

OFDM FFT용 저전력 Radix-4 나비연산기 구조

*김도한, 김비철, 허은성, 이원상, 장영범

상명대학교 정보통신공학부

e-mail : doha0801@smu.ac.kr, beechul@smu.ac.kr, heaven81@smu.ac.kr,
windstorm5@smu.ac.kr, ybjang@smu.ac.kr

Low-Power Radix-4 butterfly structure for OFDM FFT

*Do-Han Kim, Bee-Chul Kim, Eun-Sung Hur, Won-Sang Lee, Young-Beom Jang
Department of Information and Telecommunication Engineering,
Sangmyung University

Abstract

In this paper, an efficient butterfly structure for Radix-4 FFT algorithm using DA(Distributed Arithmetic) is proposed. It is shown that DA can be efficiently used in twiddle factor calculation of the Radix-4 FFT algorithm. The Verilog-HDL coding results for the proposed DA butterfly structure show 61.02% cell area reduction comparison with those of the conventional multiplier butterfly structure. Furthermore, the 64-point Radix-4 pipeline structure using the proposed butterfly and delay commutators is compared with other conventional structures. Implementation coding results show 46.1% cell area reduction.

I. 서론

DMB와 같은 고속 멀티미디어 시스템에서는 대역효율성이 우수한 OFDM 방식을 사용하고 있으며, OFDM 전송방식은 IFFT와 FFT로 구현이 가능한데, DMB용 OFDM에서는 2048 point의 FFT를 필요로 하므로 FFT 블록의 구현 비용과 전력소모를 줄이는 것이 핵심사항이라고 할 수 있다. 지상파 DMB용 OFDM에서는 246μs 동안에 2048 point FFT를 수행하여야 하므로 높은 처리율을 갖는 Radix-4 FFT 알고리즘이 주로 사용된다. 본문에서는 Radix-4 FFT 알고리즘 기반의 파이프라인 구조에 속하는 저전력 FFT 구조를 제안하고, 제안된 구조의 시뮬레이션 결과와 면적의 비교를 확인한다.

II. 본론

기존의 Radix-4 나비연산[1]에서 4개의 입력과 출력은

각각 복소수로 나타나며, W^n, W^{2n}, W^{3n} 도 모두 직각형 복소수로 다음과 같이 나타낼 수 있다.

$$W^n = e^{-j\frac{2\pi n}{N}} = \cos\frac{2\pi n}{N} - j\sin\frac{2\pi n}{N} = C_b + j(-S_b)$$

$$W^{2n} = e^{-j\frac{4\pi n}{N}} = \cos\frac{4\pi n}{N} - j\sin\frac{4\pi n}{N} = C_c + j(-S_c)$$

$$W^{3n} = e^{-j\frac{6\pi n}{N}} = \cos\frac{6\pi n}{N} - j\sin\frac{6\pi n}{N} = C_d + j(-S_d)$$

위의 나비연산에서 가산 연산과 Twiddle factor가 연산되어진 후 출력되는 각각의 출력 결과 값 $x'_a, y'_a, x'_b, y'_b, x'_c, y'_c, x'_d, y'_d$ 을 DA구조[2]에 적용 가능하도록 재정리 하면 아래의 식을 유도할 수 있다.

$$x'_a = x_a + x_b + x_c + x_d$$

$$y'_a = y_a + y_b + y_c + y_d$$

$$x'_b = (x_a + y_b - x_c - y_d)C_b - (y_a - x_b - y_c + x_d)(-S_b) \\ = x_1 C_b - x_2 (-S_b)$$

$$y'_b = (y_a - x_b - y_c + x_d)C_b + (x_a + y_b - x_c - y_d)(-S_b) \\ = x_2 C_b + x_1 (-S_b)$$

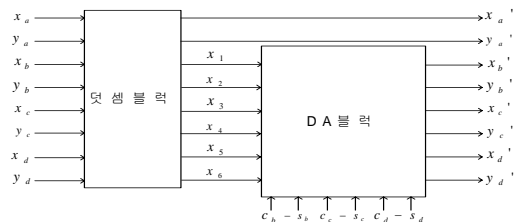
$$x'_c = (x_a - x_b + x_c - x_d)C_c - (y_a - y_b + y_c - y_d)(-S_c) \\ = x_3 C_c - x_4 (-S_c)$$

$$y'_c = (y_a - y_b + y_c - y_d)C_c + (x_a - x_b + x_c - x_d)(-S_c) \\ = x_4 C_c + x_3 (-S_c)$$

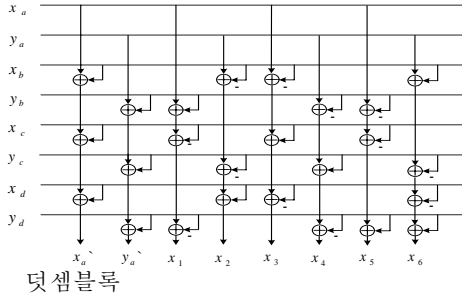
$$x'_d = (x_a - y_b - x_c + y_d)C_d - (y_a + x_b - y_c - x_d)(-S_d) \\ = x_5 C_d - x_6 (-S_d)$$

$$y'_d = (y_a + x_b - y_c - x_d)C_d + (x_a - y_b - x_c + y_d)(-S_d) \\ = x_6 C_d + x_5 (-S_d)$$

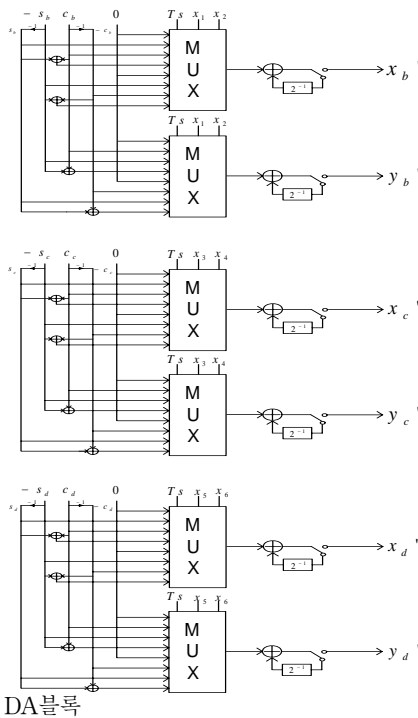
위의 식들을 아래 그림과 같이 덧셈블록과 DA블록으로 나누어 구성하는 구조를 제안하였다.



이 구조는 DA구조에의 적용을 위하여 선행계산을 위한 덧셈블록과 식의 변형 후 값을 입력시켜 트위들 벡터를 계산한 것과 같은 결과 값을 출력해주는 DA블록으로 구성되어 있으며 각각의 블록을 간단히 도식화 하면 아래 두 그림과 같이 표현된다.



덧셈블록



DA블록

III. 구현

제안된 구조의 구현면적을 Verilog-HDL 코딩을 통하여 64-point FFT의 구현면적을 비교하였다.

표 1. 제안된 구조와 기존 구조의 논리합성 결과

구분	블록명	cell area	블록 수	sub-total cell area	total cell area
제안 구조	덧셈블록	2,769	3	8,307	18,213
	DA블록	4,953	2	9,906	
승산기 구조	덧셈블록	2,131	3	6,393	46,721
	승산기블록	20,164	2	40,328	

표 1은 버터플라이/트위들 블록에 대한 논리 합성이다. 일반적으로 Radix-4의 64-point FFT는 3 스테이지로 구성되며, 스테이지 사이에 지연변환기가 사용되므로

2개의 지연변환기가 사용된다. 제안된 버터플라이/트위들 블록은 각 스테이지에서 16번의 버터플라이 연산을 반복한다. 제안 구조를 기존의 3개의 구조와 비교하였으며, 논리 합성 결과는 표 2 와 같다

표 2. Radix-4 64-point FFT 제안구조와 기존 구조들의 Cell area 비교

구분	구조 1	구조 2	구조 3	제안구조
버터플라이	44,198	27,653	46,721	18,213
지연변환기	12,170	11,853	12,170	12,170
총 cell area	56,368	39,506	58,891	30,383
상대구현면적	100%	70.9%	104.5%	53.9%

표 2의 구조 1은 Radix-4 알고리즘을 일반 승산기를 사용하여 구현한 MDC구조인 [3]에서 사용된 논리합성 결과를 참조한 것이다. 구조 2는 [3]에서 제안한 Radix-4와 Radix-2를 함께 사용하는 알고리즘에 대한 결과를 참조한 것이다. 이 구조는 64-point FFT의 구현에 지연변환기를 3개 사용하고 있다. 구조 3은 이 논문에서 제안된 구조를 따르되 DA를 사용하지 않고 일반승산기를 사용한 결과이다.

IV. 결론 및 향후 연구 방향

본문에서는 64-point Radix-4 알고리즘의 저전력 파이프라인 구조를 제안하였다. 파이프라인 FFT 구조는 버터플라이/트위들 블록과 지연변환기로 구성되는데, 이 논문은 버터플라이/트위들 블록을 덧셈블록과 DA블록으로 구현하는 저전력 구조를 제안하였다. 제안된 버터플라이/트위들 블록은 기존의 일반 승산기를 사용하는 버터플라이/트위들 블록과 비교하여 61.02%의 cell area 감소효과를 볼 수 있었다. 또한 지연변환기를 포함한 제안된 64-point FFT 블록은 기존의 64-point FFT 블록과 비교하여 46.1%의 cell area 감소효과를 나타내었으며 2048-point FFT와 같은 큰 크기의 FFT에서는 더욱 cell area 감소효과를 나타낼 것이다. 따라서 제안된 DA방식의 FFT 구조는 DMB용 OFDM 모델과 같은 큰 크기의 FFT를 요구하는 시스템에서 사용될 수 있는 효율적인 구조이다.

참고문헌

[1] Beven M. Baas, "A 9.5mW 330us 1024-point FFT Processor", IEEE Custom Integrated Circuits Conference, pp. 127-130, 1998.
 [2] Stanley A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A tutorial Review", IEEE ASSP MAGAZINE July 1989.
 [3] 정운호, 김재석, "고속 멀티미디어 통신시스템을 위한 효율적인 FFT 알고리즘 및 하드웨어 구현", 전자공학회논문지 제41권 SD편 제3호, pp. 55-64, 2004년 3월.