient fast Fourier transform (FFT)/inverse FFT processors. SMSS can significantly reduce the total number of complex multipliers up to 28%. The proposed mixed-radix multipath delay commutator processors can support128/256 and 256/512-point FFTs using SMSS. The proposed processors have been designed and implemented with 90-nm CMOS technology, which can reduce the total hardware complexity by 20%. The proposed processors having eight-parallel data paths can achieve a high throughput rate up to 27.5 GS/s at 430 MHz. In addition, the proposed processors can support any FFT size using additional stages.**

***Index Terms*- Fast Fourier transform (FFT), mixed-radix multipath delay commutator (MRMDC)**

## 1. INTRODUCTION FAST FOURIER TRANSFORMS (FFT)

The Fourier transform is the method of changing time representation to frequency representation. The discrete Fourier transform (DFT) is a one of the Fourier transform, used in Fourier analysis. It transforms one function that is time into another that is frequency, so as to get discrete signals, hence called the DFT, of the original function. The DFT of a given sequence x[n] can be computed using the formula

$$X(k) = \sum_{n=0}^{n-1} x(n) w_N^{kn} , 0 \le k \le N-1$$

$W_N = e^{-j2\pi/N}$

Where, $W_N$ is twiddle factor. Twiddle factors referred to as the root of- unity complex multiplicative constants in the butterfly operations of the FFT algorithm, used to recursively combine smaller discrete Fourier transforms. Practically, at the input there is time domain so only real values should be present But for our convenience we apply input data sequence x(n) having both real and imaginary term We observe that for each value of k, direct computation of X(k) involves N complex multiplications (4N real multiplications) and N-1 complex additions (4N-2 real additions). Consequently, to compute all N values of the DFT requires N 2 complex multiplications and N 2-N complex additions.

The Discrete Fourier transform is used to produce frequency analysis of discrete non periodic signals. The FFT is another method of achieving the same result, but with less overhead involved in the calculations. Transforms basically convert a function from one domain to another with no loss of information. Fourier Transform converts a function from the time (or spatial) domain to the frequency domain. The mathematical formula used for the Fourier transform is as follows

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)\, e^{-iwt}\, dt$$

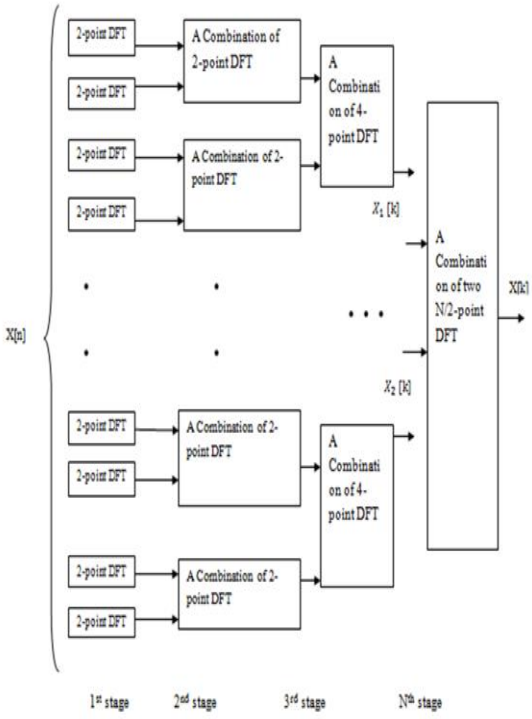
FFT radix-2 algorithm the number of complex multiplications and additions will be reduced to (N/2) log2N and Nlog2N to compute the DFT of a given complex x[n]. Hence in this project the Decimation in Time FFT radix-2 algorithm is implemented to compute the DFT of a sequence.

In summary, the comparison of the computation load is listed as following:

| | Direct Computation load of DFT | | FFT | |
|---|---|---|---|---|
| N | Complex multiplication $N^2$ | Complex Addition N(N-1) | Complex Multiplication (N/2) $log_2 N$ | Complex Addition $Nlog_2 N$ |

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 2 | 1 | 2 |
| 8 | 64 | 56 | 12 | 24 |
| 32 | 1024 | 922 | 80 | 160 |
| 64 | 4096 | 4022 | 192 | 384 |
| 128 | 16384 | 16256 | 448 | 896 |
| $2^{10}$ | 1048576 | 1047522 | 5120 | 10240 |
| $2^{20}$ | $\sim 10^{12}$ | $\sim 10^{12}$ | $\sim 10^{7}$ | $\sim 2 * 10^{7}$ |

In general, an N-point FFT (a radix-2 FFT and N= 2") is evaluated by The radix-2 algorithms are the simplest FFT algorithms. The decimation-in-time (DIT) radix-2 FFT recursively partitions a DFT into two half-length DFTs of the even indexed and odd-indexed time samples. The outputs of these shorter FFTs are reused to compute many outputs, thus greatly reducing the total computational cost. The radix-2 decimation-in-time and decimation-in-frequency fast Fourier transforms (FFTs) are the simplest FFT algorithms. Like all FFTs, they gain their speed by reusing the results of smaller, intermediate computations to compute multiple DFT frequency outputs.




2. SYSTEM DESIGN: EXISTING SYSTEM

In this section, we show the existing 128/256-point MRMDC FFT/IFFT processor to derive known FFT architectures in general. Fig. 1 shows the eight-parallel 128/256-point FFT/IFFT architecture, which consists of BUs, delay commutators, and twiddle factor multipliers. The input sequence of the ith OFDM symbol is split in eight-parallel data paths, where j stands for the OFDM symbol index. In the first stage, the radix-2/4 BU can perform one radix-4 or two radix-2 operations to compute the 128- and 256-point FFTs. The second and third stages employ a modified radix-8 BU proposed which is suitable for the pipelined structure.

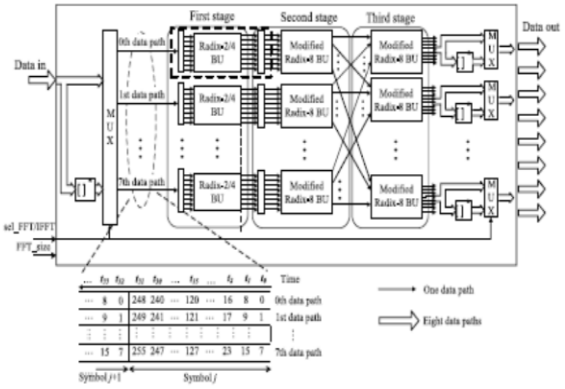



*Fig.1. Eight-parallel 128/256-point MRMDC FFT/IFFT processor*

The operation of the FFT or IFFT is selected by the control signal, sel_FFT/IFFT. To perform the IFFT computation, the signs of the imaginary parts in the input and output sequences are changed using complex conjugate operations.

The input and output sequences have specified orders, as shown in Fig. 1. There are three stages based on the radix-2, radix-4, and modified radix-8 algorithms for the 128/256-point FFT. The input sequence of the processor is split into eight streams for eight data paths, as shown at the bottom of Fig. 1, and can be expressed as follows

$$in1(k, p) = x(n)$$

where in1(k, p) represents the kth input data for the pth data path of the first stage. k and p are denoted as follows:

$$n = k \times 8 + p, p = 0, 1. . . 7.$$

A. *First Stage for 256-Point FFT*

Even though Rabiner and Gold [37] did not show the MRMDC architecture, the architecture shown in Fig. 2 can be inferred from the MDC architecture. Fig. 2 shows the first and second stages of the existing 256-point MRMDC architecture which are represented by the dotted box shown in Fig. 1. The first and second stages consist of the input buffer, radix- 2/4 BU,

modified radix-8 BU in [36], and commutator. The existing radix-8 BU in [38] consists of seven phases and requires a long critical path because the complex multipliers are located in two phases; hence, it has the long critical path delay that is 2tmul + 3tadd, where tmul and tadd are the delays of the complex multiplier and adder, respectively. To reduce the critical path, the odified radix-8 BU proposed, is more suitable for efficient implementation. Reference [36] has incorporated twiddle factors and adder tree matrices into a single phase of calculation. In the modified radix-8 BU, the first phase has 11 complex multipliers, and the other three phases consist of adder trees, and thus, the modified radix-8 BU can have the critical path delay of tmul + 3tadd. Therefore, the modified radix-8 BU can reduce the critical path delay by tmul. The FFT/IFFT processor in Fig. 1 also uses the modified radix-8 BUs used to implement high-speed FFT processors.

In the existing MRMDC architecture [37] in Fig. 2, the input sequence of the zeroth data path is split into eight data streams from A to H. The four input sequences of the upper BU include streams A, C, E, and G, and those of the lower BU include streams B, D, F, and H. In Fig. 2, indexed input samples in the horizontal arrive at the same stream at different time instants, whereas input samples in the vertical arrive at the same time from different streams. All the data streams are delayed by the delay elements, Di, to maintain the proper cycles, where i is the size of the delay element. For the first four cycles from t0 to t3, the input stream A of the zeroth data path, x(0), x(8), x(16), and x(24), is fed and stored in D28. The input stream B is fed and stored in D24 for the next four cycles. Every four cycles, the input sequence is switched to the next stream. The input samples in the vertical arrived at the same time from different streams in the upper and lower BUs in Fig. 2 can be formulated as in1,u (m) = {x(8m), x(8m + 64), x(8m + 128), x(8m + 192)} in1,l (m) = {x(8m + 32), x(8m + 96), x(8m + 160), x(8m + 224)} (10)

where m is from 0 to 3. in1,u(m) and in1,l (m) are the input samples in the upper and lower BUs, respectively. m is 0 when the computation period begins at t28, and m is incremented by one for each cycle. For example, at t28, m is 0, and then, the input data for the upper BU consist of {x(0), x(64), x(128), x(192)}, and the input data for the lower BU consist of {x(32), x(96), x(160), x(224)}. In the next cycle, t29, m becomes 1, and the input data for the upper BU are {x(8), x(72), x(136), x(200)}, and those of the lower BU are {x(40), x(104), x(168), x(232)}. As shown in Fig. 2, the FFT processor using eight-parallel data paths takes 28 cycles for reordering the radix-2/4 butterfly computation during the idle period from t0 to t27. After the idle period, the radix-2/4 BU is active during the computation period from t28 to t31, as shown in Fig. 2.

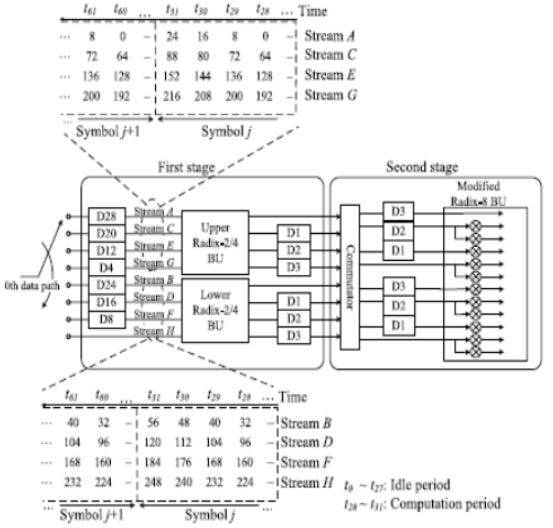



*Fig. 2. Existing 256-point FFT structure using the modified radix-8 BU*

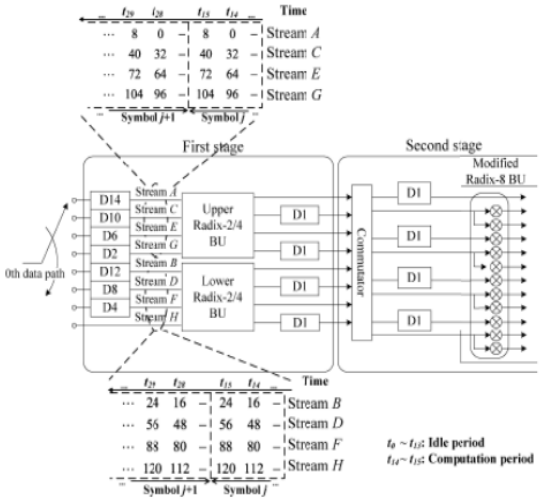



*Fig. 3. First stage of the 128-point MRMDC FFT*

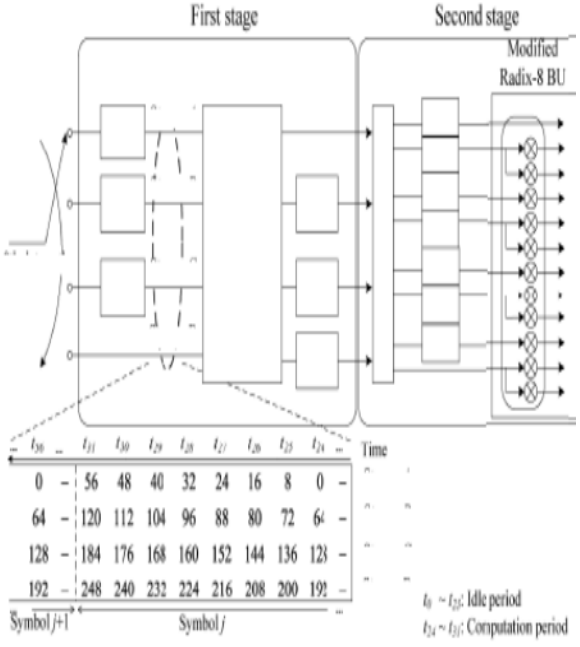
B. *First Stage for 128-Point FFT*

Fig. 3 shows the first stage of the existing 128-point MRMDC architecture, which is represented by the dotted box shown in Fig. 1. The processor can support the 128-point FFT/IFFT in a similar manner

to compute the 256-point FFT described in Section III-A. In the existing MRMDC, the input sequence of the eight parallel data paths is split into eight data streams. Further, each data stream is delayed by the delay elements. The subsequences in the vertical arrived at the same time from different streams in the upper and lower BUs in Fig. 3 can be formulated as in1,u (m)={x(8m), x(8m+32), x(8m+64), x(8m+96)} in1,l (m)={x(8m+16), x(8m+48), x(8m+80), x(8m + 112)} (11) where m is 0 or 1. in1,u(m) and in1,l(m) are the input samples in the upper and lower BUs, respectively. In contrast to the 256-point FFT in Fig. 2, the number of delay elements for input reordering in the 128-point FFT decreases by half. To reorder the input sequences for the first stage, the existing structure in Fig. 3 takes 14 cycles using D2, D4, D6, D8, D10, D12, and D14 during the idle period and two cycles during the computation period when the FFT size is 128. Therefore, the first stage of the existing processor requires 16 cycles.

Proposed System

This section proposes novel eight-parallel MRMDC FFT/IFFT processors that offer a high throughput and low hardware complexity using SMSS. In contrast with the existing processor [37], the first stage uses the shared multipliers and the second stage requires the modified radix-8 BUs without complex multipliers (w/o mul), as shown in Fig. 4.



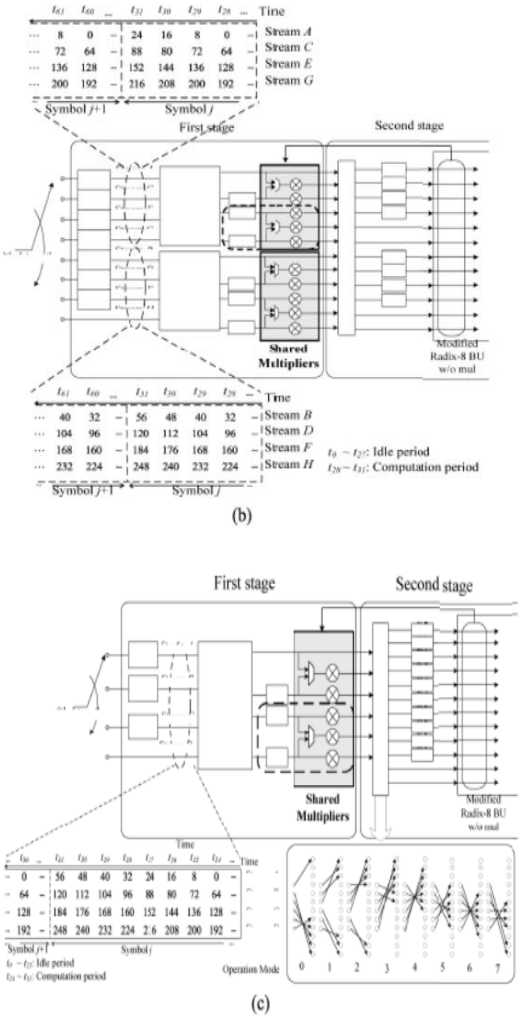




*Fig. 4. Proposed first stage of the zeroth data path for the 256-point FFT. (a) Reducing the number of butterflies (Type I). (b) Reducing the number of complex multipliers (Type II). (c) Architecture employing the scheduling scheme (Type III).*

A. *First Stage for 256-Point FFT*

Fig. 4 shows three proposed architecture types. Type I in Fig. 4(a) can reduce the number of radix-2/4 BUs from two in Fig. 2 to one in the first stage without additional clock cycles. Type II in Fig. 4(b) uses SMSS, as described in the next paragraphs, and thus, it can reduce the number of complex multipliers from 11 to 10 compared with the existing architecture. In addition, Type III shown in Fig. 4(c) uses SMSS in Type II and one radix-2/4 BU in Type I, and it can reduce the number of complex multipliers from 11 to 5 in the second stage.

In the Type I architecture shown in Fig. 4(a), the input sequence of the zeroth data path is split into four data streams A, B, C, and D. For the first eight cycles, the input stream A of the zeroth data path [i.e., x(0), x(8), . . . , x(56)] is fed and stored in D24. For the next eight cycles, the input stream B is fed and stored in D16. Unlike the existing architecture in Fig. 2, where the input sequence is switched to the next stream every four cycles, the input sequence of Type I is switched to the next stream every eight cycles. Type I takes 24 cycles to reorder the input data of the radix-2/4 BU and eight cycles to perform the radix-2/4 operations shown in the bottom of Fig. 4(a). The total computation in the first stage for one symbol can be completed in 32 cycles. Therefore, both the existing architecture shown in Fig. 2, and Type I require the same number of clock cycles, i.e., 32 cycles. Therefore, the proposed FFT/IFFT processor in the first stage can reduce the radix-2/4 BUs from two in the existing architecture (Fig. 2) to one in Type I [Fig. 4(a)] without any additional clock cycles compared with the existing architecture.

B. *First Stage for 128-Point FFT*

The proposed FFT processor can support the 128-point FFT/IFFT in a similar manner to compute the 256-point FFT described in Section IV-A. As shown in Fig. 7, the proposed structure reduces the number of BUs from two to one in the first stage compared with the existing structure in Fig. 3. In Fig. 7, the input sequence of each data path is split into four data streams, and it takes 12 cycles using D4, D8, and D12 to start the first butterfly computation and four cycles to perform the radix-2/4 operations. To finish the radix-2 computation using one radix-2/4 BU, the proposed structure requires four cycles. Therefore, the first stage of the proposed processor also requires 16 cycles, even with one radix-2/4 BU. Thus, the structure consisting of one radix-2/4 BU in the first stage can reduce the hardware complexity without increasing the number of clock cycles compared with the existing architecture. In the first stage in Fig. 7, the radix-2/4 BU can perform two radix-2 butterfly computations.

The proposed structure performs complex multiplications for the second stage before the delay commutator using the shared multipliers. The commutator is configured by the operation mode. In the 128-point FFT, the operation mode number is calculated by t modulo four. The commutator operates in four different operation modes for performing the 128-point FFT. Fig. 8 shows the proposed first stage in the dotted box shown in Fig. 1, which consists of the input buffer, butterfly processing element, and commutator. In the first stage, the input sequence of each data path is divided into four data streams (A, B, C, and D), which are delayed by the delay elements to synchronize proper cycles. The butterfly operations in the first stage are performed by four data streams. The output data of the first stage are delivered to the second stage through the delay elements and the delay commutator by the operation modes in Figs. 4(c) and 7.

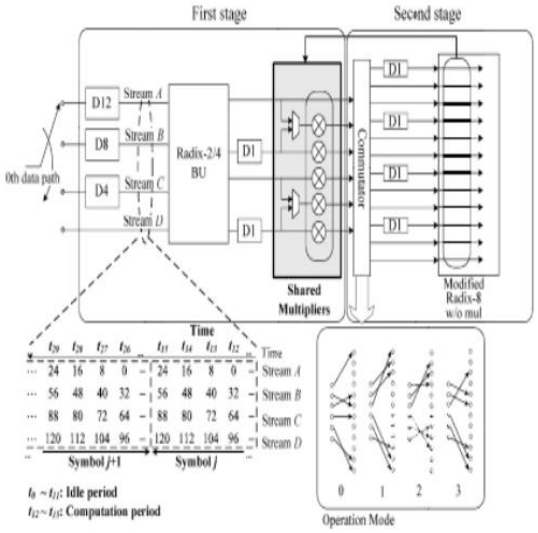



*Fig.7. Proposed first stage employing the scheduling scheme Of the 128-point FFT.*

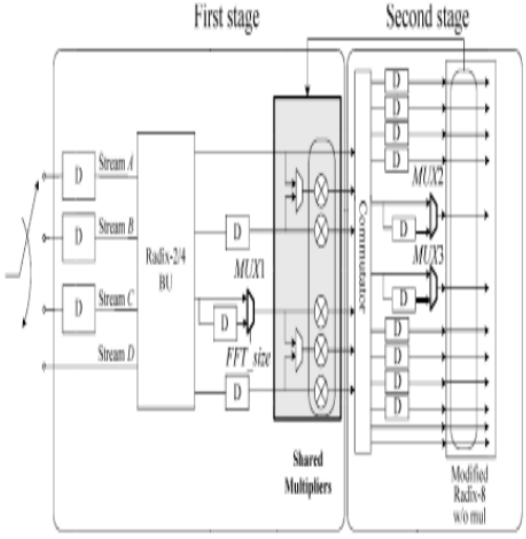



*Fig 8.Structure of the proposed first stage for the 128/256- point FFT processor*

C. *Second and Third Stage Structure*

The input data multiplied by the appropriate twiddle factors are fed into the second stage for the radix-8 butterfly operation through the delay commutator, as shown in Fig. 1. In the second stage, the remaining radix-8 calculation without multiplications is performed because all the multipliers of the second stage in the existing one are moved to the shared multipliers in the proposed first stage. A suitable structure is required to ensure the correction of the FFT output data because the third stage in Fig. 1 is different from the second stage. All the output data generated by the radix-8 butterfly in the second stage are fed to the third stage by a specific order

$$in3(p, l) = out2(l, p) \quad (13)$$

D. *Proposed 256/512-Point FFT/IFFT Processor*

This subsection presents the proposed MRMDC 256/512-point FFT/IFFT processor. Fig. 9 shows the proposed processor that consists of four stages. The radix-2 BUs in the first stage are added to support the 512-point FFT compared with that shown in Fig. 1. The second, third, and fourth stages are the same, as shown in Fig. 1. The processor shown in Fig. 9 performs the 256-point FFT, which is similar to the 256-point FFT in Fig. 4(c). As shown in Fig. 9, the input sequence is split into eight-parallel data paths that are delayed to arrange the input data order in the first stage. Fig. 10 redrawn from the dotted box in Fig. 9 shows one data path of the existing and proposed structures for the 512-point FFT, respectively. As shown in Fig. 10, all the multipliers of the third stage are moved to the shared multipliers in the second stage as we proposed in Type III.

The SMSS shown in Fig. 5 can also be applied to the 512-point FFT. To perform the 512-point FFT, the proposed structure computes twiddle factor multiplications for the second stage using the shared multipliers on each parallel data path. By employing SMSS, the proposed processor can support both the 256- and 512-point FFTs. In addition, the proposed MRMDC can be applied to larger-size FFTs, such as 1024, 2048, and 4096, using additional stages. For example, the 2048-point FFT processor consists of one radix-4 BU, the shared multipliers, one radix-8 BU without multipliers, and two radix-8 BUs.

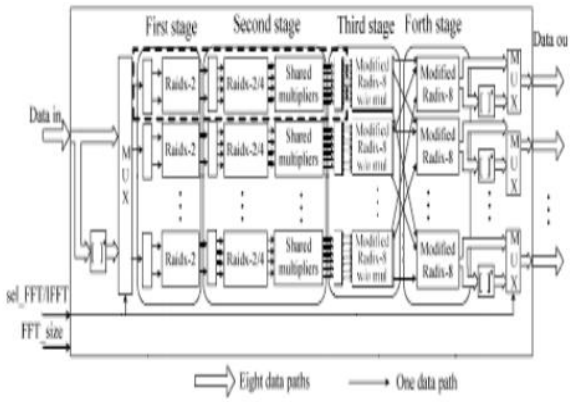



*Fig. 9. Proposed eight-parallel 256/512-point MRMDC FFT/IFFT processor.*

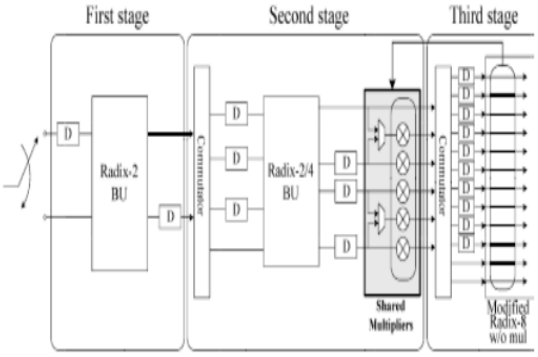



*Fig. 10. Structure of the proposed first and second stages for the 512-point FFT*

## 3. IMPLEMENTATION RESULTS

Assuming that all processors have the same throughput, Table I compares the hardware complexities among the existing and proposed processors. Since the existing processors have been implemented on FPGA, comparing them with the proposed ones may not be meaningful. Hence, as shown in Table I, we consider the number of complex multipliers as they occupy most of the area of the FFT processor .In practice, one stage using radix-2/4 is equivalent to two stages using radix-2. Thus, Table I shows the number of complex adders for two stages using the radix-2 algorithm in and the numbers of complex adders for the first stage employing the radix-2/4 algorithm in and the proposed processors.

As shown in Table I, need 64 and 32 more complex multipliers compared with Type III, respectively. In addition, Type III needs only half the number of adders compared therefore, the existing processors based on radix-2 in may not be suitable for high throughput and low power designs. The existing 128/256-point MRMDC has not been implemented by Rabiner and Gold. For fair comparisons, we implemented the existing and the proposed architectures employing the eight-parallel data paths. In Table I, the existing MRMDC requires 128 adders and 176 complex multipliers.

TABLE I

| | Radix-2 [32] | Radix-$2^4$ [34] | Existing MRMDC [37] | Proposed Type I | Proposed Type II | Proposed Type III |
|---|---|---|---|---|---|---|
| # of multipliers | 192 | 160 | 176 | 176 | 168 | 128 |
| # of adders for the first stage | 128* | 128* | 128 | 64 | 128 | 64 |
| Throughput (R: clock rate) | 64R | 64R | 64R | 64R | 64R | 64R |
| Clock rate | N/A | N/A | 430 MHz | 430 MHz | 430 MHz | 430 MHz |
| Total gate count (excl. ROM) | N/A | N/A | 945 000 (100%) | 907 000 (95.9%) | 920 500 (97.4%) | 760 000 (80%) |

Comparisons of Hardware Complexity for Parallel Pipelined 128/256-Point FFTS

For fair comparisons, Yang et al. introduced the normalized area $A_{Yang}$ as follows:

$$A_{Yang} = \frac{\text{Area}}{(\text{Tech}/0.09\ \mu\text{m})^2 \times P \times \log_2 N}$$

Where P, N, and Tech are the number of the parallel data paths, the FFT size, and the process technology in micrometers, respectively.

To consider the normalized gate count ($G_{norm}$), this paper revises as follows:

$$G_{norm} = \frac{\text{Gate count}}{P \times \log 2\ N}$$

However, the normalized area and gate count do not consider the throughput rate. For a fairer comparison, we consider both the throughput rate and normalized gate count and define the throughput rate-to-gate count ratio (TGR) as follows:

$$TGR = \frac{\text{Throughput rate}}{\text{Gnorm}}$$

In above, a larger value of TGR means a higher throughput and a smaller gate count, which demonstrates the area efficiency. Table IV presents the performance comparisons between the proposed 128/256- and 256/512-point FFT/IFFT processors using Type III and the other existing FFT processors. The existing processors were designed for WPAN standards providing up to 2.6 GS/s, whereas the proposed processors are designed for 25 GS/s throughput required in the O-OFDM standards.

Therefore, the comparisons among the existing and proposed processors may not be meaningful. Therefore, for a fair comparison, we define the TGR to consider the throughput rates, parallel data paths, FFT sizes, and gate counts simultaneously.

## 4. CONCLUSION

This paper proposed area-efficient and high-throughput 128/256- and 256/512-point MRMDC FFT/IFFT processors using a novel scheduling scheme. SMSS can reduce the total number of complex multipliers from 176 to 128, and thus, the hardware complexity of the proposed processor is decreased by 20% when compared with the existing MRMDC.

The performance results show that the proposed Type III architecture can achieve 27.5 GS/s at 430 MHz. Therefore, the proposed processors achieve a value of TGR that is higher than those of the other processors, which can meet the data rates of the high-speed OFDM standards such as O-OFDM and IEEE 802.11ac. In addition, the proposed architectures can apply any FFT size greater than 512 points using additional stages.

## REFERENCES

[1] M. Garrido and K. K. Parhi, "FFT architectures for real-valued signals based on radix-23 and radix-24 algorithms," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 9, pp. 2422–2643, Sep. 2013.

[2] E. Brigham, The Fast Fourier Transform and Its Applications. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[3] W. W. Smith and J. M. Smith, Handbook of Real-Time Fast Fourier Transforms: Algorithms to Product Testing. New York, NY, USA: Wiley, 1995.

[4] M.-H. Cheng, L.-C. Chen, Y.-C. Hung, and C. M. Yang, "A real-time maximum-likelihood heart-rate estimator for wearable textile sensors,"

in Proc. IEEE 30th Ann. Int. Conf. EMBS, Aug. 2008, pp. 254–257.

[5] R. F. Yazicioglu, P. Merken, R. Puers, and C. Van Hoof, "Lowpower low-noise 8-channel EEG front-end ASIC for ambulatory acquisition systems," in Proc. Eur. Solid-State Circuits Conf., Sep. 2006,pp. 247–250.

[6] J.-R. Choi, S.-B. Park, D.-S. Han, and S.-H. Park, "A 2048 complex point FFT architecture for digital audio broadcasting system," in Proc.IEEE Int. Symp. Circuits Syst., May 2000, pp. 693–696.

[7] B. G. Jo and M. H. Sunwoo, "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.

[8] S.-J. Huang and S.-G. Chen, "A green FFT processor with 2.5-GS/s for IEEE 802.15.3c (WPANs)," in Proc. Int. Conf. Green Circuits Syst., Jun. 2010, pp. 9–13.

[9] P.-Y. Tsai, T.-H. Lee, and T.-D. Chiueh, "Power-efficient continuous flow memory-based FFT processor for WiMax OFDM mode," in Proc. Int. Symp. Intell. Signal Process. Commun. Syst., Dec. 2006, pp. 622–625.

[10] P.-Y. Tsai and C.-Y. Lin, "A generalized conflict-free memory addressing scheme for continuous-flow parallel-processing FFT processors with rescheduling," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 12, pp. 2290–2302, Dec. 2011.