

# Low-latency and memory-efficient SDF IFFT processor design for 3GPP LTE

Yeon-Jin Kim<sup>1</sup>, In-Gul Jang<sup>2</sup>, Kyung-Ju Cho<sup>3a)</sup>,  
and Jin-Gyun Chung<sup>1b)</sup>

<sup>1</sup> Division of Electronic Engineering, Chonbuk National University, Jeonju, Korea

<sup>2</sup> Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea

<sup>3</sup> Department of Electronic Engineering, Wonkwang University, Iksan, Korea

a) [kjcho@wku.ac.kr](mailto:kjcho@wku.ac.kr); Co-corresponding Author

b) [jgchung@jbnu.ac.kr](mailto:jgchung@jbnu.ac.kr); Co-corresponding Author

**Abstract:** This paper presents a low latency IFFT design method for 3rd generation partnership project long term evolution (3GPP LTE). The proposed method focuses on reducing the delay buffer size in the first stage of single-path delay feedback (SDF) IFFT architectures since the first stage occupies about 50% of the overall delay buffer. In order to reduce the buffer size, we propose the reordering scheme of IFFT input data. By using the reordered input data, both the latency and the memory in the first stage are significantly reduced. Simulation results show that the latency for 2048-point IFFT is reduced about 41% compared with conventional architecture.

**Keywords:** SDF, IFFT, low latency, memory reduction, 3GPP LTE

**Classification:** Circuits and modules for electronic instrumentation

## References

- [1] B. M. Baas: "A low-power, high-performance, 1024-point FFT processor," IEEE J. Solid-State Circuits **34** (1999) 380 (DOI: [10.1109/4.748190](https://doi.org/10.1109/4.748190)).
- [2] S. Y. Peng, *et al.*: "Energy-efficient 128~2048/1536-point FFT processor with resource block mapping for 3 GPP-LTE system," IEEE Int. Conf. Green Circuits Syst. (2010) 14 (DOI: [10.1109/ICGCS.2010.5543106](https://doi.org/10.1109/ICGCS.2010.5543106)).
- [3] M. Khelifi, *et al.*: "Parallel independent FFT implementation on Intel processors and Xeon Phi for LTE and OFDM systems," NORCAS (2015) 1 (DOI: [10.1109/NORCHIP.2015.7364402](https://doi.org/10.1109/NORCHIP.2015.7364402)).
- [4] S. He and M. Torkelson: "Designing pipeline FFT processor for OFDM (de)modulation," IEEE URSI Int. Symp. Signals. Syst., Electron. (1998) 257 (DOI: [10.1109/ISSSE.1998.738077](https://doi.org/10.1109/ISSSE.1998.738077)).
- [5] J. Y. Oh and M. S. Lim: "New radix-2 to the 4th power pipeline FFT processor," IEICE Trans. Electron. **E88-C** (2005) 1740 (DOI: [10.1093/ietele/e88-c.8.1740](https://doi.org/10.1093/ietele/e88-c.8.1740)).
- [6] T. S. Cho and H. H. Lee: "A high-speed low-complexity modified radix-2<sup>5</sup> FFT processor for high rate WPAN applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **21** (2013) 187 (DOI: [10.1109/TVLSI.2011.2182068](https://doi.org/10.1109/TVLSI.2011.2182068)).
- [7] C. Yu and M. Yen: "Area-efficient 128- to 2048/1536-point pipeline FFT processor for LTE and mobile WiMAX systems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23** (2015) 1793 (DOI: [10.1109/TVLSI.2014](https://doi.org/10.1109/TVLSI.2014)).

- 2350017).
- [8] S. N. Tang, *et al.*: “An area- and energy-efficient multimode FFT processor for WPAN/WLAN/WMAN systems,” *IEEE J. Solid-State Circuits* **47** (2012) 1419 (DOI: [10.1109/JSSC.2012.2187406](https://doi.org/10.1109/JSSC.2012.2187406)).
  - [9] I. G. Jang, *et al.*: “Memory efficient IFFT design for OFDM-based applications,” *IEICE Electron. Express* **10** (2013) (DOI: [10.1587/elex.10.20130530](https://doi.org/10.1587/elex.10.20130530)).
  - [10] 3GPP LTE: “Evolved universal terrestrial radio access (E-UTRA): LTE physical layer,” 3GPP TS 36.201 v13.0.0 (2016) <http://www.3gpp.org>.

## 1 Introduction

IFFT/FFT is one of the key components for wireless applications based on the OFDM. For hardware implementation, the various IFFT/FFT processors have been developed. These implementations can be mainly classified into two types, the memory-based architecture [1, 2, 3], and the pipelined one [4, 5, 6, 7, 8, 9]. The memory-based architecture provides a low-area and low-power solution. However, this kind of architecture style has long latency and low throughput. On the other hand, the pipelined architecture style can get rid of the disadvantages of the forgoing style at the cost of a reasonable hardware overhead [7].

Among the various pipelined IFFT/FFT architectures, SDF approach based on radix-2<sup>r</sup> algorithm [4, 5, 6] is frequently used for its low cost and high efficiency. This approach includes  $N-1$  delay buffers, where  $N$  means the processing length.

The aim of LTE is to provide an increased data rate and reduced transmission delays compared with older wireless telecommunications. LTE signal processing relies heavily on channel coding/decoding, channel estimation, IFFT/FFT and other processing blocks. All these physical layer processing should be accomplished within the slot duration of 0.5 ms in 3GPP LTE standard [10].

IFFT calculation for 3GPP LTE has the long processing time since the processing length is up to 2048. The parallel pipelined IFFT is a good solution for the applications. However, the solution suffers from high hardware costs. In this paper, we propose a low latency SDF IFFT design method based on IFFT input data reordering.

## 2 Backgrounds

In 3GPP LTE standard [10], the processing length of IFFT/FFT varies from 128 to 2048 for all specified channel bandwidths. Table I summaries 3GPP LTE physical layer parameters. The OFDM spectrum can be classified into the data band and the

**Table I.** 3GPP LTE downlink physical layer parameters

Channel bandwidth (MHz)	1.25	2.5	5	10	15	20
Sampling frequency (MHz)	1.92	3.84	7.68	15.36	23.04	30.72
IFFT size ( $N$ )	128	256	512	1024	1536	2048
Number of data ( $N_d$ )	72	150	300	600	900	1200
Number of nulls ( $N_n$ )	56	112	212	424	636	848

guard band. The data band is used for data transmission while the guard band is used to prevent interference. For example, in the case of  $N = 2048$ , 1200 data and 848 nulls are allocated in data band and guard band, respectively.

In the OFDM transmitters, the processing length  $N$  of IFFT can be expressed as

$$N = N_d + N_n, \quad (1)$$

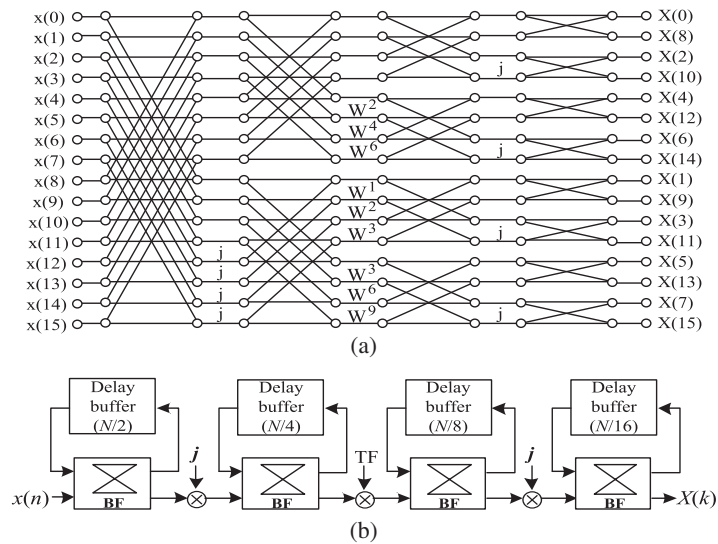
where  $N_d$  and  $N_n$  represent the number of data and that of nulls, respectively.

The signal flow graph for 16-point radix-2<sup>2</sup> IFFT is shown in Fig. 1(a). The twiddle factor (TW)  $W^i$  stands for  $e^{j2\pi i/N}$ . Fig. 1(b) shows the pipelined SDF architecture of Fig. 1(a). The butterfly operation in stage 1 can be expressed as

$$B(k_1, n_2) = x(n_2) + (-1)^{k_1} x(n_2 + N/2), \quad (2)$$

where  $k_1 = 0, 1$  and  $n_2 = 0, 1, \dots, N/2-1$ .

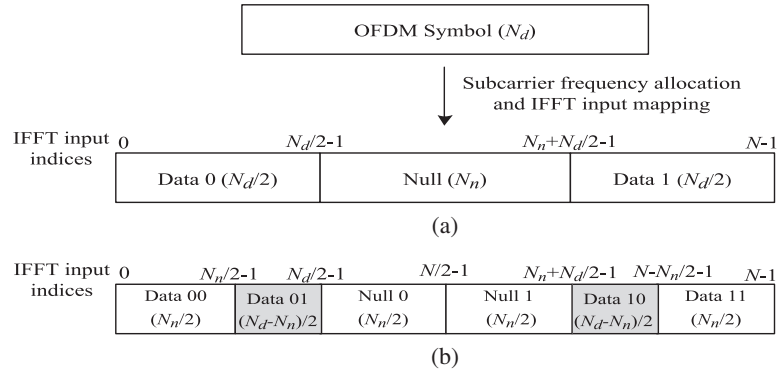
In radix-2<sup>r</sup> SDF IFFT computations, for the first  $N/2$  clock cycles, IFFT input signals are bypassed at butterfly and stored at the delay buffer in stage 1. For the next  $N/2$  clock cycles, the outputs corresponding to  $k_1 = 0$  in (2) are sent to stage 2 while the outputs corresponding to  $k_1 = 1$  are stored in the delay buffer in stage 1.



**Fig. 1.** (a) Signal flow graph of 16-point radix-2<sup>2</sup> IFFT and (b) Pipelined SDF architecture of (a).

Fig. 2(a) shows the conventional mapping of IFFT inputs for an OFDM symbol. Based on the subcarrier frequency allocation and IFFT inputs mapping rules,  $N_d$  data are mapped at two side subcarrier frequencies as *Data 0* and *Data 1*, and  $N_n$  nulls are mapped at center subcarrier frequencies. Fig. 2(b) shows the conventional IFFT input mapping scheme expressed for butterfly computations based on (2). In Fig. 2(b), the length of *Data 00* is the same as that of *Null 1*, and the length of *Data 11* is the same as that of *Null 0*. The butterfly output  $B(k_1, n_2)$  in (2) can be computed directly without addition or subtraction operations when either  $x(n_2)$  or  $x(n_2 + N/2)$  is a null signal. Thus, the butterfly operation in (2) is required only when  $n_2$  satisfies the following condition:

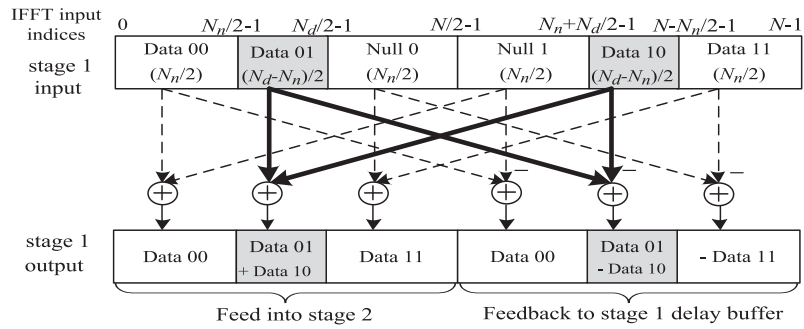
$$N_n/2 \leq n_2 \leq N_d/2 - 1. \quad (3)$$



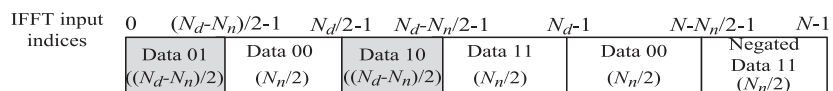
**Fig. 2.** (a) Conventional IFFT input allocation, and (b) Detailed expression of (a) for butterfly operations.

### 3 Proposed low latency IFFT design

Fig. 3 shows the inputs and outputs of butterfly operations in stage 1. Actual addition or subtraction operations (solid line) are performed only between *Data 01* and *Data 10*. Based on this observation, the latency of IFFT computation can be reduced. To remove the unnecessary butterfly operation (dashed line), we propose the reordered IFFT input mapping scheme as shown in Fig. 4, based on the butterfly output at stage 1.



**Fig. 3.** BF input and output signal in stage 1.

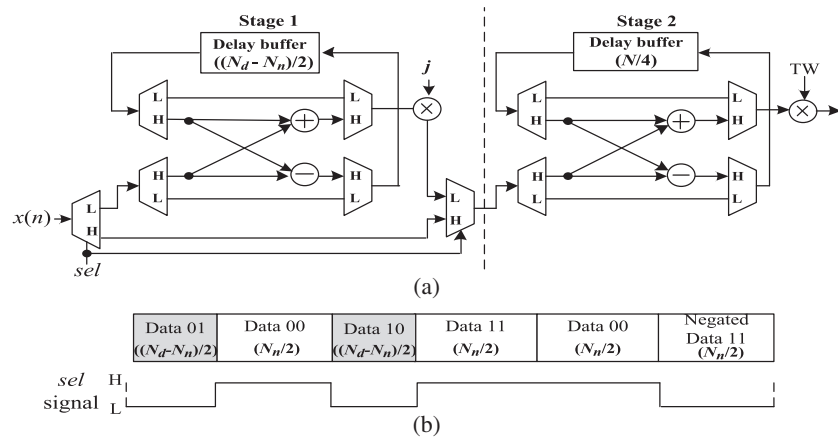


**Fig. 4.** Proposed IFFT input mapping with reordering.

Fig. 5(a) shows the proposed butterfly architecture in stage 1. Fig. 5(b) shows the bypass control signal for stage 1. As can be seen from Fig. 5(b), *Data 00* can be sent to stage 2 after  $(N_d - N_n)/2$  clock cycles while  $N/2$  clock cycles are required in conventional architectures. In addition, the memory size in stage 1 is reduced from  $N/2$  to  $(N_d - N_n)/2$ . Note that the efficiency of the proposed method depends on the number of null signals in the IFFT input. By using the proposed architecture, the latency of IFFT can be derived as

$$T_{\text{latency}} = N - N_n - 1. \quad (4)$$

As an example of  $N = 16$ , assume that  $N_d = 10$  and  $N_n = 6$ , the original IFFT inputs are  $\{x(0), x(1), x(2), x(3), x(4), 0, 0, 0, 0, 0, x(11), x(12), x(13), x(14), x(15)\}$ . The size of the delay buffer in stage 1 is 2 since  $(N_d - N_n)/2 = 2$ . Table II shows the data processing flow of radix-2<sup>2</sup> SDF IFFT for the proposed architecture



**Fig. 5.** (a) Proposed butterfly structure and (b) bypass control signals.

in Fig. 5. The original inputs are reordered by proposed method as the second column in Table II. If bypass control signal (*sel*) is 0, the reordered inputs are fed into the delay buffer which consists of {buffer (1), buffer (0)} at stage 1. Otherwise, the reordered inputs bypass butterfly operation in stage 1 and directly feed to stage 2. Also, signals at delay buffer in stage 1 are latched. To obtain the first IFFT output at stage 4, the required number of clocks is 9. The proposed scheme in this example can reduce 6 clocks compared to conventional scheme.

**Table II.** Data processing of proposed radix-2<sup>2</sup> SDF IFFT with  $N = 16$

# of clocks	Reordered inputs	Bypass signal ( <i>sel</i> )	Stored delay buffer signals at stage 1		TW	2 <sup>nd</sup> stage inputs	4 <sup>th</sup> stage BF outputs
			buffer (1)	buffer (0)			
0	$x(3)$	0	0	$x(3)$	1	0	-
1	$x(4)$	0	$x(3)$	$x(4)$	1	0	-
2	$x(0)$	1	$x(3)$	$x(4)$	1	$x(0)$	-
3	$x(1)$	1	$x(3)$	$x(4)$	1	$x(1)$	-
4	$x(2)$	1	$x(3)$	$x(4)$	1	$x(2)$	-
5	$x(11)$	0	$x(4)$	$x(3) - x(11)$	1	$x(3) + x(11)$	-
6	$x(12)$	0	$x(3) - x(11)$	$x(4) - x(12)$	$j$	$j[x(4) + x(12)]$	-
7	$x(13)$	1	$x(3) - x(11)$	$x(4) - x(12)$	$j$	$jx(13)$	-
8	$x(14)$	1	$x(3) - x(11)$	$x(4) - x(12)$	$j$	$jx(14)$	-
9	$x(15)$	1	$x(3) - x(11)$	$x(4) - x(12)$	$j$	$jx(15)$	X(0)
10	$x(0)$	1	$x(3) - x(11)$	$x(4) - x(12)$	1	$x(0)$	X(8)
11	$x(1)$	1	$x(3) - x(11)$	$x(4) - x(12)$	1	$x(1)$	X(4)
12	$x(2)$	1	$x(3) - x(11)$	$x(4) - x(12)$	1	$x(2)$	X(12)
13	$-x(13)$	0	$x(4) - x(12)$	$-x(13)$	1	$x(3) - x(11)$	X(2)
14	$-x(14)$	0	$-x(13)$	$-x(14)$	1	$x(4) - x(12)$	X(10)
15	$-x(15)$	0	$-x(14)$	$-x(15)$	1	$-x(13)$	X(6)

#### 4 Performance comparisons

To evaluate the performance of SDF IFFT designs applicable to 3GPP LTE, the latency and memory (delay buffer) size of the conventional method, the previous method [9] and the proposed method are listed in Table III. Compared to the conventional and previous method, the proposed method achieves about 41% reduction in latency for 2048-point IFFT as shown in Table IV.

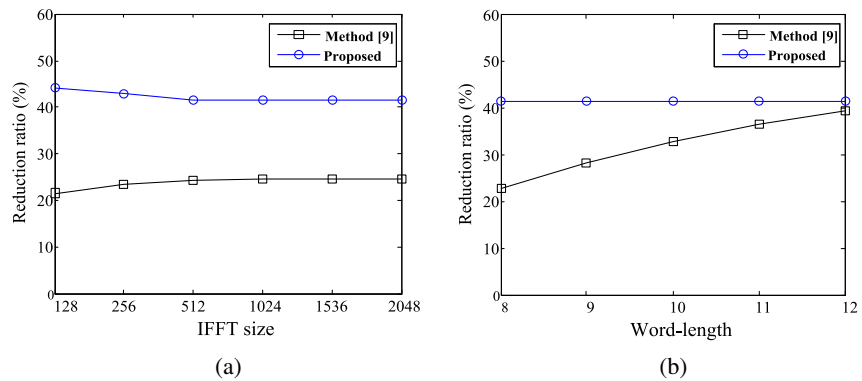
For memory size comparison, we assume that 64-QAM modulation scheme is used, and IFFT processors have fixed-width property to truncate least significant bits of the output signals in each stage. In [9], the memory size reduction approach of IFFT for OFDM applications has been proposed by combined integer mapping for pilot and null signal. In Table III,  $N_p$  is the number of pilot signals. Fig. 6 shows the comparison of memory reduction according to IFFT size and word-length for 3GPP LTE applications. It can be seen that the proposed method achieves up to 44% memory reduction compared to the conventional method for  $N = 128$ .

**Table III.** Comparison of latency, memory and additional hardware ( $W$ : word size).

	Conventional	[9]	Proposed
Latency (clocks)	$N-1$	$N-1$	$(N_d - N_n)/2 + N/2-1$
Memory of stage 1	$2 \times W \times N/2$	$2 \times 5 \times N/2$	$2 \times W \times (N_d - N_n)/2$
Memory of stage 2	$2 \times W \times N/4$	$2 \times 6 \times N/4$	$2 \times W \times N/4$
Memory of rest stages	$2 \times W \times (N/4-1)$	$2 \times W \times (N/4-1)$	$2 \times W \times (N/4-1)$
Look-up table	-	$9 \times W + 2 \times N_p$	-
others	-	2 adders + 3 MUX + sign inverter	2 MUX

**Table IV.** Comparison of latency for IFFT sizes

$N$	128		256		512		1024		1536		2048	
Method	Conv.	Prop.	Conv.	Prop.	Conv.	Prop.	Conv.	Prop.	Conv.	Prop.	Conv.	Prop.
Latency	127 (1)	71 (0.56)	255 (1)	146 (0.57)	511 (1)	299 (0.59)	1023 (1)	599 (0.59)	1535 (1)	899 (0.59)	2047 (1)	1199 (0.59)



**Fig. 6.** (a) Memory reduction for IFFT size ( $W = 12$ ), and (b) Memory reduction for word-length ( $N = 2048$ ).

## 5 Conclusions

In this paper, a low latency radix-2<sup>r</sup> SDF IFFT architecture was proposed. In order to reduce the latency, we proposed a reordering method of the IFFT input data based on the fact that IFFT input includes a specified number of null signals. By the proposed method, the latency is reduced about 41% in 3GPP LTE applications, compared with conventional architecture.

## Acknowledgments

This paper was supported by Wonkwang University in 2015.