



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0037236
(43) 공개일자 2019년04월05일

- (51) 국제특허분류(Int. Cl.)
G06F 17/17 (2006.01) **G06F 7/483** (2006.01)
G06F 7/544 (2017.01) **G06F 9/30** (2018.01)
- (52) CPC특허분류
G06F 17/17 (2013.01)
G06F 7/483 (2013.01)
- (21) 출원번호 10-2019-7002280
- (22) 출원일자(국제) 2017년06월02일
심사청구일자 없음
- (85) 번역문제출일자 2019년01월23일
- (86) 국제출원번호 PCT/US2017/035803
- (87) 국제공개번호 WO 2018/022191
국제공개일자 2018년02월01일
- (30) 우선권주장
15/224,237 2016년07월29일 미국(US)

- (71) 출원인
퀄컴 인코포레이티드
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 별명자
매튜, 디팍
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
잉글, 아제이 아난트
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
 (뒷면에 계속)
- (74) 대리인
특허법인 남엔남

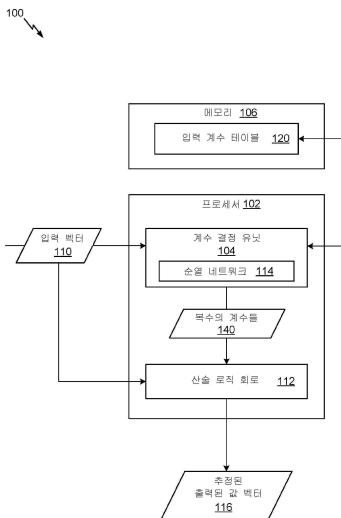
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 구분적 선형 근사화를 위한 시스템 및 방법

(57) 요 약

장치는 입력 값들의 벡터를 저장하도록 구성되는 하나 이상의 레지스터들을 포함한다. 장치는 또한 단일 명령의 프로세서에 의한 실행에 대한 응답으로, 복수의 구분적 분석 계수들을 선택하도록 구성되는 계수 결정 유닛을 포함한다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응한다. 장치는 적어도 단일 명령의 실행에 대한 응답으로, 복수의 구분적 분석 계수들 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하도록 구성되는 산술 로직 회로를 더 포함한다.

대 표 도 - 도1



(52) CPC특허분류

G06F 7/544 (2018.05)

G06F 9/3001 (2013.01)

G06F 9/30036 (2013.01)

G06F 9/3004 (2013.01)

(72) 발명자

션, 유통

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

주, 지안밍

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

호프만, 마크

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

장치로서,

입력 값들의 벡터를 저장하도록 구성되는 하나 이상의 레지스터들;

단일 명령의 프로세서에 의한 실행에 대한 응답으로, 복수의 구분적 분석 계수들을 선택하도록 구성되는 계수 결정 유닛 – 상기 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 상기 입력 값들의 벡터의 입력 값에 대응함 –; 및

적어도 상기 단일 명령의 실행에 대한 응답으로, 상기 복수의 구분적 분석 계수들 및 상기 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하도록 구성되는 산술 로직 회로를 포함하는, 장치.

청구항 2

제1 항에 있어서,

상기 단일 명령의 실행에 대한 응답으로, 상기 복수의 구분적 분석 계수들을 포함하는 구분적 분석 계수들의 벡터를 생성하도록 구성되는 순열 네트워크를 더 포함하는, 장치.

청구항 3

제1 항에 있어서,

상기 계수 결정 유닛은 상기 프로세서의 하나 이상의 레지스터들로부터 테이블을 리트리브하도록 추가로 구성되고, 상기 테이블은 구분적 분석 계수들의 다수의 세트들을 포함하는, 장치.

청구항 4

제3 항에 있어서,

상기 테이블은 구분적 분석 계수들의 다수의 세트들을 포함하는 제1 부분 및 하나 이상의 시프트 값을 포함하는 제2 부분을 포함하는, 장치.

청구항 5

제3 항에 있어서,

구분적 분석 계수들의 다수의 세트들의 각각의 세트는 제1 타입의 적어도 하나의 계수 및 제2 타입의 적어도 하나의 계수를 포함하는, 장치.

청구항 6

제5 항에 있어서,

상기 테이블은 상기 제1 타입의 다수의 구분적 분석 계수들을 포함하는 제1 섹션 및 상기 제2 타입의 다수의 구분적 분석 계수들을 포함하는 제2 섹션을 포함하는, 장치.

청구항 7

제3 항에 있어서,

상기 입력 값들의 벡터 및 상기 테이블에 기초하여 상기 복수의 구분적 분석 계수들을 선택하도록 구성되는 순열 네트워크를 더 포함하는, 장치.

청구항 8

제1 항에 있어서,

상기 산술 로직 회로는 곱을 생성하기 위해, 상기 입력 값들의 벡터의 특정 입력 값을 상기 구분적 분석 계수들의 하나 이상의 세트들 중 구분적 분석 계수들의 특정 세트의 제1 구분적 분석 계수와 곱하도록 추가로 구성되는, 장치.

청구항 9

제8 항에 있어서,

상기 산술 로직 회로는 상기 곱 및 상기 구분적 분석 계수들의 특정 세트의 제2 구분적 분석 계수에 기초하여 합을 생성하도록 추가로 구성되고, 상기 합은 상기 추정된 출력 값들의 제1 추정된 출력 값에 대응하는, 장치.

청구항 10

제1 항에 있어서,

상기 하나 이상의 레지스터들은,

상기 입력 값들의 벡터를 저장하도록 구성되는 제1 레지스터;

구분적 분석 계수들의 다수의 세트들을 저장하도록 구성되는 제2 레지스터; 및

인덱스 값 벡터를 저장하도록 구성되는 제3 레지스터를 포함하고, 상기 인덱스 값 벡터는 상기 입력 값들의 벡터 및 하나 이상의 시프트 값들에 기초하여 생성되는, 장치.

청구항 11

제10 항에 있어서,

상기 하나 이상의 레지스터들은,

상기 복수의 구분적 분석 계수들을 포함하는 구분적 분석 계수들의 벡터를 저장하도록 구성되는 제4 레지스터; 및

상기 추정된 출력 값을 저장하도록 구성되는 제5 레지스터를 포함하고,

상기 인덱스 값 벡터에 기초하여, 구분적 분석 계수들의 벡터를 생성하기 위해 상기 제2 레지스터로부터 상기 구분적 분석 계수들의 다수의 세트들 중 하나 이상을 상기 제4 레지스터에 라우팅하도록 구성되는 순열 네트워크를 더 포함하는, 장치.

청구항 12

값들을 추정하는 방법으로서,

입력 값들의 벡터를 수신하는 단계;

복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하는 단계 – 상기 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 상기 입력 값들의 벡터의 입력 값에 대응함 –; 및

상기 복수의 구분적 분석 계수들 및 상기 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하는 단계를 포함하는, 값들을 추정하는 방법.

청구항 13

제12 항에 있어서,

상기 복수의 구분적 분석 계수들을 포함하는 구분적 분석 계수들의 벡터를 생성하는 단계를 더 포함하고, 상기 구분적 분석 계수들의 벡터를 생성하는 단계는 상기 입력 값들의 벡터의 각각의 입력 값에 대해, 상기 입력 값에 기초하여 테이블로부터 구분적 분석 계수들의 세트를 선택하는 단계를 포함하고, 상기 테이블은 구분적 분석 계수들의 다수의 세트들을 포함하는, 값들을 추정하는 방법.

청구항 14

제12 항에 있어서,

상기 복수의 구분적 분석 계수들을 선택하기 위해 상기 단일 명령을 실행하는 단계는,

상기 입력 값들의 벡터의 제1 입력 값에 대응하는 복수의 비트들에 기초하여 구분적 분석 계수들의 테이블에 대한 루업을 수행하는 단계; 및

상기 루업에 기초하여, 상기 제1 입력 값에 대응하는 구분적 분석 계수들의 제1 세트를 결정하는 단계를 포함하는, 값을 추정하는 방법.

청구항 15

제14 항에 있어서,

상기 복수의 구분적 분석 계수들을 선택하기 위해 상기 단일 명령을 실행하는 단계는 상기 제1 입력 값 및 시프트 값에 기초하여 상기 복수의 비트들을 생성하는 단계를 더 포함하는, 값을 추정하는 방법.

청구항 16

제14 항에 있어서,

상기 함수에 기초하여 복수의 테이블들로부터 구분적 분석 계수들의 테이블을 선택하는 단계를 더 포함하는, 값을 추정하는 방법.

청구항 17

제14 항에 있어서,

상기 테이블은 구분적 분석 계수들의 다수의 그룹들을 포함하고, 상기 구분적 분석 계수들의 다수의 그룹들 중 적어도 2개의 그룹들의 구분적 분석 계수들은 동일한, 값을 추정하는 방법.

청구항 18

제12 항에 있어서,

상기 복수의 구분적 분석 계수들은 선형 분석 계수들에 대응하는, 값을 추정하는 방법.

청구항 19

제12 항에 있어서,

상기 복수의 구분적 분석 계수들은 구분적 분석 계수들의 다수의 세트들을 포함하고, 상기 구분적 분석 계수들의 다수의 세트들 중 구분적 분석 계수들의 각각의 세트는 2개의 구분적 분석 계수들을 포함하는, 값을 추정하는 방법.

청구항 20

제12 항에 있어서,

상기 함수는 비선형 함수를 포함하고, 상기 입력 값들의 벡터는 상기 함수의 입력 값들에 대응하는, 값을 추정하는 방법.

청구항 21

제12 항에 있어서,

상기 추정된 출력 값을 제2 단일 명령을 실행함으로써 결정되고, 상기 제2 단일 명령을 실행하는 단계는,

상기 복수의 구분적 분석 계수들의 제1 구분적 분석 계수와 상기 입력 값들의 벡터의 제1 입력 값의 곱을 생성하는 단계; 및

상기 곱과 상기 복수의 구분적 분석 계수들의 제2 구분적 분석 계수의 합에 기초하여 제1 추정된 출력 값을 생

성하는 단계를 포함하는, 값들을 추정하는 방법.

청구항 22

제12 항에 있어서,

상기 단일 명령을 실행하는 단계는 상기 함수의 추정된 출력 값들을 결정하는 단계를 더 포함하는, 값들을 추정하는 방법.

청구항 23

제12 항에 있어서,

상기 입력 값들의 벡터 및 타겟 범위에 기초하여 시프트 값을 계산하는 단계; 및

상기 입력 값들의 벡터의 각각의 입력 벡터에 대해, 상기 입력 값 및 상기 시프트 값에 기초하여 복수의 비트들을 생성하는 단계를 더 포함하고, 상기 복수의 비트들의 상위 비트들의 수는 타겟 범위 내에 있는, 값들을 추정하는 방법.

청구항 24

장치로서,

입력 값들의 벡터를 수신하기 위한 수단;

복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하기 위한 수단 – 상기 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 상기 입력 값들의 벡터의 입력 값에 대응함 –; 및

상기 복수의 구분적 분석 계수들 및 상기 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하기 위한 수단을 포함하는, 장치.

청구항 25

제24 항에 있어서,

상기 복수의 구분적 분석 계수들을 포함하는 구분적 분석 계수들의 벡터를 저장하기 위한 수단을 더 포함하는, 장치.

청구항 26

제24 항에 있어서,

상기 수신하기 위한 수단, 상기 실행하기 위한 수단 및 상기 결정하기 위한 수단은 모바일 폰, 셀룰러 폰, 컴퓨터, 휴대용 컴퓨터, 튜너, 라디오, 위성 라디오, 통신 디바이스, 모뎀, 휴대용 뮤직 플레이어, 휴대용 디지털 비디오 플레이어, 내비게이션 디바이스, PDA(personal digital assistant), 모바일 위치 데이터 유닛 또는 이들의 조합에 통합되는, 장치.

청구항 27

적어도 단일 명령을 포함하는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 단일 명령은 프로세서에 의해 실행되는 경우, 상기 프로세서로 하여금,

입력 값들의 벡터에 기초하여 복수의 구분적 분석 계수들을 선택하게 하고 – 상기 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 상기 입력 값들의 벡터의 입력 값에 대응함 –;

상기 선택된 복수의 구분적 분석 계수들에 기초하여 구분적 분석 계수들의 벡터를 생성하게 하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 28

제27 항에 있어서,

적어도 상기 단일 명령은 상기 프로세서에 의해 실행되는 경우, 추가로 상기 프로세서로 하여금, 상기 구분적 분석 계수들의 벡터 및 상기 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하게 하는, 비일시적 컴퓨터 관독가능 저장 매체.

청구항 29

제27 항에 있어서,

적어도 상기 단일 명령은, 상기 프로세서에 의해 실행되는 경우 상기 프로세서로 하여금, 상기 구분적 분석 계수들의 벡터 및 상기 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하게 하는 제2 명령을 포함하는, 비일시적 컴퓨터 관독가능 저장 매체.

청구항 30

제27 항에 있어서,

적어도 상기 단일 명령은 상기 프로세서에 의해 실행되는 경우, 추가로 상기 프로세서로 하여금,
상기 입력 값들의 벡터의 제1 입력 값 및 시프트 값에 기초하여 복수의 비트들을 생성하게 하고;
상기 복수의 비트들에 기초하여 계수 테이블에 대한 루업을 수행하게 하고;
상기 루업에 기초하여, 상기 제1 입력 값에 대응하는 구분적 분석 계수들의 제1 세트를 결정하게 하는, 비일시적 컴퓨터 관독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2016년 7월 29일에 출원된 미국 특허 출원 제15/224,237호에 대해 우선권을 주장하며, 상기 출원의 전체 내용은 인용에 의해 본원에 통합된다.

[0002] 본 개시는 일반적으로 입력 값들의 벡터를 사용한 구분적 선형 근사화에 관한 것이다.

배경 기술

[0003] 기술에서의 진보들은 더 작고 더 강력한 컴퓨팅 디바이스들을 도출해왔다. 예를 들어, 작고, 경량이고, 사용자들에 의해 소지하기 쉬운 모바일 및 스마트 폰들, 태블릿들 및 랩탑 컴퓨터들과 같은 무선 전화들을 포함하는 다양한 휴대용 개인 컴퓨팅 디바이스들이 현재 존재한다. 이러한 디바이스들은 무선 네트워크들을 통해 음성 및 데이터 패킷들을 통신할 수 있다. 추가로, 많은 이러한 디바이스들은 디지털 스틸 카메라, 디지털 비디오 카메라, 디지털 레코더 및 오디오 파일 플레이어와 같은 추가적인 기능을 통합한다. 또한, 이러한 디바이스들은 인터넷에 액세스하기 위해 사용될 수 있는 소프트웨어 애플리케이션들, 예를 들어, 웹 브라우저 애플리케이션을 포함하는 실행 가능한 명령들을 프로세싱할 수 있다. 따라서, 이러한 디바이스들은 상당한 컴퓨팅 능력을 포함할 수 있다.

[0004] 진보된 컴퓨팅 능력을 가능하게 하기 위해, 특정 기능들(예를 들어, 수학적 컴퓨터이션들, 예를 들어, 역함수, 제곱근 함수, 역 제곱근 함수, 지수 함수, 로그 함수, 삼각 함수 등)이 프로세서에 의해 근사화되어, 컴퓨터이셔널 시간을 감소시키고 컴퓨터이셔널 효율을 증가시킬 수 있다. 예를 들어, 특정 함수는 다수의 차수 다항식들에 기초하는 테일러 확장들을 생성함으로써 근사화될 수 있다. 그러나, 함수를 근사화하는 다중 차수 다항식들로부터 생성된 테일러 확장들은 비교적 많은 수의 명령들을 사용할 수 있다. 많은 수의 명령들을 실행 및 프로세싱하는 것은 상당한 컴퓨터이셔널 자원들, 예를 들어, 비교적 많은 수의 프로세싱 사이클들, 비교적 많은 양의 메모리 저장 공간, 상당한 수의 실행 유닛들 또는 이들의 조합을 사용할 수 있다.

발명의 내용

[0005] 특정 양상에서, 장치는 입력 값들의 벡터를 저장하도록 구성되는 하나 이상의 레지스터들을 포함한다. 장치는 또한 단일 명령의 프로세서에 의한 실행에 대한 응답으로, 복수의 구분적 분석 계수들을 선택하도록 구성되는 계수 결정 유닛을 포함한다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을

포함하고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응한다. 장치는 적어도 단일 명령의 실행에 대한 응답으로, 복수의 구분적 분석 계수들 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하도록 구성되는 산술 로직 회로를 더 포함한다.

[0006] 다른 특정 양상에서, 방법은 입력 값들의 벡터를 수신하는 단계를 포함한다. 방법은 또한 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하는 단계를 포함한다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응한다. 방법은 복수의 구분적 분석 계수들 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하는 단계를 더 포함한다.

[0007] 다른 특정 양상에서, 장치는 입력 값들의 벡터를 수신하기 위한 수단을 포함한다. 장치는 또한 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하기 위한 수단을 포함한다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응한다. 장치는 복수의 구분적 분석 계수들 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하기 위한 수단을 더 포함한다.

[0008] 다른 특정 양상에서, 비일시적 컴퓨터 판독가능 매체는, 프로세서에 의해 실행되는 경우, 프로세서로 하여금 입력 값들의 벡터에 기초하여 복수의 구분적 분석 계수들을 선택하게 하는 적어도 단일 명령을 포함한다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함하고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응한다. 적어도 단일 명령은 프로세서로 하여금, 선택된 복수의 구분적 분석 계수들에 기초하여 구분적 분석 계수들의 벡터를 생성하게 한다.

[0009] 본 개시의 다른 양상들, 이점들 및 특징들은, 하기 섹션들, 즉, 도면의 간단한 설명, 상세한 설명 및 청구항들을 포함하는 본 출원의 검토 후 명백해질 것이다.

도면의 간단한 설명

[0010] 도 1은 함수의 추정된 값을 생성하기 위해 구분적 분석을 사용하기 위한 시스템의 예의 블록도이다.

[0011] 도 2는 벡터화된 테이블 룩업 명령을 실행하는 예시적인 프로세스의 도면이다.

[0012] 도 3은 벡터 산술 명령을 실행하는 예시적인 프로세스의 도면이다.

[0013] 도 4는 벡터화된 테이블 룩업 명령을 실행하는 다른 예시적인 프로세스의 도면이다.

[0014] 도 5는 벡터 산술 명령을 실행하는 다른 예시적인 프로세스의 도면이다.

[0015] 도 6은 단일 구분적 근사화 명령을 실행하는 예시적인 프로세스의 도면이다.

[0016] 도 7은 단일 구분적 근사화 명령을 실행하는 다른 예시적인 프로세스의 도면이다.

[0017] 도 8은 함수의 추정된 값을 생성하는 방법의 예를 예시하는 흐름도이다.

[0018] 도 9는 함수의 추정된 값을 생성하도록 동작가능한 디바이스의 특정한 예시적 예의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0019] 도면들을 참조하여 본 개시의 특정 구현들이 아래에서 설명된다. 설명에서, 공통 특징들은 도면들 전반에 걸쳐 공통 참조 번호들로 지정된다. 본 명세서에서 사용되는 바와 같이, 다양한 용어는 오직 특정 구현들을 설명하기 위한 목적이며, 구현들의 제한으로 의도되지 않는다. 예를 들어, 본 명세서에서 사용되는 바와 같이, 단수형 형태들은, 문맥상 명확하게 달리 표시되지 않으면, 복수형 형태들을 또한 포함하도록 의도된다. "포함하다" 및 "포함하는"이라는 용어들은 "구비하다" 또는 "구비하는"과 상호교환가능하게 사용될 수 있음을 추가로 이해할 수 있다. 추가적으로, "여기서"라는 용어는 "여기에서"와 상호교환가능하게 사용될 수 있음을 이해할 것이다. 본 명세서에서 사용되는 바와 같이, 엘리먼트, 예를 들어, 구조, 컴포넌트, 동작 등을 수정하기 위해 사용되는 서수적 용어(예를 들어, "제1", "제2", "제3" 등)는 그 자체로 엘리먼트의 다른 엘리먼트에 대한 어떠한 우선순위 또는 순서를 표시하는 것이 아니라, 오히려 엘리먼트를 동일 이름을 갖는(그러나 서수적 용어를 사용한) 다른 엘리먼트로부터 단순히 구별한다. 본 명세서에서 사용되는 바와 같이, "세트"라는 용어는 특정 엘리먼트 중 하나 이상을 지칭하고, "복수"라는 용어는 다수의(예를 들어, 둘 이상의) 특정 엘리먼트를 지칭한다.

[0020] 본 개시는 감소된 수의 명령들(예를 들어, 하나 또는 두개의 명령들)을 실행함으로써 구분적 선형 근사

화를 사용하여 함수의 출력 값들을 추정하는 것을 설명한다. 예를 들어, 프로세서는 입력 값들의 벡터(함수의 입력 값들에 대응함)를 수신할 수 있고, 복수의 구분적 분석 계수들을 선택할 수 있다. 일부 구현들에서, 프로세서는, 복수의 구분적 분석 계수들을 포함하는 구분적 분석 계수들의 벡터를 생성할 수 있다. 특정 구현에서, 함수는 비선형일 수 있고, 구분적 분석 계수들은 비선형 함수의 1차 테일러 확장(예를 들어, 선형 근사화)에 기초할 수 있다. 복수의 구분적 분석 계수들은 다수의 구분적 분석 계수들을 포함하는 테이블로부터 선택될 수 있다. 입력 값들의 벡터 및 테이블에 기초하여 복수의 구분적 분석 계수들을 선택하기 위해 (또는 구분적 분석 계수들의 벡터를 생성하기 위해) 프로세서에 의해 순열 네트워크가 사용될 수 있다. 예시하자면, 순열 네트워크는 입력 값들의 벡터에 기초하여 복수의 구분적 분석 계수들(예를 들어, 구분적 분석 계수들의 벡터)을 포함하는, 제1 레지스터로부터의 테이블에 포함된 다수의 구분적 분석 계수들 중 하나 이상을 선택하여 제2 레지스터에 라우팅하기 위해 사용될 수 있다. 입력 값들의 벡터의 각각의 입력 값은 구분적 분석 계수들의 세트에 대응할 수 있다. 추가로, 복수의 구분적 분석 계수들을 선택하는 것은 단일 명령의 실행을 통해 수행될 수 있다.

[0013]

[0021] 추가적으로, 프로세서는 복수의 구분적 분석 계수들 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정(예를 들어, 생성)할 수 있다. 예를 들어, 산술 로직 회로는 제1 구분적 분석 계수를 제1 입력 값과 곱함으로써 곱을 생성할 수 있다. 산술 로직 회로는 곱 및 제2 구분적 분석 계수를 가산함으로써 제1 추정된 출력 값을 생성할 수 있다. 일부 구현들에서, 함수의 추정된 출력 값을 결정하는 것은 다른 단일 명령을 실행함으로써 수행될 수 있다. 다른 구현들에서, 복수의 구분적 분석 계수들을 선택하는 것 및 추정된 출력 값을 결정하는 것 둘 모두는 동일한 단일 명령에 대한 응답으로 수행될 수 있다.

[0014]

[0022] 다른 근사화 기술들에 비해, 비선형 함수의 값을 추정하기 위해 선형 근사화를 사용함으로써, 함수의 추정된 출력 값을 생성하기 위해 더 적은 명령들이 사용될 수 있다. 다른 근사화 기술들에 비해 추정된 출력 값을 생성하기 위해 사용되는 명령들의 수를 감소시킴으로써, 추정된 출력 값을 생성하기 위해 사용되는 컴퓨터이셔널 자원들의 양에서의 감소가 달성될 수 있다. 추가적으로, 엘리먼트마다 또는 레인마다의 루프 테이블에 비해 구분적 분석 계수들을 라우팅하기 위해 순열 네트워크를 사용함으로써, 테이블을 저장하는 것, 추정된 출력 값을 생성하는 것 또는 둘 모두를 위해 사용되는 메모리의 양에서의 감소가 달성될 수 있다.

[0015]

[0023] 도 1을 참조하면, 하나 이상의 입력 값들에 대응하는 함수의 추정된 출력 값을 생성하기 위해 구분적 분석을 사용하기 위한 시스템(100)이 도시된다. 일부 구현들에 따르면, 구분적 분석은 1차 테일러 확장을 사용하여 함수의 선형 근사화(예를 들어, $y = a*x+b$)에 기초할 수 있다. 일부 구현들에서, 함수는 비선형 함수를 포함하거나 그에 대응할 수 있다. 예를 들어, 함수는 역함수, 제곱근 함수, 역 제곱근 함수, 지수 함수, 로그 함수, 삼각 함수, 역 삼각 함수(예를 들어, 아크 탄젠트 함수) 또는 이들의 조합을 포함하거나 그에 대응할 수 있다.

[0016]

[0024] 시스템(100)은 프로세서(102) 및 메모리(106)를 포함한다. 메모리(106)는 프로세서(102)에 커플링될 수 있고, 하나 이상의 계수 테이블들(예를 들어, 루프 테이블들)과 같이 데이터를 저장하도록 구성될 수 있다. 예를 들어, 메모리(106)는 입력 계수 테이블(120)을 저장할 수 있다. 입력 계수 테이블(120)은 특정 함수(예를 들어, 역 제곱근 함수)의 구분적 분석 근사화와 연관된 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 예를 들어, 입력 계수 테이블(120)은 구분적 분석 계수들 a_0 및 b_0 의 제1 세트 및 구분적 분석 계수들 a_1 및 b_1 의 제2 세트를 포함할 수 있다. 다른 예들에서, 입력 계수 테이블(120)은 둘보다 적은 또는 둘보다 많은 세트들의 구분적 분석 계수들을 포함할 수 있다. 일부 구현들에서, 메모리(106)는 다수의 입력 계수 테이블들을 저장할 수 있고, 각각의 입력 계수 테이블은 상이한 함수, 상이한 추정 정확도(예를 들어, 구분적 분석에서 사용되는 다수의 세그먼트들) 또는 이들의 조합에 대응할 수 있다. 메모리(106)는 도 1의 프로세서(102)와 별개인 것으로 예시되지만, 다른 구현들에서, 메모리(106)는 프로세서(102)에 포함될 수 있다.

[0017]

[0025] 프로세서(102)는 테이블 로드 명령을 수신하는 것에 대한 응답으로 프로세서(102)의 하나 이상의 레지스터들(예를 들어, 레지스터 파일)에 하나 이상의 계수 테이블들을 로딩하도록 구성될 수 있다. 예시하자면, 프로세서(102)는 메모리(106)로부터 입력 계수 테이블(120)을 리트리브(또는 수신)하고 프로세서(102)의 하나 이상의 레지스터들에 입력 계수 테이블(120)을 저장하도록 구성될 수 있다.

[0018]

[0026] 프로세서(102)는 하나 이상의 벡터 명령들(도 1에 도시되지 않음)을 수신하도록 구성될 수 있다. 하나 이상의 벡터 명령들은 도 2 내지 도 7을 참조하여 설명된 벡터 명령들을 포함하거나 그에 대응할 수 있다. 일부 구현들에서, 프로세서(102)는 하나 이상의 벡터 명령들을 수신하기 전에 하나 이상의 계수 테이블들을 하나 이상의 레지스터들에 로딩할 수 있다. 하나 이상의 벡터 명령들의 실행 동안, 프로세서(102)는 입력 벡터(110) 및 입력 계수 테이블(120)을 수신할 수 있고, 프로세서(102)는 추정된 출력 값 벡터(116)를 생성할 수 있다.

입력 벡터(110)(예를 들어, 입력 값들의 벡터)는 특정 함수에 대한 하나 이상의 입력 값을 포함하거나 그에 대응할 수 있다. 추정된 출력 값 벡터(116)는 입력 벡터(110)에 대응하는 입력 값을 갖는 특정 함수의 하나 이상의 추정된(예를 들어, 근사화된) 출력 값을 포함하거나 그에 대응할 수 있다.

[0019]

[0027] 프로세서(102)는 계수 결정 유닛(104) 및 산술 로직 회로(112)를 포함할 수 있다. 계수 결정 유닛(104)은 입력 벡터(110) 및 입력 계수 테이블(120)을 수신(또는 리트리브)하고 복수의 계수들(140)을 선택하도록 구성될 수 있다. 일부 구현들에서, 계수 결정 유닛(104)은 프로세서(102)의 하나 이상의 레지스터들로부터 입력 계수 테이블(120)을 리트리브(또는 수신)할 수 있다. 다른 구현들에서, 계수 결정 유닛(104)은 메모리(106)로부터 입력 계수 테이블(120)을 리트리브(또는 수신)할 수 있다. 복수의 계수들(140)은 도 2 내지 도 5를 참조하여 설명된 바와 같이 계수 벡터 또는 도 6 또는 도 7을 참조하여 설명된 바와 같이 구분적 분석 계수들의 하나 이상의 선택된 세트들을 포함하거나 그에 대응할 수 있다. 복수의 계수들(140)은 구분적 분석 계수들의 하나 이상의 세트들을 포함할 수 있다. 예를 들어, 입력 벡터(110)의 특정 입력 값에 대해, 프로세서(102)의 계수 결정 유닛(104)은 대응하는 복수의 비트들(예를 들어, 비트 값을 이진 형태로 표시하는 다수의 비트들)을 결정할 수 있다. 복수의 비트들은 특정 입력 값의 이진 표현과 연관된 다수의 상위 비트들에 기초하여 결정될 수 있다. 복수의 비트들은 입력 계수 테이블(120)에 대한 루업을 수행하기 위한 인덱스 값으로서 사용될 수 있다. 다른 구현들에서, 단일 비트 값이 계수 결정 유닛(104)에 의해 결정되고 루업을 수행하기 위한 인덱스 값으로서 사용될 수 있다. 루업에 기초하여, 프로세서(102)의 계수 결정 유닛(104)은 특정 입력 값에 대응하는 구분적 분석 계수들의 특정 세트를 결정할 수 있다. 예시하자면, 입력 벡터(110)의 각각의 입력 값에 대해, 계수 결정 유닛(104)은 입력 계수 테이블(120)로부터 구분적 분석 계수들의 쌍과 같은 특정 세트를 선택할 수 있다. 복수의 계수들(140)은 구분적 분석 계수들의 선택된 세트들을 포함할 수 있다. 따라서, 복수의 계수들(140)은 입력 벡터(110)의(예를 들어, 그에 포함된) 각각의 입력 값에 대한 구분적 분석 계수들의 대응하는 세트를 포함할 수 있다. 예를 들어, 구분적 분석 계수들의 각각의 선택된 세트는 입력 벡터(110)의 입력 값에 대응할 수 있다.

[0020]

[0028] 일부 구현들에서, 계수 결정 유닛(104)은 하나 이상의 시프트 값을 적용함으로써 입력 벡터(110)의 입력 값들에 대응하는 비트 값을 결정할 수 있다. 예를 들어, 계수 결정 유닛(104)은 하나 이상의 시프트 값을 사용하여 하나 이상의 이진 시프트 연산들을 수행할 수 있다. 예시하자면, 계수 결정 유닛(104)은 입력 값을 우측 시프트(예를 들어, 나눗셈)할 수 있다. 입력 값(예를 들어, 이의 이진 표현)은 비트 값을 생성하기 위해 시프트 값을(예를 들어, 다수의 제로들)만큼 우측 시프트될 수 있다. 비트 값의 특정 수의 "최상위" 숫자들은 인덱스 값을 생성하기 위해 사용될 수 있다. 특정 구현에서, 입력 값은 계수 결정 유닛(104)에 의한 수신 전에 정규화되고, 입력 값은 인덱스 값에 대응하는 비트 값의 다수(예를 들어, 2, 3, 4, 5 등)의 후속 숫자들을 결정하기 위해 시프트 값을 만큼 우측 시프트된다.

[0021]

[0029] 일부 구현들에서, 입력 계수 테이블(120)은 하나 이상의 시프트 값을 포함한다. 특정 구현에서, 하나 이상의 시프트 값을 동일한 값을 가질 수 있다. 예를 들어, 동일한 시프트 값이 입력 벡터(110)의 각각의 입력 값에 적용될 수 있다. 다른 구현들에서, 프로세서(102)는 시프트 값을 벡터를 수신 또는 생성할 수 있다. 예를 들어, 프로세서(102)는 도 9를 참조하여 설명된 바와 같이 시프트 값을 생성기를 포함할 수 있다.

[0022]

[0030] 일부 구현들에서, 계수 결정 유닛(104)은 순열 네트워크(114)를 포함할 수 있다. 순열 네트워크(114)는 복수의 계수들(140)을 선택하기 위해 구분적 분석 계수들을 선택 및 라우팅하도록 구성될 수 있다. 예를 들어, 순열 네트워크(114)는 입력 벡터(110)의 각각의 입력 값에 대해, 입력 계수 테이블(120)로부터 구분적 분석 계수들의 특정 세트(예를 들어, 쌍)를 선택할 수 있다. 순열 네트워크(114)는 입력 계수 테이블(120)을 저장하는 레지스터로부터 구분적 분석 계수들의 선택된 세트들을 다른 레지스터로 라우팅하여, 복수의 계수들(140)을 생성할 수 있다. 따라서, 복수의 계수들(140)은 입력 벡터(110)에 포함된 각각의 입력 값에 대한 구분적 분석 계수들의 대응하는 세트를 포함할 수 있다. 순열 네트워크(114)는 도 1의 계수 결정 유닛(104)에 포함된 것으로 예시되지만, 다른 구현들에서, 순열 네트워크(114)는 계수 결정 유닛(104)과 별개일 수 있다.

[0023]

[0031] 일부 구현들에서, 계수 결정 유닛(104)은 복수의 계수들(140)에 기초하여 구분적 분석 계수들의 계수 벡터(예를 들어, 도 2 내지 도 5를 참조하여 설명된 바와 같은 계수 벡터)를 생성할 수 있다. 예를 들어, 순열 네트워크(114)는 계수 벡터를 생성하기 위해 구분적 분석 계수들을 선택 및 라우팅하도록 구성될 수 있다. 예시하자면, 순열 네트워크(114)는 입력 벡터(110)의 각각의 입력 값에 대해, 입력 계수 테이블(120)로부터 구분적 분석 계수들의 대응하는 세트(예를 들어, 쌍)를 선택할 수 있다. 순열 네트워크(114)는 입력 계수 테이블(120)을 저장하는 레지스터로부터 구분적 분석 계수들의 선택된 세트들을 다른 레지스터로 라우팅하여, 계수 벡

터를 생성할 수 있다. 따라서, 계수 벡터는 복수의 계수들(140)을 포함할 수 있다.

[0024] [0032] 프로세서(102)는 또한 산술 연산들을 수행하도록 구성된 산술 로직 회로(112)를 포함할 수 있다. 산술 로직 회로(112)는 하나 이상의 가산기들, 감산기들, 곱셈기들, 나눗셈기들 또는 이들의 조합을 포함할 수 있다. 프로세서(102)는 산술 로직 회로(112)를 사용하여 입력 벡터(110) 및 복수의 계수들(140)에 대한 하나 이상의 산술 연산들을 수행하도록 구성될 수 있다. 예를 들어, 산술 로직 회로(112)는 입력 벡터(110) 및 복수의 계수들(140)을 수신할 수 있고, 산술 로직 회로(112)는 입력 벡터(110) 및 복수의 계수들(140)에 대해 하나 이상의 산술 연산들을 수행할 수 있다. 예시하자면, 산술 로직 회로(112)는 구분적 분석 계수들의 세트의 제1 구분적 분석 계수와 대응하는 입력 값의 곱을 생성하기 위해 곱셈 연산을 수행할 수 있다. 추가로, 산술 로직 회로(112)는 곱셈 연산의 곱과 구분적 분석 계수들의 세트의 제2 구분적 분석 계수의 합을 생성하기 위해 가산 연산을 수행할 수 있다. 따라서, 입력 벡터(110)에 포함된 각각의 입력 값에 대해, 시스템(100)은 특정 함수와 연관된 대응하는 추정된 출력 값을 생성하도록 구성될 수 있다. 추정된 출력 값들은 추정된 출력 값 벡터(116)로서 저장될 수 있다.

[0025] [0033] 일부 구현들에서, 프로세서(102)(예를 들어, 계수 결정 유닛(104))는 복수의 계수들(140)을 선택하기 위해 제1 단일 명령(예를 들어, 도 2를 참조하여 설명된 바와 같은 벡터화된 테이블 루업 명령(201))을 실행하도록 구성될 수 있다. 추가로, 프로세서(102)(예를 들어, 산술 로직 회로(112))는 추정된 출력 값 벡터(116)를 생성하기 위해 제2 단일 명령(예를 들어, 도 3을 참조하여 앞서 설명된 바와 같은 벡터 산술 명령(301))을 실행하도록 구성될 수 있다. 다른 구현들에서, 프로세서(102)는 복수의 계수들(140) 및 추정된 출력 값 벡터(116)를 선택하기 위해 단일 명령(예를 들어, 도 6을 참조하여 설명된 바와 같은 단일 구분적 근사화 명령(601))을 실행하도록 구성될 수 있다.

[0026] [0034] 일부 구현들에서, 프로세서(102)는 다양한 벡터들을 저장하도록 구성된 하나 이상의 레지스터들(미도시)을 포함할 수 있다. 예를 들어, 프로세서(102)는 입력 벡터(110), 입력 계수 테이블(120), 복수의 계수들(140) 및 추정된 출력 값 벡터(116) 또는 이들의 조합을 저장하도록 구성된 레지스터들을 포함할 수 있다. 레지스터들은 도 3, 도 4 및 도 6을 참조하여 더 상세히 설명된다.

[0027] [0035] 일부 구현들에서, 입력 벡터(110)의 입력 값들 중 하나 이상은 타겟 범위로 정규화될 수 있다. 타겟 범위는 공칭 값들의 범위를 포함하거나 그에 대응할 수 있다. 타겟 범위는 입력 계수 테이블(120)의 구분적 분석 계수들의 다수의 세트들의 값들의 비트 크기를 감소시키도록 선택될 수 있다. 예시적이고 비제한적인 예들로서, 타겟 범위는 대략 0.5 내지 1 또는 대략 0 내지 1의 범위를 포함할 수 있다. 시프트 값을 입력 값에 적용하는 것은 타겟 범위 내의 값(예를 들어, 정규화된 값)을 생성할 수 있다.

[0028] [0036] 일부 구현들에서, 프로세서(102)는 계수 결정 유닛(104)에서 입력 값들의 수신 전에 정규화된 입력 값을 생성하기 위해 특정 입력 값을 시프트 값만큼 좌측 시프트(예를 들어, 곱셈)할 수 있다. 프로세서(102)(예를 들어, 계수 결정 유닛(104))는 인덱스 값을 생성하기 위해 특정 입력 값을 시프트 값만큼 우측 시프트(예를 들어, 나눗셈)할 수 있다. 일부 구현들에서, 계수 결정 유닛(104)은 정규화되지 않은 입력 값들을 수신할 수 있고, 계수 결정 유닛(104)은 인덱스 값을 생성하기 위해 입력 값을 우측 시프트하기 전에 입력 값을 정규화(예를 들어, 좌측 시프트)할 수 있다. 특정 구현에서, 입력 벡터(110)의 입력 값들 중 하나 이상은 도 2를 참조하여 설명된 바와 같이 인덱스 값 벡터를 생성하기 전에 타겟 범위로 정규화될 수 있다. 정규화된 값을 생성하기 위해 사용되는 시프트 값(예를 들어, 다수의 제로들)은 인덱스 값을 생성하기 위해 사용되는 시프트 값과 동일할 수 있거나 그와 상이할 수 있다. 다른 구현들에서, 계수 결정 유닛(104)은 비트 값을 결정하기 위해 입력 값을 하나 이상의 시프트 값들만큼 좌측 시프트(예를 들어, 곱셈)할 수 있다.

[0029] [0037] 연산 동안, 프로세서(102)는 입력 벡터(110)를 수신할 수 있다. 입력 벡터(110)는 도 2를 참조하여 설명된 바와 같이 프로세서(102)의 제1 레지스터에 저장될 수 있다. 계수 결정 유닛(104) 및 산술 로직 회로(112)는 입력 벡터(110)를 수신(또는 리트리브)하기 위해 제1 레지스터에 액세스할 수 있다. 일부 구현들에서, 입력 벡터(110)는 벡터 명령의 피연산자에 의해 표시될 수 있다.

[0030] [0038] 프로세서(102)는 복수의 계수들(140)을 생성하기 위해 단일 명령을 실행할 수 있다. 프로세서(102)의 계수 결정 유닛(104)은 복수의 계수들(140)을 선택하기 위해 입력 계수 테이블(120)을 리트리브, 수신 또는 액세스할 수 있다. 입력 계수 테이블(120)은 프로세서(102)의 제2 레지스터에 저장될 수 있다. 일부 구현들에서, 입력 벡터(110)의 각각의 입력 값에 대해, 복수의 계수들(140)은 도 2를 참조하여 추가로 설명된 바와 같이 구분적 분석 계수들의 대응하는 세트를 포함할 수 있다.

[0031]

[0039] 프로세서(102)는 입력 벡터(110) 및 복수의 계수들(140)에 기초하여 추정된 출력 값 벡터(116)를 생성할 수 있다. 예시하자면, 산술 로직 회로(112)는 추정된 출력 값 벡터(116)의 추정된 출력 값을 생성하기 위해 입력 벡터(110)의 각각의 입력 값 및 복수의 계수들(140)의 구분적 분석 계수들의 대응하는 세트에 대해 벡터 산술 연산들을 수행할 수 있다. 예를 들어, 산술 로직 회로(112)는 곱을 생성하기 위해 입력 벡터(110)의 입력 값을, 구분적 분석 계수들의 대응하는 세트의 제1 구분적 분석 계수와 곱할 수 있다. 산술 로직 회로(112)는 추정된 출력 값 벡터(116)의 추정된 출력 값을 생성하기 위해, 곱과 구분적 분석 계수들의 대응하는 세트의 제2 구분적 분석 계수를 가산할 수 있다.

[0032]

[0040] 단일 벡터 명령의 실행 동안 복수의 계수들(140)을 생성함으로써, 시스템(100)은 다수의 벡터 명령들의 실행 동안 복수의 계수들(140)을 선택하는 다른 시스템들에 비해 더 적은 메모리 및 전력을 사용할 수 있고, 증가된 속도를 가질 수 있다. 추가적으로, 하나의(또는 두개의) 벡터의 실행 동안 추정된 출력 값 벡터(116)를 생성함으로써, 명령(들)은 둘 이상의 벡터 명령들의 실행 동안 추정된 출력 값 벡터(116)를 생성하는 다른 시스템들에 비해, 속도를 추가로 증가시킬 수 있고 전력 소비를 추가로 감소시킬 수 있다. 추가로, (예를 들어, 복수의 계수들(140)을 선택하도록 구분적 분석 계수들을 라우팅하기 위해 산술 네트워크(114)를 사용하여) 벡터화된 테이블 루업을 수행하는 것은 엘리먼트마다 또는 레인마다 루업 테이블(예를 들어, 각각의 입력 값에 대해 별개의 루업 테이블)을 사용하는 다른 시스템들에 비해 덜 복잡하고, 더 적은 메모리 및 프로세싱 자원들을 사용할 수 있다.

[0033]

[0041] 앞서 설명된 설명의 양상들에서, 수행되는 다양한 기능들은 도 1의 시스템(100)의 특정 회로 또는 컴포넌트들에 의해 수행되는 것으로 설명되었다. 그러나, 회로 및 컴포넌트들의 이러한 분할은 오직 예시를 위한 것이다. 대안적인 예들에서, 특정 회로 또는 컴포넌트에 의해 수행되는 기능은 그 대신 다수의 회로들 또는 컴포넌트들 사이에 분할될 수 있다. 또한, 다른 대안적인 예들에서, 시스템(100)의 둘 이상의 회로들 또는 컴포넌트들은 단일 회로 또는 컴포넌트로 통합될 수 있다. 도 1에 예시된 각각의 회로 및 컴포넌트는 하드웨어(예를 들어, ASIC(application-specific integrated circuit), 프로세싱 유닛, 예를 들어, CPU(central processing unit), DSP(digital signal processor), 제어기, 다른 하드웨어 디바이스, 펌웨어 디바이스 또는 이들의 임의의 조합)를 사용하여 구현될 수 있다.

[0034]

[0042] 도 2를 참조하면, 벡터화된 테이블 루업 명령(201)을 실행하는 예시적인 프로세스의 도면(200)이 개시되어 있다. 벡터화된 테이블 루업 명령(201)은 도 1의 프로세서(102)와 같은 프로세서에 의해 실행될 수 있다. 도 2에 예시된 바와 같이, 벡터화된 테이블 루업 명령(201)은 명령 명칭(203)(vlut4segh)을 포함할 수 있다(예를 들어, opcode). 명령 명칭(203)은 세그먼트 표시자(204) 및 단어 길이 표시자(205)를 포함할 수 있다. 세그먼트 표시자(204)의 값은 구분적 분석의 다수의 세그먼트들을 표시할 수 있다. 세그먼트 표시자(204)의 값은 또한 구분적 분석 계수들의 다수의 세트들을 표시하거나 그에 대응할 수 있다. 도 2에 예시된 예에서, 세그먼트 표시자(204)는 구분적 분석의 4개의 세그먼트들이 존재함(예를 들어, 구분적 분석 계수들의 4개의 세트들이 존재함)을 표시한다. 단어 길이 표시자(205)의 값은 입력 값의 비트 크기를 표시할 수 있다. 도 2에 예시된 예에서, 단어 길이 표시자(205)의 값은 "h"이고, 명령 단어(예를 들어, 16 비트들)의 "절반"인 입력 값의 크기를 표시하기 위해 사용될 수 있다. 다른 구현들에서, 단어 길이 표시자(205)는 상이한 비트 크기를 표시하는 상이한 값을 가질 수 있다.

[0035]

[0043] 벡터화된 테이블 루업 명령(201)은 또한 제1 필드(202)(Vdd), 제2 필드(206)(Vuu) 및 제3 필드(207)(Vv)와 같은 하나 이상의 필드들(예를 들어, 피연산자들)을 포함할 수 있다. 제1 필드(202)에 저장된 제1 값은 도 3을 참조하여 설명된 바와 같이 벡터 산술 명령의 실행 동안 사용을 위한 계수 벡터(240)(예를 들어, 벡터 Vdd)를 표시할 수 있다. 제2 필드(206)에 저장된 제2 값은 입력 계수 테이블(220)(예를 들어, 벡터 Vuu)을 표시할 수 있고, 제3 필드(207)에 저장된 제3 값은 입력 벡터(210)(예를 들어, 벡터 Vv)를 표시할 수 있다. 3개의 필드들이 예시되지만, 다른 구현들에서 벡터화된 테이블 루업 명령(201)은 3개 초과의 필드들 또는 3개 미만의 필드들을 포함할 수 있다.

[0036]

[0044] 프로세서는 벡터화된 테이블 루업 명령(201)의 실행 동안 다양한 동작들을 수행하도록 구성될 수 있다. 프로세서는 x0, x1, x2 및 x3과 같은 복수의 입력 값들을 포함하는 입력 벡터(210)(Vv)를 수신하도록 구성될 수 있다. 입력 벡터(210)(예를 들어, 입력 값들의 벡터)는 프로세서의 제1 레지스터(212)에 저장될 수 있다. 프로세서는 도 1의 메모리(106)와 같은 메모리로부터 입력 계수 테이블(220)을 수신(또는 리트리브)하도록 구성될 수 있다. 입력 계수 테이블(220)은 a0 및 b0, a1 및 b1, a2 및 b2, 및 a3 및 b3과 같은 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 구분적 분석 계수들의 다수의 세트들 중 구분적 분석 계수들의 각각의 세트는 0, 1, 2 및 3과 같은 인덱스 값에 대응할 수 있다. 예를 들어, a0 및 b0은 0의 인덱스 값에 대응할 수

있고, a1 및 b1은 1의 인덱스 값에 대응할 수 있다. 입력 계수 테이블(220)은 또한 shiftvalue0 - shiftvalue3과 같은 하나 이상의 시프트 값을 포함할 수 있다. 특정 구현에서, 프로세서는 프로세서의 제2 레지스터(222)에 입력 계수 테이블(220)을 저장할 수 있다. 다른 구현들에서, 프로세서는 벡터화된 테이블 루업 명령(201)의 실행 전에 입력 계수 테이블(220)을 제2 레지스터(222)에 로딩할 수 있다. 프로세서는 벡터화된 테이블 루업 명령(201)의 실행 전에 입력 계수 테이블(220)을 제2 레지스터(222)로부터 리트리브(또는 수신)할 수 있다.

[0037] 다른 특정 구현에서, 입력 계수 테이블(220)은 프로세서의 다수의 레지스터들에 저장될 수 있다. 예를 들어, 입력 계수 테이블(220)은 하나 이상의 시프트 값을 포함하는 제1 부분 및 구분적 분석 계수들의 다수의 세트들을 포함하는 제2 부분을 포함할 수 있다. 입력 계수 테이블(220)의 제1 부분은 제2 레지스터(222)에 저장될 수 있고, 입력 계수 테이블(220)의 제2 부분은 다른 레지스터에 저장될 수 있다.

[0038] 프로세서는 인덱스 값 벡터(230)를 생성하기 위해 입력 계수 테이블(220)을 사용하여 입력 벡터(210)를 프로세싱할 수 있다. 인덱스 값 벡터(230)는 프로세서의 제3 레지스터(232)에 저장될 수 있고, 2, 3, 0 및 1과 같은 다수의 인덱스 값을 포함할 수 있다. 다수의 인덱스 값을 각각의 인덱스 값은 입력 벡터(210)의 입력 값에 대응할 수 있다. 도 2는 인덱스 값 벡터(230)에서 어떠한 인덱스 값도 반복(예를 들어, 한번 초과로 등장)되지 않는 예를 예시한다. 다른 예들에서, 특정 인덱스 값은 인덱스 값 벡터(230)에서 한번 초과로 등장 할 수 있다. 추가적으로 또는 대안적으로, 가능한 인덱스 값은 인덱스 값 벡터(230)에서 등장하지 않을 수 있다. 다수의 인덱스 값을 각각의 인덱스 값은 입력 계수 테이블(220)의 구분적 분석 계수들의 세트에 대응할 수 있다. 예를 들어, 0의 인덱스 값은 a0 및 b0에 대응할 수 있고, 1의 인덱스 값은 a1 및 b1에 대응할 수 있다.

[0039] 인덱스 값 벡터(230)의 다수의 인덱스 값을 입력 벡터(210)의 입력 값들 및 입력 계수 테이블(220)의 하나 이상의 시프트 값을 기초하여 생성될 수 있다. 예를 들어, 프로세서는 입력 계수 테이블(220)의 하나 이상의 시프트 값을 사용하여 입력 벡터(210)의 입력 값들에 대해 하나 이상의 이진 시프트 연산들을 수행할 수 있다. 예시하자면, 프로세서는 각각의 입력 값을 대응하는 시프트 값만큼 우측 시프트(예를 들어, 나눗셈) 할 수 있다. 예를 들어, 프로세서는 제1 비트 값(예를 들어, 다중 숫자 비트 값)을 생성하기 위해 입력 벡터(210)의 제1 입력 값을 입력 계수 테이블(220)의 제1 시프트 값 shiftvalue0만큼 우측 시프트할 수 있다. 제1 비트 값의 특정 수의 상위 숫자들이 인덱스 값 벡터(230)의 제1 인덱스 값 2를 생성하기 위해 사용될 수 있다. 일부 구현들에서, 시프트 값을 각각은 동일한 값을 가질 수 있다. 다른 구현들에서, 하나 이상의 시프트 값을 상이한 값을 가질 수 있다.

[0040] 인덱스 값 벡터(230)의 다수의 인덱스 값을 계수 벡터(240)를 생성하기 위해 프로세서에 의해 사용될 수 있다. 계수 벡터(240)는 제4 레지스터(242)에 저장될 수 있다. 프로세서는 계수 벡터(240)를 생성하기 위해 도 1의 순열 네트워크(114)와 같은 순열 네트워크에 대한 입력으로서 인덱스 값 벡터(230)를 사용할 수 있다. 예시하자면, 순열 네트워크는 계수 벡터(240)를 생성하기 위해 인덱스 값 벡터(230)에 기초하여, 제2 레지스터(222)(입력 계수 테이블(220)을 포함함)부터의 구분적 분석 계수들의 하나 이상의 세트들을 선택하여 제4 레지스터(242)에 라우팅할 수 있다. 계수 벡터(240)는 도 3을 참조하여 설명된 바와 같이 벡터 산술 명령의 실행 동안 사용될 수 있다. 예를 들어, 순열 네트워크는 인덱스 값 벡터(230)의 제1 인덱스 값 2에 기초하여 구분적 분석 계수들 a2 및 b2의 제1 세트를 선택할 수 있다. 순열 네트워크는 대응하는 입력 값인 제1 입력 값 x0에 기초하여 구분적 분석 계수들의 제1 세트를 제4 레지스터(242)에 라우팅할 수 있다. 입력 벡터(210)의 제1 입력 값 x0에 대응하는 구분적 분석 계수들의 제1 세트는 계수 벡터(240)의 제1 부분에 저장될 수 있다. 계수 벡터(240)의 제1 부분은 짹수 입력 값을(예를 들어, x0 및 x2)에 대한 구분적 분석 계수들에 대응할 수 있다.

[0041] 다른 예로, 제2 입력 값 x1에 대응하는 구분적 분석 계수들 a3 및 b3의 제2 세트는 계수 벡터(240)의 제2 부분에 저장될 수 있다. 계수 벡터(240)의 제2 부분은 홀수 입력 값을(예를 들어, x1 및 x3)에 대한 구분적 분석 계수들에 대응할 수 있다. 일부 구현들에서, 계수 벡터(240)의 제2 부분은 다른 레지스터에 저장될 수 있다.

[0042] 연산 동안, 프로세서는 벡터화된 테이블 루업 명령(201)을 수신할 수 있다. 프로세서는 각각의 입력 값에 대해, 인덱스 값 벡터(230)의 인덱스 값을 생성하기 위해, 입력 벡터(210)의 입력 값들 및 입력 계수 테이블(220)의 하나 이상의 시프트 값을 사용하여 벡터화된 테이블 루업 명령(201)을 실행할 수 있다. 프로세서는 입력 벡터(210)를 수신하고 입력 벡터(210)를 제1 레지스터(212)에 저장할 수 있다. 특정 구현에서, 입력 벡터

(210)의 입력 값들은 프로세서에 의해 수신되기 전에 타겟 범위로 정규화될 수 있다.

[0043]

[0051] 프로세서는 메모리로부터 입력 계수 테이블(220)을 리트리브(또는 수신)할 수 있다. 프로세서는 제2 레지스터(222)의 입력 계수 테이블(220)을 저장할 수 있다. 일부 구현들에서, 프로세서는 벡터화된 테이블 루업 명령(201)을 실행하기 전에 입력 계수 테이블(220)을 제2 레지스터(222)에 로딩할 수 있다. 프로세서는 테이블을 로딩 명령을 수신하는 것에 대한 응답으로 제2 레지스터(222)에 입력 계수 테이블(220)을 로딩할 수 있다. 특정 구현에서, 프로세서(102)는 입력 계수 테이블(220)의 제1 부분을 제2 레지스터(222)에 그리고 입력 계수 테이블(220)의 제2 부분을 다른 레지스터에 저장할 수 있다. 프로세서는 인덱스 값들을 생성하기 위해 입력 벡터(210)의 입력 값들에 하나 이상의 시프트 값을 적용할 수 있다. 예로서, 프로세서는 인덱스 값들을 생성하기 위해 입력 값을 6의 시프트 값만큼 우측 시프트(예를 들어, 나눗셈)할 수 있다(예를 들어, 입력 값들의 이진 표현들의 시작에 6개의 제로들을 추가할 수 있다). 인덱스 값들은 도 1을 참조하여 설명된 바와 같이 시프트된 입력 값의 이진 표현과 연관된 다수의 상위 비트들을 표현할 수 있다.

[0044]

[0052] 벡터화된 테이블 루업 명령(201)의 실행은 또한, 계수 벡터(240)를 생성하도록 입력 계수 테이블(220)의 구분적 분석 계수들의 하나 이상의 세트들을 선택하기 위해 인덱스 값 벡터(230)의 인덱스 값을 사용하는 것을 포함할 수 있다. 따라서, 프로세서는 각각의 입력 값에 대해, 계수 벡터(240)에 포함될 구분적 분석 계수들의 대응하는 세트를 선택할 수 있다. 예를 들어, 순열 네트워크는, 인덱스 값을 사용하여, 입력 계수 테이블(220)의 구분적 분석 계수들의 하나 이상의 세트들을 선택할 수 있고, 계수 벡터(240)를 생성하기 위해 제2 레지스터(222)로부터의 구분적 분석 계수들의 하나 이상의 세트들을 제4 레지스터(242)에 라우팅할 수 있다.

[0045]

[0053] 계수 벡터(240)는 도 2에 예시된 바와 같이 "크로스바" 구성을 사용하여 구분적 분석 계수들의 하나 이상의 선택된 세트들(예를 들어, 대응하는 세트들)을 저장할 수 있다. 예를 들어, 크로스바 구성은, 짹수 입력 값들(예를 들어, x_0 , x_2 등)에 대응하는 계수들을 계수 벡터의 제1 부분(예를 들어, 제4 레지스터(242) 또는 제4 레지스터(242)의 제1 부분)에 저장하고, 홀수 입력 값들(예를 들어, x_1 , x_3 등)에 대응하는 계수들을 계수 벡터의 제2 부분(예를 들어, 제4 레지스터(242)의 제2 부분 또는 다른 레지스터)에 저장하기 위해 사용될 수 있다. 크로스바 구성에 따라 구분적 분석 계수들의 하나 이상의 선택된 세트들을 라우팅 및 저장(예를 들어, 크로스바 구성을 포함하는 계수 벡터(240)를 생성)함으로써, 순열 네트워크는 수열에 따라 구분적 분석 계수들의 하나 이상의 선택된 세트들을 라우팅 및 저장하여 계수 벡터(240)를 더 신속하게 라우팅 및 생성할 수 있다. 추가적으로, 크로스바 구성에 따라 데이터를 저장함으로써, 벡터화된 테이블 루업 명령(201)은 종래의 순열 네트워크들과 호환가능(예를 들어, 그에 의해 실행가능)할 수 있다.

[0046]

[0054] 예시하자면, 입력 벡터(210)의 제1 입력 값 x_0 의 경우, 순열 네트워크는 인덱스 값 벡터(230)의 제1 인덱스 값에 기초하여 구분적 분석 계수들의 대응하는 세트를 선택할 수 있다. 예를 들어, 순열 네트워크는 제1 인덱스 값 2에 대응하는 구분적 분석 계수들 a_2 및 b_2 의 제1 세트를 선택한다. 순열 네트워크는 구분적 분석 계수들의 제1 세트의 제1 구분적 분석 계수 a_2 및 제2 구분적 분석 계수 b_2 를 제2 레지스터(222)(입력 계수 테이블(220)을 포함함)로부터 제4 레지스터(242)에 라우팅할 수 있다. 구분적 분석 계수들의 제1 세트의 제1 구분적 분석 계수 a_2 및 제2 구분적 분석 계수 b_2 는 제1 입력 값에 기초하여 제4 레지스터(242)에 저장될 수 있다. 예를 들어, 제1 입력 값 x_0 이 짹수 값이기 때문에, 구분적 분석 계수들의 제1 세트의 제1 구분적 분석 계수 a_2 및 제2 구분적 분석 계수 b_2 는 제4 레지스터(242)의 제1 절반에 저장될 수 있다. 구분적 분석 계수들이 저장되는 제4 레지스터(242)의 제1 절반의 위치는 입력 벡터(210)의 대응하는 입력 값의 위치에 대응할 수 있다. 예를 들어, 구분적 분석 계수들 a_2 및 b_2 의 제1 세트는 제4 레지스터(242)의 제1 절반의 처음 2개의 위치들에 저장될 수 있고, 다음 짹수 입력 값(예를 들어, x_2)에 대응하는 구분적 계수들은 제4 레지스터(242)의 제1 절반의 다음 2개의 위치들에 저장될 수 있다. 따라서, 순열 네트워크는 계수 벡터(240)에 포함될 구분적 분석 계수들의 하나 이상의 세트들 중 구분적 분석 계수들 제1 세트를 선택(또는 생성)할 수 있다.

[0047]

[0055] 입력 벡터(210)의 제2 입력 값 x_1 의 경우, 순열 네트워크는 인덱스 값 벡터(230)의 제2 인덱스 값에 기초하여 구분적 분석 계수들의 대응하는 세트를 선택할 수 있다. 예를 들어, 순열 네트워크는 제2 인덱스 값 3에 대응하는 구분적 분석 계수들 a_3 및 b_3 의 제2 세트를 선택한다. 순열 네트워크는 구분적 분석 계수들의 제2 세트의 제1 구분적 분석 계수 a_3 및 제2 구분적 분석 계수 b_3 를 제2 레지스터(222)(입력 계수 테이블(220)을 포함함)로부터 제4 레지스터(242)에 라우팅할 수 있다. 구분적 분석 계수들의 제2 세트의 제1 구분적 분석 계수 a_3 및 제2 구분적 분석 계수 b_3 는 제2 입력 값에 기초하여 제4 레지스터(242)에 저장될 수 있다. 예를 들어, 제2 입력 값 x_1 이 홀수 값이기 때문에, 구분적 분석 계수들의 제2 세트의 제1 구분적 분석 계수 a_3 및 제2 구분적 분석 계수 b_3 는 제4 레지스터(242)의 제2 절반(또는 다른 레지스터에) 저장될 수 있다. 제4 레지스터(242)의 제2 절반의 구분적 분석 계수들의 위치는 앞서 설명된 바와 같이, 입력 값 벡터(210)의 대응하는 입력

값의 위치에 대응할 수 있다.

[0048]

[0056] 추가로 예시하자면, 입력 벡터(210)의 제3 입력 값 x_2 의 경우, 순열 네트워크는 인덱스 값 벡터(230)의 제3 인덱스 값에 기초하여 구분적 분석 계수들의 대응하는 세트를 선택할 수 있다. 예를 들어, 순열 네트워크는 제3 인덱스 값 0에 대응하는 구분적 분석 계수들 a_0 및 b_0 의 제3 세트를 선택한다. 순열 네트워크는 구분적 분석 계수들의 제3 세트의 제1 구분적 분석 계수 a_0 및 제2 구분적 분석 계수 b_0 을 제2 레지스터(222)(입력 계수 테이블(220)을 포함함)로부터 제4 레지스터(242)에 라우팅할 수 있다. 구분적 분석 계수들의 제3 세트의 제3 구분적 분석 계수 a_0 및 제2 구분적 분석 계수 b_0 은 제3 입력 값에 기초하여 제4 레지스터(242)에 저장될 수 있다. 예를 들어, 제3 입력 값 x_2 가 짹수 값이기 때문에, 구분적 분석 계수들의 제3 세트의 제1 구분적 분석 계수 a_0 및 제2 구분적 분석 계수 b_0 은 제4 레지스터(242)의 제1 절반 내의 제2 위치 내에 저장될 수 있다.

[0049]

[0057] 입력 벡터(210)의 제4 입력 값 x_3 의 경우, 순열 네트워크는 인덱스 값 벡터(230)의 제4 인덱스 값에 기초하여 구분적 분석 계수들의 대응하는 세트를 선택할 수 있다. 예를 들어, 순열 네트워크는 제4 인덱스 값 1에 대응하는 구분적 분석 계수들 a_1 및 b_1 의 제4 세트를 선택한다. 순열 네트워크는 구분적 분석 계수들의 제4 세트의 제1 구분적 분석 계수 a_1 및 제2 구분적 분석 계수 b_1 을 제2 레지스터(222)(입력 계수 테이블(220)을 포함함)로부터 제4 레지스터(242)에 라우팅할 수 있다. 구분적 분석 계수들의 제4 세트의 제1 구분적 분석 계수 a_1 및 제2 구분적 분석 계수 b_1 은 제4 입력 값에 기초하여 제4 레지스터(242)에 저장될 수 있다. 예를 들어, 제4 입력 값 x_3 이 홀수 값이기 때문에, 구분적 분석 계수들의 제4 세트의 제1 구분적 분석 계수 a_1 및 제2 구분적 분석 계수 b_1 은 제4 레지스터(242)의 제2 절반 내의 제2 위치에(또는 다른 레지스터에) 저장될 수 있다. 따라서, 순열 네트워크는 구분적 분석 계수들의 하나 이상의 세트들을 선택하여 제4 레지스터(242)에(또는 제4 레지스터(242) 및 다른 레지스터에) 라우팅함으로써 계수 벡터(240)를 생성할 수 있다.

[0050]

[0058] 계수 벡터(240)는 도 3 및 도 5를 참조하여 설명된 바와 같이 벡터 산술 명령에 대한 입력으로서 사용될 수 있다. 단일 명령의 실행 동안 계수 벡터(240)를 생성함으로써, 시스템은 계수 벡터(240)를 생성하기 위해 다수의 명령들을 실행하는 시스템들에 비해, 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다. 계수 벡터(240)를 생성하기 위해 순열 네트워크를 사용함으로써, 시스템은 레인마다 또는 엘리먼트마다의 루프 테이블(예를 들어, 각각의 입력 값에 대한 별개의 루프 테이블)에 비해 시스템의 속도를 추가로 증가시키고 복잡도를 감소시킬 수 있다.

[0051]

[0059] 도 3을 참조하면, 벡터화 산술 명령(301)을 실행하는 예시적 프로세스의 도면(300)이 개시되어 있다. 벡터 산술 명령(301)은 도 1의 프로세서(102)와 같은 프로세서에 의해 실행될 수 있다. 도 3에 예시된 바와 같이, 벡터 산술 명령(301)은 명령 명칭(303)(vmpyaddeoh)을 포함할 수 있다(예를 들어, opcode). 도 3의 특정 구현에서, 명령 명칭은 벡터 곱셈 연산(예를 들어, mpy) 및 벡터 가산 연산(예를 들어, add)을 표시할 수 있다. 명령 명칭은 벡터 연산들의 순서를 표시할 수 있다. 예를 들어, "mpy"는 제1 타입의(예를 들어, 갖는) 계수들을 입력 값들과 곱하고, "add"는 제2 타입의(예를 들어, 갖는) 계수들을, 제1 타입의 계수들과 입력 값들의 곱에 가산하는 것을 표시할 수 있다. 명령 명칭(303)은 구성 표시자(304) 및 단어 길이 표시자(305)를 포함할 수 있다. 구성 표시자(304)의 값(eo)은 계수 벡터(240)의 순서를 표시할 수 있다. 구성 표시자(304)의 값(eo)은 계수 벡터(240)의 짹수-홀수 방향을 표시할 수 있다. 예를 들어, 짹수-홀수 방향은 "크로스바" 구성을 포함하거나 그에 대응할 수 있고, 여기서 짹수 입력 값들(예를 들어, x_0 , x_2 등)에 대응하는 계수들은 계수 벡터(240)의 제1 부분에 저장될 수 있고, 홀수 입력 값들(예를 들어, x_1 , x_3 등)에 대응하는 계수들은 계수 벡터(240)의 제2 부분에 저장될 수 있다. 단어 길이 표시자(305)의 값은 입력 값들의 비트 크기를 표시할 수 있다. 도 2에 예시된 예에서, 단어 길이 표시자(305)의 값은 "h"이고, 명령 단어(예를 들어, 16 비트들)의 "절반"인 입력 값의 크기를 표시하기 위해 사용될 수 있다.

[0052]

[0060] 벡터 산술 명령(301)은 또한 제1 필드(302)(Vd), 제2 필드(306)(Vdd) 및 제3 필드(307)(Vv)와 같은 하나 이상의 필드들(예를 들어, 피연산자들)을 포함할 수 있다. 제1 필드(302)에 저장된 제1 값은 결과 벡터(350)(예를 들어, 벡터 Vd)를 표시할 수 있다. 제2 필드(306)에 저장된 제2 값은 계수 벡터(240)(예를 들어, 벡터 Vdd)를 표시할 수 있고, 제3 필드(307)에 저장된 제3 값은 입력 벡터(210)(예를 들어, 벡터 Vv)를 표시할 수 있다. 3개의 필드들이 예시되지만, 다른 구현들에서 벡터 산술 명령(301)은 3개 초과의 필드들 또는 3개 미만의 필드들을 포함할 수 있다.

[0053]

[0061] 프로세서는 벡터 산술 명령(301)의 실행 동안 다양한 동작들을 수행하도록 구성될 수 있다. 도 2를 참조하여 설명된 바와 같이, 벡터화된 테이블 루프 명령(201)의 실행 동안, 입력 벡터(210)(Vv)는 제1 레지스터(212)에 저장되고, 계수 벡터(240)(Vdd)는 제4 레지스터(242)에 저장된다. 프로세서는 입력 벡터(210) 및 계수

벡터(240)를 프로세싱할 수 있고, 결과 벡터(350)를 생성할 수 있다. 결과 벡터(350)는 제5 레지스터(352)에 저장될 수 있다. 결과 벡터(350)는 함수의 추정된 출력 값들을 포함할 수 있다. 도 3에서, 추정된 출력 값들은 $y_0 - y_3$ 을 포함한다. 추정된 출력 값들은 입력 값들에 기초한 함수의 근사화들에 대응할 수 있다. 예시하자면, 벡터 산술 명령(301)을 실행하는 것은 입력 벡터(210)의 특정 입력 값 및 계수 벡터(240)의 구분적 분석 계수들의 대응하는 세트에 대한 하나 이상의 산술 연산들을 수행함으로써 특정한 추정된 출력 값을 생성할 수 있다.

[0054] 일부 구현들에서, 결과 벡터(350)는 도 1의 산술 로직 회로(112)과 같은 프로세서의 산술 로직 회로에 의해 생성될 수 있다. 산술 로직 회로는 입력 벡터(210) 및 계수 벡터(240)에 대한 산술 연산들을 수행하도록 구성될 수 있다. 예를 들어, 산술 로직 회로(112)는 곱들을 생성하기 위해 입력 벡터(210)의 입력 값들을, 제1 타입의 계수 벡터(240)의(예를 들어, 갖는) 구분적 분석 계수들과 곱할 수 있다. 산술 로직 회로는 결과 벡터(350)에 포함되는 추정된 출력 값들을 생성하기 위해 제2 타입의 계수 벡터(240)의(예를 들어, 갖는) 구분적 분석 계수들 및 곱들에 기초하여 합들을 생성할 수 있다. 예시하자면, 산술 로직 회로는 제1 곱 a_2*x_0 을 생성하기 위해 제1 타입의 제1 구분적 분석 계수 a_2 와 제1 입력 값 x_0 을 곱할 수 있다. 산술 로직 회로는 제1 추정된 출력 값 $y_0 = a_2*x_0+b_2$ 를 생성하기 위해 곱 a_2*x_0 을 제2 타입의 제2 구분적 분석 계수 b_2 와 가산할 수 있다. $y_1 - y_3$ 의 추정된 출력 값은 유사한 방식으로 생성될 수 있다.

[0055] 연산 동안, 프로세서는 벡터 산술 명령(301)을 수신할 수 있다. 프로세서는 결과 벡터(350)의 추정된 출력 값들을 생성하기 위해 입력 벡터(210)의 입력 값들 및 계수 벡터(240)의 구분적 분석 계수들을 사용하여 벡터 산술 명령(301)을 실행할 수 있다. 프로세서의 산술 로직 회로는 결과 벡터(350)를 생성하기 위해 입력 벡터(210) 및 계수 벡터(240)를 사용하여 하나 이상의 산술 연산들을 수행할 수 있다. 예시하자면, 프로세서의 산술 로직 회로는 입력 벡터(210)의 각각의 입력 값에 대해, 곱을 생성하기 위해 (계수 벡터(240)로부터) 구분적 분석 계수들의 대응하는 세트의 제1 구분적 분석 계수와 입력 값들을 곱할 수 있다. 프로세서의 산술 로직 회로는 구분적 분석 계수들의 대응하는 세트의 제2 구분적 분석 계수 및 곱에 기초하여 합을 생성할 수 있다. 합은 결과 벡터(350)의 추정된 출력 값으로서 저장될 수 있다. 추정된 출력 값들은 함수의 입력 값들의 추정들(또는 근사화들)을 표현할 수 있다. 단일 명령의 실행 동안 결과 벡터(350)를 생성함으로써, 시스템은 결과 벡터(350)를 생성하기 위해 다수의 명령들을 실행하는 시스템들에 비해, 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다.

[0056] 곱셈 및 가산 연산들이 설명되었지만, 벡터 산술 명령(301)은 이에 제한되지 않는다. 예를 들어, 벡터 산술 명령(301)은 입력 벡터(210) 및 계수 벡터(240)를 사용하여 수행될 하나 이상의 산술 연산들을 표시할 수 있다. 하나 이상의 산술 연산들은 가산 연산들, 감산 연산들, 곱셈 연산들, 나눗셈 연산들 또는 이들의 조합을 포함할 수 있다. 하나 이상의 산술 연산들은 명령 명칭(303)에 의해 표시될 수 있다. 일부 구현들에서, 함수의 출력 값들을 추정하기 위해 사용되는 하나 이상의 산술 연산들은 함수의 비선형 근사화(예를 들어, $y(a, x, b)$)에 대응할 수 있다. 예시적인 비제한적 예로서, 비선형 근사화는 계수들을 입력 값들로 나눗셈하는 것을 포함할 수 있다(예를 들어, a/x).

[0057] 도 4를 참조하면, 벡터화된 테이블 루프 명령(401)을 실행하는 다른 예시적인 프로세스의 도면(400)이 개시되어 있다. 벡터화된 테이블 루프 명령(401)은 도 1의 프로세서(102)와 같은 프로세서에 의해 실행될 수 있다. 도 4에 예시된 바와 같이, 벡터화된 테이블 루프 명령(401)은 opcode와 같은 명령 명칭(203)(vlutNseg h)을 포함할 수 있다. 벡터화된 테이블 루프 명령(401)은 도 2를 참조하여 설명된 바와 같이 하나 이상의 표시자들, 하나 이상의 필드들(예를 들어, 피연산자들) 또는 이들의 조합을 포함할 수 있다. 세그먼트 표시자 "N"은 구분적 분석의 다수의 세그먼트들을 표시할 수 있다. 예시적인 비제한적 예들로서, N은 2, 4, 8, 16, 32 등의 값에 대응할 수 있다.

[0058] 프로세서는 벡터화된 테이블 루프 명령(401)의 실행 동안 다양한 동작들을 수행하도록 구성될 수 있다. 프로세서는 입력 벡터(410)를 수신할 수 있다. 입력 벡터(410)는 입력 값들의 수 M을 포함할 수 있다. 특정 구현에서, M은 64이다. 다른 구현들에서, M은 64 미만 또는 64 초과일 수 있다. 프로세서(또는 계수 결정 유닛)는 입력 계수 테이블(420)을 리트리브(또는 수신)할 수 있다. 입력 계수 테이블(420)은 다수의 시프트 값들을 포함할 수 있다. 입력 계수 테이블(420)에 포함된 시프트 값들의 수는 입력 값들의 수(예를 들어, M)와 동일할 수 있다. 입력 계수 테이블(420)은 또한 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 구분적 분석 계수들의 세트들의 수는 구분적 분석의 세그먼트들의 수(예를 들어, N)와 동일할 수 있다. 프로세서는 도 2를 참조하여 설명된 방식으로 인덱스 값 벡터(430)를 생성할 수 있다. 인덱스 값 벡터(430)는 다수의 인덱스 값들을 포함할 수 있다. 입력 계수 테이블(420)에 포함된 인덱스 값들의 수는 입력 값들의 수(예를 들어, M)와

동일할 수 있다.

- [0059] [0067] 프로세서는 도 2를 참조하여 설명된 이러한 방식으로 계수 벡터(440)를 생성할 수 있다. 계수 벡터(440)는 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 계수 벡터(440)에 포함된 구분적 분석 계수들의 세트들의 수는 입력 값들의 수(예를 들어, M)와 동일할 수 있다. 구분적 분석 계수들의 세트들은 도 2를 참조하여 설명된 바와 같이 크로스바 구성으로 계수 벡터(440)에 저장될 수 있다. 예를 들어, 짹수 입력 값들(예를 들어, x0, x2, . . . , xM-2)에 대응하는 계수들은 계수 벡터(440)의 제1 부분에 저장될 수 있고, 홀수 입력 값들(예를 들어, x1, x3, . . . , xM-1)에 대응하는 계수들은 계수 벡터(440)의 제2 부분에 저장될 수 있다. 추가적으로, 입력 계수 테이블(420) 및 계수 벡터(440)는 입력 벡터(410) 및 인덱스 값 벡터(430) 또는 둘 모두와 동일한 수의 비트들의 2배를 포함할 수 있다. 예를 들어, 2개의 계수들은 각각의 입력 값에 대해 선택될 수 있고, 따라서 계수 벡터(440)는 입력 벡터(410)와 동일한 수의 엘리먼트들의 2배를 저장할 수 있다.
- [0060] [0068] 일부 구현들에서, 프로세서의 레지스터들은 특정 크기일 수 있다(예를 들어, 제1 수의 비트들을 저장하도록 구성될 수 있다). 입력 벡터(410), 인덱스 값 벡터(430) 또는 둘 모두는 특정 크기일 수 있다. 입력 계수 테이블(420) 및 계수 벡터(440)는 특정 크기보다 클 수 있다. 일부 구현들에서, 입력 계수 테이블(420) 및 계수 벡터(440)는 레지스터들의 크기의 2배일 수 있다. 예를 들어, 레지스터들은 1024 비트들을 저장할 수 있고, 입력 계수 테이블(420) 및 계수 벡터(440)는 2048 비트들일 수 있다. 따라서, 입력 계수 테이블(420) 및 계수 벡터(440) 각각은 2개의 레지스터들과 같은 다수의 레지스터들에 저장될 수 있다. 예를 들어, 짹수 입력 값들(예를 들어, x0, x2 등)에 대응하는 구분적 분석 계수들의 세트들을 포함하는 계수 벡터(440)의 제1 부분은 제1 레지스터에 저장될 수 있고, 홀수 입력 값들(예를 들어, x1, x3 등)에 대응하는 구분적 분석 계수들의 세트들을 포함하는 계수 벡터(440)의 제2 부분은 제2 레지스터에 저장될 수 있다.
- [0061] [0069] 연산 동안, 프로세서는 벡터화된 테이블 루업 명령(401)을 수신할 수 있다. 프로세서는 각각의 입력 값에 대해, 인덱스 값 벡터(430)의 인덱스 값을 생성하기 위해, 입력 벡터(410)의 입력 값들 및 입력 계수 테이블(420)의 하나 이상의 시프트 값을 사용하여 벡터화된 테이블 루업 명령(401)을 실행할 수 있다. 인덱스 값들은 도 1 및 도 2를 참조하여 설명된 바와 같이 시프트된 입력 값의 이진 표현과 연관된 다수의 상위 비트들을 표현할 수 있다.
- [0062] [0070] 벡터화된 테이블 루업 명령(401)의 실행은 또한, 각각의 입력 값에 대해, 계수 벡터(440)의 구분적 분석 계수들의 대응하는 세트를 생성(또는 선택)하도록 인덱스 값 벡터(430)의 인덱스 값을 및 입력 계수 테이블(420)의 구분적 분석 계수들의 하나 이상의 세트들을 사용하는 것을 포함할 수 있다. 예시하자면, 입력 벡터(410)의 제1 입력 값 x0의 경우, 순열 네트워크는 인덱스 값 벡터(430)의 제1 인덱스 값 2에 기초하여 구분적 분석 계수들 a2 및 b2의 대응하는 세트를 선택할 수 있다. 계수 벡터(440)는 도 2를 참조하여 설명된 바와 같이 크로스바 구성으로 구분적 분석 계수들의 하나 이상의 세트들(예를 들어, 대응하는 세트들)을 저장할 수 있다.
- [0063] [0071] 계수 벡터(440)는 도 3 및 도 5를 참조하여 설명된 바와 같이 벡터 산술 명령에 대한 입력으로서 사용될 수 있다. 단일 명령의 실행 동안 계수 벡터(440)를 생성함으로써, 시스템은 계수 벡터(440)를 생성하기 위해 다수의 명령들을 실행하는 시스템들에 비해, 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다. 계수 벡터(440)를 생성하기 위해 순열 네트워크를 사용함으로써, 시스템은 레인마다 또는 엘리먼트마다의 루업 테이블(예를 들어, 각각의 입력 값에 대한 별개의 루업 테이블)에 비해 추가로 더 적은 메모리를 사용할 수 있고 증가된 속도를 가질 수 있다.
- [0064] [0072] 도 5을 참조하면, 벡터화 산술 명령(501)을 실행하는 다른 예시적 프로세스의 도면(500)이 개시되어 있다. 벡터 산술 명령(501)은 도 1의 프로세서(102)와 같은 프로세서에 의해 실행될 수 있다. 도 5에 예시된 바와 같이, 벡터 산술 명령(501)은 opcode와 같은 명령 명칭(vmpyaddeoh)을 포함할 수 있다. 벡터 산술 명령(501)은 도 3을 참조하여 설명된 바와 같이 하나 이상의 표시자들, 하나 이상의 필드들(예를 들어, 피연산자들) 또는 이들의 조합을 포함할 수 있다.
- [0065] [0073] 프로세서는 벡터 산술 명령(501)의 실행 동안 다양한 동작들을 수행하도록 구성될 수 있다. 도 4를 참조하여 설명된 바와 같이, 벡터화된 테이블 루업 명령(401)의 실행 동안, 입력 벡터(410)(Vv)가 수신되고, 계수 벡터(440)(Vdd)가 생성된다. 벡터 산술 명령(501)의 실행 동안, 프로세서는 입력 벡터(410) 및 계수 벡터(440)를 프로세싱할 수 있고, 결과 벡터(550)를 생성할 수 있다. 결과 벡터(550)는 함수의 다수의 추정된 출력 값을 포함할 수 있다. 함수의 다수의 추정된 출력 값을들은 입력 벡터(410)의 입력 값들의 수와 동일할 수 있다(예를 들어, 입력 벡터(410) 및 결과 벡터(550)는 M개의 엘리먼트들을 포함할 수 있다).

- [0066] [0074] 연산 동안, 프로세서는 벡터 산술 명령(501)을 수신할 수 있다. 프로세서는 도 3을 참조하여 설명된 바와 같이 결과 벡터(550)의 추정된 출력 값들을 생성하기 위해 입력 벡터(410)의 입력 값들 및 계수 벡터(440)의 구분적 분석 계수들을 사용하여 벡터 산술 명령(501)을 실행할 수 있다. 예시하자면, 산술 로직 회로는 제1 곱 a2*x0을 생성하기 위해 제1 타입의 제1 구분적 분석 계수 a2와 제1 입력 값 x0을 곱할 수 있다. 산술 로직 회로는 제1 추정된 출력 값 $y_0 = a2*x0+b2$ 를 생성하기 위해 곱 a2*x0을 제2 타입의 제2 구분적 분석 계수 b2와 가산할 수 있다. $y_1 - y_{M-1}$ 의 추정된 출력 값은 유사한 방식으로 생성될 수 있다. 단일 명령의 실행 동안 결과 벡터(550)를 생성함으로써, 시스템은 결과 벡터(550)를 생성하기 위해 다수의 명령들을 실행하는 시스템들에 비해, 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다.
- [0067] [0075] 도 6을 참조하면, 단일 구분적 근사화 명령(601)을 실행하는 특정한 예시적 프로세스의 도면(600)이 개시되어 있다. 단일 구분적 근사화 명령(601)은 단일 벡터 명령일 수 있다. 단일 구분적 근사화 명령(601)은 도 1의 프로세서(102)와 같은 프로세서에 의해 실행될 수 있다.
- [0068] [0076] 도 6에 예시된 바와 같이, 단일 구분적 근사화 명령(601)은 명령 명칭(603)(pwlalut2segh)을 포함할 수 있다(예를 들어, opcode). 명령 명칭(603)은 세그먼트 표시자(604) 및 단어 길이 표시자(605)와 같은 하나 이상의 표시자들을 포함할 수 있다. 세그먼트 표시자(604)의 값은 구분적 분석의 다수의 세그먼트들을 표시할 수 있다. 세그먼트 표시자(604)의 값은 또한 입력 계수 테이블(620)에서 구분적 분석 계수들의 그룹을 형성하는 구분적 분석 계수들의 수를 표시하거나 또한 이에 대응할 수 있다. 구분적 분석 계수들(예를 들어, 구분적 분석 계수들의 그룹)은 입력 계수 테이블(620)에서 한번 이상 반복될 수 있다. 도 6에 예시된 예에서, 세그먼트 표시자는 구분적 분석의 2개의 세그먼트들이 존재함(예를 들어, 그룹에 구분적 분석 계수들의 2개의 세트들(또는 4개의 구분적 분석 계수들)이 존재함)을 표시한다. 구분적 분석 계수들의 반복된 그룹들을 가짐으로써, 벡터 연산들은, 입력 값들의 수가 세그먼트 표시자(604)에 의해 표시된 세그먼트들의 수보다 큰 경우 단일 명령에서(또는 단일 연산으로) 수행될 수 있다. 벡터 연산들은 단일 명령에서 수행될 수 있는데, 이는 구분적 분석 계수들의 세트들의 수가 입력 값들의 수에 매칭할 수 있기 때문이다. 예를 들어, 도 6에 예시된 바와 같이, 입력 벡터(610)는 4개의 입력 값들(x0-x3)을 포함하고, 세그먼트 표시자는 2개의 세그먼트 구분적 근사화(구분적 분석 계수들 a0-a1, b0-b1의 2개의 세트들)를 표시한다. 따라서, 제2 그룹(그룹 2)을 가짐으로써, 구분적 분석 계수들(예를 들어, 4(그룹 1의 a0-a1, b0-b1 및 그룹 2의 a0-a1, b0-b1))의 세트들의 수가 입력 값들의 수(예를 들어, 4)와 매칭할 수 있다. 따라서, 단일 구분적 근사화 명령(601)은 구분적 근사화의 복잡도를 증가시키지 않고 더 큰 입력 벡터들을 수신할 수 있다.
- [0069] [0077] 단어 길이 표시자(605)의 값은 입력 값의 비트 크기를 표시할 수 있다. 도 6에 예시된 예에서, 단어 길이 표시자(605)의 값은 "h"이고, 명령 단어(예를 들어, 16 비트들)의 "절반"인 입력 값의 크기를 표시하기 위해 사용될 수 있다. 입력 계수 테이블(620)이 2개의 그룹들을 갖는 것으로 예시되어 있지만, 입력 계수 테이블(620)은 더 많은 그룹들 또는 더 적은 그룹들을 포함할 수 있다. 입력 계수 테이블(620)의 그룹들의 수는 세그먼트 표시자(604), 입력 벡터(610)의 입력 값들의 수, 입력 값들의 단어 길이 또는 이들의 조합에 기초할 수 있다.
- [0070] [0078] 단일 구분적 근사화 명령(601)은 또한 제1 필드(602)(Vd), 제2 필드(606)(Vuu) 및 제3 필드(607)(Vv)와 같은 하나 이상의 필드들(예를 들어, 피연산자들)을 포함할 수 있다. 제1 필드(602)에 저장된 제1 값은 결과 벡터(650)(예를 들어, 벡터 Vd)를 표시할 수 있다. 제2 필드(606)에 저장된 제2 값은 입력 계수 테이블(620)(예를 들어, 벡터 Vu)을 표시할 수 있고, 제3 필드(607)에 저장된 제3 값은 입력 벡터(610)(예를 들어, 벡터 Vv)를 표시할 수 있다. 3개의 필드들이 예시되지만, 다른 구현들에서 단일 구분적 근사화 명령(601)은 3개 초과의 필드들 또는 3개 미만의 필드들을 포함할 수 있다.
- [0071] [0079] 프로세서는 단일 구분적 근사화 명령(601)의 실행 동안 다양한 동작들을 수행하도록 구성될 수 있다. 프로세서는 x0, x1, x2 및 x3과 같은 복수의 입력 값들을 포함하는 입력 벡터(610)(vv)를 수신하도록 구성될 수 있다. 입력 벡터(610)(예를 들어, 입력 값들의 벡터)는 프로세서의 제1 레지스터(612)에 저장될 수 있다. 입력 벡터(610)의 복수의 입력 값들은 그룹들로 체계화될 수 있다. 예를 들어, x0 및 x1은 입력 값들(618)의 제1 그룹(예를 들어, 그룹 1)의 일부일 수 있고, x2 및 x3은 입력 값들의 제2 그룹(예를 들어, 그룹 2)의 일부일 수 있다. 그룹들은 입력 계수 테이블(620)에서 구분적 분석 계수들의 그룹들에 대응할 수 있다. 예를 들어, 입력 값들(618)의 제1 그룹(예를 들어, 그룹 1)은 입력 계수 테이블(620)에서 구분적 분석 계수들(628)의 제1 그룹(예를 들어, 그룹 1)에 대응할 수 있다.
- [0072] [0080] 프로세서는 인덱스 값 벡터(630)를 생성하기 위해 하나 이상의 시프트 값들을 사용하여 입력 벡터(610)

를 프로세싱할 수 있다. 일부 구현들에서, 단일 시프트 값이 사용될 수 있다. 예를 들어, 단일 시프트 값(예를 들어, 5개의 제로들)이 입력 벡터(610)의 각각의 입력 값에 적용될 수 있거나, 또는 동일한 값을 갖는 시프트 값들의 벡터가 입력 벡터(610)의 입력 값들에 적용될 수 있다. 다른 구현들에서, 상이한 시프트 값들이 사용될 수 있다. 예시적인 비제한적 예로서, 제1 시프트 값(예를 들어, 3개의 제로들)이 제1 입력 값에 적용될 수 있고, 제2 시프트 값(예를 들어, 6개의 제로들)이 제2 입력 값에 적용될 수 있다. 하나 이상의 시프트 값들은 고정(예를 들어, 저장 또는 미리 결정)될 수 있다. 하나 이상의 시프트 값들은 프로세서에 의해 생성되거나, 또는 메모리에 저장되고 프로세서에 의해 액세스될 수 있다. 특정 구현에서, 하나 이상의 시프트 값들은 도 9를 참조하여 설명된 바와 같이 시프트 값 생성기에 의해 생성될 수 있다. 하나 이상의 시프트 값들은 입력 벡터(610)의 입력 값들을 타겟 범위로 정규화하기 위해 프로세서에 의해 사용될 수 있다. 인덱스 값 벡터(630)는 프로세서의 제2 레지스터(622)에 저장될 수 있고, 다수의 인덱스 값들을 포함할 수 있다. 다수의 인덱스 값들의 각각의 인덱스 값은 입력 벡터(610)의 입력 값에 대응할 수 있다. 도 6는 인덱스 값 벡터(630)의 각각의 그룹에서 인덱스 값들이 반복(예를 들어, 한번 초과로 등장)될 수 있는 예를 예시한다. 다른 예들에서, 특정 인덱스 값은 인덱스 값 벡터(630)의 각각의 그룹에서 한번 등장할 수 있다. 추가적으로 또는 대안적으로, 가능한 인덱스 값은 인덱스 값 벡터(630)의 하나 이상의 그룹들에서 등장하지 않을 수 있다.

[0073]

[0081] 프로세서는 도 1의 메모리(106)와 같은 메모리로부터 입력 계수 테이블(620)을 수신(또는 리트리브)하도록 구성될 수 있다. 프로세서는 제3 레지스터(632)에 입력 계수 테이블(620)을 저장하도록 구성될 수 있다. 입력 계수 테이블(620)은 제1 세트(a0 및 b0) 및 제2 세트(a1 및 b1)와 같은 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 입력 계수 테이블(620)의 구분적 분석 계수들의 다수의 세트들이 반복될 수 있다. 예를 들어, 입력 계수 테이블(620)은 구분적 분석 계수들의 다수의 그룹들을 포함할 수 있다. 구분적 분석 계수들의 각각의 그룹은 구분적 분석 계수들의 다수의 세트로부터 구분적 분석 계수를 포함할 수 있다. 도 6에 예시된 바와 같이, 구분적 분석 계수들(628)의 제1 그룹은 a0 및 a1을 포함하고, 제2 그룹은 a0 및 a1을 포함한다. 다른 구현들에서, 프로세서는 단일 구분적 근사화 명령(601)의 실행 전에 메모리로부터의 입력 계수 테이블(620)을 제3 레지스터(632)에 로딩할 수 있다. 프로세서는 단일 구분적 근사화 명령(601)의 실행 동안 입력 계수 테이블(620)을 제3 레지스터(632)로부터 리트리브(또는 수신)할 수 있다.

[0074]

[0082] 다른 특정 구현에서, 입력 계수 테이블(620)은 프로세서의 다수의 레지스터들에 저장될 수 있다. 예를 들어, 입력 계수 테이블(620)은, 제1 타입 (a)의 다수의 구분적 분석 계수들을 포함하는 제1 섹션(또는 부분) 및 제2 타입 (b)의 다수의 구분적 분석 계수들을 포함하는 제2 섹션(또는 부분)을 포함할 수 있다. 입력 계수 테이블(620)의 제1 섹션은 제3 레지스터(632)에 저장될 수 있고, 입력 계수 테이블(620)의 제2 섹션은 다른 레지스터에 저장될 수 있다. 입력 계수 테이블(620)의 각각의 섹션은 입력 벡터의 그룹들에 대응하는 구분적 분석 계수들(또는 이들의 일부분)의 그룹을 포함할 수 있다. 예를 들어, 입력 계수 테이블(620)의 제1 섹션 및 제2 섹션 각각은 입력 값들(618)의 제1 그룹(예를 들어, 그룹 1)에 대응하는 구분적 분석 계수들(628)의 제1 그룹(예를 들어, 그룹 1)의 일부분을 포함할 수 있다. 입력 계수 테이블(620)의 제1 섹션 및 제2 섹션 각각은 또한 입력 값들의 제2 그룹(예를 들어, 그룹 2)에 대응하는 구분적 분석 계수들의 제2 그룹(예를 들어, 그룹 2)의 일부분을 포함할 수 있다.

[0075]

[0083] 인덱스 값 벡터(630)의 다수의 인덱스 값들은 결과 벡터(650)를 생성하기 위해 프로세서에 의해 사용될 수 있다. 결과 벡터(650)는 제4 레지스터(642)에 저장될 수 있다. 프로세서는 결과 벡터(650)를 생성하기 위해 도 1의 순열 네트워크(114)와 같은 순열 네트워크에 대한 입력으로서 인덱스 값 벡터(630)를 사용할 수 있다. 예시하자면, 순열 네트워크는 인덱스 값 벡터(630)에 기초하여, 제3 레지스터(632)(입력 계수 테이블(620)을 포함함)부터의 구분적 분석 계수들의 하나 이상의 세트들을 선택하여 산술 로직 회로에 라우팅할 수 있다. 구분적 분석 계수들의 하나 이상의 선택된 세트들은 도 1의 복수의 계수들(140)을 포함하거나 그에 대응할 수 있고, 산술 로직 회로는 도 1의 산술 로직 회로(112)를 포함하거나 그에 대응할 수 있다. 산술 로직 회로는 결과 벡터(650)를 생성하기 위해 입력 벡터(610) 및 입력 계수 테이블(620)로부터의 구분적 분석 계수들의 선택된 세트들에 대해 산술 연산들을 수행하도록 구성될 수 있다. 결과 벡터(650)는 함수의 추정된 출력 값들을 포함할 수 있다. 도 6에서, 추정된 출력 값들은 y0 - y3을 포함한다. 추정된 출력 값들은 입력 값들에 기초한 함수의 근사화들에 대응할 수 있다.

[0076]

[0084] 연산 동안, 프로세서는 단일 구분적 근사화 명령(601)을 수신할 수 있다. 프로세서는 각각의 입력 값에 대해, 인덱스 값 벡터(630)의 인덱스 값을 생성하기 위해, 입력 벡터(610)의 입력 값들 및 하나 이상의 시프트 값들을 사용하여 단일 구분적 근사화 명령(601)을 실행할 수 있다.

[0077]

[0085] 프로세서(또는 계수 결정 유닛)는 인덱스 값들을 생성하기 위해 입력 벡터(610)의 입력 값들에 하나 이

상의 시프트 값들을 적용할 수 있다. 예로서, 프로세서는 인덱스 값들을 생성하기 위해 입력 값들을 우측 시프트할 수 있다. 인덱스 값들은 도 1을 참조하여 설명된 바와 같이 시프트된 입력 값의 이진 표현과 연관된 특정 수의 상위 비트들을 표현할 수 있다. 특정 구현에서, 입력 벡터(610)의 입력 값들은 프로세서에 의해 수신되기 전에 타겟 범위로 정규화될 수 있다.

[0078] [0086] 프로세서(또는 계수 결정 유닛)은 메모리로부터 입력 계수 테이블(620)을 리트리브(또는 수신)할 수 있고, 프로세서는 입력 계수 테이블(620)을 제3 레지스터(632)에 저장할 수 있다. 특정 구현에서, 프로세서는 입력 계수 테이블(620)의 제1 섹션을 제3 레지스터(632)에 그리고 입력 계수 테이블(620)의 제2 섹션을 다른 레지스터에 저장할 수 있다. 일부 구현들에서, 프로세서는 단일 구분적 근사화 명령(601)을 실행하기 전에 입력 계수 테이블(620)을 제3 레지스터(632)에 로딩할 수 있다. 프로세서는 테이블 로딩 명령을 수신하는 것에 대한 응답으로 제3 레지스터(632)에 입력 계수 테이블(620)을 로딩할 수 있다.

[0079] [0087] 단일 구분적 근사화 명령(601)의 실행은 또한, 각각의 입력 값에 대해, 계수 벡터(640)의 구분적 분석 계수들의 대응하는 세트를 선택하도록 인덱스 값 벡터(630)의 인덱스 값들 및 입력 계수 테이블(620)의 구분적 분석 계수들의 하나 이상의 세트들을 사용하는 것을 포함할 수 있다. 일부 구현들에서, 순열 네트워크는, 인덱스 값들을 사용하여, 입력 계수 테이블(220)의 구분적 분석 계수들의 하나 이상의 세트들을 선택할 수 있고, 제3 레지스터(632)로부터의 구분적 분석 계수들의 하나 이상의 세트들을 프로세서의 산술 로직 회로에 라우팅할 수 있다. 순열 네트워크는 도 1의 순열 네트워크(114)를 포함하거나 그에 대응할 수 있다.

[0080] [0088] 산술 로직 회로는 곱들을 생성하기 위해 입력 벡터(610)의 입력 값들을, 제1 타입 (a)의 구분적 분석 계수들과 곱할 수 있다. 산술 로직 회로는 결과 벡터(650)에 대한 추정된 출력 값들을 생성하기 위해 제2 타입 (b)를 갖는 구분적 분석 계수들 및 곱들($a*x$)에 기초하여 합들을 생성할 수 있다. 예시하자면, 산술 로직 회로는 제1 곱 $a1*x0$ 을 생성하기 위해 구분적 분석 계수들(628)의 제1 그룹의 제1 구분적 분석 계수 $a1$ 과 입력 값들(618)의 제1 그룹의 제1 입력 값 $x0$ 을 곱할 수 있다. 산술 로직 회로는 제1 추정된 출력 값 $y0 = a1*x0+b1$ 을 생성하기 위해 곱 $a1*x0$ 을 구분적 분석 계수들(628)의 제1 그룹의 제2 구분적 분석 계수 $b1$ 과 가산할 수 있다.

[0081] [0089] 다른 예시로서, 산술 로직 회로는 제2 곱 $a1*x3$ 을 생성하기 위해 구분적 분석 계수들의 제2 그룹의 제3 구분적 분석 계수 $a1$ 과 입력 값들의 제2 그룹의 제2 입력 값 $x3$ 을 곱할 수 있다. 산술 로직 회로는 제2 추정된 출력 값 $y3 = a1*x3+b1$ 을 생성하기 위해 제2 곱 $a1*x3$ 을 구분적 분석 계수들의 제2 그룹의 제4 구분적 분석 계수 $b1$ 과 가산할 수 있다. 제3 구분적 분석 계수 $a1$ 은 제1 구분적 분석 계수 $a1$ 과 동일한 값을 가질 수 있지만, 제3 구분적 분석 계수 $a1$ 은 상이한 그룹(예를 들어, 그룹 2)으로부터 선택될 수 있다. $y1$ 및 $y2$ 의 추정된 출력 값들은 유사한 방식으로 생성될 수 있다. 단일 명령의 실행 동안 결과 벡터(650)를 생성함으로써, 시스템은 결과 벡터(650)를 생성하기 위해 다수의 명령들을 실행하는 시스템들에 비해, 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다.

[0082] [0090] 도 7을 참조하면, 단일 구분적 근사화 명령(701)을 실행하는 다른 예시적 프로세스의 도면(700)이 개시되어 있다. 단일 구분적 근사화 명령(701)은 단일 벡터 명령일 수 있다. 도 7에 예시된 바와 같이, 단일 구분적 근사화 명령(701)은 opcode와 같은 명령 명칭(pwlalut4segh)을 포함할 수 있다. 단일 구분적 근사화 명령(701)은 도 6를 참조하여 설명된 바와 같이 하나 이상의 표시자들, 하나 이상의 필드들(예를 들어, 피연산자들) 또는 이들의 조합을 포함할 수 있다. 도 7에 예시된 예에서, 단일 구분적 근사화 명령(701)은 4 세그먼트 구분적 분석에 대응할 수 있다. 그룹들의 수(k)는 세그먼트들의 수, 입력 벡터(710)의 입력 값들의 수(M), 입력 값들의 비트들의 수, 또는 이들의 조합에 대응할 수 있다. 예시적인 비제한적 예로서, k는 16의 값을 가질 수 있다. 다른 예들에서, k의 값은 16 미만 또는 초과일 수 있다. 구분적 분석 계수들의 반복된 세트들(예를 들어, 그룹들)을 가짐으로써, 벡터 연산들은, 도 7에 예시된 바와 같이 입력 값들의 수가 4개의 세그먼트들(a0-a3, b0-b3)과 같은 구분적 근사화의 세그먼트들의 수보다 큰 경우 단일 명령에서(또는 단일 연산으로) 수행될 수 있다. 벡터 연산들은 단일 명령에서 수행될 수 있는데, 이는 구분적 분석 계수들의 세트들의 수가 입력 값들의 수에 매칭할 수 있기 때문이다. 예를 들어, 도 7이 4 세그먼트 연산을 예시하기 때문에, 64개의 입력 값들이 존재하면, 64개의 입력 값들에 매칭하도록 (각각 구분적 분석 계수들의 4개의 세트들의) 구분적 분석 계수들의 16개의 그룹들이 존재할 수 있다. 따라서, 단일 구분적 근사화 명령(701)은 구분적 근사화의 복잡도를 증가시키지 않고 더 큰 입력 벡터들을 수신할 수 있다.

[0083] [0091] 단일 구분적 근사화 명령(701)은 도 1의 프로세서(102)와 같은 프로세서에 의해 실행될 수 있다. 프로세서는 복수의 입력 값들을 포함하는 입력 벡터(710)를 수신할 수 있다. 프로세서는 도 6를 참조하여 설명된 바와 같이 복수의 입력 값들을 프로세싱할 수 있고, 결과 벡터(750)의 다수의 추정된 출력 값들(예를 들어, 콘

텐츠들)을 동시에 생성할 수 있다.

- [0084] [0092] 연산 동안, 프로세서는 단일 구분적 근사화 명령(701)을 수신할 수 있다. 프로세서는 각각의 입력 값에 대해, 인덱스 값 벡터(730)의 인덱스 값을 생성하기 위해, 입력 벡터(710)의 입력 값들 및 하나 이상의 시프트 값들을 사용하여 단일 구분적 근사화 명령(701)을 실행할 수 있다. 프로세서는 입력 계수 테이블(720)을 리트리브 또는 수신할 수 있다. 프로세서는 도 6를 참조하여 설명된 바와 같이 입력 계수 테이블(720)로부터 복수의 계수들을 결정 또는 선택할 수 있다. 도 7는 각각의 그룹의 인덱스 값 벡터(730)에서 어떠한 인덱스 값도 반복(예를 들어, 한번 초과로 등장)되지 않는 예를 예시한다. 다른 예들에서, 특정 인덱스 값은 인덱스 값 벡터(730)의 각각의 그룹에서 한번 초과로 등장할 수 있다. 추가적으로 또는 대안적으로, 가능한 인덱스 값은 인덱스 값 벡터(730)의 하나 이상의 그룹들에서 등장하지 않을 수 있다.
- [0085] [0093] 프로세서는 도 6를 참조하여 설명된 바와 같이 입력 벡터(710)의 입력 값들 및 복수의 계수들을 사용하여 결과 벡터(750)를 생성할 수 있다. 예시하자면, 산술 로직 회로는 제1 곱 $a1*x0$ 을 생성하기 위해 입력 계수 테이블(720)의 구분적 분석 계수들의 제1 그룹의 제1 구분적 분석 계수 $a1$ 과 제1 입력 값 $x0$ 을 곱할 수 있다. 산술 로직 회로는 제1 추정된 출력 값 $y0 = a1*x0+b1$ 을 생성하기 위해 곱 $a1*x0$ 을 입력 계수 테이블의 구분적 분석 계수들(720)의 제2 그룹의 제2 구분적 분석 계수 $b1$ 과 가산할 수 있다. $y1 - yM-1$ 의 추정된 출력 값은 유사한 방식으로 생성될 수 있다. 단일 명령의 실행 동안 결과 벡터(750)를 생성함으로써, 시스템은 결과 벡터(750)를 생성하기 위해 다수의 명령들을 실행하는 시스템들에 비해, 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다.
- [0086] [0094] 도 8을 참조하면, 추정된 값들을 생성하는 방법의 특정한 예시적인 예의 흐름도가 개시되고 포괄적으로 800으로 지정된다. 방법(800)은 도 1의 프로세서(102)와 같은 프로세서에 의해 수행될 수 있다. 방법(800)은 802에서 입력 값들의 벡터를 수신하는 단계를 포함한다. 예를 들어, 입력 값들의 벡터는 도 1의 입력 벡터(110), 도 2 및 도 3의 입력 벡터(210), 도 4 및 도 5의 입력 벡터(410), 도 6의 입력 벡터(610) 또는 도 7의 입력 벡터(710)를 포함하거나 그에 대응할 수 있다. 일부 구현들에서, 입력 값들의 벡터는 함수의 입력 값들에 대응할 수 있다. 예시하자면, 프로세서(102)의 계수 결정 유닛(104)은 도 1를 참조하여 설명된 바와 같이 입력 벡터(110)를 수신할 수 있다.
- [0087] [0095] 방법(800)은 또한 804에서 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하는 단계를 포함한다. 예를 들어, 복수의 구분적 분석 계수들을 선택하는 단일 함수는 도 2의 벡터화된 테이블 툭업 명령(201), 도 1의 벡터화된 테이블 툭업 명령(401), 도 6의 단일 구분적 근사화 명령(601) 또는 도 7의 단일 구분적 선형 근사화 명령을 포함하거나 그에 대응할 수 있다. 복수의 구분적 분석 계수들은 도 1의 복수의 구분적 분석 계수들, 도 2 및 도 3의 계수 벡터(240), 도 4 및 도 5의 계수 벡터(340) 또는 도 6 및 도 7의 복수의 구분적 분석 계수들의 하나 이상의 선택된 세트들을 포함하거나 그에 대응할 수 있다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함할 수 있고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응할 수 있다. 예시하자면, 계수 결정 유닛(104)의 순열 네트워크(114)는 도 1을 참조하여 설명된 바와 같이 복수의 계수들(140)을 선택할 수 있다.
- [0088] [0096] 방법(800)은 806에서, 구분적 분석 계수들의 벡터 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하는 단계를 더 포함한다. 함수의 추정된 출력 값들은 도 1의 추정된 출력 값 벡터(116), 도 3의 결과 벡터(350), 도 5의 결과 벡터(550), 도 6의 결과 벡터(650) 또는 도 7의 결과 벡터(750)를 포함하거나 그에 대응할 수 있다. 예시하자면, 프로세서(102)의 산술 로직 회로(112)는 도 1를 참조하여 설명된 바와 같이 추정된 출력 값 벡터(116)를 결정할 수 있다.
- [0089] [0097] 일부 구현들에서, 방법(800)은 복수의 구분적 분석 계수들을 포함하는 구분적 분석 계수들의 벡터를 생성하는 단계를 포함할 수 있다. 예를 들어, 도 1의 프로세서(102)의 순열 네트워크(114)는 도 2 및 도 3의 계수 벡터(240) 또는 도 4 및 도 5의 계수 벡터(440)를 생성할 수 있다. 특정 구현에서, 구분적 분석 계수들의 벡터를 생성하는 것은 입력 값들의 벡터의 각각의 입력 값에 대해, 입력 값에 기초하여 테이블로부터 구분적 분석 계수들의 세트를 선택하는 단계를 포함할 수 있다. 테이블은 도 1의 입력 계수 테이블(120), 도 2 및 도 3의 입력 계수 테이블(220), 도 4 및 도 5의 입력 계수 테이블(420), 도 6의 입력 계수 테이블(620) 또는 도 7의 입력 계수 테이블(720)을 포함하거나 그에 대응할 수 있다. 테이블은 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 순열 네트워크를 사용함으로써, 방법(800)은, 함수를 근사화하기 위해 사용되는 계수들을 선택하기 위해 레인마다 또는 엘리먼트마다의 툭업 테이블(예를 들어, 각각의 입력 값에 대한 별개의 툭업 테이블)을 사용하는 시스템들에 비해 더 적은 메모리를 사용할 수 있고 증가된 속도를 가질 수 있다.

[0090]

[0098] 일부 구현들에서, 단일 명령을 실행하는 것은 함수의 추정된 출력 값들을 결정하는 것을 더 포함한다. 예를 들어, 도 6의 단일 구분적 근사화 명령(601) 또는 도 7의 단일 구분적 근사화 명령(701)을 실행하는 것은 결과 벡터(650) 및 결과 벡터(750)를 각각 결정할 수 있다. 결과 벡터(650) 및 결과 벡터(750)는 다수의 추정된 출력 값들을 포함할 수 있다. 선형 근사화(예를 들어, 1차 테일러 확장) 및 벡터화된 테이블 루업을 사용함으로써, 단일 명령의 실행은 추정된 출력 값들을 결정할 수 있다. 단일 명령으로 추정된 출력 값들을 결정함으로써, 프로세서는 비선형 근사화, 엘리먼트마다의 루업, 레인마다의 루업 또는 이들의 조합을 사용하는 프로세서보다 더 적은 자원들 및 프로세싱 사이클들을 사용할 수 있다.

[0091]

[0099] 일부 구현들에서, 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하는 것은 입력 값들의 벡터의 제1 입력 값에 대응하는 복수의 비트들에 기초하여 구분적 분석 계수들의 테이블에 대한 루업을 수행하는 것을 포함할 수 있다. 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하는 것은 또한, 루업에 기초하여, 제1 입력 값에 대응하는 구분적 분석 계수들의 제1 세트를 결정하는 것을 포함할 수 있다. 예를 들어, 벡터화된 테이블 루업 명령(201)을 실행하는 것은 도 2를 참조하여 설명된 바와 같이 프로세서가 입력 벡터(210)의 제1 입력 값 x_0 에 대응하는 복수의 비트들(예를 들어, 다중 숫자 비트 값)에 기초하여 입력 계수 테이블(220)에 대한 루업을 수행하는 것 및 제1 입력 값 x_0 에 대응하는 구분적 분석 계수들 a_2 및 b_2 의 제1 세트를 결정하는 것을 포함할 수 있다.

[0092]

[0100] 일부 구현들에서, 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하는 것은 제1 입력 값 및 시프트 값에 기초하여 복수의 비트들을 생성하는 것을 포함할 수 있다. 예를 들어, 복수의 비트들(예를 들어, 비트 값)은 도 2를 참조하여 설명된 바와 같이 제1 입력 값 x_0 및 제1 시프트 값 $shiftvalue_0$ 에 기초하여 생성될 수 있다.

[0093]

[0101] 일부 구현들에서, 방법(800)은 함수에 기초하여 복수의 테이블들로부터 구분적 분석 계수들의 테이블을 선택하는 것을 포함할 수 있다. 예를 들어, 프로세서(102)는 프로세서(102)의 하나 이상의 레지스터들로부터 입력 계수 테이블(120)을 리트리브할 수 있다. 프로세서(102)는 도 1을 참조하여 설명된 바와 같이 입력 벡터(110)를 수신하기 전에 입력 계수 테이블(120)을 수신(또는 리트리브)할 수 있다.

[0094]

[0102] 일부 구현들에서, 구분적 분석 계수들의 다수의 세트들의 각각의 세트는 제1 타입의 적어도 하나의 계수 및 제2 타입의 적어도 하나의 계수를 포함한다. 예를 들어, 도 1을 참조하여, 구분적 분석 계수들의 각각의 세트는 적어도 하나의 "a" 계수 및 적어도 하나의 "b" 계수를 포함한다. 일부 구현들에서, 테이블은 도 2 내지 도 5를 참조하여 설명된 바와 같이, 구분적 분석 계수들의 다수의 세트들을 포함하는 제1 부분 및 하나 이상의 시프트 값들을 포함하는 제2 부분을 포함한다. 다른 구현들에서, 테이블은 도 6 및 도 7를 참조하여 설명된 바와 같이, 제1 타입의 다수의 구분적 분석 계수들을 포함하는 제1 섹션 및 제2 타입의 다수의 구분적 분석 계수들을 포함하는 제2 섹션을 포함한다.

[0095]

[0103] 일부 구현들에서, 테이블은 구분적 분석 계수들의 다수의 그룹들을 포함할 수 있다. 예를 들어, 도 6의 입력 계수 테이블(620) 및 도 7의 입력 계수 테이블(720)은 그룹 1 및 그룹 2와 같은 다수의 그룹들을 포함할 수 있다. 구분적 분석 계수들의 다수의 그룹들 중 적어도 2개의 그룹들의 구분적 분석 계수들은 동일할 수 있다. 예를 들어, 도 6의 입력 계수 테이블(620)의 제1 섹션의 그룹 1 및 그룹 2 각각은 a_0 및 a_1 을 포함할 수 있다.

[0096]

[0104] 일부 구현들에서, 복수의 구분적 분석 계수들은 구분적 분석 계수들의 다수의 세트들을 포함할 수 있고, 구분적 분석 계수들의 다수의 세트들 중 구분적 분석 계수들의 각각의 세트는 2개의 구분적 분석 계수들을 포함할 수 있다. 예를 들어, 도 2의 계수 벡터(240)는 순열 네트워크에 의해 선택된 제1 세트 a_0 및 b_0 및 제2 세트 a_1 및 b_1 과 같은 구분적 분석 계수들의 다수의 세트들을 포함할 수 있다. 특정 구현에서, 구분적 분석 계수들의 다수의 세트들의 각각의 세트는 제1 타입의 적어도 하나의 계수 및 제2 타입의 적어도 하나의 계수를 포함할 수 있다. 예를 들어, 제1 타입은 "a" 타입 계수들을 포함하거나 그에 대응할 수 있고, 제2 타입은 "b" 타입 계수들을 포함하거나 그에 대응할 수 있다. 일부 구현들에서, 복수의 구분적 분석 계수들은 선형 분석 계수들에 대응할 수 있다.

[0097]

[0105] 일부 구현들에서, 함수는 비선형 함수를 포함할 수 있다. 예를 들어, 함수는 도 1을 참조하여 설명된 바와 같이 역함수, 제곱근 함수, 지수 함수, 로그 함수, 아크 탄젠트 함수 또는 이들의 조합을 포함할 수 있다.

[0098]

[0106] 일부 구현들에서, 추정된 출력 값들은 제2 단일 명령을 실행함으로써 결정될 수 있다. 예를 들어, 제2 단일 명령은 도 3의 벡터 산술 명령(301) 또는 도 5의 벡터 산술 명령(501)을 포함하거나 그에 대응할 수 있다.

제2 단일 명령을 실행하는 것은 복수의 구분적 분석 계수들의 제1 구분적 분석 계수와 입력 값들의 벡터의 제1 입력 값의 곱을 생성하는 것을 포함할 수 있다. 제2 단일 명령을 실행하는 것은 또한 곱과 복수의 구분적 분석 계수들의 제2 구분적 분석 계수의 합에 기초하여 제1 추정된 출력 값을 생성하는 것을 포함할 수 있다.

- [0099] [0107] 일부 구현들에서, 방법(800)은 입력 값들의 벡터 및 타겟 범위에 기초하여 시프트 값을 계산하는 단계를 포함할 수 있다. 예를 들어, 도 1의 프로세서 또는 도 9의 시프트 값 생성기(990)는 시프트 값을 계산할 수 있다. 방법(800)은 또한 입력 값들의 벡터의 각각의 입력 값에 대해, 입력 값 및 시프트 값에 기초하여 복수의 비트들을 생성하는 것을 포함할 수 있다. 복수의 비트들의 상위 비트들의 수는 타겟 범위 이내일 수 있다. 다른 구현들에서, 단일 비트가 생성될 수 있다.
- [0100] [0108] 따라서, 방법(800)은 프로세서가 감소된 수의 명령들을 사용하여 함수의 추정된 출력 값을 생성하게 할 수 있다. 예를 들어, 프로세서는 추정된 출력 값을 생성하기 위해 하나 또는 두개의 명령들을 실행할 수 있다. 감소된 수의 명령들은 더 적은 프로세싱 자원들을 사용할 수 있다. 더 적은 프로세싱 자원들을 사용함으로써, 방법(800)은 더 적은 메모리 및 전력을 사용하고 증가된 속도를 가질 수 있다.
- [0101] [0109] 특정 양상들에서, 도 8의 방법(800)은 FPGA(field-programmable gate array) 디바이스(예를 들어, ASIC, DSP, 제어기, FPGA 디바이스 등), 소프트웨어(예를 들어, 프로세서 등에 의해 실행가능한 명령들), 또는 이들의 임의의 조합에 의해 구현될 수 있다. 예로서, 도 8의 방법(800)은 도 9에 대해 설명된 바와 같이 명령들을 실행하는 프로세서에 의해 수행될 수 있다. 예시하자면, 도 8의 방법(800)의 일부분은 도 8의 방법(800)의 제2 부분과 조합될 수 있다. 추가적으로, 도 8을 참조하여 설명된 하나 이상의 동작들은 선택적일 수 있거나, 적어도 부분적으로 동시에 수행될 수 있거나, 도시되거나 설명된 것과 상이한 순서로 수행될 수 있거나, 또는 이들의 조합일 수 있다.
- [0102] [0110] 도 9을 참조하면, 디바이스(예를 들어, 무선 통신 디바이스)의 특정한 예시적 구현의 블록도가 도시되고 포괄적으로 900으로 지정된다. 다양한 구현들에서, 디바이스(900)는 도 9에 예시된 것보다 많거나 적은 컴포넌트들을 가질 수 있다. 예시적인 예에서, 디바이스(900)는 도 8의 방법(800)에 따라 동작할 수 있다.
- [0103] [0111] 특정 구현에서, 디바이스(900)는 프로세서(906)(예를 들어, CPU) 및 메모리(932)를 포함한다. 디바이스(900)는 프로세서(910)(예를 들어, DSP)와 같은 하나 이상의 추가적인 프로세서들을 포함할 수 있다. 프로세서(906), 프로세서(910) 또는 둘 모두는 도 1의 프로세서(102)를 포함하거나 그에 대응할 수 있고, 메모리(932)는 도 1의 메모리(106)를 포함하거나 그에 대응할 수 있다. 프로세서(910)는 레지스터들(980) 및 시프트 값 생성기(990)를 포함할 수 있다. 레지스터들(980)은 도 1 내지 도 7을 참조하여 설명된 바와 같이 레지스터들을 포함하거나 그에 대응할 수 있다. 시프트 값 생성기(990)는 하나 이상의 시프트 값을 생성(또는 계산)하도록 구성될 수 있다. 시프트 값 생성기(990)는 입력 값들의 범위 및 입력 값들의 타겟 범위에 기초하여 시프트 값을 생성(또는 계산)할 수 있다. 하나 이상의 시프트 값을 도 1 및 도 2를 참조하여 설명된 바와 같이 인덱스 값을 생성하기 위해 입력 값들에 적용될 수 있다. 추가적으로 또는 대안적으로, 하나 이상의 시프트 값을 입력 값을 타겟 범위로 정규화하기 위해 사용될 수 있다.
- [0104] [0112] 프로세서(910)는 도 1의 계수 결정 유닛(104), 산술 로직 회로(112) 및 순열 네트워크(114)를 포함할 수 있다. 레지스터들(980)은 프로세서(910)의 컴포넌트로서 예시되어 있지만, 다른 예들에서 레지스터들(980)의 하나 이상의 컴포넌트들은 프로세서(906), CODEC(coder-decoder)(934), 다른 프로세싱 컴포넌트 또는 이들의 조합에 포함될 수 있다. 추가적으로, 도 1의 계수 결정 유닛(104), 산술 로직 회로(112) 및 순열 네트워크(114)는 프로세서(906), CODEC(934), 다른 프로세싱 컴포넌트 또는 이들의 조합에 포함될 수 있다.
- [0105] [0113] 프로세서(910)는 스피치 및 뮤직 CODEC(908) 및 간접 제거기(912)를 포함할 수 있다. 스피치 및 뮤직 CODEC(908)은 보코더 인코더와 같은 인코더(992)를 포함할 수 있다. 스피치 및 뮤직 CODEC(908)은 프로세서(910)의 컴포넌트로서 예시되어 있지만, 다른 예들에서 스피치 및 뮤직 CODEC(908)의 하나 이상의 컴포넌트들은 프로세서(906), CODEC(934), 다른 프로세싱 컴포넌트 또는 이들의 조합에 포함될 수 있다.
- [0106] [0114] 디바이스(900)는 메모리(932) 및 CODEC(934)을 포함할 수 있다. CODEC(934)은 DAC(digital-to-analog converter)(902) 및 ADC(analog-to-digital converter)(904)를 포함할 수 있다. 스피커(936), 마이크로폰(938) 또는 둘 모두는 CODEC(934)에 커플링될 수 있다. CODEC(934)은 마이크로폰(938)으로부터 아날로그 신호들을 수신하고, 아날로그-디지털 변환기(904)를 사용하여 아날로그 신호들을 디지털 신호들로 변환하고, 디지털 신호들을 스피치 및 뮤직 CODEC(908)에 제공할 수 있다. 스피치 및 뮤직 CODEC(908)은 디지털 신호들을 프로세싱할 수 있다. 일부 구현들에서, 스피치 및 뮤직 CODEC(908)은 CODEC(934)에 디지털 신호들을 제공할 수 있다.

CODEC(934)은 디지털-아날로그 변환기(902)를 사용하여 디지털 신호들을 아날로그 신호들로 변환할 수 있고, 아날로그 신호들을 스피커(936)에 제공할 수 있다.

- [0107] [0115] 디바이스(900)는 트랜시버(950)(예를 들어, 송신기, 수신기 또는 이들의 조합)를 통해 안테나(942)에 커플링된 무선 제어기(940)를 포함할 수 있다. 디바이스(900)는 컴퓨터 판독가능 매체 또는 컴퓨터 판독가능 저장 디바이스와 같은 메모리(932)를 포함할 수 있다. 메모리(932)는 도 8의 방법(800)을 수행하기 위해 프로세서(906), 프로세서(910) 또는 이들의 조합에 의해 실행가능한 하나 이상의 명령들과 같은 명령들(960)을 포함할 수 있다.
- [0108] [0116] 예시적인 예로서, 메모리(932)는 프로세서(906), 프로세서(910) 또는 이들의 조합에 의해 실행되는 경우 프로세서(906), 프로세서(910) 또는 이들의 조합으로 하여금, 입력 값들의 벡터에 기초하여 복수의 구분적 분석 계수들을 선택하게 하는 적어도 단일 명령을 저장할 수 있다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함할 수 있고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응할 수 있다. 적어도 단일 명령은 실행되는 경우 추가로, 프로세서(906), 프로세서(910) 또는 이들의 조합으로 하여금, 선택된 복수의 구분적 분석 계수들에 기초하여 구분적 분석 계수들의 벡터를 생성하게 할 수 있다. 예를 들어, 프로세서(906), 프로세서(910) 또는 이들의 조합은 순열 네트워크로 하여금, 구분적 분석 계수들의 벡터를 생성하기 위해 선택된 복수의 구분적 분석 계수들을 하나 이상의 레지스터들에 라우팅하게 할 수 있다.
- [0109] [0117] 일부 구현들에서, 적어도 단일 명령은 실행되는 경우 추가로, 프로세서(906), 프로세서(910) 또는 이들의 조합으로 하여금, 구분적 분석 계수들의 벡터 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값을 결정하게 할 수 있다. 다른 구현들에서, 적어도 단일 명령은 실행되는 경우 추가로, 프로세서(906), 프로세서(910) 또는 이들의 조합으로 하여금, 구분적 분석 계수들의 벡터 및 입력 값들의 벡터에 기초하여 구분적 분석 계수들의 벡터를 생성하게 할 수 있다. 예를 들어, 프로세서(906), 프로세서(910) 또는 이들의 조합은 순열 네트워크로 하여금, 구분적 분석 계수들의 벡터를 생성하기 위해 선택된 복수의 구분적 분석 계수들을 하나 이상의 레지스터들에 라우팅하게 할 수 있다.
- [0110] [0118] 일부 구현들에서, 적어도 단일 명령은 실행되는 경우 추가로, 프로세서로 하여금, 입력 값들의 벡터의 제1 입력 값 및 시프트 값에 기초하여 복수의 비트들을 생성하게 하고, 복수의 비트들에 기초하여 계수 테이블에 대한 투입을 수행하게 하고, 투입에 기초하여 제1 입력 값에 대응하는 구분적 분석 계수들의 제1 세트를 결정하게 할 수 있다.
- [0111] [0119] 일부 구현들에서, 메모리(932)는 프로세서(906), 프로세서(910) 또는 둘 모두로 하여금 도 1의 프로세서(102)를 참조하여 설명된 바와 같은 기능들을 수행하게 하거나, 도 8의 방법(800)의 적어도 일부를 수행하게 하거나, 이들의 조합을 수행하게 하도록 프로세서(906), 프로세서(910) 또는 둘 모두에 의해 실행될 수 있는 코드(예를 들어, 해석되거나 컴파일된 프로그램 명령들)를 포함할 수 있다. 추가로 예시하자면, 예 1 내지 예 3은 컴파일되고 메모리(932)에 저장될 수 있는 예시적인 명령들 및 예시적인 의사-코드(예를 들어, 단순화된 C-코드)를 도시한다. 의사-코드는 도 1 내지 도 8에 대해 설명된 양상들의 가능한 구현을 예시한다. 의사-코드는 실행가능한 코드의 일부가 아닌 코멘트들을 포함한다. 의사-코드에서, 코멘트의 시작은 사선(forward slash) 및 별표(예를 들어, "/")로 표시되고, 코멘트의 종료는 별표 및 사선(예를 들어, "/*")으로 표시된다. 예시하자면, 코멘트 "COMMENT"는 /* COMMENT */로서 의사-코드에 등장할 수 있다.
- [0112] [0120] 제공된 예들에서, "==" 연산자는 동일성 비교를 표시하여, "A==B"는 A의 값이 B의 값과 동일한 경우 참의 값을 갖고, 그렇지 않으면 거짓의 값을 갖는다. "&&" 연산자는 로직 AND 연산을 표시한다. "||" 연산자는 로직 OR 연산을 표시한다. ">"(초과) 연산자는 "보다 큼"을 표현하고, ">=" 연산자는 "보다 크거나 그와 동일"을 표현하고, "<" 연산자는 "보다 작음"을 표시한다.
- [0113] [0121] 제공된 예에서, "*"는 곱셈 연산을 표현할 수 있고, "+" 또는 "합"은 가산 연산을 표현할 수 있고, "-"는 감산 연산을 표시할 수 있고, "/"는 나눗셈 연산을 표현할 수 있다. "=" 연산자는 할당을 표현한다(예를 들어, "a=1"은 1의 값을 변수 "a"에 할당한다). ">>"는 우측 시프트 연산을 표현할 수 있다. 다른 구현들은 예 1 내지 예 3의 조건들의 세트에 추가로 또는 그 대신에 하나 이상의 조건들을 포함할 수 있다.
- [0114] 예 1
- [0115] Vdd=v1ut32segh(Vuu,Vv)
- [0116] /* Vv는 64개의 16들의 값 x63, ..., x3, x2, x1, x0을 포함하고, Vuu.H는 32x (b,a) 세그먼트들을 포함하고, Vuu.L은 스플랫된 16들의 시프트 값을 포함한다 */

[0116] for (i = 0; i < 1024/32; i++) {

Vdd.V16s[2*i] = Vuu.V16s[64+Vv.V16s[2*i]>>Vuu.V16s[2*i]] modulo
2^5];

Vdd.V16s[2*i+1] = Vuu.V16s[64+Vv.V16s[2*i]>>Vuu.V16s[2*i]] modulo
2^5 +1];

Vdd.V16s[64+2*i] = Vuu.V16s[64+Vv.V16s[2*i+1]>>Vuu.V16s[2*i+1]]
modulo 2^5];

[0117] Vdd.V16s[64+2*i+1] =

Vuu.V16s[64+Vv.V16s[2*i+1]>>Vuu.V16s[2*i+1]] modulo 2^5 +1];

[0118] };

[0119] 예 2

[0120] Vd=vmpyaddeoh(Vdd,Vv)

[0121] /* Vv는 64개의 16들의 값 x63, , x3, x2, x1, x0을 포함하고, Vdd.L은 짝수 엘리먼트들에 대한
(b,a) 쌍들을 포함하고, Vdd.H는 홀수 엘리먼트들에 대한 (b,a) 쌍들을 포함한다 */

[0122] for (i = 0; i < 1024/32; i++) {

Vd.V16s[2*i]=sat_16(Vdd.V16s[2*i+1]+(((Vdd.V16s[2*i] *

Vv.V16s[2*i])+0x4000)>>15));

Vd.V16s[2*i+1]=sat_16(Vdd.V16s[64+2*i+1]+(((Vdd.V16s[64+2*i] *

Vv.V16s[2*i+1])+0x4000)>>15));

[0123] };

[0124] 예 3

[0125] Vd=pwlalut4segh(Vuu,Vv)

[0126] /* Vu는 x3, x2, x1, x0을 포함하고, Vu.L은 a3, a2, a1, a0을 포함하고, Vu.H는 b3, b2, b1, b0을 포함한다 */

[0127] for (i = 0; i < 1024/16; i++) {

Vd.V16s[i]=sat_16(Vuu.V16s[Vv.V16s[i]>>12&3+((64+i)>>2)<<2]+(((V
uu.V16s[Vv.V16s[i]>>12&3+(i>>2)<<2] * Vv.V16s[i])+0x4000)>>15));

[0128] };

[0129] [0122] 메모리(932)는 도 8의 방법(800)과 같이 본원에 개시된 방법들 및 프로세스들을 수행하기 위해, 프로세

서(906), 프로세서(910), CODEC(834), 디바이스(900)의 다른 프로세싱 유닛, 또는 이들의 조합에 의해 실행가능한 명령들(960)을 포함할 수 있다. 도 1의 시스템(100)의 하나 이상의 컴포넌트들은 전용 하드웨어(예를 들어, 회로)를 통해, 하나 이상의 작업들을 수행하기 위해 명령들(예를 들어, 명령들(960))을 실행하는 프로세서에 의해, 또는 이들의 조합으로 구현될 수 있다. 예로서, 프로세서(906), 프로세서(910), CODEC(934) 또는 이들의 조합의 메모리(932) 또는 하나 이상의 컴포넌트들은 메모리 디바이스, 예를 들어, RAM(random access memory), MRAM(magnetoresistive random access memory), STT-MRAM(spin-torque transfer MRAM), 플래시 메모리, ROM(read-only memory), PROM(programmable read-only memory), EPROM(erasable programmable read-only memory), EEPROM(electrically erasable programmable read-only memory), 레지스터들, 하드 디스크, 착탈식 디스크 또는 CD-ROM(compact disc read-only memory)일 수 있다. 메모리 디바이스는, 컴퓨터(예를 들어, CODEC(934)의 프로세서, 프로세서(906), 프로세서(910) 또는 이들의 조합)에 의해 실행되는 경우 컴퓨터로 하여금 도 8의 방법(800)의 적어도 일부를 수행하게 할 수 있는 명령들(예를 들어, 명령들(960))을 포함할 수 있다. 예로서, 프로세서(906), 프로세서(910), CODEC(934)의 메모리(932) 또는 하나 이상의 컴포넌트들은 컴퓨터(예를 들어, CODEC(934)의 프로세서, 프로세서(906), 프로세서(910) 또는 이들의 조합)에 의해 실행되는 경우 컴퓨터로 하여금 도 8의 방법(800)의 적어도 일부를 수행하는 명령들(예를 들어, 명령들(960))을 포함하는 비일시적 컴퓨터 판독가능 매체일 수 있다.

[0128] [0123] 특정 구현에서, 디바이스(900)는 시스템-인-패키지 또는 시스템-온-칩 디바이스(922)에 포함될 수 있다. 일부 구현들에서, 메모리(932), 프로세서(906), 프로세서(910), 디스플레이 제어기(926), CODEC(934), 무선 제어기(940) 및 트랜시버(950)는 시스템-인-패키지 또는 시스템-온-칩 디바이스(922)에 포함된다. 일부 구현들에서, 입력 디바이스(930) 및 전원(944)은, 시스템-온-칩 디바이스(922)에 커플링된다. 또한, 특정한 구현에서, 도 9에 예시된 바와 같이, 디스플레이(928), 입력 디바이스(930), 스피커(936), 마이크로폰(938), 안테나(942), 및 전원(944)은 시스템-온-칩 디바이스(922) 외부에 있다. 다른 구현들에서, 디스플레이(928), 입력 디바이스(930), 스피커(936), 마이크로폰(938), 안테나(942), 및 전원(944) 각각은, 시스템-온-칩 디바이스(922)의 인터페이스 또는 제어기와 같은 시스템-온-칩 디바이스(922)의 컴포넌트에 커플링될 수 있다. 예시적 예에서, 디바이스(900)는 통신 디바이스, 모바일 통신 디바이스, 스마트폰, 셀룰러 폰, 랩톱 컴퓨터, 컴퓨터, 태블릿 컴퓨터, 개인 휴대 정보 단말, 셋탑 박스, 디스플레이 디바이스, 텔레비전, 게임 콘솔, 뮤직 플레이어, 라디오, 디지털 비디오 플레이어, DVD(digital video disc) 플레이어, 광 디스크 플레이어, 튜너, 카메라, 내비게이션 디바이스, 디코더 시스템, 인코더 시스템 또는 이들의 임의의 조합에 대응한다.

[0129] [0124] 예시적인 예에서, 프로세서(910)는 도 1 내지 도 8을 참조하여 설명된 방법들 또는 동작들의 전부 또는 일부를 수행하도록 동작가능할 수 있다. 예를 들어, 프로세서(910)는 입력 값들의 벡터를 수신할 수 있다. 입력 값들은 안테나(942)에 의해 수신된 신호의 샘플들을 포함하거나 그에 대응할 수 있다. 계수 결정 유닛(104)은 입력 계수 테이블을 리트리브할 수 있다. 프로세서(910)는 입력 계수 테이블로부터 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행할 수 있다. 일부 구현들에서, 순열 네트워크(114)는 복수의 구분적 분석 계수들을 포함하는 계수 벡터를 생성할 수 있다. 산술 로직 회로(112)는 추정된 출력 값들을 포함하는 결과 벡터를 생성할 수 있다.

[0130] [0125] 설명된 양상들과 관련하여, 장치는 입력 값들의 벡터를 수신하기 위한 수단을 포함할 수 있다. 예를 들어, 입력 값들의 벡터를 수신하기 위한 수단은, 도 1의 프로세서(102), 메모리(106), 계수 결정 유닛(104), 순열 네트워크(114), 산술 로직 회로(112), 도 2의 제1 레지스터(212), 도 4의 제1 레지스터(412), 도 6의 제1 레지스터(612), 도 7의 제1 레지스터(712), 도 9의 트랜시버(950), 무선 제어기(940), 레지스터들(980), 명령들(960)을 실행하도록 프로그래밍된 프로세서들(906, 910) 중 하나 이상, 입력 값들의 벡터를 수신하기 위한 하나 이상의 다른 구조들, 디바이스들, 회로들, 모듈들 또는 명령들, 또는 이들의 조합을 포함하거나 그에 대응할 수 있다.

[0131] [0126] 장치는 또한 복수의 구분적 분석 계수들을 선택하기 위해 단일 명령을 실행하기 위한 수단을 포함할 수 있다. 예를 들어, 실행하기 위한 수단은, 도 1의 프로세서(102), 계수 결정 유닛(104), 순열 네트워크(114), 도 9의 명령들(960)을 실행하도록 프로그래밍된 프로세서(910, 906) 중 하나 이상, 복수의 구분적 분석 계수들을 선택하도록 단일 명령들을 실행하기 위한 하나 이상의 다른 구조들, 디바이스들, 회로들, 모듈들 또는 명령들, 또는 이들의 조합을 포함하거나 그에 대응할 수 있다. 복수의 구분적 분석 계수들은 구분적 분석 계수들의 하나 이상의 세트들을 포함할 수 있고, 구분적 분석 계수들의 각각의 세트는 입력 값들의 벡터의 입력 값에 대응할 수 있다.

[0132] [0127] 장치는 복수의 구분적 분석 계수들 및 입력 값들의 벡터에 기초하여 함수의 추정된 출력 값들을 결정하

기 위한 수단을 더 포함할 수 있다. 예를 들어, 결정하기 위한 수단은, 도 1의 프로세서(102), 산술 로직 회로(112), 도 9의 명령들(960)을 실행하도록 프로그래밍된 프로세서(910, 906) 중 하나 이상, 함수의 추정된 출력값들을 결정하기 위한 하나 이상의 다른 구조들, 디바이스들, 회로들, 모듈들 또는 명령들, 또는 이들의 조합을 포함하거나 그에 대응할 수 있다.

[0133] [0128] 장치는 또한 구분적 분석 계수들의 벡터를 저장하기 위한 수단을 포함할 수 있다. 예를 들어, 구분적 분석 계수들의 벡터를 저장하기 위한 수단은, 도 1의 프로세서(102), 메모리(106), 계수 결정 유닛(104), 순열 네트워크(114), 산술 로직 회로(112), 도 2의 제4 레지스터(242), 도 4의 제4 레지스터(442), 도 9의 레지스터들(980), 명령들(960)을 실행하도록 프로그래밍된 프로세서(910, 906) 중 하나 이상, 구분적 분석 계수들의 벡터를 저장하기 위한 하나 이상의 다른 구조들, 디바이스들, 회로들, 모듈들 또는 명령들, 또는 이들의 조합을 포함하거나 그에 대응할 수 있다. 구분적 분석 계수들의 벡터는 복수의 구분적 분석 계수들을 포함할 수 있다.

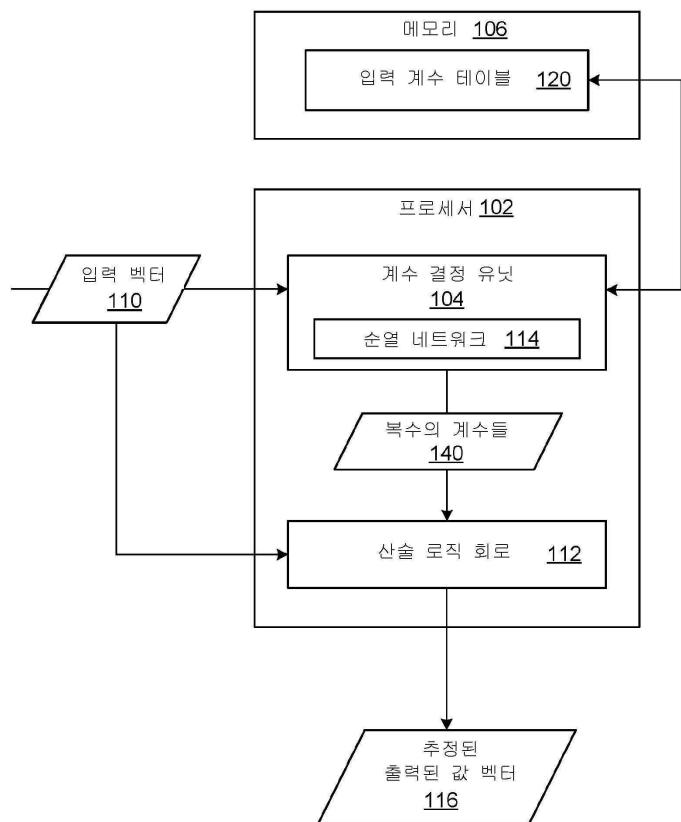
[0134] [0129] 일부 구현들에서, 수신하기 위한 수단, 실행하기 위한 수단, 및 결정하기 위한 수단은 모바일 폰, 셀룰러 폰, 컴퓨터, 휴대용 컴퓨터, 튜너, 라디오, 위성 라디오, 통신 디바이스, 모뎀, 휴대용 뮤직 플레이어, 휴대용 디지털 비디오 플레이어, 내비게이션 디바이스, PDA(personal digital assistant), 모바일 위치 데이터 유닛 또는 이들의 조합에 통합된다.

[0135] [0130] 앞서 설명된 설명의 양상들에서, 수행되는 다양한 기능들은 도 1의 시스템(100)의 회로 또는 컴포넌트들, 도 9의 디바이스(900) 또는 이들의 조합과 같은 특정 회로 또는 컴포넌트들에 의해 수행되는 것으로 설명되었다. 그러나, 회로 및 컴포넌트들의 이러한 분할은 오직 예시를 위한 것이다. 대안적인 예들에서, 특정 회로 또는 컴포넌트에 의해 수행되는 기능은 그 대신 다수의 회로들 또는 컴포넌트들 사이에 분할될 수 있다. 또한, 다른 대안적인 예들에서, 도 1 내지 도 7 및 도 9의 둘 이상의 회로들 또는 컴포넌트들은 단일 회로 또는 컴포넌트로 통합될 수 있다. 도 1 내지 도 7 및 도 9에 예시된 각각의 회로 및 컴포넌트는 하드웨어(예를 들어, ASIC, DSP, 제어기, FPGA 디바이스 등), 소프트웨어(예를 들어, 로직, 모듈들, 프로세서에 의해 실행 가능한 명령들 등) 또는 이들의 조합을 사용하여 구현될 수 있다.

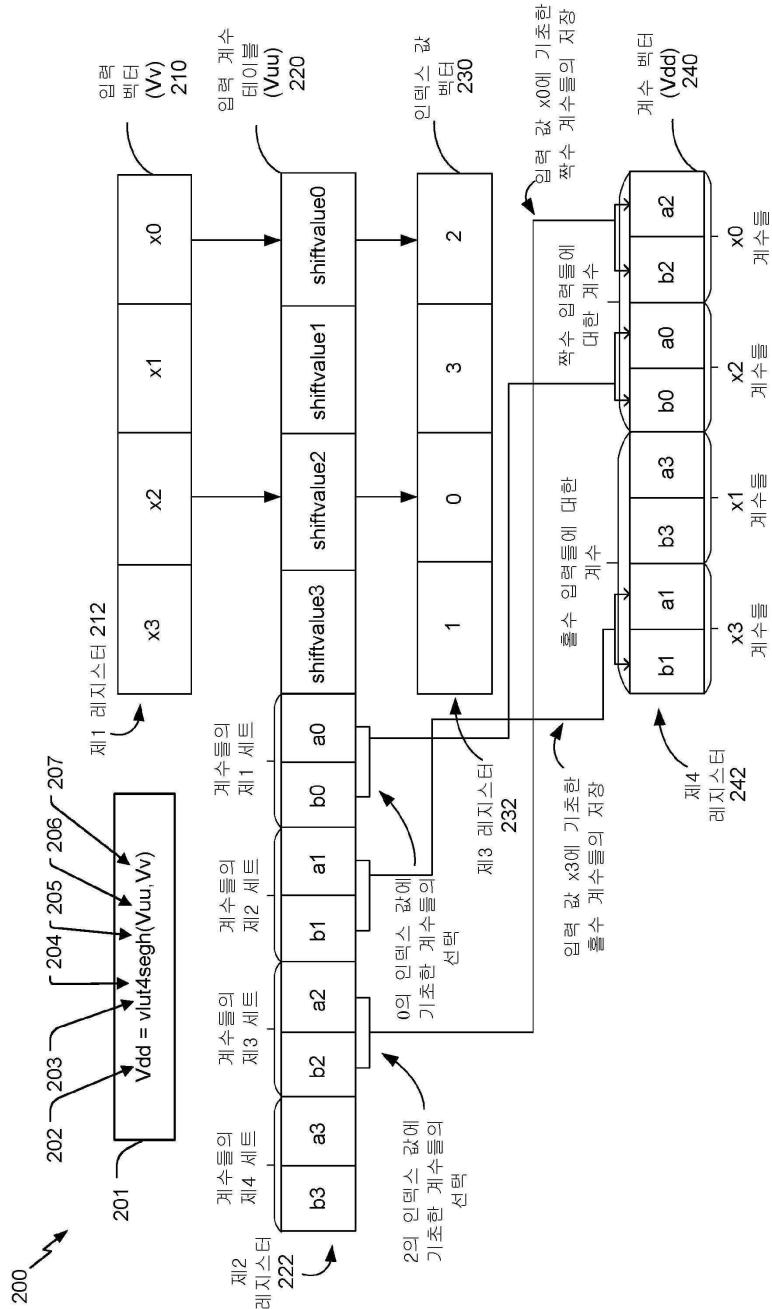
[0136] [0131] 당업자들은, 본 명세서에 개시된 양상들과 관련하여 설명된 다양한 예시적인 로직 블록들, 구성들, 모듈들, 회로들, 및 알고리즘 단계들이 전자 하드웨어, 프로세서에 의해 실행되는 컴퓨터 소프트웨어, 또는 둘 모두의 결합들로서 구현될 수 있음을 추가적으로 인식할 것이다. 다양한 예시적인 컴포넌트들, 블록들, 구성들, 모듈들, 회로들, 및 단계들은 이들의 기능 관점들에서 일반적으로 상술되었다. 이러한 기능이 하드웨어로 구현되는지 또는 프로세서 실행가능 명령들로 구현되는지 여부는 특정 애플리케이션, 및 전체 시스템에 부과된 설계 제약들에 의존한다. 당업자들은 설명된 기능을 각각의 특정 애플리케이션에 대해 다양한 방식들로 구현할 수 있고, 이러한 구현 결정들이 본 개시의 범주를 벗어나는 것으로 해석되어서는 안 된다.

[0137] [0132] 본 명세서에 개시된 양상들과 관련하여 설명되는 방법 또는 알고리즘의 단계들은 직접 하드웨어로, 프로세서에 의해 실행되는 소프트웨어 모듈로, 또는 이 둘의 결합으로 포함될 수 있다. 소프트웨어 모듈은 RAM, 플래시 메모리, ROM, PROM, EPROM, EEPROM, 레지스터들, 하드 디스크, 착탈식 디스크, CD-ROM, 또는 업계에 공지된 임의의 다른 형태의 비일시적 저장 매체에 상주할 수 있다. 특정 저장 매체는, 프로세서가 저장 매체로부터 정보를 판독하고, 저장 매체에 정보를 기록할 수 있도록 프로세서에 커플링될 수 있다. 대안적으로, 저장 매체는 프로세서에 통합될 수 있다. 프로세서 및 저장 매체는 ASIC에 상주할 수 있다. ASIC은 컴퓨팅 디바이스 또는 사용자 단말에 상주할 수 있다. 대안적으로, 프로세서 및 저장 매체는 컴퓨팅 디바이스 또는 사용자 단말에서 개별 컴포넌트들로서 상주할 수 있다.

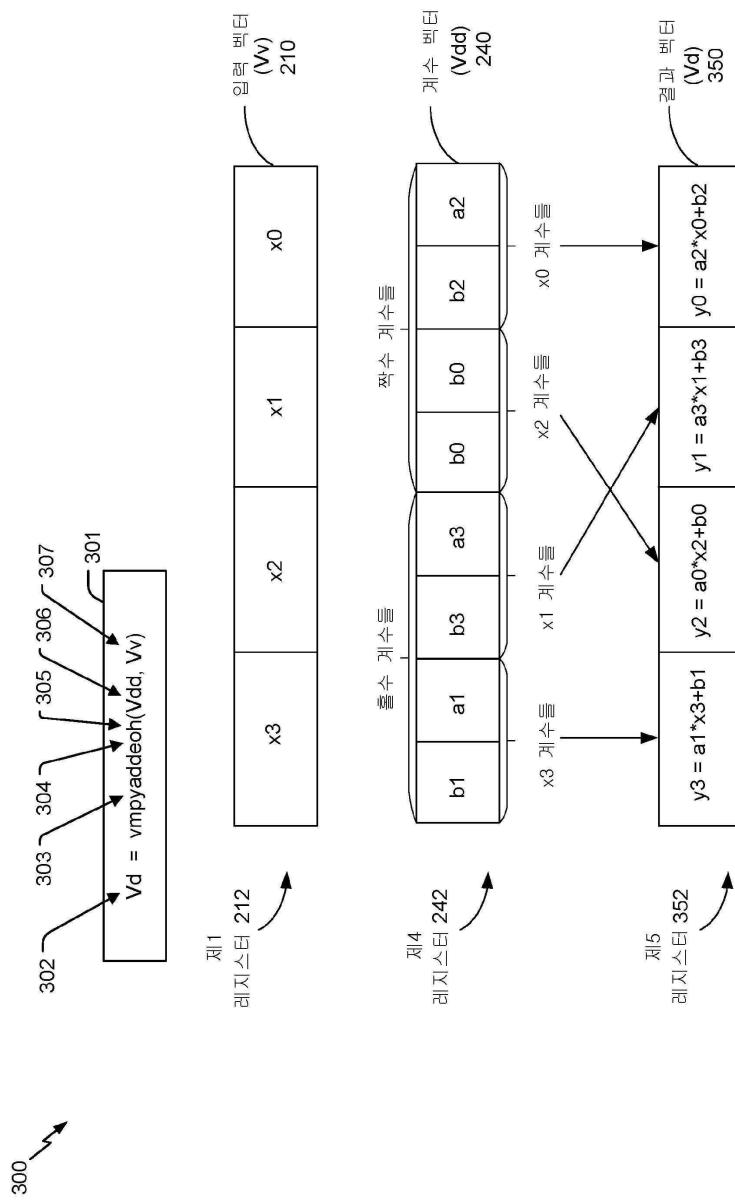
[0138] [0133] 이전의 설명은 당업자가 개시된 양상들을 실시하거나 또는 사용할 수 있도록 제공된다. 이러한 양상들에 대한 다양한 변형들은 당업자들에게 용이하게 명백할 것이며, 본 명세서에 정의된 원리들은 본 개시의 범위를 벗어나지 않으면서 다른 양상들에 적용될 수 있다. 따라서, 본 개시는 본 명세서에 나타낸 양상들로 제한되도록 의도되지 않으며, 다음의 청구항들에 의해 정의된 바와 같은 원리들 및 신규한 특성들과 일치하는 가능한 가장 넓은 범위에 부합할 것이다.

도면**도면1**100
↓

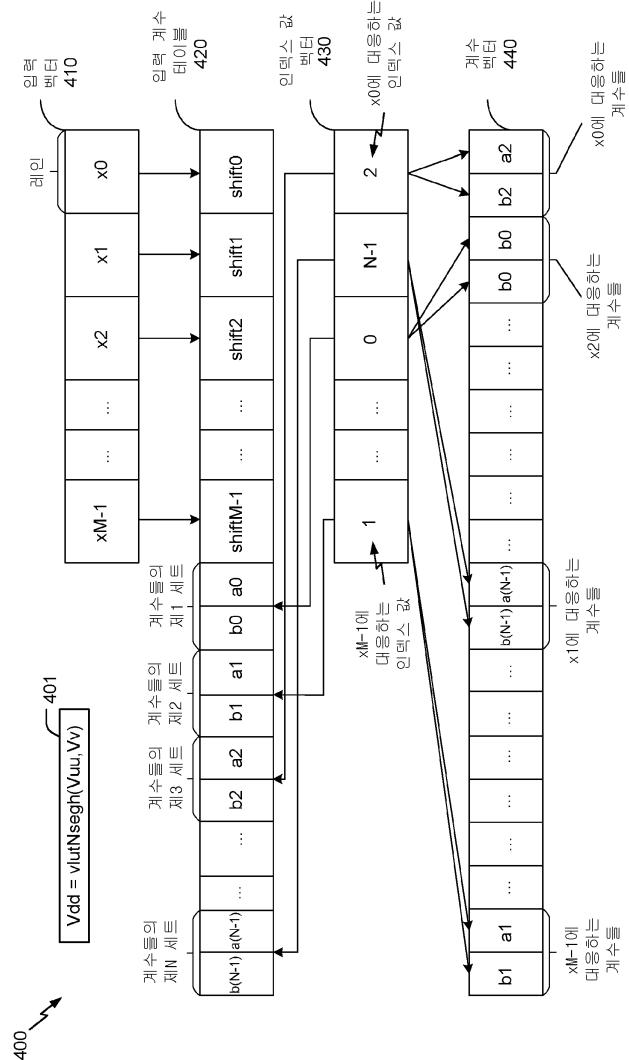
도면2



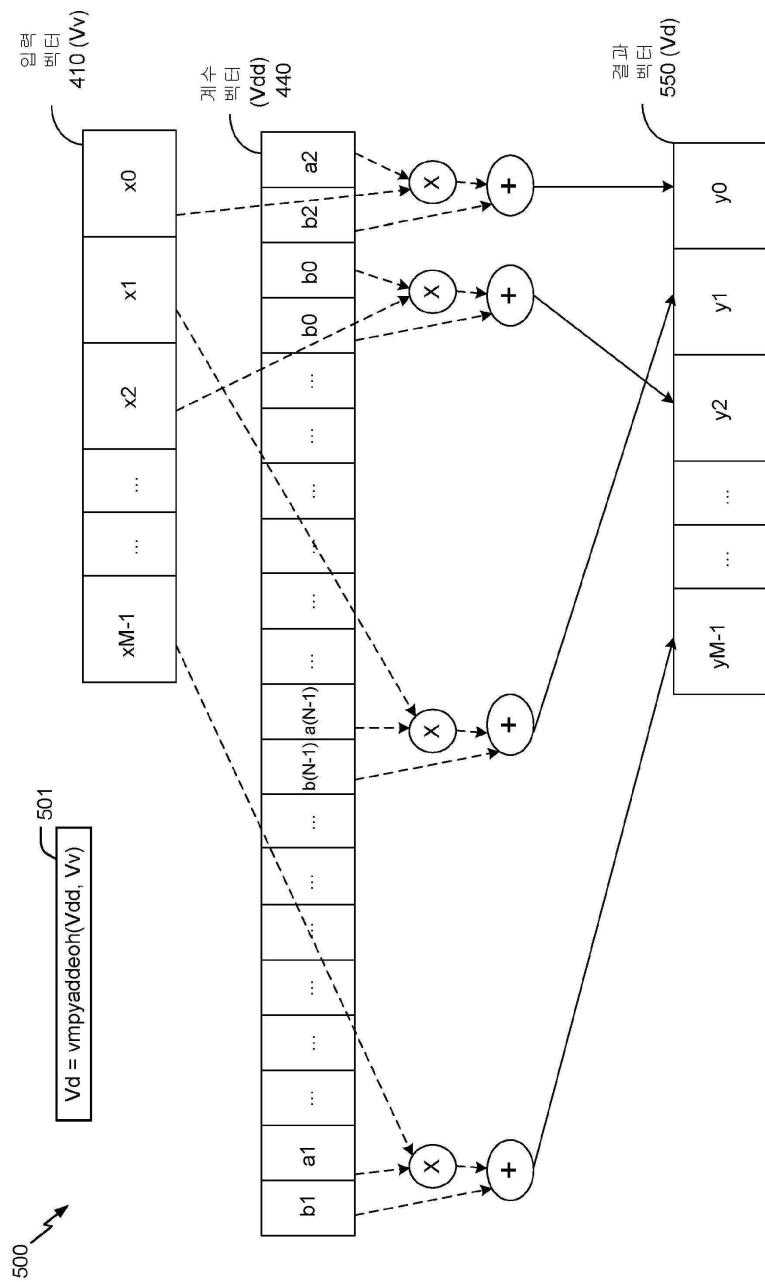
도면3



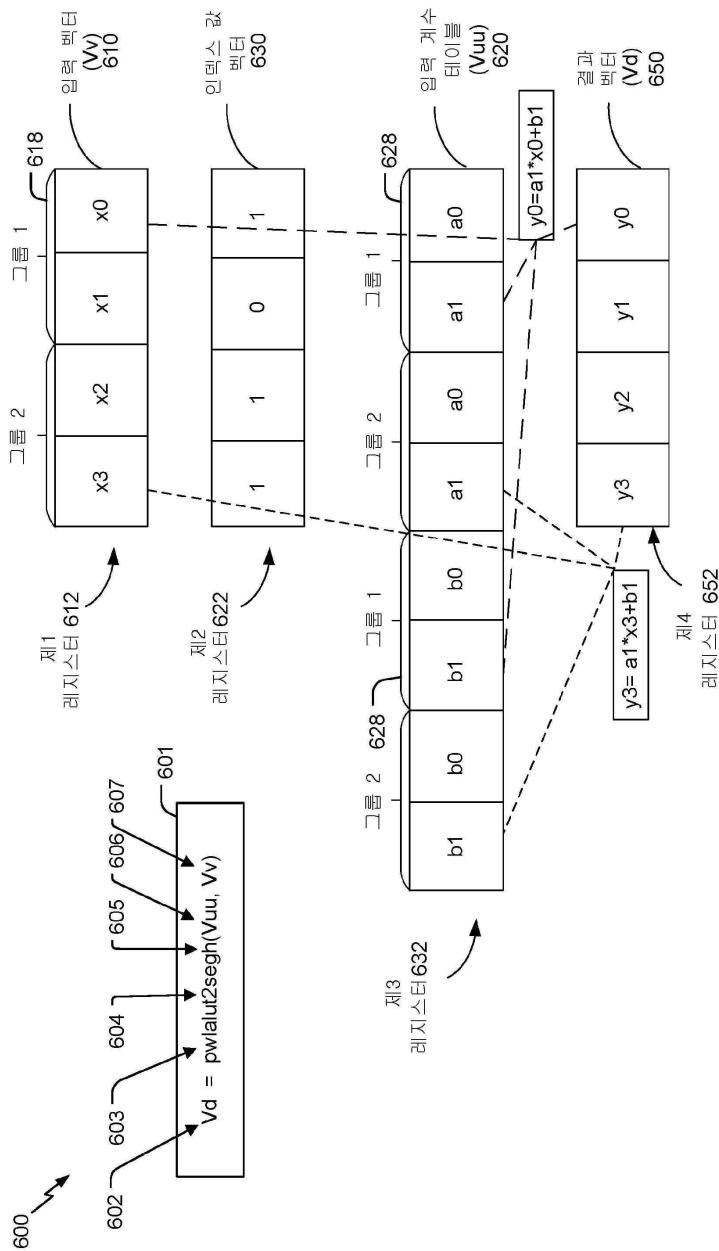
도면4



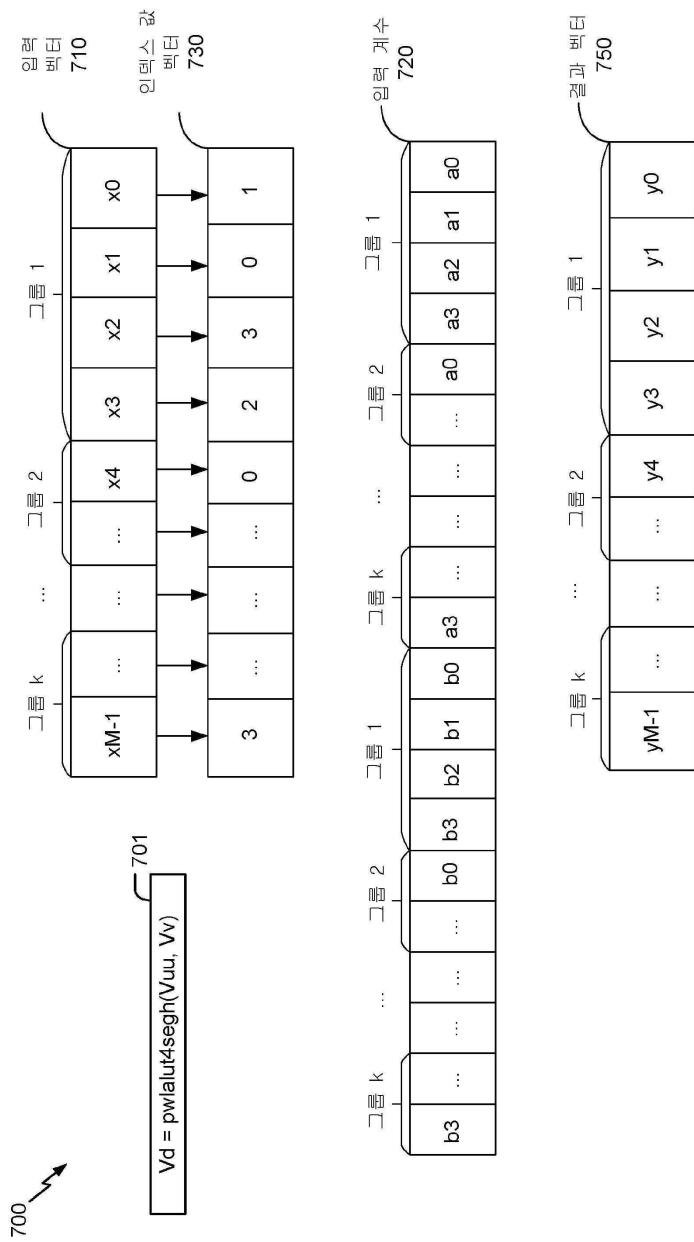
도면5



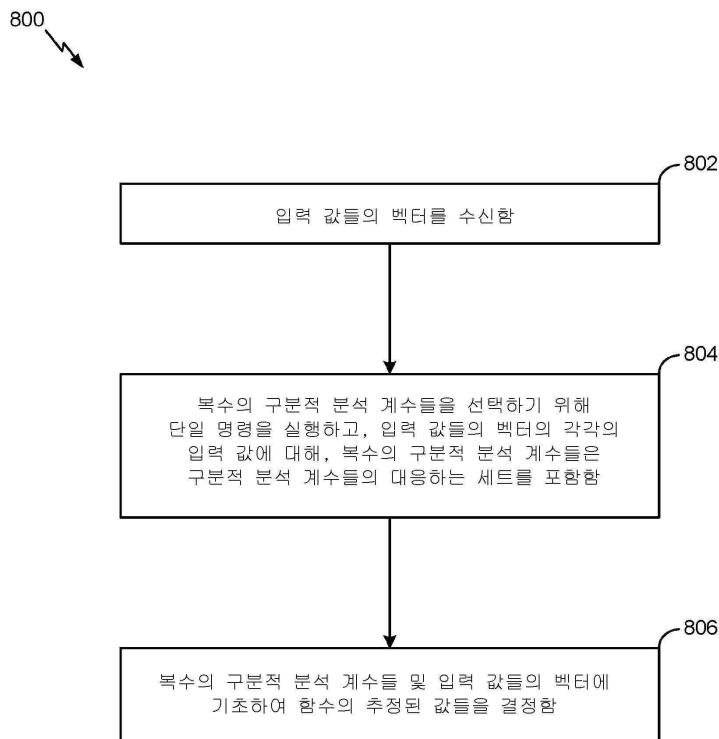
도면6



도면7



도면8



도면9

