



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0084290
(43) 공개일자 2014년07월04일

(51) 국제특허분류(Int. Cl.)
H04L 25/02 (2006.01)
(21) 출원번호 10-2014-7014136
(22) 출원일자(국제) 2012년10월26일
심사청구일자 없음
(85) 번역문제출일자 2014년05월26일
(86) 국제출원번호 PCT/US2012/062186
(87) 국제공개번호 WO 2013/063443
국제공개일자 2013년05월02일
(30) 우선권주장
61/552,242 2011년10월27일 미국(US)

(71) 출원인
엘에스아이 코퍼레이션
미국 캘리포니아주 95131, 새너제이, 라이더 파크
드라이브 1320
(72) 발명자
아자데트 카메란
미국 캘리포니아주 91101 파사데나 노스 엘 몰리
노 에비뉴-아파트먼트 #305 33
유 멩-린
미국 뉴저지주 07751 모건빌 요크셔 드라이브 135
(뒷면에 계속)
(74) 대리인
제일특허법인

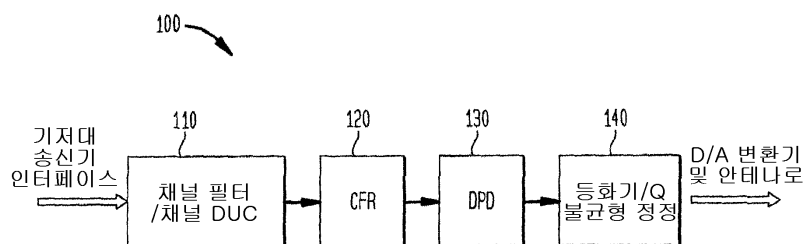
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **디지털 전치 왜곡(DPD) 및 다른 비선형 애플리케이션을 위해 사용자 정의의 비선형 함수와 함께 명령어 집합을 갖는 프로세서**

(57) 요약

디지털 전치 왜곡(DPD; digital pre-distortion)과 다른 비선형 애플리케이션을 위한 사용자 정의의 비선형 함수(user-defined non-linear functions)와 함께 명령어 집합을 갖는 프로세서가 제공된다. 예컨대 DPD같은 신호 처리 함수는 입력값 x 에 대해 적어도 하나의 비선형 함수를 수행하는 적어도 하나의 소프트웨어 명령어를 획득-상기 적어도 하나의 비선형 함수는 적어도 하나의 사용자 특정 변수(user-specified parameter)를 포함함-하고; 적어도 하나의 사용자 특정 변수를 갖는 적어도 하나의 비선형 함수를 위한 적어도 하나의 소프트웨어 명령어에 응답하여, 다음의 단계들, 즉, 입력값 x 에 비선형 함수를 적용하는 적어도 하나의 소프트웨어 명령어를 구현하는 적어도 하나의 기능성 유닛을 호출(invoking)하는 단계와, 상기 입력값 x 에 대한 비선형 함수에 대응하는 출력을 발생하는 단계를 수행함으로써 소프트웨어로 구현된다. 사용자 특정 변수는 메모리로부터 적어도 하나의 레지스터로 선택적으로 로딩될 수 있다.

대표도 - 도1



(72) 발명자

피나울트 스티븐 씨

미국 펜실베이니아주 18104 알렌타운 노스 31번 스트리트 127

윌리엄스 조셉

미국 뉴저지주 07733 홀름델 비컨 코트 8

모리나 알버트

스페인 이-03660 노벨다 (알리칸테) 4비 칼레 알칼데 마누엘 알베롤라 10

특허청구의 범위

청구항 1

신호 처리 기능을 소프트웨어로 구현하는 프로세서에 의해 수행되는 방법으로서,

입력값 x 에 대해 적어도 하나의 비선형 함수를 수행하는 적어도 하나의 소프트웨어 명령어를 획득하는 단계—상기 적어도 하나의 비선형 함수는 적어도 하나의 사용자 특정 변수(user-specified parameter)를 포함함—와,

적어도 하나의 사용자 특정 변수를 갖는 적어도 하나의 비선형 함수를 위한 상기 소프트웨어 명령어 중 적어도 하나의 소프트웨어 명령어에 응답하여,

상기 입력값 x 에 상기 비선형 함수를 적용하기 위해 상기 적어도 하나의 소프트웨어 명령어를 구현하는 적어도 하나의 기능성 유닛을 호출(invoking)하는 단계와,

상기 입력값 x 에 대해 상기 비선형 함수에 대응하는 출력을 발생하는 단계를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 신호 처리 기능은 디지털 전치 왜곡(digital pre-distortion)을 포함하는

방법.

청구항 3

제1항에 있어서,

상기 적어도 하나의 사용자 특정 변수를 메모리로부터 적어도 하나의 레지스터로 로딩하는 단계를 더 포함하는

방법.

청구항 4

제1항에 있어서,

상기 사용자 특정 변수는 유한 개수의 입력값에 대한 상기 비선형 함수의 값을 저장하는 룩업 테이블(look-up table)을 포함하는

방법.

청구항 5

제4항에 있어서,

상기 사용자 특정 변수는 상기 룩업 테이블의 엔트리들 사이에 다항식 보간(polynomial interpolation)을 수행하기 위해 상기 비선형 함수에 의해 채용되는 하나 이상의 계수를 포함하는

방법.

청구항 6

제1항에 있어서,
상기 프로세서는 벡터 프로세서를 포함하는
방법.

청구항 7

제1항에 있어서,
상기 입력값 x 는 벡터를 포함하고, 상기 출력은 벡터를 포함하는
방법.

청구항 8

신호 처리 기능을 소프트웨어로 구현하도록 구성된 프로세서로서,
메모리와,
상기 메모리에 결합된 적어도 하나의 하드웨어 장치를 포함하되,
상기 적어도 하나의 하드웨어 장치는,

입력값 x 에 대해 적어도 하나의 비선형 함수를 수행하는 적어도 하나의 소프트웨어 명령어를 획득—상기 적어도 하나의 비선형 함수는 적어도 하나의 사용자 특정 변수를 포함함—하고,

적어도 하나의 사용자 특정 변수를 갖는 적어도 하나의 비선형 함수를 위한 상기 소프트웨어 명령어 중 적어도 하나의 명령어에 응답하여,

상기 입력값 x 에 상기 비선형 함수를 적용하기 위해 상기 적어도 하나의 소프트웨어 명령어를 구현하는 적어도 하나의 기능성 유닛을 호출하고,

상기 입력값 x 에 대해 상기 비선형 함수에 대응하는 출력을 발생하도록 동작하는
프로세서.

청구항 9

제8항에 있어서,
상기 신호 처리 기능은 디지털 전치 왜곡을 포함하는
프로세서.

청구항 10

제8항에 있어서,
상기 적어도 하나의 하드웨어 장치는 상기 적어도 하나의 사용자 특정 변수를 메모리로부터 적어도 하나의 레지스터로 로딩하도록 또한 구성되는
프로세서.

청구항 11

제8항에 있어서,

상기 사용자 특정 변수는 유한 개수의 입력값에 대한 상기 비선형 함수의 값을 저장하는 룩업 테이블을 포함하는

프로세서.

청구항 12

제11항에 있어서,

상기 사용자 특정 변수는 상기 룩업 테이블의 엔트리들 사이에 다항식 보간을 수행하기 위해 상기 비선형 함수에 의해 채용되는 하나 이상의 계수를 포함하는

프로세서.

청구항 13

제8항에 있어서,

상기 프로세서는 벡터 프로세서를 포함하는

프로세서.

청구항 14

제8항에 있어서,

상기 입력값 x 는 벡터를 포함하고, 상기 출력은 벡터를 포함하는

프로세서.

청구항 15

디지털 전치 왜곡을 소프트웨어로 구현하는 프로세서에 의해 수행되는 방법으로서,

입력값 x 에 대해 적어도 하나의 비선형 함수를 수행하는 적어도 하나의 소프트웨어 명령어를 획득하는 단계—상기 적어도 하나의 비선형 함수는 적어도 하나의 사용자 특정 변수를 포함함—와,

적어도 하나의 사용자 특정 변수를 갖는 적어도 하나의 비선형 함수를 위한 상기 소프트웨어 명령어 중 적어도 하나의 소프트웨어 명령어에 응답하여,

상기 입력값 x 에 상기 비선형 함수를 적용하기 위해 상기 적어도 하나의 소프트웨어 명령어를 구현하는 적어도 하나의 기능성 유닛을 호출하는 단계와,

상기 입력값 x 에 대해 상기 비선형 함수에 대응하는 출력을 발생하는 단계를 포함하는 방법.

청구항 16

제15항에 있어서,

상기 적어도 하나의 사용자 특정 변수를 메모리로부터 적어도 하나의 레지스터로 로딩하는 단계를 더 포함하는 방법.

청구항 17

제15항에 있어서,
 상기 사용자 특정 변수는 유한 개수의 입력값에 대한 상기 비선형 함수의 값을 저장하는 룩업 테이블을 포함하는 방법.

청구항 18

제17항에 있어서,
 상기 사용자 특정 변수는 상기 룩업 테이블의 엔트리들 사이에 다항식 보간을 수행하기 위해 상기 비선형 함수에 의해 채용되는 하나 이상의 계수를 포함하는 방법.

청구항 19

제15항에 있어서,
 상기 프로세서는 벡터 프로세서를 포함하는 방법.

청구항 20

제15항에 있어서,
 상기 입력값 x 는 벡터를 포함하고, 상기 출력은 벡터를 포함하는 방법.

명세서

기술분야

- [0001] 관련 출원에 대한 상호 참조
- [0002] 본 출원은 "Software Digital Front End(SoftDFE) Signal Processing and Digital Radio"라는 명칭으로 2011년 10월 27일에 미국 출원된 미국 특허 가출원번호 제61/552,242호를 우선권 주장하며, 그 내용은 본원에서 참조하고 있다.
- [0003] 발명의 분야
- [0004] 본 발명은 디지털 신호 처리 기술에 관한 것으로, 더 구체적으로는 사용자 정의의 비선형 함수(user-defined non-linear functions)를 평가하는 기술에 관한 것이다.

배경 기술

- [0005] 디지털 전치 왜곡(DPD; digital pre-distortion)은 전력 증폭기의 효율을 개선하기 위해 송신기의 전력 증폭기를 선형화하는데 이용되는 기술이다. 송신기의 전력 증폭기는 전형적으로 거의 선형이어야 하고, 따라서 신호는 정확하게 재생된다. 입력 신호의 압축(compression) 또는 입력 신호와 출력 신호 사이의 비선형 관계는 출력 신호 스펙트럼이 인접 채널들로 누설되게 하여, 간섭을 유발한다. 이러한 효과는 흔히 스펙트럼 재생장(spectral re-growth)이라고 한다.

- [0006] 디지털 전치 왜곡 회로는 전력 증폭기의 이득과 위상 특성을 역으로 모델링하며, 이것이 증폭기와 조합될 때, 전체 시스템을 보다 선형적이게 하면서 전력 증폭기에 의해 초래될 수 있는 왜곡이 감소되게 한다. 역 왜곡 (inverse distortion)은 증폭기의 입력에 도입되고, 이로써 발생 가능성이 있던 증폭기의 임의의 비선형을 감소시킨다.
- [0007] 디지털 전치 왜곡은 전형적으로 높은 샘플링율로 인해 하드와이어드 로직(hardwired logic)을 이용하여 구현된다. 이러한 하드웨어 기반 DPD 기술이 전력 증폭기를 효율적으로 선형화하지만, 전력 증폭기는 여전히 많은 제약을 받는데, 만약 이런 제약들이 극복된다면, DPD 회로의 효율성과 융통성은 크게 개선될 수 있다. 예를 들어, 기존의 하드웨어 기반 DPD 기술은 새로운 RF 디자인을 위해 DFE 설계를 수정하는데 융통성이 부족하고, 비용도 비싸며, 시간 소모적인 어려움을 겪는다.
- [0008] 디지털 전치 왜곡 및 다른 비선형 애플리케이션은 예컨대 필터 계수값 또는 룩업 테이블(look-up table)의 값처럼 사용자에 의해 특정된 하나 이상의 변수를 포함하는 하나 이상의 비선형 함수를 종종 처리해야 한다. 그러므로, 예컨대 DPD의 고성능 소프트웨어 구현을 보장하기 위해 디지털 전치 왜곡(DPD)과 다른 비선형 애플리케이션을 위한 하나 이상의 사용자 정의의 비선형 함수와 함께 명령어 집합을 갖는 프로세서가 필요하다.

발명의 내용

- [0009] 전반적으로, 디지털 전치 왜곡(DPD)과 다른 비선형 애플리케이션을 위한 하나 이상의 사용자 정의의 비선형 함수와 함께 명령어 집합을 갖는 프로세서가 제공된다. 본 발명의 한 양상에 따르면, 예컨대 DPD같은 신호 처리 함수는 입력값 x 에 대해 적어도 하나의 비선형 함수를 수행하는 적어도 하나의 소프트웨어 명령어를 획득—상기 적어도 하나의 비선형 함수는 적어도 하나의 사용자 특정 변수(user-specified parameter)를 포함함—하고; 적어도 하나의 사용자 특정 변수를 갖는 적어도 하나의 비선형 함수를 위한 적어도 하나의 소프트웨어 명령어에 응답하여, 다음의 단계들, 즉, 입력값 x 에 비선형 함수를 적용하는 적어도 하나의 소프트웨어 명령어를 구현하는 적어도 하나의 기능성 유닛을 호출(invoking)하는 단계와, 상기 입력값 x 에 대한 비선형 함수에 대응하는 출력을 발생하는 단계를 수행함으로써 소프트웨어로 구현된다.
- [0010] 또한, 사용자 특정 변수는 메모리로부터 적어도 하나의 레지스터로 로드될 수 있다. 사용자 특정 변수는 예컨대 유한 개수의 입력값에 대한 비선형 함수의 값들을 저장하는 룩업 테이블을 포함한다. 사용자 특정 변수는 룩업 테이블의 엔트리 사이에 다항식 보간(polynomial interpolation)을 수행하기 위해 비선형 함수에 의해 활용되는 하나 이상의 계수들 포함할 수도 있다.
- [0011] 본 발명의 더 많은 특징 및 이점들과 더불어 본 발명에 대한 보다 완전한 이해는 이후의 상세한 설명과 도면을 참조하여 가능할 것이다.

도면의 간단한 설명

- [0012] 도 1은 본 발명의 양상들이 채용될 수 있는 예시적인 송신기의 부분들을 예시한다.
- 도 2는 본 발명의 양상들이 채용될 수 있는 다른 예시적인 송신기의 부분들을 예시한다.
- 도 3은 사용자 정의의 비선형 명령어 $f_{m,1}$ 을 이용하여 16개 성분 벡터의 벡터 프로세서에 대해 DPD 기능을 소프트웨어로 구현하는 예시적인 의사 코드(pseudo code)를 예시한다.
- 도 4a 및 도 4b는 예시적인 함수 블록도의 그래픽적인 예시이다.
- 도 5a는 $x(n)$ 의 함수로서 개별적인 사용자 정의의 비선형 명령어 $f_{m,1}$ 을 예시한다.
- 도 5b는 도 5a의 개별적인 사용자 정의의 비선형 명령어 $f_{m,1}$ 의 예시적인 근사(approximation)를 예시한다.
- 도 6은 테일러 총합 계산 블록(Taylor Sum computation block)을 예시한다.
- 도 7은 본 발명의 일 실시예에 따라 하나 이상의 복소수에 대해 동시에 사용자 정의의 비선형 함수를 평가하는 실시예적인 벡터 기반 디지털 프로세서의 개략적인 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0013] 도 1은 본 발명의 양상들이 채용될 수 있는 예시적인 송신기(100)의 부분을 예시한다. 도 1에 도시된 바와 같이, 예시적인 송신기 부분(100)은 채널 필터 및 디지털 상향 변환(DUC; digital up conversion) 스테이지(110), 과고율 저감(CFR) 스테이지(120), 디지털 전치 왜곡(DPD) 스테이지(130) 및 선택성 등화 및/또는 IQ 불균형 정정 스테이지(140)를 갖는다. 일반적으로, 채널 필터 및 디지털 상향 변환 스테이지(110)는 예컨대 디지털화된 기저대 신호를 무선 주파수(RF)로 변환하기 위해 유한 임펄스 응답(FIR) 필터와 디지털 상향 변환을 이용하여 채널 필터링을 수행한다. 위에서 지적했듯이, 과고율 저감 스테이지(120)는 송신된 신호의 PAR(peak-to-average ratio)을 제한한다. 디지털 전치 왜곡 스테이지(130)는 전력 증폭기를 선형화하여 효율을 개선한다. 등화 스테이지(140)는 RF 채널 등화를 채용하여 채널 장애를 완화시킨다.
- [0014] 본 발명의 한 양상에 따르면, 디지털 전치 왜곡 비선형 처리 및 다른 비선형 애플리케이션은 DPD 및 다른 비선형 애플리케이션을 위한 하나 이상의 사용자 정의의 비선형 함수와 함께 명령어 집합을 갖는 프로세서에서 소프트웨어로 수행된다. 사용자 정의의 비선형 명령어는 사용자에게 의해 특정되어야 하는 적어도 하나의 변수를 갖는 비선형 함수를 계산하는데 이용된다. 사용자 정의의 비선형 명령어는 입력 스칼라 또는 벡터와 함께 수신되어, 출력 스칼라 또는 벡터를 발생한다. 입력 벡터인 경우, 출력 벡터는 입력 샘플의 비선형 함수인 출력 샘플들을 포함한다.
- [0015] 본 발명이 디지털 전치 왜곡에 관해서 예시되었지만, 본 발명은 하나 이상의 사용자 정의의 비선형 함수를 채용하는 모든 비선형 애플리케이션에 이용될 수 있다.
- [0016] 본 발명은 핸드세트(handsets), 기지국 및 다른 네트워크 요소에 적용될 수 있다.
- [0017] 도 2는 본 발명의 양상들이 채용될 수 있는 다른 예시적인 송신기(200)의 부분들을 예시한다. 도 2에 도시된 것처럼, 두 개의 펄스 성형(pulse shaping) 및 저역 통과 필터(LPF) 스테이지(210-1, 210-2)와, 복소 신호 I, Q를 처리하는 두 개의 디지털 상향 변환기(220-1, 220-2)를 포함한다. 도 2의 예시적인 송신기 부분(200)은 도 1의 과고율 저감 스테이지(120)을 포함하지 않지만, CFR 스테이지는 선택 사양으로서 포함될 수 있다. 다음, 복소 입력(I, Q)은 도 2의 디지털 전치 왜곡기(pre-distorter)(230)에 인가되는데, 이것이 본 발명의 예시적인 실시예에서 주목할 부분이다. 도 2의 디지털 전치 왜곡기(230)는 이후에 예컨대 도 3 및 도 4와 함께 더 자세히 설명된다.
- [0018] 디지털 전치 왜곡기(230)의 출력은 두 개의 디지털-아날로그 변환기(DAC)(240-1, 240-2)에 병렬로 인가되고, 아날로그 신호들은 신호들을 RF 신호로 더욱 상향 변환시키는 직교 변조 스테이지(250)에 의해 처리된다.
- [0019] 직교 변조 스테이지(250)의 출력(255)은 예컨대 도허티 증폭기(Doherty amplifier) 또는 드레인 변조기(drain modulator)같은 전력 증폭기(260)로 인가된다. 위에서 지적하였듯이, 디지털 전치 왜곡기(230)는 전력 증폭기의 선형 범위를 더 높은 송신 전력까지 확대함으로써 전력 증폭기(260)의 효율을 개선하도록 전력 증폭기(260)를 선형한다.
- [0020] 피드백 경로(265)에서, 전력 증폭기(260)의 출력은 신호를 기저대역으로 하향 변환시키는 복조 스테이지(280)에 인가되기 전에 감쇠기(270)에 인가된다. 하향 변환된 신호는 신호를 디지털화하는 ADC(290)로 인가된다. 다음, 디지털화된 샘플은 디지털 전치 왜곡기(230)를 위한 변수 w 를 발생시키는 복소 적응 알고리즘(295)에 의해 처리된다. 복소 적응 알고리즘(295)은 본 출원의 범주 밖이다. 공지 기술들이 디지털 전치 왜곡기(230)를 위한 변수들을 발생하기 위해 채용될 수 있다.
- [0021] *디지털 전치 왜곡기의 비선형 필터 구현*
- [0022] 디지털 전치 왜곡기(230)는 비선형 시스템의 볼테라 급수(Volterra series) 모델을 이용하여 비선형 필터로 구현될 수 있다. 볼테라 급수는 테일러 급수와 유사한 방식의 비선형 동작을 위한 모델이다. 볼테라 급수와 테일러 급수의 차이점은 볼테라 급수가 "메모리" 효과를 포착할 수 있다는 점이다. 특정 시간에 비선형 시스템의 출력이 입력에 정확히 따르는 경우라면, 테일러 급수는 주어진 입력에 대한 비선형 시스템의 응답을 근사화하는데 이용될 수 있다. 볼테라 급수에서는, 그 외의 시간에 비선형 시스템의 출력이 시스템에 대한 입력에 따른다. 따라서, 볼테라 급수는 장치의 "메모리" 효과가 포착될 수 있게 한다.

[0023] 일반적으로, 메모리를 갖는 인과성 시스템(causal system)은 다음과 같이 표현될 수 있다.

[0024]
$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$$

[0025] 또한, 메모리가 없는 약 비선형 시스템(weakly non-linear system)은 다항식을 이용해 다음과 같이 모델링될 수 있다.

[0026]
$$y(t) = \sum_{k=1}^{\infty} a_k [x(t)]^k$$

[0027] 볼테라 급수는 다음의 두 개의 식의 조합으로 간주될 수 있다.

[0028]
$$y(t) = \sum_{k=1}^K y_k(t)$$

[0028]
$$y_k(t) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_k(\tau_1, \dots, \tau_k)x(t-\tau_1)\dots x(t-\tau_k)d\tau_1\dots d\tau_k$$

[0029] 이산 영역(descrete domain)에서, 볼테라 급수는 다음과 같이 표현될 수 있다.

[0030]
$$y(n) = \sum_{k=1}^K y_k(n)$$

[0030]
$$y_k(n) = \sum_{m_1=0}^{M-1} \dots \sum_{m_k=0}^{M-1} h_k(m_1, \dots, m_k) \prod_{l=1}^k x(n-m_l)$$

[0031] 볼테라 급수의 복잡성은 지수적으로 증가하여, DPD같은 많은 일반적인 응용에서 그 이용을 비실용적으로 만들 수 있다. 그러므로, 비선형 시스템에 대한 다수의 간략화된 모델이 제안되었다. 예를 들면, 메모리 다항식 모델이 흔히 이용되는 모델이다.

[0032]
$$y_{MP}(n) = \sum_{k=1}^K \sum_{m=0}^{M-1} h_k(m, \dots, m)x^k(n-m)$$

$$= \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} h_{km}x(n-m) |x(n-m)|^k$$

[0033] 일반화 메모리 다항식 모델로 불리는 다른 간략화된 모델은 다음과 같이 표현될 수 있다(여기서 M은 메모리 용량(memory depth)을 나타내고, K는 다항식 차수를 나타낸다).

[0034]
$$y(n) = \sum_{m=0}^{M-1} \sum_{l=0}^{M-1} \sum_{k=0}^{K-1} h_{k,m,l} |x(n-l)|^k x(n-m)$$

$$y(n) = \sum_{m=0}^{M-1} \sum_{l=0}^{M-1} x(n-m) \sum_{k=0}^{K-1} h_{k,m,l} |x(n-l)|^k$$

[0035] 외적(cross-product)과 함께 일반화 메모리 다항식의 등가식은 다음과 같이 표현된다.

수학식 1

[0036]
$$y(n) = \sum_{m=0}^{M-1} \sum_{l=0}^{M-1} x(n-m) \cdot f_{m,l}(|x(n-l)|)$$

[0037] 여기서 $f_{m,l}(|x(n-l)|)$ 은 다음과 같다.

수학식 2

$$f_{m,l}(|x(n-l)|) = \sum_{k=0}^{K-1} h_{k,m,l} |x(n-l)|^k$$

[0038]

[0039]

여기서 $f(x)$ 는 사용자 정의의 비선형 명령어 `vec_nl`을 이용해 본 발명의 한 양상에 따라 가속화될 것으로 가정되는 하나 이상의 사용자 특정 변수를 갖는 비선형 함수로, 이와 관련해서는 다음에 논의된다. 비선형 분해(non-linear decomposition)를 위해 x_k 가 아닌 다른 기저 함수도 가능함을 주목해야 한다.

[0040]

이후에 논의되듯이, 사용자 정의의 비선형 명령어 $f_{m,l}$ 는 예컨대 벡터 프로세서에 의해 처리될 수 있다. $f_{m,l}$ 은 $m \times 1$ 어레이의 비선형 함수이다. 각각의 비선형 함수는 예컨대 룩업 테이블 또는 계수같은 사용자 특정 변수를 가질 수 있다. 룩업 테이블은 사용자 정의 비선형 명령어 $f_{m,l}$ 의 다항식 근사(polynomial approximation)일 수 있다. 도 7과 함께 이후에 더 자세히 설명되듯이, $m \times 1$ 어레이의 각각의 사용자 정의 비선형 명령어 $f_{m,l}$ 에 대한 룩업 테이블은 메모리에 저장될 수 있고, 명령어가 프로세서에 의해 처리될 때 기능성 유닛과 연계된 레지스터로 로드될 수 있다. 이후, 입력 샘플은 $m \times 1$ 어레이의 개별적인 비선형 명령어 $f_{m,l}$ 에 대해 평가될 수 있다.

[0041]

도 3은 수학식(1)의 사용자 정의의 비선형 명령어 $f_{m,l}$ 을 이용하여 16개 성분 벡터의 벡터 프로세서상에서 소프트웨어로 DPD 기능을 구현하기 위한 예시적인 의사 코드(300)를 예시한다. 예시적인 의사 코드(300)는 입력 x 의 크기를 계산하는 제 1 부분(310)을 포함한다. 라인(320)에서, $m \times 1$ 어레이의 개별적인 비선형 명령어 $f_{m,l}$ 은 레지스터로 로드될 수 있다. 그 이후, 예시적인 의사 코드(300)는 수학식(1)을 구현하는 부분(330)을 포함한다(예컨대, 샘플을 입력하기, 샘플에 대한 자승 연산 수행하기, 비선형 함수를 계산한 후 그 결과를 곱셈-누적하기).

[0042]

도 4a는 수학식(1)을 구현하는 예시적인 함수 블록도(400)의 그래프적인 예시이다. 본원에 설명된 예시적인 실시예에서, $|x|^{2k}$ 는 $|x|^k$ 를 대신하여 이용된다. 도 4a에 도시된 바와 같이, 예시적인 회로(400)는 복수의 지연 소자를 포함하는데, 예컨대 자승 연산(410)의 출력을 지연시킴으로써 수학식(1)의 $x(n-m)$ 항목을 발생하는 지연 소자(405-1 내지 405-5)와, 수학식(2)의 $|x(n-1)|^2$ 항목을 발생하는 지연 소자(405-6 내지 405-9)를 포함한다. 또한, 예시적인 함수 블록도(400)는 기능성 유닛(420-1,1 내지 420-4,4)의 어레이를 포함하는데, 이들은 적절한 $|x(n-1)|^2$ 항목을 수신하여 수학식(2)를 구현한다. 예시적인 함수 블록도(400)는 또한 복수의 곱셈기(x)를 포함하는데, 이 곱셈기들은 적절한 $x(n-m)$ 항목을 수신하고, 이 항목을 대응하는 m,l 기능성 유닛(420)의 출력과 곱한다. 각각의 로우(row)에서의 곱셈의 출력은 덧셈기(+)(430)에 의해 더해지고, 소정의 로우의 각각의 덧셈기(430)의 출력들은 대응하는 가산기(440)에 의해 합산되어 출력 $y(n)$ 를 발생한다.

[0043]

도 4b는 감축된 개수의 곱셈 연산으로 수학식(1)을 구현하는 예시적인 대안의 함수 블록도(450)의 그래프적인 예시(450)이다. 도 4b에 도시된 것처럼, 예시적인 회로(450)는 복수의 지연 소자를 포함하는데, 예컨대 자승 연산(460)의 출력을 지연시킴으로써 수학식(1)의 $x(n-m)$ 항목을 발생하는 지연 소자(455-1 내지 455-5)와, 수학식(2)의 $|x(n-1)|^2$ 항목을 발생하는 지연 소자(455-7 내지 455-9)를 포함한다. 또한, 예시적인 함수 블록도(450)는 기능성 유닛(470-1,1 내지 470-4,4)의 어레이를 포함하는데, 이들은 적절한 $|x(n-1)|^2$ 항목을 수신하여 수학식(2)를 구현한다. 덧셈기(480)는 비선형 이득(입력의 크기의 비선형 함수의 합)을 계산한다.

[0044]

예시적인 함수 블록도(450)는 또한 복수의 곱셈기(x)(475)를 포함하는데, 이 곱셈기들은 적절한 $x(n-m)$ 항목을 수신하고, 이 항목을 대응하는 m,l 기능성 유닛(470)의 컬럼(column)의 합산된 출력의 출력과 곱한다. 이런 방식으로, 덧셈기(480)으로부터의 비선형 이득이 입력 데이터에 인가된다(복소 곱셈-누적(CMAC) 연산). 곱셈의 출력은 덧셈기(+)(485)에 의해 더해져서, 출력 $y(n)$ 을 발생한다.

[0045]

도 5a는 $x(n)$ 의 함수로서 개별적인 사용자 정의의 비선형 명령어 $f_{m,l}$ (500)을 예시한다. 도 5b는 도 5a의 개별적인 사용자 정의의 비선형 명령어 $f_{m,l}$ 의 예시적인 근사(550)를 예시한다. 도 5b의 예시적인 근사(550)는 세그먼트화된 테일러 급수 룩업 테이블(segmented Taylor series look-up tables)을 이용한다. 비선형 함수

$f_{m,1}(500)$ 는 j 개 세그먼트로 분해된다. 각각의 세그먼트와 연관된 샘플(560-1 내지 560- j)는 룩업 테이블에 저장된다. 만약 샘플이 소정의 x 에 대해 룩업 테이블에 저장된다면, 이 샘플은 룩업 테이블로부터 검색될 수 있고, 비선형 함수 평가에 직접적으로 활용될 수 있다. 만약 원하는 x 가 룩업 테이블의 2개의 값 사이에 있다면, 선형 보간 또는 보다 일반적으로는 테일러 급수 기반의 보간이 기능성 유닛 내부의 하드웨어에서 수행되어 결과물을 얻는데, 이것은 도 6과 함께 이후에 더 자세히 설명된다. 이런 방식으로, 비선형 디지털 전치 왜곡 동작은 입력 신호(550)의 상이한 세그먼트에서의 테일러 급수 계수들에 의해 설명될 수 있다. 32개의 세그먼트를 갖는 일 실시예로, 4개의 3차 근사 계수(cubic polynomial approximations coefficients)를 이용하여 표현된 계수들인 경우, 룩업 테이블에는 128개의 복소 엔트리(16비트 복소수 및 16비트 실수)가 존재한다. 128개 세그먼트를 갖고 세그먼트 당 하나의 계수를 갖는 또다른 변형안에서는, 선형 보간을 위해 128개 복소 계수들(16비트 복소수 및 16비트 실수)이 존재한다. 선택적으로, 세그먼트에 대해 32개 복소 엔트리는 3차 보간의 경우는 세그먼트당 4개 계수를 갖는다.

[0046] 위에서 지적하였듯이, 만약 원하는 x 값이 룩업 테이블에 존재하지 않고 오히려 이 룩업 테이블내의 2개의 값 사이에 있다면, 결과물을 얻기 위해 기능성 유닛 내부의 하드웨어에서 선형 보간이 수행된다. 테일러 급수 계산은 다음과 같이 작은 3차식을 평가하는 3차 보간(cubic interpolation)으로서 수행될 수 있다.

[0047]
$$f(\epsilon) = a_0 + a_1 \cdot \epsilon + a_2 \cdot \epsilon^2 + a_3 \cdot \epsilon^3$$

[0048] 여기서 계수 a 는 룩업 테이블로부터 입수된다. 그러나 이 수식의 복잡성은 상당하다(곱셈과 자승 연산을 수행하는 다수의 곱셈기로 인해).

[0049] 진술한 복잡성은 호너 알고리즘(Horner algorithm)(인수분해(factorization))를 이용해 감소될 수 있고, 따라서 $f(\epsilon)$ 는 다음과 같이 계산될 수 있다. 또한, 본원에서 참조로 이용되고 있는 "Digital Signal Processor With One Or More Non-Linear Functions Using Factorized Polynomial Interpolation"이라는 발명의 명칭으로 2008년 11월 28일에 미국 출원된 미국 특허 출원 번호 제12/324,934호를 참고하라.

수학식 3

[0050]
$$f(\epsilon) = ((b_3 \cdot \epsilon + b_2) \cdot \epsilon + b_1) \cdot \epsilon + b_0$$

[0051] 수학식(3)의 복잡성은 단지 3개의 곱셈과 3개의 덧셈 연산으로 감소되었다. $f(\epsilon)$ 는 룩업 테이블에 저장된 값으로부터의 오프셋(offset)이다.

[0052] 도 6은 수학식(3)을 구현하는 테일러 총합 계산 블록(600)을 예시한다. 계수들 b_0, b_1, b_2, b_3 는 룩업 테이블(650)로부터 검색된다. 테일러 총합 계산 블록(600)은 단지 3개의 곱셈(610) 연산과 3개의 덧셈(620) 연산만으로 수학식(3)을 구현한다.

[0053] 도 7은 본 발명의 일 실시예에 따라 하나 이상의 복소수에 대해 동시에 사용자 정의의 비선형 함수를 평가하는 실시예적인 벡터 기반 디지털 프로세서(700)의 개략적인 블록도이다. 일반적으로, 도 7의 벡터 기반 구현은 상이한 연산들을 동시에 수행한다. 그러므로, 벡터 기반 디지털 프로세서(700)는 사용자 정의의 비선형 함수를 평가하기 위한 복수의 기능성 유닛(710-1 내지 710-N)을 포함한다.

[0054] 일반적으로, 벡터 기반 디지털 프로세서(700)는 입력 벡터 x 를 처리하고, 출력 벡터 $y(n)$ 를 발생한다. 예시적인 벡터 기반 디지털 프로세서(700)는 다음과 같이 구현된 16-방식 벡터 프로세서 $n1$ 명령어에 대해 도시된다.

[0055] $vec_n1(x1, x2, \dots, x16)$, 여기서 $x[k]$ 의 범위는 0에서 1까지임

[0056] 이런 방식으로, 벡터 기반 디지털 프로세서(700)는 16가지의 이러한 비선형 연산을 수행할 수 있고, 단일 사이클내에 이들을 선형적으로 조합할 수 있다. 예를 들어, 사용자 정의의 비선형 함수는 다음과 같이 표현될 수 있다.

[0057]
$$f(x) = \sum_{k=0}^{15} a_k \cdot x^k$$

[0058] 보다 일반적인 경우, 상이한 함수들이 벡터 프로세서의 벡터 데이터의 각각의 성분들에 적용될 수도 있음을 유

의하라.

[0059] 도 7에 도시된 것처럼, 기능성 유닛(710)은 레지스터로의 저장을 위해 메모리로부터 예컨대 룩업 테이블 또는 계수같은 사용자 명세(user specification)를 수신한다.

[0060] 결론

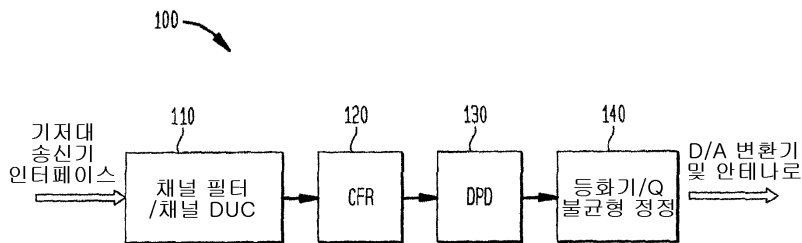
[0061] 본 발명의 예시적인 실시예들이 디지털 프로세서 내부의 디지털 논리 블록과 메모리 테이블과 관련하여 설명되었지만, 본 기술분야에 숙련된 사람에게는, 다양한 함수들이 디지털 영역에서 소프트웨어 프로그램의 처리 단계들로서 구현될 수도 있고, 회로 소자 또는 상태 머신에 의해 하드웨어로 구현될 수도 있으며, 또는 소프트웨어와 하드웨어의 조합으로 구현될 수도 있음이 자명하다. 이러한 소프트웨어는 예를 들어 디지털 신호 프로세서, 주문형 반도체 또는 마이크로 컨트롤러에 채용될 수 있을 것이다. 이러한 하드웨어와 소프트웨어는 집적회로 내부에 구현된 회로내에 구현될 수도 있다.

[0062] 그러므로, 본 발명의 함수들은 방법과 그 방법을 실시하는 장치의 형태로 구현될 수 있다. 본 발명의 하나 이상의 양상들은 예를 들어 저장 매체에 저장되거나, 머신에 로드되거나 또는 머신에 의해 실행되는 것에 무관하게 프로그램 코드의 형태로 구현될 수 있고, 이 프로그램 코드가 예컨대 프로세서같은 머신에 로드되어 실행될 때, 머신은 본 발명을 실시하는 장치가 된다. 범용 프로세서에 구현될 때, 프로그램 코드 세그먼트는 프로세서와 조합하여 특정 논리 회로와 유사하게 동작하는 장치를 제공한다. 본 발명은 또한 하나 이상의 집적 회로, 디지털 프로세서, 마이크로 프로세서 및 마이크로 컨트롤러에 구현될 수 있다.

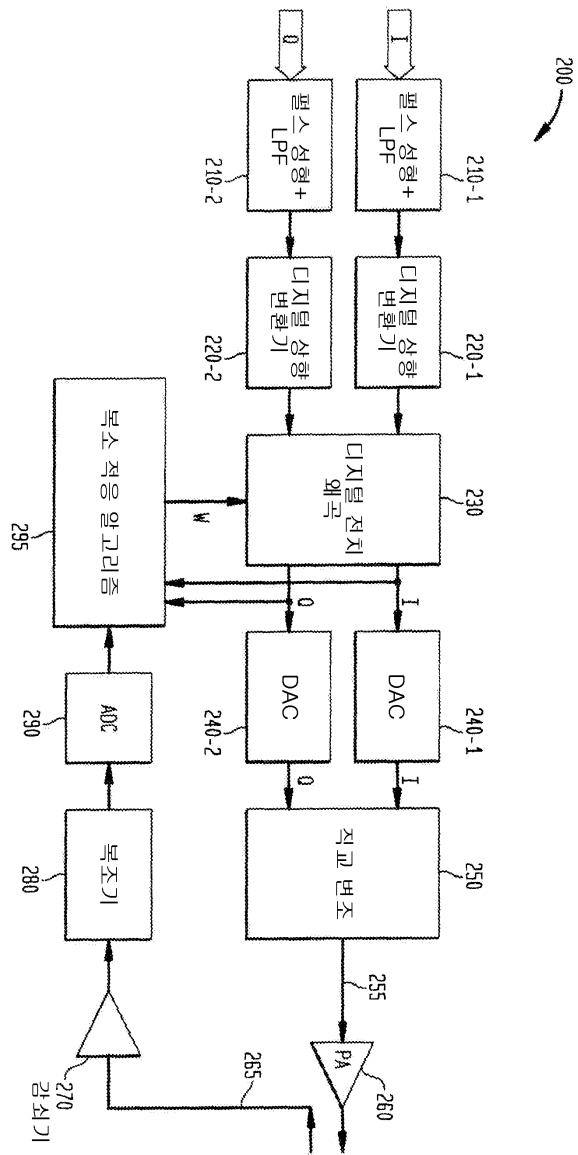
[0063] 본원에 도시 및 설명된 실시예 및 그 변형안들은 단지 본 발명의 원리를 예시하기 위한 것이고, 본 발명의 사상과 범주를 벗어나지 않으면서 당업자에 의해 다양한 수정안들이 구현될 수도 있음이 이해되어야 한다.

도면

도면1



도면2



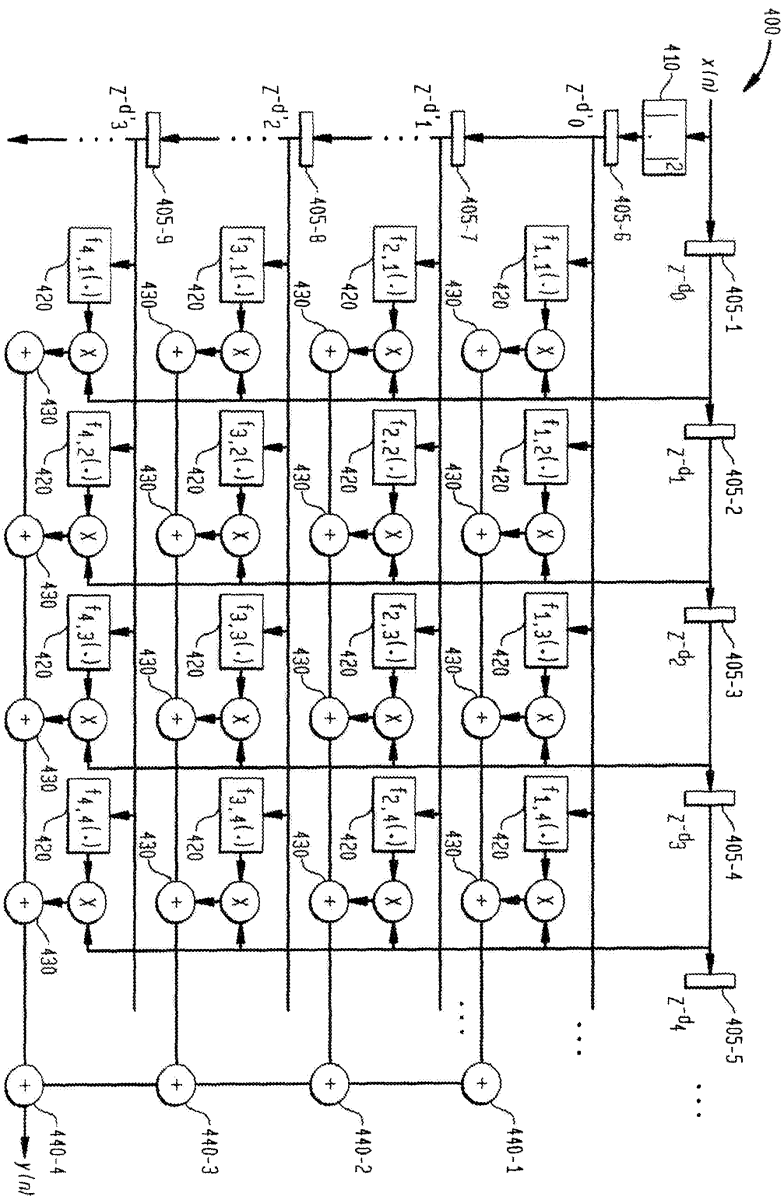
도면3

300 →

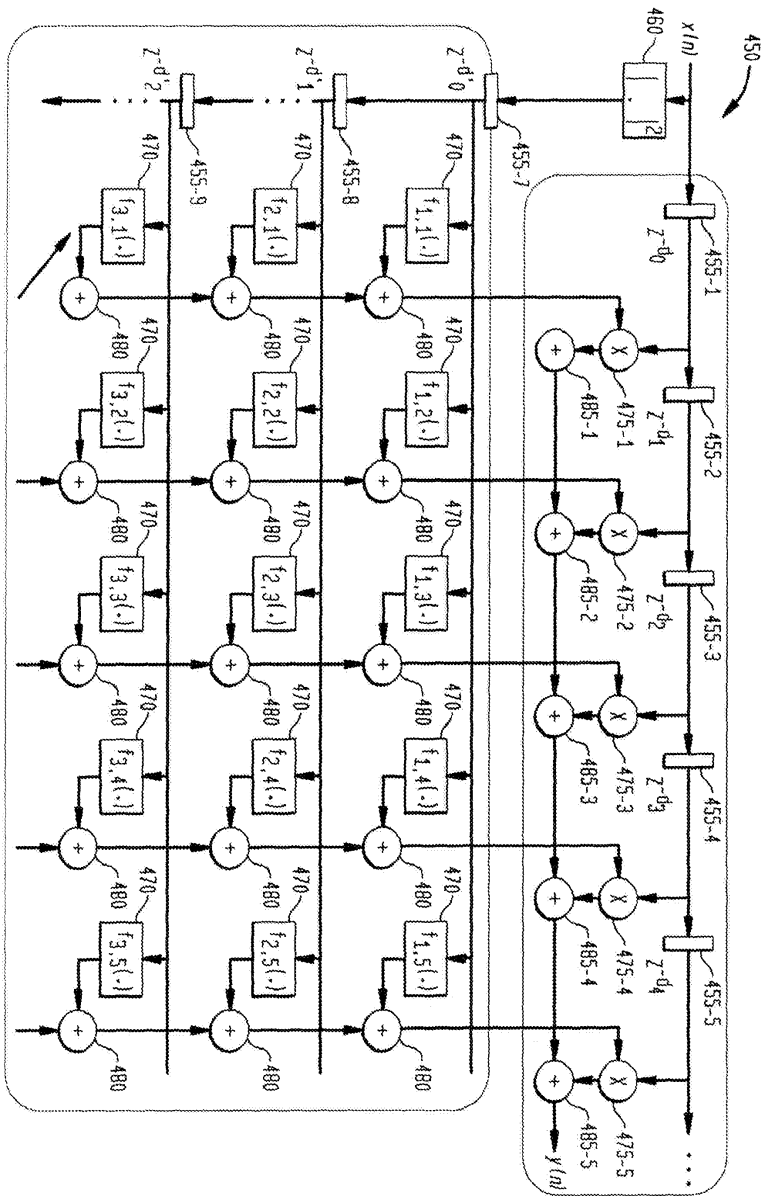
```

FOR i=0; i<256; i+=16
    {
        LOAD 16 INPUT SAMPLES (16b/16b COMPLEX)
        COMPUTE |x|^2 FOR 16 SAMPLES
        STORE 16 SQUARED SAMPLES x2 (16b/16b COMPLEX)
    }
    1 CYCLE FOR 16 SAMPLES, LOAD/MULT/STORE PIPELINED IN PARALLEL
}
FOR m=0; m<20; m++
    {
        LOAD 128 COEFFICIENTS FOR PIECEWISE POLYNOMIAL fm
        FOR i=0; i<256; i+=16
            {
                LOAD 16 INPUT SAMPLES AT OFFSET u(m)
                LOAD 16 SQUARED SAMPLES x2 AT OFFSET v(m)
                COMPUTE NONLINEAR FUNCTION fm(x2(v(m)))
                MULTIPLY/ACCUMULATE x(u(m))*fm
            }
    }
}
    
```

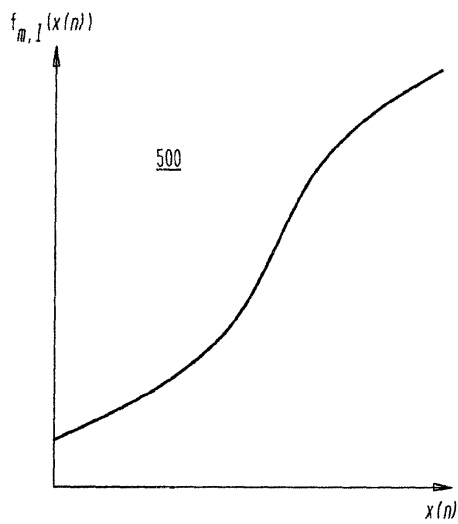
도면4a



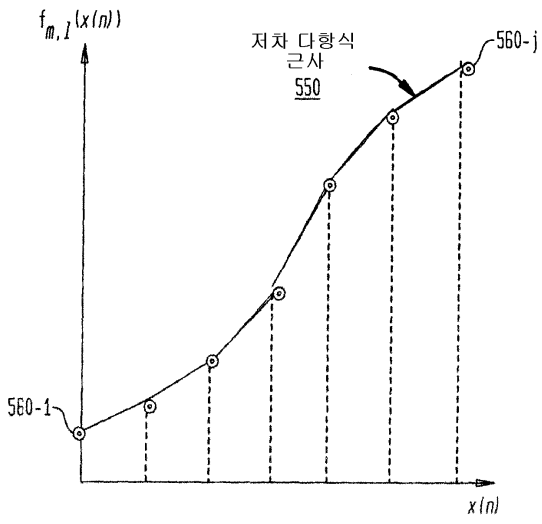
도면4b



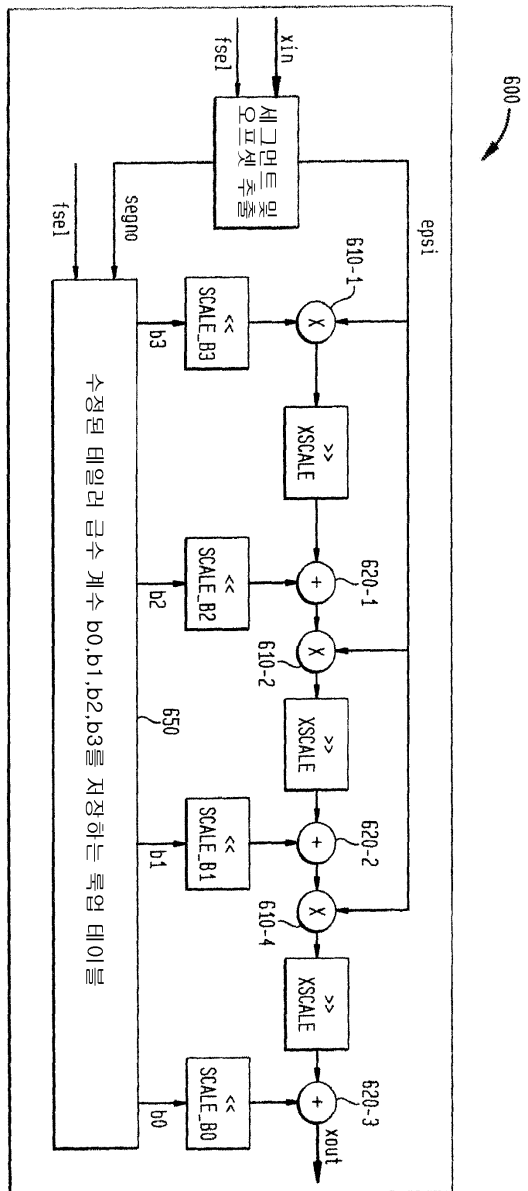
도면5a



도면5b



도면6



도면7

