

Wireless PAN기반의 근거리 해상통신용 OFDM 송수신회로에 관한 연구

OFDM System for Wireless-PAN related short distance Maritime Data Communication

조승일* · 차재상** · 박계각*** · 양충모**** · 김성권*[©]

Seung-Il Cho, Jae-Sang Cha, Gye-Kack Park, Chung-Mo Yang and Seong-Kweon Kim

* 목포해양대학교 대학원 해양전자통신공학과

** 서울산업대학교 매체공학과

*** 목포해양대학교 해상운송시스템학부

**** 삼성전기 중앙연구소

요 약

유비쿼터스 네트워크의 실현을 위한 4세대 통신방식의 유력한 후보로 부상하는 OFDM (Orthogonal Frequency Division Multiplexing) 통신방식이 육상에서 주목받고 있으며, 고속 데이터 전송을 위한 Wireless LAN의 표준기술로 확정되어 있다. 해상통신의 경우에서도 OFDM 통신방식은 VHF 대역을 이용한 데이터 전송방식으로 제안되고 있으며, ITU (International Telecommunication Union)는 해상통신에서 32-point FFT (Fast Fourier Transform)를 사용하도록 권고하고 있다. 해상 통신에서는 해양사고 및 조난 시에도 통신이 이루어져야 하는 한계상황을 고려하면, OFDM 통신시스템의 중요 디바이스인 FFT는 저전력으로 동작되어야 한다. 따라서 본 논문에서는 OFDM 방식의 중요 디바이스인 32-point FFT를 저전력으로 동작시키기 위해 radix-2와 radix-4를 이용하여 저전력 32-point FFT 알고리즘을 제안한다. 최적화된 설계로 32-point FFT를 저전력 동작이 가능하도록 설계하였으며, 제안한 알고리즘은 VHDL (VHSIC hardware description language)로 구현하고 FPGA (field-programmable gate array) Spartan3 board에 장착하여 Matlab의 이론값과 비교, 검증하였다.

키워드 : OFDM (Orthogonal Frequency Division Multiplexing), 단파대, 저전력, 32-point FFT

Abstract

Orthogonal Frequency Division Multiplexing (OFDM) has been focused on as 4th generation communication method for realization of Ubiquitous Network in land mobile communications services, and has been a standard technology of Wireless Local Area Network (WLAN) for a High Data Rate communication. And in maritime data communication using high frequency (HF) band, 32-point FFT OFDM system is recommended by International Telecommunication Union (ITU). Maritime communication should be kept on connecting when maritime accident or the maritime disaster happen. Therefore, main device FFT should be operated with low power consumption.

In this paper we propose a low power 32-point FFT algorithm using radix-2 and radix-4 for low power operation. The proposed algorithm was designed using VHSIC hardware description language (VHDL), and it was confirmed that the output value of Spartan-3 field-programmable gate array (FPGA) board corresponded to the output value calculated using Matlab. The proposed 32-point FFT algorithm will be useful as a leading technology in a HF maritime data communication

Key Words : OFDM (Orthogonal Frequency Division Multiplexing), HF(High Frequency), low-power, 32-point FFT

1. 서 론

유비쿼터스 시대를 지향하는 현대 사회는 음성, 사진, 동영상 등의 멀티미디어 통신 및 고속 데이터 통신에 대한 수

요가 증가함에 따라 다양한 무선통신 방식의 출현과 시스템 광대역화 현상이 두드러지게 나타나고 있다. 차세대 무선통신기술 개발에 대한 연구가 활발한 가운데 OFDM (Orthogonal Frequency Division Multiplexing) 기술은 4세대 통신방식 중 하나의 후보로 주목받고 있다 [1]. 또한 유비쿼터스 네트워크 실현을 위한 다양한 통신방식의 연구 가운데, 근거리 통신방식으로 대표되는 Wireless PAN 등의 다양한 기술이 소개되며, OFDM 통신방식의 적용이 또한 적극 검토되고 있다.

OFDM은 다중 반송파 전송 기술의 범주에 속하며, 변조

접수일자 : 2008년 4월 5일

완료일자 : 2008년 10월 8일

본 연구내용의 일부는 2008년도 정부 (과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임. (No. R01-2006-000-11183-0)

기술 혹은 다중화 기술로 정의된다. OFDM의 핵심 기술은 기존의 데이터를 하나의 반송파로 전송하는 것이 아니라 직렬로 입력되는 데이터 열을 N 개의 병렬 데이터 열로 변환하여 직교성 (orthogonality)을 보장하는 많은 수의 반송파 (carrier)에 데이터를 분산하여 전송하는 것이다. 단일 반송파 시스템에서는 하나의 페이딩 신호 및 간섭신호가 전체의 링크 성능에 영향을 줄 수 있지만 OFDM은 일부분의 부반송파에만 페이딩 영향을 받게 되어 에러 정정 기술에 의한 정정을 통해 성능 개선 효과를 얻을 수 있으며, 다중경로 채널에 강한 심플을 생성하여 멀티패스(multi-path) 및 이동수신 환경에서 우수한 성능을 발휘하기 때문에 지상과 디지털 TV 및 디지털 음성 방송에 적합한 변조방식으로 응용되고 있다. OFDM은 주로 통신 분야에서 연구가 진행되어 왔으나 EBU(European Broadcasting Union)가 제안한 디지털 음성 방송 시스템의 변조방식으로 채택되면서 방송 분야에서도 연구 개발이 진행되고 있으며, IEEE802.11a 및 IEEE802.11g의 작업그룹에 의하여 5GHz대역 및 2GHz대역에서 고속 데이터 전송을 위한 WLAN(Wireless Local Area Network)의 표준기술로 확정되어 있다. 특히 국내에서는 2.3GHz 대역에서 무선(Wireless)과 광대역(Broadband)이 합쳐진 WiBro라는 명칭으로 휴대인터넷 표준기술이 크게 부각되고 있다[2].

육상 통신뿐만 아니라 해상 통신에서도 OFDM 통신방식에 대해 주목되고 있다. 해상통신에서 데이터 통신을 위해 위성통신을 사용하지만, 통신료가 비싸므로 중, 소형 선박에서는 사용할 수 없기 때문에 음성 통신이 주로 사용되었던 HF (High Frequency) 및 비교적 근거리 대역 통신을 위한 VHF(Very High Frequency) 대역에서 데이터 통신을 하기 위한 연구가 활발히 진행 되고 있고[3], HF 및 VHF 대역의 주파수의 효율적인 사용과 대용량 고속 데이터 통신을 위하여 OFDM 통신방식이 추천되고 있다. 따라서, ITU (International Telecommunication Union)에서는 해상통신에서 OFDM을 적용할 경우 32-point FFT (Fast Fourier Transform)를 사용하도록 권고하였다 [4].

무선통신의 가장 큰 쟁점은 통신의 지속성과 이동성이며, 해상통신에서도 예외일 수는 없다. 특히 긴급한 대처가 요구되며, 자체 전력을 상실하는 해양사고 및 조난 시에도 통신이 이루어져야 하므로, 이러한 상황에서도 통신의 지속성과 이동성을 보장 받기 위해서는 통신시스템의 전력소비 문제를 해결해야 한다.

OFDM의 부반송파는 직교성을 유지할 수 있도록 IFFT (Inverse Fast Fourier Transform)와 FFT 프로세서를 이용하여 신호를 변조하기 때문에 고성능의 FFT 프로세서를 구현하는 것이 OFDM 방식의 고속 무선 데이터 통신을 구현하기 위한 핵심 사항이라 할 수 있다. 또한 FFT/IFFT 프로세서는 OFDM 방식의 물리계층에서 가장 큰 면적과 전력을 소모한다 [6-9]. 따라서 저전력 OFDM 해상통신시스템을 위해서는 저전력 32-point FFT 구현이 필요하다.

본 논문에서는 Wireless PAN 기반에서의 근거리 해상 통신을 위한 OFDM 적용시스템을 구현하기 위하여 국제적으로 제안되고 있는 32-point OFDM 시스템을 겨냥하여 32-point FFT를 설계하였고, 저전력으로 동작시키기 위해 radix-2와 radix-4를 이용하여 곱셈기의 수를 줄인 저전력 32-point FFT 알고리즘을 제안한다.

2. FFT 알고리즘

2.1 FFT의 개념

푸리에 변환은 시간영역의 신호를 주파수 성분으로 변환하는 방법으로 신호가 가지고 있는 주파수 성분을 통해 스펙트럼 분석에 사용된다. 또한 신호처리 연산의 계산량을 줄이는데 푸리에 변환이 이용된다[5].

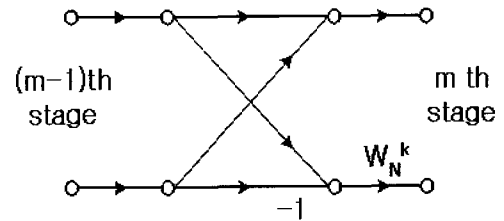


그림 1. radix-2 버터플라이 구조
Fig. 1. radix-2 butterfly structure

디지털 시스템의 이산 샘플신호의 스펙트럼 분석에는 DFT (Discrete Fourier Transform)가 사용된다. N 개의 유한한 샘플 시퀀스를 가지는 DFT 정의는 식 (1)과 같고 행렬로 나타내면 식 (2)와 같다. 여기서 N 은 이산 샘플 값의 개수이다.

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad k=0,1,\dots,N-1 \quad (1)$$

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & \dots & W_N^0 \\ W_N^0 & W_N^1 & \dots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ W_N^0 & W_N^{(N-1)} & \dots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (2)$$

N -point 시퀀스의 DFT를 직접 구현하는 계산량은 $N(N-1)$ 의 복소 덧셈과 N^2 의 복소 곱셈연산이 필요하다. DFT는 N^2 에 비례하는 방대한 계산량과 이에 따른 계산 시간 때문에 N 의 크기가 큰 경우 산술 연산을 직접 계산하기 어려워지며, 연산기 수의 증가로 H/W 크기도 커지게 된다. 그러나 회전인자 W_N 의 대칭적이고 주기적인 성질은 DFT의 계산 효율을 향상시킴으로써 이러한 문제점들을 개선할 수 있다. 1965년에 Cooley와 Tukey는 DFT 연산 중에 수행할 계산의 양을 실질적으로 줄이는 알고리즘을 개발하였다. 이러한 효율적인 알고리즘을 FFT라고 한다[6].

2.2 Radix-2 알고리즘

N 의 크기가 2의 승수일 경우, N 은 $2 \times 2 \times 2 \times \dots \times 2$ 로 인수 분해되며, N -point Cooley-Tukey 알고리즘은 radix-2 알고리즘으로 전개할 수 있다. radix-2 알고리즘구현방식을 이해하기 위해 처음 N 을 $2 \times N/2$ 으로 분해하면 $N_1=2$, $N_2=N/2$ 이 되고 n 과 k 를 식 (3)으로 나타낼 수 있으며 식 (1)은 식 (4)로 나타낼 수 있다[6].

$$\begin{aligned} n &= \frac{N}{2}n_1 + n_2, \quad \left(n_1 = 0, 1, n_2 = 0, 1, \dots, \frac{N}{2} - 1 \right) \\ k &= k_1 + 2k_2, \quad \left(k_1 = 0, 1, k_2 = 0, 1, \dots, \frac{N}{2} - 1 \right) \end{aligned} \quad (3)$$

$$\begin{aligned}
 X[k_1 + 2k_2] &= \sum_{n_2=0}^{N/2-1} \left[\sum_{n_1=0}^1 x \left[\frac{N}{2} n_1 + n_2 \right] W_2^{k_1 n_1} \right] W_N^{k_1 n_2} W_{N/2}^{k_2 n_2} \\
 &= \sum_{n_2=0}^{N/2-1} \left[x[n_2] + W_2^{k_1} x \left[n_2 + \frac{N}{2} \right] \right] W_N^{k_1 n_2} W_{N/2}^{k_2 n_2} \\
 &= \sum_{n_2=0}^{N/2-1} \widetilde{B}_{N/2}^{k_1} [n_2] W_{N/2}^{k_2 n_2} \quad (4)
 \end{aligned}$$

여기서 $\widetilde{B}_{N/2}^{k_1} [n_2]$ 는 그림 1과 같이 radix-2 버터플라이 연산을 의미한다. 첫 번째 버터플라이 연산은 $N/2$ 의 거리를 갖는 두 입력 시퀀스 간에 radix-2 버터플라이 연산을 한다. 첫 번째 버터플라이 연산 후 회전인자와 곱한다. 결국 각 열의 $N/2$ -point 변환을 수행하여 출력 시퀀스를 얻게 된다. $N/2$ -point 변환 역시 radix-2 알고리즘의 반복적인 처리를 통해 최종적으로 2-point 변환만 남을 때까지 수행하게 된다.

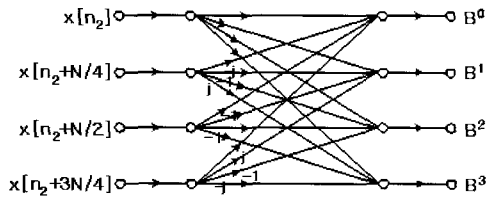


그림 2. radix-4 버터플라이 구조
Fig. 2. radix-4 butterfly structure

2.3 Radix-4 알고리즘

N 의 크기가 4의 승수일 경우, N 은 $4 \times 4 \times 4 \dots \times 4$ 로 인수 분해되며, N -point Cooley-Tukey 알고리즘은 radix-4 알고리즘으로 전개할 수 있다. 처음 N 을 $4 \times N/4$ 으로 분해하면 $N_1=4$, $N_2=N/4$ 이 되고, n 과 k 를 식 (5)로 나타낼 수 있으며 식 (1)은 식 (6)으로 나타낼 수 있다 [6].

$$\begin{aligned}
 n &= \frac{N}{4} + n_2, \quad \left(n_1 = 0, 1, \dots, 3, n_2 = 0, 1, \dots, \frac{N}{4} - 1 \right) \\
 k &= k_1 + 4k_2, \quad \left(k_1 = 0, 1, \dots, 3, k_2 = 0, 1, \dots, \frac{N}{4} - 1 \right) \quad (5)
 \end{aligned}$$

$$\begin{aligned}
 X[k_1 + 4k_2] &= \sum_{n_2=0}^{n/4-1} \left[\sum_{n_1=0}^3 x \left[\frac{N}{4} n_1 + n_2 \right] W_4^{k_1 n_1} \right] W_N^{k_1 n_2} W_{N/4}^{k_2 n_2} \\
 &= \sum_{n_2=0}^{n/4-1} \left[x[n_2] + W_4^{k_1} x \left[n_2 + \frac{N}{4} \right] + W_2^{k_1} x \left[n_2 + \frac{N}{2} \right] + W_4^{3k_1} x \left[n_2 + \frac{3N}{4} \right] \right] \\
 &\quad W_N^{k_1 n_2} W_{N/4}^{k_2 n_2} = \sum_{n_2=0}^{n/4-1} \widetilde{B}_{N/4}^{k_1} [n_2] W_{N/4}^{k_2 n_2} \quad (6)
 \end{aligned}$$

여기에서 $\widetilde{B}_{N/4}^{k_1} [n_2]$ 는 그림 2와 같이 radix-4 버터플라이 연산을 의미한다. 첫 번째 버터플라이 연산은 행의 $N/4$ 의 거리를 갖는 인접한 4개의 원소 간에 radix-4 버터플라이 연산을 한다. 첫 번째 버터플라이 연산 후 회전인자와 곱한다. 결국 각 열의 $N/4$ -point 변환을 수행하여 출력 시퀀스를 얻게 된다. $N/4$ -point 변환 역시 radix-4 알고리즘의 반복적인 처리를 통해 최종적으로 4-point 변환만 남을 때까지 수행하게 된다.

3. 파이프라인 FFT 프로세서

OFDM 시스템을 구현하기 위해서는 point수가 큰 FFT가 사용되어지므로 하드웨어의 크기가 큰 병렬 FFT 구조는 구현이 어렵기 때문에 파이프라인 구조로 FFT를 구현해야 한다. 파이프라인은 하드웨어의 구조가 규칙적이므로 제어가 간단하며 순차적 입력과 순차적 출력을 제공하기 때문에 높은 성능을 요구하는 분야에서 가장 많이 사용되고 있는 구조이다. 파이프라인 구조는 경로의 수와 radix-2, 4를 사용하는 것에 따라 구분된다[7-8].

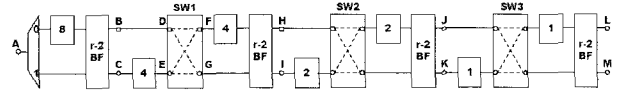


그림 3. Radix-2 MDC FFT 구조
Fig. 3. the structure of R2MDC 16-point FFT

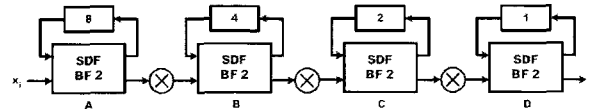


그림 4. Radix-2 SDF FFT 구조
Fig. 4. the structure of R2SDF 16-point FFT

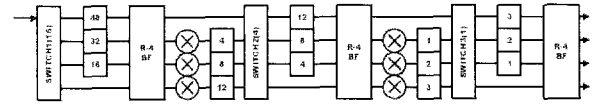


그림 5. Radix-4 MDC FFT 구조
Fig. 5. the structure of R4MDC 64-point FFT

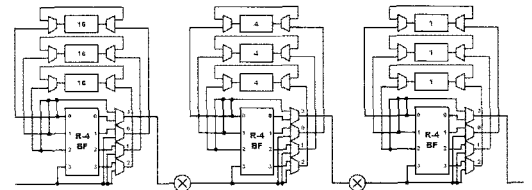


그림 6. Radix-4 SDF FFT 구조
Fig. 6. the structure of R4SDF 64-point FFT

3.1 R2 MDC(Multipath Delay Commutator)

R2MDC FFT구조는 파이프라인 구조를 사용하여 radix-2 알고리즘을 구현하는 가장 기본적인 방법이다. 다음 그림 3에서 R2MDC 16-point FFT를 보인다.

처음 8개의 데이터가 MUX에 의해서 8스트림의 레지스터에 저장되어서 지연되고, 그다음 8개의 데이터가 들어올 때 순서대로 r-2 BF에 입력되어 진다. 즉 첫 번째 데이터 $x(0)$ 와 아래쪽 데이터인 $x(8)$ 이 r-2 BF에 입력되어 버터플라이 연산이 되고, 회전인자가 곱해진 후 다시 스위칭에 의해 경로가 결정되어지는 구조로 되어있다. 그리고 각 stage의 스위치 블록은 이전 stage 스위치보다 두 배 빠르게 스위칭을 수행한다. 이 구조는 다중경로와 지연블록, 스위치블록을 가지고 있으며, 하나의 입력데이터만 처리하기 때문에 50%의 하드웨어 이용도를 갖는다.

3.2 R2 SDF(Singlepath Delay Feedback)

이 구조는 R2MDC구조의 입력을 두 개로 나누는 MUX 대신에 하나의 입력열만 받아 케환과 지연을 이용하여 SDF BF2에 입력하고 계산한다. 여기에 사용되는 SDF BF2는 버터플라이와 MUX로 구성되며, 제어신호에 따라 케환이 되는 위쪽 출력과 회전인자와 곱을 하는 아래쪽 출력으로 나뉜다. 첫 번째 단계에서는 매 $N/2$ 클럭마다 제어 신호가 변하고 그 다음 단계에서는 $N/4$, 그 다음 단계에서는 $N/8$ 과 같은 방식으로 제어신호가 변한다. 케환과 지연에 필요한 메모리는 $N-1$ 개가 필요하다.

이와 같이 하나의 입력 열만을 받아서 처리할 경우에는 R2SDF 구조가 훨씬 더 효율적이다. 그림 4는 R2SDF 16-point FFT에 대한 구조와 기본 버터플라이 구조를 보여 주고 있다.

3.3 R4 MDC(Multipath Delay Commutator)

R4MDC는 R2MDC의 radix-2를 radix-4로 확장한 구조로서 R2MDC와 동일하게 순차적인 입력 데이터를 R-4 BF의 입력으로 사용하기 위해 1 to 4 MUX와 스위치가 필요하다. 동작원리는 R2MDC 구조와 동일하지만, 그림 5와 같이 radix-4 버터플라이를 사용하므로 복소 곱셈기의 수가 많아진다. radix-4 버터플라이 연산자는 8개의 복소 가산기를 필요로 하고, 그 뒤에 3개의 복소 곱셈기가 추가된다.

3.4 R4 SDF(Singlepath Delay Feedback)

R4SDF 또한 R2SDF를 radix-4로 확장한 구조로서 동작 원리는 R2SDF와 같다. 이 구조는 R4MDC 구조에 비해 적은 메모리를 사용한다. 그리고 R4SDF에서 사용되는 다중화기는 R2SDF에 비해서 복잡하지만 그림 6과 같이 적은 복소 곱셈기를 갖는 장점을 가지고 있기 때문에 하드웨어 수와 면적을 줄일 수 있다.

4. 제안된 FFT 알고리즘

FFT 알고리즘 중에서 radix-2는 덧셈기 2개, 곱셈기 1개, radix-4는 덧셈기 8개, 곱셈기 3개를 필요로 한다. 본 논문에서는 저전력 32-point FFT 알고리즘을 위해 radix-2와 radix-4의 조합을 통하여 곱셈기를 줄이는 알고리즘을 제안한다.

$$32 = 2^x 4^y \quad (7)$$

32-point에서 radix-2와 radix-4를 혼합할 경우 사용되는 stage 수는 식 (7)과 같다 $x=5, y=0$ 인 경우 radix-2만 5 stage가 필요하며, $x=3, y=1$ 인 경우 radix-2는 3 stage, radix-4는 1 stage가 필요하다. 그리고 $x=1, y=2$ 인 경우 radix-2는 1 stage, radix-4는 2 stage가 필요하다. 여기서 32는 4의 승수가 아니기 때문에 radix-4는 첫 단계 배치할 수 없다.

4.1 알고리즘에 대한 곱셈기 수 산출 및 비교

R2-R2-R2-R2-R2 알고리즘은 32-point FFT를 radix-2를 이용하여 5개의 stage로 나타낸 것이다. 이 알고리즘은 DFT의 기본식 (1)을 식 (8)과 같이 나타낼 수 있다.

$$X[k_1 + 2k_2 + 4k_3 + 8k_4 + 16k_5] = \sum_{n_5=0}^1 \left[\left(\sum_{n_4=0}^1 \left[\left(\sum_{n_3=0}^1 \left[\left(\sum_{n_2=0}^1 \left[\left(\sum_{n_1=0}^1 x[16n_1 + 8n_2 + 4n_3 + 2n_4 + n_5] W_{32}^{16kn_1} \right) W_{32}^{8kn_2} \right) W_{32}^{4kn_3} \right) W_{32}^{2kn_4} \right) W_{32}^{kn_5} \right] \right) \right] \right] \right] \quad (8)$$

총 버터플라이 stage 수는 5이며, 160개의 복소 덧셈기와 64개의 복소 곱셈기가 필요하다.

표 1. 32-point FFT 조합 알고리즘

Table 1. 32-point FFT hybrid algorithm

x	y	stage	알고리즘
5	0	5	R2-R2-R2-R2-R2
3	1	4	R4-R2-R2-R2
			R2-R4-R2-R2
			R2-R2-R2-R4
1	2	3	R2-R2-R4

R2-R4-R2-R2 알고리즘은 32-point FFT를 radix-2와 radix-4를 이용하여 4개의 stage로 나타낸 것이다. 이 알고리즘은 DFT의 기본식 (1)을 식 (9)와 같이 나타낼 수 있다.

$$X[k_1 + 2k_2 + 8k_3 + 16k_4] = \sum_{n_4=0}^1 \left[\left(\sum_{n_3=0}^1 \left[\left(\sum_{n_2=0}^1 \left[\left(\sum_{n_1=0}^1 x[16n_1 + 4n_2 + 2n_3 + n_4] W_{32}^{16kn_1} \right) W_{32}^{8kn_2} \right) W_{32}^{4kn_3} \right) W_{32}^{2kn_4} \right] \right) \right] \right] \quad (9)$$

총 버터플라이 stage 수는 4이며, 160개의 복소 덧셈기와 56개의 복소 곱셈기가 필요하다.

R2-R2-R2-R4 알고리즘은 32-point FFT를 radix-2와 radix-4를 이용하여 4개의 stage로 나타낸 것이다. 이 알고리즘은 DFT의 기본식 (1)을 식 (10)과 같이 나타낼 수 있다.

$$X[k_1 + 2k_2 + 4k_3 + 8k_4] = \sum_{n_4=0}^1 \left[\left(\sum_{n_3=0}^1 \left[\left(\sum_{n_2=0}^1 \left[\left(\sum_{n_1=0}^1 x[16n_1 + 8n_2 + 4n_3 + n_4] W_{32}^{16kn_1} \right) W_{32}^{8kn_2} \right) W_{32}^{4kn_3} \right) W_{32}^{2kn_4} \right] \right) \right] \right] \quad (10)$$

총 버터플라이 stage 수는 4이며, 160개의 복소 덧셈기와 48개의 복소 곱셈기가 필요하다.

R2-R4-R4 알고리즘은 32-point FFT를 radix-2와 radix-4를 이용하여 3개의 stage로 나타낸 것이다. 이 알고리즘은 DFT의 기본식 (1)을 식 (11)과 같이 나타낼 수 있다.

$$X[k_1 + 2k_2 + 8k_3] = \sum_{n_3=0}^1 \left[\left(\sum_{n_2=0}^1 \left[\left(\sum_{n_1=0}^1 x[16n_1 + 4n_2 + n_3] W_{32}^{16kn_1} \right) W_{32}^{8kn_2} \right) W_{32}^{4kn_3} \right] \right] \quad (11)$$

총 버터플라이 stage 수는 4이며, 160개의 복소 덧셈기와 40개의 복소 곱셈기가 필요하다.

각 알고리즘에 대해 산출한 덧셈기수와 곱셈기수를 비교하면 표 2와 같다. 32-point FFT는 알고리즘에 관계없이

덧셈기의 수는 160개로 일정하지만 곱셈기의 수는 알고리즘마다 다르다. radix-4를 마지막 stage에 사용하는 것이 곱셈기를 줄이는 가장 효율적인 방법이며, R2-R4-R4 알고리즘이 가장 적은 곱셈기를 사용하는 것을 알 수 있다. 따라서 저전력 32-point FFT 프로세서의

표 2. 제안된 32-point FFT 알고리즘 비교
Table 2. comparison of proposed 32-point FFT algorithm

x	y	Algorithm	stage	Add	Mul
5	0	R2-R2-R2-R2-R2	5	160	64
3	1	R2-R4-R2-R2	4	160	56
		R2-R2-R2-R4	4	160	48
1	2	R2-R4-R4	3	160	40

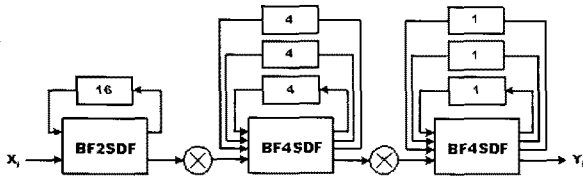
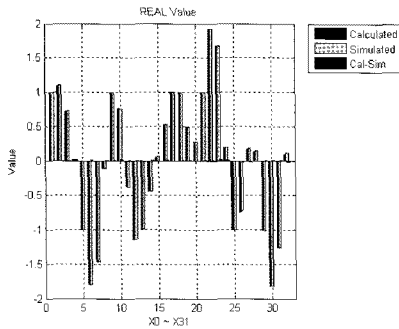
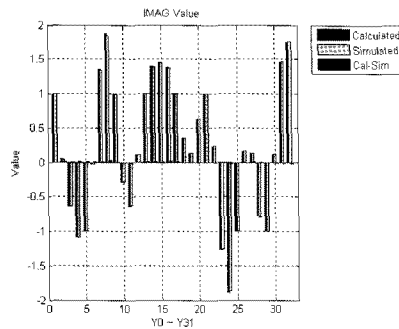


그림 7. 32-point FFT R2-R4-R4 SDF의 block diagram
Fig. 7. block diagram of 32-point FFT R2-R4-R4 SDF structure



(a) 실수 결과 값 비교



(b) 허수 결과 값 비교

그림 8. Matlab을 이용한 결과 값 비교

Fig. 8. comparison between results using Matlab

구현을 위해서는 곱셈기를 가장 적게 사용하는 R2-R4-R4 알고리즘이 적합하다.

4.2 제안된 FFT 프로세서의 구현 및 결과

파이프라인 방식 중 SDF는 MDC보다 제어가 간단하고 메모리를 적게 사용함으로써 저전력으로 동작이 가능하기 때문에 본 논문에서 제안한 32-point FFT R2-R4-R4 알고리즘은 SDF를 이용하여 구현하였다. 알고리즘은 ModelSim XE III를 이용하여 VHDL(VHSIC Hardware Description Language)로 설계하였으며, Spartan3 FPGA(Field-Programmable Gate Array) board의 출력값과 Matlab을 이용해 계산된 이론값을 비교 검증하였다.

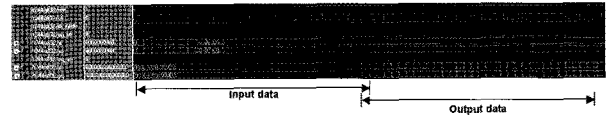


그림 9. 스파르탄3 FPGA board의 입출력 값
Fig. 9. the waveform of Spartan3 FPGA board

radix-2와 radix-4를 혼합한 32-point FFT R2-R4-R4 알고리즘을 SDF 구조로 구현할 경우 전체적인 block diagram은 그림 7과 같다. SDF 구조의 32-point FFT R2-R4-R4 알고리즘은 각 stage마다 radix-2 버터플라이 연산과 radix-4 버터플라이 연산을 수행하는 BF2SDF 1개와 BF4SDF 2개로 구성되며, 버터플라이 연산을 위해 지연 feedback을 하고, 연산된 데이터를 다음 stage에 직렬로 보내기 위한 31개의 지연소자가 있다. 그리고 버터플라이 연산이 끝난 후 다음 stage로 넘어갈 때 연산의 결과 값과 회전인자 곱을 하는 복소 곱셈기가 있으며, 회전인자 값을 저장해 두고 제어신호에 따라 정해진 값이 출력 되는 회전인자 lookup table이 있다.

VHDL로 설계된 32-point FFT R2-R4-R4 SDF에 임의의 32개 값을 입력하였으며, 그 결과 waveform은 그림 9와 같다. 32 clock 동안 데이터가 입력이 되고, 32 clock에서 출력 값이 나타나기 시작하였으며, 63 clock에서 출력 값이 모두 나와서 파이프라인 SDF의 동작을 만족한다.

입력된 32개 데이터의 이론값은 Matlab 7.0을 이용한 계산으로 산출하였다. 계산된 이론값과 VHDL로 구현한 FFT 프로세서의 출력값은 그림 8과 같다. 그림 8에 나타난 바와 같이 설계된 FFT 프로세서의 출력값은 이론값과 일치한다.

5. 결 론

대용량 고속 데이터 통신을 지향하는 차세대 무선통신으로 OFDM 통신방식이 주목받고 있는 가운데 육상통신뿐만 아니라 해상통신에서도 적용을 추진 중이다.

OFDM 통신방식이 해상 통신에 적용될 경우 ITU에서는 32-point FFT 프로세서의 사용을 권고하였다. 또한 해상 통신에서는 해양사고 및 조난 시에도 반드시 통신이 이루어져야 하므로 OFDM 통신시스템은 저전력으로 동작하여야 한다. 따라서 OFDM 통신시스템에서 전력 소모가 가장 큰 FFT 프로세서는 저전력으로 구현되어야 한다.

본 논문에서는 radix-2와 radix-4를 조합한 저전력 32-point FFT R2-R4-R4 알고리즘을 제안하고, 이를 기반으로 파이프라인 구조 32-point FFT R2-R4-R4 SDF 방식을 VHDL로 설계하여 알고리즘을 검증하였다.

radix-2와 radix-4를 조합할 경우 32는 4의 승수가 아니

기 때문에 32-point에서는 radix-4를 첫 stage에 적용할 수 없으며, radix-2만을 적용할 수 있다. 그리고 마지막 stage에는 radix-4를 적용하는 것이 곱셈기를 줄이는 데 효율적인 방법이라는 것을 확인하였다. 또한 파이프라인 방식 중에 SDF를 사용했기 때문에 radix-2나 radix-4를 사용하더라도 메모리는 포인트수 N 에서 1을 뺀 31개가 필요하고, 계산속도는 $2N-1$ clock으로 63 clock에 출력 값이 모두 출력이 된다.

제안된 알고리즘 R2-R2-R2-R2-R2, R2-R4-R2-R2, R2-R2-R2-R4 그리고 R2-R4-R4 중에서 덧셈기의 수는 모두 160개로 같지만 R2-R4-R4의 곱셈기 수가 40개로 제안된 알고리즘 중에서 가장 적게 사용함을 확인하였다. 즉 R2-R4-R4 알고리즘은 기본 알고리즘이라고 할 수 있는 R2-R2-R2-R2-R2 알고리즘의 곱셈기 수보다 24개의 곱셈기를 줄임으로써 하드웨어의 부담을 줄이고, 전력 소비도 줄여 저전력화 하였다. VHDL로 구현된 파이프라인 구조 32-point FFT R2-R4-R4 SDF 방식의 출력값은 Matlab의 이론값과 일치하였다.

FFT는 OFDM 통신시스템의 중요 디바이스이므로 제안된 32-point FFT 알고리즘은 Wireless PAN 기반의 근거리 해상통신에서 OFDM을 적용한 저전력 데이터 통신을 위한 선도 기술로 유용할 것이다.

참 고 문 헌

- [1] 정연호, "고속 다중 사용자 데이터 전송 환경에서 고유의 펄스 형성화 기술을 적용한 적응 OFDM 시스템의 개발에 관한 연구", 기초과학연구사업 학술기사, 한국과학재단, 2004.5.
- [2] 오정렬, 저 면적 및 저 전력 복소 곱셈기를 갖는 파이프라인 방식의 FFT 프로세서 설계에 관한 연구, 박사학위논문, 전북대학교 대학원, 2005.8.
- [3] 김정년, 디지털 漁業通信을 위한 SSB 모델 開發에 관한 研究, 박사학위논문, 목포해양대학교 대학원, 2007.6
- [4] ITU, Document 8/161-E, 2006.9.
- [5] 장영범, DSP 이론과 실무, 생능출판사 2004. 3.
- [6] J. W. Cooley, J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Math. Comp.*, Vol.19, pp.297-301, 1965.4.
- [7] E. H. Wold, A. M. Despain, "Pipeline and Parallel Pipeline FFT Processors for VLSI Implementation", *IEEE Trans. Comput.*, C-33(5), pp.414-426, May 1984.
- [8] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)Modulation", in Proc. *IEEE URSI Int. Symp. Signals, Syst., Electron.*, pp.257-262, 1998.

저 자 소 개



조승일(Seung-II Cho)
 2006년 : 목포해양대학교 해양전자통신공학부 졸업 (공학사)
 2008년 : 목포해양대학교 대학원 해양전자통신공학과 (공학석사)
 2008년 3월~현재 : 동 대학원 해양전자통신공학과 박사과정

관심분야 : 무선통신용 LSI 설계, 디지털 통신회로 설계, 주파수간섭분석

Phone : 061-245-1107
 E-mail : whtmddlf@mmu.ac.kr



차재상(Jae-Sang Cha)
 1991년 : 성균관대학교 전기공학과 졸업 (공학사)
 1997년 : 성균관대학교 전기공학과 (공학석사)
 2000년 : 일본 Tohoku 대학교 대학원 전자공학과 (공학박사)

2000년~2002년 : 한국전자통신연구원(ETRI) 무선방송기술 연구소 선임연구원

2002년~2005년 : 서경대학교 정보통신공학과 전임강사
 2005년~현재 : 서울산업대학교 메체공학과 조교수

관심분야 : 디지털 방송전송기술, Cognitive Radio, UWB, 홈네트워크 무선통신기술, 대역확산 및 다중접속기술, 4세대 이동통신기술

Phone : 02-970-6431
 Fax : 02-974-6123
 E-mail : chajs@snu.ac.kr



박계각(Gye-Kack Park)
 1982년 : 한국해양대학교 항해학과 졸업 (공학사)
 1986년 : 한국해양대학교 항해학과 수송공학 전공(공학석사)
 1993년 : 동경공업대학 시스템과학전공 (공학박사)

1993~1994년 : 일본통신성 국제퍼지공학연구소(선임연구원)
 2001~2002 : University of Cincinnati 방문교수

2005~현재 : 목포해양대학교 해상운송시스템학부 교수

관심분야 : 지능시스템, 해양정보시스템, 국제경제학
 Phone : 061-240-7164
 E-mail : gkpark@mmu.ac.kr



양충모(Chung-Mo Yang)
1996년: 경북대학교 졸업(공학사)
1998년: 경북대학교 대학원(공학석사)
2004년: 일본 Tohoku 대학교 대학원 전
자공학과 (공학박사)
2004년 10월~현재: 현재 삼성전기 중앙
연구소 eMD Lab. 공정개발센터
책임연구원

관심분야 : 무선통신용 LSI 설계, 무선통신시스템
Phone : 031-218-2862
E-mail : chungmo.yang@samsung.com



김성권(Seong-Kweon Kim)
1996년: 인하대 전자재료공학과 졸업
(공학사)
1996~1999년: 삼성전자 시스템 LSI사업
부 주임연구원
2002년: 일본 Tohoku 대학교 대학원 전
자공학과 (공학박사)

2002~2004년: 일본 Tohoku 대학교 전자통신연구소
Research Fellow
2004년~현재: 목포해양대학교 해양전자통신공학부 조교수

관심분야 : 무선통신용 LSI 설계, 주파수분배정책 및 주파
수의 효율적 사용에 관한 연구
Phone : 061-240-7265
Fax : 061-240-7283
E-mail : skkim12632@mmu.ac.kr