

# OFDM 송신단의 지연을 줄이기 위한 IFFT Processor의 설계

정회원 김 준 우\*, 박 윤 옥\*, 종신회원 김 환 우\*\*

## A Design of IFFT Processor for Reducing OFDM Transmitter Latency

Jun-Woo Kim\*, Youn-Ok Park\* *Regular Members*, Whan-Woo Kim\*\* *Lifelong Member*

### 요 약

본 논문에서는 OFDM(Orthogonal Frequency Division Multiplexing) 송신단에서 IFFT 를 수행한 후 Cyclic Prefix를 첨부하여 OFDM 심볼을 생성하는데 필요한 지연을 감소시킬 수 있는 IFFT(Time shifted DIT IFFT)의 구조를 제안하고, IEEE 802.16e Mobile WiMax OFDMA/TDD규격에 적합한, 1024크기의 FFT에 1/8 cyclic prefix를 가지는 시스템의 송신단 IFFT를 VHDL로 설계한 결과를 제시한다.

본 논문에서 제안하는 IFFT는 OFDM송신단의 지연을 줄이기 위하여 IFFT에 역비트(Bit-Reversed) 순서로 데이터를 입력하고, FFT의 Frequency Translation 특징을 이용해 IFFT의 출력이 cyclic prefix의 길이만큼 시간영역 쉬프트(Time-shift) 되어 나오도록 구현되었다. 이 과정은 cyclic prefix의 길이 특징을 이용하여 부가적인 복소곱셈기(Complex Multiplier)없이 구현되었고, OFDM 송신단의 지연과 함께 IFFT 결과를 저장하는 메모리의 크기도 줄일 수 있다. 송신단의 최종 출력이 통상적인 OFDM 심볼과 완전히 동일하기 때문에 전체 시스템의 성능에도 영향을 미치지 않는다.

**Key Words** : IFFT, FFT, OFDM, CP, Cyclic Prefix

### ABSTRACT

In This Paper, we introduce an efficient IFFT design technique named for transmitter of OFDM (Orthogonal Frequency Division Multiplexing) system. In OFDM system, a cyclic prefix is inserted in forepart of OFDM symbol to prevent ICI(Inter-channel Interference) and ISI (Inter-symbol Interference). Attaching cyclic prefix causes delay in storing and copying IFFT result. The proposed IFFT removes this delay because its output is cyclic shifted by the length of cyclic prefix. So we can make a complete OFDM symbol by just copying the forepart of IFFT output to the end. In many cases, the length of cyclic prefix is 1/2n of FFT size, and this IFFT does not require additional hardware complexity and it does not cause any performance degradation.

### 1. 서 론

OFDM (Orthogonal Frequency Division Multiplexing) 기술은 무선 환경에서 다중 경로 페이딩에 강하여

무선 인터넷 접속 등 고속 데이터 전송에 적합한 방식으로 각광받고 있다. OFDM의 송신단에서는 IFFT (Inverse Fourier Transform)를 수행하여 OFDM 심볼을 생성한 뒤, 인접한 OFDM 심볼 사이에 채

\* 한국전자통신연구원 이동단말모뎀연구팀(jwkim74@etri.re.kr)

\*\* 충남대학교 전자전파정보통신공학과(wwkim@cnu.ac.kr)

논문번호 : KICS2008-11-511, 접수일자 : 2008년 11월 3일, 최종논문접수일자 : 2009년 11월 17일

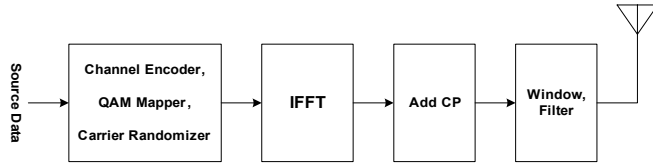


그림 1. OFDM 송신단의 구조

널의 임펄스 응답보다 긴 cyclic prefix(순환전치)를 보호구간(Guard Interval)에 삽입하여 ISI(Inter-symbol Interference)와 ICI (Inter-channel Interference)의 영향을 제거한다<sup>1)</sup>.

cyclic prefix는 시간영역에서 OFDM 심볼의 뒷부분을 앞부분으로 복사하는 것이므로 이로 인한 지연이 발생하게 되는데, 본 논문에서는 이러한 지연 없이 OFDM 심볼에 cyclic prefix를 첨부하는 IFFT를 설계하였다.

본 논문의 구성은 다음과 같다. 2장에서는 IFFT와 FFT의 관계 및 기준에 송신단의 지연을 줄이기 위해 제시된 방법을 살펴보고, 3장에서는 전치순환부를 사용하는 OFDM 시스템 송신부의 IFFT를 하드웨어로 효율적으로 구현하여 부가적인 복잡도 없이 송신단의 지연과 하드웨어의 복잡도를 줄일 수 있는 방법을 제안하고 4장에서는 구현된 결과를 기존의 DIF방식의 IFFT를 사용해 시뮬레이션한 결과와 비교하였다.

본 논문에서 제안하는 IFFT는 모든 OFDM 시스템에 적용이 가능하다. 이 방법은 2장에서 OFDM 송신단의 지연을 감소시키기 위해 사용된 것으로 소개된 Cyclic Suffix(순환후치)를 사용하는 방식과 송신단의 지연이 같으며, 출력은 cyclic prefix를 첨부한 OFDM 심볼과 동일하고 성능의 열화도 야기하지 않는다. 하드웨어의 복잡도에 있어서도 필요한 메모리의 크기가 감소되고, cyclic prefix의 길이가 FFT 크기의 1/2<sup>n</sup>인 경우에는 부가적인 복소 곱셈기(Complex Multiplier)도 필요하지 않다.

## II. OFDM 송신단 IFFT의 구현

### 2.1 FFT와 IFFT의 관계

N-point FFT와 IFFT는 다음과 같은 수식으로 표시된다<sup>2)</sup>.

$$\text{FFT연산: } F_k = \sum_{n=1}^N f_n e^{-j\frac{2\pi nk}{N}} \quad (1)$$

$$\text{IFFT연산: } f_n = \frac{1}{N} \sum_{k=1}^N F_k e^{j\frac{2\pi nk}{N}} \quad (2)$$

IFFT연산은 FFT프로세서를 이용해 수행되는데, 그 관계는 다음과같은 식으로 표현될 수 있다.

$$f_n = \text{IFFT}(F_k) = \frac{1}{N} (\text{FFT}(F_k^*))^* \quad (3)$$

$$= \frac{1}{N} (j \times \text{FFT}(j \times F_k^*))^* \quad (4)$$

(\*)는 공액연산(Conjugate)을 의미한다. (3), (4)식은 IFFT연산이 FFT 프로세서를 이용해 구현될 수 있다는 것을 보여준다. (3)식으로 IFFT를 구현하면 FFT에 공액연산용 1비트 덧셈기가 2개 추가되는 반면, (4)식으로 구현하면 부가적인 하드웨어가 필요없게 된다. j(\*)는 단순히 실수부와 허수부를 바꾸어주는 것이므로 IFFT는 그림 2에서처럼 FFT 프로

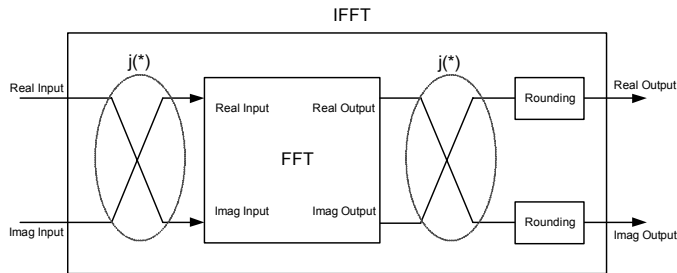


그림 2. FFT를 이용해 IFFT를 수행하는 회로

세서 입력의 실수부와 허수부를 바꾸고 FFT 프로세서의 출력에서도 실수부와 허수부를 바꿔준 뒤, 1/N을 곱하면 되기 때문이다. 실제 OFDM 송신단의 구조에서는 1/N을 곱하지 않고 송신단 마지막에 있는 D/A 변환기의 입력 비트에 맞게 Rounding한다.

### 2.2 OFDM 송신단의 지연을 감소시키기 위한 기존의 방법

그림 3에서 볼 수 있듯이, cyclic prefix는 OFDM 심볼의 바로 앞 보호구간에 그 OFDM 심볼의 마지막 부분을 보호구간의 길이만큼 복사하여 구성하는데, 이 부분을 하드웨어로 구현하려면 IFFT의 출력을 메모리에 저장했다가 심볼 뒷부분에 해당하는 부분부터 읽어내야 하기 때문에 송신단의 지연과 하드웨어의 복잡도가 증가하게 된다.

구체적으로는, IFFT가 DFT (Decimation In Frequency) 방식인 경우<sup>[5]</sup>, IFFT결과가 역비트(Bit-Reversed) 순서로 출력되므로 그림 4에서처럼 OFDM 심볼 전체를 메모리에 저장해서 데이터 재정렬(Reordering)과 cyclic prefix의 첨부를 수행하기 때문에 그림 5에서처럼 FFT 길이의 2배 시간만큼 송신단 지연이 발생한다. IFFT가 결과값을 순차적으로 출력하는 DIT (Decimation In Time) 방식인 경우에도 IFFT 출력의 뒷부분을 먼저 전송해야 하기 때문에, 송신단의 지연과 하드웨어의 복잡도 측면에서 큰 개선을 기대할 수 없다.

만약 이러한 송신단의 지연을 줄일 수 있다면 일

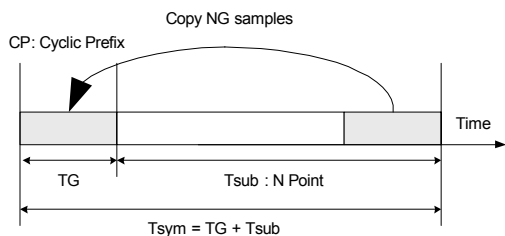


그림 3. OFDM 심볼의 시간영역 구조

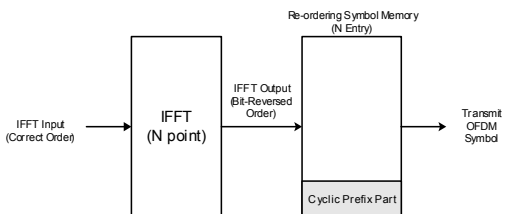


그림 4. 데이터 재정렬과 cyclic prefix 첨부를 위한 메모리

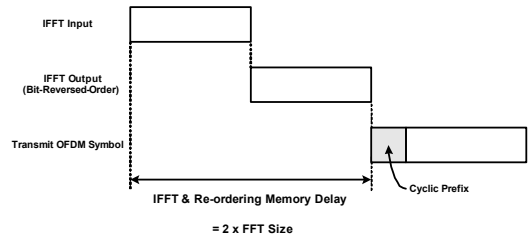


그림 5. Cyclic prefix 첨부로 인한 지연

반적인 데이터를 전송할 때에도 요긴하지만, 특히 긴급한 제어 메시지등을 전송할 때 잊점이 될 수 있다. 예를 들어 IEEE 802.16e Mobile WiMax<sup>[7]</sup>의 단말기에서 핸드오버(Hand Over)를 수행하기 위해 기지국으로 전송하는 Handover Ranging 메시지 등은 상위계층에서 핸드오버를 결정하는 순간 채널 부호화기도 거치지 않고 OFDM 송신단에 전달되며, 최대한 빨리 기지국으로 전송되어야 하는데, 이러한 경우 송신단의 지연이 짧으면 유리할 것이다.

이러한 OFDM송신단의 지연을 줄이고 부가적으로 하드웨어의 복잡도도 줄이기 위한 시도로서 cyclic prefix 대신 Cyclic Suffix를 사용하는 방법이 제시된 바 있다<sup>[2][4]</sup>. 이는 그림 6와 그림 7에서 보인 것처럼 OFDM 심볼의 뒷부분을 앞으로 복사하는 대신, 앞부분을 뒤로 복사하는 방식으로 송신단의 지연을 줄이는 방법이다. IFFT의 출력이 역비트 순서가 아니라 순차적으로(Correct Order) 나가

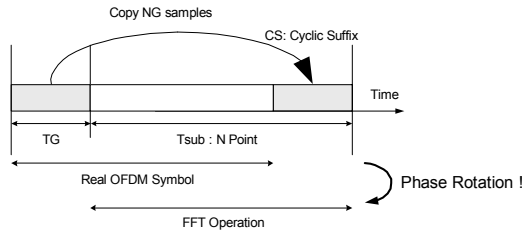


그림 6. cyclic suffix를 사용한 OFDM 심볼

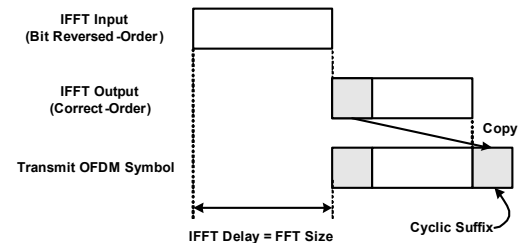


그림 7. cyclic suffix를 사용할 때의 지연

야 하므로 DIT 방식의 IFFT를 이용한다.

그러나 cyclic suffix를 사용하려면 이미 cyclic prefix를 사용하도록 규격이 정해진 대부분의 OFDM 시스템에의 규격 변경이 필요하며, 실제 OFDM 심볼 구간과 수신단에서 FFT를 수행하는 영역이 일치하지 않아 FFT를 수행한 결과에서 위상 회전(Phase Rotation)이 발생하게 된다. 이러한 위상의 회전을 혼련 심볼과 주파수 영역 등화기를 이용해 보상하거나<sup>[2]</sup>, 수신단에서 대용량 FIFO(First-In First-Out) 메모리를 사용하여 위상회전을 방지하는<sup>[3]</sup> 방법이 제시되었지만, 전자의 경우 혼련심볼 없이 인접한 파일럿 부채널(Pilot Subcarrier)들을 보간(Interpolation)해서 채널추정을 수행할 때 성능열화를 유발할 가능성이 있으며, 후자의 경우에는 송신단의 지연은 감소하더라도 수신단의 지연이 그만큼 증가하여 결국 전체 전송 경로의 지연은 줄어들지 않게 된다.

### III. 제안된 OFDM 송신단 FFT의 구조

#### 3.1 송신단의 지연을 감소시키는 방법

(5)식은 FFT의 Frequency Translation 특징<sup>[5]</sup>을 나타내고 있다. FFT의 결과가 원하는 길이만큼 쉬프트된 값이 나오게 하려면 FFT의 입력에 그만큼의 위상회전값을 곱해주면 된다. IFFT에서 결과값을 쉬프트 시키려면 (6)에서 볼 수 있듯이 FFT의 경우와 반대의 위상값을 곱해주면 된다.

여기서 N을 FFT 크기인  $L_{FFT}$ , M은 cyclic prefix의 길이인  $L_{CP}$ 로 두면 IFFT의 결과를  $L_{CP}$ 만큼 시간지연되어 나오게 하기 위해 필요한 회전인자  $W(L_{CP}, L_{FFT}, k)$ 는 식(7)과 같게 된다. 즉, 그림 8에서처럼 IFFT입력에 (7)에서 제시한 위상회전값을 곱해주거나 그림 9에서처럼 IFFT내의 FFT프로세서에서 그 반대의 위상회전값을 곱해주면 된다.

$$F_{(k+M)} = FFT(f_n e^{j\frac{2\pi Mn}{N}}) \quad (5)$$

$$f_{(n+M)} = IFFT(F_k e^{-j\frac{2\pi Mk}{N}}) \\ = \frac{1}{N} \left[ FFT(F_k^* e^{j\frac{2\pi Mk}{N}}) \right]^* \quad (6)$$

$$W(L_{CP}, L_{FFT}, k) = e^{-j\frac{2\pi \times L_{CP} k}{L_{FFT}}} \quad (7)$$

이렇게 하면 IFFT연산 결과는 cyclic prefix의 길

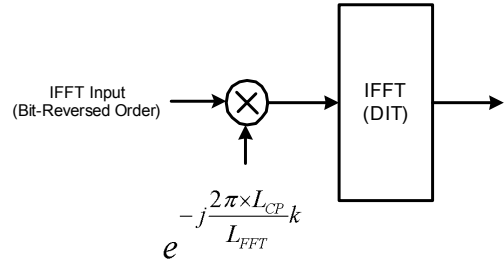


그림 8. IFFT의 입력을 위상 회전

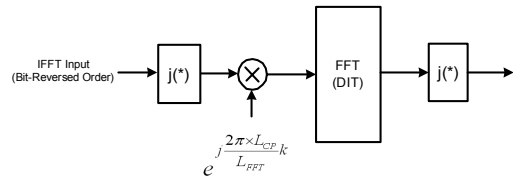


그림 9. FFT의 입력을 위상 회전

이만큼 쉬프트되어 나오게 된다. 즉, IFFT의 결과에서 이미 Cyclic Prefix에 해당하는 부분이 먼저 나오게 되므로, 전체 OFDM심볼은 IFFT에서 나온 값을 그대로 전송한 뒤 앞의 cyclic prefix에 해당하는 부분만 별도로 저장했다가 한 번 더 전송하면 된다. 송신단의 지연은 그림 7의 cyclic suffix를 사용할 때와 같은 값으로 감소되지만 전송되는 OFDM 심볼은 cyclic prefix가 첨부되어 전송되므로 cyclic suffix를 사용할 때와 같은 문제가 없다.

IFFT 뒷단에서도 그림 4에서처럼 FFT 결과 전체를 저장하는 대신 cyclic prefix에 해당하는 부분만 저장하면 되므로 메모리를 절약하는 효과가 있다.

본 논문에서 제시하는 IFFT는 DIT 방식의 FFT 프로세서를 사용하며, IFFT의 출력이 cyclic prefix 구간만큼 시간영역에서 쉬프트되어 나오므로 편의상 TS-DIT IFFT(Time Shifted DIT IFFT)로 부르기로 한다.

#### 3.2 부가적인 하드웨어를 쓰지 않는 방법

IFFT를 수행할 때, 위상회전을 위해 IFFT의 입력값에 미리 (7)의 값을 곱하는 것은, (7)의 k도 역비트 순서이므로 구현이 간단하지 않으며 부가적인 복소 곱셈기와 회전인자 테이블이 필요하다는 단점이 있다.

그런데, IEEE 802.16e Mobile WiMax<sup>[7]</sup>, IEEE 802.11a 무선랜<sup>[10]</sup>, DVB-T<sup>[11]</sup>등 많은 OFDM 시스템에서는 Cyclic Prefix의 길이(그림 1의 TG)가

FFT 크기(그림 1의 Tsub)의 1/2<sup>n</sup>인데, 이런 경우에는 (3.2)에서 위상회전을 위해 곱해주는 값이 k에 따라 주기성을 가지게 된다. 따라서 부가적인 Complex 곱셈기 대신 FFT 프로세서에서 회전인자를 곱하는 과정에서 위상 회전값을 함께 곱할 수 있다.

예를 들어 그림 10은 FFT Size는 8이고 cyclic prefix의 길이가 2인 경우의 IFFT 프로세서의 Radix-2 FFT부분을 예로 든 것이다. cyclic prefix가 FFT 크기의 1/4이므로 FFT 입력 신호의 위상을 회전시키기 위해 곱해지는 인자들(WN0, WN4, WN6, WN2)은 4의 주기를 가진다. 즉,  $\hat{y}(0)$ 와  $\hat{y}(4)$ 에 같은 값이 곱해지고,  $\hat{y}(1)$ 과  $\hat{y}(5)$ 에도 같은 값이 곱해지게 된다. 따라서 이 회전인자들이 그림 11에서처럼 첫 번째 Butterfly 뒤에서 FFT의 고유한 회전인자(Twiddle Factor)와 함께 곱해져도 됨을 알 수 있다.

추가로 곱해지는 회전인자의 값과 곱하는 위치는 cyclic prefix의 크기에 따라 달라진다. 예를 들어 cyclic prefix의 길이가 FFT 크기의 1/4이면 주파수 영역에서 곱해지는 회전인자의 주기는 4가 되고, 1/8이면 8이 된다. 즉 그림 10에서 cyclic prefix가 1/4이기 때문에 입력 x0와 x4에 같은 위상 회전인

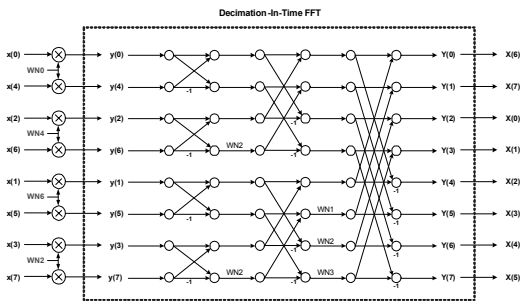


그림 10. 출력을 쉬프트하는 Radix 2 FFT의 예

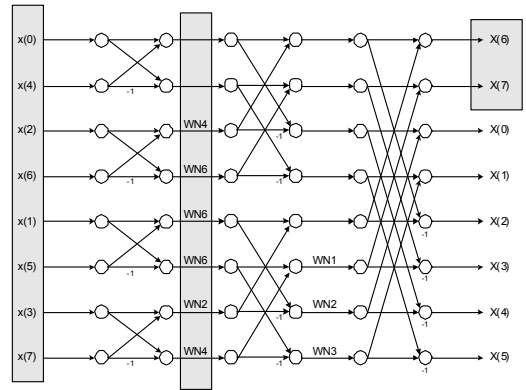


그림 11. 부가적인 복소 곱셈기 없이 출력을 쉬프트하는 Radix 2 FFT

자가 곱해지게 되므로 추가적인 회전인자는 첫번째 Butterfly단을 전후하여 곱해지면 된다. 만약 cyclic prefix의 길이가 1이거나 FFT크기의 1/2<sup>n</sup>이 아니면 입력 데이터에 곱해지는 회전인자의 주기 특성이 없게되므로 별도의 복소 곱셈기를 쓸 수 밖에 없다.

이 때 회전인자의 최소값은 다음과 같이 정의한다.

$$WN = e^{-\frac{j2\pi}{8}} \quad (8)$$

회전인자를 곱하는 위치에 대해 좀더 자세히 살펴보면, IEEE 802.16e Mobile WiMAX의 1024 Point FFT, cyclic prefix 1/8 의 경우에는 7번째 Butterfly 이전 단계에 바뀐 회전인자를 곱하면 되고, IEEE 802.11a 무선랜의 64 Point FFT, cyclic prefix 1/4의 경우에는 5번째 Butterfly 이전 단계에 바뀐 회전인자를 곱하면 된다. 이 관계를 그림으로 표시하면 그림 12와 같다.

이러한 특성을 이용하면 cyclic prefix의 길이 조

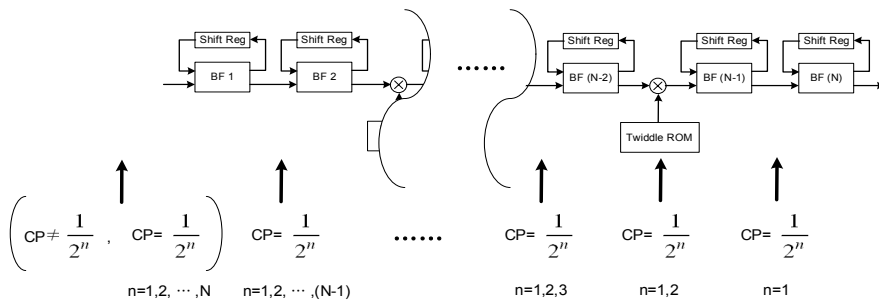


그림 12. Cyclic Prefix의 길이에 따라 위상회전이 첨부될 수 있는 위치

건이 맞을 경우 부가적인 복소곱셈기를 사용하지 않고 FFT 연산 수행 과정에서 적절한 회전인자를 선택하는 것만으로 FFT 프로세서의 결과를 cyclic prefix 길이만큼 쉬프트시킬 수 있다. 물론 불가피하게 부가적인 복소곱셈기를 사용하는 경우에도 송신단의 지연을 줄이고 cyclic prefix를 첨부하기 위해 필요한 메모리의 크기를 줄이는 효과가 있음에는 변함이 없다.

#### IV. 구현 및 시뮬레이션 결과

##### 4.1 Mobile WiMax용 IFFT의 구현

3장에서 제시된 방법에 따라 IEEE 802.16e Mobile WiMax의 송신단용 IFFT를 설계하였다. Mobile WiMax는 1024크기의 FFT를 사용하며 Cyclic Prefix길이는 FFT크기의 1/8인 128 샘플이므로 Cyclic Prefix가 더해진 OFDM 심볼의 길이는 1152 샘플이 된다.

입력 비트의 크기는 14비트로, 이는 Mobile WiMax에서 요구하는 64 QAM의 정상도와 3dB만큼 Boosting되어있는 파일럿 부채널(Pilot subcarrier) 및 프리앰블(Preamble) 신호를 처리하기 위해 요구되는 최소값등을 고려한 값이다. FFT의 회전인자(Twiddle Factor)의 값도 14비트이다<sup>8)</sup>. 출력값은 18비트이며, D/A 변환기로 전송되는 최종 출력 16비트는 필요에 따라 IFFT의 출력으로부터 조절할 수 있게 하였다.

구현은 VHDL을 사용해 Radix-2 Square 구조<sup>6)</sup>로 설계하였고, 동작시클 속도는 10 MHz Clock이다. 이 IFFT는 기지국 송신단과 단말국 송신단 모두에서 사용 가능하다. Radix-2 Square FFT는 필요한 복소곱셈기의 개수가 Radix-4 FFT와 같은 반면 Butterfly의 구조가 더 간단한 장점이 있다. 구조는 Pipelined FFT인 SDF (Single Path Delay Feedback) 구조이다<sup>12)</sup>.

FFT크기의 1/8 Cyclic Prefix인 128 샘플만큼 쉬프트된 출력을 얻기 위해서는 IFFT입력에 128 주기로 동일한 위상회전값을 곱해야하므로 8번째 Butterfly 이전에 Time-shift용 회전인자를 곱해야 한다.

따라서 7번째 Butterfly 앞에 있는 세 번째 회전인자 테이블에 수정된 회전인자를 곱한다. 첫번째나 두번째 회전인자 테이블 값에 위상회전값을 곱해도 되지만, DIT방식의FFT에서는 뒷쪽의 회전인자 테이블크기가 더 작기 때문에 가급적 뒷쪽의 테이블을 수정하기로 하였다.

수정된 회전인자는 FFT고유의 회전인자 64개에 [WN0(8개) WN512(8개) WN256(8개) WN768(8개) WN128(8개) WN640(8개) WN384(8개) WN896(8개)]를 더 곱한 값이다. 입력값이 역비트 순서이기 때문에, 곱해지는 위상은  $WN128 * [0\ 4\ 2\ 6\ 1\ 5\ 3\ 7]$ 이 된다.  $[0\ 4\ 2\ 6\ 1\ 5\ 3\ 7]$ 는  $[0\ 1\ 2\ 3\ 4\ 5\ 6\ 7]$ 의 3비트 역비트 순서이다.

이 값들을 Radian으로 표현하면 다음과 같다.

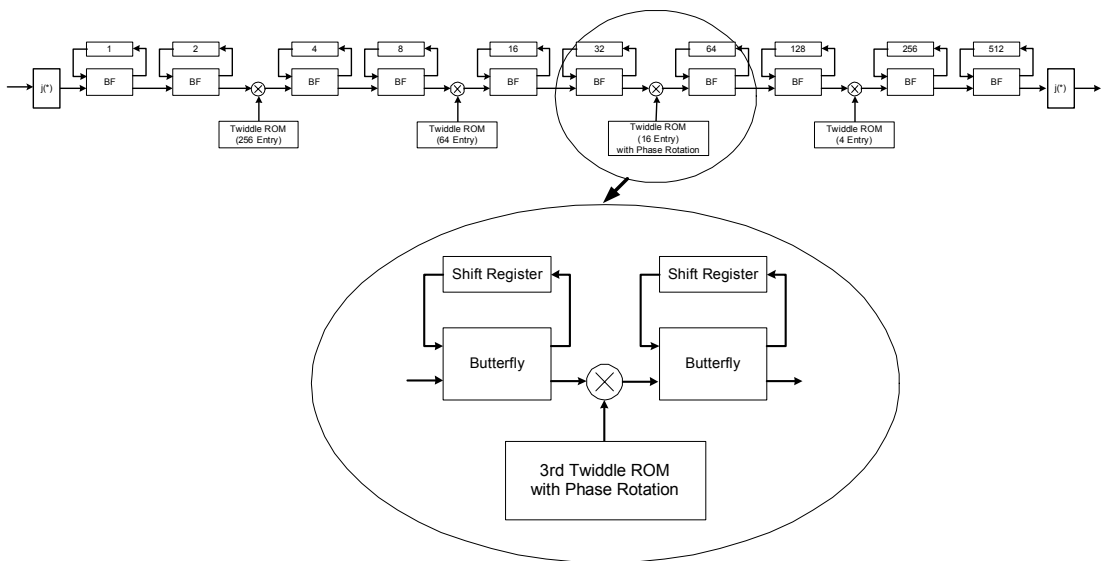


그림 13. 1/8 Cyclic Prefix 첨부기능을 가진 1024 Point Radix-2 Square IFFT 프로세서의 구조

$$\left[ 0 \quad \pi \quad \frac{\pi}{2} \quad -\frac{\pi}{2} \quad \frac{\pi}{4} \quad -\frac{3\pi}{4} \quad \frac{3\pi}{4} \quad -\frac{\pi}{4} \right]$$

### 4.2 시뮬레이션 결과

본 논문에서 제안한 IFFT와 기존의 방법으로 DIF IFFT를 이용해 설계된 OFDM 송신단을 시뮬레이션한 결과를 비교하였다.

그림 14는 시뮬레이션을 수행한 환경이며, 그림 15는 Modelsim으로 시뮬레이션을 수행한 결과이다. 두 IFFT는 모두 VHDL로 설계되었으며 IFFT의 입력과 출력 비트 및 회전인자(Twiddle Factor) 테이블의 크기, Rounding 방법<sup>1)</sup>등을 동일하게 구현하였다.

그림 15의 (A)는 본 논문에서 제시한 TS-DIT 방법으로 구현한 IFFT의 입력으로, 역비트 순서이며, (B)는 이 IFFT의 출력이다. (B)는 Cyclic Prefix 부분이 먼저 출력되기 때문에 송신단의 최종 출력인 (C)를 생성하기 위해 별도의 Delay가 없음을 알 수 있다. 최종적으로 Cyclic Prefix부분을 뒤에 첨부하여 OFDM 심볼을 완성한 것이 (C)이다.

반면, 통상적인 DIF FFT를 이용해 송신단을 구현한 경우, 입력(D)가 순차적으로 들어가면 출력 (E)가 역비트 순서로 나오고, 이를 재정렬하고 cyclic prefix를 첨부하는 과정을 끝내 최종 출력인 (F)가 구해진다. 따라서 (F)는 (C)와 동일한 값을 가지지만 1024 Cycle만큼의 지연을 더 요구함을 알 수 있다. 이 차이는 10MHz Clock과 1024 크기의 FFT

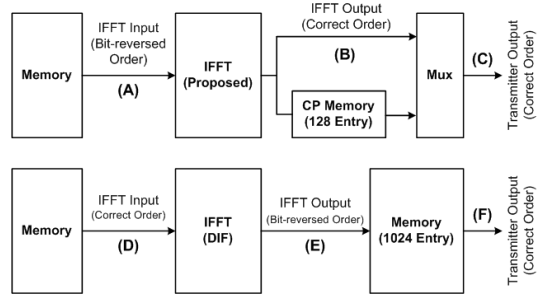


그림 14. 제안한 구조와 기존 구조의 비교해 시뮬레이션한 환경

를 사용하는 IEEE 802.16e Mobile WiMax에서 102.4 us의 시간에 해당한다.

### 4.3 성능 비교

IFFT에서 Cyclic Prefix부분을 먼저 출력하기 위해 위상회전값을 곱하는 과정에서 성능 감소가 없는지를 시뮬레이션으로 확인하였다.

QAM Mapper를 통해 64 QAM으로 변조된 신호를 VHDL로 1K IFFT를 한 후 1/8 Cyclic Prefix까지 첨부된 OFDM 심볼로 완성한다. 그 결과를 Matlab에서 Cyclic Prefix를 제거한 후, 이상적인 FFT로 복조해서 Null Carrier들을 제거하고 구해진 부반송파(Subcarrier)들을 원본 데이터와의 오차를 구해 EVM(Error Vector Magnitude)을 계산하였다. EVM은 두 신호를 비교한 모든 에러의 Root Mean Square값을 구한 것으로, IEEE 802.16e Mobile

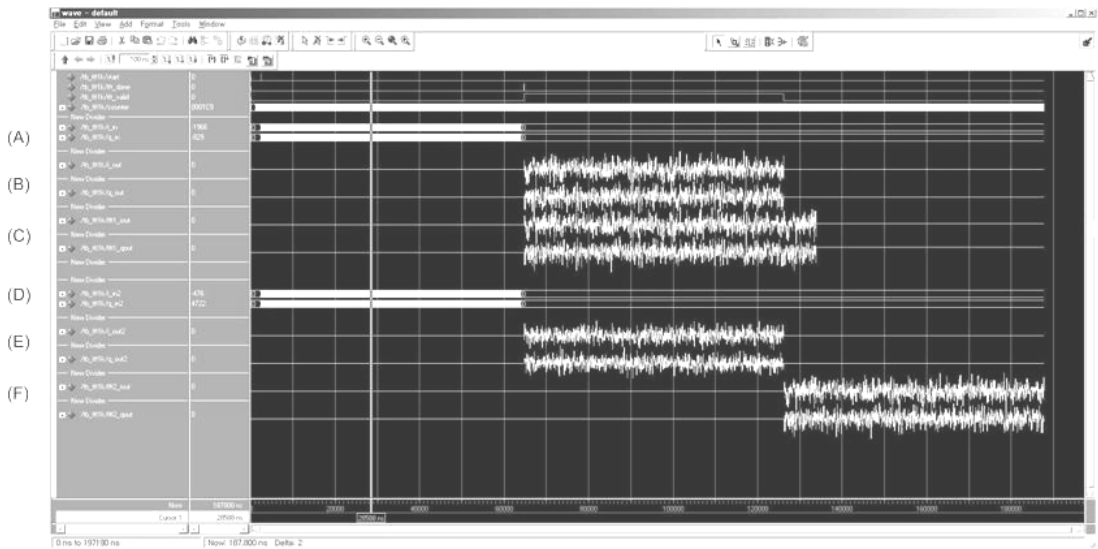


그림 15. TS-DIT IFFT구조와 기존 구조의 송신단 Latency의 비교

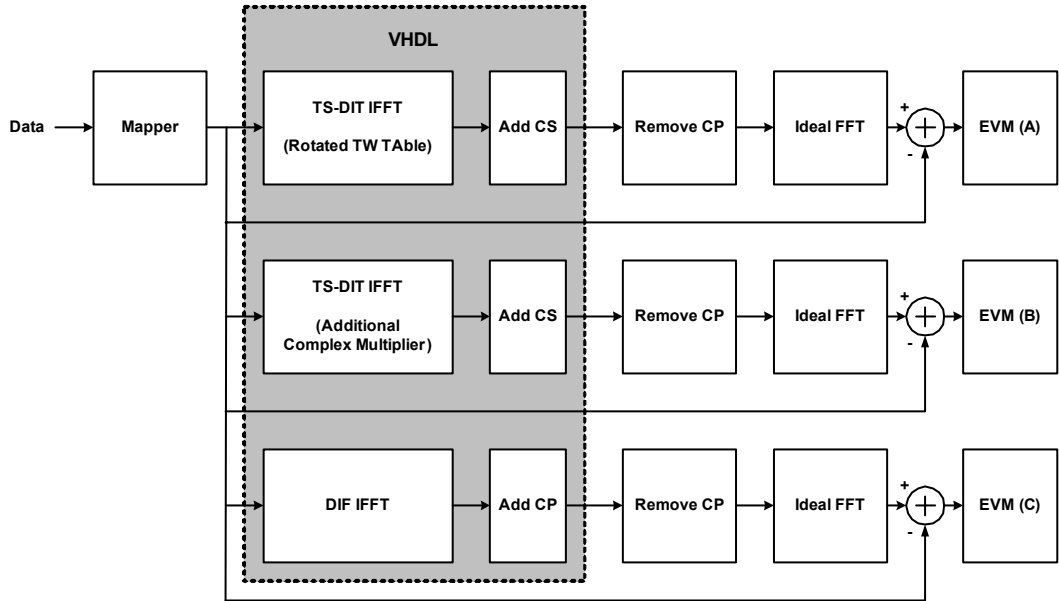


그림 16. 오차를 측정하는 시뮬레이션 환경

WiMax규격에 계산식이 규정되어 있다.

비교한 IFFT는 3가지이다. 그림 13 과 같이 3번째 회전인자(Twiddle Factor) 테이블에 위상회전값을 반영한 TS-DIT (A), 그림 9에서처럼 회전인자 테이블에 위상회전값을 반영하는 대신 별도의 복소곱셈기를 이용해 위상회전을 시켜주는 TS-DIT2 (B), 그림 4와 같이 일반적인 DIF IFFT를 이용한 (C)이다. 세 가지 설계 모두 14 비트 입력, 18비트 출력이며, FFT 내부의 위상회전인자(Twiddle Factor)와 TS-DIT2에서 별도의 위상회전을 위해 사용한 회전인자는 모두 14비트이다.

표 1은 이 세가지 IFFT를 사용했을 때 계산된 EVM값을 보여준다. 다른 선행 구현의 예<sup>[8]</sup>와 비교해도 위상회전이 추가됨으로서 성능열화가 생기지 않는다는 것을 알 수 있다. Cyclic Prefix 가 1/2<sup>n</sup> 길이가 아니라서 불가피하게 부가적인 Complex 곱셈기를 사용하는 TS-DIT2의 경우에도 이로 인한 성능열화는 아주 근소함을 알 수 있다.

표 1. EVM 비교 결과

	TS-DIT (A)	TS-DIT2(B)	DIF (C)
EVM	0.0270%	0.0283%	0.0276%

#### 4.4 하드웨어 복잡도의 비교

본 논문에서 제안한 구조의 IFFT는 Cyclic Prefix가 FFT크기의 1/2<sup>n</sup>인 특성을 이용해 별도의 복소곱셈기를 사용하지 않고 구현될 수 있었다. IFFT의 결과를 저장하는 메모리의 크기는 본 논문에서 제안한 구조를 택했을 경우 128 개의 샘플을 저장하면 되는 반면, 기존 구조의 FFT를 사용하면 1024 개의 샘플을 저장해야 한다. IFFT의 출력이 18비트이므로 그 차이는 16,128 비트에 해당한다.

반면 본 논문에서 제시한 IFFT는 DIT 방식의 FFT 프로세서를 이용해 구현되어야 하므로 피드백 쉬프트 레지스터(Feedback Shift Register)의 크기가 증가하는 단점이 있다. FFT의 각 Butterfly에서 입력이 N비트이면 출력은 (N+1)비트가 되게 되므로,

표 2. 필요한 하드웨어의 비교

	TS-DIT (A)	DIF (B)	Difference (A-B)
Complex Multipliers	4	4	0
Re-Ordering Memory Size	128 x 18 (bits)	1024 x 18 (bits)	-16,128
D F/Fs in Shift Register	36,776	30,556	6,220



FFT의 출력쪽으로 갈수록 Butterfly와 쉬프트 레지스터가 다루는 값은 더 많은 비트를 요구하는데, DIF 방식의 FFT는 길이가 긴 쉬프트 레지스터들이 앞쪽에 위치하는 반면, DIT 방식의 FFT는 길이가 긴 쉬프트 레지스터들이 뒷쪽에 위치하므로 상대적으로 더 많은 면적을 차지하게 된다. 이는 SDF가 아닌 다른 구조로 FFT를 설계해도 동일하게 나타난다. 이 때문에 증가하는 쉬프트 레지스터의 크기는 FFT의 입력 비트 크기와 Rounding에 따라 달라지는데, 본 구현에서 입력이 14비트이고 출력을 18비트로 Saturation시켰을 때 6,220 비트가 된다.

따라서 본 논문에서 제안한 구조는 DIF FFT를 사용하는 방법에 비해 쉬프트 레지스터에서 다소 많은 면적을 필요로 하지만 메모리 사용량이 더 크게 줄어들기 때문에 하드웨어 복잡도 면에서 더 유리함을 알 수 있다.

### V. 결 론

본 논문은 OFDM 시스템의 송신단에서 IFFT를 수행한 뒤, Cyclic Prefix 를 첨부하여 심볼을 생성하는데 드는 지연을 감소시키는 방법을 제안하고 있다. 기존에 이러한 지연을 감소시키는 방식으로 제안된 Cyclic Suffix 방식은 Cyclic Prefix 방식과 달리 OFDM 심볼의 뒷부분을 앞으로 복사하는 대신, 앞부분을 뒷부분으로 복사하는 방식이다. 그러나 이 방식을 사용하기 위해서는 전반적인 OFDM 시스템 규격변경이 불가피하고 FFT 수행한 후의 결과가 위상회전이 발생하게 되므로 수신단에서 대용량의 메모리를 사용하거나 성능열화가 발생할 가능성이 남아있다.

반면에 본 논문에서 제안된 방식은 Cyclic Prefix 방식을 사용하므로 OFDM 시스템의 규격변경이 필요없고 전체 OFDM 심볼의 IFFT 결과값을 그대로 전송한 뒤, 앞의 Cyclic Prefix에 해당하는 부분만 별도로 저장했다가 한번 더 전송하면 되므로 송신단의 지연을 Cyclic Suffix 사용할 때와 동일하게 감소시킬 수 있게 될 뿐만 아니라, 메모리 절약의 효과도 있다. 또한 Cyclic Prefix의 길이가 OFDM 심볼의 1/2인 경우 별도의 복소곱셈기가 필요하지 않아 하드웨어의 복잡도도 줄일 수 있다. 성능열화가 발생하지 않고 송신단의 지연을 줄일 수 있으며 하드웨어의 복잡도 또한 개선할 수 있으므로 본 논문에서 제시된 방식은 앞으로 OFDM 시스템 송신단의 효율적인 구성에 기여를 할 것이라고 생각한다.

### 참 고 문 헌

- [1] Richard van Nee, Ramjee Prasad, "OFDM Wireless Multimedia Communications," Artech House, PP.39-45, 2000.
- [2] 박경원, 박세현, 양원영, 조용수, "Cyclic Suffix 를 사용한 OFDM기반의 25 Mbps 무선 ATM 모뎀의 ASIC Chip 설계," 한국통신학회논문지 00-5 Vol.25 No.5B, pp.859-870.
- [3] Fredrik Kristensen, Peter Nilsson, Anders Olsson, "Reduced transceiver-delay for OFDM systems," IEEE Vehicular Technology Conference, VTC, Spring 2004.
- [4] 위정욱, 양원영, 백종호, 유영환, 조진웅, 조용수, "효율적 Cyclic Extension을 갖는 Zipper 방식의 VDSL모뎀," 한국통신학회논문지 00-10 Vol.25 No.10B, pp.1793-1802.
- [5] Alan V. Oppenheim, Ronald W. Schaffer, "Discrete-Time Signal Processing, Second Edition," PP.650-652, Prentice Hall, 1999.
- [6] Shouheng He, Mats Torkelson, "A New Approach to Pipeline FFT Processor," The 10th Proceedings or IPSP, 1996, pp.766-770.
- [7] IEEE Standard 802.16-2004, "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," 2006.2.
- [8] 박형숙, 박윤옥, 김철성 "OFDMA/TDD 시스템의 하향링크 빔형성 송신기 설계," 한국통신학회 논문지 06-5 Vol.31 No.5A, pp.493-500.
- [9] IEEE Standard for Binary Floating-Point Arithmetic. New York, ANSI/IEEE 754-1985, 1985.
- [10] IEEE Standard 802.11a-1999: Wireless LAN MAC and PHY Specifications-High Speed Physical Layer in the 5 GHz Band, New York, IEEE, 2000.
- [11] ETSI, "Digital video broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television," EN 300 744, V1.4.1 Jan., 2001.
- [12] Seungbeom Lee, Sin-Chong Park, "Modified SDF Architecture for Mixed DIF/DIT FFT," Circuits and Systems, 2007, pp.2590-2593

김 준 우 (Jun-Woo Kim)

정회원



1996년 2월 경북대학교 전자공학과 학사

1998년 2월 한국과학기술원 전기 및 전자공학과 석사

2006년 3월~현재 충남대학교 정보통신공학부 박사과정

1998년 1월~2001년 9월 데이

콤 종합연구소 연구원

2001년 10월~현재 한국전자통신연구원 연구원

<관심분야> 디지털 신호처리, 디지털 통신

김 환 우 (Whan Woo Kim)

중신회원



1977년 2월 서울대학교 전자공학과 학사

1979년 2월 한국과학기술원 전기 및 전자공학과 석사

1988년 6월 University Of Utah 전자공학과 박사

1980년 6월~현재 충남대학교 정

보통신공학부 교수

<관심분야> 디지털 신호처리, 초고속 디지털 통신

박 윤 옥 (Youn-Ok Park)

정회원



1986년 2월 한양대학교 전자공학과 학사

1997년 2월 충남대학교 컴퓨터 공학과 석사

2001년 3월~현재 충남대학교 정보통신공학과 박사과정

1985년 12월~1987년 1월 삼성

전자 종합연구소 연구원

1987년 2월~현재 한국전자통신연구원 팀장(책임연구원)

<관심분야> 디지털 통신, CDMA, OFDMA, MIMO