

2단계 수렴 블록 부동점 스케일링 기법을 이용한 8192점 파이프라인 FFT/IFFT 프로세서

준회원 이승기*, 양대성**, 정회원 신경욱*

A 8192-point pipelined FFT/IFFT processor using two-step convergent block floating-point scaling technique

Seung-Ky Lee*, Dae-Sung Yang** Associate Member, Kyung-Wook Shin* Regular Members

요 약

DMT 기반의 VDSL 모뎀, OFDM 방식의 DVB 모뎀 등 다중 반송파 변조 시스템에서 핵심 블록으로 사용되는 8192점 FFT/IFFT 프로세서를 설계하였다. 새로운 2단계 수렴 블록 부동점 (two-step convergent block floating-point; TS_CBFP) 스케일링 방법을 제안하여 설계에 적용하였으며, 이를 통해 FFT/IFFT 출력의 신호 대 양자화 잡음 비 (signal-to-quantization-noise ratio; SQNR)가 크게 향상되도록 하였다. 제안된 TS_CBFP 스케일링 방법은 별도의 버퍼 메모리를 사용하지 않아 기존의 방법에 비해 메모리를 약 80% 정도 감소시키며, 따라서 칩 면적과 전력소모를 크게 줄일 수 있다. 입력 10-비트, 내부 데이터와 회전인자 14-비트, 그리고 출력 16-비트로 설계된 8192점 FFT/IFFT 코어는 약 60-dB의 SQNR 성능을 갖는다. 0.25- μ m CMOS 셀 라이브러리로 합성한 결과, 약 76,300 게이트와 390K 비트의 RAM, 그리고 39K 비트의 ROM으로 구현되었다. 시뮬레이션 결과, 50-MHz@2.5-V 로 안전하게 동작할 것으로 평가되었으며, 8192점 FFT/IFFT 연산에 약 164- μ s가 소요될 것으로 예상된다. 설계된 코어는 Xilinx FPGA에 구현하여 정상 동작함을 확인하였다.

ABSTRACT

An 8192-point pipelined FFT/IFFT processor core is designed, which can be used in multi-carrier modulation systems such as DMT-based VDSL modem and OFDM-based DVB system. In order to improve the signal-to-quantization-noise ratio (SQNR) of FFT/IFFT results, two-step convergent block floating-point (TS_CBFP) scaling is employed. Since the proposed TS_CBFP scaling does not require additional buffer memory, it reduces memory as much as about 80% when compared with conventional CBFP methods, resulting in area- and power-efficient implementation. The SQNR of about 60-dB is achieved with 10-bit input, 14-bit internal data and twiddle factors, and 16-bit output. The core synthesized using 0.25- μ m CMOS library has about 76,300 gates, 390K bits RAM, and twiddle factor ROM of 39K bits. Simulation results show that it can safely operate up to 50-MHz clock frequency at 2.5-V supply, resulting that a 8192-point FFT/IFFT can be computed every 164- μ s. It was verified by Xilinx FPGA implementation.

1. 서론

고속 푸리에 변환 (Fast Fourier Transform; FFT)

은 시간영역의 데이터를 주파수영역으로 변환하는 이산 푸리에 변환 (Discrete Fourier Transform; DFT)의 고속 연산 알고리즘이며, 디지털 신호처리

* 금오공과대학교 전자공학부 VLSI 설계 연구실 (kwshin@kumoh.ac.kr), ** (주)KEC 종합연구소 IC Design Center

논문번호 : 020225-0509, 접수일자 : 2001년 5월 9일

※ 본 연구는 시스템집적반도체기반기술개발사업의 선행핵심IP개발과제 지원과 반도체설계교육센터(IDEC)의 CAD Tool 지원으로 수행되었습니다.

및 고속 유·무선 디지털 통신 시스템의 설계에 폭 넓게 사용된다^[1]. 최근, 디지털 통신 및 무선 전송기술의 발달로 인하여 FFT와 Inverse FFT (IFFT)의 응용분야가 급속히 확대되고 있으며, 특히 직교 주파수 분할 다중화 (Orthogonal Frequency Division Multiplexing; OFDM) 방식을 사용하는 IEEE 802.11a 고속 무선 LAN, 유럽형 디지털 오디오 방송 (Digital Audio Broadcasting; DAB) 및 디지털 비디오 방송 (Digital Video Broadcasting; DVB) 시스템, DMT (Discrete Multi-Tone) 방식의 디지털 가입자 회선 (Digital Subscriber Line; xDSL) 모델 등에서 핵심 기능블록으로 사용된다^[2-4].

고속 디지털 통신 시스템에 사용되는 FFT/IFFT 프로세서는 대량의 데이터에 대한 실시간 연산이 필수적이다. 예를 들어, 유럽형 DVB-T 규격에서는 8192점 FFT를 896- μ s 이내에 처리해야 하며, DMT 기반의 VDSL (Very high-speed DSL) 모델에서는 8192점 FFT/IFFT를 250- μ s 이내에 처리해야 한다. 따라서 고 해상도 FFT/IFFT의 실시간 처리를 위해서는 전용 ASIC (Application-Specific Integrated Circuit) 구현이 필수적이다. 특히, 시스템 수준의 집적회로 (System-On-a-Chip; SOC) 설계가 보편화 되는 추세에 따라, 고속/저전력 특성을 갖는 FFT/IFFT 코어의 IP (Intellectual Property) 개발이 필요하며, 최근에는 IComm^[5], QuickLogic^[6], Xilinx^[7] 등에서 1,024점 FFT/IFFT 코어를 상용화하고 있다.

높은 성능이 요구되는 응용분야에서는 FFT의 각 연산단계에 하나 또는 그 이상의 나비 연산기를 할당하고 각 연산단계를 파이프라인 방식으로 동작시키는 파이프라인 FFT 구조가 사용된다. 파이프라인 FFT 구조는 radix와 지연블록의 구현 방식에 따라 radix-2 multipath delay commutator (R2MDC) 구조, radix-2 single-path delay feedback (R2SDF) 구조, radix-4 multipath delay commutator (R4MDC) 구조, radix-4 single-path delay feedback (R4SDF) 구조, radix-4 single-path delay commutator (R4SDC) 구조, radix-2² single-path delay commutator (R2²SDC) 등 다양한 형태가 제안되고 있다^[9]. 본 논문에서는 나비 연산기의 이용 효율이 가장 좋으며, CBFP 스케일링의 구현이 용이한 R4SDC 구조를 채택하였다.

FFT/IFFT 연산은 나비연산과 복소수 곱셈으로 구성되는 연산단계를 거치면서 내부 중간결과 값의 비트 수가 증가하게 되므로, 고정점 연산을 사용하는 FFT/IFFT 프로세서는 각 연산단계의 중간결과

값을 적절한 내부 비트 수로 제한해야 한다. 각 연산단계의 출력을 지정된 비트 수로 잘라내는 단순 절사 방법은 부가 하드웨어가 필요치 않다는 장점이 있으나, FFT/IFFT 출력의 정확도가 떨어져 신호 대 양자화 잡음 비 (Signal-to-Quantization-Noise Ratio; 이하 SQNR로 약칭함)가 나빠지는 단점을 갖는다.

이를 개선하기 위해, 고정점 (fixed-point) 표현과 부동점 (floating-point) 표현을 혼합한 블록 부동점 (Block Floating Point; 이하 BFP로 약칭함) 스케일링 기법^[11]이 제안되었다. BFP 스케일링은 각 연산단계의 중간결과 데이터 전체를 버퍼 메모리에 저장한 후, 이들 중에서 가장 큰 값을 기준으로 데이터를 스케일링하여 내부 비트 수를 제한하는 방법이다. 그러나, 이 방법은 이전 연산단계의 출력 데이터 전체가 메모리에 저장되어야 하므로 파이프라인 방식의 FFT/IFFT 프로세서에는 적용이 불가능하며, 또한 FFT 길이에 비례하는 큰 버퍼 메모리를 필요로 하여 칩 면적과 전력소모가 커지는 단점을 갖는다.

파이프라인 FFT 프로세서의 SQNR 향상을 위해 수렴 블록 부동점 (Convergent Block Floating Point; 이하 CBFP로 약칭함) 스케일링 기법^[12]이 제안되었다. CBFP 스케일링은 각 연산단계의 데이터 블록들이 독립적으로 연산되는 성질을 이용한 것으로, 각 데이터 블록의 연산이 종료된 직후 이 중에서 가장 큰 값을 기준으로 데이터를 스케일링하여 내부 비트 수를 제한하는 방법이다. 따라서, 각 연산단계에서 블록 크기에 비례하는 버퍼 메모리만을 사용하므로 BFP 방법에 비하여 스케일링을 위한 메모리를 크게 줄일 수 있다. 그러나, 8192점 FFT와 같이 길이가 큰 경우에는 여전히 큰 버퍼 메모리를 필요로 하므로, 칩 면적과 전력소모 측면에서 실용성이 매우 떨어진다. 또한, 각 연산단계마다 블록 스케일링 지수 값을 찾는 시간만큼 출력이 지연되므로 전체 FFT 프로세서의 출력 레이턴시가 길어지는 단점을 갖는다.

본 논문에서는 CBFP 스케일링을 갖는 파이프라인 FFT 프로세서의 효율적인 구현을 위해, 별도의 버퍼 메모리를 사용하지 않는 새로운 2단계 CBFP 스케일링 방법을 제안하고, 이를 적용한 8192점 파이프라인 FFT/IFFT 프로세서를 설계하였다.

본 논문의 II장에서는 CBFP 스케일링의 기본 개념과 함께 일반적인 구현방법을 소개하고, 종래 방법의 단점을 개선하기 위해 본 논문에서 제안하는

2단계 CBFP 스케일링 방법에 대해 설명한다. III장에서는 제안된 2단계 CBFP 스케일링 방법을 적용한 8192점 FFT/IFFT 코어의 아키텍처 및 회로설계를 기술하고, IV장에서는 기능레벨 검증과 FPGA 구현을 통한 검증결과를 기술하고, V장에서 결론을 제시하였다.

II. 2단계 CBFP 스케일링 기법

본 장에서는 고정점 연산을 갖는 파이프라인 FFT/IFFT 프로세서의 연산 정확도를 향상시키기 위한 CBFP 스케일링의 기본 개념과 기존의 CBFP 스케일링 방법이 갖는 단점에 대해 언급하고, 이를 개선하기 위해 본 논문에서 제안되는 2단계 CBFP 스케일링 방법에 대해 설명한다.

2.1. 종래의 CBFP 스케일링 방식

그림 1은 일반적인 CBFP 스케일링 방법의 메커니즘을 나타내는 개념도이다. 설명의 편의를 위하여, 데이터 블록의 크기는 $B=128$ 이고 연산단계의 중간 결과 값을 $m=25$ 비트에서 $d=12$ 비트로 스케일링하는 경우를 예로 들었다. 그러나, 블록의 크기 (B)나 중간결과 데이터의 비트 수 (m) 및 내부 비트 수 (d)에 제한 받지 않고 일반적으로 적용이 가능하다.

나비 연산과 복소수 승산을 거쳐 CBFP 처리부에 입력되는 25비트 데이터의 실수부와 허수부에서 부호비트 (즉, 데이터의 최상위 비트)와 동일하면서 연속된 '0' 또는 '1'의 개수 (Number of Leading Bit; 이하 NLB로 약칭함)를 찾는다. 이를 블록 스케일링 지수 (Block Scaling Index; 이하 BSI로 약칭함) 레지스터 값과 비교하여 작은 값을 BSI 레지스터에 갱신하고, 25비트 데이터는 순차적으로 버퍼 메모리에 임시 저장된다. 블록내의 모든 데이터에 대해 상기 과정을 반복하면 BSI 레지스터에는 가장 작은 NLB 값이 저장되고, 이 값을 이용하여 버퍼 메모리에 저장된 25비트의 데이터를 순차적으로 꺼내어 12비트로 스케일링하여 다음 연산단계로 넘겨준다. 이와 같은 CBFP 스케일링 과정이 각 연산단계의 모든 데이터 블록들에 대해 반복 수행되며, 각 연산단계에서 얻어진 BSI 값은 연산단계마다 누적 가산된다. 마지막 연산단계에는 누적 가산된 CBFP 지수를 이용하여 최종 FFT/IFFT 출력에 대한 CBFP 디코딩 과정이 수행된다.

그림 1의 예를 구체적으로 설명하면 다음과 같다. 첫 번째 데이터의 경우, 실수부와 허수부의 NLB

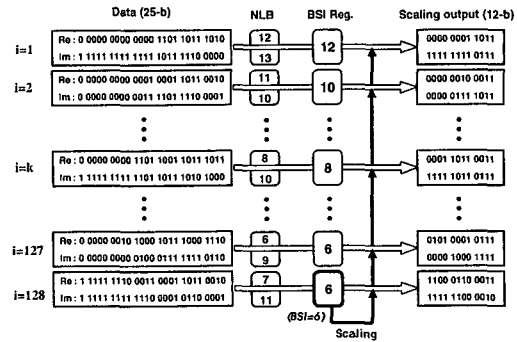


그림 1. 종래의 CBFP 스케일링 방법

값은 각각 12와 13이므로 BSI 레지스터에 12가 저장되며, 두 번째 데이터의 실수부와 허수부의 NLB 값은 각각 11과 10이므로, 이들을 BSI 레지스터에 저장되어 있는 값 12와 비교하여 가장 작은 값인 10이 BSI 레지스터에 저장된다. 동일한 과정을 나머지 데이터에 대해 반복하면, BSI 레지스터의 최종 값은 6이 된다. 따라서, 25비트의 중간결과 데이터를 12비트로 제한하기 위하여, 버퍼 메모리에 저장된 각 데이터의 최상위 비트 쪽 6비트와 최하위 비트 쪽 7비트를 버리고 중간부분의 나머지 12비트만을 취하여 다음 연산단계로 넘겨주게 된다.

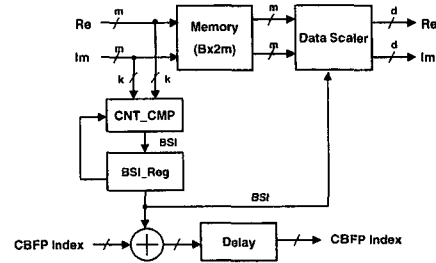


그림 2. 종래의 CBFP 처리블록

그림 2는 그림 1의 메커니즘을 기반으로 하는 종래의 CBFP 스케일링 처리부의 구성도이다. 계수/비교기 (CNT_CMP)는 CBFP 처리부에 입력되는 m -비트 데이터의 NLB 값을 구하고, 이 값을 BSI 레지스터에 저장된 값과 비교하여 작은 값을 BSI 레지스터에 갱신한다. 데이터 블록내의 모든 데이터에 대한 계수/비교 과정이 완료되어야만 해당 데이터 블록의 스케일링 지수가 찾아지므로, 데이터 블록내의 모든 m -비트 데이터는 버퍼 메모리에 일시 저장되어야 하며, 따라서 버퍼 메모리는 $(B \times 2m)$ -비트의 용량을 필요로 한다. 여기서 B 는 데이터 블록의 크기를 나타내며, FFT 연산단계마다 $1/r$ 씩 감소한다

(단, r 은 FFT 알고리즘의 radix 값을 나타냄). 데이터 스케일러는 해당 데이터 블록의 BSI 값이 결정된 후, 버퍼 메모리에 저장된 데이터를 순차적으로 꺼내어 d -비트 (단, $m > d$ 임)로 스케일링하여 다음 연산단계로 보낸다. 한편, 현재 연산단계의 BSI 값은 이전 연산단계에서 입력되는 CBFP 지수와 가산된 후, 데이터와 CBFP 지수간의 동기를 맞추기 위해 지연기를 거쳐 다음 연산단계의 CBFP 스케일링 처리부로 보내진다.

이와 같은 종래의 CBFP 스케일링 방법은 각 연산단계의 CBFP 처리부에서 $(B \times 2^m)$ -비트의 버퍼 메모리가 필요하며, 또한 각 연산단계마다 BSI 값을 구하는 시간만큼 출력이 지연되므로 FFT 프로세서의 출력 레이턴시가 길어진다. 따라서, 8192점 FFT 경우에는 매우 큰 버퍼 메모리를 필요로 하므로 칩면적과 전력소모 측면에서 실용성이 떨어진다.

2.2. 2단계 C3FP 스케일링 방식

그림 3은 본 논문에서 제안하는 2단계 CBFP (Two-Step CBFP; 이하 TS_CBFP로 약칭함) 스케일링 메커니즘을 나타내는 개념도이며, 그림 1과 동일한 블록 크기와 데이터를 사용하였다.

CBFP 처리부에 입력되는 25비트 데이터의 실수부와 허수부에서 NLB를 찾아 이를 예비 스케일링 지수 (Pre-Scaling Index; 이하 PSI라 약칭함) 레지스터 값과 비교하여 작은 값을 PSI 레지스터에 저장하고, 이를 이용하여 해당 데이터의 실수부와 허수부를 12비트로 예비 스케일링한 후 PSI 값과 함께 지연 변환기에 저장한다. 블록내의 모든 데이터에 대해 이와 같은 예비 스케일링 과정이 완료되면 최종 BSI 값이 결정되고, 지연 변환기 메모리에 저장된 데이터와 해당 PSI 값을 순차적으로 꺼내서 BSI 값과의 차이만큼 스케일링 보정을 행한다. 이와 같은 CBFP 스케일링 과정이 각 연산단계의 모든 데이터 블록들에 대해 반복 수행되며, 각 연산단계에서 얻어진 BSI 값은 연산단계마다 누적 가산되어 최종 출력에 대한 CBFP 디코딩 과정에서 사용된다.

그림 3의 예를 구체적으로 설명하면 다음과 같다. 첫 번째 데이터의 경우, 실수부와 허수부의 NLB 값은 각각 12와 13이므로 PSI 레지스터에 12가 저장되며, 25비트의 중간결과 데이터에서 최상위 비트 쪽 12비트와 최하위 비트 쪽 1비트를 버리고 중간부분의 나머지 12비트만을 취하여 PSI 값 12와 함께 지연 변환기에 저장한다. 두 번째 데이터의 경우, 실수부와 허수부의 NLB 값은 각각 11과 10이

므로, PSI 레지스터 값은 10으로 갱신되며, 25비트의 데이터에서 최상위 비트 쪽 10비트와 최하위 비트 쪽 3비트를 버리고 중간부분의 나머지 12비트만을 취하여 PSI 값 10과 함께 지연 변환기에 저장한다. 한편, 128번째 데이터의 경우, 실수부와 허수부의 NLB 값은 각각 7과 11이나, 127번째 데이터까지 구해진 PSI 값이 6이므로, PSI 레지스터는 6을 유지하며, 25비트의 데이터에서 최상위 비트 쪽 6비트와 최하위 비트 쪽 7비트를 버리고 중간부분의 나머지 12비트만을 취하여 PSI 값 6과 함께 지연 변환기에 저장한다. 따라서, BSI 값은 PSI 레지스터의 최종 값인 6으로 결정되며, 이는 예비 스케일링된 데이터에 대한 스케일링 보정에 사용된다.

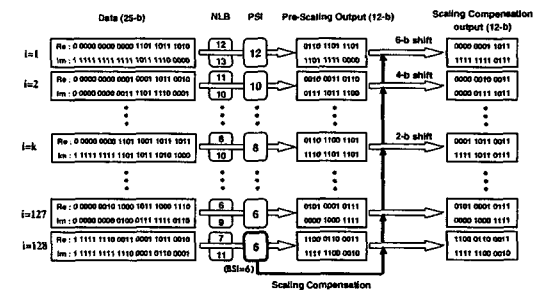


그림 3. 제안된 TS_CBFP 스케일링 방법

스케일링 보정은 PSI 값과 BSI 값의 차에 해당하는 비트 수만큼 오른쪽 쉬프트를 통해 이루어지며, 이는 PSI 값이 BSI 값 보다 항상 같거나 크기 때문이다. 한편, 스케일링 보정을 위한 오른쪽 쉬프트는 각 데이터의 최상위 비트를 확장한 만큼 최하위 비트 쪽을 버리는 동작을 의미한다. 첫 번째 데이터의 경우 $PSI=12$ 로 예비 스케일링되었으므로 $BSI=6$ 과의 차이인 6비트만큼의 스케일링 보정이 필요하며, 지연 변환기에 저장된 12비트의 실수부 데이터 "0110 1101 1101"와 허수부 데이터 "1101 1111 0000"은 각각 6비트 오른쪽 쉬프트를 통해 "0000 0001 1011"와 "1111 1111 0111"로 스케일링 보정된다. 한편, 127번째와 128번째 데이터의 경우에는 PSI 값과 BSI 값이 같으므로 스케일링 보정이 필요 없다. 그림 3의 제안된 방법으로 스케일링 보정된 최종 데이터는 그림 1의 결과와 동일함을 알 수 있다.

2.1절에서 설명된 종래의 방법과 비교할 때, 본 논문에서 제안되는 TS_CBFP 방법은 스케일링 처리부에 별도의 버퍼 메모리를 사용하지 않는다는 장점을 갖는다. 즉, m -비트의 데이터를 d -비트 (단,

$m>d$)로 예비 스케일링하고, 이 데이터를 별도의 버퍼 메모리 대신에 지연 변환기에 저장함으로써, $(B \times 2m)$ -비트의 버퍼 메모리가 필요한 종래의 방법에 비해 칩 면적과 전력소모를 크게 감소시키는 효과를 얻을 수 있다.

III. 회로 설계

3.1. 아키텍처 개요

제안된 TS_CBFP 스케일링 방법을 적용한 8192점 파이프라인 FFT/IFFT 프로세서 (이하 PFFT8k 코어로 약칭함)는 그림 4와 같은 구조를 가지며, R4SDC 구조를 기반으로 하여 6개의 radix-4 연산단계와 1개의 radix-2 연산단계, 그리고 CBFP 디코더로 구성된다. 연산단계-1은 스케일링을 갖지 않는 일반적인 radix-4 연산단계이며, 연산단계-2~연산단계-6은 TS_CBFP 스케일링 기능을 갖는 radix-4 연산단계이고, 연산단계-7은 radix-2 연산단계이다. 입력단과 출력단의 선택기는 모드선택 신호에 의해 FFT 연산과 IFFT 연산을 선택적으로 수행하도록 하며, CBFP 디코더는 연산단계-6까지 누적 가산된 CBFP 지수와 외부에서 인가되는 출력 이득조정 신호(OSF)에 의해 최종 FFT/IFFT 출력을 역 스케일링하는 기능을 수행한다. 한편, 내부 비트 수는 고정점 시뮬레이션을 통해 결정하였으며, 입력 데이터는 10비트, 회전인자 값과 CBFP 스케일링 후의 내부 데이터는 14비트, 그리고 최종 출력은 16비트로 결정하였다.

연산단계-2~연산단계-6을 구성하는 TS_CBFP 스케일링 기능을 갖는 radix-4 연산단계는 그림 5와 같이 CBFP 예비 스케일러, 스케일링 보정 지연 변환기(SC-DC), radix-4 나비 연산기, 복소수 곱셈기

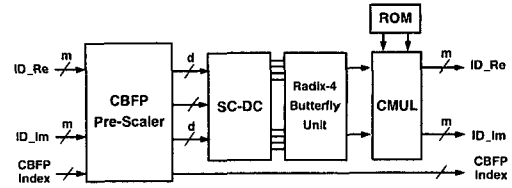


그림 5. 제안된 TS_CBFP 스케일링을 갖는 radix-4 연산단계

와 회전인자 롬 등으로 구성된다. CBFP 예비 스케일러와 스케일링 보정 지연 변환기는 이전 연산단계의 복소수 승산기에서 출력되는 28비트 중간결과 데이터를 각 데이터 블록내의 가장 큰 데이터를 기준으로 14비트로 스케일링함으로써 연산오차를 최소화하는 기능을 수행한다.

설계된 PFFT8k 코어의 동작 타이밍 도는 그림 6과 같다. 초기에 리셋신호(RST)가 인가된 후, 코어의 초기상태 설정을 위한 configuration load enable(CLE) 신호가 인가되면 FFT/IFFT를 결정하는 IFFT 신호(IFTT=0이면 FFT, IFFT=1이면 IFFT)와 출력 이득조정을 위한 3비트의 output scaling factor(OSF) 신호가 로드된다. 데이터 입력이 시작되면 첫 번째 데이터가 입력되었음을 알리는 신호(DIS)가 외부로 출력되며, 그로부터 8,904 클럭 주기의 레이턴시를 지나 FFT/IFFT 결과가 출력되기 시작하며,

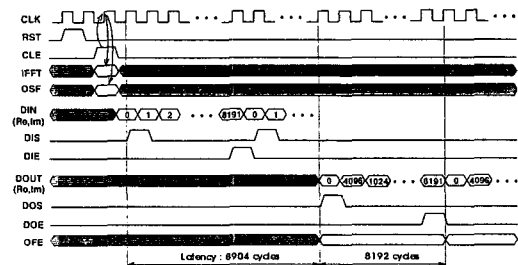


그림 6. PFFT8k 코어의 동작 타이밍

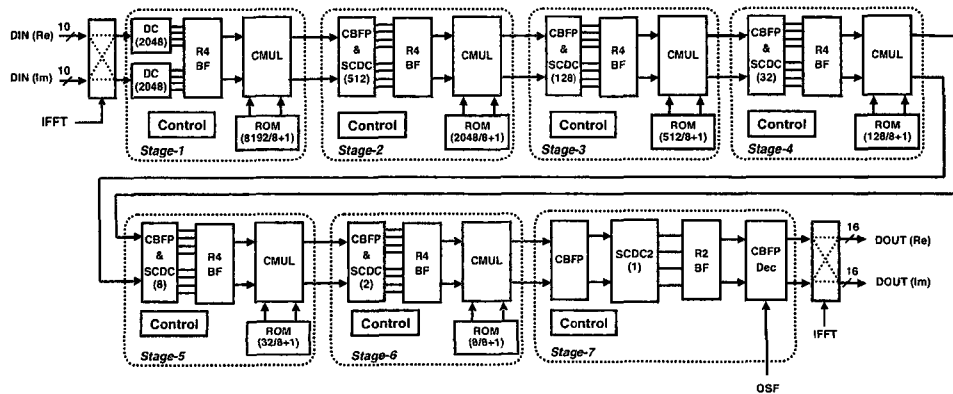


그림 4. 제안된 TS_CBFP 스케일링을 적용한 8192점 FFT/IFFT 프로세서의 구조

데이터 출력이 시작되었음을 알리는 신호 (DOS)가 외부로 출력된다. 한편, OSF 신호를 이용한 출력 이득조정 과정에서 오버플로가 발생하는 경우, 오버플로 에러 신호인 OFE가 'HIGH'로 출력된다.

3.2. CBFP 예비스케일링 처리부

그림 7은 CBFP 예비 스케일링 처리부의 상세도이며, 그림 3의 메커니즘을 기반으로 한다. 복소수 곱셈기로부터 입력되는 m -비트 데이터는 레지스터에 저장됨과 동시에 계수/비교기 (CNT_CMP)에서 NLB 값을 구하여 각 데이터의 PSI 값을 결정한다. 레지스터에 임시 저장된 m -비트 데이터는 예비 스케일링에 의해 해당 PSI 값만큼 예비 스케일링된 d -비트 (단, $m > d$)로 변환되어 스케일링 보정을 갖는 지연 변환기로 입력된다. 블록내의 모든 데이터에 대한 처리가 완료되면, PSI 레지스터의 최종 값이 해당 데이터 블록의 BSI 값으로 결정되어 BSI 레지스터에 저장된 후, 스케일링 보정 지연 변환기로 입력된다. 현재 연산단계의 BSI 값은 이전 연산 단계에서 입력되는 CBFP 지수와 가산된 후, 다음 연산단계의 CBFP 예비 스케일링 처리부로 보내진다. 한편, CBFP z 수의 누적 가산과정에서 발생하는 오버플로를 감지하기 위하여 레지스터의 최상위 비트와 가산기 출력의 최상위 비트를 논리 OR 연산하여 레지스터의 최상위 비트에 저장하며, 이 오버플로 신호는 CBFP 디코더에서 사용된다.

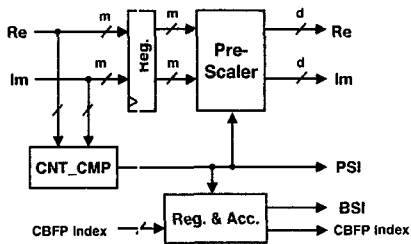


그림 7. CBFP 예비 스케일링

3.3. 스케일링 보정을 갖는 지연 변환기

지연 변환기는 이전 연산단계에서 순차적으로 입력되는 데이터가 radix-4 나비연산에 사용될 수 있도록 4개씩 짝을 만드는 데이터 섞음 (shuffling) 기능을 수행한다. 지연 변환기에 저장된 d -비트의 데이터는 각각의 PSI 값으로 예비 스케일링된 데이터이므로, CBFP 예비 스케일링 처리부에서 결정된 BSI 값과 각 데이터의 PSI 값의 차이만큼 스케일링 보정이 필요하다. 본 논문에서는 스케일링 보정과 데이터 섞음 기능을 효율적으로 구현하기 위해 그

림 8에서 보는 바와 같이 7개의 ($B_k \times 2d$)-비트 버퍼 메모리 블록, 4개의 ($B_k \times 4$)-비트 PSI 메모리 및 스케일링 보정기 블록, 그리고 3개의 선택기 (MUX)로 구현하였다. 여기서, B_k 는 k -번째 연산단계의 블록 크기를 나타낸다. 스케일링 보정은 PSI 값과 BSI 값의 차이만큼 오른쪽 쉬프트를 통해 이루어지며, 오른쪽 쉬프트 연산은 각 데이터의 최상위 비트를 확장한 만큼 최하위 비트 쪽을 버리는 동작을 의미한다. 출력단의 MUX는 radix-4 나비연산 회로의 동작에 맞도록 메모리 블록의 출력을 선택하는 기능을 수행한다.

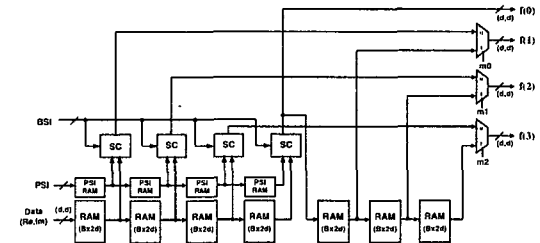


그림 8. 스케일링 보정을 갖는 지연 변환기

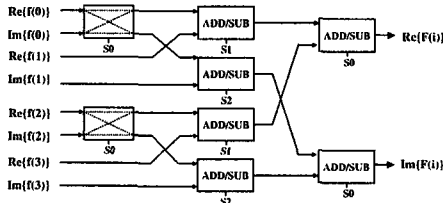
3.4. Radix-4 나비 연산기와 복소수 승산기

Radix-4 나비 연산기는 그림 9-(a)과 같이 6개의 가/감산기와 2개의 입력 스위치를 사용하여 설계하였다. 스위치는 가/감산기로 입력되는 데이터의 실수부와 허수부의 경로를 결정하기 위해 사용되며, 그림 9-(b)는 제어신호에 따른 스위치와 가/감산기의 동작을 보인 것이다. 가/감산기는 캐리선택 가산기 (carry-select adder)를 사용하여 설계하였다.

복소수 승산기는 나비 연산기를 거친 데이터와 격자계수의 곱셈을 연산하며, 연산시간, 칩 면적, 전력소모 등의 측면에서 전체 FFT 프로세서의 성능에 큰 영향을 미치는 부분이므로 설계의 최적화가 요구된다. 본 논문의 8192점 FFT/IFFT 프로세서는 연산단계-7을 제외한 각 연산단계 마다 1개씩 총 6개의 복소수 승산기가 사용되며, 나비 연산기를 거친 14비트 데이터와 14비트의 격자계수를 곱셈하여 28비트의 결과를 출력한다. 본 논문에서는 디지털 통신 및 신호처리에 적합하도록 개발된 복소수 승산기 IP^[10]를 사용하였다. 사용된 복소수 승산기는 실수 승산기를 이용하는 고전적인 방법 대신에 RB (Redundant Binary) 수치계와 Booth 알고리즘을 혼합한 방식과 개선된 Booth 인코딩/디코딩 방식을 적용함으로써 기존의 방식에 비해 고속/고집적/저전력의 특성을 갖는다. 14비트×14비트 복소수 승산기

는 6,530개의 게이트로 구현되었다.

복소수 곱셈에 사용되는 격자계수는 cosine 파형과 sine 파형의 한 주기를 ($B_k \times 4$)개로 샘플링 하여 얻을 수 있으나, 8192점 FFT/IFFT와 같이 데이터 길이가 큰 경우에는 격자계수를 저장하기 위한 ROM의 크기가 매우 커지게 되어 칩 면적과 전력 소모 측면에서 바람직하지 않다. 본 논문에서는 sine 파형과 cosine 파형의 1/8 주기만을 ROM에 저장한 후, ROM의 읽기 주소를 적절히 제어하여 각 연산단계에서 필요한 격자계수가 생성되도록 하였으며, 이를 통해 격자계수 ROM의 크기를 약 1/8로 감소시켰다.



(a) 내부 구조

Time	S0	S1	S2
1		ADD	ADD
2		ADD	SUB
3		SUB	SUB
4		SUB	ADD

(b) 제어신호와 동작

그림 9. Radix-4 나비 연산기

3.5. CBFP 디코더

각 연산단계에서 사용된 CBFP 스케일링은 m -비트의 중간결과 데이터를 d -비트 (단, $m > d$)로 제한하기 위해 블록내의 가장 큰 데이터를 기준으로 정렬 증가 (scaling-up) 시키는 과정이다. 따라서, 마지막 연산단계를 거친 데이터는 최종 CBFP 지수 값만큼 정렬 감소 (scaling-down)시켜 출력해야 한다. 이와 같은 CBFP 디코딩 과정은 CBFP 지수를 이용한 오른쪽 쉬프트로 처리된다. 한편, CBFP 디코더에서 출력의 최하위 비트쪽을 잘라내면 최종 출력의 정확성이 떨어져 SQNR이 감소된다. 이를 방지하기 위해, 외부로부터 인가되는 출력 이득조정 신호 (OSF)에 의해 출력의 정렬 감소 크기를 조정함으로써 최종 출력의 SQNR이 향상되도록 하였다.

출력 이득조정을 갖는 CBFP 디코더는 그림 10과

같이 설계되었으며, 그림 4에서 보는 바와 같이 최종 출력단에 위치한다. CBFP 지수와 이득조정 신호의 차를 구한 후, 쉬프트에서 해당 비트만큼 오른쪽 쉬프트시켜 최종 데이터를 출력한다. 한편, OSF 값이 CBFP 지수 값 보다 큰 경우에는 오버플로 감지회로에 의해 오버플로 에러신호 (OFE)가 발생되며, 이때에는 '00...00'이 출력되도록 설계하였다.

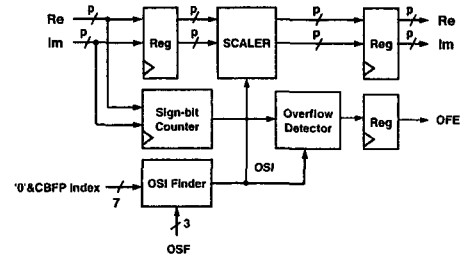


그림 10. 출력 이득 조정을 갖는 CBFP 디코더

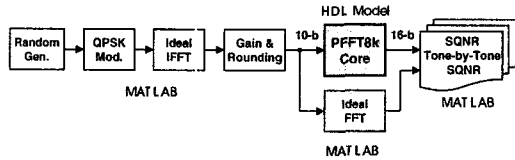
IV. 설계 검증 및 회로 합성

VHDL로 모델링된 PFFT8k 코어는 Synopsys와 ModelSim을 이용하여 논리합성과 검증을 하였으며, 그림 11-(a)는 논리검증과 성능평가 과정을 보인 것이다. 2진 랜덤 신호를 생성하여 QPSK 변조한 후, 부동점 연산을 갖는 ideal IFFT와 이득 조정을 거쳐 10비트로 양자화된 데이터를 시뮬레이션 입력으로 사용하였으며, 그림 11-(b)는 시뮬레이션에 사용된 입력 데이터의 실수부와 허수부를 보인 것이다. 그림 11-(c)는 FFT 출력의 연산오차를 보인 것이며, 85-dB의 신호전력에 대해 양자화 잡음 전력은 약 25-dB로 나타나나 약 60-dB의 SQNR을 갖는 것으로 분석되었다. 그림 11-(d)는 FFT 출력의 tone-by-tone SQNR을 분석한 결과이며, 전체 주파수 영역에서 우수한 SQNR 특성을 나타냈다.

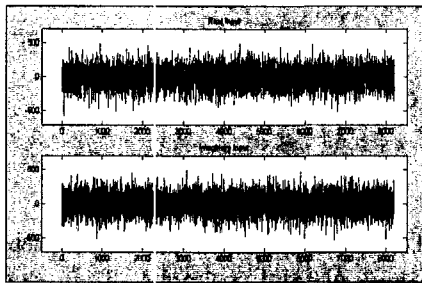
그림 12는 설계된 PFFT8k 코어의 출력 이득조정 비트 수에 따른 SQNR 특성을 분석한 결과이다. CBFP 스케일링과 함께 FFT 출력을 5비트 scale-up 시킨 경우 약 60-dB의 SQNR이 얻어졌으며, 이는 CBFP 스케일링을 적용하지 않는 경우의 37.7-dB에 비해 약 22-dB의 SQNR이 향상되었다. 그림 11-(d)와 그림 12에서 보는 바와 같이, 설계된 PFFT8k 코어는 우수한 SQNR 및 tone-by-tone SQNR 특성을 나타내어 다수 반송파 변조 시스템의 요구 조건을 충분히 만족하는 것으로 평가되었다.

검증이 완료된 HDL 모델은 0.25- μ m CMOS 셀 라이브러리를 이용하여 합성한 결과는 표 1과 같으

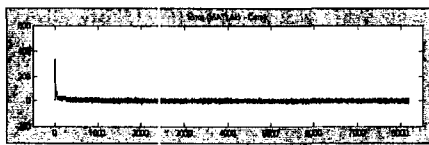
며, 76,300 게이트와 390K 비트의 RAM, 그리고 39K 비트의 ROM으로 구현되었다. 본 논문에서 제안된 TS_CBFP 스케일링 방법은 PSI 값 저장을 위한 30K 비트의 메모리만 사용되었으며, 153K 비트의 버퍼 메모리를 별도로 사용하는 기존의 방법에 비해 약 80%의 메모리를 줄일 수 있다.



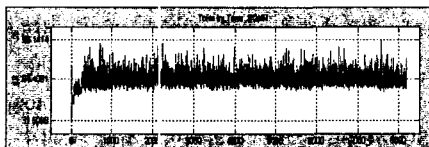
(a) simulation procedure



(b) input data



(c) quantization error of FFT output



(d) Tone-by-Tone SQNR

그림 11. 설계된 PFFT8k 코어의 검증 결과

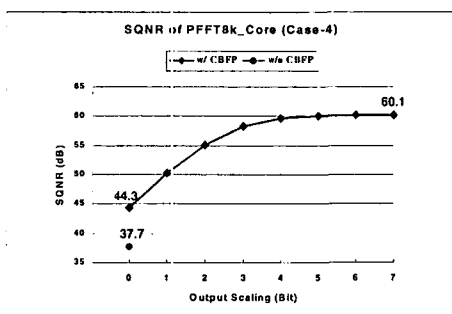


그림 12. 설계된 PFFT8k 코어의 SQNR 특성

표 1. 설계된 PFFT8k 코어의 합성 결과

Logic (gate)	R4BF	7,836	10.3%
	CMUL	37,615	49.3%
	Control & Reg.	13,850	18.2%
	CBFP & Reg.	16,925	22.2%
	Total	76,226	100%
RAM (bit)	DC	360,000	92.3%
	CBFP	29,920	7.7%
	Total	389,920	100%

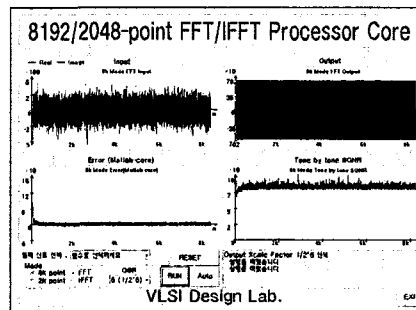
Twiddle Factor ROM (bit)	38,556
--------------------------	--------

시뮬레이션 결과, 최대 지연시간은 약 8-ns로 나타났다. 레이아웃 후의 배선에 의한 지연을 고려하더라도 2.5-V 전원전압에서 50-MHz로 안전하게 동작 가능할 것으로 평가된다. 따라서, 8192점 FFT연산에 164- μ s가 소요되어 DMT 기반의 VDSL 모델과 유럽형 DVB 시스템의 사양을 만족하는 것으로 평가되었다.

설계된 PFFT8k 코어를 Xilinx FPGA 디바이스 (XCV2000e-6)에 구현한 후, PC와 PCI 인터페이스를 통해 동작을 검증하였으며, 그림 13은 검증 시스템과 그 실행화면을 보인 것이다. FPGA 구현 결과, XCV2000e-6 디바이스 내부의 Slice는 62%, 블록 RAM은 81%가 사용되어 등가 게이트 수는 약 2,288,000이었으며, 최대 동작 주파수는 20.5-MHz로 나타났다.



(a) FPGA 구현 검증 시스템



(b) 실행 화면

그림 13. 설계된 PFFT8k 코어의 FPGA 구현 검증

V. 결론

OFDM 및 DMT 방식의 다수 반송파 변조 시스템에서 핵심 기능블록으로 사용될 수 있는 8192점 파이프라인 FFT/IFFT 프로세서 코어를 설계하였다. SNR 향상을 위해 새로운 2단계 CBFP 기법을 고안하여 적용함으로써 기존의 방식에서 필요로 하는 메모리의 약 20%만으로 CBFP 스케일링을 구현하였으며, 따라서 칩 면적과 전력소모를 줄임과 동시에 CBFP에 의한 출력 레이턴시의 증가를 제거하였다. 또한, CBFP 디코더에서 출력신호의 이득을 조정할 수 있도록 설계함으로써 출력의 정확도를 향상시켜 SQNR과 tone-by-tone SQNR이 크게 개선되도록 하였다. 설계된 PFFT8k 코어는 0.25- μ m CMOS 셀 라이브러리로 합성한 결과, 약 76,300 게이트와 390K 비트의 RAM, 그리고 39K 비트의 ROM으로 구현되었으며, 약 60-dB의 SQNR이 얻어졌다. 시뮬레이션 결과, 50-MHz@2.5-V로 안전하게 동작할 것으로 판단되며, 8192점 FFT/IFFT 연산에 164- μ s가 소요되어 DMT 기반의 VDSL 모델과 DAB 시스템의 사양을 만족하는 것으로 평가되었다. 설계된 PFFT8k 코어는 Xilinx FPGA 디바이스에 구현하여 정상 동작함을 확인하였으며, 반도체 설계자산인 IP (Intellectual Property)로 가공되어 고속 통신용 모델의 단일 칩 설계에 폭넓게 사용될 수 있을 것이다.

참고 문헌

[1] E.C. Ifeachor and B.W. Jervis, *Digital Signal Processing : A Practical Approach*, Addison-Wesley, 1996

[2] 김재석, 조용수, 조중휘, *이동통신용 모델의 VLSI 설계*, 대영사, 2001.

[3] T. de Couasnon, R. Monnier, and J.B. Rault, "OFDM for digital TV broadcasting", *Signal Processing*, vol. 39, pp. 1-39, Jan., 1994.

[4] John A.C. Bingham, *ADSL, VDSL and Multi-Carrier Modulation*, Wiley, 2000.

[5] *1024-points Complex FFT/IFFT Engine in ASIC/FPGA*, IComm Technologies, 1998.

[6] *CSC2411QL 1024-point FFT/IFFT*, Quick Logic Corp., 2001.

[7] *High-performance 1024-point complex FFT/*

IFFT, Xilinx Inc., 1999.

[8] E.O. Brigham, *The Fast Fourier Transform and Its Application*, Prentice Hall, 1998.

[9] S. He and M. Torkelson, "Design and implementation of a 1024-point pipelined FFT processor", *IEEE 1998 Custom Integrated Circuits Conference*, pp. 131-134, 1998.

[10] 양대성, 이승기, 신경욱, "복소수 승산기 코어의 파라미터화된 소프트 IP 설계", *한국통신학회 논문지*, vol. 26, no. 10B, 2001, 10.

[11] R.G. Lyons, *Understanding Digital Signal Processing*, Addison-Wesley, 1997.

[12] E. Bidet, C. Joanblanq, and P. Senn, "A fast single chip implementation of 8192 complex points FFT", *IEEE 1994 Custom Integrated Circuits Conference*, pp. 207-210, 1994.

이 승 기(Seung-Ky Lee)

준회원



2001년 2월 : 금오공과대학교
전자공학과 졸업
2001년 3월 ~ 현재 :
금오공과대학교
전자공학과
석사과정 재학 중

<주관심 분야> 통신 및 신호처리용 집적회로 설계, 로봇 제어

양 대 성(Dae-Sung Yang)

준회원



2000년 2월 : 금오공과대학교
전자공학과 졸업
2002년 2월 : 금오공과대학교
전자공학과 졸업
2002년 8월 ~ 현재 : (주)KEC
종합연구소 IC Design Center

<주관심 분야> 통신 및 신호처리용 집적회로 설계, 암호 프로세서 설계

신 경 욱(Kyung-Wook Shin)

정회원

1984년 2월 : 한국항공대학교 전자공학과 졸업
1986년 2월 : 연세대학교 대학원 전자공학과
(공학석사)
1990년 8월 : 연세대학교 대학원 전자공학과
(공학박사)



1990년 9월~1991년 6월: 한국
전자통신연구소
반도체연구단
(선임연구원)

1991년 7월~현재: 금오공과대
학교 전자공학부(교수)

1995년 8월~1996년 7월: University of Illinois at
Urbana-Champaign (방문교수)

<주관심 분야> 통신 및 신호처리용 집적회로 설계,
암호 프로세서 설계, 반도체 IP 설계