

An Area-Efficient Multimode FFT Circuit for IEEE 802.11 ax WLAN Devices

Phuong T.K. Dinh^{*†}, Leonardo Lanante^{*}, Minh D. Nguyen[†], Masayuki Kurosaki^{*} and Hiroshi Ochi^{*} ^{*}Graduate School of Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka, Japan

[†]School Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam

Email:phuongminhdud@gmail.com

Abstract—Multi-mode fast Fourier Transform (FFT) circuits are essential in orthogonal frequency domain (OFDM) based systems which supports multiple bandwidth. Typically, hardware implementation employs a single FFT circuit for the highest supported bandwidth and using oversampling, the same FFT circuit is used to support lower bandwidth. For the new 802.11ax wireless local area network (WLAN) standard whose frame consists of the regular 3.2us length symbol as well as a longer 12.8us symbol, a fast switchable double-mode FFT circuit is required. In addition, the 802.11ax SIG-B symbol contains a maximum of two independent symbol streams that requires two FFT circuits for the 3.2us symbol length. Our proposed FFT architecture is optimized to support the 802.11ax standard with low latency, area and power requirements. FPGA implementation results show that our proposed circuit has efficiency 13.7% lower area compared to conventional architecture.

Keywords—OFDMA, IFFT/FFT, parallel FFT, SDF, MDC, 802.11ax.

I. INTRODUCTION

IEEE 802.11ax is the newest amendment to the 802.11 wireless local area network (WLAN) standard whose purpose is to increase the network efficiency in highly density areas. To reduce the degradation resulting from the high number of users simultaneously trying to access the channel, 802.11ax has must implemented orthogonal frequency division multiple access (OFDMA).

OFDMA involves allocating resource units in the form of clusters of subcarriers to users. To effectively implement this, OFDMA based systems typically feature longer OFDM symbol lengths and hence which provides support to higher number of distributable subcarriers per symbol. In 802.11ax, due to the requirement of backward compatibility, two pre-guard interval symbol lengths are needed; one is the 3.2us symbol length while the other is 12.8us.

While 802.11 devices utilise multiple FFT sizes including 64, 128 and 256 to support 20MHz, 40MHz and 80MHz bandwidths, an area optimized design will only implement a single FFT circuit that supports the highest signal bandwidth and use the same FFT circuit with the lower bandwidth signals after oversampling it accordingly.

Recently, many architectures have been configured to a variety of FFT sizes that support both FFT and flexible FFT. Most architects focus on designs with minimum energy consumption where deep pipelines are used to improve energy efficiency

of FFT processors. In addition to, Single-path Delay Feedback (SDF) architecture, Multipath Delay Commutator (MDC) are attracting attention and encouraging architects to design IFFT/FFT models which can provide high throughput and high efficiency of memory usage. Some designs combine MDC and SDF to create pipeline FFT which has low latency[8], as well as power consumption and area minimization [6]. Even though they use mixed radix 4 and radix 8 to design variable length FFT processor that integrate two radix 2 stages and three radix 2³ stages for FFT size 512, 1024, 2048 was proposed in [11].

However, these designs are not suitable for 802.11 ax. This standard requires many specifications in the medium access control (MAC) sublayer and the physical layer (PHY) for high efficiency (HE) operation in frequency bands between 1GHz and 6GHz for Wireless LAN. In addition, PHY guarantee very high throughput (VHT) when HE STA is operating in the 5GHz band and high throughput (HT) when it is operating in 2.4 GHz. The HE PHY extends the maximum number of users supported for downlink multi-user MIMO (MU-MIMO) transmissions to eight and provides support for downlink (DL) and uplink (UL) OFDMA as well as for uplink multiuser (MU)-MIMO. Both downlink and uplink MU-MIMO transmissions are supported on the Physical Layer Convergence Protocol Packet Data Unit (PPDU) bandwidth. In an MU-MIMO resource unit, there is support for up to eight users with up to four space-time streams per user with the total number of space-time streams not exceeding eight.

In addition, 802.11ax is developed from previous 802.11 resulting in its PHY has two parts. The first part is pre HE modulated which includes L-STF (Non-HT short training field), L-LTF (Non-HT long training field), L-SIG (Non-HT SIGNAL field)[which is the same as in legacy 802.11], RL-SIG (Repeated Non-HT SIGNAL field), HE-SIG A (HE-SIGNAL A field) and HE-SIG B (HE-SIGNAL B) support to HE PPDU. The second part is HE modulated and is used only for HE PPDU in 802.11ax. It includes HE-STF (HE-short training field), HE-LTF (HE long training field), data and PE (packet extension field). Subcarrier frequency spacing for the pre HE modulated fields requires 312.5 kHz and for HE modulated fields requires 78.125 kHz on each 20Mhz band [1] [2] [3]. Therefore, it needs FFT 64 n (n depends on the bandwidth of the PHY frame) point for pre HE modulated fields and FFT 256 n point for HE modulated fields in HE PPDU. Furthermore, in 802.11ax, HE PPDU format consists of HE SU PPDU (HE single user PPDU), HE MU PPDU

(HE multiuser PPDU), HE EXT SU PPDU (HE extended range single user PPDU) and HE trigger based PPDU, so that the transmitter block diagram for each part in every HE PPDU is different. This is especially in the case of HE MU PPDU using HE SIG B, which includes using a common block field. This block field contains; a) carriers' RU allocation of subfields dependent on the PPDU bandwidth, b) user specific fields to show the position of the user field, and c) the RU used to transmit STAs data which supports the multiplexing of users using MU-MIMO. In 802.11ax standard, subcarriers transmit signals by non-HT, HT and VHT, which details are as follows [1]; for a 20MHz non OFDMA or OFDMA HE PPDU transmission, the 20 MHz is divided into 256 subcarriers; for a 40 MHz non OFDMA or OFDMA HE PPDU transmission, the 40 MHz is divided into 512 subcarriers; the 80 MHz non OFDMA or OFDMA HE PPDU transmission is divided into 1024 subcarriers. In the case of a non-contiguous 80+80 MHz transmission, each is divided into 1024 subcarriers, identical to that of a singular 80MHz HE PPDU transmission. As a result, new reconfigurable FFT architecture that can support devices using 802.11ax must be designed.

This architecture can compliant implement with many modes: 8 channels of FFT 256 points; 2 channels of FFT 1024 points; 4 channels of FFT 512 points; 1 channel of FFT 2048 points. Depend on the mode of operation of the device. Furthermore, this flexible FFT architecture has high throughput, high efficiency of memory and low latency. This paper presents the combination of implementation of Radix 2^4 SDF and radix 2 MDC pipeline FFT processor on FPGA for 802.11ax. The paper is organized as follows: Section I is an overview of WLAN and WLAN 802.11ax, Section II gives a brief review of FFT operation and Section III discusses the FFT architecture. Section IV presents the results of FFT implementation and Section V is the conclusion.

II. FFT ALGORITHMS AND ARCHITECTURES

A. IDFT/DFT

Signal of Fourier transform is decomposed into orthogonal frequency so we can use this transform to perform modulation and demodulation OFDM based Wireless LAN in 802.11ax. According to [13], N-point Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) of an input sequence is defined as (1):

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad (1)$$

Where $k = 0, 1, 2, N-1$ and $W_N^{nk} = e^{-\frac{2\pi j}{N}nk}$ is known as the twiddle factor for DFT, $W_N^{nk} = e^{\frac{2\pi j}{N}kn}$ is known as the twiddle factor for IDFT.

Direct implementation (1) needs N^2 complex multiplications and $N(N-1)$ complex additions. As proposed in [10], FFT algorithm can compute the DFT/IDFT by decomposing the input sequence into smaller sized DFT/IDFT. There are two basics of decomposing which correspond to two algorithm FFT: decimation in time (DIT) FFT and decimation in frequency (DIF) FFT. Implementation of these basic algorithm

FFT reduce to the number of complex multiplications and complex additions respectively is $\left\lceil N \cdot \frac{r-1}{r} \right\rceil [\log_r N - 1]$ and $N \cdot \log_r N$ where r is the number of sequence decomposed (radix of FFT). In fact, implementation FFT often uses DIF FFT because we do not need to reverse data before feeding them into input, we will reorder data after finishing the FFT processing.

B. FFT/IFFT architecture

Reducing number of multiplications and additions based on radix FFT is negligible. The Cooley-Turkey algorithm [13] proposes a divide to partition a DFT of length $N = M \cdot L$ into smaller DFT of length M and L such as:

$$X(k) = \sum_{l=0}^{L-1} \left\{ \sum_{m=0}^{M-1} [x(l, m) \cdot W_M^{mq} \cdot W_N^{lq}] \cdot W_L^{lp} \right\} \quad (2)$$

Equation (2) calculation of $X(k)$ is implemented follow three steps: 1) compute M-point DFT, 2) multiply the outputs by twiddle factor, and 3) compute L-point DFT. So that, number of multiplications and addition remain $N \cdot (M + L + 1)$, $N \cdot (M + L - 2)$ respectively. Symmetry and periodicity of the twiddle factor can result in efficient DFT computation. As a results, all multipliers can be implemented as constant multiplies to reduce area and power.

C. SDF, MDC architectures

Pipeline architectures have the advantage of parallelism and pipelining. These architectures can perform very quickly but require more hardware complexity. There are many pipelines such as MDC, SDF architecture.

The radix 2 MDC is pipeline implementation of radix 2 FFT algorithm. In this architecture, the input sequence is divided into two parallel data streams by a commutator and then with the proper delay of one of two streams, the butterfly operation and twiddle factor multiplication are carried out [5]. Generally, using radix r MDC FFT, the input sequence is divided into r parallel data streams by a r input commutator and then with proper delays of each streams. An MDC stage consists of a radix r butterfly, which includes $(r-1)$ complex multipliers and two sets of shift registers in this architecture. There are many registers on each data stream and using higher radix butterfly we must design more a complicated commutator and more hardware. SDF pipeline FFT has one path between stages as shown in [5]. There are pipeline feedback registers in each stage. These registers store previous stage outputs for use by butterfly. Each SDF stage is comprised of a radix r butterfly followed by a complex multiplier, with the exception of the last stage, and shift registers to hold intermediate values. Both of these architectures have the same number of butterflies and multipliers. However, MDC architecture has more complex commutators and more registers than SDF. Single path delay Feedback FFT architectures have the most efficient memory utilization for pipeline FFT processors.

Due to the exponential growth of the number of memory words required with respect to the number of FFT stages, there

TABLE I: Value of n,k in FFT architecture

Point	n	k	Radix
2048	$n_4 + 16n_3 + 256n_2 + 512n_1 + 1024n_0$	$k_0 + 2k_1 + 4k_2 + 8k_3 + 128k_4$	3 stages radix 2 and 2 stages radix 2^4
1024	$n_4 + 16n_3 + 256n_2 + 512n_1$	$k_1 + 2k_2 + 4k_3 + 64k_4$	2 stages radix 2 and 2 stages radix 2^4
512	$n_4 + 16n_3 + 256n_2$	$k_2 + 2k_3 + 32k_4$	1 stages radix 2 and 2 stages radix 2^4
256	$n_4 + 16n_3$	$k_3 + 16k_4$	2 stages radix 2^4

will always be a point at which memory dominates circuit area and power consumption. As results, larger point FFT often uses SDF architecture. In addition, designing SDF for higher radix is difficult to operate because of the multiplicity of control multiplexers and utilization of registers banks. In fact, to implement higher radix butterfly we can using radix 2^k SDF architecture as proposed in [5]. Using these architectures, it is simple to implement larger numbers of point of FFT and decrease the number of multipliers and adders, which results in a smaller area.

D. Reconfigurebale FFT

Using the advantages of the Cooley-Turkey algorithm[13], MDC and SDF architecture design FFT. We implement 2048 point FFT, 1024-point FFT, 512 point FFT and 256 point FFT by deconstructing(1) by using radix 2 and radix 2^4 butterfly. In addition, for 512 point FFT, a radix 2 butterfly is added from the 256 point FFT cores. This method is also applied to the other FFT sizes. Table I shows the values for n and k for each FFT size based on the radix. The equation for 2048 point IFFT/FFT is shown as in (3); here, the value for n and k is constructed based on the radix configuration as in (2). As a result of the substitution of n and k, we can obtain the equation for 2048 point FFT as shown in (3). Similar to FFT 2048, we will obtain the formulas for 1024 point FFT, 512 point FFT and 256 point FFT by substituting the values as shown in Table I. Equation for IFFT/FFT 2048 point shown as (1): Then n and k value of (1) is decomposed based on radix configuration based on (2). These n, k are chosen from Table I. By substituting each value correspond number of point of FFT we obtain equation for FFT 2048 point and FFT 1024 point as shown in (3), (4). Similar FFT 2048 we will obtain formulas for FFT 1024, FFT 512, FFT 256 point.

$$X(k) = \sum_{n_4=0}^{15} \sum_{n_3=0}^{15} \sum_{n_2=0}^1 \sum_{n_1=0}^1 \sum_{n_0=0}^1 x(n_4 + 16n_3 + 256n_2 + 512n_1 + 1024n_0) W_{2048}^{nk} \quad (3)$$

$$X(k) = \sum_{n_4=0}^{15} \sum_{n_3=0}^{15} \sum_{n_2=0}^1 \sum_{n_1=0}^1 x(n_4 + 16n_3 + 256n_2 + 512n_1) W_{1024}^{nk} \quad (4)$$

Equation derivation of FFT 2048 point uses 5 stages in which radix butterfly shown in Table I. Generally, we can prove for FFT 1024, 512, 256 point. To prove these equation are used each stage we expand these equations for FFT 1024 point using only 4 stages from(2^{nd} stage to 5^{th} stage) as follow:

1) $2^{nd}, 3^{rd}$ stage: During these stages, we use radix 2 so we can determine that, n_1 has two values 0 and 1. Using the values from the corresponding number point of FFT from Table I, substitute the value of n, k into the equation for FFT 1024 point and then expand based on and then expand based on n_1 . Equation of FFT 1024 point transforms as shown in (5) for use in the 2^{nd} in our design [8]. This is done by generating the twiddle factor for use in the multiplier in the 2^{nd} stage, 3^{rd} stage as shown in (5), (6) respectively.

$$X(k) = \sum_{n_4=0}^{15} \sum_{n_3=0}^{15} \sum_{n_2=0}^1 X_2(n_4 + 16n_3 + 256n_2 + 512k_1) \cdot W_{512}^{(n_4+16n_3+256n_2)(k_2+2k_3+32k_4)} \quad (5)$$

$$X(k) = \sum_{n_4=0}^{15} \sum_{n_3=0}^{15} X_3(n_4 + 16n_3 + 256k_2 + 512k_1) \cdot W_{256}^{(k_3+16k_4).(n_4+16n_3)} \quad (6)$$

where X_2 and X_3 are defined as (7), (8):

$$X_2(n_4 + 16n_3 + 256n_2 + 512k_1) = \{x(n_4 + 16n_3 + 256n_2) + x(n_4 + 16n_3 + 256n_2 + 512) \cdot W_{1024}^{k_1.(n_4+16n_3+256n_2)}\} \quad (7)$$

$$X_3(n_4 + 16n_3 + 256k_2 + 512k_1) = \{X_2(n_4 + 16n_3 + 512k_1) + X_2(n_4 + 16n_3 + 256 + 512k_1) \cdot W_{512}^{k_2.(n_4+16n_3)}\} \quad (8)$$

2) $4^{th}, 5^{th}$ stage: Expanding (6) based on butterfly radix 2^4 we achieve (9) for 4^{th} stage. Then by expanding (9) we have (10) for 4^{th} stage. Parameters δ, σ, α and β as proposed in [7]:

$$X(k) = \sum_{n_4=0}^{15} \{ \sum_{\sigma_2=0}^1 \{ \sum_{\sigma_3=0}^1 \{ \sum_{\sigma_4=0}^1 \{ X_3(n_4 + 16.(8\sigma_1 + 4\sigma_2 + 2\sigma_3 + \sigma_4) + 256k_2 + 512k_1) \cdot W_2^{\sigma_1.\delta_1} \cdot (-j)^{\sigma_2\delta_1} \cdot W_2^{\sigma_2.\delta_2} \cdot W_{16}^{(\delta_1+2\delta_2)(2\sigma_3+\sigma_4)} \cdot W_2^{\sigma_3.\delta_3} \cdot (-j)^{\sigma_4\delta_3} \cdot W_2^{\sigma_4.\delta_4} \cdot W_{256}^{(\delta_1+2\delta_2+4\delta_3+8\delta_4)} \cdot W_{16}^{n_4.k_4} \} \} \} \} \quad (9)$$

$$X(k) = \sum_{\alpha_1=0}^1 \{ \sum_{\alpha_2=0}^1 \{ \sum_{\alpha_3=0}^1 \{ \sum_{\alpha_4=0}^1 X_4(8\alpha_1 + 4\alpha_2 + 2\alpha_3 + \alpha_4 + 16.(8\sigma_1 + 4\sigma_2 + 2\sigma_3 + \sigma_4) + 256k_2 + 512k_1) \cdot W_2^{\beta_1.\alpha_1} \cdot (-j)^{\beta_1\alpha_2} \cdot W_2^{\beta_1.\alpha_2} \cdot W_{16}^{(\beta_1+2\beta_2)(2\alpha_3+\alpha_4)} \cdot W_2^{\beta_3.\alpha_3} \cdot (-j)^{\beta_3\alpha_4} \cdot W_2^{\beta_4.\alpha_4} \} \} \} \} \quad (10)$$

These help us prove all equation used for FFT 2048 point, FFT 512 point and FFT256 point. Based on the value of k, n we can calculate index of output to re-order the signals of FFT processor correctly.

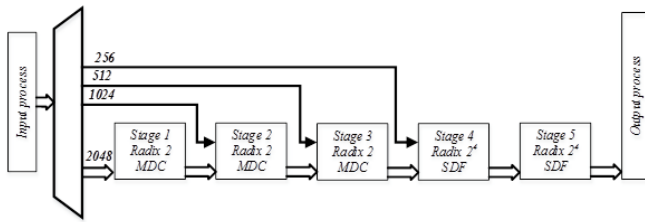


Fig. 1: Choose multimode.

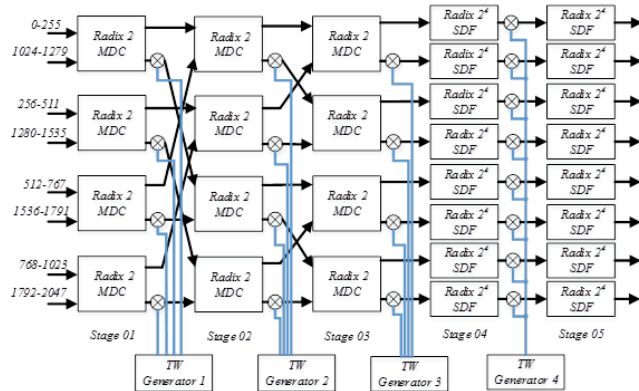


Fig. 2: System architecture.

III. MULTIMODE FFT

As mentioned above, all 802.11ax devices have many modes to create HE PPDU. This standard requires implementation of FFT 1024/256 and optional 2048/512. From the equations proved in Section II, we can design multimode FFT as shown in Fig. 1: implementation of FFT 1024 uses stages 2nd, 3rd, 4th and 5th and implementation of FFT 256 only uses stage 4th, 5th. Implementation of FFT 2048 have to use all of stages(from 1st stage to 5th stage). The choice for the FFT mode is based on control signal that selects correct input, output and circuit. To enhance efficiency and throughput of FFT processor, we proposed mix parallel, MDC and SDF architecture. In the scope of this paper we will analysis dual mode FFT 2048/512 in 802.11ax devices as in Fig. 2. Although parallel structures has high throughput but we have mention about decrease area and low latency so we use parallel architecture, radix 2 butterfly in three first stages and radix 2⁴ butterfly in the last two stages. FFT 2048, 1024, 512, 256 point architectures always use the 4th, 5th stages so we can decrease the area in the FFT. Fig. 2 shows that all index data needed in the first stage are available at the first clocks. The output of the first stage can be fed into directly 2nd stage and output of the 2nd stage can fed be into directly 3rd stage. Since there is no RAM used to buffer data out from the 1st, 2nd and 3rd stages. This architecture has high memory efficiency. In the 2 last stages, 8 parallel radix 2⁴ SDF architecture is used, with radix 2⁴ butterfly being applied with shift registers and adders. The architecture in the last stage does not use any complex

multiplier as there are any multiplications with twiddle factor.

In addition, based on the pipeline SDF architecture, a multimode FFT architecture can be implemented by cascading several radix 2^k stages in order to accommodate different FFT sizes. The signal-flow graph for radix 2⁴ is proposed in[6]. In this paper, we show four basic processing units (PU) which include basic butterfly and constant multipliers. All the intra-stage multipliers inside the elements for radix 2⁴- point FFT are constant multipliers. Full multipliers are only used for inter-stage twiddle factors. Since the inter-stage full multipliers cost more than the intra- stage constant multipliers, radix factorization should minimize the number of full multipliers. Radix 2⁴ is performed by connecting processing units together [6].

Furthermore, when using modes that do not require all blocks that are unused can be turned off to conserve energy. Designing the control signal for dual mode FFT is importance to achieve efficient FFT, as well as, enabling us to control many modes operating simultaneously. In our design, we can operate by selecting mode 1 to implement one FFT 2048, mode 2 to implement four FFT 512, mode 3 to implement two FFT 1024, mode 4 to implement eight FFT 256...depend on the system. Implementation complex multipliers with twiddle factors between of stages shown in above equation correspond each stage. Choosing this architecture does not waste performing of multiplier because changing mode does not change architecture that ignore some of blocks or stages in our architecture.

Our proposed architecture uses DIF algorithm so that input data does not need reordered but output data must be reordered correctly based on FFT function model. To reorder the output data of FFT, all data from 5th stage output is needed. Therefore, the clock latency of FFT with reorder unit become twice as long compared to the latency of FFT operation in the 1st stage until the 5th stage.

IV. RESULTS

Numerous parallel configuration of Radix 2 and Radix 2⁴ SDF were synthesized to observe the effects of parallelism on the throughput, area and power consumption of the circuit. The proposed FFT architecture was verified functionally and synthesized in Quartus II. Table II shows the clock latency and area of FFT without reorder. The table III shows the number of logic gates and memories used in the architecture proposed. The results of multimode FFT is compared to results of FFT function Matlab with error around 10⁻⁶ and using multipliers and adders with fraction 16. To have this results we have to simulate our design 10000 times in Matlab 2014 with random data inputs.

V. CONCLUSION

In this paper, a multimode FFT processor that is designed with high throughput and efficient area 256/512/1024/2048-point FFT for optional WLAN 802.11ax and 256/1024 for standard WLAN 802.11ax is proposed. A novel sub channel based on OFDMA random access scheme in WLAN 802.11ax needs the multimode FFT proposed. The use of constant

TABLE II: Latency and throughput

Point	Design [8]	Design [15]	Proposed design
Clock	530	4096	299
2048 point			without re-order
Frequency / Device	67.28 MHz Altera Stratix IV EP4SGX530KH40C3	69.36 MHz Altera Stratix II EP2S60F1020C3	111 MHz Altera Stratix IV EP4SGX530KH40C3

TABLE III: Numbers of logic stages

Parameters	Symphony Design	Propose Design
LUT	92855	80088
Memory Bits	238204	116992
I/O pins	713	616
Total Registers	78090	47129

multipliers for intra-stage twiddle factors enables a decrease in area and power savings compared to the use of full multiplications. Using radix 2^4 creates a larger number of constant multiplications that significantly reduce the number of full multiplications. The radix 2^4 single path delay used in this architecture makes some advantages that reduce the number of butterflies and multipliers. Multimode FFT proposed is efficiently area.

The results of multimode FFT is compared to results of FFT function Matlab. This architecture of FFT has been designed using Symphony and verified based on fixed point using Quartus II. Synthesis result with FPGA and calculate area using ASIC.

REFERENCES

- [1] Robert Stacey, *doc: IEEE 802.11-15/0132r8*, TGax Spec Framework, September 2015.
- [2] Matthew S. Gast, *802.11 ac: A survival Guide*, published by O'Reilly, 2015.
- [3] Matthew S. Gast, *802.11 n: A survival Guide*, published by O'Reilly, 2013.
- [4] Herbert L. Groginsky and George A. Works, *A pipeline fast fourier transform*, IEEE TRANSACTIONS ON COMPUTERS, VOL. C-19, NO. 11, NOVEMBER 1970.
- [5] Tzi-Dar Chiueh and Pei-Yun Tsai, *OFDM baseband receiver design for wireless communications*, Jony Willey and Sons (Asia), pp195-232, 2007.
- [6] Chia-Hsiang Yang and Tsung-han Yu and Dejan Markovic, *Power and area minimization of reconfigurable FFT processors: A 3 GPP-LTE Example*, IEEE Journal of Solid-State circuits, VOL.47, No3, March 2012.
- [7] Song-Nien Tang, Chi-Hsiang Liao and Tsin-Yang Chang, *An area and energy-efficient multimode FFT processor for WPAN/WLAN/WMAN systems*, IEEE Journal of Solid-State circuits, VOL.47, No6, June 2012.
- [8] Trio Adiono and Rella Mareta, *Low latency parallel-pipeline configurable FFT-IFFT 128/256/1024/2048 for LTE*, ICIAS 2012 4th.
- [9] Nguyen Hung Cuong, Nguyen Tung Lam and Nguyen Duc Minh, *Multiplier-less based architecture for variable-length FFT hardware implementation*, Communications and Electronics (ICCE), 2012 Fourth International Conference.
- [10] Yuan-Chu, Yu and Yuan-Tse Yu, *Design of a high efficiency reconfigurable pipeline processor on next generation portable device*, 978-1-4799-1616-0/13, 2013 IEEE.
- [11] Y.-T. Lin, P.-Y. Tsai and T.-D. Chiueh, *Low-power variable length fast Fourier transform processor*, IEE Proc.- Comput. Digit. Tech., vol.152, No. 4, pp.499-506, Jul.2005.
- [12] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithm, and Applications*, 3rd ed. IEnglewood Cliffs, NJ: Prentice Hall, 1996.
- [13] J.W. Cooley, J.W. Turkey, *An algorithm for machine calculation complex Fourier series*, Math. Computation, vol.19, pp.297-301, 1965.
- [14] Eun Ji kim and Myung hoon Sunwoo, *High speed eight-parallel mixed-radix FFT processor for OFDM systems*, 978-1-4244-9474-3/11, 2011 IEEE.
- [15] Adiono, T.; Fourman, D.A.S Andyes; Salman, Amy H.; *Configurable 2k/4k/8k FFT-IFFT core for DVB-T and DVB-H*, AVLSI, 2011, IEEE.



Phuong T. K. Dinh received the B.E in Radio and Communication from University of Transport and Communications, Vietnam in 2001 and M.E in Information Processing and Communications from Hanoi University of Science and Technology, Vietnam in 2006. She is currently a PhD student in Hanoi University of Science and Technology, Vietnam. From March 2016 to January 2017, she is a research student in Kyushu Institute of Technology. Her research interests include algorithms in TI-ADC as well as FFT/IFFT for wireless communication.



Leonardo Lanante Jr. received the B.S. in Electronics and Communications Engineering degree and M.S. in Electrical Engineering both from University of the Philippines in 2005 and 2007. He received his Ph.D. degree in Information Systems from Kyushu Institute of Technology in 2009 and currently an assistant professor in this university. His research interests include synchronization algorithms in wireless systems as well as signal processing in MIMO OFDM. He is a member of IEEE and IEICE.



Minh D. Nguyen obtained a PhD in Electrical Engineering from University of Kaiserslautern in 2009. He worked as a scientific staff at University of Kaiserslautern, Germany. From 2009 to 2016, he worked as Researcher and Lecture in the School of Electronics and Telecommunications at Hanoi University of Science and Technology. His research activities involve digital hardware design, embedded system design, formal verification of digital design and embedded systems.



Masayuki Kurosaki received his B.E (2000), M.E (2002) and Ph.D (2005) degrees from Tokyo Metropolitan University. He was with Kyushu Institute of Technology from 2005 to 2011 as an assistant professor. Since 2011, he has been with Kyushu Institute of Technology as an associate professor. His research interests include image processing and wireless communication for multimedia. He is a member of the IEEE.



Hiroshi Ochi received the B.S. and M.S. degree in electronics engineering from Nagaoka Institute of Technology in 1981 and 1984. He received his Ph.D. degree in electrical engineering from Tokyo Metropolitan University in 1991. He is currently a professor in computer and electronics engineering department in Kyushu Institute of Technology. His research interests include signal processing and VLSI design. He is a member of IEEE and IEICE.