

OFDM 모뎀용 FFT/IFFT IP 자동 생성기

준회원 이진우*, 정회원 신경욱**, 준회원 김종환**, 정회원 백영석***, 어익수***

FFT/IFFT IP Generator for OFDM Modems

Jin-Woo Lee* *Associate Member*, Kyung-Wook Shin** *Reguler Member*,
Jong-Whan Kim** *Associate Member*, Young-Seok Baek***, Ik-Soo Eo*** *Reguler Members*

요약

본 논문은 OFDM 변복조 모뎀 설계에 핵심 IP로 사용될 수 있는 파라미터화된 FFT/IFFT 코어 생성기 FCore_GenSim(Parameterized FFT Core Generation & Simulation Program)에 대해 기술한다. FCore_GenSim은 FFT 코어의 Verilog-HDL 모델을 생성하는 parameterized 코어 생성기(PFFT_CoreGen)와 생성된 코어의 연산 정밀도를 분석해주는 fixed-point 시뮬레이터(FXP_FFTSim)로 구성된다. PFFT_CoreGen은 FFT 길이(64점~2048점 범위)와 입력/출력/중간결과/격자계수의 word-length(8-b~24-b 범위, 2-b 단위)를 지정하면, 지정된 사양을 갖는 FFT 코어의 Verilog-HDL 모델을 생성하며, 총 43,659 종류의 코어를 생성할 수 있다. 또한, 사용자의 필요에 따라 CBFP(Convergent Block Floating Point) 스케일링의 적용 여부를 지정할 수 있다. 생성되는 코어의 내부 구조는 FFT 길이에 따라 radix-2, radix-2/4, radix-2/4/8 알고리즘의 혼합구조가 적용되도록 하였으며, 또한 CBFP 스케일링의 적용 여부에 따라 R2SDF 단일구조 또는 R2SDF/R2SDC 복합구조가 적용되도록 함으로써 생성되는 코어의 회로 복잡도와 성능이 최적화되도록 하였다.

Key Words : FFT, IFFT, OFDM, IP, Core Generator

ABSTRACT

This paper describes a FCore_GenSim(Parameterized FFT Core Generation & Simulation Program), which can be used as an essential IP(Intellectual Property) in various OFDM modem designs. The FCore_GenSim is composed of two parts, a parameterized core generator(PFFT_CoreGen) that generates Verilog-HDL models of FFT cores, and a fixed-point FFT simulator(FXP_FFTSim) which can be used to estimate the SQNR performance of the generated cores. The parameters that can be specified for core generation are FFT length in the range of 64~2048-point and word-lengths of input/output/internal/twiddle data in the range of 8-b~24-b with 2-b step. Total 43,659 FFT cores can be generated by FCore_GenSim. In addition, CBFP(Convergent Block Floating Point) scaling can be optionally specified. To achieve an optimized hardware and SQNR performance of the generated core, a hybrid structure of R2SDF and R2SDC stages and a hybrid algorithm of radix-2, radix-2/4, radix-2/4/8 are adopted according to FFT length and CBFP scaling.

※ 본 논문은 2005년도 ETRI 위탁연구비 지원에 의한 연구결과의 일부임.

※ 반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

* C&S Technology, Inc.,

** 금오공과대학교 전자공학부 VLSI 설계 연구실 (kwshin@kumoh.ac.kr)

*** IT융합부품연구소, 한국전자통신연구원(ETRI)

논문번호 : KICS2004-12-311, 접수일자 : 2005년 12월 9일, 최종논문접수일자 : 2006년 3월 7일

I. 서론

직교 주파수 분할 다중화(Orthogonal Frequency Division Multiplexing; OFDM) 방식은 고속의 데이터 전송을 위한 기술로 유럽, 일본 및 호주의 디지털 TV 표준으로 채택되었으며, 무선 랜, 휴대 인터넷(WiBro), 지상파 DMB 및 UWB(Ultra Wideband) 등 디지털 무선 통신 및 방송 시스템에 폭넓게 이용되고 있다. OFDM의 원리는 고속의 data stream을 여러 개의 저속 stream으로 분할하여 이들을 다수개의 반송파(subcarrier)에 실어 동시에 전송하는 것이다. 저속의 병렬 반송파를 사용함으로써 심벌구간이 증가되고 이에 의해 multi-path delay spread에 의한 시간영역에서 상대적인 분산이 감소하게 된다. OFDM은 FEC(Forward Error Correction) coding, interleaving, modulation, IFFT(Inverse Fast Fourier Transform)/FFT 등의 기능블록으로 구성되며, 그 중 IFFT/FFT는 다수 반송파의 변·복조를 수행하는 핵심 블록이다. OFDM 신호의 전송은 심벌 단위로 이루어지며, 하나의 심벌에 다수의 반송파가 존재하게 된다. 이때, 하나의 OFDM 심벌이 갖는 반송파의 수는 FFT/IFFT 길이를 결정하게 되며, FFT 처리시간은 유효 심벌기간에 의해 결정된다¹⁾.

FFT/IFFT는 OFDM 모델에서 가장 많은 연산량과 하드웨어 복잡도를 필요로 하는 핵심 블록이며, 모바일 기기의 OFDM 변복조기는 저전력, 작은 칩면적 등이 중요한 요소가 된다. 한편, OFDM 기반 변복조기는 응용 시스템에 따라 64점~8192점 범위의 다양한 FFT 프로세서가 사용되며, SoC 기술의 보편화 추세에 따라 고속/저전력 FFT 프로세서 IP의 중요성이 증대되고 있다.

현재 상용화된 대부분의 FFT 프로세서 IP들은 특정 길이의 FFT를 연산하는 구조를 가져 응용범위가 제한적이고, 입력/출력/내부데이터/회전인자 등의 비트 수가 고정되어 있어 연산 정밀도 조정이 어렵다²⁻³⁾. 따라서 FFT 길이와 데이터의 word-length를 선택적으로 조정할 수 있는 parameterized FFT 프로세서 IP 개발이 필요하다. 특히, 알고리즘, 아키텍처, 연산회로 등 전체 design hierarchy에 걸친 설계 최적화와 연산 정밀도 향상을 위한 스케일링 기법 등 저전력/고성능 FFT 프로세서 관련 설계기술은 OFDM 모델 SoC 개발에 있어 핵심 요소가 된다.

본 논문에서는 OFDM 변복조 모델 설계에 핵심 IP로 사용될 수 있는 파라미터화된 FFT/IFFT 코어 생성기인 FCore_GenSim(Parameterized FFT Core

Generation & Simulation Program)를 개발하였다. FCore_GenSim은 FFT 코어의 Verilog-HDL 모델을 생성하는 파라미터화된 FFT 코어 생성기(PFFT_CoreGen)와 생성된 코어의 연산 정밀도를 분석해주는 fixed-point FFT 시뮬레이터(FXP_FFTSim)로 구성된다. PFFT_CoreGen은 FFT 길이에 따라 radix-2, radix-2/4, radix-2/4/8 알고리즘을 복합적으로 적용하였으며, CBFP 적용 여부에 따라 R2SDF 단일구조와 R2SDF/R2SDC 복합구조로 코어를 생성한다. 또한 FFT 연산에 사용되는 복소수 승산기를 생성하는 parameterized 복소수 승산기 생성기(PCMUL_GEN)를 포함하고 있다.

II. FFT 알고리즘과 아키텍처

2.1 FFT 알고리즘

데이터 x_n 에 대한 N 점 이산 푸리에 변환(Discrete Fourier Transform; DFT)은 식 (1)과 같이 정의된다.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot W_N^{nk} \quad (1)$$

$$W_N^{nk} = e^{-j\frac{2\pi}{N}nk}, \quad k = 0, 1, 2, \dots, N-1$$

DFT는 N^2 번의 복소수 승산과 $N(N-1)$ 의 복소수 가산으로 계산되므로 N 이 증가함에 따라 연산량이 급격히 증가한다. FFT는 식(1)의 연산량을 줄인 고속 연산 알고리즘이며, 대표적인 알고리즘으로는 Cooley-Turkey FFT 알고리즘, Winograd FFT 알고리즘, Prime Factor 알고리즘, Split-Radix FFT 알고리즘 등이 있다. FFT 알고리즘은 크게 나누어 radix-2, radix-4, radix-8과 같은 fixed-radix 알고리즘과 radix-2/4, radix-2/8, radix-2/4/8 등과 같은 mixed-radix 알고리즘으로 구분할 수 있다. Fixed-radix 알고리즘은 연산구조가 규칙적이므로 VLSI 구현에 적합하다는 장점은 있지만, mixed-radix 알고리즘에 비해 많은 연산량을 필요로 한다. 반면에 mixed-radix 알고리즘은 연산량은 작으나 연산구조의 불규칙성을 갖는다.

FFT 알고리즘들은 각기 고유의 연산구조와 연산량을 가지며, 이들은 연산 시간, 칩 면적, 전력소모 등에 직접적인 영향을 미친다. 따라서 효율적인 FFT 프로세서의 구현을 위해서는 연산량과 연산 흐름의 규칙성이 함께 고려되어야 한다⁴⁻⁵⁾.

본 논문의 FCore_GenSim은 radix-2/4/8 알고리즘을 기반으로 하며, FFT 길이에 따라 radix-2, radix-2/4 알고리즘을 복합적으로 적용하여 코어를 생성하며, CBFP 스케일링의 효율적인 적용을 위해 나비연산 후에 격자계수 곱셈이 이루어지는 DIF (Decimation-in-Frequency) 방식을 채택하였다.

2.2 FFT 아키텍처

FFT의 구현 방식은 시스토크 배열(systolic array) 구조, 파이프라인 구조, 그리고 column FFT 구조 등 다양한 하드웨어 구조로 구현 될 수 있다.

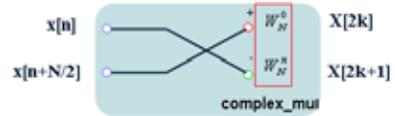
파이프라인 FFT 구조는 각 연산 stage에 하나 또는 그 이상의 나비 연산기를 할당하고 연산 stage들을 파이프라인 방식으로 동작시키는 구조이다. 이 방식은 다른 구조에 비해 비교적 적은 하드웨어로 높은 처리율을 얻을 수 있고 순차적인 입·출력을 제공하므로 고속 연산이 요구되는 분야에 많이 사용된다. 파이프라인 FFT 구조는 지연 메모리의 구현 방식에 따라 delay feedback(DF) 방식과 delay commutator(DC) 방식으로 구분되며, FFT 알고리즘의 radix와 병렬성에 따라 R2MDC, R2SDF, R2SDC, R4MDC, R4SDF, R4SDC, R2²SDC 등 다양한 형태로 구현이 가능하다. DF 구조는 적은 메모리를 필요로 하나, 나비 연산기의 효율이 작고 CBFP 스케일링의 구현이 용이하지 않다. 반면에 DC 구조는 DF 구조에 비해 많은 메모리를 필요로 하지만 나비 연산기의 이용 효율이 높고 CBFP 스케일링의 구현이 용이하다는 장점을 갖는다.

본 논문의 FCore_GenSim은 CBFP가 적용되지 않는 경우에는 R2SDF 단일 구조를 적용함으로써 적은 메모리로 구현되도록 하였으며, CBFP가 적용되는 경우에는 R2SDC 구조와 R2SDF 구조를 혼합한 hybrid 구조를 적용하여 코어를 생성함으로써 최소의 메모리로 구현되도록 하였다. 한편, CBFP 스케일링은 버퍼 메모리를 사용하지 않는 TS_CBFP (Two-Step CBFP) 방법⁶⁾을 적용하였다.

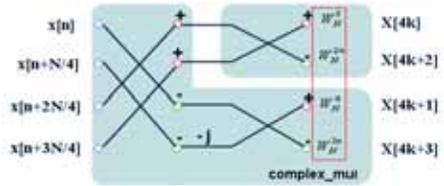
III. FFT 프로세서 구조 설계

본 장에서는 64점~2048점 범위의 FFT 코어를 생성하기 위해 FCore_GenSim에 사용된 FFT 프로세서 구조에 대해 언급한다.

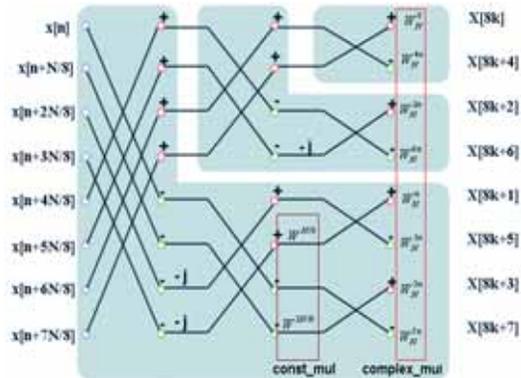
II장에서 논의된 여러 가지 FFT 알고리즘과 아키텍처의 장·단점 분석을 토대로, 그림 1에 제시된 radix-2, radix-2/4, radix-2/4/8 구조를 기본 구조로



(a) radix-2 algorithm



(b) radix-2/4 algorithm



(c) radix-2/4/8 algorithm

그림 1. FCore_GenSim에 사용된 알고리즘
Fig. 1. Algorithms used in FCore_GenSim

사용하였다. radix-2 구조는 그림 1-(a)처럼 '+', '-' 연산과 복소수 승산기로 구성되며, 그림 1-(b)의 radix-2/4 구조는 '+', '-', 'j' 연산과 복소수 승산기로 이루어지고, radix-2/4/8 구조는 그림 1-(c)와 같이 '+', '-', 'j' 연산, 상수 곱셈기, 복소수 승산기로 구성된다.

그림 2는 FCore_GenSim에서 생성되는 FFT 코어의 구조를 FFT 길이별로 나타낸 것이며, 코어의 내부 구조는 다음과 같이 결정된다. Radix-2/4/8 알고리즘을 기본으로 하여 FFT 길이가 $N=8^k$ 인 경우에는 radix-2/4/8 블록을 직렬 연결하여 구현하고, $N=2 \times 8^k$ 인 경우에는 radix-2 블록을, $N=4 \times 8^k$ 인 경우에는 radix-2/4 블록을 추가함으로써 코어 구조가 완성된다. 그림 2에서 보는 바와 같이, 64점~512점 FFT 코어는 64점 블록을 기본으로 하여 radix-2, radix-2/4, radix-2/4/8 구조를 앞단에 추가한 형태로 생성된다. 1024점과 2048점 코어의 경우에는 그림 2-(e)와 그림 2-(f)와 같은 변형된 구조를 갖도록 설계되었으며, 이를 통해 상수 곱셈기가 제거되고 메모리 크기가 감소하는 장점이 얻어졌다.

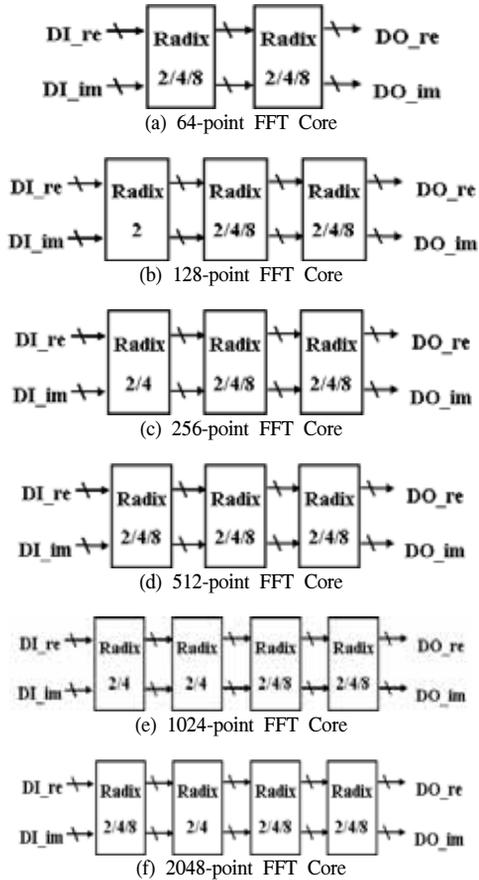


그림 2. FCore_GenSim에서 생성되는 FFT 코어의 구조
Fig. 2. Architectures of FFT cores generated by FCore_GenSim

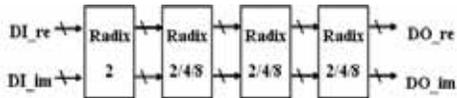


그림 3. Radix-2/4/8 기반의 1024점 FFT 코어의 구조
Fig. 3. Architecture of 1024-point FFT core based on radix-2/4/8

한편, $N=2 \times 8^3 = 1024$ 인 경우, 그림 3과 같이 3개의 radix-2/4/8 블록과 1개의 radix-2 블록을 사용하여 구현될 수 있다. 그러나 본 논문에서 적용한 그림 2-(e)의 구조는 2개의 radix-2/4 블록과 2개의 radix-2/4/8 블록을 사용하는 변형된 구조를 적용함으로써 상수 곱셈기 1개가 제거되었으며, 따라서 상수 곱셈기에 의한 절사오차가 발생하지 않아 연산 정밀도가 향상된다. 또한 CBFP 스케일링이 적용되는 경우에 R2SDC 구조가 사용되는 연산단계의 메모리 크기가 감소되는 장점도 갖는다.

표 1은 1024점 FFT 코어에 대해 그림 3의 방식과 본 논문에서 적용한 그림 2-(e)의 구조를 비교한

표 1. 1024점 FFT 코어의 비교
Table 1. Comparison of 1024-point FFT cores

	RAM	Gate Count	SQNR(dB)
그림 3의 경우	53,240(1.0)	63,903(1.0)	27.29(1.0)
본 논문의 경우	41,848(0.79)	61,768(0.97)	37.91(1.4)

※ 입력:12-b, 출력:16-b, 내부연산비트:16-b, 격자계수:14b

것이다. 코어의 생성 조건은 입력 12-b, 출력 16-b, 내부연산 16-b, 격자계수 14-b이고, CBFP를 적용한 경우이다. 본 논문에서 적용된 구조가 그림 3의 일반적인 구조에 비해 메모리는 약 21% 감소하였으며, 연산정밀도 SQNR은 약 40% 증가하였다.

IV. Parameterized FFT Core Generation & Simulation Program

본 장에서는 파라메터화된 FFT 코어 생성기인 FCore_GenSim의 내부 구조와 기능에 대해 기술한다. 그림 4는 FCore_GenSim의 GUI 화면을 보인 것이며, 고정점 FFT 시뮬레이터(FXP_FFTSim)와 FFT 코어 생성기(PFFT_CoreGen)로 구성된다. FCore_GenSim에서 지정 가능한 파라메터의 범위는 표 2와 같으며, 총 43,659개의 FFT 코어를 생성할 수 있다. 64점 FFT 코어에는 1개의 복소수 승산기만 사용되므로 CBFP 스케일링을 적용하지 않더라도 충분한 연산 정밀도가 얻어지므로 CBFP를 적용하지 않는다.

4.1 고정점 FFT 시뮬레이터(FXP_FFTSim)

OFDM 변복조기는 그 응용 시스템에 따라 요구되는 연산성능이 달라지며, FFT 프로세서의 연산 정밀도는 FFT length, 입력/내부데이터/회전인자의



그림 4. FCore_GenSim의 GUI 화면
Fig. 4. GUI of FCore_GenSim

표 2. FCore_GenSim의 파라미터 선택 범위
Table 2. Parameter ranges of FCore_GenSim

Parameters		Ranges
FFT Length		64~2048
Word-length	Input data	6-b~18-b(2-b step)
	Output data	8-b~24-b(2-b step)
	Internal result	8-b~20-b(2-b step)
	Twiddle factor	8-b~24-b(2-b step)
CBFP		On/OFF

비트 수 등에 영향을 받는다. 따라서 시스템에서 요구되는 연산 성능을 만족하도록 FFT 코어의 내부 데이터 및 회전인자의 비트 수를 결정하기 위해서는 fixed-point 시뮬레이션이 필요하다.

본 논문에서 개발된 FXP_FFTSim은 FFT 길이, 데이터의 비트 수, CBFP 스케일링의 사용 여부 및 출력 이득을 지정하면 해당 사양을 갖는 FFT 코어의 연산 성능을 예측해주는 fixed-point 시뮬레이터이다. FXP_FFTSim은 PFFT_CoreGen에서 생성되는 FFT 코어의 실제 회로와 동일한 알고리즘과 하드웨어 구조가 반영되도록 하여 시뮬레이션 결과의 신뢰성을 높였다. Radix-2/4/8를 기본으로 하여 FFT 길이에 따라 radix-2, radix-2/4, radix-2/4/8 연산 단계가 복합적으로 연산되며, 격자계수 생성, 상수 곱셈, 복소수 승산, 나비연산 등이 처리되도록 구현되었다. CBFP 스케일링은 FFT 길이가 128-point 이상인 경우에만 사용할 수 있으며, 이 경우 CBFP의 출력 이득 조정(Output Scaling Factor; OSF)은 $1(2^0) \sim 128(2^7)$ 범위에서 선택할 수 있다. 또한 사용자의 편의성을 위해 입력, 출력, 중간연산결과, 격자계수의 비트 수 중 1개를 Auto_Run 모드로 설정할 수 있도록 하였다. Auto 모드를 사용하는 경우, 선택된 파라미터의 가변 범위내의 모든 FFT 연산결과와 SQNR을 확인할 수 있도록 함으로써 사용자 편의성을 높였다. 그림 5는 128점 FFT 연산에 대해 Auto 모드를 적용하여 입력 데이터의 비트 수를 6-b~18-b 범위로 가변시키며 시뮬레이션 한 결과를 보인 것이다.

그림 6은 FXP_FFTSim의 내부 연산 흐름도를 보이고 있다. 입력 파라미터 중 FFT 길이, CBFP, mode(FFT 또는 IFFT) 선택에 따라 시뮬레이션 될 코어의 전체 구조가 결정된다. FFT length에 따라 연산 stage의 위치, 복소수 승산기의 개수, 상수 곱셈기의 적용 여부 등이 결정된다. 나머지 파라미터들은 각 연산단계의 입력, 출력 데이터의 비트 수를



그림 5. FXP_FFTSim의 Auto 실행 모드의 화면
Fig. 5. Result of FXP_FFTSim with Auto_run mode

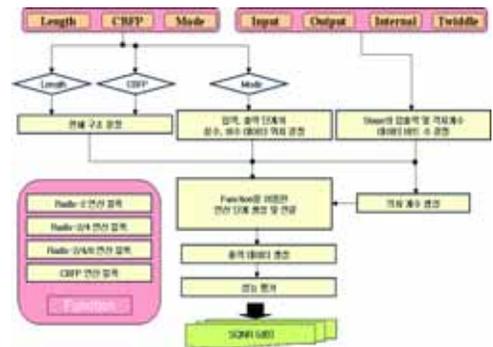


그림 6. FXP_FFTSim의 흐름도
Fig. 6. Flow diagram of FXP_FFTSim

결정하는 변수들이다. 먼저, 전체 구조가 결정되면 함수로 구현된 각 연산블록들이 호출되고, 각 연산 단계의 데이터 비트 수에 따라 시프트 연산자를 이용하여 비트 수를 맞추고 고정점 시뮬레이션이 수행된다. 연산 정밀도 SQNR은 최종 출력 데이터와 Matlab에서 계산된 결과 값을 비교해 계산된다.

4.2 FFT 코어생성기(PFFT_CoreGen)

PFFT_CoreGen는 응용분야에 따라 FFT 길이, 입력/출력/내부데이터/회전인자 등의 비트 수를 선택적으로 지정하여 해당 사양을 갖는 FFT 코어의 Verilog-HDL 모델을 생성하는 parameterized 코어 생성기이다. FFT 길이에 따라 그림 2의 구조를 갖는 FFT 코어가 생성되며, 표 2에 나열된 파라미터의 선택 범위를 가져 총 43,659 종류의 FFT 코어를 생성할 수 있다. 또한, 연산 정밀도 향상을 위해 생성되는 코어에 CBFP 스케일링의 삽입 여부를 선택할 수 있도록 하였다. 한편, 1024점과 2048점 코어의 경우에는 아키텍처 레벨의 최적화를 통해 첫 번째 복소수 승산 후에 radix-2/4 구조를 적용함으로써 CBFP

가 적용되는 코어의 전체적인 메모리 크기를 감소시켰으며, 또한 1024점 코어에서는 상수 곱셈기 1개를 제거함으로써 연산 정밀도가 향상되도록 하였다.

PFFFT_CoreGen의 내부 흐름도는 그림 7과 같으며, 크게 FFT 코어 생성부와 복소수 승산기 생성부로 구성된다. 복소수 승산기 생성부는 FFT 길이, 입력/내부데이터 비트 수에 따라 필요한 복소수 승산기 모듈을 생성한다. FFT 코어 생성부는 FFT 코어 생성에 필요한 모듈들을 라이브러리에서 호출하여 Verilog 코드를 생성하며, FFT 길이에 따라 전체 코어의 구조가 결정되고 입력과 내부 데이터의 비트 수에 의해 버티플라이 연산에 필요한 메모리와 가산기 블록을 생성한다. 그 다음, 상수 곱셈기 블록을 생성하고, 버티플라이 연산블록, 메모리, 상수 곱셈기 블록을 포함하는 각 연산단계 블록을 생성하게 된다. 마지막으로, 각 연산 단계 모듈, 격자 계수 블록, 복소수 승산 블록을 포함하는 TOP 모듈을 생성함으로써 코어의 Verilog-HDL 모델 생성이 완료된다. 그림 8은 PFFFT_CoreGen으로 1024점 FFT 코어를 생성한 결과 화면을 보인 것이다.

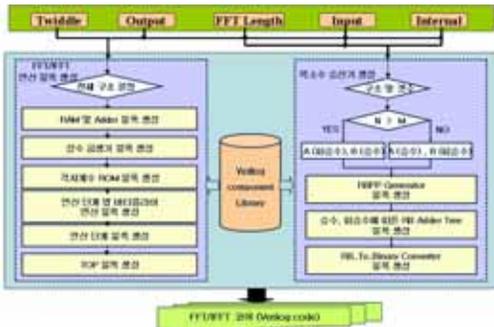


그림 7. PFFFT_CoreGen의 내부 흐름도
Fig. 7. Flow diagram of PFFFT_CoreGen



그림 8. PFFFT_CoreGen의 FFT core 생성
Fig. 8. FFT core generation of PFFFT_CoreGen

V. 검증 및 성능 평가

PFFFT_CoreGen에서 생성된 FFT 코어의 Verilog-HDL 모델은 ModelSim과 Matlab을 이용하여 그림 9와 같은 과정으로 논리검증과 성능평가를 수행하였으며, 0.25- μm CMOS 셀 라이브러리로 합성하여 게이트 수와 동작속도를 예측하였다.



그림 9. 성능 평가 방법
Fig. 9. Performance evaluation method

PFFFT_CoreGen에서 생성된 FFT 코어의 연산 정밀도를 평가하기 위해 그림 10의 방법으로 SQNR을 분석하였다. Matlab을 이용하여 만들어진 데이터는 생성된 FFT 코어의 ModelSim 시뮬레이션 입력 벡터와 Matlab을 이용한 부동점(floating-point) FFT 연산의 입력으로 사용된다. ModelSim 시뮬레이션 결과와 Matlab 시뮬레이션 결과로부터 연산오차와 식 (2)로 정의되는 SQNR을 분석하였다. 식 (2)에서 A는 Matlab을 이용한 부동 소수점 FFT 결과 값을 나타내며 B는 생성된 FFT 코어의 ModelSim 시뮬레이션 결과 값을 나타낸다.

$$SQNR = \frac{\sum [Re(A)]^2 + \sum [Im(A)]^2}{\sum [Re(A) - Re(B)]^2 + \sum [Im(A) - Im(B)]^2} \quad (2)$$

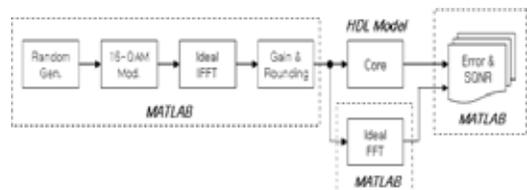
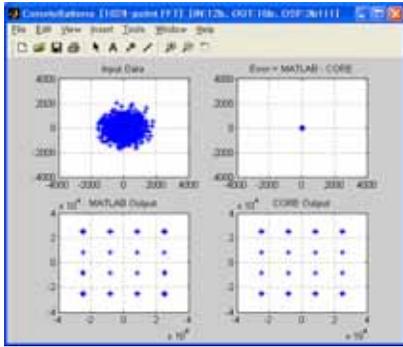
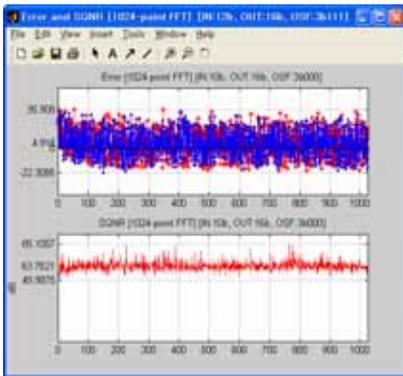


그림 10. FFT 코어의 연산 정밀도 분석 방법
Fig. 10. SQNR evaluation method of FFT core.



(a) constellations



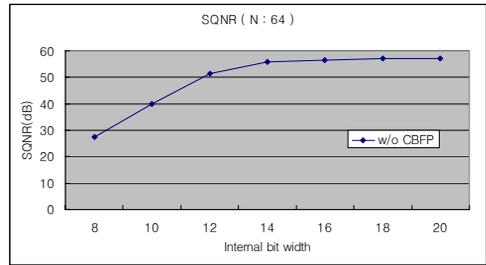
(b) error & SQNR

그림 11. 1024점 FFT 코어의 성능 평가
Fig. 11. Performance of 1024-point FFT core.(with CBFP, OSF=7)

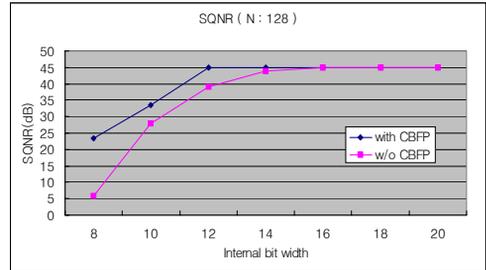
그림 11은 FCore_GenSim에서 생성된 1024점 FFT 코어의 성능을 평가한 것이다. CBFP를 적용하여 출력 이득을 7로 설정하고, 입력은 12-b, 출력은 16-b, 내부 비트는 16-b, 격자계수는 14-b인 경우이다. 입력 데이터는 그림 11-(a)에서 좌측 상단의 성상도이며, 좌측 하단의 성상도는 Matlab에서 얻어진 이상적 FFT 출력이고, 우측 하단의 성상도는 생성된 코어에서 얻어진 FFT 출력이다. 연산 오차는 우측 상단의 성상도와 같이 매우 작은 범위이다. 따라서 생성된 코어의 연산 정확도가 우수함을 알 수 있다.

연산 정확도는 SQNR 분석을 통해서도 알 수 있으며, 그림 11-(b)는 연산 오차와 SQNR 특성을 주파수 별로 보인 것이다. 평균 약 63-dB 정도의 연산 정밀도와 함께 전체 주파수 영역에서 우수한 성능을 나타내고 있음을 알 수 있다.

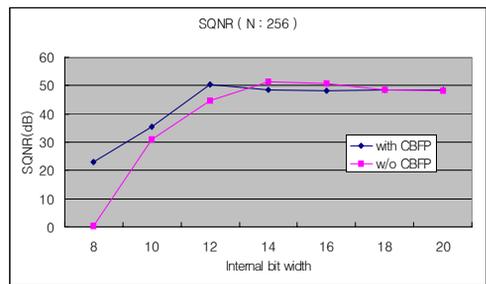
그림 12는 CBFP 스케일링 회로가 적용된 코어(OSF=0인 경우)와 적용되지 않은 코어의 SQNR 특성을 비교한 것이다. 코어의 생성 조건은 입력이 12-b, 출력이 16-b, 격자계수 14-b이며 내부 연산결과를 8-b-20-b 범위에서 변화시켰다.



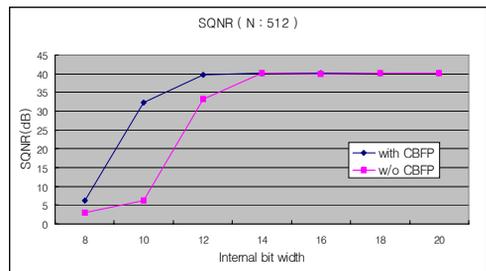
(a) 64점 FFT 코어



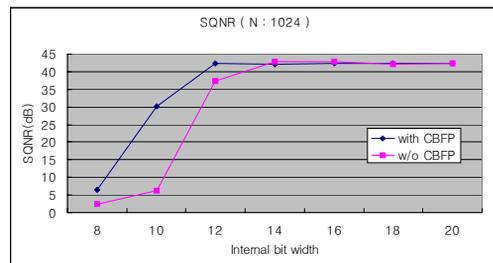
(b) 128점 FFT 코어



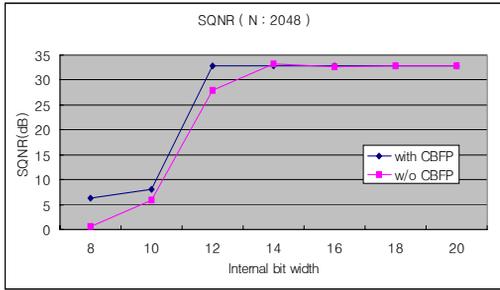
(c) 256점 FFT 코어



(d) 512점 FFT 코어



(e) 1024점 FFT 코어



(f) 2048점 FFT 코어

그림 12. FFT 코어의 SQNR 성능
Fig. 12. SQNR performance of FFT cores

기능 검증이 완료된 코어는 0.25- μm CMOS 셀 라이브러리를 이용하여 합성하여 게이트 수와 지연 시간을 추출하였다. 표 3은 PFFT_CoreGen에서 생성된 코어(입력 데이터 12-b, 출력 16-b, 격자계수 14-b, 내부 연산결과 16-b)의 합성결과를 요약한 것이다. CBFP가 적용되지 않은 경우, 64점 FFT 코어는 약 25,400개의 게이트와 1,500 비트의 RAM으로 구현되었으며, 1024점 FFT 코어는 54,100개의 게이트와 29,200 비트의 RAM으로 구현되었다. CBFP가 적용된 코어의 경우, 1024점 FFT 코어는 약 61,800개의 게이트와 41,900 비트의 RAM으로 구현되었으며, 2048점 FFT 코어는 약 67,400개의 게이트와 71,600 비트의 RAM으로 구현되었다. 표 3의 데이터를 비교하면, CBFP 스케일링을 갖는 코어는 CBFP 스케일링을 갖지 않는 코어에 비해 게이트 수와 메모리 크기가 각각 약 10%와 20% 정도 증가한다. 그러나 그림 12에서 보는 바와 같이 CBFP 스케일링을 갖는 코어의 연산 정밀도가 더 우수하므로 사용자의 시스템 사양에 따라 적합한 코어를 생성하여 사용할 수 있다.

생성된 FFT 코어의 논리합성 결과로부터, 내부의 최대 지연시간을 분석한 결과 약 11.4-ns로 예측되

표 3. FFT 코어의 하드웨어 복잡도
Table 3. Hardware complexity of FFT cores

FFT cores without CBFP scaling						
FFT Length	64	128	256	512	1024	2048
Gate Count	25,346	39,321	40,651	45,659	54,034	61,076
RAM(bit)	1,472	3,468	7,104	14,271	29,120	57,792
FFT cores with CBFP scaling						
FFT Length	-	128	256	512	1024	2048
Gate Count	-	43,727	44,957	50,105	61,767	67,344
RAM(bit)	-	6,124	10,592	17,504	41,848	71,544

※ 입력:12-b, 출력:16-b, 내부연산비트:16-b, 격자계수:14b

있으며, 레이아웃 후의 배선에 의한 지연을 고려하더라도 2.5-V 전원전압에서 60-MHz로 안전하게 동작 가능할 것으로 평가된다. 따라서 64점 FFT 코어는 77 cycles의 latency와 0.9- μs 의 연산시간을 가지며, 2048점 FFT 코어는 2,244 cycles의 latency와 30.7- μs 의 연산시간을 갖는다. 표 4는 PFFT_CoreGen에 의해 생성되는 코어의 latency와 FFT 연산시간을 요약한 결과이다.

표 4. FFT 코어의 연산 성능
Table 4. Computation time of FFT cores

FFT Length	Latency (cycles)	Computation Time	
64	77	0.9 us (64x15 ns)	
w/o CBFP	128	144	1.9 us (128x15 ns)
	256	273	3.8 us (256x15 ns)
	512	532	7.7 us (512x15 ns)
	1024	1,045	15.4 us (1024x15 ns)
	2048	2,072	30.7 us (2048x15 ns)
with CBFP	128	186	1.9 us (128x15 ns)
	256	315	3.8 us (256x15 ns)
	512	574	7.7 us (512x15 ns)
	1024	1,217	15.4 us (1024x15 ns)
	2048	2,244	30.7 us (2048x15 ns)

VI. 결론

OFDM 변복조 모델 설계에 핵심 IP로 사용될 수 있는 FFT 코어 생성기 FCore_GenSim를 개발하였다. FCore_GenSim은 고정점 시뮬레이터(FXP_FFTSim)와 코어 생성기(PFFT_CoreGen)로 구성된다. PFFT_CoreGen은 64점~2048점 범위의 FFT 코어 43,659 종류를 Verilog-HDL로 모델링하여 생성한다. 사용되는 시스템에 따라 입력/출력/내부연산 격자계수의 word-length를 8-b~24-b 범위에서 2-b 단위로 선택하여 코어를 생성할 수 있다. 연산 정밀도 향상을 위해 CBFP 스케일링 회로를 선택적으로 적용할 수 있다. CBFP가 적용되는 경우 R2SDF와 R2SDC의 혼합구조로 설계하여 R2SDC 단일 구조에 비해 메모리를 감소시켰다. PFFT_CoreGen에는 FFT 연산에 필요한 복소수 승산기를 생성하는 모듈이 포함되어 있다.

기능검증 결과 PFFT_CoreGen에서 생성된 1024점 코어(입력 12-b, 출력 16-b, 내부연산비트 16-b, 격자계수 14-b)는 CBFP를 적용한 경우(OSF=7) 평균 62 dB의 연산 정밀도가 얻어졌으며, 합성 결과

약 61,800개의 게이트와 41,900 비트의 RAM으로 구현된다. FCore_GenSim은 무선 랜에서부터 DMB 모델에 이르기까지 OFDM 기반의 다양한 시스템 설계에 폭넓게 사용될 수 있을 것이다.

참 고 문 헌

[1] 조용수, “무선 멀티미디어 통신을 위한 OFDM 기초”, 대영사, 2001.
 [2] “High-performance 1024-point complex FFT/IFFT V2.0”, Xilinx Inc., 2001.
 [3] “FFT MegaCore Function V1.2.0”, <http://www.altera.com/mysupport>, 2004.
 [4] L. Jia, Y. Gao, J. Isoaho and H. Tenhuen, “A New VLSI-Oriented FFT Algorithm And Implementation” *Proceeding of 1998 IEEE ASIC Conference*, pp. 337-341, 1998.
 [5] Y.J. Hongil and J. Kim, “New efficient FFT algorithm and pipeline implementation result for OFDM/DMT applications”, *IEEE Trans. on Consumer Electronics*, vol. 49, no. 1, pp. 14-20, Feb. 2003.
 [6] 신경욱, “R4SDF/R4SDC Hybrid 구조를 이용한 메모리 효율적인 2k/8k FFT/ IFFT 프로세서 설계”, *한국해양정보통신학회 논문지*, 제8권2호, pp. 430-439, 2004.
 [7] 양대성, 이승기, 신경욱, “2단계 수평 블록 부동점 스케일링을 이용한 8192점 파이프 라인 FFT/IFFT 프로세서”, *한국통신학회 논문지*, vol. 27, no. 10C, pp. 963-972, 2002. 10.

이진우 (Jin-Woo Lee)

정회원



2004년 2월 금오공과대학교 전자공학과 졸업
 2006년 2월 금오공과대학교 전자공학과 석사
 C&S Technology 연구원
 <관심분야> 마이크로프로세서 설계, SoC 설계, 통신/신호처리

집적회로 설계

신경욱 (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 한국전

자통신연구소 반도체연구단(선임연구원)
 1991년 7월~현재 금오공과대학교 전자공학부(교수)
 1995년 8월~1996년 7월 University of Illinois at Urbana-Champaign (방문교수)
 2003년 1월~2004년 1월 University of California at San Diego(방문교수)
 <관심분야> 통신 및 신호처리용 SoC 설계, 네트워크 정보보호 SoC 설계, 반도체 IP 설계

김종환 (Jong-Hwan Kim)

준회원



2005년 2월 금오공과대학교 전자공학과 졸업
 2005년 3월~현재 금오공과대학교 전자공학과 석사과정 재학 중
 <관심분야> SoC 설계, 반도체 IP 설계

백영석 (Young-Seok Baek)

정회원



1985년 2월 한양대학교 전자공학과 졸업(공학사)
 1987년 2월 한양대학교 대학원 전자공학과 졸업(공학석사)
 1989년 8월~현재 한국전자통신연구원 IT융합부품연구소, 책임연구원

어익수 (Ik-Soo Eo)

정회원



1985년 2월 부산대학교 전자공학과 졸업
 1987년 8월 한국과학기술원 전기 및 전자공학과 졸업
 1987년 10월~현재 한국전자통신연구원 IT융합부품연구소 책임연구원

<관심분야> VLSI 설계, 무선모뎀 구현, 통신신호처리