

Memory efficient DIT-based SDF IFFT for OFDM systems

Ho-Yun Lee¹, Jun-Ho Kim¹, In-Gul Jang², Kyung-Ju Cho³,
and Jin-Gyun Chung^{1a)}

¹ Division of Electronic Engineering, Chonbuk National University, Jeonju, Korea

² Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea

³ Department of Electronic Engineering, Wonkwang University, Iksan, Korea

a) jgchung@chonbuk.ac.kr (Corresponding Author)

Abstract: This paper presents a new memory reduction method of IFFT for OFDM systems, based on a mapping of three IFFT input signals which consist of modulated data, pilot and null signals. The proposed method focuses on reducing the memory size in the bit-reversal part which requires the largest number of memory cells in IFFT architectures. To reduce the memory size, we propose a decimation-in-time (DIT) twiddle factor shifting algorithm to remove the multiplications in the first two stages. It is shown that the proposed method achieves a memory reduction of about 30% compared to previous methods.

Keywords: IFFT, memory reduction, DIT, SDF

Classification: Integrated circuits

References

- [1] S. He and M. Torkelson: IPPS (1996) 766.
- [2] J. Y. Oh and M. S. Lim: IEICE Trans. Electron. **E88-C** (2005) 1740.
- [3] T. S. Cho and H. H. Lee: IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **21** (2013) 187.
- [4] I. G. Jang, K. J. Cho, Y. E. Kim and J. G. Chung: IEICE Trans. Commun. **E95-B** (2012) 2095.
- [5] I. G. Jang, K. J. Cho, H. Y. Kim and J. G. Chung: IEICE Electron. Express **10** (2013) 20130530.

1 Introduction

IFFT/FFT is one of the key components in the OFDM systems. For the implementation of IFFT/FFT, the pipelined architectures are widely used since they offer high throughput and low latency, as well as a reasonable low area and power consumption. Among the various pipelined IFFT/FFT architectures, the single-path feedback (SDF) approach based on the radix- 2^r algorithm is frequently used for its low cost and high efficiency [1, 2, 3].

In N -point radix- 2^2 SDF IFFT/FFT design, the number of the total required feedback memory location is $(N-1)$. Also, the bit-reversal part requires $2 \times N$ memory locations for reordering the input sequence in decimation-in-time (DIT) algorithm, and reordering the output sequence in decimation-in-frequency (DIF) algorithm. Note that the required memory

size in the bit-reversal part nearly doubles the feedback memory in SDF IFFT.

In [4], to reduce the memory size of IFFT for OFDM systems, the combined integer mapping which generates mapped data and pilot/null signals was proposed. In [5], a signed integer mapping with a simple post-processing was proposed to further reduce the memory size of IFFT. However, the approaches in [4, 5] cannot be used to reduce the memory size in the bit-reversal part.

2 Backgrounds

Each subcarrier in most OFDM systems is modulated using either BPSK, QPSK, 16-QAM or 64-QAM modulation schemes. To achieve the same average power for all modulation schemes, the final modulated data (d) is formed by multiplying the mapped value ($I + jQ$) by a normalization factor (K_{MOD}) as

$$d = (I + jQ) \times K_{MOD} \quad (1)$$

where K_{MOD} is 1, $1/\sqrt{2}$, $1/\sqrt{10}$, and $1/\sqrt{42}$ for BPSK, QPSK, 16-QAM and 64-QAM, respectively. In all of OFDM symbols, to obtain several advantages in the transmitter and receiver, pilot and null signals should be inserted in specified subcarriers. Pilot signals are BPSK-modulated, and null signals are set to zero.

Fig. 1 shows the subcarrier modulation procedures. In Fig. 1 (a), the value d , pilot and null signals are quantized with W -bit, depending on the required SQNR before IFFT operation. However, in Fig. 1 (b), mapped value ($I + jQ$) is mapped by signed integer mapping with W_m -bit (for example, 4-bit for 64-QAM) and pilot signals are mapped into zero. The stage 2 outputs are multiplied by K_{MOD} and added the compensation values (CVs) outputs to correct the error due to zero-mapping of the pilot. The compensated input signals to stage 3 are expressed as

$$\begin{aligned} stg3_in = & (stg2_out_{re} \cdot K_{MOD} + CV_{re}) \\ & + j(stg2_out_{im} \cdot K_{MOD} + CV_{im}) \end{aligned} \quad (2)$$

where the subscript re and im mean real and imaginary parts, respectively.

Fig. 2 shows N -point radix-2² DIF-based SDF IFFT architecture. Each

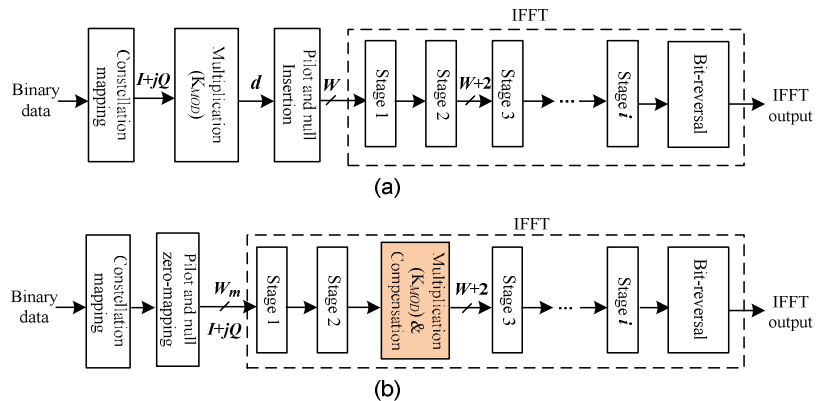


Fig. 1. Modulation procedures: (a) conventional and (b) mapping [5].

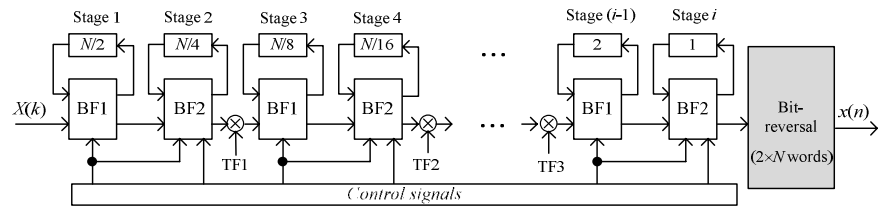


Fig. 2. N -point radix- 2^2 DIF-based SDF IFFT architecture.

number inside the boxes represents the feedback memory size. The feedback memory at stage i stores $2 \times N/2^i$ pieces of complex data. Butterfly type 2 (BF2) has the same structure with butterfly type 1 (BF1) except for trivial multiplication by $-j$. The number of memory cells required for i -th stage is $2 \times N/2^i \times W_i$, where W_i is the word-length of the output signal from i -th stage. In the bit-reversal part, the size of memory cells is $2 \times 2 \times N \times W_i$ for reordering the output signal.

3 Proposed DIT-based IFFT design

In DIF-based SDF IFFT architectures, the bit-reversal part requires the largest amount of memory. The word-length reduction approach in [5] can only reduce the memory sizes at stages 1 and 2. However, if the bit-reversal part is located before stage 1 as shown in Fig. 3, the word-length reduction technique can be applied to in the bit-reversal part. To apply the memory reduction technique, there is a restriction that the computations of stages 1 and 2 should not contain multiplication.

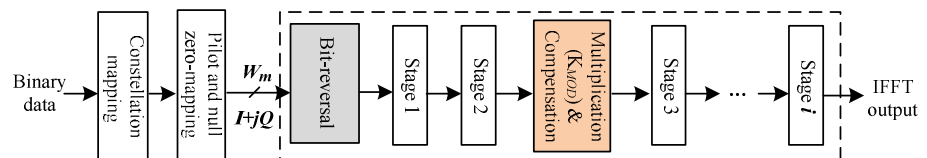


Fig. 3. Proposed DIT-based IFFT procedure.

3.1 2^{2l} -point DIT-based SDF IFFT architecture

2^{2l} -point radix- 2^2 DIT-based SDF IFFT architectures satisfy the restriction and consequently the technique in [5] can be directly applied. As an example of memory reduction, in 64-QAM scheme, modulated signals have a word-length of 4-bits. It is possible to reduce the memory size of bit-reversal by $2 \times 2 \times N \times (W-4)$ since the memory size of bit-reversal in DIF-based architectures is $2 \times 2 \times N \times W$.

3.2 2^{2l+1} -point DIT-based IFFT architecture

The signal flow graph of 2^5 -point DIT-based IFFT is shown in Fig. 4 (a). The memory reduction technique in [5] cannot be directly applied since complex multiplications exist between stages 1 and 2. For stages 1 and 2, only a group of 8 signals are added, subtracted or multiplied as dashed line in Fig. 4 (a). The numbers on branches mean an exponent of twiddle factor base, $W_N = e^{2\pi/N}$. Moreover, the twiddle factors in stages 1 and 2 have $W_N^{(N/4)}$ as a common factor. Based on these observations, a twiddle factor

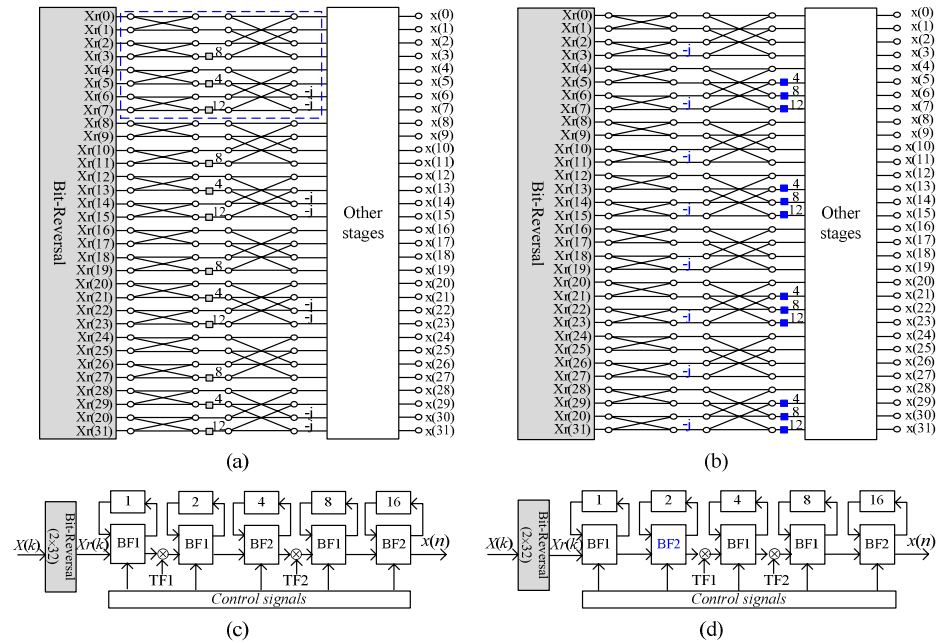


Fig. 4. Signal flow graph and SDF architectures for radix- 2^2 32-point IFFT: (a) original, (b) proposed, (c) original SDF and (d) proposed SDF.

shifting algorithm is proposed as follows:

Twiddle Factor Shifting Algorithm

In 2^{l+1} -point radix- 2^2 DIT-based SDF IFFT architecture, all the twiddle factors in stages 1 and 2 are $W_N^{1/2(N/4)}$, $W_N^{(N/4)}$ and $W_N^{3/2(N/4)}$, where $W_N^k = e^{2\pi k/N}$.

1. Decomposed $W_N^{3/2(N/4)}$ into $W_N^{1/2(N/4)}$ and $W_N^{(N/4)}$.
2. Shift $W_N^{1/2(N/4)}$ to butterfly output edges, using the relation in Fig. 5.

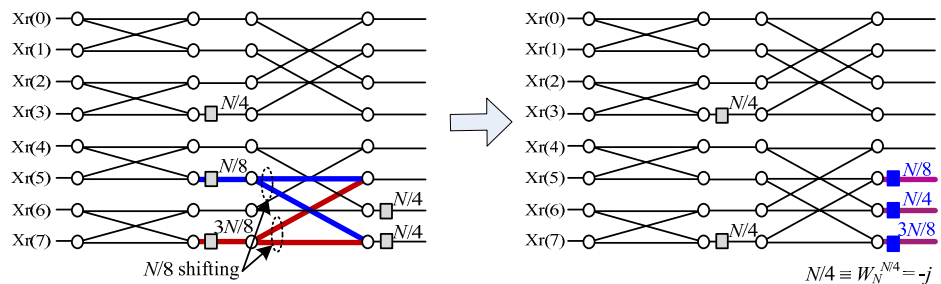


Fig. 5. (a) Principle of twiddle factor shifting.

By using DIT-based twiddle factor shifting algorithm, the signal flow in Fig. 4 (b) can be obtained from Fig. 4 (a). Notice that, since $W_N^{N/4}$ is $-j$, no multiplication is needed by $W_N^{N/4}$. By the algorithm, multiplications are moved after stage 2. Thus, the technique in [5] can be applied to DIT-based SDF IFFT architecture.

BF at stage i generates the addition output (a_{oi}) and subtraction output (s_{oi}) which are feed into the next stage and the feedback memory, respectively. If bit-reversed inputs in DIT-based SDF IFFT architectures is $X_r(n)$, the output signals from BF at stage 2 in Fig. 4 (b) can be expressed as

$$\begin{aligned}
 a_{o2}(4m) &= X_r(4m) + X_r(4m+1) + X_r(4m+2) + X_r(4m+3) \\
 s_{o2}(4m) &= X_r(4m) + X_r(4m+1) - [X_r(4m+2) + X_r(4m+3)] \\
 a_{o2}(4m+1) &= X_r(4m) - X_r(4m+1) - j[X_r(4m+1) - X_r(4m+3)] \\
 s_{o2}(4m+1) &= X_r(4m) - X_r(4m+1) + j[X_r(4m+1) - X_r(4m+3)] \\
 \text{for } m &= 0, 1, \dots, (N/4 - 1)
 \end{aligned} \quad (3)$$

From (3), considering the feedback path at stage 2 as shown in Fig. 4 (d), stage 3 input signals before the multiplication by twiddle factor can be expressed as

$$\begin{aligned}
 stg3_in(4m) &= a_{o2}(4m), stg3_in(4m+1) = a_{o2}(4m+1) \\
 stg3_in(4m+2) &= s_{o2}(4m), stg3_in(4m+3) = s_{o2}(4m+1)
 \end{aligned} \quad (4)$$

Assuming that an IFFT input signal $X_r(n_0)$ is a pilot signal p_0 . From (4), it can be noticed that $X_r(n_0)$ has effects on stage 3 input signals $stg3_in(4m)$, $stg3_in(4m+1)$, $stg3_in(4m+2)$ and $stg3_in(4m+3)$. Thus, if zero input is used instead of the real pilot p_0 , the p_0 should be added to the four stage 3 input signals as a CV.

Table I summarizes the compensation values to the third stage input according to four different pilot signal locations. In Table I, the number of parentheses is binary representation of input order at each stage. Similar to [5], the quantized value of the compensated stage 3 signals can be obtained using two look-up tables and compensation circuits derived from Table I.

Table I. CV to stage 3 inputs (k : no. of stage, %: modulo).

Pilot location in reordered input ($n = l_{k-1}l_{k-2} \dots l_0$)	Pilot signal	Stage 3 input order ($l_{k-1}l_{k-2} \dots l_0$)	CV
$n \% 4 = 0$ ($n = l_{k-1}l_{k-2} \dots l_2 00$)	p_0	$n \ (l_{k-1}l_{k-2} \dots l_2 00)$ $n+1 \ (l_{k-1}l_{k-2} \dots l_2 01)$ $n+2 \ (l_{k-1}l_{k-2} \dots l_2 10)$ $n+3 \ (l_{k-1}l_{k-2} \dots l_2 11)$	p_0 p_0 p_0 p_0
$n \% 4 = 1$ ($n = l_{k-1}l_{k-2} \dots l_2 01$)	p_1	$n-1 \ (l_{k-1}l_{k-2} \dots l_2 00)$ $n \ (l_{k-1}l_{k-2} \dots l_2 01)$ $n+1 \ (l_{k-1}l_{k-2} \dots l_2 10)$ $n+2 \ (l_{k-1}l_{k-2} \dots l_2 11)$	p_1 $-p_1$ p_1 $-p_1$
$n \% 4 = 2$ ($n = l_{k-1}l_{k-2} \dots l_2 10$)	p_2	$n-2 \ (l_{k-1}l_{k-2} \dots l_2 00)$ $n-1 \ (l_{k-1}l_{k-2} \dots l_2 01)$ $n \ (l_{k-1}l_{k-2} \dots l_2 10)$ $n+1 \ (l_{k-1}l_{k-2} \dots l_2 11)$	p_2 $-jp_2$ $-p_2$ jp_2
$n \% 4 = 3$ ($n = l_{k-1}l_{k-2} \dots l_2 11$)	p_3	$n-3 \ (l_{k-1}l_{k-2} \dots l_2 00)$ $n-2 \ (l_{k-1}l_{k-2} \dots l_2 01)$ $n-1 \ (l_{k-1}l_{k-2} \dots l_2 10)$ $n \ (l_{k-1}l_{k-2} \dots l_2 11)$	p_3 jp_3 $-p_3$ $-jp_3$

4 Performance comparisons

To compare the memory efficiency in SDF IFFT design, the memory size for the feedback path and bit-reversal part, and additional hardware of the conventional method, previous method [5], and proposed method are listed in Table II. In Table II, N_p is the number of pilot signals. We assume that 64-QAM modulation scheme is used, and IFFT processors have fixed-width property to truncate least significant bits of the output signals in each stage. In stages 1 and 2 of the proposed method and [5], the sign extension

Table II. Comparison of the memory size and additional H/W.

	Conventional DIF-based IFFT	DIF-based IFFT [5]	Proposed DIT-based IFFT
Bit-reversal	$2 \times 2 \times W \times N$	$2 \times 2 \times W \times N$	$2 \times 2 \times 4 \times N$
Stage 1	$2 \times W \times N/2$	$2 \times 5 \times N/2$	$2 \times 5 \times 1$
Stage 2	$2 \times W \times N/4$	$2 \times 6 \times N/4$	$2 \times 6 \times 2$
Rest stages	$2 \times W \times (N/4 - 1)$	$2 \times W \times (N/4 - 1)$	$2 \times W \times (N - 4)$
Look-up table	-	$9 \times W + 2 \times N_p$	$9 \times W + 2 \times N_p$
Additional	-	2 adders + 3 MUX + sign inverter	2 adders + 3 MUX + sign inverter

should be considered for accurate computation before multiplication with K_{MOD} . Notice that in the bit-reversal part, the memory size of the proposed method does not depend on IFFT input word-length. However, the conventional and previous methods depend on the IFFT input word-length.

Fig. 6 shows the comparison of memory reduction according to IFFT size and word-length. It can be seen that the proposed method achieves a memory reduction about 30% reduction compared to the method in [5].

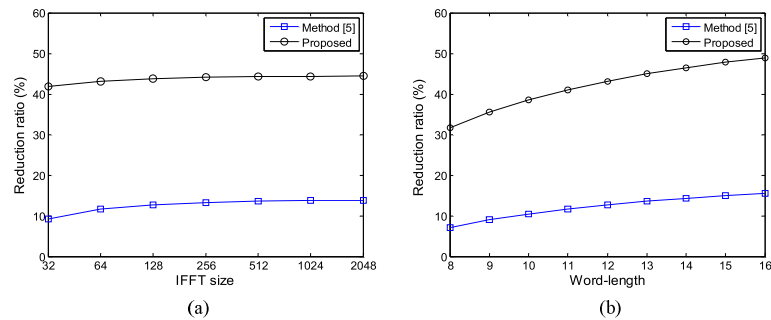


Fig. 6. Comparison of memory reduction: (a) according to IFFT size ($W=16$) and (b) according to word-length ($N=128$).

5 Conclusions

We proposed a new IFFT design method to reduce the memory size of IFFT for OFDM systems. Since the bit-reversal part requires the largest amount of memory, the proposed method focuses on reducing the memory cells in the bit-reversal part using DIT-based twiddle factor shifting algorithm. The benefits of the proposed method can be maximized in OFDM systems with long word-length.

Acknowledgments

This work was supported by the IT R&D program of MOTIE/KEIT 10044092, Development of Core IPs of OFDM PHY and RF Transceiver for 60 GHz Wireless LAN/PAN in application of 7 Gbps Wireless Multimedia Services.