

OFDM용 FFT 프로세서의 설계

배영재, 조원경

경희대학교 전자공학과

toughguy@csvlsi.kyunghee.ac.kr

Design of FFT Processor for OFDM

YoungJe Bae, WonKyung Cho

School of Electronic Engineering KyungHee University

toughguy@csvlsi.kyunghee.ac.kr

Abstract

This paper present the architecture and design of FFT processor for the OFDM modulation. The OFDM modulation have a merit that use frequency efficiently and robust ISI. It needs FFT to have fast and large number of points. Moreover, this FFT design has pipeline architecture. R^2SDF architecture for FFT processor has more advantage others. Therefore this paper present FFT processor used R^2SDF architecture.

I. 서론

미래의 이동통신 시스템에서 멀티미디어 서비스를 지원하기 위해서는 무선구간에서의 고속 및 채널 환경에 강한 전송방식이 필수적이다. 그러나 수킬로바이트급의 음성서비스를 제공하는 셀룰라 시스템으로는 음성, 데이터, 이미지등을 포함한 멀티미디어에 대한 서비스를 할 수 없다. 이러한 고속, 고품질의 무선 멀티미디어 서비스로 제공하기 위해서는 더 빠른 전송속도의 통신방식이 필요하다.[1]

최근들어, 차세대 멀티미디어 이동통신 서비스를 위한 변복조 방식으로 OFDM(Orthogonal Frequency Division Multiplex)에 대한 많은 연구가 진행되어지고 있다. OFDM은 다수의 반송파로 변조하여 병렬 전송하는 다중 반송파 변조기술로서, 1950년대부터 연구가

시작되었으나 그 구현의 복잡성으로 인하여 실용화되지 못하였다. 그러나 최근에 들어 FFT와 같은 고속 디지털 신호처리와 VLSI의 고속, 고집적도 기술이 발전함에 따라 다중 반송파 변조 방식의 구현상의 문제점을 해결할 수 있게 되었다.[2][3]

본 논문에서는 OFDM의 기본 구조와 여러 가지의 Pipeline FFT를 고찰하였으며[1][4][5], 하드웨어를 효율적으로 사용하고 간단하게 구현할 수 있는 radix-2² algorithm을 이용한 Pipeline 구조의 1024-point FFT를 설계하였다. FFT 연산기 설계 시에 오버플러워를 방지하기 위해 기존의 스케일링 또는 라운딩 방법이 아닌 오버플러워를 고려한 가산기를 설계하여 하드웨어로 구현하였다.

II. OFDM 기본구조

OFDM의 개념은 고속의 데이터를 다수의 직교 부반송파를 사용하는 다수 반송파 변조방식으로서 주파수 사용 효율을 극대화시킬 수 있고, ISI(inter symbol interference)와 ICI(inter-carrier interference) 영향에 강하다.[1][2]

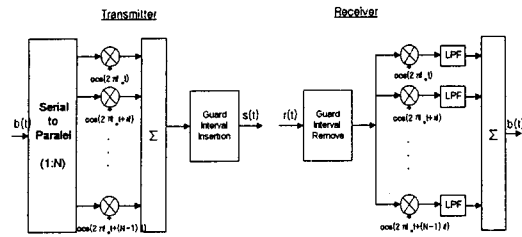


그림 1 OFDM 기본적인 구조
Fig. 1 The Basic OFDM Structure

OFDM의 기본적인 구조는 그림 1과 같다. 송신부는 QPSK(혹은 QAM) 변조된 데이터가 OFDM multiplexer에 의해 처리된 후에 ISI를 방지하기 위해 보호 구간을 삽입하는 구조로 되어 있으며, 수신부는 먼저 보호구간을 제거하고 OFDM demultiplexer와 QPSK에 의해 복조하는 구조로 되어있다.

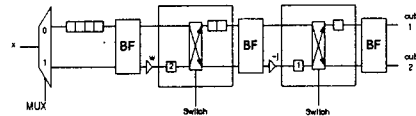
주어진 OFDM의 다수 반송파의 배열로 신호를 생성해 내는 효율적인 방법은 Inverse Discrete Fourier Transform을 사용하는 방법이다. 그러나 하드웨어 구현시에는 계산량을 줄이기 위해 Fast Fourier Transform(FFT)을 사용한다.[3]

III. FFT Processor 구조의 비교

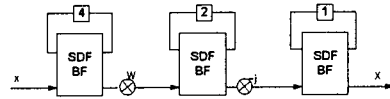
FFT 구조는 크게 완전 병렬 FFT와 Pipeline FFT가 있다. OFDM 시스템을 구현하기 위해서는 큰 point의 FFT가 사용되어지므로 병렬 FFT와 같은 구조는 하드웨어 크기가 커지므로 구현이 불가능하다. 그러므로 하드웨어의 병렬성을 이용하는 Pipeline FFT로 구현하여야 한다. 이 Pipeline은 하드웨어의 구조가 규칙적이므로 제어가 간단하며 순차적 입력과 순차적 출력을 제공하기 때문에 높은 성능을 요구하는 응용분야에서 가장 많이 사용되고 있는 구조이다.

Pipeline FFT를 위한 구조는 그림 2와 같으며, 표1은 6종류의 Pipeline FFT 구조를 비교한 한 것이다. Radix-2 Multi-path Delay Commutator(R2MDC)는 radix-2 FFT 알고리즘에 가장 전형적인 방법으로 입력이 순차적으로 들어오면 두 개의 병렬 데이터 열로 바뀐다. 하나의 데이터는 flow graph에서 다음의 Butterfly(BF)로 들어오는 입력 데이터까지 지연된 후에 계산되어진다. Radix-2 Single-path Delay Feedback (R2SDF)은 BF의 출력을 피드백 쉬프트 레지스터를 사용하여 R2MDC 구조보다 레지스터를 적게 사용하는 구조이다. Radix-4 Multi-path Delay Commutator (R4MDC)은 R2MDC를 radix-4로 구현한 구조이다. 이 구조는 가장 많은 하드웨어와 메모리가

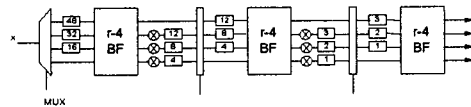
필요하다. Radix-4 Single-path Delay Feedback (R4SDF) 또한 R3SDF을 radix-4로 구현한 구조로서 R4MDC 구조보다는 메모리 사용이 적으나, R2SDF 구조에 비해 제어가 복잡하다. Radix-4 Single Delay Commutator (R4SDC) 구조는 한번에 하나의 출력을 내보내는 변형된 radix-4 butterfly를 사용하여 R4MDC를 간략화시켰으나 구조상 제어가 가장 복잡하다.



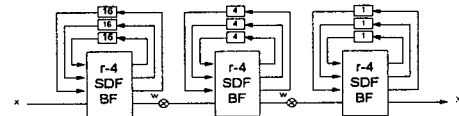
(a) R2MDC(N=8)



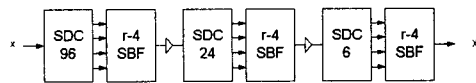
(b) R2SDF(N=8)



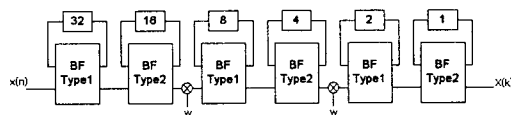
(c) R4MDC(N=64)



(d) R4SDF(N=64)



(e) R4SDC(N=64)



(f) R2²SDF(N=64)

그림 2 여러 가지 파이프라인 FFT processor 구조
Fig. 1 Various architecture for pipeline FFT processor

표 1 Pipeline FFT 구조 비교

Table 1 Pipeline FFT architecture comparison

	multiplier	adder	memory size	control
R2MDC	$2(\log_4 N - 1)$	$4\log_4 N$	$3N/2 - 2$	간단
R2SDF	$2(\log_4 N - 1)$	$4\log_4 N$	$N - 1$	간단
R4MDC	$3(\log_4 N - 1)$	$8\log_4 N$	$5N/2 - 4$	간단
R4SDF	$\log_4 N - 1$	$8\log_4 N$	$N - 1$	중간
R4SDC	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$	복잡
$R2^2$ SDF	$\log_4 N - 1$	$4\log_4 N$	$N - 1$	간단

Radix-2² Single-Path Delay Commutator(R2²SDF) 구조는 radix-2 DIF FFT 알고리즘에서 Single Flow Graph(SFG)을 그림 3과 같이 변형하여 Pipeline화 한 것이다. 그리하여 이 R2²SDF 구조는 radix-4 구조의 R4SDF와 R4SDC와 같은 수의 곱셈기를 사용하여 구현이 가능하다. 그리고 기본적으로 SDF 구조이므로 N-1 크기의 메모리와 $4\log_4 N$ 만큼의 가산기를 사용하여 효율적으로 하드웨어를 사용하고 또한 구조도 간단히 구현할 수 있다.

본 논문에서는 이러한 장점을 가진 R2²SDF 구조를 사용하여 FFT Processor를 구현하였다.

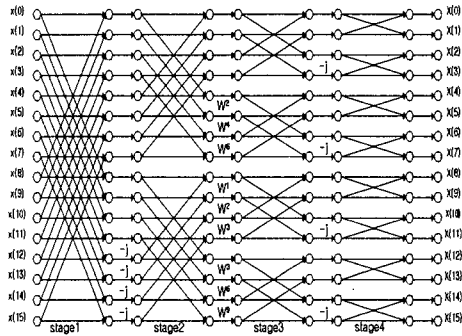


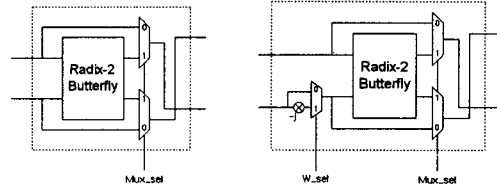
그림 3 Radix-2² DIF FFT flow graph (N=16)

Fig. 3 Radix-2² DIF FFT flow graph (N=16)

IV. R2²SDF FFT Processor의 설계

R2²SDF 구조는 그림 3과 같이 스테이지1과 스테이지2 사이에는 곱셈기가 들어가지 않고 스테이지3과 스테이지2 사이에만 "Twiddle factor"가 있는, 스테이지 사이를 규칙적인 구조로 SFG를 변형하여 그림3 (f)와 같이 Pipeline화 한 것이다. 그림 4는 "Twiddle factor"가 들어가지 않는 기본적인 두 가지의 BF의 구조이다. BF1 구조(Type1)는 Mux_sel 신호에 제어되며, BF2 구조(Type2)는 BF1의 구조와 비슷하나 BF 입력단전에 W_sel 신호에 의해 제어되는 구조이다. "Twiddle

factor", -j의 승산은 실수데이터는 허수부 입력으로, 허수 데이터는 실수부 입력으로 바꾸어서 BF에 입력되어지는 구조로서, 하드웨어를 절약할 수 있다.



(a) Butterfly Type1

(b) Butterfly Type2

그림 4 R2²SDF의 Butterfly Type

Fig. 4 R2²SDF's Butterfly Type

(가) 오버플러위를 고려한 가산기 설계

일반적으로 FFT 내부 연산시에 발생하는 오버플러위를 방지하기 위해 연산결과값을 스케일링을 수행한다. 그림 4는 BF의 출력값을 1/2를 스케일링함으로써 오버플러위를 방지하는 회로를 나타내고 있다. 이 구조는 입력값 x0와 x1의 가산기 연산에서 오버플러위를 위해서 1비트가 더 사용되어야 하고 연산 데이터를 가공하므로써 하드웨어 낭비와 데이터의 오차를 발생시킨다. 그림 5는 이러한 낭비를 없애기 위한 가산기 구조이다. 이 구조는 4비트 가산기라 할 때, 입력 X, Y는 $X+Y/2$ 로 연산되어 범위가 연산 결과가 -8에서 7 사이에 있으면 결과를 그대로 출력하고, 결과가 -8보다 작거나 7보다 크면 결과는 -8 혹은 7이 출력함으로써 스케일링되어진다. 이 구조를 사용하면 최소한의 하드웨어 추가로 스케일링 문제를 해결할 수 있고 출력을 무조건 1/2 곱해주는 방식에 비해 보다 개선된 성능을 얻을 수 있다.

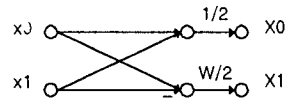


그림 4 1/2 스케일링 Butterfly

Fig. 4 1/2 scaling Butterfly

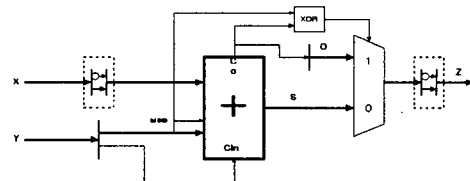


그림 5 Overflow를 고려한 가산기

Fig. 5 Adder in considered of Overflow

(나) 승산기 설계

승산기는 FFT Processor에서 가장 중요한 모듈이다. 승산기는 하드웨어 크기가 가장 클 뿐 아니라 Processor의 성능을 결정하는 중요한 요소이다. FFT에 사용되는 승산기는 두 개의 실수와 허수 입력에 대한 complex 승산기이다. 입력을 $a+jb$, $c+jd$ 라 했을 때, 출력 $e+jf$ 는 식(1)과 같다.

$$e+jf = (a+jb)(c+jd) = (ac-bd)+j(bc+ad) \quad (1)$$

$$e+jf = a(c+d)-d(a+b)+j(a(c+d)+c(a-b)) \quad (2)$$

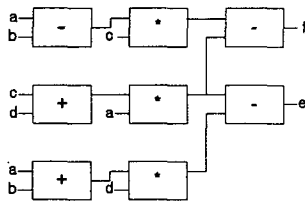


그림 6 승산기 구조

Fig.6 Multiplier architecture

하나의 복소수 승산기는 식(1)과 같이 4개의 실수 승산기와 2개의 실수 가산기가 필요하나, 식을 식(2)와 같이 변형시켜 구현하면 실수 승산기 3개와 실수 가산기 5개가 사용되므로, 한 개의 승산기를 줄여 하드웨어 크기를 크게 줄일 수 있다.[5]

(다) 제어부 설계

R^2SDF 구조의 제어는 클럭에 맞춰 2048 point의 경우 11비트의 카운터가 사용되는데, 이 카운터는 BF1, BF2 제어와 ROM table에 "Twiddle factor"값의 입력 제어에 사용되어진다. 카운터에 의한 제어부 설계는 구현이 간단하다.[6]

VI. 실험 및 고찰

본 논문에서는 R^2SDF pipeline 구조의 FFT 프로세서 설계는 하드웨어 기술 언어인 VHDL을 이용, 모델링하여 기능을 검증하였고 Altera Flex10K에서 합성한 결과 critical path의 지연시간이 최대 40ns 정도였으며, 그림 7은 시뮬레이션 결과를 보여준다. 처음 데이터는 N 클럭후에 나오기 시작하여 연속적으로 데이터가 나온다.

V. 결론

본 논문에서 여러 가지 구조의 Pipeline FFT를 고찰

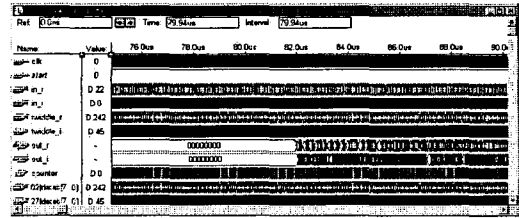


그림 7 Simulation 결과 파형

Fig. 7 Simulation result wave

하였고 그 중 메모리, 승산기 그리고 가산기 등의 하드웨어를 적게 사용하고 설계 또한 간단한 R^2SDF 구조를 이용하여 설계하였다. 데이터 폭은 실수와 허수 입력을 위해 8비트가 사용되어졌으며, 출력 또한 실수와 허수 8비트로 구성되어 있으며 FFT 내부연산시에 생기는 오버플러위를 방지하기 위해 출력에 기존의 1/2 스케일링 구조를 오버플러위를 고려한 가산기를 설계하여 하드웨어를 절약하고 성능의 향상시켰으며, 복소수 승산기를 줄여 성능이 향상되도록 설계하였다. 향후 연구과제로서 FFT의 성능을 가장 큰 영향을 미치는 연산인 고속 승산의 구조에 대해 연구가 이루어져야 할 것이다.

참고문헌

[1] 연구보고서, "무선멀티미디어기술 연구", 한국전자통신연구원, 1997.12
 [2] 최진형, 김동규, et., "유럽 지상파 디지털 TV 방송용 COFDM 방식과 수신 기술의 고찰", Telecommunication REVIEW, 1998 제 8권 2호, pp. 222-237
 [3] Ali N. Akansu, "Orthogonal Transmultiplexers in Communication: A Review", IEEE. Trans on SIGNAL PROCESSING, VOL.46, NO 4, APRIL 1998
 [4] Lawrence R. Rabiner, Bernard Gold, "Theory and application of digital signal processing", Prentice-hall, 1975 pp. 573-625
 [5] D. Peter, and D. Renshaw, "VLSI signal processing a bit-serial approach", Addison-Wesley, 1985 pp. 147-199
 [6] Alvin M. Despain, "Very Fast Fourier Transform Algorithms Hardware for Implementation", IEEE. Trans on COMPUTER, VOL. c-28, NO. 5, May 1979