

GPU 기반 주파수 영역 다중 채널 광대역 신호처리부 설계

Design of GPU-Based Frequency Domain Multi-Channel Wideband Signal Processing Unit

이현휘* · 박경태* · 윤현철* · 이광용*

Hyeon-Hwi Lee · Kyung-Tae Park · Hyun-Chul Yoon · Kwang-yong Lee

요 약

National Instrument사 USRP(universal software radio peripheral) 제품과 최신 계측기 등 상용 제품에서는 신호를 수신하여 분석하는 기능을 제공한다. 이러한 시스템에서는 대상 신호의 특성이 매우 다양하기 때문에 기본적으로 광대역 신호부터 협대역 신호를 수신할 수 있도록 설계하고, 필요에 따라서 원하는 신호를 추출하여 상세하게 분석할 수 있는 기능을 제공한다. 과거 이러한 기능을 실시간성의 이유로 FPGA(field programmable gate array), DSP(digital signal processor) 등의 하드웨어 형태로 구현하였다. 하지만 소프트웨어 기술의 발달에 힘입어 수집 시스템의 신호처리부를 소프트웨어로 대체할 수 있었다. 기존 GPU 기반 신호처리부는 연산량과 메모리 사용량이 과도하여 소수의 채널의 실시간 처리가 가능하였다. 이러한 한계점을 극복하기 위하여 주파수 영역 신호처리를 적용하였고, 연산량과 메모리 사용량을 획기적으로 감소시켰다. 그 결과, 최대 수 백 채널의 실시간 처리가 가능함을 확인하였다. 본 논문에서는 고가의 하드웨어 장비보다는 CPU, GPU 프로세서(processor)와 같은 프로그램 가능한 장비를 이용하여 다중 채널 광대역 신호처리부를 설계 구현하였다.

Abstract

Commercial products, such as the National Instrument's USRP (universal software radio peripheral) products and the latest instruments, provide functions to receive and analyze signals. Because the characteristics of the target signal are diverse, these systems are designed to receive a wideband signal to a narrowband signal and provide functions for extracting a desired signal and analyzing it in detail if necessary. In the past, these functions were implemented in the form of hardware, such as FPGA (field programmable gate array) and DSP(digital signal processor), for real-time processes. However, owing to the development of software technology, it is possible to replace the signal-processing part of the acquisition system with software. The existing GPU-based signal processing unit is capable of processing a few channel only in real-time owing to its excessive computation and memory usage. To overcome this limitation, frequency-domain signal processing is applied and the computation time and memory usage are considerably reduced. Thus, real-time processing of hundreds of channels is possible. In this study, a multi-channel wideband signal processing unit is designed and implemented using programmable equipment, such as CPU and GPU processors, rather than expensive hardware equipment.

Key words: GPU, SDR(Software Defined Radio), CUDA, SRC, Parallel Processing

LIG넥스원 전자전연구소(Electronic Warfare R&D Center, LIG Nex1 Co., Ltd.)

· Manuscript received January 22, 2022 ; Revised February 8, 2022 ; Accepted February 20, 2022. (ID No. 20220122-008)

· Corresponding Author: Hyeon-Hwi Lee (e-mail: hyeonhwi.lee2@lignex1.com)

I. 서 론

전자전 환경에서 신호 수집 시스템은 광범위한 주파수 대역을 감시하며, 동시에 미상 신호를 수집하고 분석한다. 그림 1은 전자전 신호 수집 시스템을 축약하여 RF(radio frequency) 신호수신기와 RF 신호처리기로 나타냈다.

안테나에서 수신한 RF 신호를 RF 신호수신기에서 입력받아 IF 신호로 변환 후 디지털 데이터로 양자화한다. RF 신호처리기에서는 ADC(analog digital converter) data 신호 처리 및 분석 후 신호정보 데이터를 생성한다. 여기서 신호정보는 탑재되는 알고리즘에 따라 변조방식, 심볼율, PDW(pulse description word), HDW(hopping description word), 복조데이터 등이 될 수 있다.

그림 2의 RF 신호수신기는 넓은 주파수 범위의 신호를 수신하기 위해 슈퍼헤테로다인 구조로 주로 구성되며, 안테나, RF front end, mixer, local oscillator, ADC로 구성된다.

RF front end는 각종 filter, LNA(low noise amplifier), Switch가 주요부품이며, 수신감도를 높이기 위한 RF 회로로 구성한다. Mixer에서는 local oscillator에서 생성한 관심대역의 동조 주파수를 이용하여 RF신호가 IF 주파수로 하향 변환한다. ADC에서는 넓은 동적범위를 갖는 양자화 bit를 결정하여 디지털 데이터로 변환한다.

RF 신호처리기는 광대역 스펙트럼을 감시하며, 탐지되는 신호를 저장 및 분석해야 한다. 기존 RF 신호처리기 Platform은 광대역 신호데이터 수신과 실시간 신호처리

성능 문제로 하드웨어에서 주로 처리하였다. RF 신호처리기는 다음의 성능을 포함하여 복잡한 DDC(digital down converter) architecture가 필요했다.

1. Real-time processing 능력
2. 다중 채널 협대역 변환 능력(최대 250)
3. 높은 decimation factor(최대 3,072)
4. 불요 신호 억압 능력
5. Pre-buffering 기능

이러한 성능이 발휘되도록 하드웨어를 구성하기 위해서는 복잡한 로직 구현에 대한 위험도가 높다. 그리고 채널을 추가할수록 추가하는 채널당 하드웨어 비용이 기하급수적으로 증가하였다.

RF 신호처리기는 그림 3과 같이 SW modularity를 높이기 위하여 하부 광대역 신호수신부, 광대역 신호탐지부, 광대역 신호처리부와 협대역 신호분석부를 각각 독립 프로세스로 구성하였다.

기존 연구^[1]에서는 광대역 ADC data를 PC platform에서 수신하기가 어려웠다. 그래서 digitizer board를 PC에 탑재하여 ADC data를 PCI-E interface로 획득하였다. 10 G ethernet interface 적용으로 PC에서 고속의 ADC 데이터를 수신할 수 있게 되었다.

광대역 신호처리부의 최대 분석 대역폭은 80 MHz이다. 80 MHz는 4개의 부 채널로 각각 나뉘어져 있다. 각 부 채널 샘플링률은 30.72 MHz이고 Usable bandwidth는 20 MHz이다. Data throughput은 식 (1)과 같이 계산한다. 여기서 샘플링률은 30.72 MHz, ADC bit는 16, 부 채널은 4개이며 complex data를 수신하므로 2를 곱셈한다.

$$\text{Data throughput} = \text{Sampling rate} \times \text{ADC bit} \times \text{Sub channel} \times 2 \quad (1)$$

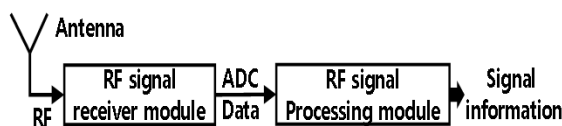


그림 1. 전자전 신호 수집 시스템
Fig. 1. Electronic warfare signal acquisition system.

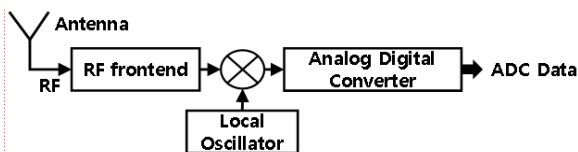


그림 2. RF 신호수신기
Fig. 2. RF signal receiver module.

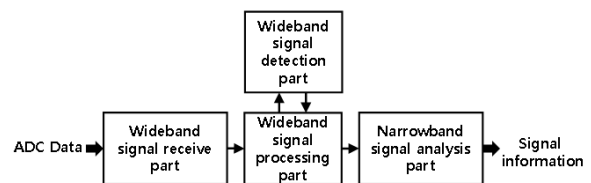


그림 3. RF 신호처리기
Fig. 3. RF signal processing module.

광대역 신호수신부는 식 (1) 계산 결과, 약 4 Gbps 속도로 고속 전송되는 광대역 신호를 실시간으로 수신 가능하다. Ethernet 프로토콜 중 UDP를 사용하였으며 광대역 신호처리부로 데이터 전송은 프로세스 간 통신 기법 중 한 종류인 공유 메모리(shared memory)를 사용하였다. UDP에서 data를 수신하여 버퍼에 넣는 thread와 공유 메모리에 데이터를 이동하는 thread로 구성된다. ADC Data 수신 시 packet loss가 발생하면 신호가 왜곡된다. Network interface card 속성(jumbo packet, 송·수신 버퍼 크기 등)을 tuning 후 UDP datagram 크기별 packet loss를 측정하였다. 8 k(8,192 Byte) 이상에서 packet loss가 발생하지 않았다. 또한, 광대역 신호 수신 프로세스가 단일 Core를 할당 받고 CPU 우선순위를 가장 높여야 소프트웨어 부하 시에도 packet loss가 발생하지 않는다.

광대역 신호탐지부는 광대역 신호처리부에서 생성하는 광대역 스펙트럼을 이용하여 미상신호를 탐지한다. 광대역 스펙트럼은 설정된 주기마다 광대역 신호처리부에서 광대역 신호탐지부로 전송한다. 신호 ID, 주파수와 대역폭을 탐지하여 광대역 신호처리부의 DDC 입력으로 사용한다.

협대역 신호분석부 알고리즘은 시험 과정에서 수시로 변경된다. 광대역 신호처리부에서 협대역 데이터를 file로 저장 후 저장된 파일 위치와 정보를 송신한다. 협대역 신호분석부는 협대역 데이터를 읽고, 신호인식, 신호 복조 과정을 수행한다. File 기반 통신은 메모리 통신보다 중복하여 file I/O를 한다는 단점이 있다. 하지만 광대역 신호처리부와 의 응집도를 낮추고, 향후 신호분석부 알고리즘 reconfigurability를 높이기 위하여 file로 통신하였다.

광대역 신호처리부 소프트웨어 설계에서 아키텍처를 어떻게 설계할 것인가를 결정하는 것은 매우 중요하다. 광대역 신호처리부 소프트웨어 아키텍처는 광대역 신호수신부, 광대역 신호탐지부, 광대역 신호분석부와 의 연동 방법을 고려해야 한다. 그리고 고속 병렬 신호처리와 신호품질을 높일 수 있는 기술을 필요로 한다.

II. 기존연구

2-1 GPU 기반 주파수 영역 Convolution에 관한 연구

시간 영역에서 GPU를 이용하여 다중 채널의 DDC를

실시간으로 처리하기에는 성능적으로 한계에 부딪혔다. 성능 측정 결과 DDC 과정 중 LPF 과정에서 시간이 오래 소요되었다. LPF 과정의 시간을 단축하기 위하여 convolution 연산을 기존 시간 영역에서 주파수 영역으로 변경하였다. 시간 영역에서 convolution 연산은 주파수 영역에서 곱셈 연산과 동일하다^[2]. 식 (2)의 $x[n]$ 는 입력 신호, $h[k]$ 는 필터계수, $X[n]$ 은 $FT(x[n])$, $H[k]$ 는 $FT(h[k])$ 이다. FT는 Fourier transform이며 FT-1은 IFFT(inverse Fourier transform)이다.

$$y[n] = x[n] * h[k] = FT^{-1}(X[n]H[k]) \quad (2)$$

그림 4는 주파수 영역 convolution 흐름을 표현하였다. 무한히 들어오는 입력신호와 유한한 필터계수와의 linear convolution을 수행하기 위하여 IFFT 이후 overlap-add^[3] 혹은 overlap-save^[4] 과정이 필요하다.

GPU 기반 주파수 영역 convolution에 관한 최근 연구가 진행되었다. 참고문헌 [5]에서 제안한 SM-OLS 연구를 DDC 과정 중 LPF에 그림 5와 같이 적용하였다. 적용 결과 연구에서 제시한 convolution 연산 실행시간(execution time) 결과와 유사한 성능을 확인하였다. NCO(Numerically-Controlled Oscillator) complex mixer와 downsampler는 기존 연구 결과물을 사용하였다^[6].

표 1은 그림 5 DDC 실행시간 결과이다. 입력 샘플링률

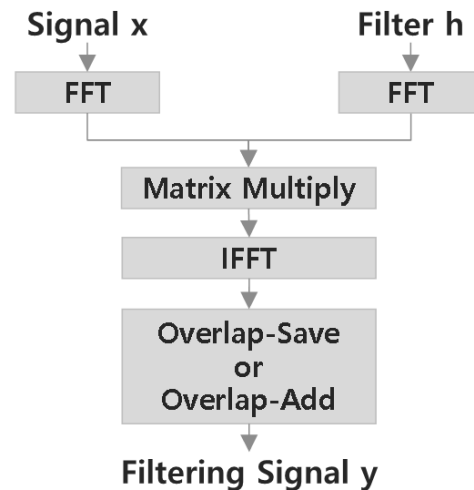


그림 4. 주파수 영역 convolution

Fig. 4. Frequency domain convolution.

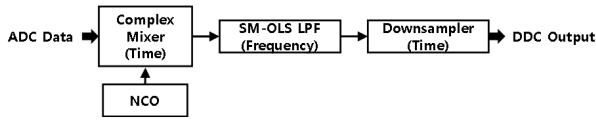


그림 5. 주파수 영역 convolution을 이용한 DDC
Fig. 5. DDC using frequency domain convolution.

표 1. 주파수 영역 convolution을 이용한 DDC 성능
Tabel 1. DDC using frequency domain convolution performance.

Channel	Execution time [sec]	GPU usage [%]	GPU RAM [%]
4	0.078	8	5
8	0.088	8	11
16	0.103	15	22
32	0.122	27	45
64	0.178	39	91

은 30.72 MHz이고, 출력 샘플링률은 64 kHz이다. 1초 분량의 ADC 데이터에서 협대역 데이터로 변환하는 시간을 측정하였다. 여기서 채널은 동시에 처리한 협대역 채널 개수이다. DDC 실행시간은 DDC 처리 thread 분기 후 최종 thread의 종료시점까지 소요시간을 측정하였다. GPU usage는 DDC 실행시간 중 GPU 최대 사용률이고, GPU RAM은 GPU 최대 메모리 사용량이다.

64개 채널부터는 GPU RAM 사용률이 90 %를 초과하였다. 250개 다중 채널 DDC를 수행하기에는 한계에 부딪혔다. 광대역 신호탐지부에서 수신하는 각각 서로 다른 주파수를 이용하여 탐지된 신호를 baseband로 옮긴다. complex mixer를 통과하면 채널 수 만큼 동일한 ADC data가 생성되고, 이를 저장할 메모리 공간이 필요한 단점이 존재한다. Complex mixer는 Sine과 Cosine 곱셈기로 구성되어 있고, ADC data 샘플 수만큼 연산을 수행해야 한다. 기존 시간영역 complex mixer를 연산 횟수와 메모리 측면에서 개선할 방안이 필요하다.

2.2 주파수 영역 Resampler에 관한 연구

기존 시간영역 SRC(sample rate conversion)는 입력 샘플링률 대비 출력 샘플링률 간 up/down 정수 인자를 결정

한다. 그 이후 결정된 인자로 interpolation과 decimation을 통한 resampling을 수행하였다. 하지만 시간 영역 SRC는 계산 복잡도가 높고, 실행 시간이 길다. 또한 입력 대비 출력 샘플링률이 정수가 아닌 분수가 나올 경우를 대비할 수 없고, 무한히 들어오는 입력신호에 대한 방안을 강구해야 했다.

주파수 영역 SRC는 참고문헌 [7]에서 설명하였다. 해당 문헌에서는 FFT 이후 몇 개 원소를 단순히 제거/추가하여 샘플링률을 감소/증가시켰다. 효율적인 FFT/IFFT 알고리즘이 결합된다면 계산 복잡도를 낮출 수 있고, 실행 시간을 단축할 수 있는 연구 결과이다. 게다가 overlapping 기술을 이용하여 입력 신호가 무한히 들어올 경우도 정확도를 향상시켰다.

III. 제안하는 광대역 신호처리부

3-1 주파수 영역 DDS Resampler를 이용한 DDC

그림 6은 제안하는 DDC 구조이다. 제안하는 DDC의 장점 중 첫 번째는 IFFT 연산량이 DF(decimation factor)만큼 줄어든다. 그 이유는 DDS(direct digital synthesis) resampler에서 입력신호 중 저장 신호 대역폭만큼 원하는 신호를 resampling하기 때문이다. 기존 주파수 영역 convolution 과정은 필터링된 전체 데이터를 획득하는 데 반하여 DDS resampler는 IFFT 수행 전 입력신호를 decimation하여 출력한다. 주파수 영역 convolution 마지막 과정의 IFFT는 전체 데이터가 입력되지만, 제안하는 DDC

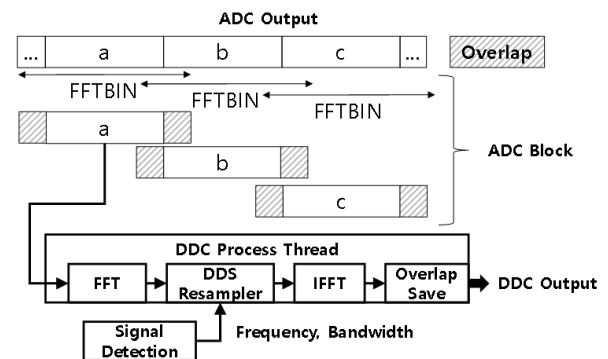


그림 6. 제안하는 DDC 구조
Fig. 6. Proposed DDC architecture.

는 DDS resampler에서 줄어든 샘플만큼 IFFT가 수행된다. 결국 DDS resampler의 출력은 decimation 결과이기 때문에 그림 4 주파수 영역 convolution 중 filter FFT 과정과 matrix multiply 연산을 수행할 필요가 없고 그림 5 downsampler 연산이 필요 없다.

제안하는 DDC의 장점 중 두 번째는 저장 채널에 따라 ADC 데이터만큼 증가하는 메모리 사용량을 감소시킨다. 그 이유는 DDS resampler는 입력 신호 중 탐지된 신호의 주파수 위치를 발췌하기 때문이다. 그림 5 기존 complex mixer 과정은 LPF 입력 전 각 탐지된 주파수를 baseband로 이동시킨다. Complex mixer를 통과하게 되면 신호탐지 주파수별 baseband로 이동한 ADC Data를 담은 메모리가 필요하다. DDS resampler를 적용하면 그림 5 complex mixer 과정이 필요 없다. 결국 채널 별 Sine과 Cosine 곱셈기를 통과한 ADC Data 메모리 양만큼 절약된다.

첫 번째 단계는 FFT이다. 들어오는 입력신호는 설정된 FFTBIN에 맞춰 ADC block으로 분할하여 입력된다. Pointer 위치 이동으로 overlap을 수행하였다. 개별 ADC block은 이전 block의 이후 block의 일부분을 포함하여 한 개의 ADC block으로 구성된다. 입력된 ADC block은 FFT를 거쳐 FFT data를 산출한다. FFT 알고리즘은 NVIDIA사가 제공한 cuFFT API를 사용하였다. FFT 알고리즘 이후 FFTShift를 수행하였다. FFTShift 연산을 수행한 이유는 DDS resampler 과정에서 간편하게 복사를 수행하기 위함이다. FFTShift 연산을 수행하지 않고 DDS resampler에서 신호 위치별 복사연산 횟수가 늘어난다.

두 번째 단계는 DDS resampler이다. DDS resampler는 memory 복사 명령으로 수행할 수 있다. Memory 복사 명령을 수행하기 위하여 복사할 샘플개수와 복사 위치를 결정하는 것이 필요하다. 복사할 샘플 개수(copysample)를 식 (3)과 같이 계산한다. 복사할 위치(copyposition)를 식 (4)와 같이 계산한다. 여기서 BW는 신호탐지부에서 수신한 탐지 대역폭 정보이고, F_c 는 탐지 중심주파수이다. Sample 개수는 ADC block의 샘플 개수이며 FFTBIN과 동일하다. F_s 는 ADC data의 샘플링률이다. DDS resampler 과정은 식 (4)번에 의하여 결정된 메모리 위치에서 식 (3)에 의하여 결정된 데이터 메모리 양만큼 복사하는 명령을 수행한다.

$$\text{CopySample} = (\text{Sample number} \times \text{BW} / F_s) \quad (3)$$

$$\text{CopyPosition} = ((F_c + F_s / 2) / F_s) \times \text{Sample number} - \text{CopySample} / 2 \quad (4)$$

세 번째 단계는 IFFT이다. 첫 번째 과정에 대비하여 DDS resampler 결과 줄어든 샘플만큼 IFFT를 수행한다. IFFT 알고리즘은 첫 번째 과정과 동일하다. IFFT 단계까지 거치게 되면 협대역 데이터를 산출한다. 현재까지 FFT 후 DDS resampler를 거쳐 IFFT까지 신호처리 흐름을 그림 7로 표현하였다.

마지막 단계는 overlap save 과정이다. Overlap 길이는 ADC block의 1/16 이상으로 수행할 때 각 ADC block 간 왜곡이 발생하지 않았다. Overlap save는 별도의 메모리 공간에 복사 명령으로 수행할 수 있다. ADC block의 overlap 길이(Sample 개수/16)를 DDC에 입력했을 때 DDC 결과의 overlap 샘플 수를 계산한다. DDC 결과의 Overlap 샘플 수(overlapsample)를 식 (5)로 계산한다. 식 (5)에서 sample 개수는 ADC block의 샘플 개수이고, BW는 신호 탐지부에서 수신한 탐지 대역폭 정보, F_s 는 ADC data의 샘플링률이다.

$$\text{OverlapSample} = ((\text{Sample 개수} / 16) \times \text{BW} / F_s) \quad (5)$$

표 2는 제안하는 DDC 구조의 pseudo-code이다.

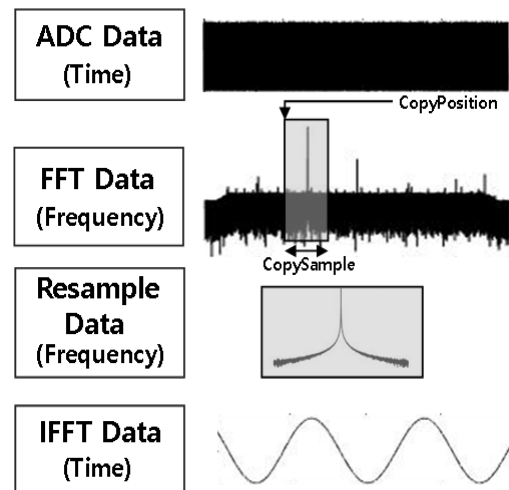


그림 7. DDS resampler를 이용한 신호처리
Fig. 7. Signal processing using DDS resampler.

표 2. 제안하는 DDC 구조 pseudo-code

Tabel 2. Pseudo-code for proposed DDC architecture.

```

FFTData = ForwardFFT(ADCData) // 1. Forward FFT of ADC Data
// 2. DDS Resampler
MemCpy(ResampleData, &FFTData[CopyPosition], CopySample)
// 3. Inverse FFT of DDS Resampler Output
IFFTData = InverseFFT(ResampleData);
// 4. Overlap Save
MemCpy(DDCOutput, &IFFTData[OverlapSample], CopySample
-OverlapSample)

```

3-2 제안하는 광대역 신호처리부 구조

그림 8은 제안하는 광대역 신호처리부 구조이다. ADC data는 8 k UDP diagram 단위만큼 packet 데이터가 수신된다. Packet 데이터 수신 후 설정된 FFTBIN에 맞춰 처리단 위까지 쌓이면 공유 메모리로 데이터를 복사한다. 공유 메모리는 광대역 신호수신부와 광대역 신호처리부와 ADC data를 공유하기 위한 메모리 영역이다. 광대역 신호처리부에서는 공유 메모리에 데이터가 쌓이게 되면 FFT를 수행 후 내부 GPU global memory에 적재한다. GPU global memory는 DDC 처리 thread 전부에서 접근 가능한 메모리 영역이다. 만약 FFT를 개별 DDC 처리 thread에서 수행할 경우, FFT 연산량이 채널 수에 비례하여 현저히 늘어난다. ADC data를 FFT 수행 후 공유 자원인 global memory에 적재하고, 개별 DDC 처리 thread에서 공통의 데이터를 참조한다면, 연산량을 대폭 감소할 수 있는 장점이 있다.

FFT를 거친 데이터는 광대역 신호탐지부로 설정된 주

기마다 전송한다. 광대역 신호탐지부는 임계치 이상의 신호를 탐지한 후 신규 신호 탐지 시 ID를 부여한다. 그리고 탐지된 신호의 주파수와 대역폭 정보를 광대역 신호처리부 DDC manager에 전달한다.

광대역 신호처리부 DDC manager에서는 광대역 신호탐지부에서 수신된 주파수와 대역폭 정보를 유틸리티 DDC thread를 할당한다. DDC thread 입력으로는 GPU global memory offset과 ID, 주파수와 대역폭이다. GPU global memory offset 적용 시 신호탐지 latency와 설정된 pre-buffering 만큼 offset 처리를 수행해야 한다. 개별 DDC 처리 thread는 global memory를 순회하면서 DDC 연산을 수행한다. 광대역 신호탐지부에서 수신되는 신호소멸 목록에는 ID가 있으며, ID와 매칭된 DDC 처리 thread를 중지한다.

Normalize 단계는 IFFT 샘플 수가 아닌 입력된 FFT 샘플 수로 수행해야 입력된 광대역 신호와 동일한 신호세기가 복원된다. 수신한 ADC data는 광대역수신기의 gain이 반영되어 수신되므로 실제 RF 신호세기와는 오차가 발생한다. 그리고 광대역 신호처리부 자료형(data format)과 저장장치에 저장할 자료형의 차이로 발생하는 신호세기 오차가 발생하므로 협대역 데이터 저장 전 gain을 반영하였다.

IV. 시험 결과

본 장에서는 앞서 제안 및 설명한 광대역 신호처리부의 성능 평가를 수행하였다. 입력 신호원은 안테나 혹은

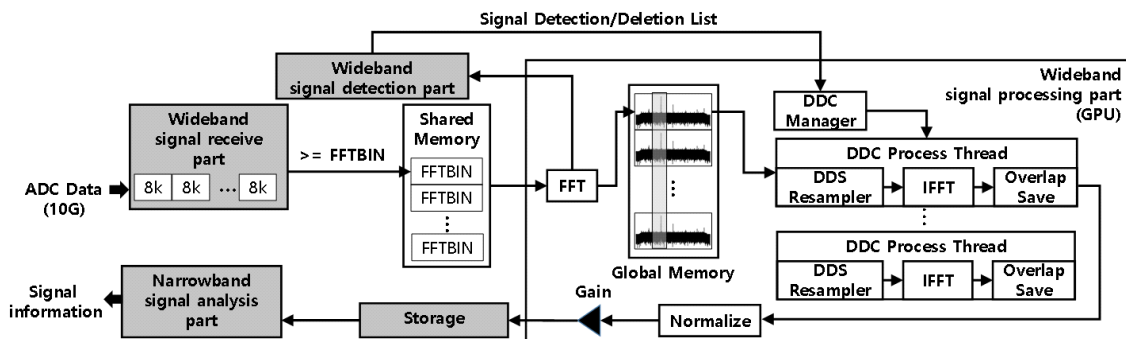


그림 8. 광대역 신호처리부 구조

Fig. 8. Wideband signal processing unit structure.

표 3. 하드웨어 사양

Tabel 3. Specification of hardware.

	CPU	GPU
Model	Intel(R) Xeon Gold 6246R	Tesla v100
Clock (GHz)	3.4	1.38
RAM (GB)	256	32
Core	6	5,120

신호발생기에서 모의한 신호이다. 안테나로부터 입력한 신호원은 상용 FM 대역이다. 신호발생기에서는 다중 tone을 발생하여 혼합한 CW 신호를 발생하였다. 저장한 광대역 데이터는 부 채널에 해당하는 30.72 MHz의 ADC 데이터이다. 표 3은 성능평가에 사용된 하드웨어 사양이다. File I/O는 NAS storage를 사용하였고, 실시간 저장 가능한 사양으로 RAID가 적용되었다.

4.1 신호처리 성능

표 4는 제안한 광대역 신호처리부의 성능 측정 자료이다. 표 1에서 제시한 자료와 동일한 환경에서 측정하였다. 기존연구와 비교 결과 연산량이 감소하여 GPU 사용률이 감소하였다. 또한 GPU RAM 사용률이 현저히 줄어든 것을 확인할 수 있다. 기존 연구에서는 GPU RAM 한계로 최대 64개까지 처리하였지만, 제안하는 광대역 신호처리부는 최대 256개까지 다중 채널 협대역 변환 능력을 보유한 것을 확인하였다. 적재된 ADC data를 밀리지 않고 처리할 수 있기 때문에 latency가 있는 실시간 처리 능력을 보유함을 확인하였다.

표 4. 제안하는 광대역 신호처리부 성능

Tabel 4. Proposed wideband signal processing unit performance.

Channel	Execution time [sec]	GPU usage [%]	GPU RAM [%]
4	0.052	9	3.93
8	0.057	8	3.94
16	0.062	10	3.95
32	0.093	10	3.98
64	0.146	10	4.03
128	0.340	9	4.12
256	0.703	11	4.28

4.2 신호처리 품질

제안하는 DDC 구조는 광대역데이터 중 일부 부분만 잘라내기 때문에 이상적인 rectangle filter의 성능을 발휘할 수 있다. 그림 9는 신호발생기에서 모의한 다중 CW를 발생하여 혼합한 광대역 신호에서 DF를 가변 하여 얻은 DDC 결과이다.

시간영역 DDC의 경우, 높은 샘플링률에서 낮은 샘플링률로 변환하기 위해서는 다단의 필터링 과정이 필요하다. 불요신호 억압 능력을 위하여 filter tab 길이를 높이면 계산복잡도가 증가하는 비례 관계에 있다. 그리고 filter의 squirt 문제로 filter edge 부분의 필터링 성능이 passband 영역만큼 억압도가 높지 않다. 시간영역 필터를 설계하는 방식에 따라 성능은 차이가 있겠지만, 혼재된 신호원을 완벽히 차단하면서 filter edge 부분의 필터링 성능까지 확보하기는 어렵다. 그림 10은 800 필터 차수 등리플 fir filter를 적용한 시간영역 DDC 성능이다. 인접 신호에 의한 불요신호도 존재하며 passband 신호가 filter edge 특성에 의하여 감소되었다.

안테나에서 수신받은 상용 FM 방송신호 음성청취로도 신호처리 품질을 확인하였다. 그림 11(a)는 광대역 신호의 중심주파수는 110 MHz이다. 그림 11(b)는 중심주파수 107.7 MHz, 탐지대역폭 200 kHz FM 신호를 DDC 처리한

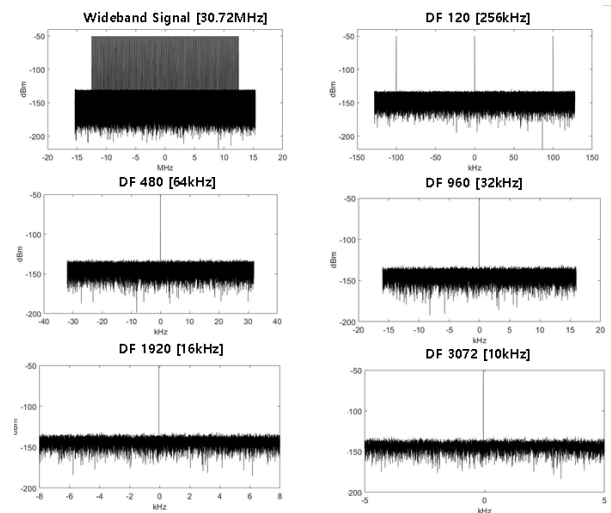


그림 9. DF를 가변한 DDC 결과

Fig. 9. DDC result with variable DF.

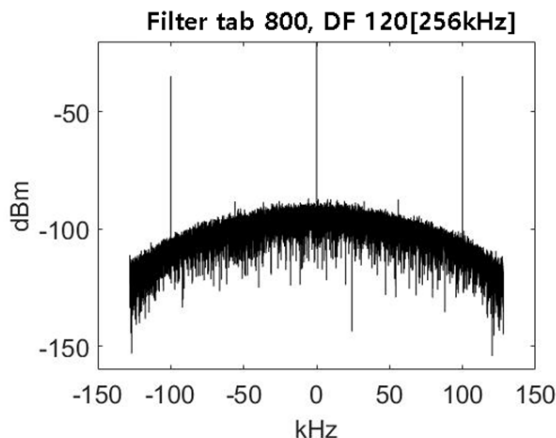


그림 10. 시간영역 DDC 성능
Fig. 10. Time domain DDC performance.

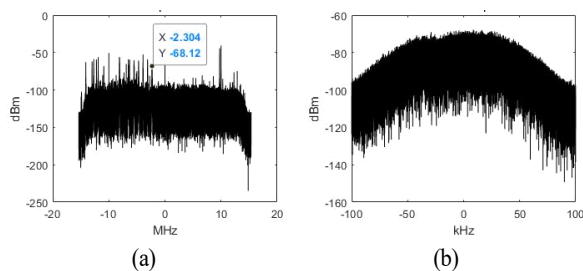


그림 11. FM 방송 DDC 결과
Fig. 11. FM broadcast DDC result.

결과이다.

V. 결 론

본 논문에서는 GPU 기반 주파수 영역 신호처리를 적용하여 다중 채널 실시간 처리가 가능함을 확인하였다. 기존 연구결과를 구현하였고, 성능을 비교하였다. 제안한 광대역 신호처리부는 기존 연구결과, 대비 연산량과 메모리 사용량은 감소시키며, 신호품질이 향상됨을 시험 결과를 통해 확인하였다.

상용제품은 수신 채널을 추가함에 따라 가격이 증가한다. 반면 GPU 기반 수신기는 상용제품군 대비 저렴하다. 상용제품에서 수신 채널을 추가하게 되면 최소 수 백만 원에서 수 천만 원까지 비용이 증가하게 된다. 본 논문

서 사용한 GPU는 천만 원 내외 비용으로 수신 채널을 수백 채널까지 구성할 수 있었다. 수신 채널을 수백 개 이상 구성해야할 수신 시스템이라면 상용제품으로 구성하는 것보다 GPU 기반 수신기를 적용하는 것이 가격 측면에서 큰 장점이다.

References

- [1] H. H. Lee, K. Y. Lee, S. Yun, Y. Park, and S. Kim, "Realtime wideband SW DDC using high-speed parallel processing," *The Journal of Korean Institute of Electromagnetic Engineering and Science*, vol. 25, no. 11, pp. 1135-1141, Nov. 2014.
- [2] T. G. Stockham Jr., "High-speed convolution and correlation," in *Proceedings of Spring Joint Computer Conference*, Boston, MA, Apr. 1966, pp. 229-233.
- [3] J. G. Proakis, D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Upper Saddle River, NJ, Prentice-Hall, pp. 430-433, 1996.
- [4] A. V. Oppenheim, R. W. Schaffer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ, Prentice-Hall, pp. 558-560, 1989.
- [5] K. Adamek, S. Dimoudi, M. Giles, and W. Armour, "GPU fast convolution via the overlap-and-save method in shared memory," *ACM Transactions on Architecture and Code Optimization*, vol. 17, no. 3, pp. 1-20, Sep. 2020.
- [6] H. C. Yoon, H. H. Lee, H. J. Kang, J. Y. Kim, B. H. Moon, and S. K. Kim, "Tuned save of DSSS signal using parallel programming skills of general purpose computing on graphics processing," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 45, no. 3, pp. 577-583, Mar. 2020.
- [7] G. Bi, S. K. Mitra, "Sampling rate conversion in the frequency domain [DSP tips and tricks]," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 140-144, May 2011.

이 현 휘 [LIG넥스원/선임연구원]

<https://orcid.org/0000-0001-5315-0625>



2013년 2월: 한동대학교 전산전자공학과
학사(공학사)
2013년~현재: LIG넥스원 전자전연구소
선임연구원
[주 관심분야] 병렬처리, 디지털 신호 처
리

윤 현 철 [LIG넥스원/선임연구원]

<https://orcid.org/0000-0001-8325-5620>



2010년 2월: 선문대학교 컴퓨터공학과 (공
학사)
2012년 2월: 한양대학교 전자컴퓨터통신
공학과 (공학석사)
2012년 1월~현재: LIG넥스원 전자전연구
소 선임연구원
[주 관심분야] 체계공학, 전자전신호처리,
GPGPU

박 경 태 [LIG넥스원/수석연구원]

<https://orcid.org/0000-0001-9284-7337>



1999년 2월: 고려대학교 산업공학 (공학
사)
2016년 8월: 아주대학교 IT융합 (공학석
사)
2002년 7월~현재: LIG넥스원 전자전연구
소 수석연구원
[주 관심분야] 소프트웨어 공학, 체계 성
능 분석, 체계 설계

이 광 용 [LIG넥스원/수석연구원]

<https://orcid.org/0000-0002-4540-8929>



2001년 2월: 전남대학교 전자공학과 (공학
사)
2021년 2월: 한양대학교 통신정보공학과
(공학석사)
2001년 7월~현재: LIG넥스원 수석연구원
[주 관심분야] 체계공학, 통신신호분석 등