



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년01월07일
(11) 등록번호 10-2199517
(24) 등록일자 2020년12월31일

(51) 국제특허분류(Int. Cl.)
G06F 7/556 (2006.01) G06F 17/16 (2006.01)
G06F 7/575 (2006.01)
(52) CPC특허분류
G06F 7/556 (2013.01)
G06F 17/16 (2013.01)
(21) 출원번호 10-2019-0095947
(22) 출원일자 2019년08월07일
심사청구일자 2019년08월07일
(56) 선행기술조사문헌
KR1020060097502 A*
KR1020090057588 A*
KR1020090071823 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
남병규
대전광역시 서구 도안북로 136, 111동 1504호
(74) 대리인
홍성욱, 심경식

전체 청구항 수 : 총 7 항

심사관 : 지정훈

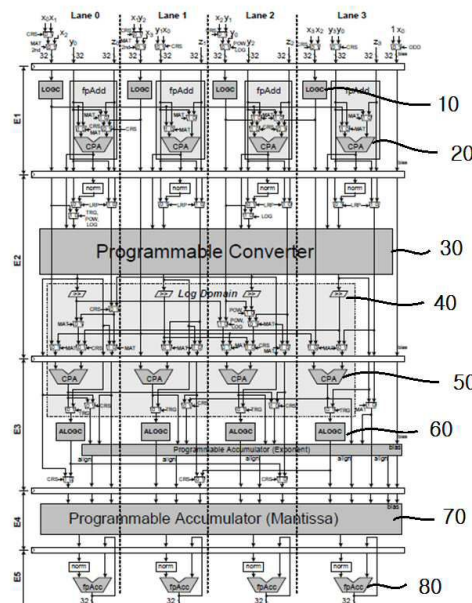
(54) 발명의 명칭 다기능 연산장치

(57) 요약

실시예는 행렬연산, 벡터연산, 초월함수의 구현대상 연산이 하나의 연산장치로 통합하여 구현되는 다기능 연산장치에 있어서, 3차원 그래픽스 연산을 위하여 입력된 값을 로그 도메인으로 변환하는 로그변환기와, 상기 로그 변환기의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 서로 합산하기 위한 제1 가산기와, 상기 제1 가산기

(뒷면에 계속)

대표도 - 도1



의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 이용하여 구현대상 연산을 수행하도록 프로그래밍되는 PCNV(Programmable Converter)와, 상기 PCNV 결과값을 자리 이동시키기 위한 쉬프터와, 상기 로그변환기의 결과값과 상기 쉬프터의 결과값을 합산하기 위한 제2 가산기와, 상기 제2 가산기의 결과값을 지수 도메인으로 변환하기 위한 역로그 변환기와, 상기 역로그 변환기 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 이용하여 구현대상 연산을 수행하도록 프로그래밍되는 PACC(Programmable Accumulator);를 포함하고, 상기 PCNV는 멱함수 계산을 수행하고, 두 번의 로그변환을 수행하여 상기 멱함수 계산을 덧셈으로 구현할 수 있으며, 있으며, 상기 PACC는 CSA 합산트리를 이용한 벡터내적 및 삼각함수 계산을 수행할 수 있다.

(52) CPC특허분류

G06F 7/575 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711093454
부처명	과학기술정보통신부
과제관리(전문)기관명	한국산업기술평가관리원
연구사업명	전자정보디바이스산업원천기술개발(R&D)
연구과제명	딥러닝 기반 상황인식용 임베디드 GP-GPU기술 개발
기 여 율	1/1
과제수행기관명	서경대학교 산학협력단
연구기간	2019.01.01 ~ 2019.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

행렬연산, 벡터연산, 초월함수의 구현대상 연산이 하나의 연산장치로 통합하여 구현되는 다기능 연산장치에 있어서,

3차원 그래픽스 연산을 위하여 입력된 값을 로그 도메인으로 변환하는 로그변환기;

상기 로그 변환기의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 서로 합산하기 위한 제1 가산기;

상기 제1 가산기의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 이용하여 구현대상 연산을 수행하도록 프로그래밍되는 PCNV(Programmable Converter);

상기 PCNV 결과값을 자리 이동시키기 위한 쉬프트;

상기 로그변환기의 결과값과 상기 쉬프트의 결과값을 합산하기 위한 제2 가산기;

상기 제2 가산기의 결과값을 지수 도메인으로 변환하기 위한 역로그 변환기;

상기 역로그 변환기 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 이용하여 구현대상 연산을 수행하도록 프로그래밍되는 PACC(Programmable Accumulator);를 포함하고,

상기 PCNV는 멱함수 계산을 수행하고, 두 번의 로그변환을 수행하여 상기 멱함수 계산을 덧셈으로 구현하고,

상기 멱함수는 두 번의 로그변환을 이용하여 로그 도메인에서 덧셈으로 수행한 다음 두 번의 역로그변환을 통해 구현하는, 다기능 연산장치.

청구항 2

제1항에 있어서,

상기 PCNV는 변환기를 재구성가능하도록 하여 행렬-벡터 곱셈 계산에 필요한 4개의 역로그 변환기, 벡터 곱셈, 나눗셈, 제곱근 계산 및 벡터 선형보간, 내적 계산에 필요한 4개의 로그 변환기, 벡터외적 계산시에 필요한 각 2개의 로그변환기 및 역로그 변환기, 상기 멱함수 계산에 필요한 1개의 로그 변환기, 삼각함수의 Taylor 급수 전개에 필요한 4개의 상수 곱셈기로 사용가능한, 다기능 연산장치.

청구항 3

제2항에 있어서,

상기 PCNV는 로그변환시 로그변환을 위한 LUT(Look Up Table)을 추가하여 합산 트리를 공유하며, 역로그변환시 역로그변환을 위한 LUT(Look Up Table)를 추가하여 합산트리를 공유하고, 상수 곱셈시 상수 곱셈을 위한 LUT(Look Up Table)를 추가하여 합산트리를 공유하도록 프로그래밍되는, 다기능 연산장치.

청구항 4

제1항에 있어서,

상기 PACC는 하나의 PACC를 재구성 가능하도록 하여 벡터 곱셈-합산 연합 (Multiply-Add), 외적 (cross product), 행렬-벡터 곱 구현을 위하여 4-way SIMD 가산기로 프로그래밍되고, 내적 (dot product) 및 삼각함수 구현을 위해서는 하나의 5-입력 CSA (carry save adder) 합산트리로 프로그래밍되며, 상기 PCNV에 역로그 변환을 위한 LUT를 추가하고 CSA 합산 트리는 공유하여 역로그 변환기로도 프로그램할 수 있도록 하여 멱함수의 2중 역로그 변환을 위한 하나의 역로그 변환기로 프로그래밍되는, 다기능 연산장치.

청구항 5

삭제

청구항 6

제2항에 있어서,

상기 벡터 선형 보간은 상기 제1 가산기를 이용하여 연산 수행한 다음 PCNV로 구현하는, 다기능 연산장치.

청구항 7

제2항에 있어서,

2개의 변수를 가지는 로그 함수는 아래 수식에 의하여,

$$\log_x y = \log_2 y / \log_2 x = 2^{\log_2(\log_2 y) - \log_2(\log_2 x)}$$

상기 로그 변환기와 상기 PCNV의 로그변환기를 직렬로 연결하여 구현하는, 다기능 연산장치.

청구항 8

제1항에 있어서,

상기 멱함수는 아래 식에 의해 구현되는 다기능 연산장치.

$$2^{(y \times \log_2 x)} = 2^{(2^{(\log_2 y + \log_2(\log_2 x))})}$$

발명의 설명

기술 분야

- [0001] 본 발명은 다기능 연산장치에 관한 것으로서, 특히 휴대단말기, 내장형 시스템 및 컴퓨터시스템에서 널리 사용되는 3차원 그래픽 프로세서 (GPU)를 위한 연산 장치를 저전력화, 소형화 및 고속화하기 위한 다기능 연산장치에 관한 것이다.

배경 기술

- [0002] 일반적으로 기존의 3차원 그래픽스 프로세서는 PC기반의 고성능 컴퓨터시스템을 대상으로 설계되었기 때문에, 면적과 전력소모가 크다.
- [0003] 현재, 실시간 3차원 그래픽스 분야는 하드웨어의 개선과 활용의 증가에 따라서 매우 빠른 속도로 발전하고 있다. 예전에는 CPU에서 구현하던 기능을 그래픽스 하드웨어로 넘겨줌으로써 성능을 높이고 CPU는 그래픽스 이외의 다른 작업에 몰두할 수 있게 되었다.
- [0004] 그러나, 최근에 스마트폰이나 스마트워치와 같은 휴대단말기 상에서 3차원 그래픽스 프로세서에 대한 수요가 증가하고 있고, 그 요구 사양 또한 점차적으로 증가하여 PC 기반의 그래픽스 프로세서에서 채택되었던 프로그램 가능한 그래픽스 파이프라인을 채택하게 되었다.
- [0005] 하지만, 휴대 시스템은 그 특성상 면적과 전력소모가 크기 때문에, 면적과 전력소모에서 많은 제약을 가지게 된다.
- [0006] 따라서 기존의 PC 기반 시스템을 대상으로 제안된 그래픽스 프로세서는 휴대 시스템에는 사용하기 적합하지 않는 문제점이 있다.
- [0007] 종래에는 행렬연산, 벡터연산, 초월함수의 구현 대상 연산이 하나로 연산장치로 통합하여 구현되는 다기능 연산 장치가 국내공개특허 제10-2009-0071823호에 개시되어 있다.
- [0008] 하지만, 종래의 다기능 연산 장치는 멱함수 계산을 위해서 Booth 곱셈기를 사용하기 때문에 불필요한 낭비에 의한 지연 시간을 감소하는데 한계가 있다.
- [0009] 또한, 종래의 다기능 연산 장치는 내적 (dot product) 및 삼각함수 구현을 위해 CPA (carry propagate adder) 트리를 사용하므로 지연시간을 감소하는데 한계가 있다.

발명의 내용

해결하려는 과제

- [0010] 상술한 문제점을 해결하기 위해 본 발명은 저전력화 및 소형화가 가능한 다기능 연산장치를 제공하는 것을 그 목적으로 한다.
- [0011] 또한, 본 발명은 지연 시간을 감소시키기 위한 다기능 연산장치를 제공함을 그 목적으로 한다.

과제의 해결 수단

- [0012] 실시예는 행렬연산, 벡터연산, 초월함수의 구현대상 연산이 하나의 연산장치로 통합하여 구현되는 다기능 연산 장치에 있어서, 3차원 그래픽스 연산을 위하여 입력된 값을 로그 도메인으로 변환하는 로그변환기와, 상기 로그 변환기의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 서로 합산하기 위한 제1 가산기와, 상기 제1 가산기의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 이용하여 구현대상 연산을 수행하도록 프로그래밍 되는 PCNV(Programmable Converter)와, 상기 PCNV 결과값을 자리 이동시키기 위한 쉬프트와, 상기 로그변환기의 결과값과 상기 쉬프트의 결과값을 합산하기 위한 제2 가산기와, 상기 제2 가산기의 결과값을 지수 도메인으로 변환하기 위한 역로그 변환기와, 상기 역로그 변환기 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 이용하여 구현대상 연산을 수행하도록 프로그래밍되는 PACC(Progammmable Accumulator);를 포함하고, 상기 PCNV는 곱 함수 계산을 수행하고, 두 번의 로그변환을 수행하여 상기 곱함수 계산을 덧셈으로 구현할 수 있다.
- [0013] 상기 PCNV는 변환기를 재구성가능하도록 하여 행렬-벡터 곱셈 계산에 필요한 4개의 역로그 변환기, 벡터 곱셈, 나눗셈, 제곱근 계산 및 벡터 선형보간, 내적 계산에 필요한 4개의 로그 변환기, 벡터외적 계산시에 필요한 각 2개의 로그변환기 및 역로그 변환기, 상기 곱함수 계산에 필요한 1개의 로그 변환기, 삼각함수의 Taylor 급수 전개에 필요한 4개의 상수 곱셈기로 사용할 수 있다.
- [0014] 상기 PCNV는 로그변환시 로그변환을 위한 LUT(Look Up Table)을 추가하여 합산 트리를 공유하며, 역로그변환시 역로그변환을 위한 LUT(Look Up Table)를 추가하여 합산트리를 공유하고, 상수 곱셈시 상수 곱셈을 위한 LUT(Look Up Table)를 추가하여 합산트리를 공유하도록 프로그래밍될 수 있다.
- [0015] 상기 PACC는 하나의 PACC를 재구성 가능하도록 하여 벡터 곱셈-합산 연합 (Multiply-Add), 외적 (cross product), 행렬-벡터 곱 구현을 위하여 4-way SIMD 가산기로 프로그래밍되고, 내적 (dot product) 및 삼각함수 구현을 위해서는 하나의 5-입력 CSA (carry save adder) 합산트리로 프로그래밍되며, 상기 PCNV에 역로그 변환을 위한 LUT를 추가하고 CSA 합산트리는 공유하여 역로그 변환기로도 프로그래밍할 수 있도록 하여 곱함수의 2중 역로그 변환을 위한 하나의 역로그 변환기로 프로그래밍될 수 있다.
- [0016] 상기 곱함수는 두 번의 로그변환을 이용하여 로그도메인에서 덧셈으로 수행한 다음 두 번의 역로그변환을 통해 구현할 수 있다.
- [0017] 상기 벡터 선형 보간은 상기 제1 가산기를 이용하여 연산 수행한 다음 PCNV로 구현할 수 있다.
- [0018] 2개의 변수를 가지는 로그 함수는 아래 수식에 의하여, 상기 로그 변환기와 상기 PCNV의 로그변환기를 직렬로 연결하여 구현할 수 있다.

$$\log_x y = \log_2 y / \log_2 x = 2^{\log_2(\log_2 y) - \log_2(\log_2 x)}$$

- [0019]
- [0020] 상기 곱함수는 아래 식에 의해 구현되는 다기능 연산장치.

$$2^{(y \times \log_2 x)} = 2^{(2^{(\log_2 y + \log_2(\log_2 x))})}$$

[0021]

발명의 효과

- [0022] 실시예는 곱함수 계산을 덧셈을 통해 구현하도록 함으로써, 지연 시간과 면적 그리고 전력소모를 개선하는 효과가 있다.
- [0023] 또한, 실시예는 내적 및 삼각함수에 필요한 합산을 CSA 합산트리를 통해 구현함으로써, 지연 시간과 전력 소모를 개선하는 효과가 있다.

도면의 간단한 설명

- [0024] 도 1은 본 발명에 따른 다기능 연산장치의 전체 구성을 나타내는 도면이다.
- 도 2a 및 2b는 본 발명에 따른 다기능 연산장치의 로그변환방법을 설명하기 위한 도면이다.
- 도 3a 및 3b는 본 발명에 따른 다기능 연산장치의 역로그 변환방법을 설명하기 위한 도면이다.
- 도 4는 본 발명에 따른 다기능 연산장치의 PCNV(programmable converter)의 구성을 나타낸 도면이다.
- 도 5는 본 발명에 따른 다기능 연산장치의 PACC(programmable accumulator)의 구성을 나타낸 도면이다.
- 도 6은 본 발명에 따른 다기능 연산장치의 PACC(programmable accumulator)의 구성을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0025] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.
- [0026] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0027] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.
- [0028] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0029] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0031] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세하게 설명한다.
- [0032] 도 1은 본 발명에 따른 다기능 연산장치의 전체 구성을 나타내는 도면이고, 도 2a 및 2b는 본 발명에 따른 다기능 연산장치의 로그변환방법을 설명하기 위한 도면이며, 도 3a 및 3b는 본 발명에 따른 다기능 연산장치의 역로그 변환방법을 설명하기 위한 도면이고, 도 4는 본 발명에 따른 다기능 연산장치의 상수 곱셈방법을 설명하기 위한 도면이며, 도 5는 본 발명에 따른 다기능 연산장치의 PCNV(programmable converter)의 구성을 나타낸 도면이며, 도 6은 본 발명에 따른 다기능 연산장치의 PACC(programmable accumulator)의 구성을 나타낸 도면이다.
- [0033] 도 1에 나타난 바와 같이, 본 발명에 따른 연산장치는 4-channel, 5-stage의 파이프라인으로 구성되어 있으며, stage 1에는 입력 데이터를 로그 도메인으로 변환하는 로그 변환기(10), stage 2에는 구현하고자 하는 연산에 따라 프로그램가능한 PCNV(programmable converter, 30)를 포함하고 있다. stage 3에는 로그도메인의 연산결과를 고정/부동 소수점 결과로 변환하기 위한 역로그 변환기(60)를 포함하고 있으며, stage 4에는 구현대상 연산에 따라 프로그램가능한 PACC(programmable accumulator, 70)를 포함하고 있다. stage 5에는 아래에서 설명할 행렬연산을 구현하기 위한 누산기(80)를 포함하고 있다.

- [0034] 여기서, 상기 stage1은 상기 로그변환기(10)의 결과값과 3차원 그래픽스 연산을 위하여 입력된 값을 서로 합산하기 위한 제1 가산기(20)를 더 포함하고, 상기 stage2는 상기 PCNV(30) 결과값을 자리 이동시키기 위한 쉬프터(40)를 더 포함하며, 상기 stage3은 상기 로그변환기(10)의 결과값과 상기 쉬프터(40)의 결과값을 합산하기 위한 제2 가산기(50)를 더 포함하여 구성된다.
- [0035] 특히, 본 발명은 저전력 동작을 위하여 입출력 데이터로 고정 소수점 수 체계 (Fixed-Point Number System) 혹은 부동 소수점 수 체계 (Floating-Point Number System)의 데이터를 처리하며, 연산의 복잡도를 줄이기 위해 입력으로 받은 고정 혹은 부동 소수점 데이터를 로그 도메인의 데이터인 로그 수 체계 (Logarithmic Number System; LNS)로 변환하여 계산한다. 이렇게 로그 수로 계산된 데이터는 입출력 형식인 고정/부동 소수점 데이터로 변환되어 출력된다.
- [0036] 이때, 고정/부동 소수점 데이터를 LNS의 데이터로 변환하는 로그 변환기의 정확도가 연산의 정확도를 결정지므로, 로그 변환기의 변환 에러를 줄이는 것이 중요하다.
- [0037] 또한, 본 발명에서는 저전력으로 로그 변환기를 구현하기 위하여 구간별 선형 근사법(piecewise linear approximation)을 사용한다. 이 로그 변환기에서는 소수부분은 [0,1]의 값에 대해서 구간을 나누어 각 구간을 선형으로 근사화 하고, 정수 부분은 고정 소수점 데이터일 경우 소수점으로부터 leading one의 위치를 count함으로써 구할 수 있으며, 부동 소수점 데이터일 경우 지수 (exponent)부분을 취함으로써 얻을 수 있다. 이 때, 로그 함수에서는 입력데이터가 '1'에 가까워질수록 출력데이터가 '0'에 가까워지므로, 이 구간에서는 작은 에러값도 원 값에 대한 비율 (%)은 크게 나타나는 문제점을 가지고 있다.
- [0038] 이를 해결하기 위해 본 발명에서는 '1'에 가까운 구간에서는 근사 구간을 더욱 세분하여 근사화 함으로써 에러를 줄이는 기법을 제안하였다.
- [0039] 도 2a 및 2b에서는 본 발명에 따른 구간별 선형 근사에 기반하는 로그 변환법과 이를 이용한 구간별 선형 근사를 LUT(lookup table, 15), CSA(Carry Save Adder, 16) 및 CPA(Carry Propagation Adder, 17)로 구성된 가산기 트리(adder tree)로 구현하는 장치를 설명하는 것으로서, '1'에 가까운 구간에서는 더욱 세분하여 근사화 함으로써 에러를 줄이기 위한 방법을 사용하였다.
- [0040] 도 3a 및 3b에서는 본 발명에 따른 역로그 변환법과 이를 이용한 장치를 설명하기 위한 것으로서, 도 2에서 처럼 로그 도메인에서 연산된 결과값을 고정 및 부동 소수점 결과, 즉 지수 도메인으로 변환하는 역로그변환법으로서, 구간별 선형근사를 위한 LUT(65), CSA(66) 및 CPA(67)로 구성된 가산기 트리로 구현되는 장치를 사용하여 에러를 줄이기 위한 방법을 사용하였다.
- [0041] 도 4에서는 본 발명에 따른 상수 곱셈장치를 설명하기 위한 것으로서 상수 곱셈을 미리 정해진 쉬프트 연산들의 조합으로 구현하는 방법으로서, 쉬프트 계수를 저장하는 LUT와 쉬프트 연산결과들의 합산을 위한 CSA 및 CPA로 구성된 가산기 트리로 구현되는 장치를 사용하였다.
- [0042] 도 5에서는 본 발명에 따른 다기능 연산장치의 PCNV(programmable converter)의 구성을 설명한다.
- [0043] 기존의 발명에서는 벡터 연산과 초월함수 연산을 함께 구현하기 위해서 stage 1에 8개의 로그 변환기를 두었으며, 로그 도메인인 stage 2에 Booth 곱셈기를 두었다. 하지만, 이는 벡터 연산에 대해서는 Booth 곱셈기가 낭비되고, 초월함수에 대해서는 7개의 로그 변환기가 낭비되는 결과를 초래하였다.
- [0044] 또 다른 기존의 발명에서는 8개의 로그 변환기 중 4개는 stage 1에 두고, 나머지 4개는 stage 2에 두고 로그 변환기 및 Booth 곱셈기에 공통으로 필요한 가산기 트리(adder tree)를 공유하여 프로그램 가능하도록 PMUL(Programmable Multiplier)을 만들어, 벡터 연산에 대해서는 로그 변환기로, 초월함수에 대해서는 Booth 곱셈기로 각각 프로그램 되도록 함으로써, 불필요한 낭비를 줄여주었지만 지연시간의 감소를 가져오지는 못하였다.
- [0045] 따라서, 수학적 식 1b에서 보인바와 같이, 본 발명에서는 두 단계의 로그 변환을 이용하여 멱함수 계산을 덧셈을 통하여 구현할 수 있고, 이를 통해 기존 수학적 식 1a를 통해 멱함수 계산에 필요하던 Booth 곱셈기를 제거함으로써 기존의 PMUL을 경량화할 수 있다.
- [0046] [수학적 식 1]

[0047]
$$x^y = 2^{y \times \log_2 x} \quad (a)$$

$$2^{(y \times \log_2 x)} = 2^{(2^{(\log_2 y + \log_2(\log_2 x))})} \quad (b)$$

[0048]

[0049] 또한, 수학식 2에서 보인바와 같이, 본 발명에서는 삼각함수의 Taylor 급수 전개에 필요한 로그 도메인의 곱셈을 상수 곱셈기로 구현함으로써 PMUL에서 Booth 곱셈기능을 완전히 제거할 수 있으며, 이를 통해 경량화된 PCNV를 제안한다.

[0050]

[수학식 2]

$$c_0 x^{k_0} \oplus c_1 x^{k_1} \oplus c_2 x^{k_2} \oplus c_3 x^{k_3} \oplus c_4 x^{k_4} = c_0 x^{k_0} \oplus 2^{(\log_2 c_1 + k_1 \times \log_2 x)} \oplus 2^{(\log_2 c_2 + k_2 \times \log_2 x)} \oplus 2^{(\log_2 c_3 + k_3 \times \log_2 x)} \oplus 2^{(\log_2 c_4 + k_4 \times \log_2 x)}$$

[0051]

[0052] 여기서, $\oplus_i \in \{+, -\}$ 이고, c_i 및 k_i 는 상수로서 각각 양의 실수 및 정수이다.

[0053]

도 4에 보인 바와 같이, PCNV는 로그 변환기, 역로그 변환기 및 상수 곱셈기에 공통으로 필요한 가산기 트리(adder tree)를 공유하여 프로그램 가능하도록 하여, 벡터 연산에 대해서는 4개의 로그 변환기로, 멱함수에 대해서는 하나의 로그 변환기로, 삼각함수에 대해서는 4개의 상수 곱셈기로 각각 프로그램 되도록 함으로써, 기존 발명에서 필요하였던 멱함수와 삼각함수를 위한 Booth 곱셈기를 없애어 지연시간을 줄여주게 된다.

[0054]

또한, PCNV에 역로그 변환을 위한 LUT(36)를 추가하고 가산기 트리는 공유하여 역로그 변환기로도 프로그램할 수 있도록 하여, 행렬-벡터 곱셈, 벡터 외적 등에도 활용할 수 있도록 하였다.

[0055]

상기 PCNV는 변환기를 재구성가능하도록 하여 행렬-벡터 곱셈 계산에 필요한 4개의 역로그 변환기, 벡터 곱셈, 나눗셈, 제곱근 계산 및 벡터 선형보간, 내적 계산에 필요한 4개의 로그 변환기, 벡터외적 계산시에 필요한 각 2개의 로그변환기 및 역로그 변환기, 멱함수 계산에 필요한 1개의 로그 변환기, 삼각함수의 Taylor 급수 전개에 필요한 4개의 상수 곱셈기로 모두 사용가능하다.

[0056]

도 6은 본 발명에 따른 다기능 연산장치의 PACC(programmable accumulator)의 구성을 설명하기 위함이다.

[0057]

이는 벡터 곱셈-합산 연합 (Multiply-Add), 외적 (cross product), 행렬-벡터 곱 구현을 위해서는 4-way SIMD(Single Instruction Multiple Data) 가산기로 프로그램되고, 내적 (dot product) 및 삼각함수 구현을 위해서는 하나의 5-입력 CSA (carry save adder) 합산 트리(5-input adder tree)로 프로그램될 수 있다. 또한, PACC에 역로그 변환을 위한 LUT(36)를 추가하고 CSA 합산 트리는 공유하여 역로그 변환기로도 프로그램할 수 있도록 하여 수식 (1b)에서 보인 멱함수의 2중 역로그 변환을 위해 하나의 역로그 변환기로 프로그램될 수 있다.

[0058]

상기와 같이 구성된 본 발명에 따른 다기능 연산장치는 GPU에 사용되는 연산들의 복잡도를 줄이기 위하여 합산과 뺄셈을 제외한 모든 연산을 로그 도메인으로 변환하여 수행하여, 로그 도메인에서는 곱셈이 합산, 나눗셈이 뺄셈, 제곱근이 쉬프트, 그리고 멱함수가 곱셈으로 연산의 복잡도가 줄어드는 장점이 있다.

[0059]

특히, 본 발명에서는 이중 로그변환을 통해 멱함수가 덧셈으로 연산의 복잡도가 들어드는 장점을 가진다.

[0060]

이를 위해서는 입력 값을 로그 도메인으로 변환해주는 로그 변환기와, 로그 도메인에서 계산된 값을 다시 원래의 도메인으로 변환해주는 역로그 변환기가 필요하다. 특히, 삼각함수들 (삼각함수, 쌍곡선함수 및 그들의 역함수)의 연산 복잡도를 감소시키기 위해서 본 발명에서는 이들을 Taylor 급수로 전개하여, 이를 로그 도메인에서 계산함으로써 삼각함수들에 대한 연산의 복잡도를 낮추어 처리할 수 있도록 하였다.

[0061]

한편, 기존 발명 중에도 로그 도메인에서의 연산을 이용함으로써 성능을 증가시킨 예가 있지만, 본 발명에서는 멱함수와 삼각함수까지 하나의 연산 장치에 통합 구현기 위해 필요하였던 Booth 곱셈기를 제거함으로써 지연시간과 면적의 개선을 이루었다. 또한 본 발명에서는 내적과 삼각함수 계산에 필요한 합산을 CSA 트리를 이용함으로써 지연시간의 개선을 이루었다.

[0062]

본 발명에서는 GPU에서 필요한 행렬 연산을 2주기 스트루풋(2-cycle throughput)으로 처리하고, 벡터 및 초월함수 연산을 단일 주기 스트루풋(single-cycle throughput)으로 처리함으로써, GPU의 스트루풋을 크게 증가시키고, 이들을 하나의 연산 장치에 통합 구현함으로써 저전력화 및 소형화가 가능하도록 하였다.

[0064]

이하, 본 발명에서 제안한 각 연산에 대한 구현 방법을 설명하면 다음과 같다.

[0065]

1. 행렬-벡터 곱 (Matrix-Vector Multiplication)

[0066] 3차원 그래픽스에서 요구되는 기하변환(geometry transformation)을 수행하기 위해서는 4x4 행렬과 4-엘리먼트(element) 벡터의 곱셈이 필요하다. 수학식 3에서 보인 바와 같이, 이는 16개의 곱셈 연산을 필요로 하는데, 이는 로그 도메인에서 20개의 로그 변환기, 16의 가산기, 16개의 역로그 변환기를 필요로 한다.

[0067] [수학식 3]

$$\begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} c_{00} \\ c_{10} \\ c_{20} \\ c_{30} \end{bmatrix} x_0 + \begin{bmatrix} c_{01} \\ c_{11} \\ c_{21} \\ c_{31} \end{bmatrix} x_1 + \begin{bmatrix} c_{02} \\ c_{12} \\ c_{22} \\ c_{32} \end{bmatrix} x_2 + \begin{bmatrix} c_{03} \\ c_{13} \\ c_{23} \\ c_{33} \end{bmatrix} x_3 =$$

$$2^{\left(\begin{bmatrix} \log_2 c_{00} \\ \log_2 c_{10} \\ \log_2 c_{20} \\ \log_2 c_{30} \end{bmatrix} + \log_2 x_0 \right)} + 2^{\left(\begin{bmatrix} \log_2 c_{01} \\ \log_2 c_{11} \\ \log_2 c_{21} \\ \log_2 c_{31} \end{bmatrix} + \log_2 x_1 \right)} + 2^{\left(\begin{bmatrix} \log_2 c_{02} \\ \log_2 c_{12} \\ \log_2 c_{22} \\ \log_2 c_{32} \end{bmatrix} + \log_2 x_2 \right)} + 2^{\left(\begin{bmatrix} \log_2 c_{03} \\ \log_2 c_{13} \\ \log_2 c_{23} \\ \log_2 c_{33} \end{bmatrix} + \log_2 x_3 \right)}$$

MAT 1st phase MAT 2nd phase

[0068]

[0069] 3차원 그래픽스에서 기하변환(geometry transformation) 행렬의 원소들은 하나의 3차원 물체를 변환하는 동안 고정되므로, 행렬원소들은 연산이 시작되기 전 미리 로그 도메인으로 변환될 수 있다.

[0070] 따라서, 수학식 1의 연산을 구현하기 위해 필요한 20개의 로그변환기 개수는 벡터 원소 변환만을 위한 4개의 로그변환기로 줄어들게 되어, 4개의 로그 변환기, 16개의 가산기, 16개의 역로그 변환기가 수학식 3을 위해 필요하게 된다.

[0071] 이를 본 발명에서 제안하는 4-way 연산기에 구현하기 위해 수학식 3에서 보인바와 같이 2 페이스(phase)로 나누어 구현하면 1 페이스에 2개의 로그 변환기, 8개의 가산기, 8개의 역로그 변환기만이 필요하게 된다. 이때 필요한 8개의 가산기는 stage 1과 stage 3의 제1 및 제2 가산기들을 이용하면 되고, 8개의 역로그 변환기는 stage 3의 4개 역로그 변환기와 stage 2의 PCNV를 4개의 역로그 변환기로 프로그램 함으로써 구현할 수 있다.

[0072] stage 2의 역로그 변환 결과들과 stage 3의 역로그 변환 결과들은 PACC를 4-way SIMD 가산기로 프로그램 하여 더함으로써 페이스 1의 연산 결과를 얻을 수 있고, 이 같은 과정을 반복하여 페이스 2결과를 얻어 stage 5의 누산기를 통해 phase 1,2의 결과를 누적함으로써 최종 연산결과를 얻을 수 있다. 이와 같은 방법으로 기존의 4-way 연산기에서 4 주기 스루풋(4-cycle throughput)으로 구현하던 행렬-벡터 곱셈을 2 주기 스루풋(2-cycle throughput)으로 향상시킬 수 있다.

[0073] 2. 벡터 합산, 뺄셈 (Add, Subtract)

[0074] 합산, 뺄셈은 로그 도메인으로 변환하지 않고, 고정/부동 소수점 도메인에서 처리한다. 이는 도 1에 나타난 stage 1의 제1 가산기(20)를 이용한다.

[0075] 3. 벡터 곱셈, 나눗셈, 제곱근 (Multiply, Divide, Square-root) 및 합산 연합연산 (Multiply-Add, Divide-Add, Sqrt-Add)

[0076] 수학식 4에서와 같이, 곱셈, 나눗셈, 제곱근은 로그 도메인에서 각각 합산, 뺄셈, 쉬프트 연산으로 변환하여 처리한다. 이를 위해 도 1에 나타난 stage 2의 PCNV(30)는 4개의 로그 변환기로 프로그램되며, stage 2의 시프터(shifter, 40)와 stage 3의 제 2가산기를 이용한다. 또한, 이들과 합산의 연합연산은 도 1에 나타난 stage 4의 가산기들을 이용한다.

[0077] [수학식 4]

$$\left(x_i \otimes y_i^p \oplus z_i \right)_{i \in \{0,1,2,3\}} =$$

$$\left(2^{(\log_2 x_i) \oplus (\log_2 y_i - q)} \oplus z_i \right)_{i \in \{0,1,2,3\}}$$

[0078]

[0079] 여기서, $\otimes \in \{\times, \div\}$, $\oplus \in \{+, -\}$, $p \in \{0.5, 1\}$, $q \in \{0, 1\}$ 이다.

[0080] 4. 벡터 선형 보간 (Linear interpolation)

[0081] 수학적 식 5와 같이, $\log_2(z_i - y_i)$ 는 뺄셈 수행 후 로그변환을 필요로 한다. 이를 위해, stage 2의 PCNV(30)는 로그 변환기로 프로그램 되며, stage 1의 제1 가산기(20)를 이용해 뺄셈을 수행한 후 로그 변환을 수행한다.

[0082] [수학적 식 5]

$$(x_i(z_i - y_i) + y_i)_{i \in \{0,1,2,3\}} = (2^{\log_2 x_i + \log_2(z_i - y_i)} + y_i)_{i \in \{0,1,2,3\}}$$

[0083]

[0084] 5. 벡터 내적 (Dot-product) 및 외적 (Cross-product)

[0085] 벡터 내적은 두 벡터의 각 요소들 간의 곱으로 이루어진 항들의 합으로 정의된다. 따라서 각 벡터 요소들 간의 곱은 2번에서 설명한 바와 같이 로그 도메인에서 수행한 후, 고정/부동 소수점 도메인으로 역로그 변환을 하여 그 곱셈 결과들을 합함으로써 구하도록 한다. 이 곱셈을 위해 stage 2의 PCNV(30)는 4개의 로그 변환기로 프로그램 되며, 곱셈 결과들의 합산을 위해 stage 4의 PACC(70)는 5-input 합산 트리로 프로그램 된다.

[0086] 벡터 외적은 수학적 식 6에서 보이는 바와 같이, 6개의 곱셈을 필요로 하므로 12개의 로그 변환기, 6개의 가산기, 6개의 역로그 변환기를 필요로 하지만 6개의 서로 다른 피연산자(operand)만을 사용하므로 필요한 로그 변환기의 수를 6개로 줄일 수 있다.

[0087] [수학적 식 6]

$$\begin{bmatrix} x_1 y_2 - y_1 x_2 \\ x_2 y_0 - y_2 x_0 \\ x_0 y_1 - y_0 x_1 \end{bmatrix} = \begin{bmatrix} 2^{\log_2 x_1 + \log_2 y_2} - 2^{\log_2 y_1 + \log_2 x_2} \\ 2^{\log_2 x_2 + \log_2 y_0} - 2^{\log_2 y_2 + \log_2 x_0} \\ 2^{\log_2 x_0 + \log_2 y_1} - 2^{\log_2 y_0 + \log_2 x_1} \end{bmatrix}$$

[0088]

[0089] 따라서, stage 2에 있는 PCNV(30)를 2개의 로그 변환기와 2개의 역로그 변환기로 프로그램 함으로써, 외적에 필요한 6개의 로그변환기 및 6개의 역로그 변환기를 stage 1, 2, 3의 변환기들로부터 마련할 수 있고 6개의 가산기들은 stage 3에 있는 제2 가산기들(60) 및 stage 1의 제1 가산기들(20)로부터 마련할 수 있다.

[0090] 6. 로그 함수 ($\log_x y$)

[0091] 기존 발명에서는 로그 함수의 밑수는 임의의 상수였지만, 본 발명에서는 2개의 변수를 가지는 로그 함수를 구현한다. 2개 변수를 가지는 로그 함수는 수학적 식 7과 같이 로그 도메인 연산을 이용하여 구현할 수 있다.

[0092] [수학적 식 7]

$$\log_x y = \log_2 y / \log_2 x = 2^{\log_2(\log_2 y) - \log_2(\log_2 x)}$$

[0093]

[0094] 7. 멱함수 (power)

[0095] 멱함수는 연산의 복잡도가 큰 함수중의 하나이지만, 로그 도메인에서는 앞서 보인 수학적 식 1b)와 같이 2번의 로그변환 및 역로그 변환을 통해 덧셈연산으로 계산이 가능하다.

[0096] 따라서, 본 발명에서는 PCNV를 로그 변환기로 프로그램 가능하도록 하고 PACC를 역로그 변환기로 프로그램 가능하도록 만듦으로써, 멱함수의 계산에 필요한 2단계의 로그변환과 2단계의 역로그변환이 각각 가능하도록 하였다.

[0097] 8. 삼각함수 (삼각(trigonometric)함수, 쌍곡선(hyperbolic)함수, 역삼각(inverse-trigonometric)함수, 역쌍곡선(inverse-hyperbolic)함수 포함)

[0098] 삼각함수(삼각함수, 쌍곡선함수, 역삼각함수, 역쌍곡선함수 포함)와 같은 경우, Taylor 급수 전개를 통하여 삼각함수도 로그 도메인으로 변환되도록 하여 연산의 복잡도를 줄였다. 삼각함수에 대한 Taylor 급수 전개에 사용되는 각 항들을 계산하기 위해서는 각 항에 대하여 입력 값의 멱함수와 계수 곱셈이 요구되는데 이를 로그 도메인으로 변환하면 수학적 식 2와 같이 상수곱셈과 합산으로 변환된다.

[0099] 여기서 필요한 상수곱셈은 PCNV를 4-way 상수 곱셈기로 프로그램 함으로써 구현한다. 이 곱셈기는 도 4에 나타나 있으며, 이는 4-way 상수 곱셈기와 계수를 위한 가산기로 함께 구성될 수 있다.

[0100] 이렇게 구현된 항들은 도 7에 나타난 바와 같이, PACC를 5-input 합산 트리로 프로그램 하여 합산함으로써 삼각 함수를 구현할 수 있다.

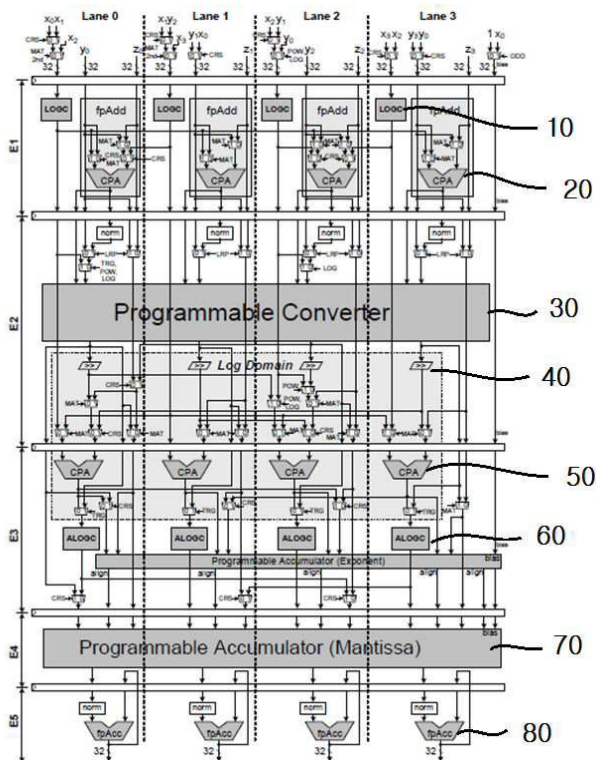
[0101] 위의 식에서 삼각함수들에 대한 Taylor 급수 전개 시 첫째 항은 항상 상수 '1' 이거나 1차항이 되므로, 이는 로그 변환기와 곱셈기를 통하지 않고 도 5에 나타난 바와 같이 합산 트리에 직접 더해지도록 함으로써 하나의 로그 변환기와 곱셈기를 줄일 수 있다. 이로 인해 각 삼각함수를 5개항의 Taylor 급수로 근사함으로써 근사화에 따르는 에러를 줄였다.

부호의 설명

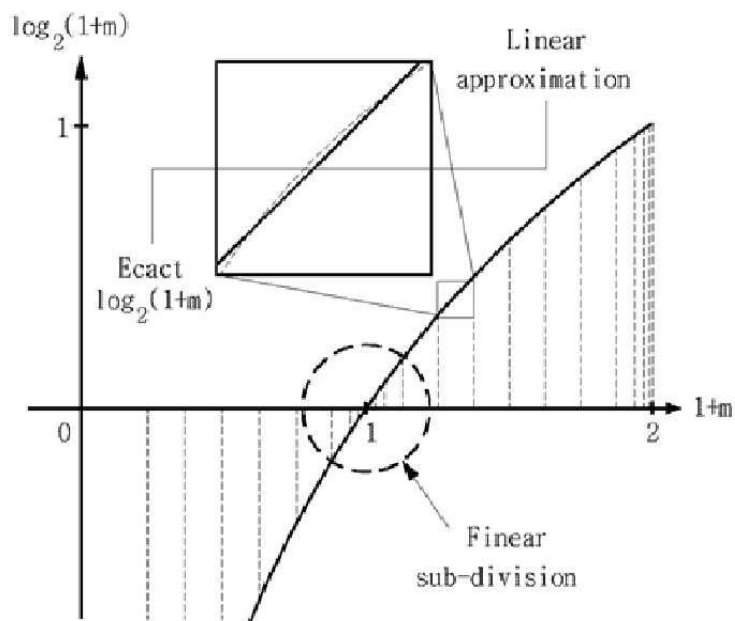
- [0102]
- 10: 로그 변환기
 - 20: 제1 가산기
 - 30: PCNV
 - 40: 쉬프터
 - 50: 역로그 변환기

도면

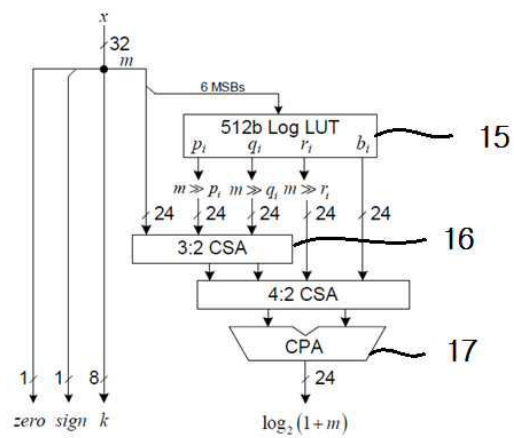
도면1



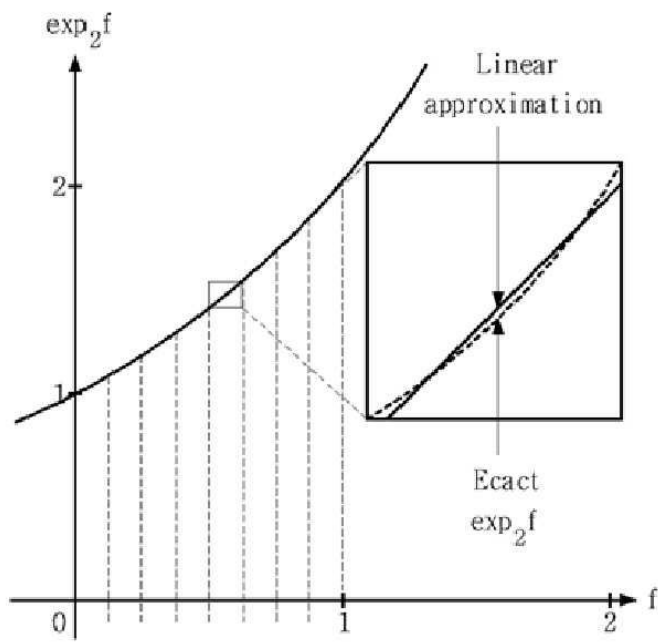
도면2a



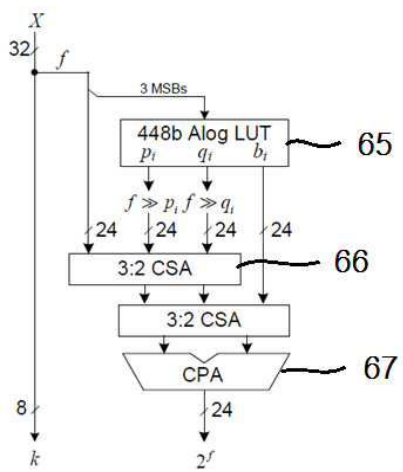
도면2b



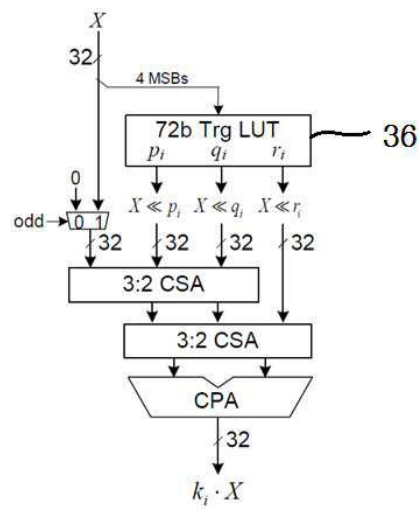
도면3a



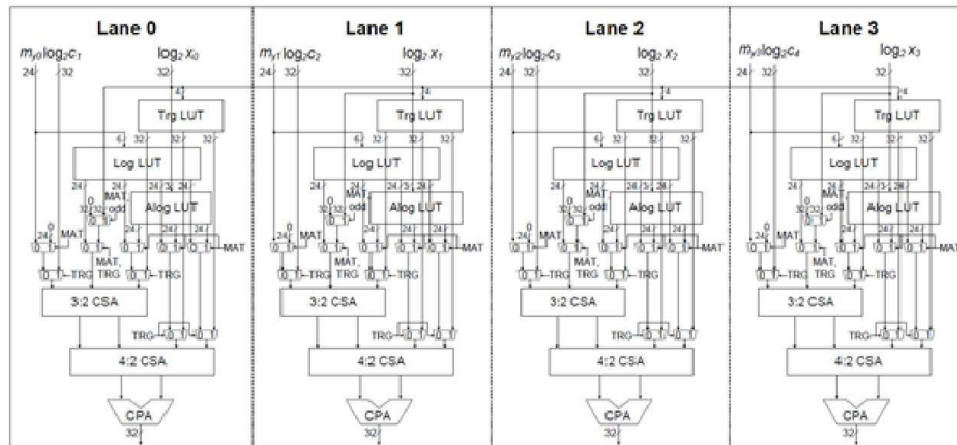
도면3b



도면4



도면5



도면6

