

온칩 컨볼루션 가속기를 포함한 대칭적 버퍼 기반 액티브 노이즈 캔슬러의 경량화된 FPGA 구현

박승현¹ · 박대진^{2*}

Lightweight FPGA Implementation of Symmetric Buffer-based Active Noise Canceller with On-Chip Convolution Acceleration Units

Seunghyun Park¹ · Daejin Park^{2*}

¹Undergraduate Student, School of Electronics Engineering, Kyungpook National University, Daegu, 41566 Korea

^{2*}Associate Professor, School of Electronics Engineering, Kyungpook National University, Daegu, 41566 Korea

요약

처리지연이 적은 노이즈 캔슬러일수록 샘플링 주파수를 높일 수 있으므로 더 좋은 품질의 출력 신호를 얻을 수 있다. 단일 버퍼를 사용할 경우 프로세서가 입력된 데이터를 처리하는 동안 새로운 데이터를 버퍼에 쓰기가 불가능하므로 처리지연이 발생한다. 이러한 처리지연은 안티-노이즈와 출력 신호를 합성시킬 때 위상을 일치시키기 위한 추가적인 버퍼링 오버헤드를 발생시킨다. 본 논문에서는 대칭적 Even-Odd-buffer 구조를 사용하여 읽기와 쓰기 작업을 번갈아 가며 수행함으로써 처리지연을 최소화하고 처리속도를 높일 수 있는 가속기의 구조를 제안한다. 또한, Fast Fourier Transform 기반 노이즈 캔슬링과 적응 Least Mean Square 알고리즘을 사용한 노이즈 캔슬링의 구조적 차이를 비교한다. 그 결과로 대칭적 Even-Odd-buffer를 사용하였을 때 단일 버퍼 대비 처리지연이 29.2% 줄어들었다. 제안하는 대칭적 Even-Odd-buffer 구조는 다양한 노이즈 캔슬링 알고리즘에 적용될 수 있다는 장점이 있다.

ABSTRACT

As the noise canceler with a small processing delay increases the sampling frequency, a better-quality output can be obtained. For a single buffer, processing delay occurs because it is impossible to write new data while the processor is processing the data. When synthesizing with anti-noise and output signal, this processing delay creates additional buffering overhead to match the phase. In this paper, we propose an accelerator structure that minimizes processing delay and increases processing speed by alternately performing read and write operations using the Symmetric Even-Odd-buffer. In addition, we compare the structural differences between the two methods of noise cancellation (Fast Fourier Transform noise cancellation and adaptive Least Mean Square algorithm). As a result, using an Symmetric Even-Odd-buffer the processing delay was reduced by 29.2% compared to a single buffer. The proposed Symmetric Even-Odd-buffer structure has the advantage that it can be applied to various canceling algorithms.

키워드: 노이즈 캔슬링, 대칭적 Even-Odd-buffer, FFT, 적응 LMS 알고리즘, FPGA

Keywords: Noise cancellation, Symmetric Even-Odd-buffer, FFT, Adaptive LMS algorithm, FPGA

Received 20 September 2022, Revised 28 September 2022, Accepted 2 November 2022

* Corresponding Author Daejin Park (E-mail: boltanut@knu.ac.kr, Tel: +82-53-950-5548)

Associate Professor, School of Electronic and Electrical Engineering, Kyungpook National University, Daegu, 41566 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.11.1713>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

노이즈는 데이터의 전달과정 중 원하지 않는 신호가 더해져 전달되는 현상이다. 노이즈 현상이 심해지면 시스템의 신뢰성이 떨어지게 된다. 음향신호의 노이즈를 제거하기 위해서 음향기기의 설계과정에서 다양한 필터를 사용하여 신호를 깨끗하게 할 수 있다. 노이즈 필터링 기술이 발달하기 전에는 주파수 영역에서 FFT를 이용하여 노이즈의 주파수를 계산하고 대역 통과 필터를 사용하여 노이즈를 제거하였다. 이 방법은 데이터의 크기가 커질수록 필터링에 오랜 시간이 필요하고 연산량 또한 많아진다. 따라서 주파수 영역 필터링 방법은 실시간 처리에 적합하지 않다. 최근 Kalman 필터링 알고리즘을 기반으로 한 재귀적인 알고리즘들이 발달하였다 [1, 2]. 노이즈 신호와 입력 신호의 차이를 계산하여 비용함수가 최소로 되게 하는 필터의 계수를 찾아내는 적응 알고리즘을 통해 필터링할 수 있다. 적응 알고리즘은 Least-Mean-Square (LMS), Normalized Least-Mean-Square (NLMS), Recursive Least-Square (RLS) 알고리즘들이 잘 알려져 있다. 본 논문에서는 FFT 기반 노이즈 필터링과 LMS 알고리즘 노이즈 필터링 프로세서를 하드웨어 기술언어 (HDL)를 통해 Register Transfer Level (RTL) 합성해보고 두 필터링 방법의 구조적 차이에 대해 알아본다.

프로세스의 스케줄링은 특정 시간에 명령을 수행할 자원들의 역할과 프로세스를 나누는 것을 의미한다. 많은 프로세스가 한 명령을 수행하는 데 필요하므로 높은 효율을 위한 유동적인 프로세스의 분배가 이루어져야 한다 [3]. 노이즈 필터링 프로세서는 신호의 오차를 계산하고 필터의 계수를 수정할 때까지 시간 지연이 발생한다. 지연시간이 길수록 출력 신호와의 싱크를 맞추기 위해 버퍼를 추가하는 과정에서 많은 오버헤드와 전력 소모가 발생한다. 본 논문에서는 이중 버퍼링 기술을 사용하여 처리지연을 줄이는 방법을 고안하였다. 이중 버퍼는 두 개의 First In First Out (FIFO)로 구성되고 Analog-to-Digital Converter (ADC)에 의해 샘플링된 데이터를 메모리에 저장한다. 많은 데이터가 하나의 버퍼를 통해 입력된다면 필터링을 하기 위해 버퍼가 채워지기까지 기다려야 하므로 지연을 발생시키고 실시간 처리를 불가능하게 한다. 따라서 두 버퍼의 적절한 스케줄링을 통해 지연시간을 효과적으로 줄일 수 있다.

II. 노이즈 캔슬링 가속기 구조

본 장에서는 노이즈를 제거하는 하드웨어 가속기의 두 가지 구조에 대해 다룬다. 또한, 각 가속기의 장단점을 비교해 보았다.

2.1. FFT 노이즈 제거 알고리즘

마이크로 입력된 신호는 FFT 과정을 거쳐 주파수 스펙트럼으로 변환할 수 있다. 주파수 스펙트럼으로 변환된 데이터는 메모리에 저장되고 주파수별 진폭 값을 참조하여 노이즈를 처리하게 된다. 이때 프로세서는 어떤 주파수가 노이즈인지 알지 못하므로 노이즈 주파수, 진폭의 임계값을 정하고 소프트웨어 러닝 알고리즘을 사용하여 필터링할 주파수를 지정해주거나 학습시켜야 한다. 그림 1은 FFT를 사용하여 노이즈를 제거하는 구조에 대해 나타내고 있다. 3장에서 살펴볼 Symmetric Even-Odd-buffer 구조를 사용하여 샘플링을 하고 샘플링된 데이터를 FFT 연산하여 주파수 성분으로 분석한다. 노이즈로 확인된 주파수와 그 진폭을 알아내고 Look Up Table (LUT)을 통해 Finite Impulse Response (FIR) 필터의 계수를 결정한다. 그 계수를 FIR 필터에 넘겨서 노이즈 제거를 수행한다. 노이즈는 버퍼, FFT, FIR 필터 순서대로 처리되어야 하므로 파이프라인 구조를 사용하여 다음의 샘플링 데이터의 처리 효율을 높일 수 있다 [4].

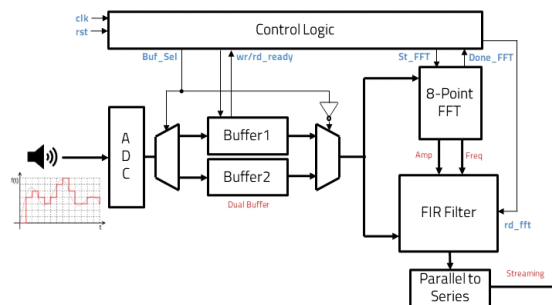


Fig. 1 FFT-based noise cancellation accelerator

2.2. LMS 노이즈 제거 알고리즘

최근에는 적응 알고리즘을 이용한 노이즈 캔슬링이 발달하고 있다. LMS 알고리즘은 적응 알고리즘 중 하나이며 하드웨어의 복잡도가 가장 낮으므로 실험을 위한 알고리즘으로 채택하였다. 그림 2는 LMS 알고리즘을 이용하여 설계한 노이즈 캔슬러의 하드웨어 다이어그램

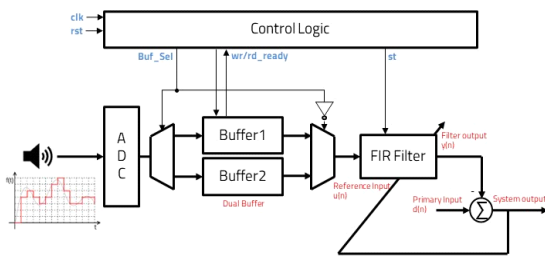


Fig. 2 LMS algorithm noise cancellation accelerator

램이다. 입력된 노이즈를 LMS 알고리즘을 통해 실제 음원과 비교하여 출력하는데 실시간으로 처리된다는 점과 필터의 계수들이 자동으로 갱신된다는 점에서 액티브 노이즈 캔슬러라고 한다. LMS 알고리즘은 입력된 신호와 필터의 출력과의 차이의 제곱 오차의 평균이 최소가 되게 하는 알고리즘이다. LMS 알고리즘 또한 FFT 기반 노이즈 제거 알고리즘과 같이 FIR 필터를 사용하므로 두 알고리즘의 성능 비교에서 이점이 있다. 하지만 필터의 계수를 계산하고 자동으로 업데이트 해주는 하드웨어가 추가되어야 하므로 구현 복잡도는 더 높다. 하지만 입력 샘플과 필터 출력값의 오차를 재귀적으로 계산하여 다른 참조변수가 없어도 노이즈 캔슬링을 할 수 있다 [5]. 노이즈의 파형을 변경하는 FIR 필터는 칩 내의 컨볼루션 유닛을 구성하고 있다. FIR 필터의 구조상 계산을 빠르게 하기 위해 Multiply and Accumulate(MAC) 연산기가 들어가고 그 연산기를 제어하는 컨볼루션 가속기가 내장되어 있다. 소프트웨어를 통해 컨볼루션을 계산하는 것보다 하드웨어 가속기로 구성하여 계산하는 것이 더욱 빠르기 때문에 노이즈 제거 알고리즘을 더욱 빠르게 계산할 수 있다.

III. 대칭적 Even-Odd-buffer

대칭적 (Symmetric) Even-Odd-buffer는 두 개의 FIFO를 사용하여 읽기와 쓰기 역할을 번갈아가며 수행하는 구조이다. 샘플링된 데이터를 실시간으로 처리하기 위해서는 처리할 준비가 된 버퍼가 항상 대기 중이어야 한다. 단일 버퍼를 사용할 경우 읽기와 쓰기 동작을 동시에 수행할 수 없으므로 데이터를 처리하는데 많은 클럭 사이클이 소요된다. 제안하는 Symmetric Even-Odd-buffer를 사용할 경우 읽기 전용 버퍼와 쓰기 전용

버퍼가 동작이 끝날 때마다 서로 바뀌기 때문에 데이터를 효율적으로 처리할 수 있다 [6].

Symmetric Even-Odd-buffer 구조에는 읽기 버퍼와 쓰기 버퍼를 구분하기 위해 파이프라인 구조를 사용하고 있다. 하나의 버퍼가 읽기 동작을 하고 있다면 다른 버퍼는 동기화되어 샘플링된 데이터들을 처리하게 된다. 프로세서에서 한 프레임의 노이즈 제거가 완료되었다면 완료를 나타내는 제어 신호를 통해 버퍼와 FFT 모듈 등 모든 데이터 처리 블록들을 동기화할 수 있게 된다. 또한, 제안하는 Even-Odd-buffer는 버퍼의 크기와 무관하게 동작하기 때문에 스위칭 동작이 빠르고 지연이 거의 발생하지 않는다. 그림 2는 Even-Odd-buffer의 구조와 제어 신호를 보여준다. 읽기/쓰기 시작 신호는 동기화를 위해 한 프레임의 처리가 완료되었을 때 방출이 된다. 또한, 오버플로우 현상을 막기 위해 버퍼의 크기를 나타내는 카운트 레지스터를 선언하였다. 읽기 전용 버퍼에는 카운트가 0이 될 때 읽기 동작을 수행할 수 있으므로 Ready 신호를 구현하였다. 반대로 쓰기 전용 버퍼에서는 데이터의 크기를 비교하여 처리하게 될 데이터가 온전히 들어있는지 확인 후 Ready 신호를 방출하게 된다. 모든 버퍼가 준비상태가 되었을 때 Finite State Machine (FSM) 이 데이터의 이동 단계로 넘어가게 된다 [7].

IV. 구현 및 실험

Even-Odd-buffer 구조는 Verilog-HDL을 이용하여 RTL 설계가 가능하다. 작성한 Even-odd-buffer 구조를 FFT 기반 노이즈 캔슬러와 LMS 알고리즘 노이즈 캔슬러에 각각 적용하여 노이즈 캔슬링 가속기를 설계한다. 설계한 RTL을 FPGA에 올려 수행시간, 전력 소모 결과를 분석한다.

Quartus II는 인텔사의 FPGA 설계 소프트웨어이다. Quartus II를 이용하여 Verilog-HDL로 설계한 하드웨어를 RTL로 합성할 수도 있고 그 성능을 분석하는 다양한 툴을 제공한다. 이 논문에서는 Quartus II의 Power analyzer 툴을 이용하여 전력을 비교하였다. ModelSim은 Verilog-HDL과 같은 하드웨어 기술언어를 사용하여 타이밍 시뮬레이션을 할 수 있는 프로그램이다. ModelSim을 사용하여 타겟 FPGA에서 프로그램을 수

행하는 데 어느 정도의 시간이 소요되는지 확인할 수 있다 [8, 9, 10].

4.1. Even-Odd-buffer 타이밍 분석

같은 사양을 가진 버퍼를 적용하여 하나의 버퍼를 사용한 데이터 전송 모델, Even-Odd-buffer를 사용한 데이터 전송 모델을 실험해 보았다. 데이터는 24bit 크기이고 버퍼는 16개의 주소를 가지고 있다. 단일 버퍼와 Even-Odd-buffer의 동작 비교를 위해서 데이터를 읽고 쓰는 작업을 하는 테스트 벤치로 실험을 진행하였다. 그림 3은 ModelSim 소프트웨어를 이용하여 비교한 단일 버퍼와 Even-Odd-buffer의 신호 파형을 나타낸다.

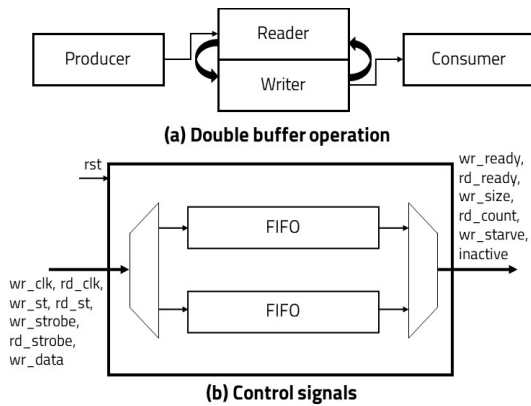


Fig. 3 Even-Odd-buffer structure

단일 버퍼의 경우 하나의 버퍼에서 입력과 출력 포트에 데이터가 순차적으로 들어온다. Even-Odd-buffer에서는 프레임의 싱크를 맞추기 위해 두 개의 프레임을 단위로 데이터 처리 알고리즘이 동작한다. 단일 버퍼의 경우 하나의 데이터를 받고 다음 데이터를 받을 때까지의 지연시간은 19 사이클로 측정되었다. 또한, 첫 번째 2프레임의 데이터를 받기까지의 시간은 74 사이클이며 다음 2프레임마다 처리 시간은 72 사이클이 소요되었다. Even-Odd-buffer의 지연시간은 11 사이클로 단일 버퍼 대비 42% 감소하였고, 첫 번째 2프레임을 받기까지 걸리는 시간은 89 사이클, 다음 2프레임마다 소요시간은 51 사이클이 소요되었다. 2프레임당 소요시간을 바탕으로 짝수 개의 데이터 프레임의 개수가 많아질 때마다 소요시간의 증가를 그림 4에 나타냈다.

4개의 프레임까지는 Even-Odd-buffer의 주소설정,

제어 신호 개수의 차이에 의해 Even-Odd-buffer의 오버헤드가 발생한다. 따라서 4프레임까지 단일 버퍼의 처리속도가 빠르지만, 그 이후로 2프레임당 소요시간이 더 작으므로 Even-Odd-buffer의 처리속도가 29.2% 더 빨라졌다. 실제 오디오의 경우, Rate size가 1,410kbps이고 200초인 파일의 크기는 282,000bit이고 프레임당 비트 크기로 나누면 734개의 프레임이 필요하다. 즉, 일반적인 음원 파일의 경우 제안한 Even-Odd-buffer의 사용이 유리하다.

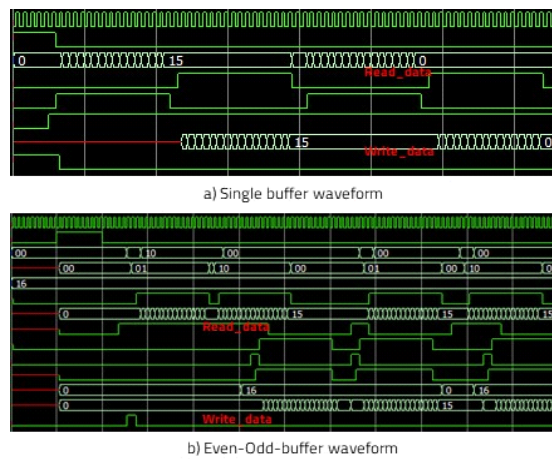


Fig. 4 Buffer's waveform

4.2. Even-Odd-buffer 성능 분석

데이터가 24bit이고, 16개의 데이터가 1프레임으로 이루어져 있어서 24bit로 이루어진 실제 음원을 메모리에서 불러들여 실험할 수 있다. 버퍼를 구동하는 클럭은 clock domain crossing 방법을 사용하여 읽기 전용, 쓰기 전용 버퍼의 동작을 제어할 수 있었다. 클럭이 분기되기 때문에 클럭의 싱크를 맞추기 위해 별도의 모듈이 필요하였다.

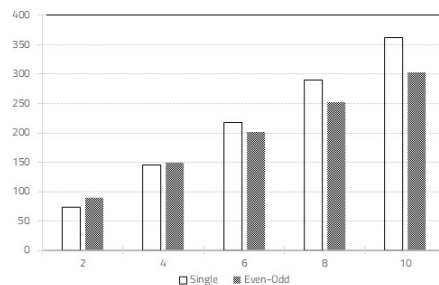


Fig. 5 Buffer's timing analysis with increasing data size

Quartus II를 이용하여 설계한 단일 버퍼와 Even-Odd-buffer의 스펙을 비교해 보았다. Intel FPGA 보드인 Cyclone V의 5CGXFC7C7F23C8 모델을 타겟으로 실험을 진행하였다. 표 1은 단일 버퍼와 Even-Odd-buffer의 자원과 성능을 나타낸다.

Even-Odd-buffer의 FPGA 내부 ARM 프로세서의 Logic utilization은 131개로 단일 버퍼 46개 대비 2.8배 증가하였고, 레지스터의 개수는 187개로 단일 버퍼 85개 대비 2.2배 증가하였다. 하지만 전력 소모는 355.44mW로 단일 버퍼 356.73mW 대비 0.36% 줄어든 것을 확인하였다. Even-Odd-buffer는 FPGA의 ARM 프로세서를 통해 데이터 이동 알고리즘을 효율적으로 관리할 수 있었고 FIFO가 하나 추가된 구조이기 때문에 전력소모량이 비슷하였다 [11]. 단일 버퍼와 Even-Odd-buffer의 전력소모량이 비슷하게 나온 이유는 RTL 합성시 단일 버퍼와 Even-Odd-buffer의 면적 차이가 FPGA 사용 면적에 비해 매우 작고, 소프트웨어가 전력을 계산하는 과정에서 leakage 등의 성분들을 계산하지 못하기 때문에 작은 오차가 발생한다.

Table. 1 Buffer's specification

	Single	Even-Odd
Family	Cyclone V	
Device	5CGXFC7C7F23C8	
Logic utilization	46/56,480	131/56,480
Total pins	71/268	77/268
Total registers	85	187
Power dissipation	356.73 mW	355.44 mW

4.3. 노이즈 제거 실험

제한한 Even-Odd-buffer와 LMS 알고리즘을 사용한 노이즈 캔슬링 가속기를 Verilog-HDL 언어로 기술하였다. 그림 6은 노이즈 캔슬링 가속기의 입출력 포트를 나타내는 Verilog-HDL 코드이다.

마이크로 입력되어 들어오는 신호는 input_data로 전송되고 실제 음원 파일은 desired_data 포트로 메모리를 통해 읽어 온다. Step_size와 reset_weights를 초기화할 수 있도록 하였고 FIR 필터의 출력값을 filtered_signal을 통해 출력하고 desired_data와 filtered_signal과의 오차를 y 출력 포트를 통해 출력한다. 이때 input_data는 제안하는 Even-Odd-buffer를 통해 16개의 데이터가 하나의 프레임으로 처리된다.

```

module Noise_Cancellation
    (clk, rst,
     input_data,
     desired_data,
     step_size,
     reset_weights,
     filtered_signal,
     y);

    input  clk, rst;
    input  signed [23:0] input_data;
    input  signed [23:0] desired_data;
    input  [23:0] step_size;
    input  reset_weights;
    output signed [23:0] filtered_signal;
    output signed [23:0] y;

```

Fig. 6 Noise cancellation verilog-HDL code for port

그림 7은 Verilog-HDL 코드를 RTL로 컴파일한 다음 게이트로 표현된 RTL을 도식하였다. LMS 알고리즘은 이전 필터의 계수를 통해 다음 필터의 계수를 바꾸기 때문에 필터의 계수를 저장하는 레지스터들이 존재한다. 또한, 입출력 포트, Even-Odd-buffer의 구조 등을 합성한 결과를 함께 볼 수 있다.

그림 8은 ModelSim의 타이밍 분석을 통해 입력 신호, 노이즈 신호, 출력 신호를 나타내었다. 사인파 입력 신호에서 LMS 알고리즘에 의해 결정된 필터의 계수들이 FIR의 계수들로 적용되어 필터링 된 출력을 만들어낸다. 필터링이 완료된 신호와 입력 신호의 차이를 뺄셈기를 통해 노이즈 신호로 출력했다. 적용 알고리즘을 사용하였기 때문에 처음 몇 개의 프레임은 출력신호가 불안정하다가 점차 안정된다.

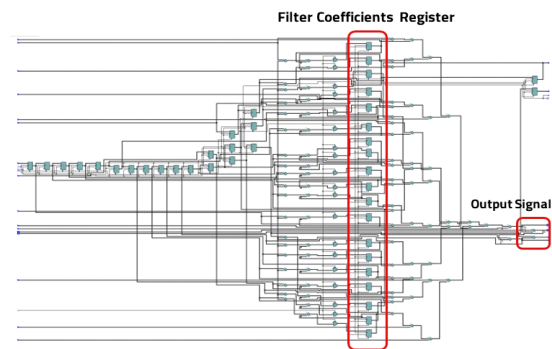


Fig. 7 Noise canceler RTL schematic

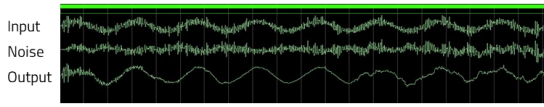


Fig. 8 Noise canceler RTL schematic

Table. 2 Noise canceller's specification

	LMS Noise Canceller
Family	Cyclone V
Device	5CGXFC7C7F23C8
Logic utilization	714/56,480
Total pins	75/268
Total registers	1092
Power dissipation	360.51 mW

V. 결 론

본 논문은 기존 단일 버퍼를 사용할 때 발생하는 데이터 처리 지연 문제와 처리속도를 개선하기 위해 입력 데이터를 두 개의 프레임 단위로 읽기 전용 버퍼와 쓰기 전용 버퍼를 Even-Odd-buffer에 효율적으로 이동시키는 노이즈 캔슬러를 개발하였다. 제안한 Even-Odd-buffer를 사용하여 노이즈 캔슬링 가속기를 설계하기 위해 FFT 기반의 노이즈 캔슬러와 LMS 적응 알고리즘을 이용한 노이즈 캔슬러라는 두 가지 경우의 가속기에 대하여 그 구조를 소개하였다. FFT 기반 노이즈 캔슬러에서는 FFT에서부터 출력된 신호의 스펙트럼 중 노이즈의 임계치를 넘는 신호를 FIR 필터를 통해 제거하는 방법을 사용하였다. 또한, 실험에서는 LMS 알고리즘을 사용하여 계수를 결정하는 알고리즘을 사용하여 노이즈 캔슬러를 설계하였다. LMS 알고리즘은 잘 알려진 노이즈 제거 방법이지만, 기존의 다중 버퍼를 사용했을 때 싱크 동기화, 데이터 이동 제어 신호의 구성 등 많은 설계 제한사항 등이 많았다. 하지만 본 논문에서는 이를 해결하기 위해 2개의 프레임을 단위로 하는 버퍼 구조를 사용하여 기존의 다중 버퍼의 문제점을 해결할 수 있었다.

Even-Odd-buffer를 사용하였을 때 단일 버퍼 대비 처리지연은 42% 감소하였고 처리속도는 29.2% 개선되었다. 이러한 성능 개선은 마이크로 입력된 신호와 음원 파일에서 불러온 신호의 싱크를 맞추기 위한 버퍼링 소

요를 줄일 수 있으므로 전력과 처리속도 측면에서 바람직하다. 또한, 단일 버퍼에서 Even-Odd-buffer로의 구조의 변경이 많지 않기 때문에 전력 소모는 0.36% 차이가 있었다.

실험은 Intel의 FPGA를 모델로 설정하여 진행하였고 arm 프로세서를 사용할 수 있었기 때문에 전력 소모의 차이가 크게 나지 않았지만, Convolution 연산을 포함하는 RTL 구조를 사용하여 하드웨어를 설계한다면 단일 버퍼와 Even-Odd-buffer의 전력 소모의 차이가 더 증가할 것이다. 그러나 clock gating 방법 등을 통해 처리속도를 높여 하드웨어의 대기시간을 줄이고 전체적으로 전력을 적게 소모하는 시스템을 설계할 수 있다.

본 논문에서 진행한 연구를 바탕으로 모든 시스템 동작을 RTL로만 설계하여 FPGA 보드에서 실제 음원과 마이크에서 들어오는 노이즈를 추출하여 제거하는 실험을 진행할 것이다. 또한, Even-Odd-buffer의 전력 소모를 줄일 수 있는 최적의 알고리즘을 개발하는 연구를 지속할 것이다.

ACKNOWLEDGEMENT

This study was supported by the BK21 FOUR project funded by the Ministry of Education, Korea (4199990113966), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1A6A1A03025109, 10%, NRF-2022R11A3069260, 20%) and by Ministry of Science and ICT (2020M3H2A1078119). This work was partly supported by an Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2021-0-00944, Metamorphic approach of unstructured validation/verification for analyzing binary code, 40%) and (No. 2022-0-00816, OpenAPI-based hw/sw platform for edge devices and cloud server, integrated with the on-demand code streaming engine powered by AI, 20%) and (No. 2022-0-01170, PIM Semiconductor Design Research Center, 10%). The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

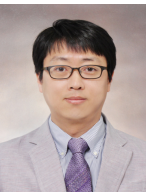
References

- [1] D. Li, S. Li, S. Zhang, J. Sun, L. Wang, and K. Wang, "Aging state prediction for supercapacitors based on heuristic kalman filter optimization extreme learning machine," *Energy*, vol. 250, pp. 123773, Jul. 2022.
- [2] S. Dixit and D. Nagaria, "LMS adaptive filters for noise cancellation: A review," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 5, pp. 2520, Oct. 2017.
- [3] Z. Li and M. Ierapetritou, "Process scheduling under uncertainty: Review and challenges," *Computers & Chemical Engineering*, vol. 32, no. 4-5, pp. 715-727, Apr. 2008.
- [4] J. Antoni and R. B. Randall, "Unsupervised noise cancellation for vibration signals: part I—evaluation of adaptive algorithms," *Mechanical Systems and Signal Processing*, vol. 18, no. 1, pp. 89-101, Jan. 2004.
- [5] J. H. Lee, L. E. Ooi, Y. H. Ko, and C. Y. Teoh, "Simulation for noise cancellation using LMS adaptive filter," in *Proceedings of IOP Conference Series: Materials Science and Engineering*, Bangkok, Thailand, vol. 211, no. 1, p. 012003, 2017.
- [6] K. Swaminathan, G. Lakshminarayanan, and S. B. Ko, "High Speed Generic Network Interface for Network on Chip Using Ping Pong Buffers," in *Proceedings of International Symposium on Electronic System Design (ISED)*, Kolkata, India, pp. 72-76, 2012.
- [7] K. Swaminathan, G. Lakshminarayanan, and S. B. Ko, "High Speed Low Power Ping Pong Buffering Based Network Interface for Network on Chip," *Journal of Low Power Electronics*, vol. 9, no. 3, pp. 322-331, Oct. 2013.
- [8] A. K. Sugiyama, R. Miyahara, and K. Oosugi, "A Noise Robust Hearable Device with an Adaptive Noise Canceller and its DSP Implementation," in *Proceedings of IEEE International Conference on Consumer Electronics (ICCE)*, Brighton, UK, pp. 1-2, 2019.
- [9] J. An and D. Park, "Fast adaptation techniques of compensation coefficient of active noise canceller using binary search algorithm," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 25, no. 11, pp. 1635-1641, Nov. 2021.
- [10] W. Fohl, J. Matthies, and B. Schwarz, "A FPGA-based Adaptive Noise Cancelling System," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, pp. 1-4, 2009.
- [11] D. Lee and D. Park, "Hardware and Software Co-Design Platform for Energy-Efficient FPGA Accelerator Design," *Journal of the Korea Institute of Information and Communication Engineering*, vol.25, no.1 pp.20-26, Jan. 2021.



박승현(Seunghyun Park)

2017년~현재 : 경북대학교 전자공학부 학사과정
 2021년~현재 : 경북대학교 인공지능 내장형 소프트웨어온칩 연구실 연구원
 2023년: 경북대학교 전자공학부 석박통합과정 입학예정
 ※관심분야 : 뉴럴네트워크 프로세서, FPGA기반 가속기 설계, 저전력 시스템



박대진(Daejin Park)

2001 경북대학교 전자전기공학부 학사
 2003 한국과학기술원(KAIST) 전기 및 전자공학과 석사
 2014 한국과학기술원(KAIST) 전기 및 전자공학과 박사
 2003~ 2014 SK Hynix/Samsung (차세대 LSI 설계) 수석연구원
 2014~ 2016 경북대학교 전자공학부 초빙교수 (2014년 대통령 Postdoctoral Fellow 선정)
 2016~ 현재 경북대학교 전자공학부 부교수
 ※관심분야 : 저전력 SoC 설계, 하드웨어-소프트웨어 Co-design, Dependable 스마트 IoT 시스템, Robust 임베디드 시스템