

## 고속 병렬처리 기법을 활용한 주파수 도약 신호 분석

# Frequency Hopping Signal Analysis Using High-Speed Parallel Processing

이 광 용 · 윤 현 철 · 이 현 휘

Kwang-Yong Lee · Hyun-Chul Yoon · Hyeon-Hwi Lee

### 요 약

본 논문에서는 주파수 도약 신호를 고속 병렬처리 구조로 설계하여 추출하는 기법에 대해 연구하였다. 주파수 도약 시스템은 반송파 주파수를 무작위한 것으로 보이는 패턴으로 변형시키기 때문에 고정 신호와는 달리 신호를 탐지하기 어렵고, 분석에 많은 시간이 소요되는 특성을 가진다. 이를 해결하기 위해 주파수 도약 신호를 분석하는 방법을 고속 병렬처리 기법을 적용하여 설계하였다. 병렬처리를 위해 GPU를 사용하는 CUDA를 사용하였고, 단일처리 사용 결과와의 성능 평가 결과를 비교하였다. 그 결과, 단일처리 대비 연산 수행 속도 면에서 약 8.53배 성능이 향상됨을 확인하였다.

### Abstract

In this paper, we studied a technique of extracting a Frequency Hopping(FH) signal for analysis using high-speed parallel processing structure. Unlike fixed frequency signal, FH signal is difficult to detect and analyze because FH systems use many random frequencies instead of a single carrier frequency. To solve this problem we designed a method that analyze FH signal using high-speed parallel processing. In order to apply parallel processing, we use CUDA using GPU and compare single processing with parallel processing. As a result, using CUDA on a GPU is about 8.53 times faster than single processing.

Key words: Frequency Hopping(FH), Signal Processing, Parallel Processing, CUDA

## I. 서 론

주파수 도약 시스템은 한정된 주파수 대역에서 다수의 사용자들이 무선통신을 수행하면서 사용자들 간의 간섭을 최소화할 수 있는 장점이 있다<sup>[1]</sup>. 또한, 재밍과 감청을 회피하기 위한 전자전 군용 통신 시스템에 널리 사용되고 있다<sup>[2][3]</sup>. 광대역 수신기를 이용하여 수신한 주파수 도약 신호를 분석하기 위해서는 중심 주파수, 샘플링 주파수, 신호 대역폭 등을 알아야 한다.

기존 주파수 도약 신호 분석은 FPGA에서 주로 수행되어

왔다. 하지만, 최근에 다양한 주파수 도약 시스템, 예측할 수 없는 신호 환경, 높은 가격 등의 이유로 FPGA에서 도약 신호 분석<sup>[4]</sup>이 한계에 부딪혔으며, 현재 PC 기반의 CUDA<sup>[5]</sup>를 이용한 병렬처리 소프트웨어로 대체되고 있는 추세이다. PC 기반의 도약 신호 분석 소프트웨어는 새로운 환경에 유연하게 대처가 가능하며, 저렴한 가격의 기술이지만 느린 처리 속도에 대한 문제점을 지니고 있었다. 본 논문에서는 주파수 도약 신호 분석 기법을 하드웨어 기반에서 소프트웨어 기반으로 설계하였으며, 단일 처리 방식에서 병렬 처리 방식으로 개선하였다. 기존 연구

「이 연구는 국방과학연구소의 지원 및 관리로 수행되었습니다.」

LIG 넥스원 전자전 연구센터(Electronic Warfare R&D Lab, LIG Nex1)

· Manuscript received November 18, 2013 ; Revised December 19, 2013 ; Accepted January 13, 2014. (ID No. 20131118-114)

· Corresponding Author: Kwang-Yong Lee (e-mail : kwangyong.lee@lignex1.com)

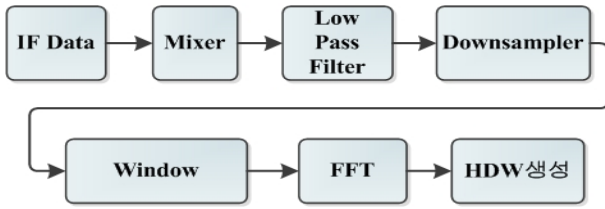


그림 1. 주파수 도약 신호 분석을 위한 블록도  
Fig. 1. Block diagram for analysis of FH.

에서는 입력 데이터로 광대역 수신기로부터 수신한 RF 신호를 IF 데이터로 변환하여 사용하고, PC 기반의 분석 컴퓨터에서 주파수 도약 신호를 병렬처리 기법을 적용하여 분석하는 방안을 제시하였다.

입력된 IF 데이터는 Digital Down Converter(DDC) 과정을 수행한 후 I/Q 데이터로 변환된다. 변환된 I/Q 데이터는 FFT 과정을 수행한 후 주파수 도약 신호 제원 추정에 필요한 Hopping Description Word(HDW)를 생성한다. 그림 1은 주파수 도약 신호를 분석하는 과정을 블록도로 표현하였다.

분석에 많은 시간이 소요되는 광대역 DDC, FFT 과정의 성능을 향상시키기 위해 고속 병렬처리 기법 중 하나인 CUDA<sup>[5]</sup>를 활용하여 주파수 도약 신호 분석 알고리즘을 설계하였고, 각 단계별 성능 향상 효과와 주파수 도약 신호 분석 결과를 CPU와 GPU<sup>[5]</sup>로 구현하여 성능을 평가하였다.

## II. 주파수 도약 신호 분석 단계

주파수 도약 신호 분석을 수행하기 위해서는 그림 1의 단계적 설계가 필요하다. 각 단계를 크게 3단계로 분류하였으며, 설계 방안은 다음과 같다.

### 2-1 DDC

광대역 신호를 파일로부터 읽어들이어 가장 먼저 수행하는 과정은 Mixer이다. 광대역 신호와 중심주파수의 Sine, Cosine 신호 곱셈을 통하여 광대역 신호를 기저대역으로 낮추며, I/Q 데이터를 생성한다. 이를 식 (1)로 표현하였다.

$$\begin{aligned} y_{re}[i] &= x[i] \times \cos(2\pi f_c t) \\ y_{im}[i] &= x[i] \times \sin(2\pi f_c t) \end{aligned} \quad (1)$$

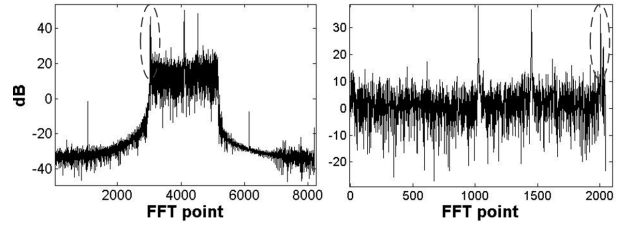


그림 2. 주파수 미러링(LPF 적용후, Downsampler 적용후)  
Fig. 2. Mirror frequency.

$y_{re}[i]$ 는 Mixer Real 성분,  $y_{im}[i]$ 는 Mixer의 Image 성분,  $x[i]$ 는 A/D 데이터,  $f_c$ 는 중심 주파수,  $t$ 는 시간 인덱스를 의미한다.

Mixer를 수행한 다음, 저대역의 주파수를 통과시키고, 차단주파수 이상의 주파수를 감쇄시키기 위하여 Low Pass Filter(LPF)를 수행한다. 식 (2)의

$$y[i] = \sum_{j=0}^M x[i] \times h[j-i] \quad (2)$$

$x[i]$ 는 입력 신호,  $h[j-i]$  필터계수,  $M$ 은 필터 길이이다.

LPF 설계시 Downsampling Rate를 고려하여 설계하였다. LPF의 차단 주파수는 Outband에 존재하는 신호가 Inband로 들어오게 되는 미러링 현상을 방지하기 위하여 Downsampling 수행 이후에 탐지할 수 있는 주파수 대역보다 작게 설계하였다.

LPF 이후에 Downsampler를 거친다. Sampling Rate를 낮추어 DDC 과정 이후 알고리즘의 데이터 처리량을 줄여준다.

### 2-2 FFT

DDC 과정을 수행한 다음, Side-lobes를 제거하기 위하여 Window 과정을 수행한다. 본 논문에서는 Hamming Window를 적용하였다.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right), 0 \leq n \leq N \quad (3)$$

식 (3)에서  $N$ 은 Window Length이다.

Hamming Window 과정을 수행한 후, Short-Time FFT를 수행한다. FFT 과정 이후에 FFT Shift 연산과 스펙트럼

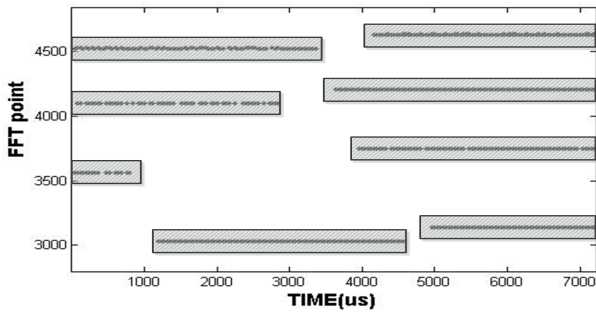


그림 3. HDW 생성  
Fig. 3. HDW creation.

생성을 위한 절대값 연산을 수행한다. 절대값 연산 시, 수행 시간 소모가 큰 Sqrt를 사용하지 않았다. 절대값 연산을 수행한 데이터는 데시벨 스케일로 변환한다. 그림 3 블록안 점선은 FFT 수행 결과를 표시하였다.

### 2-3 HDW

주파수 도약 신호 판별을 위해서는 신호 유지시간이 사용된다. FFT 데이터를 이용하여 신호를 추출하려면 각 홉마다 단일 홉 내 FFT Point 비교 빈도수가 많게 되므로, 미리 각 홉에 대한 펄스 형태로 신호를 간략화하여 도약 신호 분석 속도를 향상시키기 위해 HDW를 생성하는 방법을 모색하였다.

HDW는 주파수, 시작시간, 유지시간, 신호 세기 정보가 담겨 있으며, 이후 신호 추출 시 사용된다. 그림 3의 네모 블록은 FFT 인덱스 정보를 이용하여 HDW로 생성한 사진이다.

## III. 성능 평가

주파수 도약 신호 분석을 위한 각 단계별 설계는 병렬처리 기법 중 하나인 CUDA를 사용하여 설계 및 구현하였다. HDW 생성의 경우는 CPU에서 수행하였다. 각 신호처리 단계별 병렬처리 설계 기법은 다음과 같다.

### 3-1 DDC에 대한 병렬처리

광대역 A/D 데이터로부터 FFT Point 수만큼 Thread를 생성하여 병렬처리 연산을 수행하였다. 그림 4는 FFT Point 수를 8,192로 설정하였고, 8,192개의 Thread를 생성

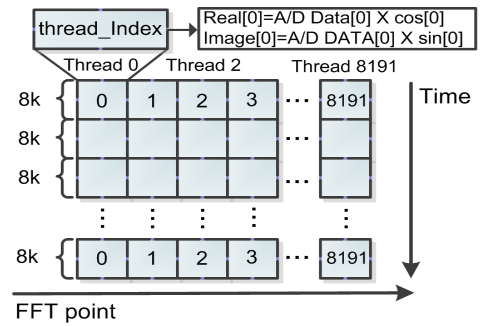


그림 4. Mixer 병렬처리 설계  
Fig. 4. Mixer parallel processing design.

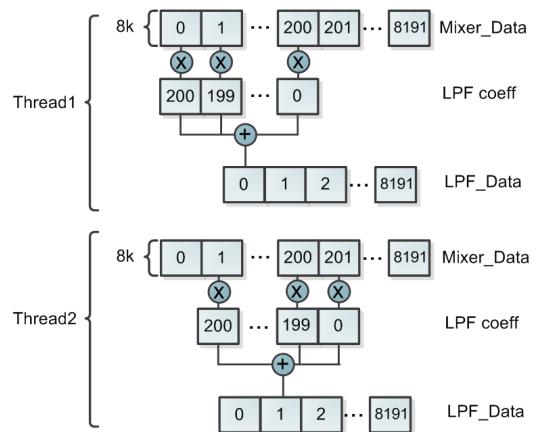


그림 5. LPF 병렬처리 설계  
Fig. 5. LPF parallel processing design.

하였다. 각 Thread는 해당 A/D 데이터에 Mixer를 수행한다.

LPF의 병렬처리 설계로 연산 수행 속도를 향상시키기 위하여 그림 5와 같이 FFT Point 수만큼 Thread를 생성하였다.

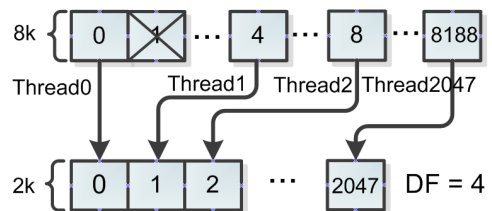


그림 6. Downsampler 병렬처리 설계  
Fig. 6. Downsampler parallel processing design.

표 1. DF에 따른 성능 측정(ms)

Tabel 1. Performance evaluation at different DF.

DF	2	4	8
속도(ms)	95	150	235

Downsampler의 병렬처리 설계 방안으로 그림 6과 같이 FFT Point 수를 Decimation Factor(DF)로 나눈 수만큼 Thread를 생성하였다. DF의 변화에 따른 성능 측정 결과는 표 1과 같다. DF가 증가될수록 Thread 수가 감소하기 때문에 처리 속도가 느려짐을 볼 수 있다. 병렬처리 특성상 많은 데이터를 묶어서 처리할 경우 성능이 향상된다.

### 3-2 FFT에 대한 병렬처리

Window, ABS, Logscale 과정은 Mixer에서 적용한 방법과 동일하게 병렬처리를 수행하였다. 다만, Downsampler을 수행하였기 때문에 FFT Point 수를 DF로 나눈 수만큼 Thread를 생성하였다.

FFT 과정은 CUFFT Library<sup>[5]</sup>를 사용하였으며, Complex FFT를 수행하였다. FFT Point 수의 변화에 따른 성능 측정 결과는 표 2와 같다. 표 2는 동일 용량에 대한 FFT 처리결과이며, 2,048 Point 수로 처리할 경우 처리 횟수가(frame수)가 증가한다. 8,192 Point 수로 처리할 경우, 처리 횟수(frame수)가 감소하므로 GPU Core 점유율(Core 사용개수)이 증가하여<sup>[5]</sup> 처리 속도가 향상됨을 볼 수 있다. FFT Point 수가 높아지면 처리 속도와 주파수 해상도는 증가하지만, 시간해상도가 낮아지는 트레이드오프 관계가 있다.

성능평가 시 사용된 CPU는 i5-3470 Quad Core를 사용하였고, GPU<sup>[5]</sup>는 GTX560을 사용하였다. 각 단계별 성능 측정 결과는 표 3과 같다.

표 2. FFT Point 수에 따른 성능 측정(ms)

Tabel 2. Performance evaluation at different FFT point.

FFT point 수	2,048	4,096	8,192
파일용량(MB)	464	464	464
Frame	237,658	118,784	59,392
속도(ms)	4.221	1.468	1.201

표 3. 단계별 성능 측정(ms)

Tabel 3. Performance evaluation at each step.

항목	단일처리	병렬처리
File read	8,500	8,500
Memcpy(CPU to GPU)	-	537
Mixer	2,375	2,003
LPF	354,583	31,762
Downsampler(DF: 4)	425	150
Window	2,547	1,500
FFT(Point 수: 2,048) ABS/Logscale	39,156	2,424
Memcpy(GPU to CPU)	-	888
Total(ms)	407,586	47,764

## IV. 결 론

본 논문에서는 고속 병렬처리 기법인 CUDA를 사용하여 주파수 도약 신호 분석 설계 방법을 제안한다. 전처리 과정인 DDC, FFT 과정을 병렬처리로 설계하여 CPU와 비교해 연산 수행 속도 면에서 CPU 대비 약 8.53배 성능이 향상됨을 확인하였다. 또한, HDW 수행과정을 통하여 주파수 도약 신호를 판별할 수 있음을 확인할 수 있었다.

주파수 도약 신호 분석 시, 본 연구의 병렬처리 설계 기법을 활용하여 성능 평가가 가능할 것이며, 향후 신호 분석의 고속 병렬처리 설계 연구에 활용이 가능할 것이다.

## References

- [1] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, Rockville, MA: Computer Science, 1985.
- [2] Robert C. Dixon, *Spread Spectrum Systems with Commercial Applications, 3rd Edition*, John Wiley & Sons, Inc., 1994.
- [3] Bernard Sklar, *Digital Communications, 2nd Edition*, 2004.
- [4] 정효진 et al., "FPGA 기반의 다중주파수 도약 신호 실시간 추적 방안에 관한 연구", 한국통신학회 학술대회 논문집, pp. 304-305, 2011년.
- [5] Jason Sanders, Edward Kandrot, *CUDA by Example*, Addison-Wesley, 2010.