



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0083360
(43) 공개일자 2022년06월20일

(51) 국제특허분류(Int. Cl.)
G01R 31/14 (2006.01) G01R 23/165 (2006.01)
G01R 31/12 (2006.01)
(52) CPC특허분류
G01R 31/14 (2013.01)
G01R 23/165 (2013.01)
(21) 출원번호 10-2020-0173528
(22) 출원일자 2020년12월11일
심사청구일자 없음

(71) 출원인
한국전력공사
전라남도 나주시 전력로 55(빛가람동)
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
김녹원
경기도 용인시 기흥구 서천서로 27, 107동 1302호(서천동, 센트럴파크원)
프억 응우옌
경기도 용인시 기흥구 덕영대로 1732(서천동)
(74) 대리인
특허법인 정안

전체 청구항 수 : 총 10 항

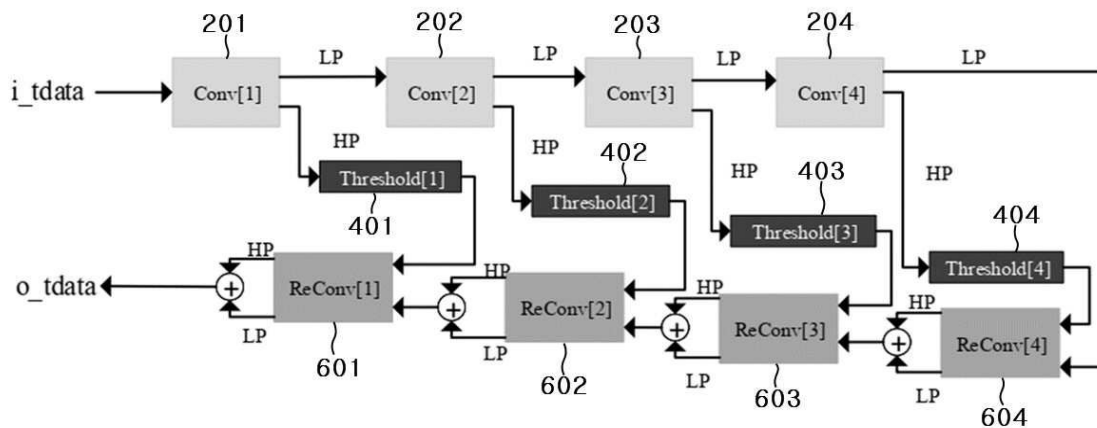
(54) 발명의 명칭 실시간 임베디드 노이즈 처리 장치

(57) 요약

본 발명의 실시간 임베디드 노이즈 처리 장치는, 센서로부터 수신한 입력신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제1 디컴포지셔; 상기 제1 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제2

(뒷면에 계속)

대표도 - 도7



디컴포지서; 상기 제2 디컴포지서가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제3 디컴포지서; 상기 제1 디컴포지서가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제1 트레시홀드; 상기 제2 디컴포지서가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제2 트레시홀드; 상기 제3 디컴포지서가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제3 트레시홀드; 상기 제1 트레시홀드가 생성한 필터링된 신호를 재결합하여 출력 신호로서 출력하는 제1 리컨스트럭처; 상기 제2 트레시홀드가 생성한 필터링된 신호를 재결합하여 상기 제1 리컨스트럭처로 전달하는 제2 리컨스트럭처; 및 상기 제3 트레시홀드가 생성한 필터링된 신호를 재결합하여 상기 제2 리컨스트럭처로 전달하는 제3 리컨스트럭처를 포함한다.

(52) CPC특허분류

G01R 31/1209 (2013.01)

G01R 31/1218 (2013.01)

명세서

청구범위

청구항 1

센서로부터 수신한 입력신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제1 디컴포지셔;

상기 제1 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제2 디컴포지셔;

상기 제2 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제3 디컴포지셔;

상기 제1 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제1 트레이시홀드;

상기 제2 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제2 트레이시홀드;

상기 제3 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제3 트레이시홀드;

상기 제1 트레이시홀드가 생성한 필터링된 신호를 재결합하여 출력 신호로서 출력하는 제1 리컨스트럭처;

상기 제2 트레이시홀드가 생성한 필터링된 신호를 재결합하여 상기 제1 리컨스트럭처로 전달하는 제2 리컨스트럭처; 및

상기 제3 트레이시홀드가 생성한 필터링된 신호를 재결합하여 상기 제2 리컨스트럭처로 전달하는 제3 리컨스트럭처

를 포함하는 실시간 임베디드 노이즈 처리 장치.

청구항 2

제1항에 있어서,

상기 제3 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제4 디컴포지셔;

상기 제4 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제4 트레이시홀드; 및

상기 제4 트레이시홀드가 생성한 필터링된 신호를 재결합하여 상기 제3 리컨스트럭처로 전달하는 제4 리컨스트럭처

를 더 포함하는 실시간 임베디드 노이즈 처리 장치.

청구항 3

제1항에 있어서,

상기 제1 내지 제3 트레이시홀드는,

소프트 트레이시홀드를 적용한 필터링을 수행하는 실시간 임베디드 노이즈 처리 장치.

청구항 4

제1항에 있어서,
 상기 제1 내지 제3 트레시홀드는,
 상기 제1 내지 제3 디컴포지서 중 하나가 출력하는 하이패스 신호를 버퍼링하는 출력 라인 버퍼; 및
 상기 하이패스 신호에 대하여, 현 시점과 이전 시점의 웨이블릿 변환 값과의 차이에 따른 벡터로서 트레시홀드를 적용하여 필터링하는 정렬된 버퍼와 스왑된 버퍼
 를 포함하는 실시간 임베디드 노이즈 처리 장치.

청구항 5

제1항에 있어서,
 상기 제1 내지 제3 디컴포지서는,
 입력되는 신호를 버퍼링하는 입력 라인 버퍼; 및
 상기 입력되는 신호에 대하여 웨이블릿 변환을 수행하는 컨벌루션 연산기
 를 포함하는 실시간 임베디드 노이즈 처리 장치.

청구항 6

제5항에 있어서,
 상기 컨벌루션 연산기는,
 상기 입력 라인 버퍼의 회선 버퍼의 각 가중치 요소와 해당 값 사이의 곱셈의 합으로 컨볼루션 결과를 생성하는
 실시간 임베디드 노이즈 처리 장치.

청구항 7

제5항에 있어서,
 상기 입력 라인 버퍼가 버퍼링 중에 다운 샘플링을 수행하거나,
 상기 컨벌루션 연산기가 웨이블릿 변환을 다운 샘플링하면서 수행하는 실시간 임베디드 노이즈 처리 장치.

청구항 8

제1항에 있어서,
 상기 입력 신호는,
 전력 설비의 부분 방전 감지를 위해 설치된 초음파 센서의 센싱 신호인 실시간 임베디드 노이즈 처리 장치.

청구항 9

제1항에 있어서,
 상기 입력 신호는,

전력 설비의 부분 방전 감지를 위해 설치된 광학 방식 센서의 센싱 신호인 실시간 임베디드 노이즈 처리 장치.

청구항 10

제1항에 있어서,

상기 출력 신호를 외부로 전송하기 위한 네트워크 전송 장치

를 더 포함하는 실시간 임베디드 노이즈 처리 장치.

발명의 설명

기술 분야

- [0001] 본 발명은 DWT 기반의 실시간 임베디드 노이즈 처리 장치에 관한 것으로, 특히 이기종 다중 센서 기반 전력망 시스템 모니터링을 위한 사물인터넷 하드웨어 플랫폼 아키텍처에 유용하게 적용될 수 있는 실시간 임베디드 노이즈 처리 장치에 관한 것이다.

배경 기술

- [0003] 전력설비는 기계적 스트레스, 온도 등의 영향으로 절연열화가 발생하여 내부 절연부분에서의 국부적인 부분방전이 발생하게 된다. 부분방전은 일반적으로 절연열화 현상에 의하여 절연파괴전압(Breakdown voltage)에 도달하였을 때 발생되며, 나타나는 물리적 현상은 빛(Light), 소리(음향신호), RF신호(전자파신호), 방전누설전류(전기적신호)의 신호가 방출된다.
- [0004] 일반적으로 수배전반, 분전반, 모터 제어반을 구성하는 전기 설비 또는 부스 / 케이블 등에서 아크 또는 부분방전 발생 시 조속히 유지보수를 시행하지 않을 경우, 추후 섬락 사고, 감전 사고, 전기 화재 등이 발생할 수 있다.
- [0005] 한편, 통신 기술의 발전은, 전력 IoT 네트워크 상의 이종 다수 센서 신호들을 처리할 수 있는 통합 HMI 플랫폼을 전력 분야에서도 도입 가능하게 하였다.
- [0006] 전력망 시스템에서 부분 방전의 발생은 일반적으로 압전 센서, 자외선 센서 또는 광학 센서와 같은 한 가지 유형의 센서를 통해 인지할 수 있다. 한 가지 유형의 장치 만 작동하면 이벤트를 누락하거나 잘못된 이벤트를 경고하는 잠재적인 문제가 있다.
- [0007] 전력망 시스템의 각 설비가 운영되는 다양한 환경 조건에 따라 다양한 형태의 노이즈(잡음)이 섞이기 때문에, 효과적으로 이를 필터링하기 위한 수단이 없었기 때문이다. 특히 연산 처리 하드웨어의 사양이 낮은 임베디드 장치에서 실시간으로 노이즈를 제거(처리)하는데 어려움이 있는 바, 이러한 어려움은 IoT 네트워크를 이용하여 전력 분야에서 센서 네트워크를 구축하는데 장애가 되었다. 다시말해, 전력 IoT 네트워크의 통합 HMI 플랫폼 상에서 부분 방전 신호를 효과적으로 처리할 수 있는 수단이 마땅치 않았다.

선행기술문헌

특허문헌

- [0009] (특허문헌 0001) 대한민국 등록공보 10-1486994호

발명의 내용

해결하려는 과제

- [0010] 본 발명은 센서나 신호 전송 경로상의 오류를 억제하여 정확하게 부분 방전을 검출할 수 있는 실시간 임베디드 노이즈 처리 장치를 제안하고자 한다.

과제의 해결 수단

- [0012] 본 발명의 일 측면에 따른 실시간 임베디드 노이즈 처리 장치는, 센서로부터 수신한 입력신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제1 디컴포지셔; 상기 제1 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제2 디컴포지셔; 상기 제2 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제3 디컴포지셔; 상기 제1 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제1 트레이시홀드; 상기 제2 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제2 트레이시홀드; 상기 제3 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제3 트레이시홀드; 상기 제1 트레이시홀드가 생성한 필터링된 신호를 재결합하여 출력 신호로서 출력하는 제1 리컨스트럭처; 상기 제2 트레이시홀드가 생성한 필터링된 신호를 재결합하여 상기 제1 리컨스트럭처로 전달하는 제2 리컨스트럭처; 및 상기 제3 트레이시홀드가 생성한 필터링된 신호를 재결합하여 상기 제2 리컨스트럭처로 전달하는 제3 리컨스트럭처를 포함할 수 있다.
- [0013] 여기서, 상기 제3 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제4 디컴포지셔; 상기 제4 디컴포지셔가 출력하는 하이패스 신호에 트레이시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제4 트레이시홀드; 및 상기 제4 트레이시홀드가 생성한 필터링된 신호를 재결합하여 상기 제3 리컨스트럭처로 전달하는 제4 리컨스트럭처를 더 포함할 수 있다.
- [0014] 여기서, 상기 제1 내지 제3 트레이시홀드는, 소프트 트레이시홀드를 적용한 필터링을 수행할 수 있다.
- [0015] 여기서, 상기 제1 내지 제3 트레이시홀드는, 상기 제1 내지 제3 디컴포지셔 중 하나가 출력하는 하이패스 신호를 버퍼링하는 출력 라인 버퍼; 및 상기 하이패스 신호에 대하여, 현 시점과 이전 시점의 웨이블릿 변환 값과의 차이에 따른 벡터로서 트레이시홀드를 적용하여 필터링하는 정렬된 버퍼와 스왑된 버퍼를 포함할 수 있다.
- [0016] 여기서, 상기 제1 내지 제3 디컴포지셔는, 입력되는 신호를 버퍼링하는 입력 라인 버퍼; 및 상기 입력되는 신호에 대하여 웨이블릿 변환을 수행하는 컨벌루션 연산기를 포함할 수 있다.
- [0017] 여기서, 상기 컨벌루션 연산기는, 상기 입력 라인 버퍼의 회선 버퍼의 각 가중치 요소와 해당 값 사이의 곱셈의 합으로 컨볼루션 결과를 생성할 수 있다.
- [0018] 여기서, 상기 입력 라인 버퍼가 버퍼링 중에 다운 샘플링을 수행하거나, 상기 컨벌루션 연산기가 웨이블릿 변환을 다운 샘플링하면서 수행할 수 있다.
- [0019] 여기서, 상기 입력 신호는, 전력 설비의 부분 방전 감지를 위해 설치된 초음파 센서의 센싱 신호일 수 있다.
- [0020] 여기서, 상기 입력 신호는, 전력 설비의 부분 방전 감지를 위해 설치된 광학 방식 센서의 센싱 신호일 수 있다.
- [0021] 여기서, 상기 출력 신호를 외부로 전송하기 위한 네트워크 전송 장치를 더 포함할 수 있다.
- [0022]

발명의 효과

- [0023] 상술한 구성의 본 발명의 사상에 따른 임베디드 노이즈 처리 장치를 실시하면, 센서나 신호 전송 경로상의 오류를 억제하여 정확하게 부분 방전을 검출할 수 있는 이점이 있다.
- [0024] 본 발명의 사상에 따른 임베디드 노이즈 처리 장치는, IoT 하드웨어 플랫폼 기술을 적용하여 많은 설치 및 유지 관리 비용없이 유연성과 배포 용이성을 확보할 수 있는 이점이 있다.
- [0025] 본 발명의 사상에 따른 임베디드 노이즈 처리 장치는, 다양한 열악한 임베디드 환경에서 센서 신호로부터 깨끗한 저노이즈 신호를 획득할 수 있는 이점이 있다.

[0026] 본 발명의 사상에 따른 임베디드 노이즈 처리 장치는, 다양한 환경에서 전력 배송 시스템을 모니터링 하는데 최적화된 이점이 있다.

도면의 간단한 설명

[0028] 도 1은 본 발명의 노이즈 처리 장치가 적용될 수 있는 시스템의 블록 다이어그램이다.
 도 2는 본 발명의 노이즈 처리 장치가 적용된 메인 프로세싱 유닛의 구성들을 도시한 유닛 구성도.
 도 3은 웨이블릿 알고리즘의 일반적인 구성을 도시한 블록도.
 도 4는 본 발명의 사상에 따라 센서 신호의 잡음 제거에 4-레벨 웨이블릿 알고리즘을 적용한 구성을 도시한 개념도.
 도 5는 symlet4 웨이블릿의 4개 필터를 나타낸 그래프들.
 도 6a는 하드웨어적인 네트워크 연결 구성을 도시한 연결도.
 도 6b는 하드웨어 플랫폼 내의 웨이블릿 노이즈 제거 IP 코어를 나타낸 구성도.
 도 7은 본 발명의 일 실시예에 따른 실시간 임베디드 노이즈 처리 장치의 개략적인 세부 구성을 나타낸 코어 구성도.
 도 8a는 도 7의 디컴포지셔(Conv) 블록 및 트레시홀드(Threshold) 블록의 세부 구성을 도시한 회로도.
 도 8b는 소프트 트레시홀드 기능을 수행하는 소프트 트레시홀드 어플리케이션의 내부 구성을 도시한 회로도.
 도 9a 및 도 9b는 다운 샘플링을 사용하는 것과 다운 샘플링을 사용하지 않는 것의 차이점을 나타내는 개념도.
 도 10은 소프트웨어 구현을 사용하여 다양한 수준의 웨이블릿 노이즈 제거 효과를 나타낸 그래프들.
 도 11은 하드웨어 구현을 사용하여 다양한 수준의 웨이블릿 노이즈 제거 효과를 나타낸 그래프들.
 도 12는 잡음 제거 IP 코어의 동작을 나타내는 타이밍 다이어그램.
 도 13은 제안된 아키텍처의 상용화를 위한 PCB 형태를 도시한 PCB도.

발명을 실시하기 위한 구체적인 내용

[0029] 본 발명을 설명함에 있어서 제 1, 제 2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 구성요소들은 용어들에 의해 한정되지 않을 수 있다. 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제 1 구성요소는 제 2 구성요소로 명명될 수 있고, 유사하게 제 2 구성요소도 제 1 구성요소로 명명될 수 있다.

[0030] 어떤 구성요소가 다른 구성요소에 연결되어 있거나 접속되어 있다고 언급되는 경우는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해될 수 있다.

[0031] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함할 수 있다.

[0032] 본 명세서에서, 포함하다 또는 구비하다 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것으로서, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해될 수 있다.

[0033] 또한, 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있다.

[0035] 현재 대부분의 IoT 하드웨어 솔루션은 주로 데이터 전달을 수행하거나 소형 소프트웨어 알고리즘을 수행하는데, 실시간 노이즈 제거에 있어서는 단일 센서를 사용하나, 다중 센서를 사용하나, 효과가 없거나 느리다. 본 발명

에서는 단일/다중 센서 신호에 대한 IoT에서 실시간 노이즈 제거를 위한 방안을 제시한다.

[0036] 먼저, 본 발명의 사상인 노이즈 처리 방안을 살펴보기에 앞서 IoT에서 임베디드 형태로 구현되기 위한 구성들부터 살펴보겠다.

[0037] 도 1은 본 발명의 노이즈 처리 장치가 적용될 수 있는 시스템의 블록 다이어그램이다.

[0038] 도시한 블록 다이어그램은 나노 물질로 제작되며 압전 방식 센서 및 자외선 센서의 이중 센서 구성을 구비한다. 또한, 단일/이중 센서와 인터페이스하고 디지털 데이터를 하드웨어 플랫폼으로 전송하는 ADC 보드, 주요 처리 제어 및 관리 작업을위한 MCU, 신호 처리 작업을위한 FPGA, AP SoC에서 데이터를 수신하고 다른 노드로 전송하여 원격에 도달하는 BLE 모듈을 포함하여, 장치를 구성하는 것을 나타낸다. 여기서 구성되는 장치는 프로그래밍 가능한 시스템 온 칩-AP SoC 형태로 구현될 수 있다.

[0040] 도 2는 본 발명의 노이즈 처리 장치가 적용된 메인 프로세싱 유닛의 구성들을 도시한다.

[0041] 도시한 프로세싱 유닛의 구성들은 제어 및 신호 처리 작업 모두에 대한 유연성과 효율성을 목표로한다. MCU만 사용하면 제어에는 잘 대처할 수 있지만 고속 데이터 처리에는 실패할 수 있다. 반면 FPGA는 로직 부분을 기반으로하는 고속 처리 작업을 효과적으로 처리 할 수있지만 로직이 식별 된 후에는 쉽게 조정할 수 없다. Zynq-7000 AP SoC 하드웨어 플랫폼은 저렴한 비용으로 MCU 작동과 FPGA 로직을 모두 제공한다. 도 3은 처리 시스템 (PS)과 프로그래밍 가능 로직 (PL)의 두 가지 주요 부분이있는 Zynq-7000 AP So C의 구성 요소를 보여준다. PS는 일정 또는 제어 작업을위한 빠른 프로토 타이핑 환경을 제공하는 반면, FPGA 기반 PL은 데이터를 샘플링하고 처리하기 위해 맞춤형 하드웨어 설계를 구현할 수 있는 유연성을 제공한다.

[0042] PS는 ARM 기반 프로세서와 SPI(직렬 주변 장치 인터페이스), I2C(통합 회로), CAN(컨트롤러 영역 네트워크), 범용 비동기 수신기와 같은 많은 내장 주변 장치를 구비한 MCU로 구현될 수 있다. 송신기(UART), 범용 입력/출력(GPIO) 등, 센서와 상호 작용하는 것이 유리하다.

[0043] PS는 특정 감지 레벨을 넘는 센서 신호 발생에 따라 PD 발생 시점을 감지한다. 구현에 유연성이 있는 FPGA 기반 PL은 PS에서 실시간 처리 서비스가 필요한 작업을 오프로딩하고, 프로세서 대역폭을 회수하는 등 다양한 유형의 하드웨어 가속을 수행할 수 있다. PL은 신호 처리 메커니즘(예 : 웨이블릿 노이즈 제거)을 구현하여 보다 정확하고 노이즈에 강하다. PL은 잡음 제거 신호 처리의 많은 계산에서 프로세서를 분리하는데 도움이 되어 전체 시스템에 유리하다.

[0045] 노이즈 제거 기술은 PL에서 하드웨어 가속기로 구현되면 보다 실시간 작동을 용이하게 한다. 고속 푸리에 변환 기반, 저역 통과 필터링, 위너 빌 분포 기반, 재귀 최소 제곱 및 지수 가중치 재귀 최소 제곱 방법과 같은 28개의 잡음 제거 기술 중에서 웨이블릿 변환 기반 잡음 제거가 잡음 유형이 다양한 전력 분야 센서 신호에 대하여 최고의 성능을 제공하는 것을 확인하였으며, 본 발명에서 실시간 잡음 제거를 위해 구현되었다.

[0046] 도 3은 웨이블릿 알고리즘의 일반적인 구성을 도시한 블록도이다.

[0047] 도시한 블록도는 노이즈 제거 처리의 기본적인 단계들만 나타내며, 웨이블릿 변환을 수행하는 분해(Decomposition), 각 노이즈 레벨에서 임계 값을 계산하고 분해 단계의 출력 일부에 임계 값을 적용하는 임계값(Threshold) 추정 및 적용, 역 웨이블릿 변환(Rcontruction)을 수행하여, 잡음이 제거된 신호를 생성한다.

[0048] 마더 웨이블릿과 관련하여 시프트되고 스케일링 된 신호를 생성하는 웨이블릿 변환은 다음 수학적 식 1로 표현될 수 있다.

수학적 식 1

$$\Psi_f^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(x) \psi^*\left(\frac{x-\tau}{s}\right) dx$$

[0049]

[0050] 여기서, τ = 변환 매개 변수, s = 척도 매개 변수, $\Psi(x)$ = 마더 웨이블릿이다. 이러한 작업의 처리 메커니즘

은 Mallat에 의해 개발 된 웨이블릿 변환을 위한 빠른 알고리즘을 기반으로 하며, 근사(cA)와 세부(cD)의 두 가지 계수 그룹을 생성한다. 도 3에서 나타난 바와 같이 잡음이 있는 신호는 잡음 제거를 위해 분해(Decomposition), 쓰레시홀드(Threshold) 및 재구성(Rccontruction)의 세 단계를 거치게 된다. 분해(Decomposition)는 웨이블릿 변환을 기반으로 하는 반면, 재구성(Rccontruction)은 역 연산의 결과이다. 각 단계의 작업 세부 사항은 4 가지 수준의 노이즈 제거를 수행하는 후술하는 도 4에서 확인할 수 있다.

[0052] 도 4는 본 발명의 사상에 따라 센서 신호의 잡음 제거에 4-레벨 웨이블릿 알고리즘을 적용한 구성을 도시한 개념도이다.

[0053] 도 5는 symlet4 웨이블릿의 4개 필터로서, 도 5(a)는 분해시 저역 통과(LoD), 도 5(b)는 하이 패스 분해(HiD), 도 5(c)는 재구성의 저역 통과(LoR), 도 5(d)는 재구성의 고역 통과(HiR)를 나타낸다.

[0054] 도시한 구성은 센서 신호의 잡음 제거에 웨이블릿 알고리즘을 비대칭적으로 적용한 구조를 설명하기 위한 것으로, 각 연산 블록은 일반적인 처리 과정들로 채워져 있으며, 상기 일반적인 처리 과정을 종래 기술로 구현하면, 연산 부하가 커서 임베디드로 적합하지 않지만, 개념 설명을 위해 먼저 도 4의 형태를 제시한다.

[0055] 도 4에서 입력 신호는 분해(Decomposition) 단계를 거쳐 근사(cA) 및 세부(cD)의 두 가지 유형의 계수로 생성된다. 분해(Decomposition) 단계에는 분해를 위한 저역 통과(LoD)와 분해를 위한 고역 통과 (HiD)의 두 가지 유형의 필터가 있다. 근사(cA) 및 세부(cD)는 입력의 컨볼루션 연산에 이어지는 LoD 및 HiD를 사용한 다운 샘플링 연산의 결과이다.

[0056] 레벨 1(cA1)의 근사(cA)는 마지막 레벨(cA4)에 도달할 때까지 다른 레벨에서 디컴포지션(분해) 블록을 계속 통과한다. 한편, 세부(cD)는 쓰레시홀드(Threshold)값 계산과 쓰레시홀드(Threshold) 적용이라는 두 가지 작업을 수행하는 쓰레시홀드(Threshold) 블록을 통과한다. 쓰레시홀드(Threshold) 계산은 해당 수준에서 입력되는 cD에서 쓰레시홀드(Threshold)를 찾는 것이다. 쓰레시홀드(Threshold)가 결정되면 원래의 세부(cD)에 적용된다. 이제, 쓰레시홀드(Threshold)값 cD(cD')을 사용할 수 있으며, 근사(cA)와 함께 재구성 블록에 들어가기 전에 다른 수준의 동일한 작업 집합의 완료를 대기한다.

[0057] 두 계수는 먼저 업 샘플링 작업을 거쳐 컨볼루션 작업을 계속 수행한다. 이때 두 필터는 재구성(Rccontruction) 단계의 필터인 LoR(재건을 위한 저역 통과) 및 재구성을 위한 고역 통과(HiR)의 시간에 따라 반전된다(도 5 참조). 그런 다음, 두 결과를 더하여 상위 수준에 대한 cA'를 생성한다. 최종 재구성 레벨(level 1)에 도달하면 잡음 제거된(denoised) 데이터가 생성된다.

[0059] 다양한 실제 신호에 적용되는 웨이블릿 변환은 희소 표현을 생성하므로 신호 표현을 큰 크기를 가진 적은 수의 웨이블릿 계수로 정착시킨다. 수축 작업은 크기가 작은 일부 계수를 제거하는데, 제거된 계수는 잡음 성분으로 기여하므로 신호의 전체 품질에 영향을 주지 않는다. 수축 작업은 다음 수학적 2와 같이 하드 또는 소프트 쓰레시홀드 함수에 의해 각 수준의 모든 세부 계수에 쓰레시홀드(Threshold) 값이 적용될 때 발생한다.

수학적 2

$$D^H(y, t) = \begin{cases} y & |y| \geq t \\ 0 & |y| < t \end{cases}$$

$$D^S(y, t) = \begin{cases} \text{sign}(y)(|y| - t) & |y| \geq t \\ 0 & |y| < t \end{cases}$$

[0060]

[0061] 하드 쓰레시홀드 기능은 일부 불연속성과 진동을 생성하기 때문에, 소프트 쓰레시홀드가 부드러운 신호 전환을 생성하는데 사용된다. 예컨대, 하기 수학적 3과 같이, 미디언(median)이 mi인 총 ni 세부 계수가 있는 레벨i(λi로 표시)에서의 쓰레시홀드 추정치는 범용 쓰레시홀드 추정기에서 계산될 수 있다.

수학식 3

$$\lambda_i = m_i / 0.6745 \cdot \sqrt{2 \cdot \log(n_i)}$$

[0062]

[0063]

Symlet 또는 Daubechies와 같은 직교 웨이블릿을 사용한 신호 처리는 가장 간결한 신호 표현을 생성한다. 따라서, 4개의 소실 모멘트(symlet4)가있는 Symlet은 노이즈 제거 가속기의 마더 웨이블릿으로 사용된다. 변환의 분해 단계 후 저장된 계수가 적기 때문에 작업 저장이 효율적이고, 재구성 후 신호는 노이즈 성분이 지워졌기 때문에 보다 명확하고 선명하다.

[0065]

도 6a는 하드웨어적인 네트워크 연결 구성을 도시한다.

[0066]

설정에는 소스 장치, 브리지 장치 및 대상 장치의 세 가지 장치가 포함되며, 소스 장치에는 센서 부품이 부착되어 있으며 BLE 모듈에 연결된다. 브리지 장치에는 두 개의 BLE 모듈이 있다. 하나는 소스 장치에서 데이터를 수신하기 위한 것이고 다른 하나는 대상 장치로 전달하기 위한 것이다. 브리지 장치의 역할은 소스 장치와 대상 장치 사이의 간격을 줄이는 것이다.

[0068]

도 6b는 하드웨어 플랫폼 내의 웨이블릿 노이즈 제거 IP 코어를 나타낸다.

[0069]

AXI DMA에 대한 2 개의 전용 포인트-투-포인트 및 효율적인 AXI-Stream 인터페이스가 있으며, 이는 차례로 메인 메모리(DDR)에서 데이터를 송수신한다. DDR의 데이터는 부분 방전이 발생하는 값을 결정하거나 BLE 모듈로 전송하는 것과 같은 다른 목적으로 사용될 수 있다.

[0071]

도 7은 본 발명의 일 실시예에 따른 실시간 임베디드 노이즈 처리 장치의 개략적인 세부 구성을 나타낸다.

[0072]

도 7에서 세 블록은 노이즈 제거 프로세스의 세 단계를 반영한다. 분해(Conv) 블록은 분해 단계에서 컨볼 루션과 다운 샘플링을 모두 포함한다. 쓰레시홀드 블록에는 쓰레시홀드 계산과 쓰레시홀드 적용의 두 가지 하위 단계가 있다. 재구성(ReConv) 블록은 재구성 단계에서 업 샘플링과 컨볼루션을 통합하고 있다.

[0073]

도시한 실시간 임베디드 노이즈 처리 장치는, 센서로부터 수신한 입력신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호(세부(cD))와 로우패스 신호(근사(cA))를 출력하는 제1 디컴포지셔(분해 블록)(201); 상기 제1 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제2 디컴포지셔(201); 상기 제2 디컴포지셔가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제3 디컴포지셔(203);

[0074]

상기 제1 디컴포지셔(201)가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제1 트레시홀드(401); 상기 제2 디컴포지셔(202)가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제2 트레시홀드(402); 상기 제3 디컴포지셔(203)가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제3 트레시홀드(403);

[0075]

상기 제1 트레시홀드(401)가 생성한 필터링된 신호를 재결합하여 출력 신호로서 출력하는 제1 리컨스트럭처(601); 상기 제2 트레시홀드(402)가 생성한 필터링된 신호를 재결합하여 상기 제1 리컨스트럭처(601)로 전달하는 제2 리컨스트럭처(602); 및 상기 제3 트레시홀드(403)가 생성한 필터링된 신호를 재결합하여 상기 제2 리컨스트럭처로 전달하는 제3 리컨스트럭처(603)를 포함한다.

[0076]

상기 정의한 것과 같이 3 레벨만으로 어느 정도 충분한 성능을 얻을 수 있으며, 5 레벨로 노이즈 제거 성능을 극대화할 수도 있다. 그러나, 4 레벨이 현재 하드웨어 수준에서 적합하므로, 4 레벨로 구체화하여 설명하지만, 권리범위가 이에 한정되지는 않는다.

[0077]

도면들에서 구체화한 4 레벨 실시간 임베디드 노이즈 처리 장치의 경우, 상기 제3 디컴포지셔(203)가 출력하는 로우패스 신호에 웨이블릿 변환 및 다운샘플링을 수행하고, 하이패스 신호와 로우패스 신호를 출력하는 제4 디컴포지셔(204); 상기 제4 디컴포지셔(204)가 출력하는 하이패스 신호에 트레시홀드를 적용하여 필터링된 하이패스 신호를 생성하는 제4 트레시홀드(404); 상기 제4 트레시홀드(404)가 생성한 필터링된 하이패스 신호를 재결합하여 상기 제3 리컨스트럭처(603)로 전달하는 제4 리컨스트럭처(604)를 포함한다.

스 신호를 생성하는 제4 트레시홀드(404); 및 상기 제4 트레시홀드가 생성한 필터링된 신호를 재결합하여 상기 제3 리컨스트럭처(603)로 전달하는 제4 리컨스트럭처(604)를 더 포함한다.

[0078] 상기 제1 내지 제4 트레시홀드는, 트레시홀드에 의해 제거되지 않는 영역의 값들은 트레시홀드값만큼 시프트시키는 방식의 소프트 트레시홀드를 적용하는 것이 유리하다.

[0080] 도 8a는 디컴포지셔(분해, Conv) 블록(200) 및 트레시홀드(Threshold) 블록(400)의 세부 구성을 도시한 회로도이다.

[0081] 도 8b는 소프트 트레시홀드 기능을 수행하는 소프트 트레시홀드 어플리케이션의 내부 구성을 도시한 회로도이다.

[0082] 디컴포지셔(분해, Conv) 블록(200)은 기본적으로 재구성 단계에서 사용되는 리컨스트럭처(ReConv) 블록과 동일한 구조 및 구성 요소(라인 버퍼, 가중치 메모리, 승수, 가산기 등)를 가지고 있다. 이들의 차이점은 로딩된 가중치의 목적에 있다 : Conv for Decomposition(분해) 및 ReConv for Reconv(재구성). 디컴포지셔(분해, Conv) 블록(200)에서 메모리 영역은 컨볼루션에 대한 가중치를 보유한다.(분해의 경우 LoD 및 HiD, 재구성의 경우 LoR 및 HiR).

[0083] 입력 라인 버퍼(210)는 새로운 데이터를 이동시키고 해당 가중치로 곱셈을 수행 한 다음 모든 결과를 더하여 디컴포지셔(Conv) 블록(200)의 출력을 생성한다. 저역 통과 필터로 컨볼루션된 데이터는 곧바로 다음 레벨로 이동하고, 다른 하나는 트레시홀드(Threshold) 블록(400)으로 입력된다. 한 레벨에서 트레시홀드(Threshold)값을 계산하기 위해 트레시홀드(Threshold) 블록에는 디컴포지셔(분해, Conv) 블록(200)에서 나오는 데이터 스트림과 동일한 길이의 출력 라인 버퍼(410)가 구비된다. 상기 버퍼(410)는 값이 결정되면 나중에 트레시홀드(Threshold)를 적용할 수 있도록 원래 값을 유지한다.

[0084] 각 컨볼루션 부분에는 symlet4의 필터 가중치가 미리로드되어 있으며 도면의 왼쪽 부분에 표시된 것처럼 계산을 위한 전용 리소스(승수, 가산기)가 있다. 라인 버퍼는 컨볼루션 연산을 수행하는 창이며 가중치와 길이가 같다. 메모리, 즉, $n = 8$ 은 symlet4에 대한 각 필터의 요소수이다. 다운 샘플링된 각 입력 신호가 라인 버퍼로 이동하면 회선 버퍼의 각 가중치 요소와 해당 값 사이의 곱셈의 합으로 컨볼루션 결과가 생성되며 다음 수학적 식 4와 같이 표현될 수 있다.

수학적 식 4

$$o_tdata_* = \sum_{j=1}^n L[i][j] \times W[i][j]$$

[0085] where $i = 0$ for LP and $i = 1$ for HP

[0086] 다시 말해, 도시한 제1 내지 제4 디컴포지셔(201 ~ 204)는, 입력되는 신호를 버퍼링하는 입력 라인 버퍼(210); 및 상기 입력되는 신호에 대하여 웨이블릿 변환을 수행하는 컨볼루션 연산기(220)를 포함할 수 있다.

[0087] 또한, 상기 컨볼루션 연산기(220)는, 상기 입력 라인 버퍼(210)의 회선 버퍼의 각 가중치 요소와 해당 값 사이의 곱셈의 합으로 컨볼루션 결과를 생성할 수 있다.

[0088] 구현에 따라, 상기 입력 라인 버퍼(210)가 버퍼링 중에 다운 샘플링을 수행하거나, 상기 컨볼루션 연산기(220)가 웨이블릿 변환을 다운 샘플링하면서 수행할 수 있다.

[0089] 예컨대, 입력 및 출력 신호 유형은 부호있는 16 비트일 수 있다. 다운 샘플링을 용이하게 하기 위해 입력 데이터에는 두 개의 입력이 포함되어 있으므로 32 비트(예 : $i_tdata_lp[31 : 0]$)이고, 출력 데이터에는 16 비트(예 : $o_tdata_lp[15 : 0]$)가 될 수 있다.

[0090] 예컨대, 디컴포지셔(Conv) 블록(200) 내에서 다운 샘플링은 다음과 같이 수행될 수 있다(*_lp 및 *_hp 모두에 적용됨). 입력 데이터는 16비트 데이터 2개($i_tdata_lp[15 : 0]$, $i_tdata_lp[31:16]$)를 포함하는 32 비트 데이터($i_tdata_lp[31 : 0]$)이다.

- [0091] 모든 클럭 사이클에서 모든 32 비트 입력 데이터는 회선 연산을 위해 라인 버퍼의 두 요소로 이동되고 하나의 16 비트 결과, 즉 $o_tdata_lp[15 : 0]$ 을 생성한다. 두 개의 16 비트 결과가 하나의 데이터로 압축되어 다음 레벨로 이동한다.
- [0093] 도 9a 및 도 9b는 다운 샘플링을 사용하는 것과 다운 샘플링을 사용하지 않는 것의 차이점을 나타내고 있다.
- [0094] 도 9a와 같이 다운 샘플링을 사용하면, 4클럭 사이클 후 8개의 입력 데이터가 4개의 출력 데이터를 생성하는 반면, 도 9b와 같이 다운 샘플링이 없으면 4클럭 사이클 후 4개의 입력 데이터가 4개의 출력 데이터를 생성한다.
- [0096] 트레이시홀드(Threshold)(400)에는 데이터 스트림 길이의 절반인 정렬된 버퍼(so)(422)와 스왑된 버퍼(sw)(426)라는 두 개의 추가 라인 버퍼가 있다. 이 두 버퍼의 각 요소는 스왑 모듈(424)을 통해 비교 및 스왑 작업을 수행한다.
- [0097] 각 버퍼의 요소수는 트레이시홀드(Threshold) 블록에 입력되는 데이터 길이의 절반이며, 트레이시홀드(Threshold) 값이 결정된 후, 제거되도록 다른 라인 버퍼에 저장된다. 매 클럭주기마다 디컴포지셔(분해, Conv) 블록에서 새 데이터가 이동하면 정렬된 버퍼(422)의 각 요소가 스왑된 버퍼(426)의 이전 요소와 비교되고 해당 내용은 하기 표 1 같은 알고리즘으로 업데이트될 수 있다.

표 1

Algorithm 1: The sorting algorithm on so and sw buffers.

Data: Stream of data with length L from the HP Conv block.

Result: Sorted data on two auxiliary buffers

for every element i in so and sw buffers do

if $so_i > sw_{i-1}$ then

$sw_i \leftarrow so_i$;

$so_i \leftarrow sw_{i-1}$;

else

$sw_i \leftarrow sw_{i-1}$;

$so_i \leftarrow so_i$ (keep old value) ;

end

end

- [0098]
- [0099] 이러한 방식으로 데이터 스트림의 미디언(중앙값)이 결정되고 트레이시홀드(Threshold)값을 얻기 위해 범용 트레이시홀드(Threshold) 추정기가 사용될 수 있다. 이 트레이시홀드(Threshold) 값은 트레이시홀드(Threshold) 애플리케이션 블록에서 사용될 수 있다.
- [0100] 다시 말해, 도시한 제1 내지 제3, 제4 트레이시홀드(401 ~ 404)는, 상기 제1 내지 제3 디컴포지셔(201 ~ 204) 중 하나가 출력하는 하이패스 신호(세부(cD))를 버퍼링하는 출력 라인 버퍼(410); 및 상기 하이패스 신호(세부(cD))에 대하여, 현 시점과 이전 시점의 웨이블릿 변환 값과의 차이에 따른 벡터로서 트레이시홀드를 적용하여 필터링하는 정렬된 버퍼(422)와 스왑된 버퍼(426)를 포함할 수 있다.
- [0101] 즉, 트레이시홀드 블록(400)에 입력되는 데이터는 두 경로를 통과한다. 첫 번째 경로는 적용할 트레이시홀드 값을 결정하기 위해 정렬 모듈(420)(스왑 모듈(424), 정렬된 버퍼(422), 스왑된 버퍼(426) 포함)로 들어가는 것이다. 두 번째 경로는 첫 번째 경로에서 트레이시홀드 값이 계산될 때까지 입력 데이터 스트림을 저장하기 위한 것이다. 트레이시홀드 값을 사용할 수 있게 되면 소프트 트레이시홀드 적용 회로를 사용하여 저장된 입력 데이터에 적용된다. 이러한 과정으로 트레이시홀드 블록(400)의 출력이 동기화되어, 해당 리컨스트럭처(ReConv) 블록(재구성 단계)에서 사용될 수 있다.

- [0103] 본 발명에서는 소프트 트레시홀드를 적용하였으며, 이를 위한 회로가 도 8b와 같다. 회로도에서 입력의 절대 값이 트레시홀드 값보다 작으면 출력은 0값을 얻으며, 그렇지 않은 경우 입력 값이 양수이면 출력은 입력 값에서 트레시홀드 값을 뺀 값으로, 입력의 절대 값이 트레시홀드 값보다 크고 음수인 경우 출력에 입력 값과 트레시홀드 값이 더해진 값으로 출력을 얻을 수 있다.
- [0104] 다운 샘플링 작업은 컨볼루션 모듈의 다른 모든 결과를 유지하며 데이터가 디컴포지션(분해, Conv) 블록에 입력될 때 수행된다. 전체 데이터 샘플의 절반을 삭제하는 작업은 데이터 양을 줄일뿐만 아니라 데이터가 재구성 단계에서 복원될 때 앨리어싱 효과를 제거하므로 유용하다.
- [0105] 트레시홀드 단계는 노이즈 제거 목적에 매우 중요하다. 이는 데이터 스트림에서 트레시홀드 값을 동적으로 결정하며, 이는 HP 컨볼루션 경로의 결과에 적용되어 특정 수준에서 세부 계수를 얻는다.
- [0106] 상술한 정렬 알고리즘은 데이터 스트림의 미디언(중앙값)이 두 보조 버퍼의 상위 끝(더 큰 인덱스)으로 밀려나도록 한다. 이 미디언(중앙값)은 한 레벨에 대한 트레시홀드 값을 계산하는 데 사용되며 트레시홀드 애플리케이션 블록은 도 9에 나타난 바와 같은 소프트 트레시홀드를 적용하도록 구현될 수 있다.
- [0108] 도 10은 소프트웨어 구현을 사용하여 다양한 수준의 웨이블릿 노이즈 제거 효과를 나타낸 그래프들이다. 도 10(a)는 잡음이 있는 신호이며, 도 10(b)는 Wavelet 잡음 제거 레벨 1을 적용한 소프트웨어 버전 잡음 제거 신호이며, 도 10(c)는 Wavelet 잡음 제거 레벨 2를 적용한 소프트웨어 버전 잡음 제거 신호이며, 도 10(d)는 Wavelet 잡음 제거 레벨 4를 적용한 소프트웨어 버전 잡음 제거 신호이다.
- [0109] 도 11은 하드웨어 구현을 사용하여 다양한 수준의 웨이블릿 노이즈 제거 효과를 나타낸 그래프이다. 도 11(a)는 잡음이 있는 신호이며, 도 11(b)는 Wavelet 잡음 제거 레벨 1을 적용한 하드웨어 버전 잡음 제거 신호이며, 도 11(c)는 Wavelet 잡음 제거 레벨 2를 적용한 하드웨어 버전 잡음 제거 신호이며, 도 11(d)는 Wavelet 잡음 제거 레벨 4를 적용한 하드웨어 버전 잡음 제거 신호이다.
- [0110] 도 12는 잡음 제거 IP 코어의 동작을 나타내는 타이밍 다이어그램이다.
- [0112] 도 10은 다양한 수준의 분해(디컴포지션, Conv)(1, 2, 4)를 사용한 웨이블릿 노이즈 제거의 소프트웨어 구현 결과를 보여준다. 특히 더 많은 레벨을 적용할수록 센서 신호가 더 명확해짐을 알 수 있다. 레벨 4 노이즈 제거를 통해 신호에 노이즈가 거의없고 신호의 피크가 PD 발생으로서 쉽게 식별된다. 제안된 아키텍처는 다양한 수준의 선택을 수용할 수 있을 뿐 아니라 프로세스의 각 단계에 대해 빠른 계산을 수행할 수 있다.
- [0113] 소프트웨어 구현 결과를 저장하고 하드웨어 구현에서 얻은 데이터와 비교하여 결과를 확인하는데 사용되었다. 도 11(b)에서 알 수 있듯이 모든 노이즈 제거 수준에서 두 가지 구현 유형의 차이는 매우 미미하고 육안으로는 거의 알아볼 수 없다. 도 11(c)에서 알 수 있듯이 하드웨어에서 얻은 데이터는 노이즈가 많은 원래 입력 데이터보다 훨씬 부드럽고 PD를 식별하기 쉽다. 레벨 4에서 하드웨어 구현의 노이즈 제거 데이터도 도 10(d)의 소프트웨어 버전에 대해 플로팅되었다. 두 데이터 표현은 대부분 겹쳤으며 데이터 양자화 프로세스 중 약간의 정확도 손실로 인해 하드웨어에서 작은 규모의 스파이크가 발생했다. 하드웨어 설계는 부호있는 16 비트 고정 소수점 데이터 유형(Q0.15)을 사용했으며 소프트웨어 버전은 64 비트 부동 소수점 데이터 유형을 사용하였다.
- [0114] 하기 표 2는 두 구현 간의 상관 계수의 높은 값을 나타내며 하드웨어 가속기의 정확성을 확인시키고 있다. 따라서, 획득된 결과는 웨이블릿 기반 잡음 제거의 하드웨어 버전의 정확성과 효율성을 나타낸다. 설계는 표 3에 표시된 하드웨어 리소스를 사용하여 FPGA에서 150MHz로 실행할 수 있다.

표 2

Denoise Level	Correlation Coefs
1	0.999881
2	0.997799
4	0.994126

[0115]

표 3

Resource	Quantity Used
LUTs	31,179
FFs	55,072
DSP	64
BRAM	0

[0116]

[0117]

노이즈 제거 IP는 1001 센서 데이터 스트림에 대해 작동을 완료하는데 1774 사이클(또는 11.84초)가 걸렸다.

[0118]

도 12는 컨볼루션, 데이터로드, 정렬 및 데이터 동기화와 같은 모든 활동에 대한 타이밍 정보를 나타낸 타이밍 다이어그램이다. $N_f = 1$ 은 디컴포지션(분해, Conv) 블록에 대한 초기 로딩 시간이고, $L_0 = 1001$ 및 $F = 8$ 은 각각 센서 데이터의 길이와 전체 필터 요소이다. 트레이스홀드 값을 찾기 위해 정렬한 후 지연주기를 삽입하여 데이터를 재컨볼루션 작업에 입력하였다. 여기에는 계산된 계수의 두 가지 세트인 세부(cd')과 근사(ca')가 필요하다. 각 레벨의 다운샘플링으로 인해 컨볼루션 시간은 4개 레벨에서 거의 동일했지만 노이즈 제거 코어의 지연 시간은 노이즈 레벨 증가에 따라 변경되지 않았다. 첫 번째 수준은 컨볼루션 및 정렬을 위한 요소수가 가장 많았으므로 다운 스트림 레벨(level 2, 3 및 4)이 실행되는 동안 가장 많은 작업 시간이 발생했다. 따라서 제안된 하드웨어 아키텍처는 더 많은 시간을 도입하지 않고도 더 나은 신호를 제공하는 이점을 달성하였다.

[0120]

도 13은 제안된 아키텍처의 상용화를위한 PCB 형태를 도시한다.

[0121]

도면에서 도시한 주요 구성 요소로는 이중 센서(압전 및 자외선), ADC 칩, AP SoC, BLE 칩, 확장용 헤더가 있다. 커버링 박스는 회로 요소를 보호하는 데 사용할 수 있으며, 관심 이벤트(예 : 부분 방전)에 대한 센서만 공개하였다.

[0122]

도시한 시스템(압전 및 자외선)의 이중 센서는 발전된 나노 기술을 적용하여 제작될 수 있다. 그러면, 크리스탈, 세라믹, 압전 센서 용 폴리머, 자외선 센서 용 진공 형 UV 감지기와 같은 일반 유형에 비해 이벤트 응답, 감도 및 정확도가 향상될 수 있다. 이러한 이중 센서에 본 발명의 사상에 따른 프로그래밍 가능 로직에 구현된 실시간 시스템 내 노이즈 제거 가속 기능을 적용하면, 감지 성능이 더욱 강력하고 안정적으로 된다.

[0123]

앞서 말한 바와 같이, 본 발명의 사상에 따른 실시간 임베디드 노이즈 처리 장치는, 전력 분야의 설비의 부분 방전을 감지하는 초음파(압전) 센서 및 광학 방식 센서로서 자외선 센서에 적용될 수 있다.

[0124]

이 경우, 실시간 임베디드 노이즈 처리 장치에 대한 상기 입력 신호는, 전력 설비의 부분 방전 감지를 위해 설치된 초음파 센서의 센싱 신호일 수 있다. 또는, 상기 입력 신호는, 전력 설비의 부분 방전 감지를 위해 설치된 광학 방식 센서의 센싱 신호일 수 있다.

[0125]

이러한 다중 센서를 결합하는 시스템은 부분 방전 이벤트 발생을 식별하기 위해 서로 다른 물리적 특성을 활용하여 이러한 유형의 실수를 저지를 가능성을 줄일 수 있다.

[0126]

또한, 상기 설명에서는 하나의 PD 센서에서의 1차원 센싱 신호에 대하여 노이즈 처리를 적용하는 것으로 구체화하였지만, 다수개의 PD 센서의 1차원 신호들을 조합하여 2차원 또는 3차원 집합 신호를 생성하여 노이즈 처리를 적용할 수 있다.

[0127]

또한, 상기 실시간 임베디드 노이즈 처리 장치는, 상기 출력 신호를 외부로 전송하기 위한 네트워크 전송 장치를 더 포함할 수 있다. 구현에 따라서는 상기 네트워크 전송 장치를 통해 입력 신호로서 센서 신호를 전송받을 수도 있다. 상기 네트워크 전송 장치에 의한 서버 통신은, Bluetooth Low Energy 모듈을 통한 ad hoc 통신 기능을 사용하여 소스 장치에서 대상 장치로 데이터를 전송할 수 있다.

[0129]

본 발명이 속하는 기술 분야의 당업자는 본 발명이 그 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있으므로, 이상에서 기술한 실시 예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로서 이해해야만 한다. 본 발명의 범위는 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내

어지며, 특허청구범위의 의미 및 범위 그리고 그 등가개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0131]

200, 201, 202, 203, 204 : 디컴포지셔

210 : 입력 라인 버퍼

220 : 컨벌루션 연산기

400, 401, 402, 403, 404 : 트레이스홀드

410 : 출력 라인 버퍼

422 : 정렬된 버퍼

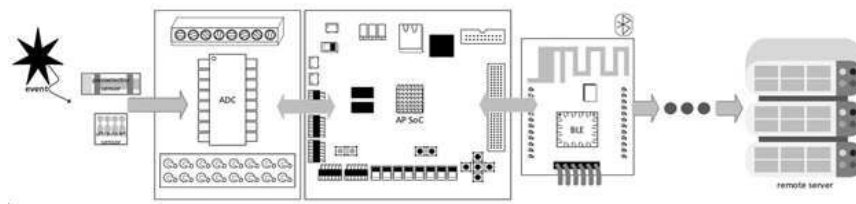
424 : 스왑 모듈

426 : 스왑된 버퍼

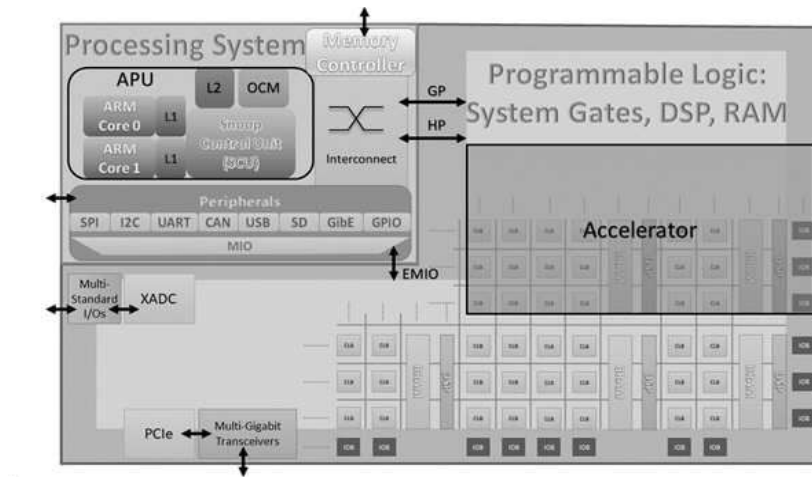
600, 601, 602, 603, 604 : 리컨스트럭처

도면

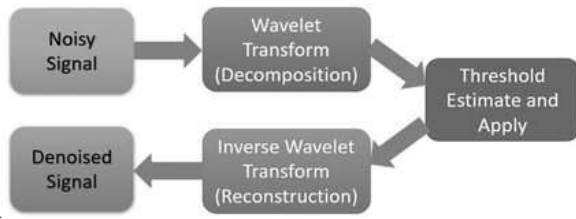
도면1



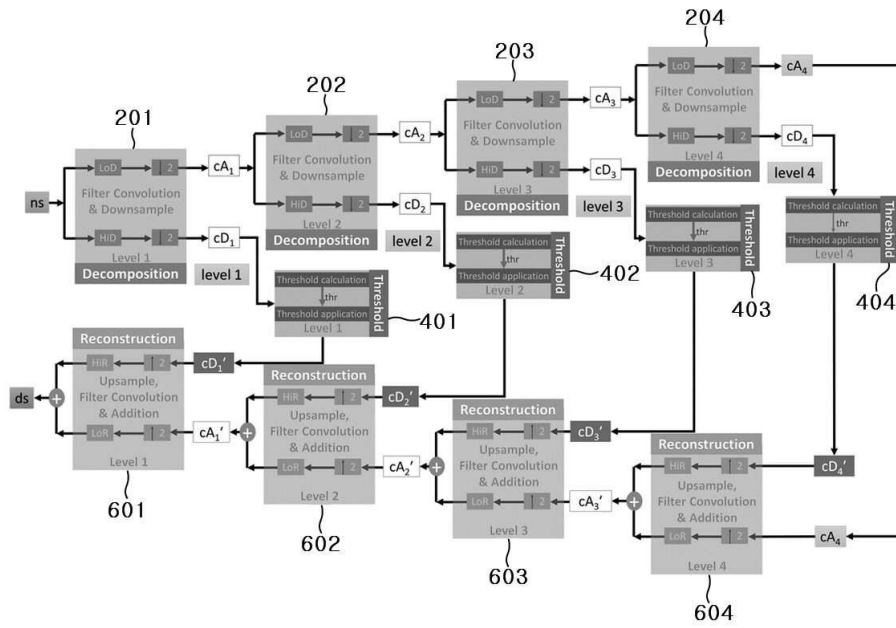
도면2



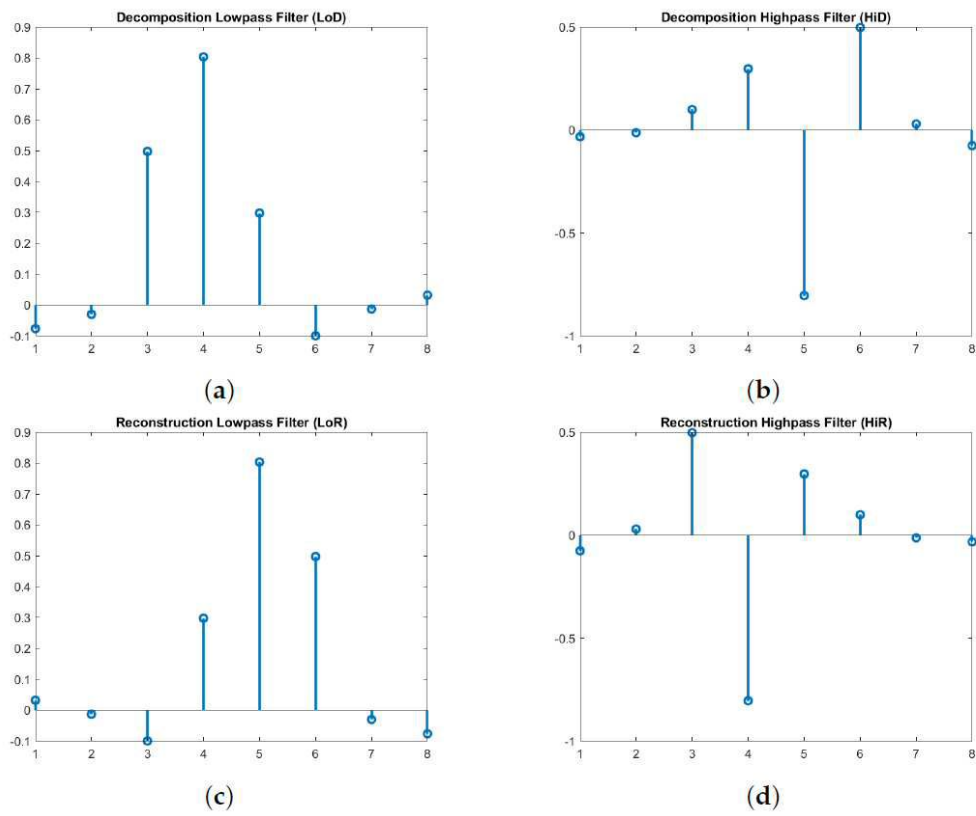
도면3



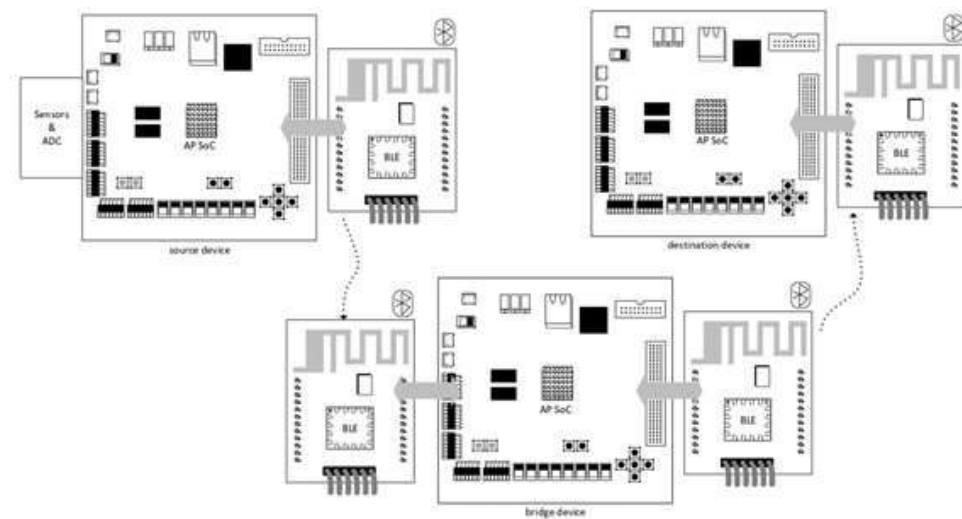
도면4



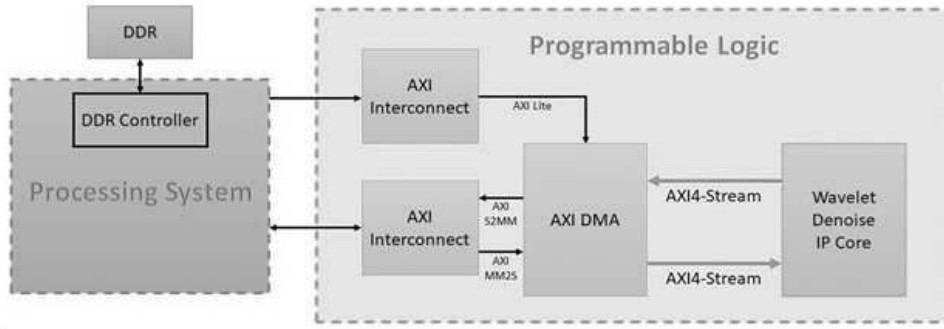
도면5



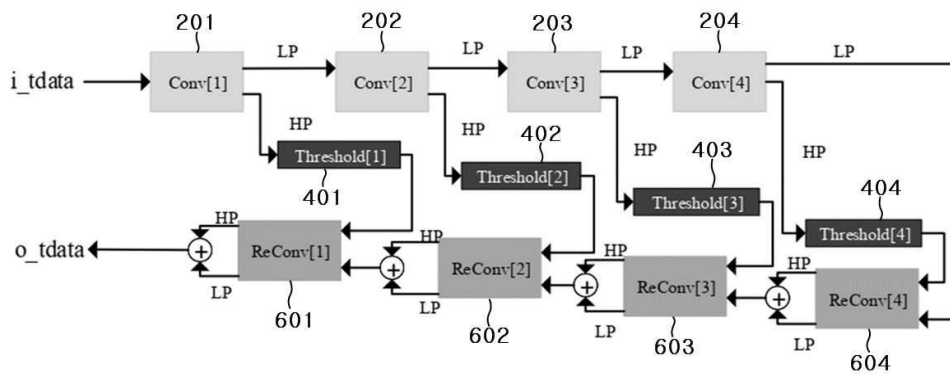
도면6a



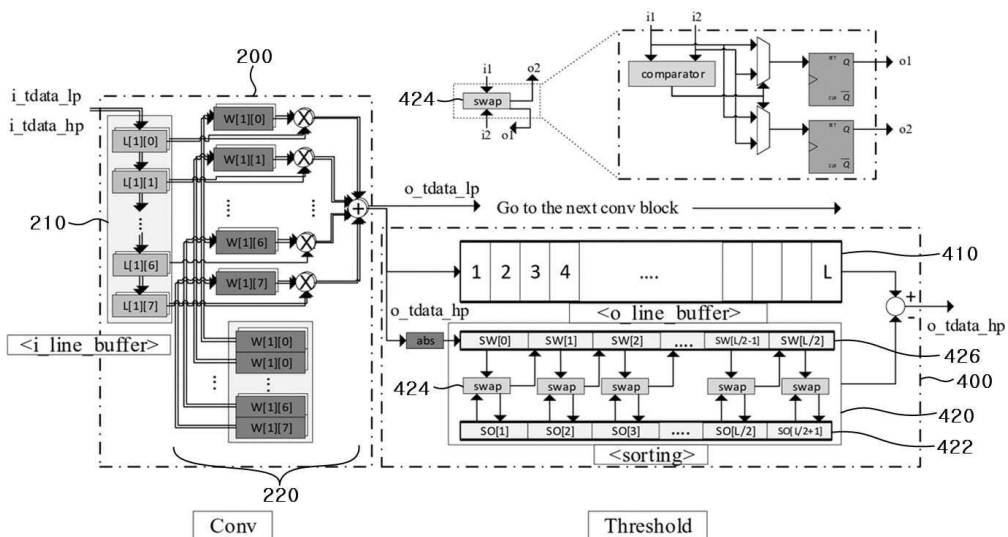
도면6b



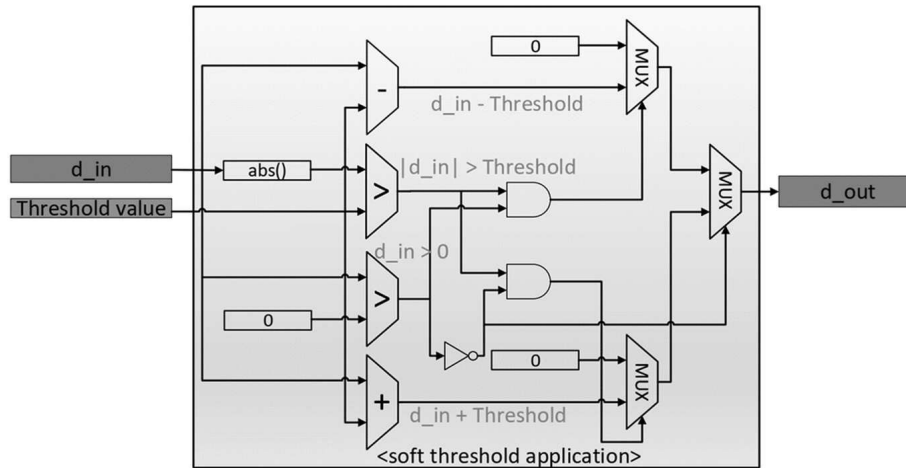
도면7



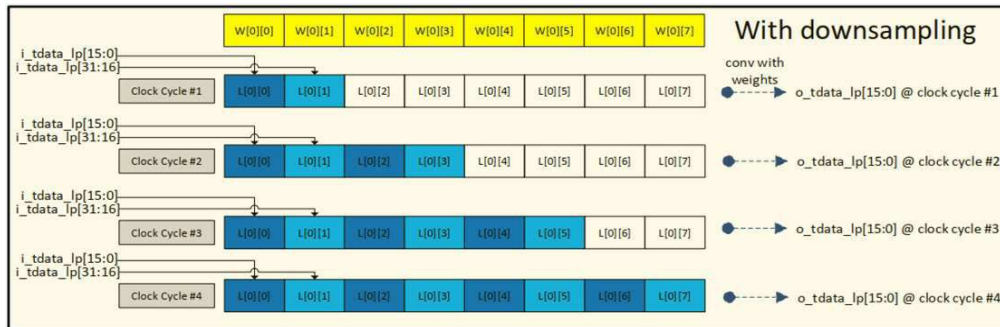
도면8a



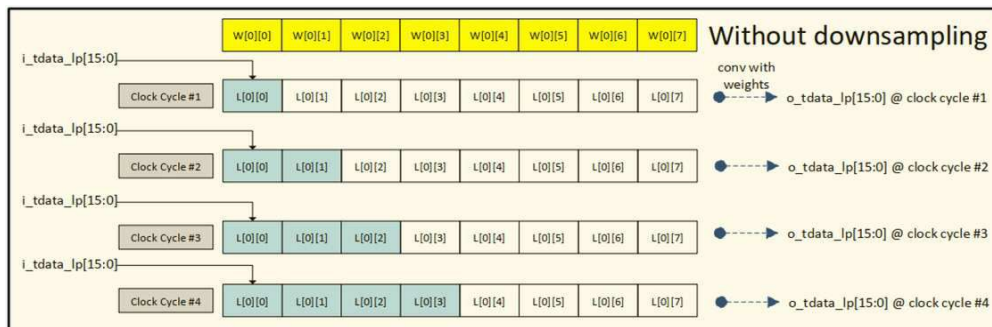
도면8b



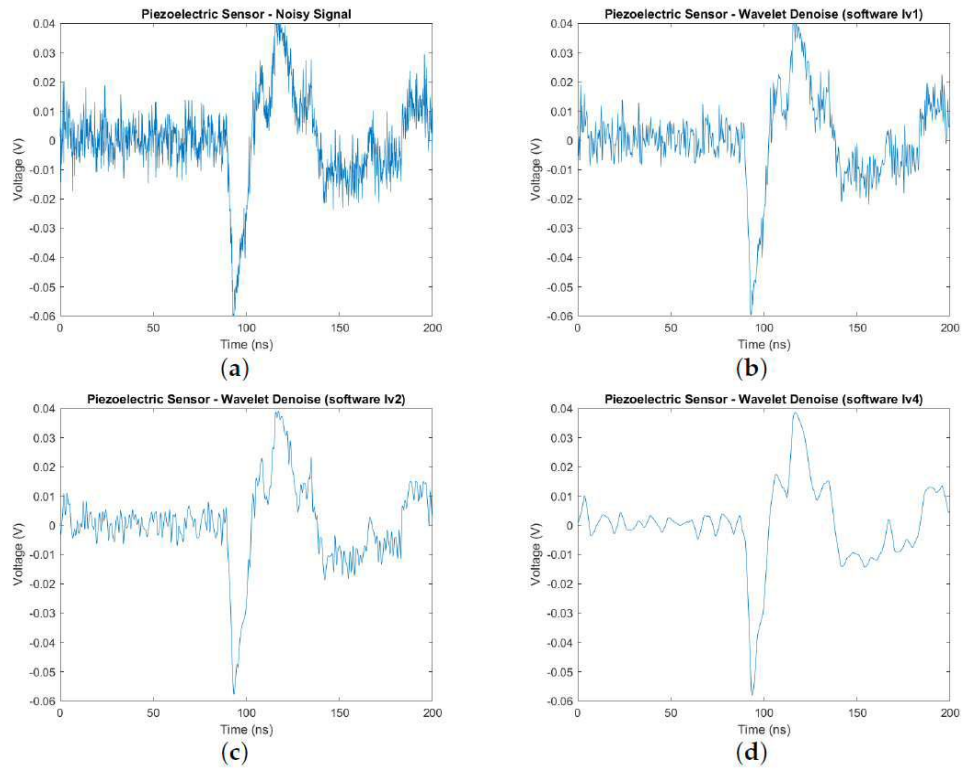
도면9a



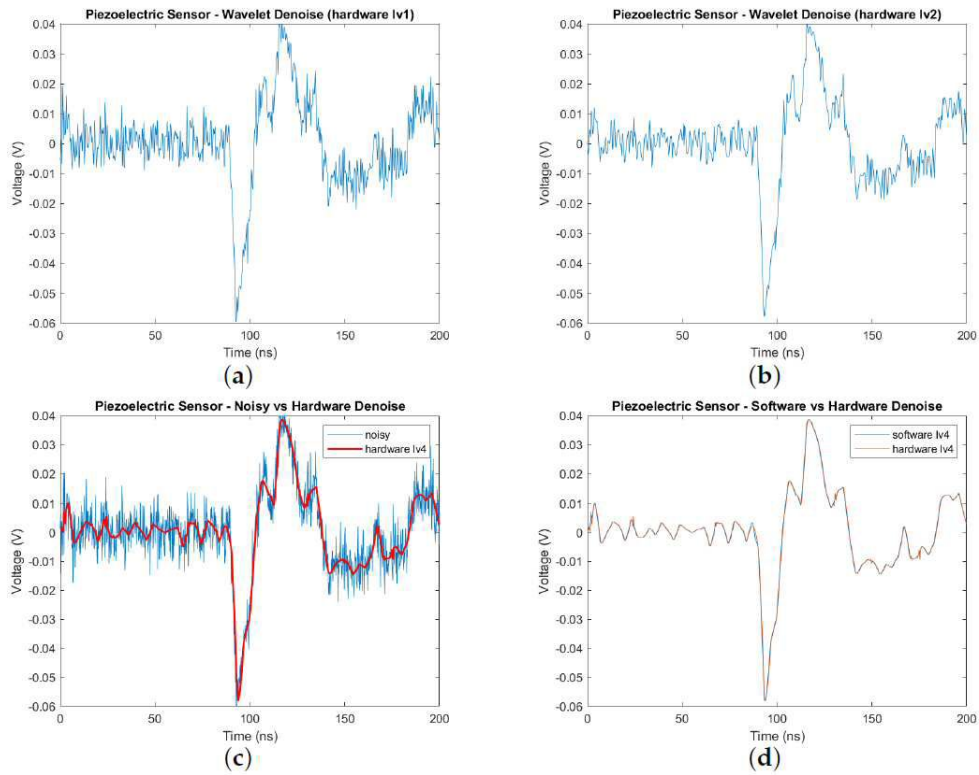
도면9b



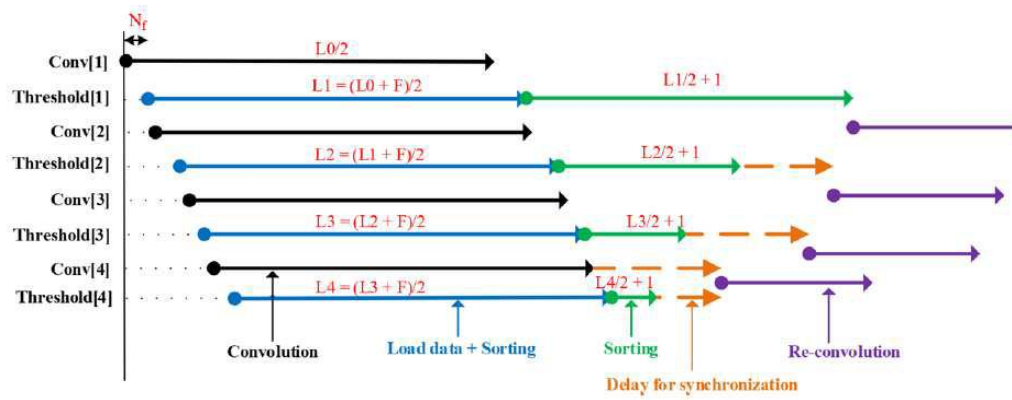
도면10



도면11



도면12



도면13

