An Indexed-Scaling Pipelined FFT Processor for OFDM-Based WPAN Applications

Yuan Chen, *Student Member, IEEE*, Yu-Chi Tsao, Yu-Wei Lin, Chin-Hung Lin, and Chen-Yi Lee, *Member, IEEE*

***Abstract*—In this brief, a high-throughput and low-complexity fast Fourier transform (FFT) processor for wideband orthogonal frequency division multiplexing communication systems is presented. A new indexed-scaling method is proposed to reduce both the critical-path delay and hardware cost by employing shorter wordlength. Together with the mixed-radix multipath delay feedback structure, the proposed FFT processor can achieve very high throughput with low hardware cost. From analysis, it is shown that the proposed indexed-scaling method can save at least 11% memory utilizations compared to other state-of-the-art scaling algorithms. Also, a test chip of a 1.2 Gsample/s 2048-point FFT processor has been designed using UMC 90-nm 1P9M process with a core area of 0.97 mm^2. The signal-to-quantization-noise ratio (SQNR) performance of this test chip is over 32.7 dB to support 16-QAM modulation and the power consumption is about 117 mW at 300 MHz. Compared to the fixed-point FFT processors, about 26% area and 28% power can be saved under the same throughput and SQNR specifications.**

***Index Terms*—Convergent block floating point (CBFP), data scaling, fast Fourier transform (FFT), indexed-scaling, mixed-radix multipath delay feedback (MRMDF), orthogonal frequency-division multiplexing (OFDM), wireless personal area network (WPAN).**

I. Introduction

RECENTLY, researches on gigabit wireless personal area network (WPAN) have attracted great attention due to the increasing demands on high-speed internet access, multimedia streaming, and wireless data bus, etc. The IEEE 802.15.3 Task Group has also been formed to develop a new WPAN solution providing over 2-Gbps data rate (3 Gbps in optional mode). Among the transmission technologies, orthogonal frequency division multiplexing (OFDM) is one of the possible candidates for gigabit WPAN. It has many advantages such as high bandwidth efficiency and excellent multipath immunity compared to other modulation schemes. However, in a gigabit OFDM-based WPAN system, the design of fast Fourier transform (FFT) processors is challenging due to the coexisting requirements of high throughput and accuracy. Although long wordlength can be used in previous high-throughput FFT designs [1]–[3] to achieve better signal-to-quantization-noise ratio (SQNR), both the critical-path delay and hardware cost are increased accordingly. As a result, the overall hardware complexity and power consumption would be very high for the new gigabit FFT processors. Besides, if long-size FFT is chosen for better bandwidth efficiency, even longer wordlength is needed to maintain the same SQNR [4].

As described above, it is demanded to design an efficient FFT processor which satisfies both high throughput and SQNR requirements with shorter wordlength. Among the possible solutions, the scaling-based algorithms have been shown to be effective in improving SQNR performance for FFT processors [4]–[8]. However, the dynamic scaling algorithm [4] is proposed for the single-memory architecture and therefore it is not suitable in high-throughput designs. As for the other scaling-based algorithms [5]–[8], although they can be adopted in the high-throughput pipelined architectures [1]–[3], the induced overhead of storage elements is large which lowers the benefits of wordlength reduction.

In this brief, we propose a new indexed-scaling method which reduces the required wordlength with low storage overhead. Thus, the total memory cost is less than the previous scaling-based algorithms. Besides, the proposed indexed-scaling method can be easily integrated with the mixed-radix multipath delay feedback (MRMDF) structure [1] to build a high-throughput low-complexity FFT processor for gigabit WPAN applications. Both area and power can be greatly saved compared to a fixed-point FFT processor.

II. Scaling Algorithm

A. Review of Previous Scaling Methods

Because of the addition and subtraction operations in FFT computations, the wordlength is increased to avoid possible overflows in a fixed-point FFT design. However, the increased wordlength has many drawbacks in FFT implementations. First, a larger memory is required to store the data which increases both chip area and power consumption. Besides, a longer wordlength results in worse critical-path timing for the arithmetic units (especially for the complex multipliers), which is not preferred in the high-throughput FFT designs.

To solve this problem, many scaling-based algorithms have been proposed for pipelined FFT processors to scale the data dynamically for wordlength reduction [5]–[8]. Unlike the floating-point arithmetic, the scaling-based algorithms use a shared-exponent concept to reduce the size of exponent table. The convergent block floating point (CBFP) algorithm [5], [6] employs intermediate buffers to store the output data, and thus only one exponent is required for each buffer group. Unfortunately, the total size of the intermediate buffers is proportional to the FFT size which causes a large amount of overhead. Also, an additional processing latency is introduced by the buffer delay. Compared

Manuscript received February 16, 2007; revised May 22, 2007, and July 28, 2007. This work was supported by the National Science Council of Taiwan R.O.C. under Grant NSC96-2221- E-009-181 and MediaTek, Inc. This paper was recommended by Associate Editor Y. Lian.

Y. Chen, Y.-C. Tsao, and C.-Y. Lee are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan 30010, R.O.C. (e-mail: ychen@si2lab.org; friendly@si2lab.org; cylee@si2lab.org).

Y.-W. Lin is with MediaTek Inc., Hsinchu, Taiwan 300, R.O.C. (e-mail: yw_lin@mtk.com.tw).

C.-H. Lin is with ICL/ITRI, Chutung, Hsinchu, Taiwan 310, R.O.C. (e-mail: linjh@itri.org.tw).

Digital Object Identifier 10.1109/TCSII.2007.910771

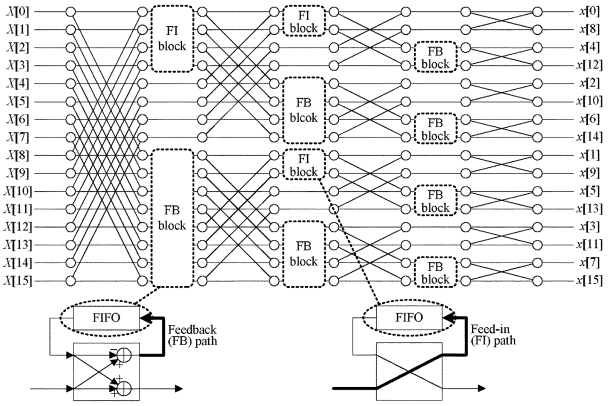


Fig. 1. Data alignment group in a 16-point FFT.

to CBFP, the data scaling algorithm [7] does not need intermediate buffer for exponent sharing. Instead, every data output in each stage has its own exponent. Since the real and imaginary parts of data share the same tailored exponent (for down scaling only), the hardware cost is less than the floating-point arithmetic. Recently, a co-optimization algorithm lying between CBFP and data scaling algorithm is proposed [8]. With small intermediate buffers and fewer exponents, the total size of storage elements for co-optimization method is less compared to CBFP and data scaling algorithms.

B. Proposed Indexed-Scaling Method

Although the scaling-based algorithms [5]–[8] can reduce the required wordlength for pipelined FFT processor effectively, additional storage overhead is still large which lowers the original benefits. Thus, we propose an indexed-scaling method which uses the inherent FIFO of delay-feedback FFT as intermediate buffers for exponent alignment. As shown in Fig. 1, the data in FIFO from both feed-in (FI) and feedback (FB) paths can be aligned to the same exponent. As a result, no additional buffer is required as adopted in the CBFP and co-optimization algorithms. Besides, the FIFO wordlength is kept in constant to reduce the memory cost. Only the transition indexes and the maximum value of data exponents are stored for further scaling and therefore low hardware overhead is achieved. The operation flow of the proposed indexed-scaling method is described in more detail below.

1) Indexed-Scaling Method in I/O Operation: When I/O operation is performed in an FFT stage, the input data from the previous FFT stage (feed-in path) are stored in the FIFO while the pre-stored data in the FIFO are scaled and outputted to the next FFT stage. As shown in Fig. 2, the FIFO outputs with smaller exponents are down scaled to the same exponent by the alignment (ALI) unit which is a variable-bit right shifter with sign-extension. The proposed scaling flow is shown in Fig. 3. As shown in Fig. 3(a), the maximum input exponent is updated in the $\mathbf{EXP}_{\mathrm{FI}}$ and the indexes for a larger input exponent are stored in $\mathbf{IND}_{\mathrm{FI}}$ for future scaling. As for the FIFO output, every output is scaled to $\mathbf{IND}_{\mathrm{FB}}$ according to the exponent change indexes $\mathbf{EXP}_{\mathrm{FB}}$ as depicted in Fig. 3(b). Since all FIFO outputs are scaled to the same exponent $\mathbf{EXP}_{\mathrm{FB}}$ after scaling, the output exponent for the next FFT stage is set to $\mathbf{EXP}_{\mathrm{FB}}$ during the whole I/O operation.

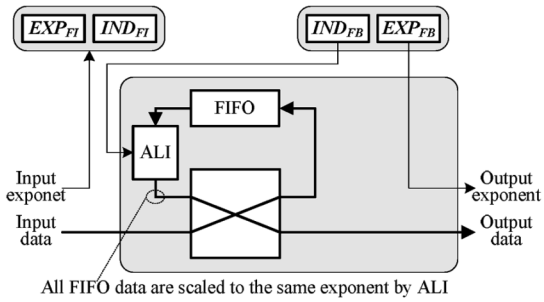


Fig. 2. Block diagram of one FFT stage in I/O operation.

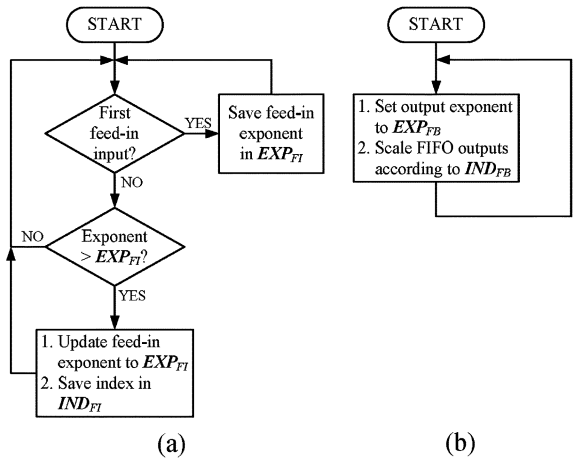


Fig. 3. Flow diagram of the indexed-scaling method in I/O operation. (a) Exponent information storage. (b) Scaling operation.

2) Indexed-Scaling Method in BU Operation: After the FIFO is filled with feed-in data, the butterfly unit (BU) is ready for FFT computations. As illustrated in Fig. 4, the FIFO output is aligned to the input exponent (down scaled if necessary) by the ALI unit for BU processing. The overflow detection and scaling (ODS) units detect the possible overflow and scale the data for BU outputs to maintain the same wordlength. If an overflow occurs, the output data is scaled to half (one-bit right shift) and the output exponent is increased by one. In our ODS design, the output data will always be considered as "overflow" if its input exponent is smaller than the previous output exponent. In this way, incremental exponents can always be assured in both feedback and output paths which simplifies the operation of data alignment. This incremental exponent technique has been proposed in [7] to reduce the number of stored exponents. The scaling flow diagram is shown in Fig. 5. To store the information for follow-up scaling, both the maximum value and the transition indexes of feedback exponents are stored in $\mathbf{EXP}_{\mathrm{FB}}$ and $\mathbf{IND}_{\mathrm{FB}}$ respectively as illustrated in Fig. 5(a) which is similar to the feed-in processing in Fig. 3(a). As for the data-path scaling, the FIFO output data are first scaled to the same exponent $\mathbf{EXP}_{\mathrm{FI}}$ according to the transition indexes $\mathbf{IND}_{\mathrm{FI}}$ and then aligned to the stage inputs. Since $\mathbf{EXP}_{\mathrm{FI}}$ is always smaller than or equal to the input exponent because of the incremental exponents, only down scaling is required. After the BU computation is finished, the overflow detection and scaling together with exponent generation are performed for feedback and output path independently. The scheduling of the proposed

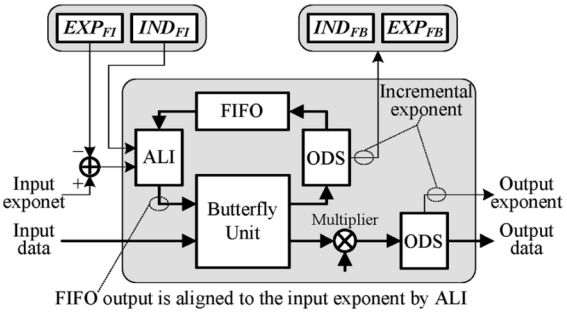


Fig. 4. Block diagram of one FFT stage in BU operation.

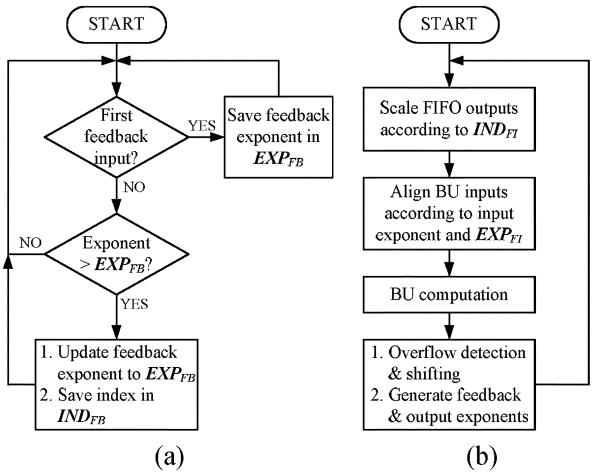


Fig. 5. Flow diagram of the indexed-scaling method in BU operation. (a) Exponent information storage. (b) Scaling operation.

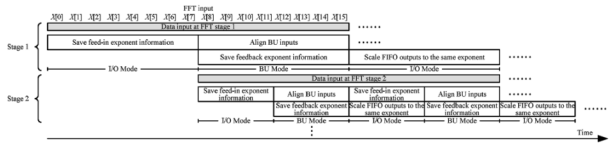


Fig. 6. Operation of the proposed indexed-scaling method for the entire FFT computation.

indexed-scaling method in both I/O and BU operations for the entire FFT stages is summarized in Fig. 6.

III. FFT Architecture

The block diagram of the proposed 2048-point indexed-scaling FFT processor is shown in Fig. 7. To achieve the high-throughput requirement with low hardware cost, both the proposed indexed-scaling method and the MRMDF architecture [1] are exploited in this design. As illustrated in Fig. 7, the 2048-point FFT processor is partitioned into four processing modules using radix-$2^2/2^3$ algorithm [9] to reduce the number of complex multiplications. The scaling controller stores the scaling information for each module and controls the scaling operations according to the proposed scaling method. The detailed description of each module will be given below.

Scaling controller: The scaling controller is responsible for storing the exponent information ($\mathbf{EXP}_{\mathrm{FI}}, \mathbf{EXP}_{\mathrm{FB}}, \mathbf{IND}_{\mathrm{FI}}, \mathbf{IND}_{\mathrm{FB}}$), generating the control signals for each FFT stage, and producing the final exponent outputs. Since multipath delay feedback structure is employed in this design, we design a single scaling controller

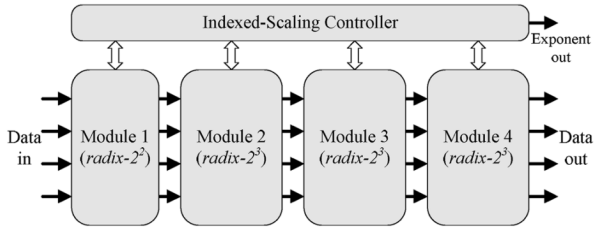


Fig. 7. Block diagram of the proposed 2048-point FFT processor.

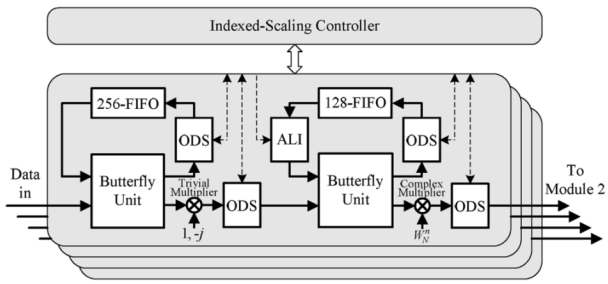


Fig. 8. Block diagram of Module 1.

generating the same control signals for four paths to reduce the hardware cost with negligible SQNR loss. Since the data have the same exponent between each path, it also eliminates the needs for intra-path data alignment for the last two stages in Module 4.

Module 1: Module 1 has a four-path radix-2^2 structure as shown in Fig. 8. Each path consists of two FIFOs, two BUs, a trivial multiplier, a complex multiplier, four ODS units, and an ALI unit. The ODS units perform the overflow detection, one-bit right shifting, and new exponent generation as described earlier. As for the data alignment to the same exponent, it is processed by the ALI unit controlled by the scaling controller. Note that ALI is not adopted in the first stage because the FFT inputs are assumed to have a fixed exponent. Theoretically, the right-shifting range of ALI unit should be increased by one bit every BU stage to support all possible exponents differences between BU inputs which causes high complexity. In realistic FFT computations; however, the required shifting range has an upper bound which can be determined from simulation. In our design, the maximum difference between exponents does not exceed 3 for all stages and thus a simple variable-bit right shifter supporting 0 to 3 bits with sign-extension is sufficient.

Module 2 and Module 3: The block diagram of Module 2 is illustrated in Fig. 9. In Module 2, radix-2^3 algorithm is employed where only two trivial multipliers and one complex multiplier are required. Since it has similar scaling control to Module 1, the control path is omitted in this figure. For Module 3, the structure is almost the same except the FIFO sizes are eight, four, and two for the three BUs.

Module 4: In Module 4, the first FFT stage is the same as the previous modules with one-word FIFO. For the final two FFT stages; however, the four data paths must be merged with each other as shown in Fig. 10 to complete the FFT computation. Since the outputs of the first-stage BUs have the same exponent, overflow-controlled shifters are employed here to replace the ALI units. All four paths are right-shifted by one bit if any of them has an overflow. As a result, the inputs of the second-stage BUs have the same exponent which eliminates the need of data alignment. Beside, the overflow information is transmitted to

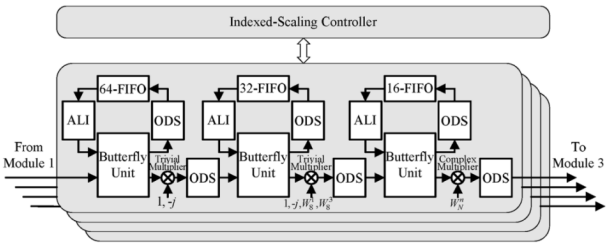


Fig. 9. Block diagram of Module 2.

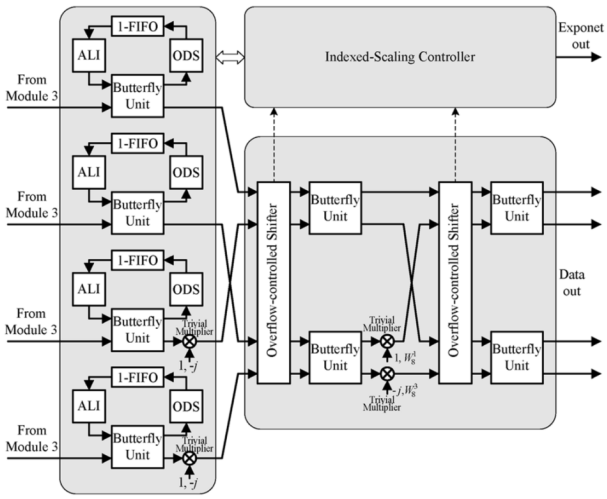


Fig. 10. Block diagram of Module 4.

the scaling controller for generating the final exponent outputs. The operation of the third-stage BUs is almost the same as that of the second-stage BUs.

IV. Analysis and Comparison

The SQNR performance of the proposed indexed-scaling method for 2048-point FFT is shown in Fig. 11. The same radix-$2^2/2^3$ algorithm is employed in floating/fixed-point FFT for comparison. As we can see, the performance of the proposed method is close to the upper bound (floating-point) and over 5-bit wordlength can be saved compared to the fixed-point arithmetic (incorporating an attenuation factor of 1/2 at each stage) [10]. Assuming the target implementation loss is less than 0.2 dB for 16-QAM signals at 17-dB signal-to-noise-ratio (SNR), a 9-bit wordlength is sufficient in our proposed design. As for the previous scaling-based algorithms [6]–[8], both the SQNR and SNR performances are listed in Table I. Although our proposed method does not have the highest SQNR, the impact to system SNR is within 0.03 dB which is negligible.

Now we compare the memory costs between the proposed indexed-scaling method and other scaling-based algorithms [6]–[8]. Here we take radix-2 single delay feedback (SDF) FFT processor as our analysis case. The relative cost of each algorithm for other FFT structures is similar. Assuming the FFT inputs have W-bit wordlength for real/imaginary part with a fixed decimal-point. Since the internal wordlength is fixed in our FFT processor, the memory size of ith-stage FIFO is

$$M_{\mathrm{FIFO}_i} = 2W \times N/2^i, \quad 1 \le i \le \log_2 N \tag{1}$$

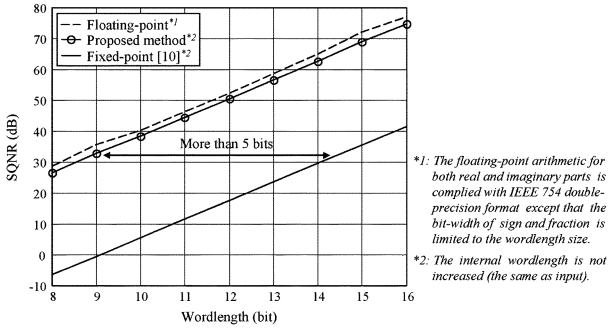


Fig. 11. SQNR performance of the indexed-scaling method (2048-point).

TABLE I
SQNR and SNR Comparisons (9-Bit Wordlength for 2048-Point FFT)

	CBFP [6]	Data Scaling [7]	Co-Optimization [8]	This Work
SQNR	31 dB	33.7 dB	33.2 dB	32.7 dB
SNR Loss	0.21 dB	0.12 dB	0.13 dB	0.15 dB

where N is the FFT size. For the feed-in path at stage i, up to i-1 indexes are required to be stored in $\mathbf{IND}_{\mathrm{FI}}$ and each of them needs $\log_2 N - i$ bits for encoding. Thus, the total size of scaling storage for feed-in path at stage i is

$$M_{\mathrm{FI}_i} = \underbrace{(i-1)(\log_2 N - i)}_{\text{Size of IND}_{\mathrm{FI}}} + \underbrace{\lfloor \log_2 (i-1) \rfloor + 1}_{\text{Size of EXP}_{\mathrm{FI}}}, \quad 2 \le i \le \log_2 N. \tag{2}$$

Similarly, the scaling storage size for feedback path at stage i is

$$M_{\mathrm{FB}_i} = \underbrace{i(\log_2 N - i)}_{\text{Size of IND}_{\mathrm{FB}}} + \underbrace{\lfloor \log_2 i \rfloor + 1}_{\text{Size of EXP}_{\mathrm{FB}}}, \quad 1 \le i \le \log_2 N \tag{3}$$

where $\lfloor \cdot \rfloor$ denotes the floor operator. By summation of (1), (2), and (3) for all FFT stages, the memory requirement of the proposed indexed-scaling method is listed in Table II. Also, a summary of the memory costs for other scaling-based algorithms is given in Table II for comparison. Note that CBFP is not employed in the first two stages of CBFP FFT to save the intermediate buffers [6] and thus the wordlength of these stages is increased to prevent overflows. Besides, we assume tailored-exponents are employed for data scaling and co-optimization FFTs as described in [8]. $D_i - 1$ is the optimal intermediate buffer size in the co-optimization algorithm which can be calculated [8]. From Table II, the memory costs of both scaling overhead and data FIFOs are plotted with 9-bit input wordlength in Figs. 12 and 13. We can find that the proposed algorithm has the minimum memory requirements in both scaling elements and FIFOs. The total memory cost of the proposed method is only about 66% of that of CBFP FFT. When compared to the co-optimization algorithm, a memory reduction of 11% can still be achieved by the proposed indexed-scaling algorithm.

V. Chip Implementation

A test chip of a 2048-point indexed-scaling FFT processor has been implemented using UMC 90 nm 1P9M CMOS technology. Our target application is a 16-QAM, 1.2 Gsample/s wideband OFDM system which can provide over 2 Gbps data rate. A 9-bit wordlength is chosen to meet 32.7 dB SQNR and the core area

TABLE II
MEMORY REQUIREMENT FOR DIFFERENT SCALING METHODS

	CBFP [6]	Data Scaling [7]	Co-Optimization [8]	This Work
Data FIFO (bit)	$(N-1)(2W+2)+N$	$(N-1)(2W+2)$	$(N-1)(2W+2)$	$(N-1)(2W)$
Intermediate buffer (bit)	$(\frac{N}{4}-\log_2 N+1)(2W+2)+\frac{N}{2}-4$	0	$(D_1-1)(2W+2)$ $+\sum_{i=1}^{\log_2 N-2}(D_{i+1}-1)(2W+\lfloor\log_2 i\rfloor+3)$	0
Scaling storage (bit)	$N-2$	$\sum_{i=1}^{\log_2 N-1}\frac{N}{2^{i+1}}(\lfloor\log_2 i\rfloor+1)$	$\sum_{i=1}^{\log_2 N-1}\frac{N}{2^{i+1}D_i}(\lfloor\log_2 i\rfloor+1)$	$\sum_{i=1}^{\log_2 N-1}(\lfloor\log_2 i\rfloor+1)+\sum_{i=1}^{\log_2 N}(\lfloor\log_2 i\rfloor+1)$ $+\frac{\log_2 N(\log_2 N-1)(2\log_2 N-1)}{6}$

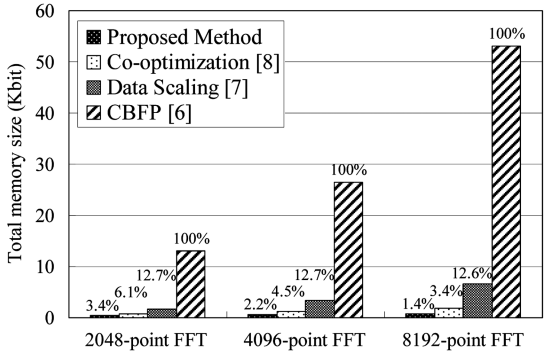


Fig. 12. Comparison of scaling overhead for different FFT processors.

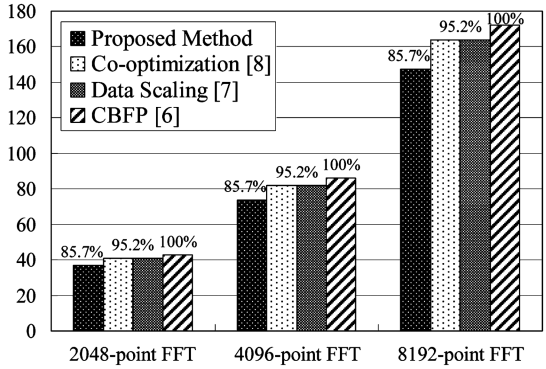
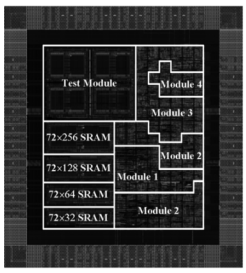


Fig. 13. Comparison of data FIFO for different FFT processor.



Technology	90 nm 1P9M
Supply voltage	1.0 volt
Wordlength	9 bits
Memory size	72×480 bits
Gate count (excl. memory)	181 K
Core area (excl. test module)	0.97 mm^2
Working frequency	300 MHz
Power consumption	117 mW

Fig. 14. Layout and chip summary of the proposed FFT processor.

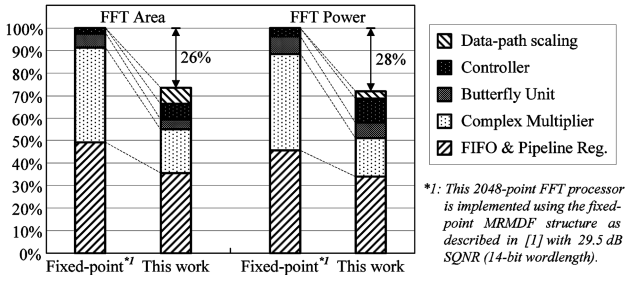


Fig. 15. Chip comparisons between fixed-point and proposed FFT processors.

of this FFT chip is 0.97 mm^2 as illustrated in Fig. 14. From post-layout prime power simulation, it is shown that our FFT processor consumes about 117 mW at 300 MHz. For comparison, a 2048-point FFT processor with fixed-point radix-$2^2/2^3$ MRMDF structure described in [1] is also implemented under the same throughput and SQNR specifications. The chip comparison between the proposed indexed-scaling FFT processor and fixed-point FFT processor is given in Fig. 15. It is clear that both area and power can be greatly reduced by the proposed scheme with little scaling overhead. Compared to the fixed-point FFT processor, about 26% area and 28% power can be saved using the proposed indexed-scaling method.

VI. CONCLUSION

In this brief, an indexed-scaling pipelined FFT processor has been proposed for wideband OFDM applications such as the next-generation gigabit WPAN systems. With a new indexed-scaling method and the MRMDF structure, both very-high-throughput requirement and lower hardware cost can be achieved. Compared to the previous scaling-based algorithms, it is shown that at least 11% memory size is reduced. Moreover, from the post-layout simulation, the proposed FFT processor can save about 26% chip area and 28% power consumption under the same throughput and SQNR specifications compared to conventional high-throughput FFT processors.

REFERENCES

[1] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications," *IEEE J. Solid-State Circuits*, vol. 40, no. 8, pp. 1726–1735, Aug. 2005.

[2] J. Lee, H. Lee, S.-I. Cho, and S.-S. Choi, "A high-speed, low-complexity radix-2^4 FFT processor for MB-OFDM UWB systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2006, pp. 4719–4722.

[3] Y.-W. Lin and C.-Y. Lee, "Design of an FFT/IFFT processor for MIMO OFDM systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 807–815, Apr. 2007.

[4] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A dynamic scaling FFT processor for DVB-T applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 2005–2013, Nov. 2004.

[5] E. Bidet, D. Castelain, C. Joanblanq, and P. Senn, "A fast single-chip implementation of 8192 complex point FFT," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 300–305, Mar. 1995.

[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. (ISCAS)*, 2000, pp. 693–696.

[7] T. Lenart and V. Öwall, "A 2048 complex point FFT processor using a novel data scaling approach," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2003, pp. 45–48.

[8] T. Lenart and V. Öwall, "Architectures for dynamic data scaling in 2/4/8K pipeline FFT cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 11, pp. 1286–1290, Nov. 2006.

[9] H. Shousheng and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. URSI Int. Symp. Signals, Syst., Electron.*, 1998, pp. 257–262.

[10] A. V. Oppenheim and C. J. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," *Proc. IEEE*, vol. 60, no. 8, pp. 957–976, Aug. 1972.