

Pipeline 방식 256-point FFT Processor의 설계

서정훈, 송인채

승실대학교 전자공학과

전화 : (02) 816-6073 / 팩스 : (02) 821-7653

Design of a 256-point FFT Processor

Jung-Hoon Seo, Inchaeh Song

Department of Electronic Engineering, Soongsil University

E-mail : hoon@hanul.soongsil.ac.kr

Abstract

In this paper, we designed a 256-point FFT processor using VHDL. We adopted Radix-2² SDC(Single-path Delay Commutator) architectures to reduce the number of complex multipliers. We confirmed the operation of the design through simulation using Altera MAX+PLUS II.

I. 서론

이동 통신의 발전으로 인하여 고 품질의 이동 멀티미디어 서비스를 위한 통신 시스템에서 고속으로 데이터를 송신하고 수신함에 따라 발생하는 인접 심벌간의 간섭(ISI : Inter-Symbol Interference) 왜곡을 줄이기 위한 방식으로 직교 주파수 분할 다중화(OFDM : Orthogonal Frequency Division Multiplexing) 방식에 대한 연구가 활발하게 진행되고 있다.[1] OFDM 시스템 내부 반송파의 수 즉, N의 증가에 따라, 많은 연산량을 효율적으로 처리할 수 있으면서 면적 및 전력 소모의 관점에서보다 효율적인 N-Point FFT 프로세서 구조에 대한 연구가 요구되고 있다. N이 비교적 큰 경우에는 파이프라인 구조의 FFT 프로세서가 면적 및 처리 속도면에서 가장 효율적인 것으로 알려져 있다.

파이프라인 구조의 FFT 프로세서에는 Radix-2 MDC(Multi-path Delay Commutator), Radix-2 SDF(Single-path Delay Feedback), Radix-4 MDC, Radix-4 SDF, Radix-4 SDC(Single Delay Commutator), Radix-2² SDC 구조들을 이용한 것들이 제안되어 있다. Radix-2² SDC 구조로 설계할 경우 요구되어지는 버터플라이의 단수는 N point의 경우 $\log_2 N$ 이 되며 각 단마다 필요한 복소승산기의 수가 반으로 주는 장점이 있다.[2] 파이프라인(pipeline) FFT구조는 높은 성능을 요구하는 응용분야에 가장 많이 사용하는 구조이며, 다양한 형태의 파이프라인 FFT가 제안되었다. 그중 Radix-2² Single-path Delay Commutator(R2² SDC)는 비교적 간단한 구조와 제어가 용이하며 다른 형태 보다 적은 메모리를 요구한다. 이에 본 논문에서는(R2² SDC) 구조를 이용하여 256 Point FFT 프로세서를 설계하였다.

II. FFT 프로세서의 구조

2-1. 256 point FFT Processor

그림 1은 256 Point FFT Processor의 블록도이다.[3] 전체 블록은 총 8단의 버터플라이단이 파이프라인구조로 연결되어 있으며, 버터플라이단 사이에는 Twiddle Factor가 곱해지게 된다.

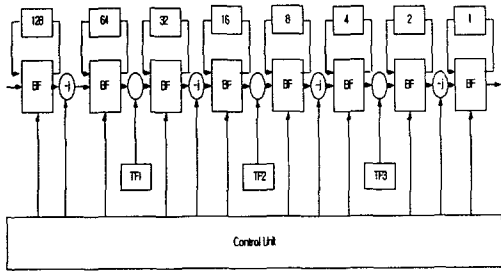


그림 1. pipeline FFT architecture for N = 256

첫 번째 버터플라이에는 128단의 Shift Register단이 연결되며 두 번째 단은 64단의 Shift Register단이 연결되며 각 단을 거칠 때마다 요구되어지는 Register량은 반으로 줄어든다. 버터플라이 단 사이마다 Twiddle factor와 $-j$ 값이 곱해지게 된다. $-j$ 의 승산은 실수데이터는 허수부의 입력으로, 허수데이터는 실수부의 입력으로 바뀌어 다음단 버터플라이의 입력으로 들어가게 되므로써 하드웨어를 절약할수 있다. 버터플라이와 승산기 Shift Register블록에 필요한 제어신호를 주는 Control Unit이 필요하다. 그림 2는 16 Point FFT의 신호 흐름도이다.[4]

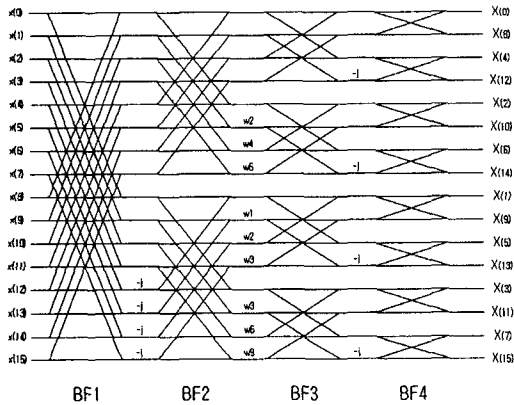


그림 2. Radix-2² FFT flow graph for N = 16

2-2. 버터플라이 구조

Radix-2² SDC버터플라이는 두가지 모드로 동작을 한다. 동작모드는 내부 Mux의 select신호에 의해 제어된다. select 신호가 0일 경우 입력단으로 들어오는 값은 Shift Register로 Clock신호와 동기되어 저장되어 있는 동시에 Register에 저장되어 있는 값을 그대로 다음 단으로 출력하게 된다. select신호가 1일 경우 입력

신호가 Register에 저장되어 있던 값과 버터플라이 연산을 수행하게 되는데, 이때 더해진 값은 다음단의 입력으로 전달이 되며 감산된 값은 다시 Register로 저장이 된다.[3][5] 그림 3에 버터플라이의 동작을 도시하였다.

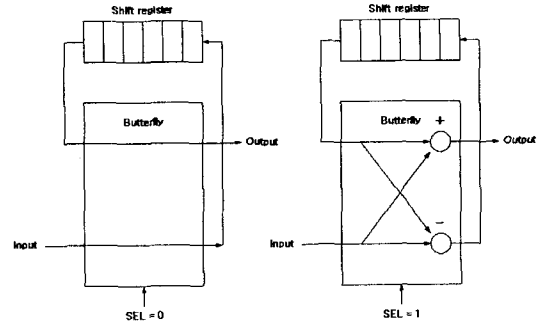


그림 3 버터플라이 동작

버터플라이는 가산기와 감산기 그리고 2개의 MUX로 구성된다. 버터플라이연산을 수행할 때마다 오버플로워가 발생할수 있는데 이를 방지하기 위해 버터플라이 출력값을 1/2 스케일링함으로써 오버플로워를 방지할수 있다. 그림 4는 버터플라이의 내부구조를 보여준다.

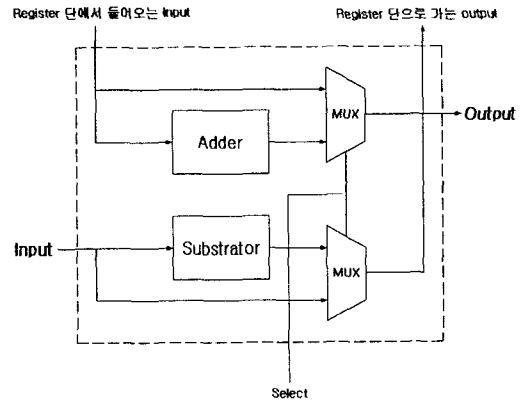


그림 4 버터플라이의 내부구조

2-3. Complex Multiplier

버터플라이 각 단사이에 Twiddle Factor가 곱해지게 된다. Twiddle Factor는 실수값이 아닌 복소수값이기 때문에 Complex Multiplier가 필요하다. (1)식에 따라 Complex연산을 수행한다.

$$Z = A B = (A_R + jA_I)(B_R + jB_I)$$

$$Z_R = \text{Re}[Z] = A_R B_R - A_I B_I$$

$$Z_I = \text{Im}[Z] = A_R B_I + A_I B_R \quad (1)$$

그림 5는 Complex Multiplier의 내부 블록도 이다. 4개의 입력신호를 받아서 각각 곱한 후 2개의 실수와 허수 출력을 내보내게 된다.

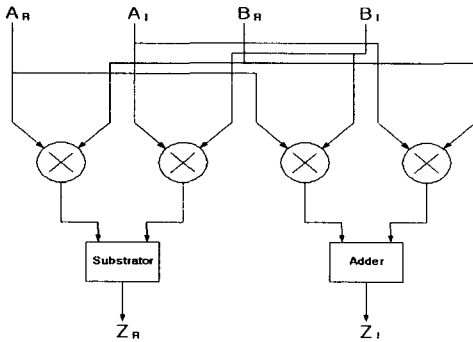


그림 5. Complex Multiplier

2-4. Twiddle Factor

Complex Multiplier로 들어가는 Twiddle Factor값은 ROM에 저장된다. 신호 흐름도에서 볼수 있듯이 계수값 들은 일정한 주기를 가지고 반복적으로 곱해지게 됨을 알수 있다. ROM에 Address신호를 주기 위하여 Counter가 필요하다. 그림 6은 Twiddle Factor의 블록도이다.

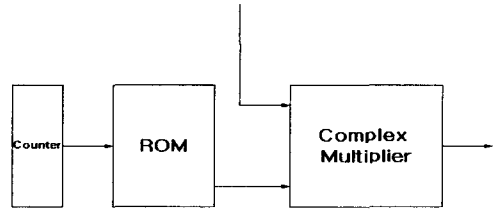


그림 6 Twiddle Factor 블록도

2-5. Control unit

Control unit 각각의 블록에 제어신호를 만들어준다. 각각의 블록들은 clock과 Control unit에서 나온 제어신호와 동기가 되어 동작한다. 표1은 각각의 블록에 필요한 Control 신호와 입,출력신호를 정리한 것이다.

clk	clock
rst	reset
bf1_sel	butterfly mode select
bf2_sel	butterfly mode select
bf3_sel	butterfly mode select
bf4_sel	butterfly mode select
bf5_sel	butterfly mode select
bf6_sel	butterfly mode select
bf7_sel	butterfly mode select
bf8_sel	butterfly mode select
j1_sel	-j term select
j2_sel	-j term select
j3_sel	-j term select
j4_sel	-j term select
real_in	실수 입력신호
im_in	허수 입력신호
real_out	실수 출력신호
im_out	허수 출력신호

표 1. Control 신호

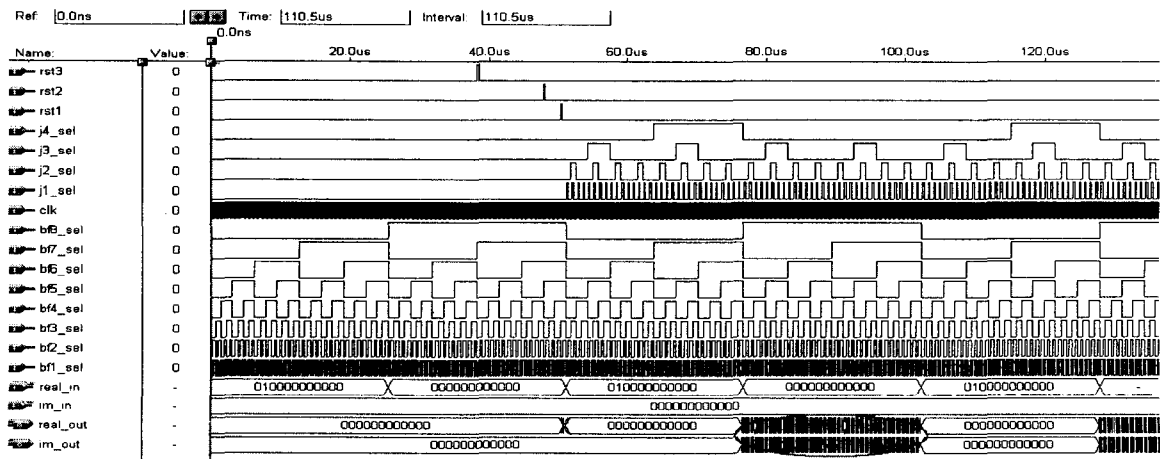


그림 7. FFT Processor의 전체시물레이션

III. Simulation

앞에서 살펴본 FFT Processor를 반도체 설계 교육센터 (IDEC)에서 지원한 Altera MAX+PLUS II를 이용하여 시뮬레이션 하였다. 입력단의 허수 파트는 전부 0값을 실수 파트중 128 clock(N/2)동안 1을 나머지 128 clock(N/2)동안 0을 입력신호로 주었다. 시뮬레이션한 결과 N클럭 후 처음 출력값이 나오기 시작해서 Bit-reverse한 순서로 출력값이 나온다. 출력단의 실수 파트는 처음 N/2 동안 0을 나머지 N/2동안은 1값을 갖는다. 허수파트는 처음 N/2동안 0값을 나머지 N/2동안은 Sync function의 형태를 유지함을 확인하였다. 그림 7은 전체시뮬레이션 결과이고, 그림8은 일부분을 확대한 그림이다.

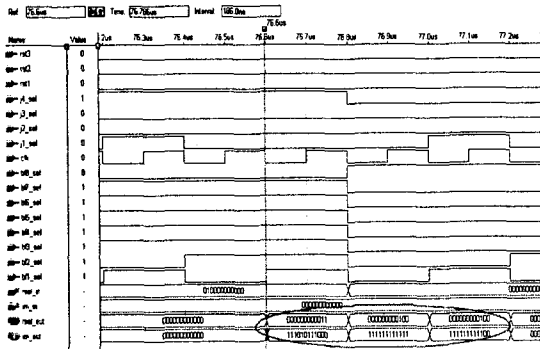


그림 8. FFT Processor의 전체시뮬레이션의 부분 확대도

IV. 결론

본 논문에서는 Radix-2² 구조를 이용한 Pipeline방식 256 point FFT Processor를 VHDL을 사용하여 설계하였다. 입력과 출력신호의 데이터 폭은 출력신호의 정확도를 위해 12bit을 사용하였다. Altera MAX+PLUS II로 simulation하여 동작을 검증하였다.

참고문헌

[1] ETRI, *A Study on Wireless Multimedia Communication Technology*, 1997.
 [2] 장시중 "OFDM을 위한 고속 FFT Processor 설계에 관한 연구," 전북대학교 대학원 정보통신공학과 1998.
 [3] M. Vergara, M. Strum, W. Eberle, and B. Gyselinx, "A 195KFFT/K (256-points)High Performance FFT/IFFT Processor for OFDM

Applications," Telecommunication Symposium, IEEE Internationl pp. 273-278, 1998.

[4] S. A. White "A Simple FFT Butterfly Arithmetic Unit," *IEEE Trans. Circuit and Systems*, vol. cas-28, no. 4, April 1981.

[5] E. H. Wold and A. M. Despain "Pipeline and Parallel-Pipeline FFT Processors for VLSI Implementations," *IEEE Trans. Comm*, vol. c-33, no. 5, pp 414-426, may 1984.