

CORDIC 알고리즘에 기반한 DVB-T용 2K/4K/8K-Point FFT 프로세서

박상윤, 조남익

서울대학교 전기컴퓨터공학부

2K/4K/8K-Point FFT Processor Based on the CORDIC Algorithm for DVB-T

Sang Yoon Park, Nam Ik Cho

School of Electrical Engineering & Computer Science, Seoul National University

e-mail : sanguni@ispl.snu.ac.kr, nicho@snu.ac.kr

요약

본 논문에서는 OFDM 시스템용 복조기의 구현에 가장 핵심적인 소자인 2K/4K/8K-point FFT 프로세서를 제안하였다. 구현된 프로세서는 30MHz 시스템 클럭에서 8,192개의 복소입력 샘플을 273μs에, 2,048개의 복소 입력 샘플을 68.26μs에 수행함으로써 OFDM 방송에서 요구하는 심볼 레이트의 데이터를 처리할 수 있다. 기본 구조는 1차원 DFT를 작은 크기의 2차원 DFT로 변환할 수 있는 콜리-투키 알고리즘을 적용하였으며 다차원 DFT 변환에 적합한 전치 메모리와 서플 메모리를 사용하였다. 복소 곱셈기는 기존의 방법보다 더 효율적인 메모리 구조를 갖는 CORDIC 프로세서를 사용하였으며 제안하는 트위들팩터 발생 방법은 트위들팩터를 저장하기 위한 ROM의 크기를 효과적으로 줄일 수 있다.

1 서론

DMT (Discrete multitone)는 xDSL과 디지털 비디오/오디오 등의 디지털 데이터 전송 시스템에 광범위하게 사용되고 있다. 특히, OFDM (Orthogonal Frequency Division Multiplexing)은 유럽의 DAB (Digital Audio Broadcasting) 및 DVB-T (Digital Terrestrial Video Transmission)에서 기본 변복조 방식으로 사용되고 있다[1]. OFDM은 광대역의 채널을 다수의 협대역 부채널로 분할하고 서로 직교하는 다수의 부반송파를 각 협대역 채널에 할당, 변조하여 다중화하는 기법이다. 직교하는 다수의 부반송파를 얻는 과정은 DFT (Discrete Fourier Transform)에 의해 수행되므로 저가, 저전력 고성능의 경쟁력 있는 FFT 프로세서의 구현은 OFDM 시스템에서 가장 중요한 문제라고 할 수 있다. OFDM 시스템의 변복조기에서 다수의 부반송파를 얻기 위한 DFT 변환을 단순히 병렬적으로 구현하는 것은 거의 불가능하므로 부반송파의 수보다 큰 2ⁿ 크기의 FFT 프로세서를 이용하여 구현한다. 그림 1은 FFT 프로세서를 사용한 OFDM 시스템의 변복조 블록도이

다. 각 직교 부반송파는 IFFT (Inverse Fast Fourier Transform)에 의해 동시 발생되며, 반송파들의 직교성을 이용하여 대역 통과 필터에 의해서가 아니라 기저대역 신호처리에 의해 변조와 다중화가 이루어진다. 본 논

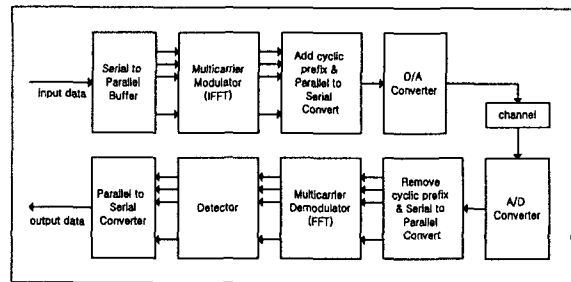


그림 1: OFDM 시스템의 변복조 블록도

문에서 구현하는 FFT 프로세서는 2K-point FFT를 기본 구조로 제작되었으므로 유럽형 OFDM용 2K/8K-point FFT뿐만 아니라 일본의 4K-point FFT 및 각종 실시간 신호처리 응용에 이용될 수 있다.

본 논문의 2장에서는 제안하는 시스템의 FFT 알고리즘에 대해 살펴보고, 3장에서는 전체 구조, 복소 곱셈기 및 메모리 시스템을 상세히 설명한다. 4장에서는 제안하는 시스템의 성능을 분석하고 5장에서 결론을 맺는다.

2 FFT 알고리즘

N-point DFT는 다음과 같이 주어진다.

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

여기서 $W_N = e^{-j(2\pi/N)}$ 이며, $x(n)$ 과 $X(k)$ 는 복소수로서 각각 DFT의 입출력을 나타낸다. DFT를 빠르게

수행하기 위한 FFT 알고리즘에 관한 연구들은 지금까지 많이 수행되었으며 일반적으로 DIT (Decimation-in-Time) 알고리즘, DIF (Decimation-in-Frequency) 알고리즘, 콜리-투키 (Cooley-Tukey) 알고리즘[2], Winograd 알고리즘 등이 널리 알려져 있다. 본 논문에서는 큰 크기의 DFT를 다차원의 짧은 크기의 DFT로 변환하여 계산량을 줄일 수 있는 콜리-투키 알고리즘을 적용한다. 즉 콜리-투키 알고리즘을 적용함으로써 2K 또는 8K의 DFT를 아주 작은 크기의 DFT들로 변환한다. 그런데 문제의 크기를 나누는 방법과 기본 DFT 유닛 (unit)의 크기에 따라서 시스템의 구성이 많이 달라질 수 있으므로 OFDM에 적합한 구조를 도출하는 것이 중요하다.

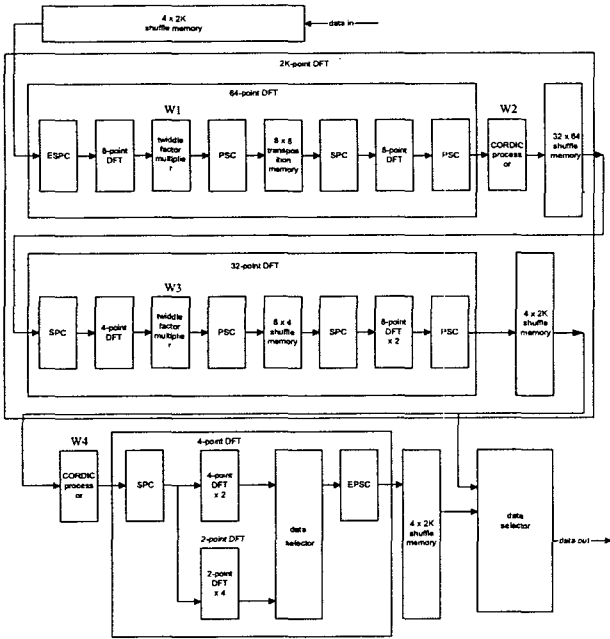


그림 2: 제안하는 2K/4K/8K-point FFT 프로세서의 블록도

3 구조

본 장에서는 고성능 연산처리를 위해 병렬 구조를 가지며, 메모리의 요구 사항 및 복소 곱셈을 위한 하드웨어 비용을 최소화한 OFDM용 2K/4K/8K-point FFT 프로세서의 구조를 소개한다.

3.1 전체 구조

제안하는 FFT 프로세서의 주요 구성 요소는 짧은 길이의 DFT, 전치 메모리 (TM : Transposition Memory), 셔플 메모리 (SM : Shuffle Memory) 및 복소 곱셈기이다. 제안하는 2K/4K/8K 복소 FFT 프로세서의 블록도가 그림 2에 나타나 있다. 8K-point DFT는 콜리-투키 알고리즘에 의해 4-point DFT와 2K-point DFT의

2차원 DFT로 나누어 지고 2K-point DFT는 다시 64-point DFT와 32-point DFT로 나누어 진다. 64-point DFT와 32-point DFT는 다시 더 작은 DFT로 나누어 진다. 제안하는 구조의 기능 블록은 4-point DFT, 8-point DFT, 복소 곱셈기(그림 2의 W1, W2, W3, W4) 그리고 메모리 컨트롤러이다. 64-point DFT는 행 우선 순서와 열 우선순서 사이의 변환을 위해 전치 메모리를 사용하며 행과 열의 크기가 다른 변환에는 셔플 메모리를 사용한다. 각 연산기와 메모리 사이에는 효율적인 파이프라이닝 (pipelining)을 위해 순차병렬변환기 (SPC : Serial-to-Parallel Converter)와 병렬순차변환기 (PSC : Parallel-to-Serial Converter)가 위치한다.

복소 곱셈기와 전치 메모리, 셔플 메모리는 다음 장에서 자세히 설명하며 순차병렬변환기와 병렬순차변환기에 대한 설명은 생략한다.

3.2 복소 곱셈기

4-point DFT와 8-point DFT는 radix-2를 기반으로 병렬구조를 갖으며 DIT 알고리즘을 적용하여 구현한다. 한편, 본 논문에서는 하드웨어의 낭비 없이 높은 정밀도를 가지는 복소 곱셈기를 구현하기 위해 세 가지의 다른 경우에 대해 각각 다른 방법의 복소 곱셈 방법을 적용한다.

3.2.1 두 개의 실수 곱셈을 갖는 복소 곱셈기

8-point DFT에서 복소 곱셈기는 $\pm j$ 를 제외하면 $\cos(0.75\pi) \pm j \sin(0.75\pi)$ 의 실수부분과 허수부분이 동일한 트위들 팩터 (twiddle factor)를 갖게 된다. 그러므로 8-point DFT는 두 개의 실수 곱셈만을 갖는 복소 곱셈기를 이용하여 구현할 수 있다. 그림 3은 8-point DFT에서 부스 알고리즘 (booth algorithm)을 이용하여 구현한 2개의 실수 곱셈만을 갖는 복소 곱셈기의 블록도를 나타낸다.

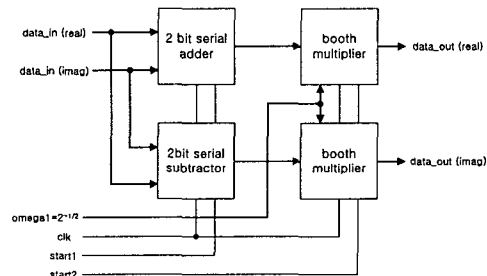


그림 3: 두 개의 실수 곱셈을 갖는 복소곱셈기의 블록도

3.2.2 세 개의 실수 곱셈을 갖는 복소 곱셈기

64-point DFT와 32-point DFT에서 복소 곱셈기 (그림 2의 W1, W3)는 3.2.1절과 마찬가지로 부스 알고리즘을 이용한 실수 곱셈기를 이용한다. 특히 간단한 삼

각함수의 성질을 이용하면 실수 곱셈의 개수를 줄일 수 있다. 트위들 팩터의 곱셈은 다음과 같이 표현된다.

$$A \times W_N^{nk} = (A_R + jA_I) \times (\cos \theta - j \sin \theta) \\ = (A_R \cos \theta + A_I \sin \theta) + j(A_I \cos \theta - A_R \sin \theta) \quad (2)$$

여기서 $A = A_R + jA_I$ 로서 곱셈기의 입력을 나타내고 $W_N^{nk} = e^{-j\frac{2\pi}{N}nk} = e^{-j\theta}$ 로서 트위들팩터를 나타낸다. 식 (2)의 우변은 적절한 변형을 통해 다음과 같이 나타낼 수 있다.

$$A_R \cos \theta + A_I \sin \theta = A_R(\sin \theta + d) + A_I \sin \theta \\ = (A_R + A_I) \sin \theta + A_R d \quad (3)$$

$$A_I \cos \theta - A_R \sin \theta = A_I \cos \theta - A_R(\cos \theta - d) \\ = (A_I - A_R) \cos \theta + A_R d \quad (4)$$

여기서 $d = \cos \theta - \sin \theta$ 와 같다. 기존의 복소 곱셈은 4개의 실수 곱셈을 필요로 하지만 사인함수와 코사인함수의 차를 이용하면 식 (3)과 식 (4)와 같이 복소 곱셈이 3개의 실수 곱셈만을 포함하게 된다. 이와 같이 삼각함수의 성질을 이용하면 표 1과 같이 약 23%의 하드웨어를 절감할 수 있으며 12비트의 무작위 정수 샘플에 대해 약 0.007만큼 낮은 오차율을 얻을 수 있다.

3.2.3 CORDIC 프로세서

2K, 4K 및 8K-point DFT의 트위들 팩터 곱셈기 (그림 2에서 W_2, W_4)는 트위들 팩터를 저장하기 위해 많은 양의 ROM이 필요하므로 기존의 부쓰 곱셈기를 이용하는 방법은 메모리를 효율적으로 사용하기 위한 VLSI 구조로는 적당하지 않다. 예를 들면, 실수 곱셈기 4개를 필요로 하는 복소 곱셈기는 트위들 팩터 마다 2-word의 ROM (cos값과 sin값)을 필요로 하므로 8K-point FFT의 경우 81,920-bit의 ROM이 필요하다. 본 논문에서는 FFT의 트위들 팩터 곱셈을 2×1 벡터의 회전변환의 꼴로 변형함으로써 하드웨어 비용을 줄일 수 있는 CORDIC (COordinate Rotation DIgital Computer) 알고리즘을 이용한다[3]. CORDIC 알고리즘은 트위들 팩터의 회전각을 미리 정의된 기본각들로 분할함으로써 쉬프트(shift)와 덧셈연산만으로 복소 곱셈을 수행할 수 있다. CORDIC 알고리즘은 기본각 만큼 연속적으로 회전 변환을 수행하는 반복 과정과 입력과 출력의 크기를 동일하게 하기 위해 크기 팩터를 곱하는 과정으로 나뉘게 된다. CORDIC의 반복 과정을 요약하면 다음과 같다.

$$x_{i+1} = x_i - \sigma_i \cdot 2^{-i} \cdot y_i \\ y_{i+1} = y_i + \sigma_i \cdot 2^{-i} \cdot x_i \\ z_{i+1} = z_i - \sigma_i \cdot \alpha_i \\ \alpha_i = \tan^{-1}(2^{-i}) \\ \sigma_i = \text{sign}(\phi_i) \quad (5)$$

여기서 $[x_i, y_i]$ 는 입력 벡터이고 이를 $\sigma_i \cdot \alpha_i$ 만큼 회전하여 $[x_{i+1}, y_{i+1}]$ 의 출력 벡터를 얻는다. 기존의 CORDIC

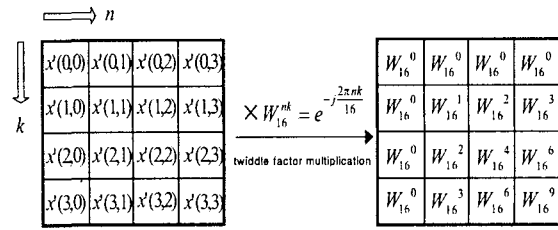


그림 4: 트위들 팩터 발생 방법

프로세서는 17번의 반복 과정을 수행한다고 가정할 때 σ_i 를 저장하기 위한 2560-word 17-bit의 ROM과 부가적인 하드웨어를 필요로 하게 된다. 본 논문에서는 이러한 ROM의 사용을 막기 위해 트위들 팩터 발생 방법을 제안한다. 제안하는 방법은 식 (1)에서 $n \times k$ 값을 안다면 CORDIC 알고리즘의 적절한 스케일링 과정을 통해 회전변환을 수행할 수 있다는 점에 착안한다. 그림 4에서 보듯이 $n \times k$ 값은 메모리 주소의 행의 값과 열의 값의 곱으로 나타내어 질 수 있으므로 트위들 팩터 발생은 그림 5와 같이 counter와 circular shifter를 이용하여 쉽게 구현할 수 있다. 제안하는 트위들 팩터 발생 방법은 콜리-루키 알고리즘 뿐만 아니라 다른 FFT 알고리즘에도 폭넓게 적용할 수 있다. 제안하는 방법을 이용한다면 앞서 언급하였듯이 기존의 방법과 비교하여 ROM을 사용할 필요가 없어지므로 word-line decoder나 pull up register와 같은 부가적인 하드웨어의 비용 또한 줄일 수 있다. 더욱이 표 1에서 보듯이 복소 곱셈기를 구현하는 데 필요한 하드웨어의 양은 기존의 방법에 비해 그리 크지 않음을 알 수 있다.

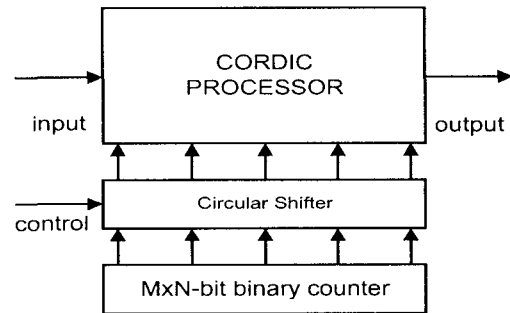


그림 5: 트위들 팩터 발생 방법을 사용하는 CORDIC 프로세서의 구조

3.3 메모리 컨트롤

콜리-루키 알고리즘을 이용하여 DFT를 구현하는 데 있어 효과적인 메모리 (RAM)의 사용과 제어의 용이성 또한 중요한 문제 중의 하나이다. 본 논문에서는 콜리-루키 알고리즘의 행방향과 열방향의 순서의 전환을 용이하게 하기 위해 [4]에서 제시된 전치 메모리와 셔플 메모리를 사용한다. $N \times N$ DFT와 같이 행의 수

표 1: 복소곱셈기의 하드웨어 비용 (sin값과 cos값에 각각 16 bit를 할당).

	Conventional approach (real multiplier×4)	Proposed scheme	
		Real multiplier×3 (W1, W3)	CORDIC (W2, W4)
ROM size	81,920-bit	0-bit	0-bit
Complex multiplier	23,589 gates	18,171 gates	26,974 gates

와 열의 수가 동일할 때에는 그림 2에서 보듯이 기존의 전치 메모리를 사용하지만 $M \times N$ DFT와 같이 M 과 N 이 서로 다른 경우에는 전치 메모리의 일반화된 형태인 셔플 메모리를 사용한다. 셔플 메모리의 기본 기능 블록은 3-T DRAM을 사용하는 메모리 셀 어레이, 주소 발생기, 워드 라인 디코더, 입력 데이터 버퍼, 감지 증폭기 (sense amplifier) 그리고 버스 드라이버로 구성되어 있다. 특히 3-T DRAM을 사용하는 메모리 셀 어레이는 읽기 모드의 비트 라인이 쓰기 모드의 비트 라인과 서로 독립적으로 작동하기 때문에 클럭 주기를 반으로 분할하여 같은 주소에서 연속적인 읽고 쓰기가 가능하게 된다. 8K-word 셔플 메모리의 전체적인 구조가 그림 6에 나타나 있다. 그림 6에서 보듯이 8K-word 셔플 메모리는 4 개의 2K-word 메모리 블록으로 분할되어 있으며 블록 선택 신호에 따라 8K 및 2K-word 메모리로서 동작한다. 더욱이 제안하는 메모리 시스템의 주소 발생기는 OFDM 전송시 데이터의 지연 확산 왜곡 (delay spread distortion)을 막기 위한 보호 구간 (guard interval)에도 데이터의 배열이 꼬이지 않도록 설계 되어 있다.

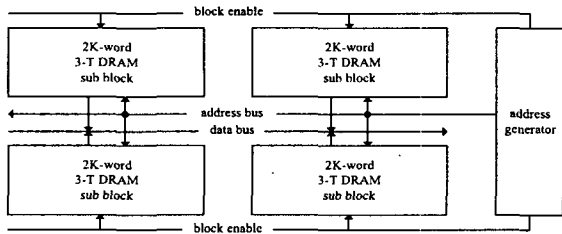


그림 6: 8K-word 셔플 메모리의 구조

4 시스템 성능

DFT의 분산 전달 모델 (variance propagation model)[5]을 통해 제안하는 시스템의 정밀도는 입력을 2×10 bit, 출력을 2×12 bit, 내부 정밀도를 2×16 bit로 결정하였다. 한편 CORDIC의 반올림 오차 (rounding error)와 근사 오차 (approximation error)를 분석한 결과 기존의 복소 곱셈기와 동일한 오차율을 얻기 위해 DFT의 내부 정밀도에 실수부분과 허수부분 각각 5 bit씩을 첨가하고 iteration 과정을 17번으로 결정하였다[6]. 그 결과

제안하는 FFT 프로세서는 디지털 지상파 방송에 충분한 약 49dB의 SNR (Signal to Noise Ratio)을 얻을 수 있었다. 제안하는 구조는 1.8V 0.18 μ m의 CMOS 기술을 통해 합성하였고 그 결과 제안하는 시스템은 logic과 메모리를 합쳐 약 600,000 게이트의 하드웨어 복잡도를 갖는다.

5 결론

본 논문에서는 OFDM 시스템의 복조기에서 사용하는 FFT 프로세서의 구조를 제안하고 구현하였다. 제안한 구조는 쿨라-투키 알고리즘에 기반하며 간단한 조작으로 쉽게 2K/4K/8K-point FFT 처리가 가능하다. 트워드 팩터 곱셈용 복소 곱셈기는 하드웨어의 증가를 억제하기 위해 부쓰복소 곱셈기와 CORDIC 프로세서를 사용하였으며 CORDIC 프로세서의 ROM의 양을 줄이기 위해 새로운 트워드 팩터 발생 방법을 제안하였다. 제안한 2K/4K/8K-point FFT 프로세서를 1.8V 0.18 μ m CMOS 기술을 통해 구현하였고 약 600,000 게이트의 하드웨어가 소요된다. 제안한 프로세서는 30MHz의 시스템 클럭에서 8K-point FFT를 매 273 μ s마다 수행하고 2K-point FFT를 매 68.26 μ s마다 수행함으로써 OFDM에서 규정하는 심볼 레이트를 충분히 넘어선다.

참조 서적

- [1] M.Alard and R.Lasalle, "Principles of modulation and channel coding for digital broadcasting for mobile receivers," *ITU WARC-ORB Conference.*, Sep. 1988.
- [2] J.W.Cooley and J.W.Tukey, "An algorithm for machine computation of complex Fourier series," *Math. Comput.*, vol. 19, 1965
- [3] Y. H. Hu, "CORDIC based VLSI architecture for digital signal processing," *IEEE Signal Processing Mag.*, pp. 16-35, July 1992.
- [4] Kichul Kim, "Shuffle memory system," *13-th International Parallel Processing Symposium.*, p p.268-272, April, 1999.
- [5] T.Thong and B.Liu, "Fixed-point fast Fourier transform error analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp.563-573, Dec. 1976.
- [6] Y.H.Hu, "The quantization effects of the CORDIC algorithm," *IEEE Transactions on Signal Processing.*, vol. 40, No. 4, April 1992.