

공유 메모리 기반 시스토크 어레이 FFT 프로세서 설계 및 구현

Design and Implementation Systolic Array FFT Processor Based on Shared Memory

정 동 민*, 노 윤 석*, 손 한 나*, 정 용 철*, 정 윤 호*

Dongmin Jeong*, yunseok Roh*, Hanna Son*, Yongchul Jung*, Yunho Jung*

Abstract

In this paper, we presents the design and implementation results of the FFT processor, which supports 4096 points of operation with less memory by sharing several memory used in the base-4 systolic array FFT processor into one memory. Sharing memory provides the advantage of reducing the area, and also simplifies the flow of data as I/O of the data progresses in one memory. The presented FFT processor was implemented and verified on the FPGA device. The implementation resulted in 51,855 CLB LUTs, 29,712 CLB registers, 8 block RAM tiles and 450 DSPs, and confirmed that the memory area could be reduced by 65% compared to the existing base-4 systolic array structure.

요 약

본 논문에서는 base-4 시스토크 어레이 FFT 프로세서에서 사용되는 여러 메모리를 하나의 메모리로 공유함으로써 기존 보다 작은 메모리 면적의 FFT 프로세서의 설계 및 구현 결과를 제시한다. 메모리를 공유하여 면적이 줄어드는 장점이 생겼 으며, 데이터의 입출력이 하나의 메모리에서 진행되므로 데이터의 흐름이 단순해졌다. 제시한 FFT 프로세서를 FPGA 디바이스 상에서 구현 및 검증하였으며, 구현 결과 4096-point FFT 기준 51,855개의 CLB LUT, 29,712개의 CLB registers, 8개의 block RAM tile과 450개의 DSP로 구현되었고, 최대 동작 주파수는 150MHz 인 것을 확인했으며 특히, 기존 base-4 시스토크 어레이 구조 대비 메모리 면적이 65% 감소 가능성을 확인하였다.

Key words : DFT, FFT, FPGA, radar, SAR, signal processing, systolic array

* School of Electronics and Information Engineering,
Korea Aerospace University

★ Corresponding author

E-mail : yjung@kau.ac.kr, Tel : +82-2-300-0133

※ Acknowledgment

The authors gratefully acknowledge the support from Next Generation SAR Research Laboratory at Korea Aerospace University, originally funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

Manuscript received Sep. 5, 2020; revised Sep. 18, 2020; accepted Sep. 21, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

레이다 시스템의 경우 표적의 거리와 속도를 추출하기 위한 FFT(fast fourier transform) 연산이 필수적이며, FFT 연산은 레이다 시스템에서 복잡도가 가장 큰 신호처리 중 하나이므로 효율적인 FFT 프로세서가 요구된다. 특히, SAR(synthetic aperture radar)의 영상형성 알고리즘의 경우 대용량의 영상정보를 처리해야 하므로 연산 복잡도가 높으며, 큰 분해능을 요구하는 특성으로 인해 높은 포인트의 연산을 지원해야 하고, SAR 시스템의 실시간 응용을 위해 고속의 FFT 프로세서 설계가 필수적이다.

FFT 프로세서의 하드웨어 구조에는 단일 버터플라이, 파이프라인, 시스토크 어레이 구조로 구분되는데, 단일 버터플라이 구조와 파이프라인 구조는 시스토크 어레이 구조에 비해 작은 면적으로 구현 가능하지만, 고속의 연산을 처리하기에는 적합하지 않다. 따라서, 실시간 응용의 경우 하드웨어 구조 중 가장 속도가 빠른 시스토크 어레이 기반의 병렬구조가 가장 적합하다.

다양한 시스토크 어레이 구조 중에서 base-4 기반의 시스토크 어레이 구조가 면적과 수행시간의 교환 관계(trade-off)를 가장 잘 만족한다[1-3]. 하지만, 시스토크 어레이 구조의 특성상 고속의 동작을 지원하는 만큼 많은 메모리 용량이 요구되는 문제가 있다[3-5]. 따라서, 본 논문에서는 이를 줄이기 위해 공유 메모리 구조를 사용하여 메모리 요구량을 줄인 면적-효율적인 FFT 프로세서 하드웨어 구조를 제안하고, 이의 설계 및 구현 결과를 제시한다.

본 논문의 구성은 다음과 같다. II장에서는 base-4 기반 시스토크 어레이 FFT의 알고리즘에 대해 설명하고, III장에서는 제안된 FFT 프로세서의 하드웨어 구조를 설명한다. IV장에서는 제안된 FFT 프로세서의 설계 및 구현, 검증 결과를 제시하며, 끝으로 V장에서 본 논문의 결론으로 마무리한다.

II. Base-4 기반의 시스토크 어레이 FFT 알고리즘

1. 알고리즘

base-4 시스토크 어레이 FFT 알고리즘은 1차원 행렬 데이터에 대하여 총 두 단계의 부분 분리(factorization) 과정이 수행된다. 첫 번째 부분 분리는 N 길이의 1차원 입력 데이터를 행(row) (N_r)과 열(column) (N_c)로 분리하는 것인데, 수행하고자 하는 입력 포인트 수를 N 이라 할 때 $N=N_rN_c$ 로 분리가 되며, $N_r(=N_3)$ 는 행, $N_c(=N_4)$ 는 열의 크기가 된다. 두 번째 부분 분리는 첫 번째 부분 분리를 통해 나뉜 값들을 $N_3=N_{1r}N_2$ 와 $N_4=N_{1c}N_2$ 로 분리하며, 여기서 N_2 는 base- b 에서의 b 의 값과 같다. 즉, N_r -포인트 크기의 FFT 연산을 N_c 번 수행하고, 아래의 수식 (1)에 표현한 N -포인트에 해당하는 회전 인자를 곱해준 뒤 N_c -포인트 크기의 FFT 연산을 N_r 번 수행한다.

$$W_N^{i,k} = e^{-j\left(\frac{2\pi ik}{N}\right)}, 0 \leq i < N_3, 0 \leq k < N_4 \quad (1)$$

이 과정에서 첫 번째와 두 번째 FFT는 계산 과정이 동일하고, 단순히 입력의 데이터가 행인지 열인지에 대한 차이만 있다. 이 과정을 식으로 정리하면, column-FFT를 수행할 때에는 $N_3=N_{1r}N_2$ 이고, row-FFT를 수행할 때에는 $N_4=N_{1c}N_2$ 이지만 두 식을 일반화 하여 $M=N_1N_2$ 라고 하면 FFT는 아래의 식 (2)와 (3)과 같이 정리된다 [6].

$$Y_b = W_M \cdot C_{M1} X_b \quad (2)$$

$$Z_b = C_{M2} Y_b^t \quad (3)$$

위 식(2)에서 \cdot 는 요소별 곱을 의미하고, ' b '는 'base'의 값을 의미하며, 본 논문에선 base-4를 다루기 때문에 b 는 4가 된다. X_b 는 $N_1 \times b$ 크기의 입력 데이터이고, Z_b 는 $N_1 \times b$ 크기의 FFT 출력 데이터이다. W_M 은 $N_1 \times N_1$ 크기의 행렬이며, C_{M1} , C_{M2} 는 ± 1 과 $\pm j$ 로 구성된 radix-4 butterfly 행렬의 배열을 일컫는다. C_{M1} , C_{M2} 는 M/b^2 개의 부분 행렬 $C_B=[c_1|c_2 \cdots |c_b]^t$ 로 구성되며, $C_{M1}=[C_B^t|C_B^t \cdots]^t$ 이 $C_{M2}=[C_B|C_B \cdots]$ 가 된다. 본 논문에서 b 는 4이기 때문에 c 와 C_B 값은 각각 다음 수식 (4)와 (5) 같이 정의된다.

$$c_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, c_2 = \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix}, c_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, c_4 = \begin{bmatrix} 1 \\ j \\ -1 \\ -j \end{bmatrix} \quad (4)$$

$$C_B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \quad (5)$$

위 수식(2)와 수식 (3)을 이용해 column-FFT의 계산 결과인 Z_b 에 N -포인트에 해당하는 회전인자 $W_N^{i,k}$ 를 곱해준 뒤 column-FFT와 같은 방법으로 row-FFT를 계산하면 최종 FFT 연산의 결과가 도출된다.

2. Base-b 기반의 시스토크 어레이 FFT 프로세서 구조

기존에 제안된 256 길이의 base-4 시스토크 어레이 FFT 프로세서의 하드웨어 구조는 그림 1과 같

다. 왼쪽에 16개의 PE(process element) 셀과 중간에 4개의 복소 곱셈기, 오른쪽에 16개의 PE 셀로 구성되어 있다. 이때, 왼쪽 PE 셀들의 묶음을 LHS(left hand side)라고 하며, 오른쪽 PE 셀들의 묶음을 RHS(right hand side)라고 한다. LHS와 RHS의 PE셀은 $(N_3/b) \times b$ 의 배열로 구성되며, 중간에 복소 곱셈기는 $(N_3/b) \times 1$ 의 배열로 구성된다. 그림 1에서 나타내었듯이 base-4 기반의 시스토크 어레이 FFT 프로세서의 구조에서 LHS와 RHS는 각각 4개의 열로 구성되며, 256-포인트의 경우 16×16 으로 분해되므로, LHS와 RHS의 PE셀의 행의 개수는 4개가 되고, 중간에 복소 곱셈의 PE셀 개수 또한 4개가 된다. C_{M1} 과 C_{M2} 는 앞서 설명한 것처럼 N 이 256일 때에 $M=N_3=N_4=16$ 이 되므로, 아래의 수식 (6)과 같다. 이 값은 그림 1에서 볼 수 있듯이 LHS PE셀에 C_{M1} 값이 미리 할당되어 있고, C_{M2} 값은 RHS의 아래에서 입력된다.

$$C_{M1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}^t, C_{M2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \quad (6)$$

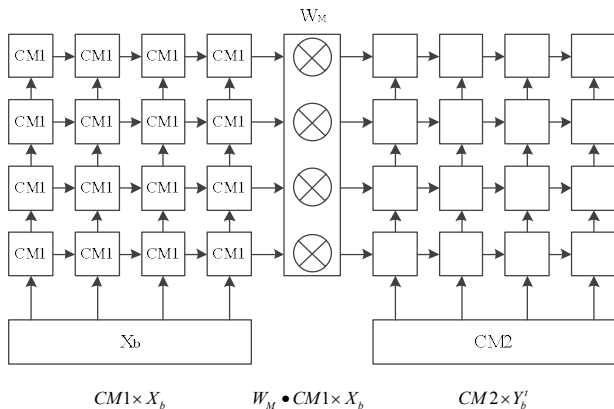


Fig. 1. Basic base-4 FFT processor structure of a 256-point. 그림 1. 256-포인트의 base-4 시스토크 어레이 FFT 프로세서의 기본 구조도

기존 base-4 시스토크 어레이 FFT 프로세서의 메모리의 경우, 그림 2와 같이 1차원의 입력 데이터들을 2차원으로 부분 분리하여 4개의 데이터가 동시에 병렬로 들어갈 수 있게 데이터를 재배치하는 input RAM(random access memory)과 column-FFT 이후 중간 결과 값을 저장할 internal RAM, row-FFT 까지 끝난 최종 결과 값을 인덱스에 맞게 re-ordering 하기 위한 output RAM으로 구성된다. in/output

RAM과 internal RAM의 크기는 N -포인트 기준 각각 N 크기만큼 필요하기 때문에 총 $3N$ 크기의 용량이 요구된다. 따라서, 이는 FFT의 포인트 수가 증가 할수록 메모리의 요구량이 커지는 문제가 있다.

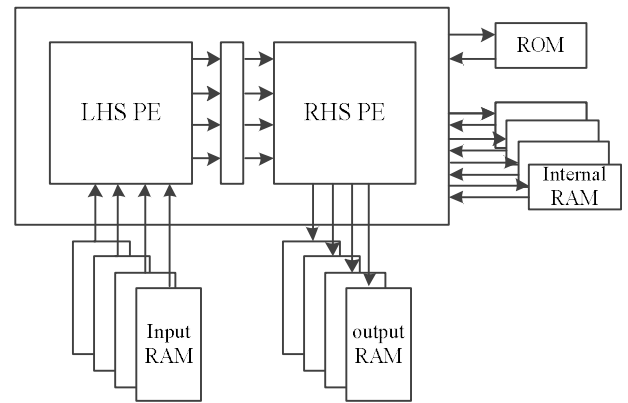


Fig. 2. Hardware structure of conventional base-4 systolic Array FFT Processor.

그림 2. 기존 base-4 시스토크 어레이 FFT 프로세서의 하드웨어 구조도

III. 제안된 FFT 프로세서의 하드웨어 구조

본 논문에서는 in/output RAM과 internal RAM을 서로 공유하여 메모리를 하나만 사용하는 하드웨어 구조를 제시한다. 제안한 FFT 프로세서의 하드웨어 구조는 기존의 FFT 프로세서에서 사용한 in/output RAM의 동작을 internal RAM에서 수행했다. 즉, 그림 3과 같이 초기 입력 데이터가 internal RAM에 저장되고 FFT 연산이 실행된 이후 나오는 column-FFT의 결과 값을 이미 사용된 초기 데이터의 주소에 re-write함으로써 메모리의 utilization을 최대로 사용하였다. 이후, row-FFT를 수행하여 나온 결과 값들을 column-FFT와 마찬가지로의 방법으로 주소 값을 컨트롤하여 저장하면 in/output RAM을 별도로 이용하지 않고 한 개의 RAM으로 동작 가능하다.

또한, 기존에는 그림 4와 같이 column-FFT 연산을 수행할 때 input RAM에서 들어온 데이터들이 LHS와 RHS의 연산 수행 후 internal RAM에 저장되고, 저장된 데이터가 row-FFT를 수행할 때에는 멀티플렉서(multiplexer)에 의해 LHS PE셀에 column-FFT의 결과 값이 저장되고, C_{M1} 값이 input RAM에서 입력되기 때문에 column, row-FFT 연산의

데이터 플로우가 다르게 진행되므로, 별도의 컨트롤러를 이용해 데이터의 흐름을 제어해주어야 한다.

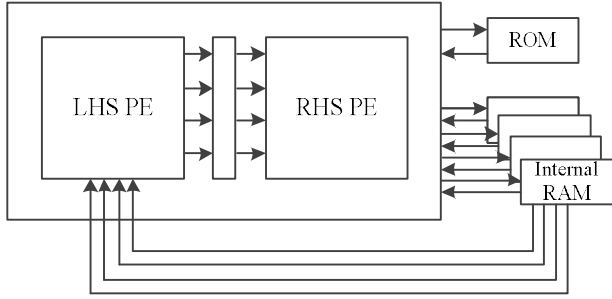


Fig. 3. Hardware structure of proposed base-4 systolic array FFT processor.

그림 3. 제안하는 base-4 시스토크 어레이 FFT 프로세서의 하드웨어 구조도

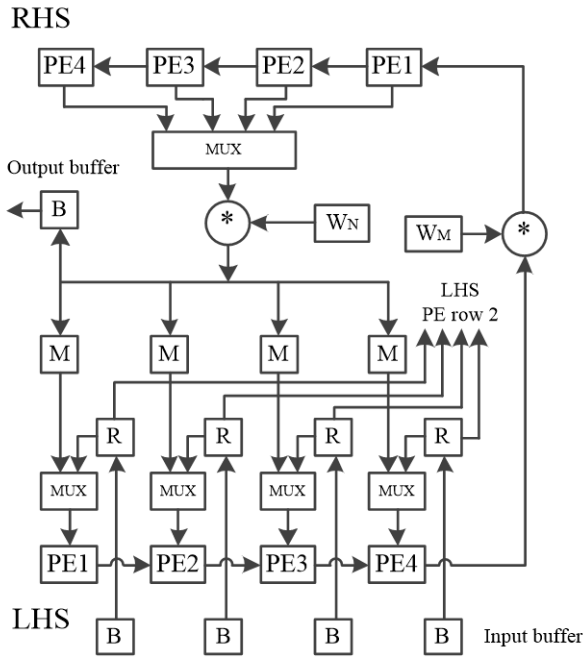


Fig. 4. Data flow of conventional base-4 systolic array FFT processor for bottom of PE row.

그림 4. PE 하단 행에 대한 기존 base-4 시스토크 어레이 FFT 프로세서의 흐름도

반면, 제안하는 FFT 프로세서의 경우 그림 5와 같이 한 개의 공유 메모리만을 이용하여 데이터가 입/출력되기 때문에 멀티플렉서의 사용 없이 column, row-FFT의 데이터 플로우가 같아진다. 따라서 기존 구조 대비 N 개의 입력 데이터 기준 \sqrt{N} 개의 멀티플렉서가 감소되었다. 또한, 기존 구조에서는 column-FFT를 진행할 때에 C_{M1} 값이 PE셀 내부에 저장되어 있어 $\pm 1, \pm j$ 중 하나만 지원하면 되지만, row-FFT의 경우 C_{M1} 값이 input RAM에서 입력되

기 때문에 한 개의 LHS PE셀이 C 가 $\pm 1, \pm j$ 인 값에 대해 모든 연산을 수행할 수 있어야 한다. 따라서, 내부 PE 셀에서도 별도의 컨트롤러를 필요로 한다. 반면, 제안하는 구조의 경우 C_{M1} 값이 항상 LHS PE셀에 저장되어 있기 때문에 각 PE 셀은 $\pm 1, \pm j$ 값들 중 한 개의 연산에 대해서만 지원하면 되어 별도의 컨트롤러 필요 없어 LHS PE 셀의 구조가 간단해진다.

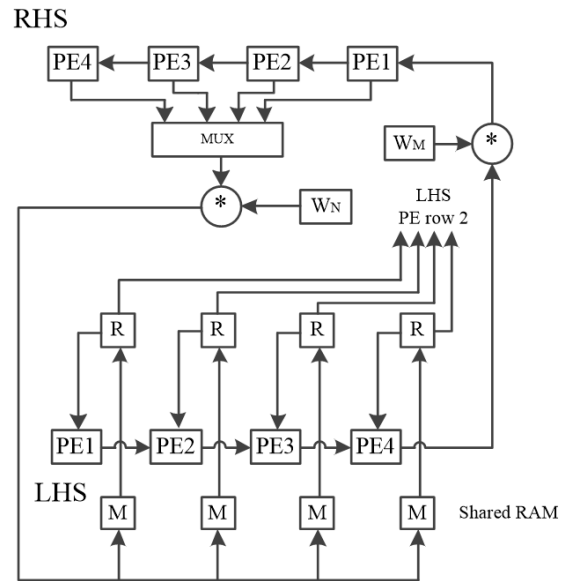


Fig. 5. Data flow of proposed base-4 systolic array FFT processor for bottom of PE row.

그림 5. PE 하단 행에 대한 제안하는 base-4 시스토크 어레이 FFT 프로세서의 흐름도

IV. 제안된 FFT 프로세서 설계 및 구현

제안된 FFT 프로세서는 Verilog-HDL을 이용해 RTL 설계 후, Xilinx의 UltraScale+ ZCU104 SoC 플랫폼을 이용하여 구현 및 검증이 진행되었다.

Table 1. Synthesis results of the proposed base-4 systolic array FFT processor.

표 1. 제안된 base-4 시스토크 어레이 FFT 프로세서의 합성 결과

Maximum Clock Speed :150MHz	
Resource	Count
LUT	51855
Flip Flop	29712
BRAM	8
DSP	450

표 1에서 제시된 바와 같이 총 51,855개의 CLB LUT, 29,712개의 CLB registers, 8개의 block RAM tile과 450개의 DSP로 구현되었으며, 최대 동작 주파수는 150MHz인 것을 확인했다. 특히, 표 2에 제시된 바와 같이 기존의 base-4 시스토크 어레이 FFT 프로세서의 경우 N -포인트 기준 메모리의 크기가 $3N+5N_3$ 개 필요한 반면 제안된 시스토크 어레이 FFT 프로세서의 구조에서는 공유 메모리를 사용해 $N+5N_3$ 개의 메모리 면적으로 구현할 수 있다. 그 결과 동일한 4096 길이의 FFT 프로세서의 경우, 기존 구조 대비 약 65% 메모리 면적 감소 효과를 보였다.

Table 2. Memory requirement of the proposed base-4 systolic array FFT processor.

표 2. 제안된 base-4 시스토크 어레이 FFT 프로세서의 메모리 요구량

Memory Requirement		
	[6]	Proposed
Input RAM	N	0
Output RAM	N	0
Internal RAM	N	N
PEs	N_3	N_3
Total	$3N+5N_3$	$N+5N_3$
Reduction Rate	65%	

V. 결론

본 논문에서는 base-4 기반 시스토크 어레이 구조의 고속 FFT 프로세서를 설계 하였으며, 기존 구조가 가지고 있는 큰 메모리 면적 요구의 단점을 공유 메모리 구조를 이용해 개선했다. 또한, row와 column-FFT의 데이터 경로를 동일하게 적용시켜, 데이터 제어에 필요한 multiplexer의 개수를 줄여 설계를 최적화 했다. 따라서, 제안된 FFT 프로세서는 공유 메모리 형태와 시스토크 어레이 FFT의 형태를 특징으로, 고속 FFT 연산의 실시간처리를 요구하는 레이다 응용에 적합할 것으로 기대된다.

References

[1] P. K. Meher, "Efficient Systolic Implementation of DFT Using a Low-Complexity Convolution-

Like Formulation," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.53, no.8, pp.702-706, 2006. DOI: 10.1109/TCSII.2006.875379

[2] Hyesook Lim and E. E. Swartzlander, "Multi-dimensional systolic arrays for multidimensional DFTs," *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol.6, pp.3276-3279, 1996. DOI: 10.1109/ICASSP.1996.550576

DOI: 10.1109/ICASSP.1996.550576

[3] J. G. Nash, "Computationally efficient systolic architecture for computing the discrete Fourier transform," in *IEEE Transactions on Signal Processing*, vol.53, no.12, pp.4640-4651, 2005. DOI: 10.1109/TSP.2005.859216

DOI: 10.1109/TSP.2005.859216

[4] J. G. Nash, "A High Performance Scalable FFT," *2007 IEEE Wireless Communications and Networking Conference*, pp.2367-2372, 2007. DOI: 10.1109/WCNC.2007.442

DOI: 10.1109/WCNC.2007.442

[5] J. G. Nash, "A New Class of High Performance FFTs," *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP '07*, vol.2, pp.22-24, 2007. DOI: 10.1109/ICASSP.2007.366162

DOI: 10.1109/ICASSP.2007.366162

[6] J. Greg Nash, "High-throughput programmable systolic array FFT architecture and FPGA implementations," *2014 International Conference on Computing, Networking and Communications (ICNC)*, pp.878-884, 2014. DOI: 10.1109/ICCNC.2014.6785453

DOI: 10.1109/ICCNC.2014.6785453

BIOGRAPHY

Dongmin Jeong (Member)



2015~present : MS degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Yunseok Roh (Member)

2015~present : BS degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Hanna Son (Member)

2017~present : BS degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Yongchul Jung (Member)

2015 : BS degree in School of Electronics and Information Engineering, Korea Aerospace University.

2017 : MS degree in School of Electronics and Information Engineering, Korea Aerospace University.

2017~present : Ph.D degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Yunho Jung (Member)

1998 : BS degree in Department of Electrical and Electronic Engineering, Yonsei University.

2000 : MS degree in Department of Electrical and Electronic Engineering, Yonsei University.

2005 : Ph.D degree in Department of Electrical and Electronic Engineering, Yonsei University.

2005~2007 : Senior Engineer, Samsung Electronics.

2007~2008 : Research professor, Institute of Information Engineering, Yonsei University.

2008~present : Professor, School of Electronics and Information Engineering, Korea Aerospace University.