

# ADSL G.LITE 모뎀을 위한 주파수 영역 프로세서의 설계

고우석, 김준석\*, 고태호\*, 윤대희  
연세대학교 전기전자공학과, (주) 인타임\*

## Frequency Domain Processor for ADSL G.LITE Modem

Woo-Suk Ko, Joon-Seok Kim\*, Taiho Ko\*, Dae-Hee Youn  
Electrical & Electronic Eng., Yonsei University, InTime Corp.\*  
E-MAIL : kws@assp.yonsei.ac.kr

### 요 약

G.LITE ADSL 모뎀에 적합한 주파수 영역 프로세서의 구조를 제안하였다. 주파수 영역의 연산과정에 대한 알고리즘 수준의 최적화를 수행하였고, 하드웨어 자원할당에 따른 설계의 효율성도 분석하였다. 제안된 프로세서는 한 개의 실수 곱셈기와 두 개의 실수 덧셈기를 병렬로 연결한 구조를 가지며, 기존의 연구결과에 비해 작은 하드웨어 크기를 차지한다. 설계된 시스템은 삼성 0.35 $\mu$ m 표준셀 라이브러리를 사용하여 합성하였으며, G.LITE ADSL 모뎀에 적합하게 적은 하드웨어 자원으로 필요한 연산을 효율적으로 수행한다.

### 2. G.LITE 시스템

G.LITE ADSL 모뎀의 개념적인 블록 다이어그램은 그림 1과 같다. 왼쪽 신호경로는 uplink(전송단)를 나타내고 오른쪽 경로는 downlink(수신단)를 나타낸다. Uplink에 대해서 보면, scramble되고 FEC 인코딩된 데이터는 전송채널에서의 burst error를 막기 위해 interleave된다. 그리고 직교채널의 전송특성에 따라 할당된 비트정보를 이용하여 QAM(Quadrature Amplitude Modulation) 신호로 바뀌고 세부적인 gain이 조절된다 [2]. QAM 신호들은 IDFT를 통해 DMT 변조한 후 DAC를 통해 아날로그 신호로 변경하여 전송한다.

### 1. 서 론

최근 인터넷과 멀티미디어 서비스에 대한 요구가 증가함에 따라 고속 데이터 통신에 대한 필요성이 부각되고 있다. ADSL (Asymmetric Digital Subscriber Lines)은 기존의 아날로그 전화선을 통해 고속 디지털 서비스를 제공하므로, 낮은 초기 투자비용으로 구현할 수 있는 효율적인 시스템이다.

ADSL은 DMT(Discrete Multi-Tone) 방식에 기반하고 있다. DMT 방식은 전송특성에 따라 독립적으로 변조되는 여러 직교채널들을 사용하는데, IDFT(Inverse DFT)를 사용하여 효율적으로 구현할 수 있다. ITU-T는 G.DMT와 G.LITE ADSL 표준안을 제정하였다. G.LITE는 G.DMT에 비해 낮은 전송률을 지원하는 대신 적은 하드웨어 비용을 요구한다 [1-2].

ADSL 모뎀이 수행해야 하는 주파수 영역 연산중에서 FFT와 FEQ는 가장 많은 연산량을 차지하는 연산과정이다. 따라서 이들 과정에 대한 효율적인 구현은 ADSL 모뎀의 하드웨어 복잡도를 크게 줄일수 있으며, 이를 위해서 많은 연구들이 수행되어 왔다 [3-4]. 그러나 기존의 연구들은 G.DMT 모뎀의 효율적인 구현을 위한 것으로, G.LITE 시스템에 그대로 적용하기에는 부적합하다. 이에 본 논문에서는 좀더 간단한 방식의 G.LITE 시스템에 적합한 주파수 영역 프로세서의 구조를 제안하였다.

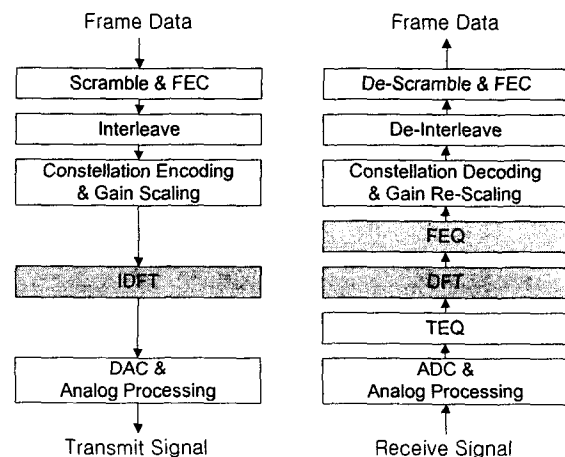


그림 1. ADSL G.LITE 모뎀의 개념적인 블록도

Downlink는 uplink에서 수행한 과정들의 역과정들을 수행한다. 단 TEQ(Time-domain Equalization)와 FEQ(Frequency-domain Equalization)의 과정이 추가적으로 필요한데, TEQ는 ISI를 줄이기 위해 신호의 유효 전송경로를 짧게 만들고, FEQ는 전송도중의 채널에 의한 영향을 제거한다.

그림 1에서 검게 칠해진 블록들은 주파수 영역의 연산 블록을 나타낸다. 주파수 영역의 연산과정은 DFT, IDFT와 FEQ로 이루어져 있고, 이들은 샘플단위가 아닌 심볼

단위로 데이터를 처리한다. 또한 매우 많은 연산량을 요구하는 반면 연산구조는 상대적으로 간단하고 규칙적인 구조를 가지고 있다.

주파수 영역 연산과정에 대한 좀더 자세한 블록도를 그림 2에 나타내었다. Downlink에 대해서는 수신된 신호로부터 cyclic prefix가 제거한 후 256-point FFT 과정을 수행한다. FEQ의 출력 신호는 QAM 신호의 원래 크기로 재조정된 후 신호판정이 내려진다. Uplink에 대해서는 QAM 신호들에 대해서 64-point IFFT 과정을 통해 DMT 변조시킨 후, cyclic prefix를 첨가하여 전송한다.

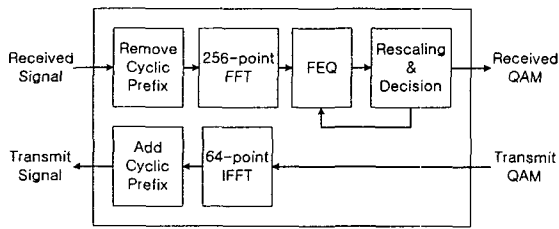


그림 2. G.LITE ADSL 모델의 주파수 영역 연산과정

그림 2의 주파수 영역의 연산과정을 수행할 프로세서는 매우 많은 연산량을 수행해야 하며, 작은 설계면적을 차지하고 전력을 적게 소모해야 한다. 상용 DSP를 이용한 구현은 효율과 성능면에서 한계가 있으므로, 전용 하드웨어로 로직으로 구현하는 것이 바람직하다. 설계할 시스템의 동작 주파수는 downlink의 샘플링 주파수의 정수배로 사용해야 아날로그 front-end와의 인터페이스 로직이 복잡해지는 것을 피할 수 있다. 본 논문에서 선택한 동작 주파수는 35.328 MHz로써, 샘플링 주파수인 1.104 MHz의 32배에 해당한다 [2].

### 3. 알고리즘 최적화

알고리즘 최적화는 구현에 용이하도록 알고리즘을 수정함으로써 연산량을 줄이고 시스템의 복잡도를 감소시킬 수 있다. 하드웨어로 로직을 설계하는 관점에서 알고리즘이 가져야 할 바람직한 특징은 다음과 같다.

- 간단한 제어로직만을 요구하는 규칙적 연산구조
- 연산자원을 공유할 수 있도록 비슷한 연산사용
- 메모리 사용량 감소를 위한 in-placement 방식지원

#### FFT and IFFT

그림 2에서 FFT 블록의 출력과 IFFT 블록의 입력 데이터는 복소수 값이고, FFT 블록의 입력과 IFFT 블록의 출력 데이터는 실수값이다. 따라서 연산량 감소 측면에서 CFFT(Complex FFT)보다 RFFT(Real-valued FFT)를 사용하는 것이 바람직하다. 표 1은 길이  $N(=2^M)$ 의 입력

데이터에 대한 CFFT와 여러 가지 RFFT 알고리즘의 연산량과 연산의 규칙성을 나타내고 있다.

표 1. Real-Valued FFT 알고리즘 분석

알고리즘	연산량/규칙성
Direct CFFT	곱셈 2MN, 덧셈 3MN 한종류의 butterfly 필요, 규칙적인 연산
Radix-2 RFFT	곱셈 $MN-2N+2$ , 덧셈 $3/2MN-5/2N+4$ 거의 CFFT에 유사한 규칙적 연산구조
Higher-radix RFFT	Radix-2 RFFT보다 적고 split-radix RFFT보다 많은 곱셈, 덧셈 radix-2 RFFT보다 복잡한 연산구조
Split-radix RFFT	곱셈 $2/3MN-19N+3$ , 덧셈 $4/3MN-17/9N+3$ 복잡한 제어와 불규칙한 데이터 재정렬

표 1에서 곱셈과 덧셈은 모두 실수 연산을 나타내고, 4개의 실수곱셈과 2개의 실수덧셈을 사용하여 1개의 복소수 곱셈을 처리하는 방식을 사용하였다. Direct CFFT는 제로값의 허수부를 갖도록 확장된 입력 데이터에 대한 연산을 의미하며, 연산량 및 복잡도 비교를 위하여 표 1에 같이 나타내었다.

표 1에서 보듯이 모든 종류의 RFFT는 Direct CFFT보다 절반이하의 연산량만이 필요함을 알 수 있다. Higher-radix RFFT와 split-radix RFFT는 radix-2 RFFT보다 연산량을 적게 요구하지만, 상대적으로 연산의 규칙성이 감소하여 제어로직과 데이터 어드레싱을 복잡하게 만든다. 한편 radix-2 RFFT는 Direct CFFT의 연산량과 필요한 메모리량을 약 1/2로 감소시키면서 CFFT의 연산 규칙성을 어느정도 유지할 수 있다. Radix-2 RFFT 알고리즘은 간단한 데이터 어드레싱 수정을 통해서 in-placement 메모리 액세스 방식을 구현할 수 있다 [5]. 이에 본 논문에서는 앞서 살펴본 설계에 적합한 알고리즘의 특징을 만족시키는 radix-2 RFFT 알고리즘을 채택하여 downlink와 uplink에 각각 필요한 FFT와 IFFT를 구현하였다.

#### FEQ

ADSL 모델에 일반적으로 사용되는 FEQ는 각 직교채널마다 1-tap FIR 필터를 갖는 구조이다. 이때 전송채널의 특성을 추적하도록 필터 계수를 갱신하는 적응 알고리즘은 다음과 같은 사항을 고려하여 채택해야 한다.

- 직교채널당 수렴성능
- 수렴속도의 조절 가능성

직교채널당 수렴성능을 동일하게 만드는 것은 각 직교채널마다 동일한 등화과정 성능을 얻기 위해 필요하다. 또한 G.LITE ADSL 시스템의 전송모드에 따라 적응계수를 다르게 사용하여 수렴속도를 조절할 수 있어야 한다.

결과적으로 채택한 필터계수를 갱신시키기 위한 적응 알고리즘은 정규화된 LMS 알고리즘이다.

본 논문에서 설계한 개념적인 FEQ 블록 다이어그램은 그림 3과 같다. 이때 입력  $x(n)$ 은 수신된 신호,  $w$ 는 필터계수,  $\mu$ 는 수렴계수,  $gain$ 은 신호전력 정규화와 미세 조정값을 포함하는 값이고, 출력  $y(n)$ 은 원래 크기가 복원된 QAM 신호에 대해 신호판정을 내린 값이다. 이때 LMS 알고리즘의 desired 입력으로서는 신호판정된 값과 송수신단간 약속된 PRD(Pseudo Random Data) 값이 전송모드에 따라 선택되어 사용된다 [2].

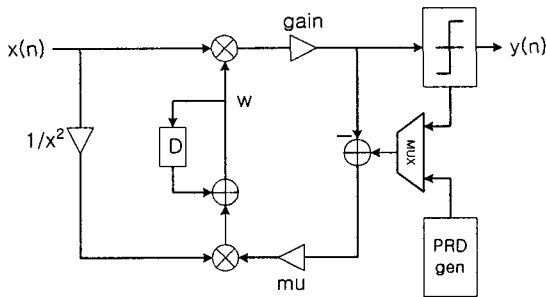


그림 3. 개념적인 FEQ 블럭도

#### 4. 하드웨어 설계 최적화

본 절에서는 주파수 영역 연산과정에 대한 연산량 분석과 하드웨어 자원 할당에 따른 시스템의 효율성 분석을 토대로 설계할 시스템의 구조를 최적화시켰다. 표 2는 주파수 영역 연산과정별 곱셈, 덧셈과 메모리 참조연산의 연산량을 나타내고 있다.

표 2. 주파수 영역 연산과정의 연산량 분석

연산과정	곱셈	덧셈	메모리 참조	총합	비율(%)
256-FFT	1,538	2,308	4,096	7,942	53.64
FEQ	2,520	1,440	1,200	5,160	34.85
64-IFFT	252	571	890	1,713	11.51

표 2에서 보듯이 FFT는 주파수 영역 연산과정 전체 연산량의 약 53%를 차지하고 FEQ는 약 35%를 차지한다. FFT와 IFFT는 메모리 참조 연산이 곱셈이나 덧셈보다 많이 사용되고, FEQ의 경우는 곱셈연산이 메모리 참조 연산보다 많이 사용된다. 이때 가장 많은 연산량을 차지하는 메모리 참조연산이 전체 시스템의 복잡도를 결정하는데 중요한 요소가 된다.

주파수 영역의 연산들은 3절에 살펴본 바와 같이 주로 곱셈과 덧셈의 산술연산들로 이루어져 있다. 따라서 본 논문에서는 곱셈기와 덧셈기가 병렬로 연결된 구조를 취하여 많은 연산량을 수행하도록 하였다. 이때 하드웨어 사용률과 실시간 동작을 고려하여 하드웨어 자원을

할당하는 문제가 설계된 시스템의 효율성을 좌우한다. 표 3은 연산기 자원할당에 따른 연산량 분석을 나타낸다. 이때 메모리 액세스 방식으로는 in-placement 방식을 사용했으므로 메모리 참조 연산은 일정한 연산량을 차지한다.

표 3. 연산자원 할당에 따른 연산량 분석

연산자원할당	산술연산	메모리참조	총 사이클 수
곱셈기 1개 덧셈기 1개	7,080	6,510	8,826
곱셈기 1개 덧셈기 2개	6,272	6,510	7,620
곱셈기 2개 덧셈기 2개	4,582	6,510	6,766

곱셈기와 덧셈기를 각 1개씩 할당하는 첫 번째 방식은 산술연산에 의한 연산량이 메모리 참조에 의한 연산량보다 많고, 세 번째 할당방식은 반대이다. 시스템 동작주파수가 35.328 MHz이고 DMT 심볼구간이 231.884  $\mu s$ 이므로, 한 심볼구간 동안 사용가능한 사이클 수는 8,192 사이클이다. 표 3에서 보듯이 첫 번째 할당방식은 연산자원을 너무 적게 할당하여 실시간으로 동작하지 못하고, 세 번째 방식은 너무 많이 할당되어 자원 활용률이 떨어진다. 반면 두 번째 할당방식은 메모리 참조연산을 수행하는 동안 연산자원을 효율적으로 사용하며, 실시간 구현조건도 만족시킨다. 이에 본 논문에서는 곱셈기 1개와 덧셈기 2개를 할당하여 프로세서를 설계하였다.

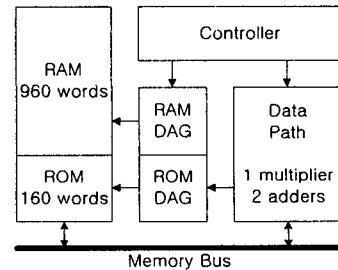


그림 4. 주파수 영역 프로세서의 전체구조

그림 4에 설계된 주파수 영역 프로세서의 전체 시스템 구성도를 나타내었다. 그림에서 보듯이 전체 시스템을 연산로직인 Data-Path와 주소발생기(DAG: Data Address Generator), 제어로직으로 이루어져 있다. 앞서 자원할당 분석에 따라 설계된 Data-path의 자세한 구조는 그림 5에 나타내었다.

Data-Path는 곱셈기와 덧셈기가 병렬로 연결된 구조를 가지며, 병렬연산과 파이프라인 방식의 작업 스케줄링을 통해 연산의 고속성과 효율성을 증대시킬수 있다. 모든

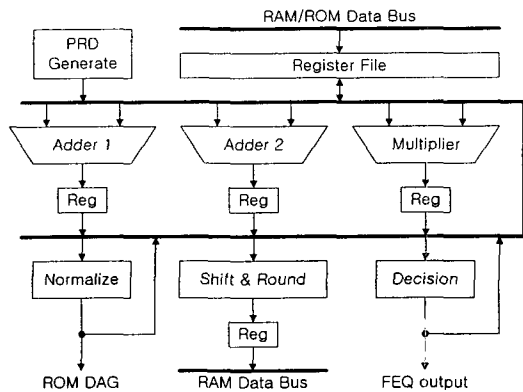


그림 5. Data-Path의 구조

메모리 액세스는 레지스터를 통해서 이루어고, 모든 연산의 결과는 레지스터에 래치되도록 설계하였다. 이것은 메모리 액세스 시간과 연산기의 시간지연을 최소화하기 위한 것이다. 그림 5에서 보듯이 고정 소수점 연산에 의한 오차를 최소화하기 위해서 round 로직을 사용하였으며, 신호전력으로 나누는 나눗셈 연산은 고속연산을 위해 ROM 테이블을 사용하여 구현하였다.

## 5. 결 과

동작주파수를 고려할 때 설계된 시스템이 한 개의 DMT 심볼을 처리하는데 사용할 수 있는 사이클수는 cyclic prefix가 첨가된 심볼의 경우 8,192 사이클이고, 첨가되지 않을 경우 8,704 사이클이다. 표 4는 설계된 시스템이 주파수 영역 연산과정별 사용하는 사이클수를 나타내고 있다. 표에서 보듯이 초기전송 모드와 데이터 전송 모드에서 모두 실시간으로 동작함을 알 수 있다.

표 4. 주파수 영역 연산과정별 동작 사이클수

주파수 영역 연산과정		사이클 수
256-point FFT		4,096
FEQ	초기전송 모드 (cyclic prefix 첨가)	2,318
	초기전송 모드	2,562
	데이터 전송 모드	2,806
64-point IFFT		962
데이터 인출력		640
평균합	초기전송 모드	8,138
	데이터 전송 모드	8,504

설계된 시스템은 VHDL로 기술되고 삼성 0.35  $\mu\text{m}$  표준셀 라이브러리를 사용하여 합성하였다. 표 5는 합성된 시스템의 각 블록별 게이트 수와 시간지연을 나타내고 있다. 전체 시스템의 게이트 수는 16,634 게이트이고, 시간지연은 12.76 ns로서, 합성후에도 정상적으로 동작함을 확인할 수 있다. 제안된 시스템은 두 개의 곱셈기를 사

용하는 Kiss의 ALU 구조[3]나 Wang이 제안한 FFT/IFFT 프로세서[4]보다 적은 하드웨어 자원을 사용하여 효율적으로 연산을 수행한다.

표 5. 설계된 시스템의 합성결과

서브블럭	게이트 수	시간지연 (ns)
Data-Path	10,408	12.76
주소발생기(DAG)	1,777	6.46
제어로직	4,449	7.58
총합	16,634	12.76

## 6. 결 론

G.LITE ADSL 모델에 적합한 주파수 영역 프로세서의 구조를 제안하였다. G.LITE 모델이 주파수 영역에서 수행하는 연산중에서 가장 많은 연산량을 차지하는 부분은 FFT와 FEQ 과정으로서, 이를 효율적으로 구현하는 것이 전체 시스템의 효율성을 결정한다.

설계과정은 크게 알고리즘 수준에서의 최적화와 하드웨어 설계상의 최적화 과정으로 나뉜다. 연산량을 줄이고 하드웨어 로직 구현에 적합하도록 radix-2 RFFT 알고리즘을 채택하였으며, FEQ 연산을 위해 정규화된 LMS 알고리즘을 사용하는 1-tap FIR 필터구조를 택하였다. 설계된 시스템은 1개의 실수 곱셈기와 2개의 실수 덧셈기가 병렬로 연결된 구조를 가지며, 이를 통해 고속의 연산을 효율적으로 수행할 수 있도록 설계되었다.

제안된 시스템은 VHDL로 기술하고 0.35  $\mu\text{m}$  기술로 합성하여 검증하였으며, 기존의 Kiss나 Wang등이 제안한 시스템에 비해 적은 하드웨어 면적을 차지하면서 효율적으로 연산을 수행함을 알 수 있다.

## 참고문헌

- [1] ITU-T Recommendation G.992.1, Asymmetrical Digital Subscriber Line (ADSL) Transceivers, Jun. 1999
- [2] ITU-T Recommendation G.992.2, Splitterless Asymmetric Digital Subscriber Line (ADSL) Transceivers, Jul. 1999
- [3] L.Kiss, E.Hanssens, et al., A Customizable DSP for DMT-Based ADSL Modem, *Proc. of the 11th IEEE International ASIC Conference*, pp. 349-353, 1998
- [4] C.Wang and C.Chang, A Novel DHT-Based FFT/IFFT Processor for ADSL Transceiver, *Proc. of ISCAS*, vol. 1, pp. 51-54, 1999
- [5] H.V.Sorensen, et al., Real-Valued Fast Fourier Transform Algorithms, *IEEE Trans. On ASSP*, vol. ASSP-35, no. 6, pp. 849-863, Jun. 1987