



공개특허 10-2022-0148328



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0148328
(43) 공개일자 2022년11월04일

- (51) 국제특허분류(Int. Cl.)
G06F 1/3287 (2019.01) *G06T 1/20* (2018.01)
(52) CPC특허분류
G06F 1/3287 (2019.01)
G06T 1/20 (2013.01)
(21) 출원번호 10-2022-7036929(분할)
(22) 출원일자(국제) 2015년07월29일
심사청구일자 없음
(62) 원출원 특허 10-2017-7005783
원출원일자(국제) 2015년07월29일
심사청구일자 2020년07월29일
(85) 번역문제출일자 2022년10월24일
(86) 국제출원번호 PCT/IB2015/001845
(87) 국제공개번호 WO 2016/016730
국제공개일자 2016년02월04일
(30) 우선권주장
62/030,913 2014년07월30일 미국(US)
(뒷면에 계속)

- (71) 출원인
모비디어스 리미티드
네털란드 엔지 스히폴-리크 1119 카프로닐란 37
(72) 발명자
배리, 브랜단
아일랜드 더블린 14 쳐치타운 랜드스케이프 가든
스 8
리치몬드, 리차드
영국 비티5 5엔유 벨페스트 안트림 선버리 애비뉴
14
(뒷면에 계속)
(74) 대리인
양영준, 김연송, 백만기

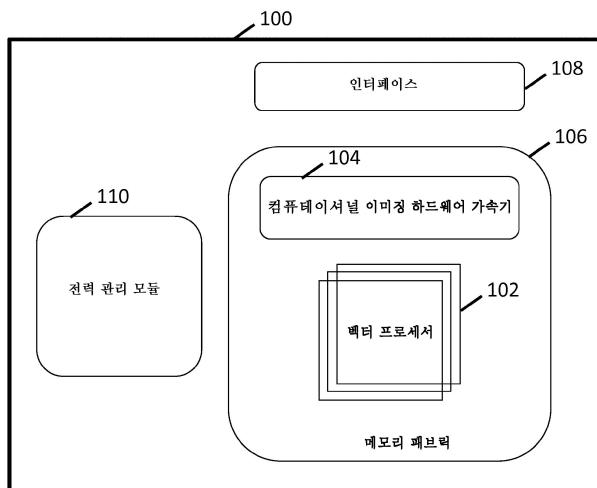
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 저전력 컴퓨터이셔널 이미징

(57) 요 약

본 출원은, 컴퓨터이셔널 이미징을 위한 저전력, 고성능 컴퓨팅 플랫폼을 제공할 수 있는 컴퓨팅 디바이스(100)를 개시한다. 이 컴퓨팅 디바이스는, 하나 이상의 처리 유닛, 예를 들어, 하나 이상의 벡터 프로세서(102) 및 하나 이상의 하드웨어 가속기(104), 지능형 메모리 패브릭(106), 주변 디바이스(108), 및 전력 관리 모듈(110)을 포함할 수 있다. 이 컴퓨팅 디바이스는, 하나 이상의 이미지 센서, 가속도계, 자이ロ스코프, 또는 기타 임의의 적절한 센서 디바이스 등의, 외부 디바이스와 통신할 수 있다.

대 표 도 - 도1



(52) CPC특허분류

Y02D 10/00 (2020.08)

(72) 발명자

코너, 퍼칼

아일랜드 엘에이치 던도크 월로우 그로브 69

몰로니, 데이비드

아일랜드 더블린 9 글라스네빈 아이오나 로드 42

(30) 우선권주장

14/458,052 2014년08월12일 미국(US)

14/458,014 2014년08월12일 미국(US)

명세서

청구범위

청구항 1

제1항에 따른 방법.

발명의 설명

기술분야

[0001] 관련 출원의 상호참조

본 출원은, 2013년 11월 6일 출원된 발명의 명칭이 "APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE"인 루마니아 특허 출원 OSIM 문서 제A/00812호와, 2013년 8월 8일 출원된 발명의 명칭이 "CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE"인 U.K 특허 출원 제GB1314263.3호에 대한 우선권을 주장하는, 2013년 11월 18일 출원된 발명의 명칭이 "APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING COMPUTATIONAL IMAGING PIPELINE"인 미국 특허 출원 제14/082,396호에 대한 우선권을 주장하며, 2014년 7월 30일 출원된 발명의 명칭이 "LOW POWER COMPUTATIONAL IMAGING COMPUTING DEVICE"인 미국 가출원 제62/030,913호에 대한 우선권을 주장하는, 2014년 8월 12일 출원된 발명의 명칭이 "LOW POWER COMPUTATIONAL IMAGING"인 미국 특허 출원 제14/458,014호의 더 이른 우선권 주장 날짜의 혜택을 주장한다. 본 출원은 또한, 2013년 11월 6일 출원된 발명의 명칭이 "APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE"인 루마니아 특허 출원 OSIM 문서 제A/00812호와, 2013년 8월 8일 출원된 발명의 명칭이 "CONFIGURABLE AND COMPOSABLE COMPUTATIONAL IMAGING PIPELINE"인 U.K 특허 출원 제GB1314263.3호에 대한 우선권을 주장하는, 2013년 11월 18일 출원된 발명의 명칭이 "APPARATUS, SYSTEMS, AND METHODS FOR PROVIDING COMPUTATIONAL IMAGING PIPELINE"인 미국 특허 출원 제14/082,396호에 대한 우선권을 주장하며, 2014년 7월 30일 출원된 발명의 명칭이 "LOW POWER COMPUTATIONAL IMAGING COMPUTING DEVICE"인 미국 가출원 제62/030,913호에 대한 우선권을 주장하는, 2014년 8월 12일 출원된 발명의 명칭이 "APPARATUS, SYSTEMS, AND METHODS FOR LOW POWER COMPUTATIONAL IMAGING"인 미국 특허 출원 제14/458,052호의 더 이른 우선권 주장 날짜의 혜택을 주장한다. 이들 출원들 각각은 그 전체가 참조로 본 명세서에 포함된다.

[0003] 기술분야

본 출원은 대체로 저전력 컴퓨테이셔널 이미징(computational imaging) 컴퓨팅 디바이스를 제공하는 것에 관한 것이다.

배경기술

[0005] 컴퓨테이셔널 이미징은, 이미지 및 비디오에 기초한 전례없는 사용자 경험 및 정보를 제공할 수 있는 새로운 이미징 패러다임이다. 예를 들어, 컴퓨테이셔널 이미징은, 이미지 및/또는 비디오를 처리하여 장면의 깊이 맵(depth map)을 제공하고, 장면의 파노라마 뷰를 제공하며, 이미지 및/또는 비디오로부터 얼굴을 추출하고, 이미지 및/또는 비디오로부터 텍스트, 피쳐, 및 메타 데이터를 추출하고, 심지어, 객체 및 장면 인식 피쳐에 기초하여 자동화된 시각적 인식 능력을 제공한다.

[0006] 컴퓨테이셔널 이미징은 흥미로운 능력을 제공할 수 있지만, 널리 채택되지는 못했다. 컴퓨테이셔널 이미징의 느린 채택은, 컴퓨테이셔널 이미징에는 기본적인 데이터 처리 해결과제가 있다는 사실에 기인할 수 있다. 종종, 이미지 해상도와 비디오 프레임 레이트는 높다. 따라서, 컴퓨테이셔널 이미징은 일반적으로, 특히 저전력 소모시 높은 메모리 대역폭에 의해 성능이 유지 및 백업될 수 있어야 하는 경우, 일반 컴퓨터 프로세서를 이용하여 획득하기에 어려울 수 있는 수백 기가플롭의 연산 자원을 요구한다. 또한, 컴퓨테이셔널 이미징은 일반적으로 레이턴시에 민감하다. 사용자는, 카메라가 물체를 인식하도록 몇 분을 기다릴 가능성이 없으므로, 컴퓨테이셔널 이미징 카메라는 일반적으로 이미지와 비디오를 신속하게 처리할 수 있도록 설계되고, 이것은 컴퓨테이셔널 이미징의 계산 요구사항을 더욱 부담스럽게 한다.

- [0007] 불행히도, 맞춤화된 하드웨어에서 컴퓨터이셔널 이미징 기술을 구현하는 것은 어렵다. 컴퓨터이셔널 이미징 분야는 비교적 초기 단계이기 때문에, 구현 기술은 지속적으로 유통적이다. 따라서, 구현 기술을 변경하려면 전체 하드웨어를 다시 설계해야 하므로 컴퓨터이셔널 이미징을 완전히 하드웨어로 맞춤화하는 것은 어렵다. 따라서, 일반적으로 유연한 하드웨어 아키텍처 및 유연한 하드웨어 인프라구조를 제공하는 것이 바람직하다.
- [0008] 동시에, 이러한 비디오 및 이미지 처리에 대한 수요는, 전력 소비가 핵심 고려사항인 휴대형 전자 디바이스, 예를 들어 태블릿 컴퓨터 및 모바일 디바이스로부터 상당히 커지고 있다. 그 결과, 제한된 전력 예산 하에서도 동작할 수 있는 유연한 컴퓨터이셔널 이미징 인프라구조가 일반적으로 필요하다.
- [0009] 개시된 주제에 따르면, 저전력 컴퓨터이셔널 이미징을 제공하기 위한 시스템 및 방법이 제공된다.
- [0010] 개시된 주제는 컴퓨팅 디바이스를 포함한다. 컴퓨팅 디바이스는 복수의 벡터 프로세서를 포함할 수 있으며, 여기서, 복수의 벡터 프로세서 중 하나는 제1 어레이의 값들에 관해 연산하는 명령어를 실행하도록 구성된다. 컴퓨팅 디바이스는 또한, 제2 어레이의 값들에 관한 필터링 연산을 수행하도록 구성된 하드웨어 가속기를 포함할 수 있다. 또한, 컴퓨팅 디바이스는, 복수의 메모리 슬라이스를 포함하는 메모리 패브릭과 제1 인터페이스 및 제2 인터페이스를 갖는 인터커넥트 시스템을 포함할 수 있으며, 여기서, 제1 인터페이스는 복수의 벡터 프로세서를 상기 다수의 메모리 슬라이스에 결합하도록 구성되며, 제2 인터페이스는 하드웨어 가속기를 복수의 메모리 슬라이스에 결합하도록 구성된다. 또한, 컴퓨팅 디바이스는, 메모리 패브릭으로 하여금 제1 인터페이스를 통해 복수의 벡터 프로세서 중 하나에 제1 어레이의 값들을 제공하게 하고, 제2 인터페이스를 통해 하드웨어 가속기에 제2 어레이의 값들을 제공하게 함으로써, 복수의 벡터 프로세서 중 하나가 명령어에 따라 제1 어레이의 값들을 처리하고 하드웨어 가속기가 필터링 연산에 따라 제2 어레이의 값들을 처리할 수 있게 하도록 구성된 호스트 프로세서를 포함할 수 있다.
- [0011] 일부 실시예에서, 컴퓨팅 디바이스는 각각이 적어도 하나의 전력 도메인을 포함하는 복수의 전력 아일랜드 (power island)를 포함할 수 있고, 여기서, 복수의 전력 아일랜드 중 제1 전력 아일랜드는 제1 공급 전압에 결합되어 복수의 벡터 프로세서 중 하나에 제1 공급 전압을 제공하고, 복수의 전력 아일랜드 중 제2 전력 아일랜드는 제2 공급 전압에 결합되어 하드웨어 가속기에 제2 공급 전압을 제공한다.
- [0012] 일부 실시예에서, 컴퓨팅 디바이스는 복수의 전력 아일랜드 중 제1 전력 아일랜드를 제1 공급 전압에 결합하는 스위치에 인에이블 신호를 제공함으로써, 복수의 벡터 프로세서 중 하나를 활성 모드(active mode)에 두도록 구성된 전력 관리 모듈을 포함할 수 있다.
- [0013] 일부 실시예에서, 복수의 벡터 프로세서 중 하나는, 제1 어레이의 값들을 처리하기 위한 로직 회로 영역 및 제1 어레이의 값들의 적어도 서브셋을 저장하기 위한 로컬 메모리를 포함할 수 있고, 전력 관리 모듈은, 제1 공급 전압이 로직 회로 영역에 제공되게 하고 제3 공급 전압이 로컬 메모리에 제공되게 하여 로직 회로 영역 및 로컬 메모리의 전력 소비를 독립적으로 제어하도록 구성될 수 있다.
- [0014] 일부 실시예들에서, 전력 관리 모듈은 스위치를 턠오프하여 제1 공급 전압으로부터 복수의 전력 아일랜드 중 제1 전력 아일랜드를 제1 공급전압으로부터 접속해제함으로써, 복수의 벡터 프로세서 중 하나를 저전력 모드에 두도록 구성될 수 있다.
- [0015] 일부 실시예에서, 전력 관리 모듈은 유효 신호를 생성하도록 구성된 유효 신호 생성기를 포함할 수 있고, 이 유효 신호는 복수의 전력 아일랜드 중 제1 전력 아일랜드의 회로 블록이 입력 데이터를 처리할 준비가 되어 있는 시간 인스턴스(time instance)를 나타내고, 유효 신호 생성기는 복수의 전력 아일랜드 중 제1 전력 아일랜드 내의 회로 블록에 제1 공급 전압을 제공하는 스위치들의 테이지 체인(daisy chain)을 포함한다.
- [0016] 일부 실시예에서, 컴퓨팅 디바이스는 복수의 입력/출력(I/O) 핀에 결합된 주변 디바이스를 포함할 수 있고, 여기서, 주변 디바이스는 복수의 벡터 프로세서 중 적어도 하나와 외부 디바이스 사이에 통신 채널을 제공하도록 구성된다.
- [0017] 일부 실시예에서, 주변 디바이스는 항상 전원이 켜진 전력 아일랜드 내에 있을 수 있다.
- [0018] 일부 실시예에서, 주변 디바이스는, 외부 디바이스로부터의 신호를 모니터링하여 복수의 벡터 프로세서 중 하나가 응답해야 하는 이벤트를 검출하고, 주변 디바이스가 이벤트를 검출할 때, 전력 관리 모듈로 하여금 복수의 벡터 프로세서 중 하나를 활성 모드로 두게 하도록 구성된다.
- [0019] 일부 실시예에서, 주변 디바이스는, 주변 디바이스로 하여금 I/O 핀들의 공통 세트를 통해 복수의 표준 프로토

콜 인터페이스의 기능을 에뮬레이트하게 하도록 구성된 에뮬레이션 모듈을 포함할 수 있다.

- [0020] 일부 실시예에서, 주변 디바이스는 I/O 핀들의 차동 쌍에 결합될 수 있고, 주변 디바이스는 극성 제어 신호에 기초하여 차동 쌍의 극성을 변경하도록 구성된다.
- [0021] 일부 실시예에서, I/O 핀들의 차동 쌍은 모바일 산업 프로세서 인터페이스(MIPI; Mobile Industry Processor Interface) 레인(lane)들의 차동 쌍을 포함할 수 있다.
- [0022] 일부 실시예에서, 주변 디바이스는, 입력 I/O 핀과 출력 I/O 핀 사이에서 바이패스를 수행함으로써, 벡터 프로세서 중 하나를 활성 모드로 두지 않고도 입력 I/O 핀과 출력 I/O 핀 사이에 통신 채널을 제공하도록 구성된 바이패스 버퍼를 포함할 수 있다.
- [0023] 개시된 주제는 방법을 포함한다. 이 방법은, 복수의 메모리 슬라이스를 포함하는 메모리 패브릭과 제1 인터페이스 및 제2 인터페이스를 갖는 인터커넥트 시스템을 제공하는 단계를 포함할 수 있다. 이 방법은 또한, 제1 인터페이스를 이용하여, 복수의 메모리 슬라이스와 복수의 벡터 프로세서를 결합하고, 제2 인터페이스를 이용하여, 복수의 메모리 슬라이스 및 하드웨어 가속기를 결합하는 단계를 포함할 수 있다. 이 방법은, 메모리 패브릭에 의해, 제1 인터페이스를 통해 복수의 벡터 프로세서 중 하나에 제1 어레이의 값들을 제공하는 단계, 제2 인터페이스를 통해 하드웨어 가속기에 제2 어레이의 값들을 제공하는 단계, 복수의 벡터 프로세서 중 하나에서, 제1 어레이의 값들에 관해 연산하는 명령어를 실행하는 단계, 및 하드웨어 가속기에 의해 제2 어레이의 값들에 관해 필터링 연산을 수행하는 단계를 포함한다.
- [0024] 일부 실시예에서, 이 방법은, 복수의 벡터 프로세서 중 하나에 제1 공급 전압을 제공하는 단계, 및 하드웨어 가속기에 제2 공급 전압을 제공하는 단계를 포함할 수 있고, 여기서, 복수의 벡터 프로세서 중 하나와 하드웨어 가속기는 각각 제1 전력 아일랜드 및 제2 전력 아일랜드와 연관된다.
- [0025] 일부 실시예에서, 이 방법은, 전력 관리 모듈에 의해, 제1 전력 아일랜드를 제1 공급 전압에 결합하는 스위치에 인에이블 신호를 제공함으로써, 복수의 벡터 프로세서 중 하나를 활성 모드에 두는 단계를 포함할 수 있다.
- [0026] 일부 실시예에서, 이 방법은, 복수의 벡터 프로세서 중 하나 내의 회로 블록에 제1 공급 전압을 제공하는 스위치들의 데이지 체인을 이용하여, 제1 전력 아일랜드 내의 회로 블록이 입력 데이터를 처리할 준비가 되어 있는 시간 인스턴스를 나타내는 유효 신호를 생성하는 단계를 포함할 수 있다.
- [0027] 일부 실시예에서, 이 방법은 복수의 입력/출력(I/O) 핀에 결합된 주변 디바이스를 제공하는 단계를 포함할 수 있고, 여기서, 주변 디바이스는 항상 전원이 켜진 전력 아일랜드와 연관된다.
- [0028] 일부 실시예에서, 이 방법은, 외부 디바이스로부터의 신호를 모니터링하여 복수의 벡터 프로세서 중 하나가 응답해야 하는 이벤트를 검출하고, 전력 관리 모듈로 하여금 복수의 벡터 프로세서 중 하나를 활성 모드에 두게 하는 단계를 포함할 수 있다.
- [0029] 일부 실시예에서, 이 방법은, 주변 디바이스에 의해, I/O 핀들의 공통 세트를 통해 복수의 표준 프로토콜 인터페이스의 기능을 에뮬레이트하는 단계를 포함할 수 있다.
- [0030] 일부 실시예에서, 주변 디바이스는 I/O 핀들의 차동 쌍에 결합되고, 이 방법은 극성 제어 신호에 기초하여 차동 쌍의 극성을 변경하는 단계를 더 포함한다.
- [0031] 일부 실시예에서, 이 방법은, 바이패스 버퍼를 이용하여 입력 I/O 핀과 출력 I/O 핀 사이에서 바이패스를 수행함으로써, 벡터 프로세서 중 하나를 활성 모드로 두지 않고도 입력 I/O 핀과 출력 I/O 핀 사이에 통신 채널을 제공하는 단계를 포함할 수 있다.
- [0032] 개시된 주제는 전자 디바이스를 포함한다. 전자 디바이스는 복수의 벡터 프로세서를 포함할 수 있고, 여기서, 복수의 벡터 프로세서 중 하나는 제1 어레이의 값들에 관해 연산하는 명령어를 실행하도록 구성된다. 전자 디바이스는 또한, 소프트웨어 모듈로부터 수신된 구성 정보를 이용하여 프로그램되는 프로그램가능한 데이터경로 파이프라인을 포함하는 하드웨어 가속기를 포함할 수 있고, 여기서, 프로그램가능한 데이터경로 파이프라인은 구성 정보에 따라 제2 어레이의 값들에 관한 필터링 연산을 수행하도록 구성된다. 전자 디바이스는 또한, 복수의 메모리 슬라이스를 포함하는 메모리 패브릭을 포함할 수 있다. 전자 디바이스는, 메모리 패브릭으로 하여금 복수의 벡터 프로세서 중 하나에 제1 어레이의 값들을 제공하게 하고, 하드웨어 가속기에 제2 어레이의 값들을 제공하게 함으로써, 복수의 벡터 프로세서 중 하나가 명령어에 따라 제1 어레이의 값들을 처리하고 하드웨어 가속기가 구성 정보에 따라 제2 어레이의 값들을 처리할 수 있게 하도록 구성된 호스트 프로세서를 더 포함할 수

있다.

- [0033] 일부 실시예에서, 하드웨어 가속기는, 프로그램가능한 데이터경로 파이프라인에 의해 처리된 이미지의 스캔-라인(scan-line)을 수신하기 위한 출력 버퍼와, 출력 버퍼가 가득 찰 때, 프로그램가능한 데이터 경로 파이프라인의 동작을 정지시키도록 구성된 파이프라인 정지 제어기를 포함할 수 있다.
- [0034] 일부 실시예에서, 하드웨어 가속기는 필터링 연산을 수행하기 위해 함께 체인화되는 복수의 기능 유닛을 포함할 수 있다.
- [0035] 일부 실시예에서, 복수의 기능 유닛이 함께 체인화되는 순서는 소프트웨어 모듈로부터 수신된 구성 정보를 이용하여 결정된다.
- [0036] 일부 실시예에서, 복수의 기능 유닛 중 제1 기능 유닛의 출력은 메모리 패브릭 내의 버퍼에 제공되고, 복수의 기능 유닛 중 제2 기능 유닛의 입력은 버퍼로부터 수신된다.
- [0037] 일부 실시예에서, 하드웨어 가속기는, 이미지의 스캔-라인 내의 팩셀에 의해 표현된 객체의 깊이를 나타내는 깊이 정보를 수신하도록 구성된 깊이 맵 클라이언트를 포함할 수 있다.
- [0038] 일부 실시예에서, 하드웨어 가속기는, 깊이 정보의 해상도를 이미지의 스캔-라인의 해상도와 정합시키기 위해 깊이 정보를 처리하도록 구성된 깊이 맵 모듈을 포함할 수 있다.
- [0039] 일부 실시예에서, 깊이 맵 모듈은 깊이 정보를 이미지의 스캔-라인에 시간-동기화하도록 구성된다.
- [0040] 일부 실시예에서, 메모리 패브릭은, 벡터 프로세서들 중 하나에 의한 공유 자원에 대한 배타적 액세스를 요청하는 배타적 액세스 요청의 상태를 모니터링하고, 벡터 프로세서들 중 하나가 공유 자원에 대한 배타적 액세스를 수신할 때, 벡터 프로세서들 중 하나가 공유 자원에 대한 배타적 액세스를 갖는다는 것을 나타내는 확인응답 메시지를 벡터 프로세서들 중 하나에 전송하도록 구성된 상호-배타(뮤텍스) 제어기를 포함할 수 있다.
- [0041] 일부 실시예에서, 메모리 패브릭은 복수의 버퍼를 포함할 수 있고, 여기서, 복수의 버퍼 중 제1 버퍼는 벡터 프로세서들 중 제1 벡터 프로세서와 연관되고, 벡터 프로세서들 중 제2 벡터 프로세서는, 복수의 버퍼 중 제1 버퍼에 데이터를 저장함으로써, 데이터를 벡터 프로세서들 중 제1 벡터 프로세서에 전송하도록 구성된다.
- [0042] 일부 실시예에서, 메모리 패브릭은 벡터 프로세서들 중 제1 벡터 프로세서에 전달되는 데이터의 양에 기초하여 복수의 버퍼 중 제1 버퍼의 용량을 동적으로 변경하도록 구성될 수 있다.
- [0043] 일부 실시예에서, 메모리 패브릭은 벡터 프로세서들 중 제1 벡터 프로세서에 전달되는 데이터의 양에 기초하여 복수의 버퍼 중 2개 이상을 벡터 프로세서들 중 제1 벡터 프로세서와 동적으로 연관시키도록 구성될 수 있다.
- [0044] 일부 실시예에서, 복수의 버퍼는 메모리 패브릭 내의 복수의 메모리 슬라이스 중 하나의 일부일 수 있다.
- [0045] 일부 실시예에서, 메모리 패브릭은 벡터 프로세서들 중 하나가 저전력 모드에 진입할 때, 벡터 프로세서들 중 하나의 상태 정보를 저장하도록 구성될 수 있다.
- [0046] 일부 실시예에서, 상태 정보는 메모리 패브릭 내의 정적 랜덤 액세스 메모리에 저장된다.
- [0047] 일부 실시예에서, 메모리 패브릭은 직접 메모리 액세스(DMA; direct memory access) 제어기를 포함할 수 있고, DMA 제어기는 DMA 동작이 수행될 순서를 나타내는 동작 리스트(operation list)를 포함한다.
- [0048] 일부 실시예에서, DMA 제어기는 인에이블 버퍼에 기초하여 동작 리스트 내의 DMA 동작들의 서브셋을 수행하도록 구성될 수 있고, 여기서, 인에이블 버퍼는 복수의 비트를 포함하며, 복수의 비트 중 하나는 DMA 동작들 중 하나와 연관되고 복수의 비트 중 하나의 값은 DMA 동작들 중 하나가 DMA 제어기에 의해 수행되어야 하는지를 나타낸다.
- [0049] 개시된 주제는 방법을 포함한다. 이 방법은, 복수의 메모리 슬라이스를 포함하는 메모리 패브릭에 의해, 제1 어레이의 값들을 복수의 벡터 프로세서 중 하나에 제공하는 단계를 포함할 수 있다. 이 방법은 또한, 메모리 패브릭에 의해, 프로그램가능한 데이터경로 파이프라인을 포함하는 하드웨어 가속기에 제2 어레이의 값들을 제공하는 단계; 복수의 벡터 프로세서 중 하나에 의해, 제1 어레이의 값들에 관해 연산하는 명령어를 실행하는 단계; 구성 정보를 이용하여 하드웨어 가속기 내의 데이터경로 파이프라인을 구성하는 단계; 및 하드웨어 가속기 내의 데이터경로 파이프라인을 이용하여, 구성 정보에 따라 제2 어레이의 값들에 관한 필터링 연산을 수행하는 단계를 포함한다.

- [0050] 일부 실시예에서, 이 방법은, 출력 버퍼에서, 프로그램가능한 데이터경로 파이프라인에 의해 처리된 이미지의 스캔-라인을 수신하는 단계; 및 출력 버퍼가 가득찰 때, 파이프라인 정지 제어기에 의해, 프로그램가능한 데이터경로 파이프라인의 동작을 정지시키는 단계를 포함할 수 있다.
- [0051] 일부 실시예에서, 하드웨어 가속기는 복수의 기능 유닛을 포함하고, 이 방법은 필터링 연산을 수행하기 위해 구성 정보에 따라 복수의 기능 유닛을 체인화하는 단계를 포함한다.
- [0052] 일부 실시예에서, 복수의 기능 유닛은 제1 기능 유닛 및 제2 기능 유닛을 포함하고, 여기서, 복수의 기능 유닛을 체인화하는 단계는 제2 기능 유닛의 입력으로의 제1 기능 유닛의 출력을 포함한다.
- [0053] 일부 실시예에서, 이 방법은, 이미지의 스캔-라인 내의 픽셀에 의해 표현된 객체의 깊이를 나타내는 깊이 정보를 수신하는 단계; 및 깊이 정보를 이미지의 스캔-라인에 동기화시키는 단계를 포함할 수 있다.
- [0054] 일부 실시예에서, 이 방법은, 메모리 패브릭 내의 메모리 제어기에 의해, 벡터 프로세서들 중 하나에 의한 공유 자원에 대한 배타적 액세스를 요청하는 배타적 액세스 요청의 상태를 모니터링하는 단계, 및 벡터 프로세서들 중 하나가 공유 자원에 대한 배타적 액세스를 수신할 때, 벡터 프로세서들 중 하나가 공유 자원에 대한 배타적 액세스를 갖는다는 것을 나타내는 확인응답 메시지를 벡터 프로세서들 중 하나에 전송하는 단계를 포함할 수 있다.
- [0055] 일부 실시예에서, 메모리 패브릭은 복수의 버퍼를 포함할 수 있고, 여기서, 복수의 버퍼 중 제1 버퍼는 벡터 프로세서들 중 제1 벡터 프로세서와 연관되며, 이 방법은, 벡터 프로세서들 중 제2 벡터 프로세서에 의해, 데이터를 복수의 버퍼 중 제1 버퍼에 저장함으로써 데이터를 벡터 프로세서들 중 제1 벡터 프로세서에 전송하는 단계를 더 포함할 수 있다.
- [0056] 일부 실시예에서, 이 방법은, 벡터 프로세서들 중 제1 벡터 프로세서에 전달되는 데이터의 양에 기초하여 복수의 버퍼 중 제1 버퍼의 용량을 동적으로 변경하는 단계를 포함할 수 있다.
- [0057] 일부 실시예에서, 이 방법은, 벡터 프로세서들 중 제1 벡터 프로세서에 전달되는 데이터의 양에 기초하여 복수의 버퍼 중 2개 이상을 벡터 프로세서들 중 제1 벡터 프로세서와 동적으로 연관시키는 단계를 포함할 수 있다.
- [0058] 일부 실시예에서, 이 방법은, 벡터 프로세서들 중 하나가 저전력 모드에 진입할 때, 벡터 프로세서들 중 하나의 상태 정보를 메모리 패브릭에 저장하는 단계를 포함할 수 있다.
- [0059] 일부 실시예에서, 상태 정보는 메모리 패브릭 내의 정적 랜덤 액세스 메모리에 저장된다.
- [0060] 일부 실시예에서, 이 방법은, 직접 메모리 액세스(DMA) 제어기에서, DMA 동작들이 수행되는 순서를 나타내는 동작 리스트를 유지하는 단계를 포함할 수 있다.
- [0061] 일부 실시예에서, 인에이블 버퍼에 기초하여 동작 리스트 내의 DMA 동작들의 서브셋을 수행하는 단계를 포함할 수 있고, 여기서, 인에이블 버퍼는 복수의 비트를 포함하며, 복수의 비트 중 하나는 DMA 동작들 중 하나와 연관되고, 복수의 비트 중 하나의 값은 DMA 동작들 중 하나가 DMA 제어기에 의해 수행되어야 하는지를 나타낸다.

도면의 간단한 설명

- [0062] 개시된 주제의 다양한 목적, 피쳐, 및 이점들은, 유사한 참조 번호가 유사한 요소를 식별하는 이하의 도면들과 연계하여 개시된 주제의 이하의 상세한 설명을 참조하면 더욱 충분히 이해될 수 있다. 첨부된 도면들은 개략도이며 축척비율대로 그려진 것은 아니다. 명료성을 위해, 모든 도면에서 모든 컴포넌트가 라벨링되지는 않았다. 또한, 본 기술분야의 통상의 기술자가 개시된 주제를 이해할 수 있도록 하기 위해 예시가 필요한 것이 아닌 경우에는 개시된 주제의 각각의 실시예의 모든 컴포넌트가 도시된 것은 아니다.
- 도 1은 일부 실시예에 따른 컴퓨팅 디바이스의 고레벨 예시를 제공한다.
- 도 2는 일부 실시예에 따른 컴퓨팅 디바이스의 상세한 예시를 나타낸다.
- 도 3은 일부 실시예에 따른 하드웨어 가속기를 나타낸다.
- 도 4는 일부 실시예에 따른 깊이 정보에 기초하여 필터링 연산을 적합화할 수 있는 하드웨어 가속기를 나타낸다.

도 5는 하드웨어 가속기를 나타낸다.

도 6은 일부 실시예에 따른 일반적인 기능에 기초한 하드웨어 가속기를 나타낸다.

도 7은 일부 실시예에 따른 이미지 신호 처리(ISP; image signal processing) 기능 모듈들 사이의 통신을 위한 FIFO(first-in-first-out) 버퍼를 포함하는 하드웨어 가속기를 나타낸다.

도 8은 일부 실시예에 따른 전력 아일랜드의 전원 케이팅을 나타낸다.

도 9는 일부 실시예에 따른 유효 신호 생성기를 나타낸다.

도 10은 일부 실시예에 따른 이벤트 신호 모니터링 메커니즘을 나타낸다.

도 11은 일부 실시예에 따른 소프트웨어 정의된 인터페이스를 도시한다.

도 12는 일부 실시예에 따른 소프트웨어 정의된 인터페이스의 상세한 구현을 도시한다.

도 13은 일부 실시예에 따른 이벤트 프로세서를 나타낸다.

도 14는 일부 실시예에 따른 이벤트 프로세서 내의 이벤트 필터를 나타낸다.

도 15는 일부 실시예에 따른 주변 디바이스의 바이패스 모드를 도시한다.

도 16은 일부 실시예에 따른 프로그램 가능한 모바일 산업 프로세서 인터페이스(MIPI) 인터페이스를 도시한다.

도 17은 일부 실시예에 따른 입/출력 인터페이스를 위한 극성 반전 메커니즘의 적용을 나타낸다.

도 18은 일부 실시예에 따른 하드웨어-기반 상호 배타(뮤텍스) 제어기를 갖는 메모리 패브릭을 나타낸다.

도 19는 일부 실시예에 따른 버퍼의 동적 할당을 나타낸다.

도 20은 일부 실시예에 따른 로직 회로 메모리 디바이스들에 상이한 전압들을 제공하는 전력 관리 메커니즘을 나타낸다.

도 21은 일부 실시예에 따른 버퍼-기반 DMA 데이터 구조 인에이블 메커니즘을 구현하는 직접 메모리 액세스(DMA) 엔진을 나타낸다.

도 22는 일부 실시예에 따른 컴퓨팅 디바이스를 포함하는 전자 디바이스를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0063]

이하의 설명에서, 개시된 주제의 철저한 이해를 제공하기 위하여, 개시된 주제의 시스템 및 방법과, 이러한 시스템 및 방법이 동작할 수 있는 환경 등에 관하여 수 많은 특정한 상세사항이 개시된다. 그러나, 본 기술분야의 통상의 기술자라면, 개시된 주제가 이러한 특정한 상세사항 없이도 실시될 수 있고, 본 기술분야의 널리 공지되어 있는 소정의 피쳐들은 개시된 주제의 복잡성을 피하기 위하여 상세히 설명되지 않는다는 것이 명백할 것이다. 또한, 이하에 제공되는 예들은 예시적인 것이며, 개시된 주제의 범위 내에 있는 다른 시스템 및 방법들이 있다고 여겨진다.

[0064]

컴퓨테이셔널 이미징은 머신이 물리적 세계를 포착하고 상호작용하는 방식을 변형시킬 수 있다. 예를 들어, 컴퓨테이셔널 이미징을 통해, 머신은 전통적인 이미징 기술을 이용하여 캡처하기가 매우 어려웠던 이미지를 포착할 수 있다. 또 다른 예로서, 컴퓨테이셔널 이미징을 통해, 머신은 주변을 이해하고 주변 환경에 따라 반응할 수 있다.

[0065]

컴퓨테이셔널 이미징을 대중 시장에 가져 오는 문제점들 중 하나는, 컴퓨테이셔널 이미징이 본질적으로 계산적으로 비싸다는 것이다. 컴퓨테이셔널 이미징은 종종, 고해상도의 많은 수의 이미지를 이용하거나 및/또는 높은 프레임 레이트를 갖는 많은 수의 비디오를 이용한다. 따라서, 컴퓨테이셔널 이미징은 종종 강력한 컴퓨팅 플랫폼을 지원할 필요가 있다. 또한, 컴퓨테이셔널 이미징은 종종, 예를 들어, 스마트 폰이나 태블릿 컴퓨터를 이용하여 모바일 설정에서 이용되기 때문에, 컴퓨팅 이미징은 종종, 저전력 예산에서 동작할 수 있는 강력한 컴퓨팅 플랫폼을 지원할 필요가 있다.

[0066]

본 출원은, 컴퓨테이셔널 이미징을 위한 저전력, 고성능 컴퓨팅 플랫폼을 제공할 수 있는 컴퓨팅 디바이스를 개시한다. 도 1은 일부 실시예에 따른 컴퓨팅 디바이스의 고레벨 예시를 제공한다. 컴퓨팅 디바이스(100)는, 하나 이상의 처리 유닛, 예를 들어, 하나 이상의 벡터 프로세서(102) 및 하나 이상의 하드웨어 가속기(104), 지능

형 메모리 패브릭(106), 주변 디바이스(108), 및 전력 관리 모듈(110)을 포함할 수 있다.

[0067] 하나 이상의 벡터 프로세서(102)는, 벡터라고 불리는 데이터의 어레이에 관해 연산하는 명령어들을 포함하는 명령어 세트를 구현하는 중앙 처리 유닛(CPU)을 포함한다. 더 구체적으로는, 하나 이상의 벡터 프로세서(102)는 대량의 데이터에 관해 일반적인 산술 연산을 동시에 수행하도록 구성될 수 있다. 일부 실시예에서, 하나 이상의 벡터 프로세서(102)는, 단일 명령어 다중 데이터, 매우 긴 명령어 워드(SIMD-VLIW; single instruction multiple data, very long instruction word) 프로세서를 포함할 수 있다. 일부 실시예에서, 하나 이상의 벡터 프로세서(102)는 컴퓨터 비전 및 이미징 애플리케이션과 연관된 명령어들을 실행하도록 설계될 수 있다.

[0068] 하나 이상의 하드웨어 가속기(104)는, 더욱 범용적인 CPU 상에서 실행되는 소프트웨어에서 가능한 것보다 빠르게 일부 기능을 수행하는 컴퓨터 하드웨어를 포함한다. 비-시각 애플리케이션의 하드웨어 가속기의 예로서는, 래스터 오퍼레이터를 이용하여 수 개의 비트 맵을 하나로 결합하도록 구성된 그래픽 처리 유닛(GPU) 내의 블리팅 가속 모듈(blitting acceleration module)이 포함된다.

[0069] 일부 실시예에서, 하나 이상의 하드웨어 가속기(104)는, 이미지 처리 및 컴퓨터 비전 애플리케이션에 맞추어진 구성가능한 인프라구조를 제공할 수 있다. 하드웨어 가속기(104)는, 애플리케이션-특유의 계산 코어를 둘러싸는 이미지 처리 및 컴퓨터 비전 동작을 가속하기 위한 일반 래퍼 하드웨어(wrapper hardware)를 포함하는 것으로 간주될 수 있다. 예를 들어, 하드웨어 가속기(104)는 이미지 필터링 연산을 수행하기 위한 전용 필터링 모듈을 포함할 수 있다. 필터링 모듈은 효율적인 방식으로 이미지에 관해 맞춤화된 필터 커널을 동작시키도록 구성될 수 있다. 일부 실시예에서, 하드웨어 가속기(104)는, 클록 사이클 당 하나의 완전히 계산된 출력 핵심을 출력할 수 있다.

[0070] 지능형 메모리 패브릭(106)은, 작은 레이턴시를 갖는 저전력 메모리 시스템을 제공하도록 구성될 수 있다. 이미지와 비디오는 많은 양의 데이터를 포함하기 때문에, 메모리와 처리 유닛 사이에 고속 인터페이스를 제공하는 것이 중요하다. 일부 실시예에서, 지능형 메모리 패브릭(106)은, 예를 들어, 64 블록의 메모리를 포함할 수 있고, 블록들 각각은 64 비트 인터페이스를 포함할 수 있다. 이러한 실시예에서, 예를 들어, 600MHz에서 동작하는 메모리 패브릭(106)은 307.2 GB/sec의 속도로 데이터를 전송할 수 있다. 다른 실시예에서, 지능형 메모리 패브릭(106)은 기타의 임의의 개수의 메모리 블록을 포함할 수 있고, 메모리 블록들 각각은 하나 이상의 인터페이스 프로토콜을 구현하는 임의의 개수의 인터페이스를 포함할 수 있다.

[0071] 주변 디바이스(108)는, 이미지 센서 및 가속도계 등의 외부 디바이스로/로부터 데이터 비트를 전송 및 수신하기 위한 통신 채널을 제공하도록 구성될 수 있다. 주변 디바이스(108)는, 벡터 프로세서(102), 하드웨어 가속기(104), 및 메모리 패브릭(106)이 외부 디바이스와 통신하기 위한 통신 메커니즘을 제공할 수 있다.

[0072] 전력 관리 모듈(110)은 컴퓨팅 디바이스(100) 내의 지정된 블록의 활동을 제어하도록 구성될 수 있다. 더 구체적으로는, 전력 관리 모듈(110)은, 컴퓨팅 디바이스(100) 내에서 전력 아일랜드(power island)라고도 지칭되는, 지정된 블록들의 전원 전압을 제어하도록 구성될 수 있다. 예를 들어, 전력 관리 모듈(110)이 전력 아일랜드의 전원을 인에이블 할 때, 컴퓨팅 디바이스(100)는 적절한 전원 전압을 전력 아일랜드에 제공하도록 트리거될 수 있다. 일부 실시예에서, 각각의 전력 아일랜드는 독립적인 전력 도메인을 포함할 수 있다. 따라서, 전력 아일랜드의 전원은 독립적으로 제어될 수 있다. 일부 실시예에서, 전력 관리 모듈(110)은 또한, 컴퓨팅 디바이스(100)의 하나 이상의 입력/출력 핀을 통해 컴퓨팅 디바이스(100)에 외부적으로 부착된 전력 아일랜드의 활동을 제어하도록 구성될 수 있다.

[0073] 도 2는 일부 실시예에 따른 컴퓨팅 디바이스의 상세한 예시를 나타낸다. 컴퓨팅 디바이스(100)는 복수의 벡터 프로세서(102)를 포함할 수 있다. 이 예시에서, 컴퓨팅 디바이스(100)는 12개의 벡터 프로세서(102)를 포함한다. 벡터 프로세서(102)는 프로세서간 인터커넥트(IPI; inter-processor interconnect)(202)를 통해 서로 통신할 수 있다. 벡터 프로세서(102)는, IPI(202) 및 가속기 메모리 제어기(AMC; Accelerator Memory Controller) 크로스바(204) 또는 메모리-맵핑된 프로세서 버스(208)를 통해, 메모리 패브릭(106) 및/또는 하드웨어 가속기(104)를 포함하는 컴퓨팅 디바이스(100) 내의 다른 컴퓨팅 디바이스들과도 통신할 수 있다.

[0074] 일부 실시예에서, 하나 이상의 벡터 프로세서(102)는 전용 명령어 세트를 실행하도록 설계될 수 있다. 전용 명령어 세트는 전용 명령어를 포함할 수 있다. 전용 명령어는, 명령어 헤더 및 하나 이상의 유닛 명령어를 포함하는 가변 길이 2진 문자열일 수 있다. 명령어 헤더는, 연관된 전용 명령어에 대한 명령어 길이 및 활성 유닛에 대한 정보를 포함할 수 있다; 유닛 명령어는 고정된 또는 가변적인 필드 개수를 포함하는 가변 길이 2진 문자열일 수 있다. 유닛 명령어의 필드는, 명령어를 식별하는 opcode와 유닛 명령어 실행에서 값 이용을 명시하

는 피연산자(operand)를 포함할 수 있다.

[0075] 벡터 프로세서(102)의 상세사항은, 참조로 그 전체가 본 명세서에 포함되는, 본 명세서와 동일한 날짜에 출원된, 변리사 도켓 번호 제2209599.127US1호로 식별되는, 발명의 명칭이 "VECTOR PROCESSOR"인 미국 특허 출원(출원번호 아직 미정)에서 제공된다.

[0076] 컴퓨팅 디바이스(100)는 하드웨어 가속기(104)를 포함할 수 있다. 하드웨어 가속기(104)는 미리정의된 처리 기능을 수행하도록 구성된 다양한 가속기 모듈을 포함할 수 있다. 일부 실시예에서, 미리정의된 처리 기능은 필터링 연산을 포함할 수 있다. 예를 들어, 하드웨어 가속기(104)는, 원시 이미지 처리 모듈, 렌즈 음영 보정(LSC; lens shading correction) 모듈, 베이어 패턴 디모자이킹 모듈(bayer pattern demosaicing module), 샤픈 필터 모듈(sharpen filter module), 다상 스케일러 모듈, 해리스 코너 검출 모듈(Harris corner detection module), 컬러 조합 모듈, 루마 채널 노이즈제거 모듈(luma channel denoise module), 크로마 채널 노이즈제거 모듈(chroma channel denoise module), 메디언 필터 모듈(median filter module), 륙업 테이블, 콘볼루션 모듈, 엣지 검출 모듈, 및/또는 기타 임의의 적절한 모듈 또는 모듈들의 조합을 포함할 수 있다. 하드웨어 가속기(104)는 메모리 패브릭(106)에 존재하는 메모리 디바이스 내의 데이터를 회수 및 저장하도록 구성될 수 있다.

[0077] 메모리 패브릭(106)은, 컴퓨팅 디바이스(100) 내의 메모리 동작을 조율하는 중앙 메모리 시스템을 포함할 수 있다. 메모리 패브릭(106)은, 벡터 프로세서(102) 및 하드웨어 가속기(104) 등의, 처리 유닛들 사이의 불필요한 데이터 전송을 감소시키도록 설계될 수 있다. 메모리 패브릭(106)은, 복수의 처리 유닛이 정지없이 데이터 및 프로그램 코드 메모리에 병렬로 액세스하는 것을 허용하도록 구성된다. 추가로, 메모리 패브릭(106)은, AXI(Advanced extensible interface) 또는 기타 임의의 적절한 버스(208) 등의 병렬 버스를 통해 호스트 프로세서가 메모리 패브릭(106) 내의 메모리 시스템에 액세스하기 위한 대책을 제공할 수 있다.

[0078] 일부 실시예에서, 처리 유닛은 그 로드-저장 유닛(LSU; load-store unit) 포트를 통해 사이클당 최대 128 비트를 판독/기입할 수 있고, 그 명령어 포트를 통해 사이클당 최대 128 비트 프로그램 코드를 판독할 수 있다. 각각 프로세서(102) 및 하드웨어 가속기(104)에 대한 IPI(202) 및 AMC(204) 인터페이스에 추가하여, 메모리 패브릭(106)은 AMBA(Advanced Microcontroller Bus Architecture) 고성능 버스(AHB) 및 AXI 버스 인터페이스를 통해 메모리 시스템으로의 동시 판독/기입 액세스를 제공할 수 있다. AHB 및 AXI는, 처리 유닛, 메모리 시스템, 및 주변 디바이스가 공유 버스 인프라구조를 이용하여 접속되는 것을 허용하는 표준 병렬 인터페이스 버스이다. 기타 임의의 적절한 버스도 이용될 수 있다. 일부 실시예에서, 메모리 패브릭(106)은 클록 사이클당 18 x 128-비트 메모리 액세스의 피크를 처리하도록 구성될 수 있다. 다른 실시예에서, 메모리 패브릭(106)은 많은 수의 비트를 갖는 고속 인터페이스를 이용하여 클록 사이클당 임의의 개수의 메모리 액세스를 처리하도록 설계될 수 있다.

[0079] 메모리 패브릭(106) 내의 메모리 시스템은 복수의 메모리 슬라이스를 포함할 수 있고, 각각의 메모리 슬라이스는 벡터 프로세서(102)들 중 하나와 연관되고 다른 벡터 프로세서(102)들에 비해 그 프로세서에게 우선적인 액세스를 제공한다. 각각의 메모리 슬라이스는 복수의 랜덤 액세스 메모리(RAM) 타일을 포함할 수 있고, 여기서, 각각의 RAM 타일은 판독 포트 및 기입 포트를 포함할 수 있다. 일부 경우에, 각각의 메모리 슬라이스에는, 관련된 메모리 슬라이스로의 액세스를 제공하기 위한 메모리 슬라이스 제어기가 제공될 수 있다.

[0080] 프로세서들 및 RAM 타일들은, IPI(202)라고도 하는 버스를 통해 서로 결합될 수 있다. 일부 경우에, IPI(202)는 벡터 프로세서(202) 중 임의의 것을 메모리 패브릭(106) 내의 메모리 슬라이스들 중 임의의 것과 결합시킬 수 있다. 적합하게는, 각각의 RAM 타일은 타일로의 액세스를 허가하기 위한 타일 제어 로직 블록을 포함할 수 있다. 타일 제어 로직 블록은 때때로 타일 제어 로직 또는 중재 블록이라 불린다.

[0081] 일부 실시예에서, 각각의 메모리 슬라이스는 복수의 RAM 타일 또는 물리적 RAM 블록을 포함할 수 있다. 예를 들어, 128kB의 크기를 갖는 메모리 슬라이스는, 4k x 32-비트 워드들로서 구성된 4개의 32kB 단일-포트형 RAM 타일(예를 들어, 물리적 RAM 요소)을 포함할 수 있다. 또 다른 예로서, 256kB의 크기를 갖는 메모리 슬라이스는, 8k × 32-비트 워드들로서 구성된 8개의 32kB 단일-포트형 RAM 타일(예를 들어, 물리적 RAM 요소)을 포함할 수 있다. 일부 실시예에서, 메모리 슬라이스는, 16kB 만큼 낮고 16MB 만큼 높은 용량을 가질 수 있다. 다른 실시예에서, 메모리 슬라이스는, 컴퓨팅 디바이스에 의해 처리되는 다양한 애플리케이션을 수용하기 위해 필요한 만큼의 용량을 갖도록 구성될 수 있다.

[0082] 일부 실시예에서, RAM 타일은, 단일 포트형 상보성 금속 산화물 반도체(CMOS) RAM을 포함할 수 있다. 단일 포

트형 CMOS RAM의 이점은, 대부분의 반도체 공정에서 일반적으로 이용가능하다는 것이다. 다른 실시예에서, RAM 타일은 멀티-포트형 CMOS RAM을 포함할 수 있다. 일부 실시예에서, 각각의 RAM 타일은, 16 비트 폭, 32 비트 폭, 64 비트 폭, 128 비트 폭이거나, 컴퓨팅 디바이스의 특정 애플리케이션에 의해 필요한 만큼 넓을 수 있다.

[0083] 단일 포트형 메모리 디바이스의 이용은, 메모리 서브시스템의 전력 및 면적 효율을 증가시키지만 메모리 시스템의 대역폭을 제한할 수 있다. 일부 실시예에서, 메모리 패브릭(106)은, 이를 메모리 디바이스들이, 복수의 소스(프로세서 및 하드웨어 블록)로부터의 복수의 동시 판독 및 기입 요청을 서비스할 수 있는 가상 멀티포트형 메모리 서브 시스템으로서 동작하는 것을 허용하도록 설계될 수 있다. 이것은, 복수의 물리적 RAM 인스턴스를 이용하고 이들로의 중재된 액세스를 제공해 복수의 소스를 서비스함으로써 수행할 수 있다.

[0084] 일부 실시예에서, 각각의 RAM 타일은 타일 제어 로직과 연관될 수 있다. 타일 제어 로직은, 벡터 프로세서(102) 또는 하드웨어 가속기(104)로부터 요청을 수신하고, 연관된 RAM 타일의 개개의 판독 및 기입 포트로의 액세스를 제공하도록 구성된다. 예를 들어, 벡터 프로세서(102)가 RAM 타일 내의 데이터에 액세스할 준비가 되면, 벡터 프로세서(102)가 메모리 데이터 요청을 RAM 타일에 직접 전송하기 전에, 벡터 프로세서(102)는 메모리 액세스 요청을 RAM 타일과 연관된 타일 제어 로직에 전송할 수 있다. 메모리 액세스 요청은, 처리 요소에 의해 요청된 데이터의 메모리 주소를 포함할 수 있다. 후속해서, 타일 제어 로직은, 메모리 액세스 요청을 분석하고 벡터 프로세서(102)가 요청된 RAM 타일에 액세스할 수 있는지를 결정할 수 있다. 벡터 프로세서(102)가 요청된 RAM 타일에 액세스할 수 있다면, 타일 제어 로직은 액세스 허가 메시지를 벡터 프로세서(102)에 전송할 수 있고, 후속해서, 벡터 프로세서(102)는 메모리 데이터 요청을 RAM 타일에 전송할 수 있다.

[0085] 일부 실시예에서, 타일 제어 로직은, 많은 처리 유닛(예를 들어, 벡터 프로세서들 및 하드웨어 가속기들)이 동일한 RAM 타일에 액세스하는 순서를 결정하고 집행하도록 구성될 수 있다. 예를 들어, 타일 제어 로직은, 2개 이상의 처리 유닛이 RAM 타일에 동시에 액세스하려고 시도하는 순간을 검출하도록 구성된 충돌 검출기를 포함할 수 있다. 충돌 검출기는 액세스 충돌이 발생했고 액세스 충돌이 해결되어야 한다는 것을 런타임 스케줄러(runtime scheduler)에 보고하도록 구성될 수 있다.

[0086] 메모리 패브릭(106)은 또한, 메모리로부터 벡터 프로세서(102) 또는 하드웨어 가속기(104)로, 또는 벡터 프로세서(102) 또는 하드웨어 가속기(104)로부터 메모리로, 데이터 비트를 전송하기 위한 메모리 버스를 포함할 수 있다. 메모리 패브릭(106)은 또한, 벡터 프로세서(102), 하드웨어 가속기(104), 및 메모리 사이의 데이터 전송을 조율하는 직접 메모리 액세스(DMA) 제어기를 포함할 수 있다.

[0087] 일부 실시예에서, 하드웨어 가속기(104)는 별도의 버스를 통해 메모리 패브릭(106)에 결합될 수 있다. 별도의 버스는, 적어도 하나의 하드웨어 가속기로부터 요청을 수신하고 관련된 메모리 슬라이스 제어기를 통해 메모리 슬라이스로의 액세스를 하드웨어 가속기에게 허가하도록 구성된 가속기 메모리 제어기(AMC; accelerator memory controller)(204)를 포함할 수 있다. 따라서, 하드웨어 가속기(104)에 의해 채용된 메모리 액세스 경로는 벡터 프로세서(102)에 의해 채용된 경로와는 상이할 수 있다는 것을 이해할 것이다. 사실상, AMC(204)는, 주소 필터링, 중재 및 멀티플렉싱을 수행할 수 있다. 일부 실시예에서, 하드웨어 가속기(104)는, 메모리 패브릭(106)에 액세스하는데 있어서의 지연을 감안하기 위해 내부 버퍼(예를 들어, FIFO 메모리)를 포함할 수 있다.

[0088] 일부 실시예에서, AMC(204)는, 예를 들어 복수의 모바일 산업 프로세서 인터페이스(MIPI) 카메라 인터페이스를 포함하는 하나 이상의 주변 디바이스(108)에 결합될 수 있다. AMC(204)는 또한, AXI 및 APB 인터페이스에 접속되어 2개의 시스템 RISC 프로세서가 AMC(204)를 통해 메모리 패브릭(106) 내의 메모리 슬라이스에 액세스하는 것을 허용할 수 있다.

[0089] 일부 실시예에서, AMC(204)는 메모리 패브릭(106)의 각각의 메모리 슬라이스 내로의 한 쌍의 64 비트 포트를 포함할 수 있다. AMC(204)는, 부분적 주소 디코딩에 의해 하드웨어 가속기(104)로부터의 요청을 적절한 메모리 슬라이스로 라우팅하도록 구성될 수 있다.

[0090] 일부 실시예에서, AMC(204)는, 메모리 패브릭(106) 내의 메모리 슬라이스로의 액세스를 제공하기 위해 다양한 처리 유닛에 결합될 수 있다. 예를 들어, AMC(204)는, 메모리 패브릭(106) 내의 메모리 슬라이스로의 액세스를 제공하기 위해 임의의 유형의 하드웨어 가속기 또는 제3자 요소에 결합될 수 있다. AMC(204)는 또한, 컴퓨팅 디바이스(100) 외부에 있는 메모리 디바이스를 포함한, 컴퓨팅 시스템의 더 넓은 메모리 공간으로의 액세스를 제공하도록 구성될 수 있다.

[0091] 일부 실시예에서, AMC(204)는, 동일한 메모리 슬라이스로의 동시 메모리 액세스 요청들을 라운드 로빈 방식(round-robin manner)으로 중재할 수 있다. 예를 들어, 하드웨어 가속기(104) 등의 처리 유닛은, 메모리 주소

를 포함하는 메모리 액세스 요청을 AMC(204)에 전송할 수 있다. AMC(204)가 메모리 액세스 요청을 수신하면, AMC(204)는 메모리 액세스 요청 내의 메모리 주소가 메모리 패브릭(106) 내의 메모리 슬라이스와 연관되어 있는지를 결정한다. 메모리 액세스 요청 내의 메모리 주소가 메모리 패브릭(106) 내의 메모리 슬라이스와 연관되어 있지 않다면, AMC(204)는 메모리 요청을 AXI 마스터에 포워딩할 수 있다. 메모리 액세스 요청 내의 메모리 주소가 메모리 패브릭(106) 내의 메모리 슬라이스와 연관되어 있다면, AMC(204)는 원하는 메모리 위치로의 액세스를 제공하도록 메모리 액세스 요청을 중재할 수 있다.

- [0092] 주변 디바이스(108)는, 복수의 이미지 센서 및 가속도계 등의 외부 디바이스로/로부터 데이터 비트를 전송 및 수신하기 위한 통신 채널을 제공하도록 구성될 수 있다. 주변 디바이스(108)는, 벡터 프로세서(102), 하드웨어 가속기(104), 및 메모리 패브릭(106)이 외부 디바이스와 통신하기 위한 통신 메커니즘을 제공할 수 있다.
- [0093] 전통적으로, 주변 디바이스의 기능은 고정되고 하드-코딩되었다.
- [0094] 예를 들어, 모바일 산업 프로세서 인터페이스(MIPI) 주변기기들은, SPI, I2C, I2S 또는 기타 임의의 적절한 표준 등의 저속 디지털 인터페이스를 역시 구현하는 외부 디바이스와만 인터페이스할 수 있었다.
- [0095] 그러나, 본 개시내용의 일부 실시예에서, 주변 디바이스(108)의 기능은 소프트웨어를 이용하여 정의될 수 있다. 더 구체적으로는, 주변 디바이스(108)는, SPI, I2C, I2S 또는 기타 임의의 적절한 프로토콜 등의, 표준화된 인터페이스 프로토콜의 기능을 에뮬레이트할 수 있는 에뮬레이션 모듈을 포함할 수 있다.
- [0096] 전력 관리 모듈(110)은, 컴퓨팅 디바이스(100) 내의 블록들의 활동을 제어하도록 구성된다. 더 구체적으로는, 전력 관리 모듈(110)은, 전력 아일랜드라고도 지칭되는, 지정된 블록의 전원 전압을 제어하도록 구성된다. 예를 들어, 전력 관리 모듈(110)이 전력 아일랜드의 전원을 인에이블 할 때, 컴퓨팅 디바이스(100)는 적절한 전원 전압을 전력 아일랜드에 제공하도록 구성된다. 전력 관리 모듈(110)은, 레지스터에서 또는 버스 상의 신호 라인 상에서 인에이블 신호를 인가함으로써 전력 아일랜드의 전원을 인에이블하도록 구성될 수 있다. 일부 실시예에서, 전력 관리 모듈(110)은 또한, 컴퓨팅 디바이스(100)의 하나 이상의 입력/출력 핀을 통해 외부 디바이스의 활동을 제어하도록 구성될 수 있다.
- [0097] 일부 실시예에서, 전력 아일랜드는 항상 전원이 켜져 있을 수 있다(예를 들어, 전원 전압은 항상 전력 아일랜드에 제공된다). 이러한 전력 아일랜드는, 상시-온 상태의 전력 아일랜드(always-on power island)라고 할 수 있다. 일부 실시예에서, 항상 온-상태의 전력 아일랜드는, 예를 들어 범용 입력-출력(GPIO) 핀, 외부 인터페이스, 및/또는 저주파수 타이머 또는 파워-온 리셋 등의 내부 기능 블록들로부터의 신호들을 모니터링하는데 이용될 수 있다. 이러한 방식으로, 컴퓨팅 디바이스(100)는 이벤트 또는 이벤트 시퀀스에 응답할 수 있고, 이벤트 또는 이벤트 시퀀스에 응답하는데 필요한 전력-아일랜드만을 적응적으로 파워-업할 수 있다.
- [0098] 도 3은 일부 실시예에 따른 하드웨어 가속기를 나타낸다. 하드웨어 가속기(104)는 하드웨어 이미지 처리 필터들의 집합을 포함할 수 있다. 하드웨어 가속기(104)는, 계산 집약적인 기능성들 중 일부가 벡터 프로세서들(102)로부터 오프로드되는 것을 가능하게 할 수 있다. 가속기(104)는, 높은 대역폭에서 메모리 패브릭(106) 내의 메모리 슬라이스에 액세스하기 위해 AMC(204)에 결합될 수 있다.
- [0099] 일부 실시예에서, 하드웨어 가속기(104)는 AMC(204)를 통해 메모리 패브릭(106)에 결합될 수 있다. 일부 실시예에서, 하드웨어 가속기(104)는, MIPI 수신기 필터 및 MIPI 전송기 필터를 포함한 하나 이상의 필터 모듈(예를 들어, 20개의 필터 모듈들)을 포함할 수 있다. 일부 실시예에서, 필터 모듈은, 하나의 판독-전용 AMC 인터페이스(판독 클라이언트 인터페이스) 및 하나의 기입-전용 AMC 인터페이스(기입 클라이언트 인터페이스)를 포함할 수 있다. 다른 실시예들에서, 필터 모듈은 또한, 복수의 판독-전용 AMC 인터페이스를 가질 수 있다. 예를 들어, 필터 모듈은, 복수의 입력 버퍼들, (동일한 버퍼로부터의) 복수의 평면들에 대한 별별 액세스를 위한 복수의 판독-전용 AMC 인터페이스를 가질 수 있다. 복수의 판독-전용 AMC 인터페이스는, 필터 모듈의 처리량을 유지하기 위해 여분의 메모리 판독 대역폭을 제공하는데 이용될 수 있다. 하드웨어 가속기(104)는 단일 필터 모듈만을 가질 수 있기 때문에, 하드웨어 가속기(104)의 설명은 각각의 필터 모듈에 동등하게 적용될 수 있다. 마찬가지로, 필터 모듈은 하드웨어 가속기 내의 유일한 필터 모듈일 수 있기 때문에, 필터 모듈의 설명은 하드웨어 가속기에 동등하게 적용될 수 있다.
- [0100] 일부 실시예에서, AMC(204)는, 메모리 패브릭(106) 내의 각각의 메모리 슬라이스로의 하나 이상의 양방향(예를 들어, 판독/기입) 포트를 갖는다. 포트는 많은 수의 비트를 수용할 수 있다. 예를 들어 포트는 64 비트 통신을 수용할 수 있다. 일부 실시예에서, AMC(204)는 또한, 외부 DRAM 디바이스로의 직접 접속을 제공하는 AXI 마스터를 포함할 수 있다.

- [0101] 일부 실시예에서, 필터 모듈은, 주로 메모리 패브릭(106) 내의 버퍼들을 처리하도록 설계될 수 있다. 예를 들어, MIPI 수신기 모듈과 MIPI 전송기 필터 모듈을 제외하고, 필터 모듈은 그 AMC 클라이언트를 통해서만 데이터를 입력 및 출력할 수 있다. 버퍼 베이스 주소를 포함한, 필터 모듈의 구성은, 수 개의 APB 슬레이브 인터페이스를 통해 달성될 수 있다.
- [0102] 일부 실시예에서, 하드웨어 가속기(104)는, MIPI 수신기 필터 모듈 및 MIPI 전송기 필터 모듈을 통해 이미지 데이터를 수신할 수 있다. MIPI 수신기 필터 모듈 및 MIPI 전송기 필터 모듈은, 하드웨어 가속기(104) 내의 다른 필터 모듈들이 MIPI 수신기 제어기 및 MIPI 트랜시버 제어기로의 직접 접속을 확립하는 것을 허용할 수 있다. MIPI 수신기 필터 모듈 및 MIPI 전송기 필터 모듈은, 별별 인터페이스를 통해 MIPI 제어기들에 접속할 수 있고, MIPI 제어기로부터/로 직접 메모리 패브릭(106) 내로/밖으로 데이터를 스트리밍하는데 이용할 수 있다.
- [0103] 일부 실시예에서, 하드웨어 가속기(106)는, AMC(204)를 통해 액세스되는, 메모리 패브릭(106)에 버퍼링된 이미지 데이터의 스캔-라인 상에서 동작할 수 있다. AMC(204)는 클라이언트 인터페이스로부터 타겟 메모리 슬라이스(또는 AXI 마스터)로 트랜잭션을 라우팅하고 각각의 메모리 슬라이스에서 상이한 클라이언트들로부터의 동시 트랜잭션들을 중재할 수 있다. 일부 실시예에서, 하드웨어 가속기(106) 내의 복수의 필터 모듈은, 하나 이상의 필터 모듈(생산자/부모라고도 함)의 출력 버퍼를 다른 필터 모듈(소비자/어린이라고도 함)의 입력 버퍼에 결합함으로써 스트리밍 방식으로 함께 접속될 수 있다.
- [0104] 일부 실시예에서, 하드웨어 가속기(104) 내의 필터 모듈은 현재 픽셀을 중심으로 하는 픽셀에 관해 2차원 커널을 동작시킬 수 있다. 커널 내의 모든 픽셀은 현재 픽셀을 중심으로 한 픽셀들의 처리에 기여할 수 있다.
- [0105] 일부 실시예에서, 하드웨어 가속기(104) 내의 필터 모듈은 이미지를 라인별로 처리할 수 있다. 예를 들어, 필터 모듈은 이미지를 위에서 아래로 스캔하여 이미지의 스캔-라인을 생성하고, 예를 들어, 왼쪽에서 오른쪽으로 이동하면서, 스캔-라인들을 처리할 수 있다. 다른 예에서, 필터 모듈은, 필터 처리에 적합한 임의의 배향/순서로 이미지를 스캐닝함으로써 이미지의 스캔-라인들을 생성할 수 있다.
- [0106] 일부 실시예에서, 필터 모듈은, 데이터를 판독하여 스캔-라인 상의 제1 픽셀에 대한 커널을 형성함으로써 이미지의 스캔-라인을 처리할 수 있다. 필터 모듈은 슬라이딩 윈도우 방식으로 커널을 슬라이딩함으로써 스캔-라인을 처리할 수 있다. 일단 처리가 완료되고 나면, 필터 모듈은 출력 픽셀들을 출력 버퍼 또는 메모리 위치에 기입할 수 있다.
- [0107] 일부 실시예에서, 필터링을 위한 커널들은 통상적으로 정사각형이며, 종종 각각의 면을 따라 홀수 개의 픽셀을 갖는다, 예를 들어, 3x3, 5x5 또는 7x7. 필터 모듈이 KxK 픽셀 커널을 이용한다면, 처리되고 출력 버퍼에 기록되는 이미지 데이터의 각각의 라인에 대해 입력 버퍼로부터 이미지 데이터의 K개의 스캔-라인들이 판독될 수 있다.
- [0108] 일부 실시예에서, 하드웨어 가속기(104)는 순환 입력 버퍼를 이용할 수 있다. 타겟 필터 모듈이 입력으로서 다른 필터 모듈(부모 필터 모듈이라고도 함)의 출력 스캔-라인을 수신하도록 구성되어 있다고 가정하자. 타겟 필터 모듈이 KxK 픽셀 커널을 이용한다고 역시 가정하자. 그렇다면, 타겟 필터 모듈에 대한 입력 버퍼는, 이미지 데이터의 적어도 $(K + 1)$ 개의 스캔-라인 : 필터 모듈에 대한 K개의 스캔-라인과 부모 필터 모듈의 출력 스캔-라인을 동시에 수신하기 위한 하나(또는 그 이상의) 스캔-라인을 유지하도록 설계될 수 있다. 이 예에서, 입력 버퍼는 원형이기 때문에, 부모 필터 모듈로부터 $(K + 1)$ 개의 스캔-라인을 수신한 후, $(K + 2)$ 번째 스캔-라인은 첫 번째 라인의 위치 위에 기입될 수 있다. 대부분의 경우, 부모 필터 모듈은 입력 이미지 내의 현재 라인 번호와 관련하여 타겟 필터 모듈보다 앞에 위치할 수 있다. 초기 구성 후, 필터 모듈의 판독 및 기입 AMC 클라이언트는, 필터 모듈의 입력 및 출력 버퍼에 액세스할 때 원형 버퍼 주소 래핑을 처리할 수 있다.
- [0109] 일부 실시예에서, 하드웨어 가속기(104) 내의 버퍼는 미리결정된 바이트 수만큼 정렬될 수 있다. 예를 들어, 하드웨어 가속기(104) 내의 버퍼는 8-바이트 경계 상에 정렬될 수 있다. 트랜잭션 라우팅을 용이하게 하기 위해, 판독 및 기입 클라이언트와 AMC는 정렬된 버퍼 액세스만을 제공하도록 구성될 수 있다. 이미지 폭이 미리 결정된 바이트 수의 배수가 아닐 때, 하드웨어 가속기(104)는 각각의 스캔-라인의 (정렬되지 않은) 끝과 다음 바이트 경계 사이에서 출력 버퍼에 널 바이트(null byte)를 기입하도록 구성될 수 있다.
- [0110] 도 3은, 입력 데이터 스트림(예를 들어, 하나 이상의 이미지의 스캔-라인)에 관하여 필터 커널 레지스터(302)에 저장된 필터 커널을 동작시키기 위한 하드웨어 가속기의 구현을 나타낸다. 입력 데이터 스트림은 하나 이상의 이미지의 픽셀에 대응할 수 있다. 하드웨어 가속기(104)는, 데이터경로 파이프라인(304), 파이프라인 정지 제어기(306), 라인 버퍼 판독 클라이언트(308), 라인 시작 제어 입력(310), 및 라인 버퍼 기입 클라이언트(310)를

포함할 수 있다. 일부 실시예에서, 하드웨어 가속기(104)는, 메모리 패브릭(106) 내의 메모리 슬라이스에 액세스하기 위해 적어도 하나의 AMC 판독 클라이언트 인터페이스(314) 및/또는 적어도 하나의 AMC 기입 클라이언트 인터페이스(316)를 포함할 수 있다. AMC(204) 상의 판독/기입 클라이언트 인터페이스의 수는 적절하게 구성 가능하다.

[0111] 일부 실시예에서, 필터 커널 레지스터(302)는, 입력 데이터 스트림에 관해 동작될 커널을 수정하도록 프로그램될 수 있다. 필터 커널 레지스터(302)는 다양한 커널 크기를 수용하도록 구성될 수 있다. 예를 들어, 필터 커널 레지스터(302)는, 3x3 커널, 5x5 커널, 7x7 커널, 9x9 커널 또는 $m \times n$ 으로 표현된 기타 임의의 커널 크기를 수용하도록 구성될 수 있다. 일부 경우에, m 은 n 과 같을 수 있다; 다른 경우에, m 은 n 과는 상이할 수 있다. 일부 실시예에서, 필터 커널 레지스터(302)는 다양한 차원의 커널을 수용하도록 구성될 수 있다. 예를 들어, 필터 커널 레지스터(302)는, 1차원 필터, 2차원 필터, 3차원 필터, 또는 임의의 정수-차원 필터를 수용하도록 구성될 수 있다.

[0112] 일부 실시예에서, 라인 버퍼 판독 클라이언트(308)는, 이미지의 스캔-라인(예를 들어, 이미지 그리드 상의 이미지의 행 또는 열)을 수신하고 스캔-라인을 데이터경로 파이프라인(304)에 제공하도록 구성된다. 라인 버퍼 판독 클라이언트(308)는 AMC 판독 인터페이스(314)를 통해 이미지의 스캔-라인을 수신할 수 있다. 일단 데이터경로 파이프라인(304)이 이미지의 커널 및 스캔-라인을 수신하고 나면, 데이터경로 파이프라인(304)은 필터링 연산을 수행할 수 있다. 일단 데이터경로 파이프라인(304)이 필터링 연산을 완료하고 나면, 데이터경로 파이프라인(304)은 결과 라인을 라인-버퍼 기입 클라이언트(312)에 저장할 수 있다. 라인 버퍼 기입 클라이언트(312)는, 선택사항으로서, 결과 라인을 AMC 기입 인터페이스(316)를 통해 메모리 슬라이스에 저장할 수 있다. 파이프라인 정지 제어기(306)는 라인-버퍼 기입 클라이언트(312)가 오버플로우되지 않도록 보장하기 위해 파이프라인의 소정 부분을 정지시킬 수 있다.

[0113] 일부 실시예에서, 라인 시작 제어기(310)는 데이터경로 파이프라인(304)이 이미지의 수신된 스캔-라인을 처리하기 시작하는 시간 인스턴스를 제어할 수 있다. 라인 시작 제어기(310)는 또한, 데이터경로 파이프라인(304)의 하나 이상의 부분이 맞춤화된 동작을 수행하는 것을 선택적으로 인에이블하도록 구성될 수 있다. 일부 경우에, 라인 시작 제어기(310)는 또한, 데이터경로 파이프라인(304)에 의한 필터링 연산 동안에 이용될 계수들을 제어 할 수 있다.

[0114] 일부 실시예에서, 데이터경로 파이프라인(304) 및 라인 시작 제어기(310)는 프로그램 가능할 수 있다. 데이터경로 파이프라인(304) 및 라인 시작 제어기(310)는, 상이한 유형들의 필터링 연산이 하드웨어 가속기(104)에 의해 수행될 수 있도록 프로그램될 수 있다. 예를 들어, 데이터경로 파이프라인(304) 및 라인 시작 제어기(310)는, 맞춤화된 필터링 연산이 하드웨어 가속기(104)에 의해 실행될 수 있도록, 계수 세트 및/또는 임계치 등의, 필터 동작 파라미터들로 프로그램될 수 있다. 필터 동작 파라미터들은 또한, 필터 커널 크기, 계수, 스케일링 비율, 이득, 임계치, 록업 테이블, 또는 기타 임의의 적절한 파라미터를 포함할 수 있다. 따라서, 하드웨어 가속기(104)는 다양한 이미지 필터링 연산을 수용하기 위한 일반 래퍼(generic wrapper)로서 간주될 수 있다.

[0115] 일부 실시예에서, 데이터경로 파이프라인(304)은 하나 이상의 숫자 포맷으로 표현된 수를 처리하도록 구성될 수 있다. 예를 들어, 데이터경로 파이프라인(304)은, 부동 소수점 숫자, 예를 들어, fp16(IEEE754와 유사한 16 비트 부동 소수점 포맷), 정수, 고정 소수점 숫자, 또는 이미지 처리에 적합한 기타 임의의 숫자 포맷에 관해 동작하도록 설계될 수 있다.

[0116] 하드웨어 가속기(104)는, 데이터경로 파이프라인(304)이 입력 데이터 버퍼(308)로부터 스캔-라인을 소비하는 방법 및 데이터경로 파이프라인(304)이 처리된 스캔-라인을 출력 데이터 버퍼(312)에 저장하는 방법을 제어하도록 구성될 수 있다. 하드웨어 가속기(104)는, 2개의 제어 모드 : 버퍼 채움 제어(BFC; buffer fill control) 모드와 동기 모드 중 하나를 구현하도록 구성될 수 있다.

[0117] 일부 실시예에서, BFC 모드 하에서, 하드웨어 가속기(104)는 채움 레벨의 내부 카운트(예를 들어, 입력 버퍼에 저장된 스캔-라인의 수)를 유지하도록 구성될 수 있다. 하드웨어 가속기(104)는, (1) 하드웨어 가속기가 인에 이를되고, (2) 입력 버퍼가 충분한 수의 스캔-라인을 가지며, (3) 처리된 스캔-라인을 저장할 공간이 출력 버퍼에 있을 때, 그 입력 버퍼로부터의 스캔-라인을 자율적으로 처리하도록 구성될 수 있다. 일부 경우에는, 데이터경로 파이프라인(304)을 실행하는데 필요한 버퍼 채움 레벨은 커널의 높이에 의존할 수 있다. 예를 들어, 커널이 3x3일 때, 하드웨어 가속기(104)는 필터를 동작시키기 위해 적어도 3개의 스캔-라인을 요구할 수 있다.

[0118] 일부 실시예에서, 동기 제어 모드 하에서, 하드웨어 가속기 내의 필터 모듈은 필터 모듈에 대한 시작 비트가 온

될 때 동작하도록 구성될 수 있다. 시작 비트는 예를 들어 소프트웨어 모듈을 이용하여 온으로 될 수 있다. 동기 제어 하에서, 소프트웨어 모듈은, 필터 모듈에 대한 입력 버퍼가 충분한 수의 스캔-라인을 갖고 있다는 것 및 필터 모듈에 대한 출력 버퍼가 필터 모듈로부터의 처리된 스캔-라인을 저장하기에 충분한 공간을 갖고 있다는 것을 결정하도록 구성될 수 있다. 일단 이들 조건이 충족되고 나면, 소프트웨어 모듈은 필터 모듈의 시작 비트를 턴온할 수 있다.

[0119] 양쪽 모드 모두에서, 일단 필터 모듈이 스캔-라인을 처리하고 나면, 필터 모듈은 자신의 버퍼 내의 및 입력 이미지 내의 현재 라인 인덱스를 업데이트할 수 있다. 일부 실시예에서, 출력 이미지가 입력 이미지와 동일한 크기를 갖지 않을 때, 필터 모듈은 출력 이미지에서 그 현재 라인 인덱스를 역시 업데이트할 수 있다. 라인 인덱스의 값(및 버퍼 채움 제어를 위한 버퍼 채움 레벨)은 필터 모듈의 내부 상태를 나타낼 수 있다. 이러한 내부 상태는 소프트웨어 모듈에 의해 액세스될 수 있고, 다음 사이클에서 필터 모듈이 실행되기 전에 필터 모듈의 컨텍스트가 전환될 수 있도록, 저장, 업데이트 및 복원될 수 있다.

[0120] 일부 실시예에서, 하드웨어 가속기(104) 내의 버퍼는 복수의 데이터 평면을 유지하도록 구성될 수 있다. 예를 들어, 하드웨어 가속기(104) 내의 버퍼는, 이미지의 적색 채널, 녹색 채널, 및 청색 채널을 별개의 평면들에 유지하도록 구성될 수 있다. 일부 예에서, 하드웨어 가속기(104) 내의 버퍼는 최대 16개의 평면을 지원하도록 구성될 수 있다. 각각의 평면 내의 이미지 데이터의 스캔-라인은 연속적으로 저장될 수 있으며, 평면들은 그들의 번호 및 평면 보폭(plane stride)에 의해 정의될 수 있다.

[0121] 일부 실시예에서, 하드웨어 가속기(104) 내의 필터 모듈은 각각의 데이터 평면으로부터 한번에 하나씩 순차적으로 스캔-라인을 처리하도록 구성될 수 있다. 순차 처리의 경우, 제어 관점에서, 모든 평면으로부터의 스캔-라인들은 동일한 타임 스탬프를 갖는 것으로 간주될 수 있다. 다른 실시예에서, 하드웨어 가속기(104) 내의 필터 모듈은 복수의 데이터 평면을 병렬로 처리하도록 구성될 수 있다.

[0122] 일부 실시예에서, 이미지/비디오 스트림을 처리하기 이전에, 또는 컨텍스트가 전환된다면, 필터 모듈은 적절히 구성되고 인에이블될 수 있다. 각각의 필터 모듈은 그 입력 버퍼(들) 및 출력 버퍼 구성을 정의하는 한 세트의 소프트웨어 프로그램 가능한 레지스터를 포함할 수 있다.

[0123] 일부 실시예들에서, 필터 모듈 내의 버퍼는 다음 파라미터들 중 하나 이상을 이용하여 프로그램될 수 있다 :

● 베이스(base) : 베이스 주소. 이 파라미터는 버퍼의 베이스 주소를 명시할 수 있다. 주소는 바이트 경계(예를 들어, AMC 클라이언트 데이터 버스의 폭) 상에서 정렬될 수 있다.

● nl : 스캔-라인 수. 원형 버퍼 모드에서, 이 파라미터는 스캔-라인에서 원형 버퍼의 크기를 지정할 수 있다. 원형 버퍼의 최대 스캔-라인 수는 1023일 수 있지만, 다른 상한도 가능하다. 버퍼가 $nl = 0$ 으로 구성된다면, 이것은 버퍼가 비-원형 모드임을 나타낸다. 따라서, $nl = 0$ 은, 버퍼에 액세스하는 판독/기입 클라이언트(들)를, 버퍼 내의 스캔-라인 수가 이미지의 높이에 대응하고 어떠한 원형 버퍼 포인터 래핑(wrapping)도 발생하지 않는 비-원형 또는 비-랩 모드에 둔다.

● ls : 라인 보폭. 라인 보폭은, 고정된 수의 바이트, 예를 들어, 8 바이트의 배수일 수 있다. 최대 라인 보폭은 미리결정될 수 있다. 예를 들어, 최대 라인 보폭은 (32MB - 8) 바이트일 수 있다. 라인 보폭 및 라인 수는 원형 버퍼 포인터 산술연산을 수행하기 위해 판독/기입 클라이언트에 의해 이용될 수 있다. 라인 보폭은 이미지 폭보다 크거나 같을 수 있다.

● np : 평면의 수. 이 파라미터는 버퍼가 나타내는 평면의 수를 나타낸다. $np = 0$ 일 때, 이것은 버퍼가 비-평면 데이터(예를 들어, 단일 평면 데이터)를 나타내는 것을 나타낸다. 버퍼 내의 라인 버퍼 저장량은 평면 수에 의해 좁해질 수 있다.

● ps : 평면 보폭. 평면 보폭은, 고정된 수의 바이트, 예를 들어, 8 바이트의 배수일 수 있다. 최대 평면 보폭은 미리결정될 수 있다. 예를 들어, 최대 평면 보폭은 (32MB - 8) 바이트일 수 있다. 일반적으로, 평면 보폭은 nl 에 ls 를 곱한 값보다 크거나 같을 수 있다. 그러나, 다른 평면 보폭이 가능할 수 있다.

● 포맷 : 버퍼 데이터 포맷. 이 파라미터는 픽셀 데이터의 크기를 바이트 단위로 명시할 수 있다. 예를 들어, FP16 버퍼의 경우, 포맷은 2로 설정되어, 픽셀당 2 바이트를 나타낼 수 있다.

[0130] 일부 실시예에서, 필터 모듈 내의 출력 버퍼는 다음 파라미터들 중 하나 이상을 이용하여 프로그램될 수 있다 :

● 오프셋: 오프셋은 베이스 주소(및 각각의 라인의 시작)로부터 첫 번째 픽셀까지의 오프셋을 명시할 수 있다.

이 파라미터는 바이트 경계 상에서 정렬되는 베피의 한도를 해결하는데 이용될 수 있다. 오프셋을 이용하여, 예를 들어 출력 베피의 소비자에 의한 수평 패딩(padding)을 위한 공간이 스캔-라인의 좌측 상에 예약될 수 있다. 디폴트 오프셋은 0이다. 비-제로 오프셋이 명시된다면, 첫 번째 출력 팩셀 전에 각각의 출력 스캔-라인에 널 바이트가 기입될 수 있다.

[0132] 일부 실시예에서, 필터 모듈은 다양한 데이터 타입을 지원할 수 있다. 필터 모듈에 의해 지원되는 가장 일반적인 데이터 타입은 다음과 같다:

- U8 - 부호없는 8비트 정수 데이터

- U8F - 부호없는 8비트 소수 데이터, 범위 [0, 1.0]

- U16 - 부호없는 16비트 정수 데이터

- U32 - 부호없는 32비트 정수 데이터

- FP16 - 절반-정밀도 (16 비트) 부동 소수점

- FP32 - 전체-정밀도 (32 비트) 부동 소수점

[0139] 일부 실시예에서, 필터 모듈의 데이터경로 파이프라인은 그 동작을 위해 최적화될 수 있다 : 절반-정밀도 부동 소수점(FP16) 산술연산은 높은 동적 범위를 수반하는 연산들에 대해 이용될 수 있다; 최적화된 고정 소수점 산술연산은 고정밀도를 유지하는 것이 더욱 중요한 경우에 이용될 수 있다.

[0140] 일부 실시예에서, FP16 산술연산을 이용하여 구현된 필터 모듈은 FP16 베피에 대해서만 판독/기입하는 것으로 제약되지 않을 수 있다. U8F 베피는 또한, 필터 모듈 내에서 자동으로 발생하는 FP16으로의/으로부터의 변환을 통해 액세스될 수 있다.

[0141] 일부 실시예에서, FP16 산술연산을 이용하여 필터 모듈이 구현되는 일부 실시예에서, 베피는 FP16이거나 U8F일 수 있다. 베피가 FP16인 경우, 베피 구성 포맷은 2로 설정될 수 있다. 베피가 U8F이면, 베피 구성 포맷은 1로 설정될 수 있다. FP16 데이터경로 파이프라인이 있는 필터 모듈의 경우, 입력 베피 포맷이 "1"이라면, 판독 클라이언트는 U8F 입력 데이터를 처리 전에 FP16으로 자동으로 변환할 수 있다. 출력 베피 포맷이 "1"이라면, 기입 클라이언트는 저장 전에 데이터경로 파이프라인의 FP16을 U8F로 변환할 수 있다.

[0142] 일부 실시예에서, U8F는, 1.0/255를 곱함으로써, [0, 1.0]의 범위에서, 정규화된 FP16으로 변환된다. 정규화된 FP16은, 255를 곱하고 반올림하여 부동 소수점 값을 효과적으로 8 비트로 양자화함으로써, U8F로 변환될 수 있다. 일부 실시예에서, FP16 데이터경로 파이프라인을 갖는 필터 모듈로부터의 출력 데이터는 선택사항으로서 정규화된 범위 [0, 1.0]로 클램프될 수 있다. U8F로의 변환이 인에이블되면, 정규화된 범위로의 클램프는 무시적으로 인에이블되고, 위에서 설명된 U8F로의 변환 이전에 수행된다. FP16 데이터경로 파이프라인을 이용하여 구현된 필터 모듈은 정규화된 범위 [0, 1.0]의 데이터 처리로 제한되지 않는다; FP16의 전체 범위도 역시 지원될 수 있다.

[0143] 일부 실시예에서, 필터 모듈은 입력 이미지에서 그 수직 위치를 추적하도록 구성된다. 필터 모듈은 이 정보를 이용하여 라인 복제 또는 반사에 의해 이미지의 상단 및 하단에서 수직 패딩을 수행할 수 있다. 수직 패딩을 수행하지 않는 필터 모듈은 입력 이미지보다 작은 출력 이미지를 생성할 수 있는데, 이것은 일부 경우에는 바람직하지 않을 수도 있다.

[0144] 일부 실시예에서, 필터 모듈이 수직 패딩을 수행하도록 구성될 때, 입력 베피에 의해 유지될 수 있는 스캔-라인의 최소 개수 M은 다음과 같을 수 있다 :

M = (K >> 1) + 1, 여기서, >>는 우측 비트-시프트 연산자를 나타낸다.

[0146] 이미지의 상단에서, (스캔-라인의 관점에서) 입력 베피의 용량이 M보다 작을 때, 필터링 연산을 수행하기에 충분한 스캔-라인이 베피 내에 없다. (스캔-라인의 관점에서) 입력 베피의 용량이 M보다 크거나 같을 때, 수직 패딩이 수행되는 경우 데이터가 처리될 수 있다. 유사하게, 이미지의 하단에서, 마지막 (K >> 1) 라인을 처리할 때, 필터 모듈은 라인 N-1(또는 라인 N-1 및 그 위의 라인들)의 복제를 수행할 수 있다.

[0147] 일부 실시예에서, 수직 패딩은 커널이 짹수 차원(dimension)을 가질 때 수행될 수 있다. 짹수 차원의 커널에 대한 수직 패딩은, 하단에서 한개 적은 라인이 패딩되어야 한다는 점을 제외하고는, 홀수 차원의 커널에 대한 수직 패딩과 사실상 동일할 수 있다.

- [0148] 일부 실시예에서, 필터 모듈은 수평 패딩을 수행할 수 있다. 픽셀 커널의 수평 패딩은, 데이터가 입력 버퍼로부터 판독되고 픽셀 커널 레지스터에 기입될 때 수행될 수 있다. 필터 모듈은, 현재 라인과 라인의 시작과 끝에서의 위치를 인식할 수 있다. 따라서, 유효한 픽셀 커널 레지스터는 유효한 데이터를 보유하지 않는 레지스터 내에 복제될 수 있다. 수직 패딩에서와 같이, 수평 패딩이 수행되는지의 여부는, 주어진 필터 모듈의 특정 기능 및 요구사항에 의존할 수 있다.
- [0149] 일부 실시예에서, 원형 버퍼 모드에서, 필터 모듈은, 그 입력 버퍼로부터 하나의 스캔-라인을 처리하고 그 처리된 스캔-라인을 그 출력 버퍼에 기입하도록 구성될 수 있다. 이 동작 세트는 필터 런(filter run)이라고 지칭될 수 있다.
- [0150] 일부 실시예에서, 유연성을 위해, 필터 런을 제어할 수 있는 2개의 상이한 제어 메커니즘이 제공될 수 있다. 버퍼 채움 제어 모드라고 불리는 제1 메커니즘에서, 필터 모듈은 원형 버퍼의 채움 레벨을 추적하고 자체적으로 실행가능 여부를 결정할 수 있다. 이 접근법은 본질적으로 비동기식이다; 필터 모듈은, 요구되는 조건이 충족되는 한, 반복적으로 실행될 수 있다. 레지스터 내의 제어 비트는, 스캔-라인이 입력 버퍼에 추가되었거나 출력 버퍼로부터 제거되었을 때 소프트웨어가 필터 모듈에 통보하는 것을 허용하기 위해 제공된다. 스캔-라인이 입력 버퍼에 추가될 때, 채움 레벨이 증가될 수 있다; 스캔-라인이 출력 버퍼로부터 제거될 때, 채움 레벨은 감소될 수 있다. 이 모드에서, 필터 모듈은, 그 입력 및 출력 버퍼와 함께, FIFO로서 간주되며, 스캔-라인이 그 엔트리를 점유하고 FIFO의 깊이는 입력 및 출력 버퍼들을 위해 프로그램된 스캔-라인들의 개수에 의해 구성되는 FIFO(first-in-first-out)로서 간주될 수 있다.
- [0151] 일부 실시예에서, 또 다른 필터 모듈은, 필터 모듈의 입력 버퍼가 가득 채워지지 않는다면 FIFO에 스캔-라인을 추가할 수 있다. 소프트웨어는, 또 다른 필터 모듈이 입력 버퍼에 스캔-라인을 추가하는 것을 허용하기 전에 입력 버퍼의 채움 레벨을 체크할 수 있다. 결과적으로, 소프트웨어 또는 필터 모듈은 입력 버퍼와 연관된 채움 레벨을 증가시킬 수 있다. 출력 측에서, 소프트웨어는 출력 버퍼의 채움 레벨을 확인하거나, (예를 들어, FIFO 판독처럼, 필터의 출력 버퍼 내의 라인이, 또 다른 필터에 의해 처리된 후) 출력 버퍼의 채움 레벨을 감소시키기 전에 필터 모듈이 새로운 스캔-라인을 그 출력 버퍼에 추가했음을 나타내는 인터럽트 이벤트에 응답할 수 있다.
- [0152] 동기식 모드라고 불리는 제2 메커니즘은 각각의 필터 런을 명시적으로 스케줄링하는 소프트웨어에 의존한다. 각각의 필터 모듈의 시작 비트는, 소프트웨어가 즉시 필터 런을 시작하기 위해 기입할 수 있는 레지스터에서 제공될 수 있다. 이 메커니즘에 의해 시작될 때, 필터 모듈은 정확히 한 번만 실행될 수 있다.
- [0153] 일부 실시예에서, 필터 모듈은 인터럽트 요청을 수신할 때 인터럽트될 수 있다. 일부 경우에, 필터 모듈은, 외부 인터럽트 요청 라인에 맵핑되고 인터럽트 제어기로 라우팅되는 복수의 인터럽트 요청 소스를 가질 수 있다. 필터 모듈이 인터럽트를 플래그(flag)하고 그 인터럽트가 인에이블되면, 대응하는 외부 인터럽트 요청 라인이 플래깅될 수 있다.
- [0154] 일부 실시예에서, 복수의 인터럽트 요청 소스로는 다음과 같은 것들이 포함될 수 있다 :
- 입력 버퍼 채움 레벨 감소 인터럽트
 - 출력 버퍼 채움 레벨 증가 인터럽트
 - 프레임 완료 인터럽트
- [0155] 출력 버퍼 채움 레벨 증가 인터럽트는 또한, 필터 모듈이 동기 모드에서 동작하도록 구성될 때 필터 모듈이 그 필터 런을 완료했음을 나타내는 것으로 간주될 수 있다.
- [0156] 일부 실시예에서, 하드웨어 가속기(104)는 깊이 정보에 기초하여 필터링 연산을 적응시킬 수 있다. 예를 들어, 하드웨어 가속기(104)는 30 야드 이상 떨어진 객체와 연관된 픽셀들만을 조건부로 흐리게 처리하도록, 또는 5 야드를 초과하는 픽셀들은 10 야드를 초과하는 객체보다 덜 흐려질 수 있도록 구성될 수 있다.
- [0157] 도 4는 일부 실시예에 따른 깊이 정보에 기초하여 필터링 연산을 적합화할 수 있는 하드웨어 가속기를 나타낸다. 깊이-인식 하드웨어 가속기(402)는, 도 3의 하드웨어 가속기(104) 내의 모듈들에 추가하여, 깊이 맵 판독 클라이언트(404) 및 깊이 맵 모듈(406)을 포함한다. 깊이 맵 판독 클라이언트(404)는, 대응하는 이미지 내의 픽셀에 의해 표현되는 객체의 깊이를 나타내는 깊이 맵을 수신하도록 구성된다.
- [0158] 예를 들어, 라인 버퍼 판독 클라이언트(308)가 이미지의 스캔-라인을 수신할 때, 깊이 맵 판독 클라이언트(404)

4)는 이미지의 스캔-라인에 대응하는 깊이 맵을 수신하도록 구성될 수 있다.

[0162] 후속해서, 깊이 맵 판독 클라이언트(404)는 깊이 맵을 깊이 맵 모듈(406)에 제공할 수 있다. 깊이 맵의 해상도가 이미지의 스캔-라인의 해상도보다 낮을 때, 깊이 맵 모듈(406)은, 깊이 맵의 해상도를 스캔의 해상도와 정합시키기 위해 깊이 맵을 업-샘플링하도록 구성될 수 있다. 깊이 맵이 이미지의 스캔-라인과 시간-동기화되지 않을 때, 깊이 맵 모듈(406)은 깊이 맵과 스캔-라인을 동기화하도록 구성될 수 있다. 깊이 맵 모듈(406)은 후속해서, 라인 시작 제어기(310)가 데이터경로 파이프라인(304)의 동작을 제어할 수 있도록, 처리된 깊이 맵을 라인 시작 제어기(310)에 제공할 수 있다. 더 일반적으로, 산술 함수는, 비교기를 이용한 픽셀 위치에서의 깊이와 하나 이상의 임계값의 비교에 기초하여 또는 대안으로서, 바이패스 멀티플렉서를 이용한 비교기 출력 대신에 적용될 수 있는 각각의 픽셀과 연관된 2진 제어 비트를 직접적으로 이용하여, 조건부로 적용될 수 있다.

[0163] 전통적으로, 이미지 처리 동작을 위한 하드웨어 가속기는, 미리결정된 순서로 배열된 고정된 세트의 하드-웨어 이미지 신호 처리(ISP) 기능들을 포함한다. 도 5는 전통적인 하드웨어 가속기를 나타낸다. 전통적인 하드웨어 가속기(500)는 메모리 디바이스(504)로부터 이미지를 수신하고, 설계시에 미리결정된 순서로 ISP 기능들(502A-502H)을 이용하여 수신된 이미지를 처리한다. 도 5에 도시된 예에서, 하드웨어 가속기(500)는 예시된 순서로 8개의 ISP 기능을 이용하여 수신된 이미지를 처리한다. 이 접근법은 다소 유연하지 못하며 하드웨어 가속기(500)가 이용될 수 있는 응용 분야를 제한할 수 있다. 이미지 센서 기술은 빠르게 움직이고 있으며 모든 현재의 및 미래의 센서들에 단일의 고정된 ISP 파이프라인을 이용하는 것을 구상하기는 어렵다. 더욱이, ISP 기능이 이미지의 복수의 스캔-라인에 관해 필터를 동작시킬 때, ISP 기능은 충분한 수의 스캔-라인이 존재할 때까지 인입되는 스캔-라인을 버퍼에 저장해야 한다. 이들 버퍼들은 전형적으로 이미지의 해상도에 따라 크기조정된 RAM 디바이스를 이용하여 구현되며, 버퍼의 크기는 하드웨어 가속기(500)의 설계시에 미리결정된다. 따라서, ISP용 버퍼는 하드웨어 가속기(500)에 의해 처리될 수 있는 이미지 해상도에 관해 사실상 엄격한 제한을 강제할 수 있다. 추가로, 버퍼는 ISP 기능에 전용되기 때문에, 버퍼는 다른 시나리오에서(예를 들어, 소프트웨어에 의해) 이용될 수 없고 많은 양의 다이 면적을 소비할 수 있다.

[0164] 일부 실시예에서, 하드웨어 가속기(104)는 일반적인 공통 ISP 기능들을 체인화함으로써 전통적인 하드웨어 가속기의 비유연성을 해결한다. 종종, 하드웨어 가속기들간의 차이점은, 하드웨어 가속기에 의해 구현된 ISP 기능의 기능성이 아니라 ISP 기능이 기동되는 순서(일부 경우에는, 횟수)에 놓여 있다. 따라서, 하드웨어 가속기(104)는, 효율적으로 구현되는 하나 이상의 일반적인 공통 기능 모듈들을 체인화함으로써 원하는 기능을 수행하도록 구성될 수 있다.

[0165] 예를 들어, 콘볼루션 연산은 곱셈 및 합산으로서 표현될 수 있다. 마찬가지로, 유한 임펄스 응답(FIR) 필터링 연산도 역시 곱셈 및 합산으로서 표현될 수 있지만, FIR 필터링 연산이 곱셈 및 합산을 수행하는 순서는 콘볼루션 연산의 순서와 상이할 수 있다. 콘볼루션 연산과 FIR 필터링 연산간의 차이점에도 불구하고, 곱셈 연산 및 합산 연산은 콘볼루션 연산 및 FIR 필터링 연산에 대한 공통된 기능이다. 따라서, 하드웨어 가속기(104)는 동일한 곱셈 모듈 및 동일한 합산 모듈을 이용하여 콘볼루션 연산 및 유한 임펄스 응답 필터링 연산을 수행하도록 설계될 수 있다.

[0166] 일부 실시예에서, 일반적인 공통 기능들이 기동되는 순서는 소프트웨어를 이용하여 결정될 수 있다. 예를 들어, 소프트웨어는, 곱셈 모듈 및 합계 모듈을 기동하여 곱셈 모듈 및 합계 모듈을 상이한 순서로 체인화함으로써 콘볼루션 연산 또는 FIR 필터링 연산을 수행하도록, 하드웨어 가속기를 프로그램할 수 있다.

[0167] 도 6은 일부 실시예에 따른 일반적인 기능에 기초한 하드웨어 가속기를 나타낸다. 하드웨어 가속기(102)는, 복수의 일반 ISP 기능 모듈(602A-602H), 처리를 위해 이미지의 하나 이상의 스캔-라인을 수신하기 위한 데이터 수신기 모듈(604), 및 하나 이상의 일반 ISP 기능 모듈(602A-602H)에 의해 처리된 하나 이상의 스캔-라인을 출력하기 위한 데이터 출력 모듈(606)을 포함할 수 있다. 일부 실시예에서, 하나 이상의 일반 ISP 기능 모듈(602A-602H)은 구성 레지스터 및 제어 레지스터를 포함할 수 있다. 이들 레지스터들의 값은 소프트웨어를 이용하여 제어될 수 있다. 일부 실시예에서, 복수의 일반 ISP 기능 모듈(602A-602H)은 데이터경로 파이프라인(304)의 일부일 수 있다.

[0168] 일부 실시예에서, 일반 ISP 기능 모듈(602A-602H) 중 하나 이상은, 직접 메모리 액세스(DMA) 능력을 역시 포함하는 자급식(self-contained) 하드웨어 필터를 포함할 수 있다. 일반 ISP 기능 모듈(602A-602H) 중 하나 이상은 DMA 능력을 이용하여 메모리 패브릭(106) 내의 메모리 슬라이스로부터 및/또는 메모리 슬라이스에 데이터를 로딩 및/또는 저장할 수 있다. DMA 능력은 소프트웨어를 이용하여 제어될 수 있다.

- [0169] 일부 실시예에서, 데이터 수신기 모듈(604)은 이미지의 하나 이상의 스캔-라인을 회수하기 위한 DMA 모듈을 포함할 수 있다. 다른 실시예에서, 데이터 수신기 모듈(604)은 MIPI 모듈 등의 센서 인터페이스 모듈을 포함할 수 있다. 일부 실시예에서, 데이터 출력 모듈(606)은 이미지의 하나 이상의 처리된 스캔-라인을 저장하기 위한 DMA 모듈을 포함할 수 있다. 다른 실시예에서, 데이터 출력 모듈(606)은 디스플레이 디바이스를 포함할 수 있다.
- [0170] 일부 실시예에서, 하드웨어 가속기(102)는 ISP 테이블을 포함하는 메모리 패브릭(106)에 결합될 수 있다. ISP 테이블은 하나 이상의 버퍼(608)를 포함할 수 있다. 각각의 버퍼는 일반적인 ISP 기능 모듈(602A-602H) 중 하나에 대한 포인터를 포함할 수 있다. 메모리 패브릭(106)은 멀티-포트형 공통(또는 균일) 메모리를 포함할 수 있기 때문에, 복수의 디바이스가 ISP 테이블 내의 하나 이상의 버퍼(608)에 액세스하여 이용가능한 일반 ISP 기능 모듈을 식별할 수 있다.
- [0171] 일부 실시예에서, 프로세서(610) 상에서 실행되는 소프트웨어 ISP 기능(612A-612C)은 하나 이상의 일반 ISP 기능 모듈(602A-602H)을 하드웨어 가속기(102)에서 실행하도록 설계될 수 있다. 예를 들어, 소프트웨어 ISP 기능(612A)은, (1) 원하는 기능을 수행하기 위해 실행될 일반 ISP 기능 모듈(602A-602H)의 리스트 및 (2) 일반 ISP 기능 모듈(602A-602H)의 리스트가 실행되어야 하는 순서를 결정할 수 있다. 그 다음, 소프트웨어 ISP 기능(612A)은, 일반 ISP 기능 모듈(602A-602H)의 리스트에 대응하는 하나 이상의 버퍼(608)를 이용하여 일반 ISP 기능 모듈들을 체인화함으로써, 원하는 기능을 수행할 수 있다. 본질적으로, 하드웨어 가속기의 기능은, 소프트웨어에 의해 ISP 테이블 내의 버퍼(608)의 룩업(look-up)에서 결정될 수 있다.
- [0172] 일부 실시예에서, ISP 기능 모듈들의 입력 인터페이스는, 작은 메모리 맵핑된 FIFO(first-in-first-out) 버퍼에 의해 다른 ISP 기능 모듈들의 출력 인터페이스에 직접 결합될 수 있다. 도 7은 일부 실시예에 따른 ISP 기능 모듈들 사이의 통신을 위한 FIFO 버퍼를 포함하는 하드웨어 가속기를 나타낸다. ISP 기능 모듈(602)은, FIFO 버퍼(704) 및 메모리 패브릭(106)에 차례로 결합되는 메모리 버스 인터페이스(702)에 결합될 수 있다.
- [0173] 제1 ISP 기능 모듈(602A)이 이미지의 스캔-라인에 관한 그 동작을 완료하면, 제1 ISP 기능 모듈(602A)은 처리된 스캔-라인을 FIFO 버퍼(704)에 저장할 수 있다. 제1 ISP 기능 모듈(602A)이 추가 스캔-라인들을 계속 처리함에 따라, 제1 ISP 기능 모듈(602A)은 FIFO 버퍼(704)가 가득 찰 때까지 처리된 스캔-라인을 FIFO 버퍼(704)에 계속해서 저장할 수 있다. FIFO 버퍼(704)가 가득 차면, 제1 ISP 기능 모듈(602A)은, FIFO 버퍼(704)가 더 이상 가득 차 있지 않을 때까지 정지될 수 있다. 한편, 제2 ISP 기능 모듈(602B)은, FIFO 버퍼(704)가 비워질 때까지, 처리를 위해 FIFO 버퍼(704)로부터 처리된 스캔-라인을 회수할 수 있다. 사실상, 제1 ISP 기능 모듈(602A)은 데이터의 생산자로서 간주될 수 있다; 제2 ISP 기능 모듈(602B)은 데이터의 소비자로서 간주될 수 있다; FIFO 버퍼(704)는 중재자(arbitrator)로서 간주될 수 있다. 제2 ISP 기능 모듈(602B)은 메모리 패브릭(106) 내의 메모리 슬라이스에 비해 낮은 레이턴시를 갖는 FIFO 버퍼(704)로부터 처리된 스캔-라인을 회수할 수 있기 때문에, FIFO 버퍼(704)는 ISP 기능 모듈(602)들의 체인의 레이턴시를 감소시킬 수 있다.
- [0174] 일부 실시예에서, 컴퓨팅 디바이스(100)는 복수의 전력 아일랜드를 포함할 수 있다. 각각의 전력 아일랜드는 전용 전력 도메인과 연관될 수 있다. 따라서, 각각의 전력 아일랜드의 전원 전압은 독립적으로 제어될 수 있다. 예를 들어, 컴퓨팅 디바이스(100)는 소정 동작을 수행하기 위해 어떤 전력 아일랜드가 필요한지를 결정할 수 있고, 필요한 전력 아일랜드의 전원 전압만을 편온할 수 있다. 이러한 방식으로, 컴퓨팅 디바이스(100)는 누설 전력 소비를 감소시킬 수 있다.
- [0175] 일부 실시예에서, 컴퓨팅 디바이스(100)가, 전력 아일랜드가 현재 저전력 모드(예를 들어, 전원 전압이 제공되지 않음)에 있고 특정 동작을 위해 전력 아일랜드가 필요하다고 결정하면, 컴퓨팅 디바이스(100)는 전력 아일랜드에 대한 파워-업 시퀀스를 기동하고 전력 아일랜드에 전원 전압을 제공할 수 있다.
- [0176] 일부 실시예에서, 벡터 프로세서(102)들 각각은 고유의 전력 아일랜드와 연관될 수 있다. 일부 실시예에서, 하드웨어 가속기(104)는 고유의 전력 아일랜드와 연관될 수 있다. 일부 실시예에서, 메모리 패브릭(106)은 고유의 전력 아일랜드와 연관될 수 있다. 일부 실시예에서, 주변 디바이스(108)는 고유의 전력 아일랜드와 연관될 수 있다.
- [0177] 일부 실시예에서, 컴퓨팅 디바이스(100)는 인에이블 신호를 전력 아일랜드에 제공함으로써 파워-업 시퀀스를 기동할 수 있다. 인에이블 신호는 후속해서 전원 전압과 전력 아일랜드 사이에 위치하는 스위치들을 단음으로써, 전원 전압을 전력 아일랜드에 제공할 수 있다. 이 동작은 때때로 전원 게이팅이라고 지칭된다.
- [0178] 도 8은 일부 실시예에 따른 전력 아일랜드의 전원 게이팅을 나타낸다. 도 8은, 입력 데이터를 처리하기 위한

회로 블록을 포함할 수 있는 전력 아일랜드(802), 전원 전압 또는 접지 신호를 전력 아일랜드(802)에 제공하는 하나 이상의 스위치(804A-804B), 및 전력 아일랜드(802)가 입력 데이터를 처리할 준비가 될 때까지 입력 데이터를 유지하기 위한 입력 레지스터(806)를 도시한다. 일부 실시예에서, 입력 레지스터(806)가, 전력 아일랜드(802)가 입력 데이터를 처리할 준비가 되었음을 나타내는, 유효 신호 생성기(808)로부터 수신된 유효 신호를 수신할 때, 입력 데이터를 입력 아일랜드(802)에 제공하도록 트리거된다.

[0179] 일부 실시예에서, 컴퓨팅 디바이스(100)는, 전력 아일랜드의 전원 전압이 적절한 동작 전압에 도달했음을 나타내는 유효 신호를 생성하도록 구성된다. 유효 신호는, 전력 아일랜드 내의 회로가 원하는 동작을 수행하는데 이용될 수 있는 시간 인스턴스를 나타낼 수 있다. 유효 신호는 유효 신호 생성기(808)에 의해 생성될 수 있다.

[0180] 유효 신호 생성기(808)는 타이머를 이용하여 유효 신호를 생성할 수 있다. 예를 들어, 유효 신호 생성기(808)는, 인에이블 신호가 전력 아일랜드에 인가되는 시간 인스턴스를 결정할 수 있고, 타이머를 이용하여 미리결정된 시간 동안 대기한 다음, 유효 신호를 생성할 수 있다. 그러나, 전력 아일랜드의 전원 전압을 상승(ramp up)시키는데 걸리는 시간은, 프로세스, 전압 및 온도(PVT) 변동에 영향을 받기 때문에, 설계시에 미리결정된 양의 시간을 결정하는 것은 어렵다. PVT 변동을 해결하기 위해, 미리결정된 양의 시간은 최악의 경우의 PVT 코너를 수용하도록 종종 보수적으로(예를 들어, 충분히 커지도록) 설정되므로, 파워-업 시퀀스에 불필요하게 레이턴시를 추가할 수도 있다.

[0181] 이들 문제를 해결하기 위해, 일부 실시예에서, 유효 신호 생성기(808)는 유효 신호를 적응적으로 생성하도록 구성된다. 더 구체적으로는, 전력 아일랜드는, 전력 아일랜드에 제공된 인에이블 신호를 적응적으로 지연시킴으로써 유효 신호를 생성하도록 구성될 수 있다.

[0182] 도 9는 일부 실시예에 따른 유효 신호 생성기를 나타낸다. 유효 신호 생성기(808)는, 복수의 전력 스위치에 결합된 로직 셀들에 전원 전압을 제공하도록 구성된 복수의 전력 스위치를 포함할 수 있다. 일부 실시예에서, 전력 스위치는 각각의 로직 셀의 일부일 수 있다. 예를 들어, 전력 스위치는, 포지티브 전원과 직렬의 하나 이상의 P-채널 디바이스 및/또는 네거티브 전원(접지)과 직렬의 하나 이상의 N-채널 디바이스를 포함할 수 있다. 이들 전력 스위치들은 전력 아일랜드를 포함하는 로직 블록 전체에 걸쳐 분산될 수 있다. 도 9에서, 간소화를 위해, N 및 P- 채널 전력 스위치는 각각의 로직 셀과 연관된 단일의 전력-스위치 블록으로서 도시되어 있다.

[0183] 일부 실시예에서, 유효 신호 생성기(808)는 인에이블 신호를 데이지 체인의 전력 스위치들에 인가하고 인에이블 신호가 데이지 체인의 전력 스위치들의 끝에 도달할 때까지 대기할 수 있다. 일단 인에이블 신호가 전력 스위치의 데이지 체인 끝에 도달하고 나면, 전력 아일랜드 내의 모든 로직 셀들의 전력이 적절하게 온으로 되는 것이 보장된다. 따라서, 유효 신호 생성기(808)는, 유효 스위치로서, 데이지 체인의 전력 스위치들에 의해 지연된 인에이블 신호를 이용할 수 있다. 이 자체-교정 메커니즘(self-calibration mechanism)은 특정 컴퓨팅 디바이스의 임의의 프로세스-전압-온도(PVT) 변동을 적응적 포착할 수 있다. 이러한 방식으로, 컴퓨팅 디바이스는 전력 아일랜드가 파워-업하기 위한 긴 기간을 불필요하게 대기할 필요가 없다; 컴퓨팅 디바이스는 전력 아일랜드를 적절히 파워-업하는데 필요한 시간만 대기할 수 있다.

[0184] 일부 실시예에서, 전력 아일랜드는 항상 파워-온일 수 있다. 즉, 전력 아일랜드는 전원 전압이 공급되지 않는 저전력 모드에 진입하지 않도록 설계될 수 있다. 이러한 전력 아일랜드는, 상시-온 상태의 전력 아일랜드(always-on power island)라고 할 수 있다.

[0185] 일부 실시예에서, 상시-온 상태의 전력 아일랜드는 외부 신호를 모니터링하는데 이용될 수 있다. 예를 들어, 상시-온 상태의 전력 아일랜드는, GPIO(General-Purpose-Input-Output) 핀, 외부 인터페이스, 및/또는 저주파수 타이머 또는 파워-온 리셋 등의 내부 기능 블록들로부터의 신호를 모니터링하는데 이용될 수 있다. 이러한 방식으로, 컴퓨팅 디바이스(100)는 외부 신호를 분석하고, 외부 신호에 응답하기 위해 하나 이상의 전력 아일랜드가 파워-업될 필요가 있는지를 결정하며, 외부 신호에 응답하는데 필요한 전력 아일랜드만을 적응적으로 파워-업할 수 있다.

[0186] 도 10은 일부 실시예에 따른 이벤트 신호 모니터링 메커니즘을 나타낸다. 도 10은 상시-온 상태의 전력 아일랜드(802) 및 전력 관리 모듈(110)을 도시한다. 상시-온 상태의 전력 아일랜드(802)는 주변 디바이스(108)에 대한 전력 도메인을 포함할 수 있다. 상시-온 상태의 전력 아일랜드(802)는 저전력 모드에 진입하지 않기 때문에, 상시-온 상태의 전력 아일랜드(802) 내의 주변 디바이스(108)는 컴퓨팅 디바이스(100)의 클록과 비동기인 신호를 모니터링할 수 있다. 주변 디바이스(108)가 컴퓨팅 디바이스(100)가 응답해야 하는 이벤트 신호를 검출하면, 주변 디바이스(108)는 전력 관리 모듈(110)에 경고할 수 있다. 차례로, 전력 관리 모듈(110)은 컴퓨

팅 디바이스(100) 내의 전력 아일랜드들 중 어느 것이 온으로 되어야 하는지를 결정할 수 있다. 후속해서, 전력 관리 모듈(110)은 전력 아일랜드들 중 하나 이상이 온으로 되게 할 수 있다.

[0187] 일부 실시예에서, 주변 디바이스(108)는, 그 기능이 소프트웨어를 이용하여 정의될 수 있는, 소프트웨어 정의된 인터페이스를 포함할 수 있다. 더 구체적으로는, 주변 디바이스(108)는, SPI, I2C, I2S 또는 기타 임의의 적절한 프로토콜 등의 표준화된 인터페이스 프로토콜의 기능을 에뮬레이트할 수 있는 인터페이스 프로토콜 에뮬레이션(IPE; interface protocol emulation) 모듈을 포함할 수 있다. 소프트웨어 정의된 인터페이스는, 주변 디바이스(108)가 하나의 특정 인터페이스 프로토콜에 각각 전용되는 복수의 인터페이스를 유지하는 것 대신에 복수의 인터페이스 프로토콜을 수용하도록 프로그램될 수 있는 단일 소프트웨어 정의된 인터페이스만을 유지할 수 있기 때문에 유용하다. 단일의 소프트웨어 정의된 인터페이스는 복수의 전용 인터페이스에 비해 상당히 적은 다이 공간을 소비할 수 있기 때문에, 단일의 소프트웨어 정의된 인터페이스는 인터페이스와 연관된 비용을 크게 절감할 수 있다.

[0188] 도 11은 일부 실시예에 따른 소프트웨어 정의된 인터페이스를 나타낸다.

[0189] 도 11은, 컴퓨팅 디바이스(100)를 위한 일반 입/출력(I/O) 인터페이스(1104), IPE 모듈(1106), 및 내부 버스(1108)를 포함하는 소프트웨어 정의된 인터페이스를 나타낸다. 일반 입력/출력 인터페이스(1104)는, 센서 또는 카메라 모듈 등의 외부 디바이스와 통신하기 위한 인터페이스를 포함할 수 있다.

[0190] I/O 인터페이스(1104)의 기능은 IPE 모듈(1106)을 이용하여 구성될 수 있다.

[0191] 예를 들어, IPE 모듈(1106)이 I/O 인터페이스(1104)가 I2C 인터페이스로서 동작해야 한다고 결정할 때, IPE 모듈(1106)은, 외부 디바이스와의 통신을 위해 I2C 인터페이스 프로토콜을 이용하도록 I/O 인터페이스(1104)를 프로그램할 수 있다. 일부 실시예에서, IPE 모듈(1106)은 소프트웨어를 이용하여 프로그램될 수 있다. IPE 모듈(1106)은, IPE 모듈(1106)이, SPI, I2C, I2S 또는 기타 임의의 적절한 표준 등의 표준화된 인터페이스 프로토콜을 구현하도록 I/O 인터페이스(1104)를 구성할 수 있도록 프로그램될 수 있다.

[0192] 도 12는 일부 실시예에 따른 소프트웨어 정의된 인터페이스의 상세한 구현을 도시한다. 소프트웨어 정의된 인터페이스(1102)는, 범용 입/출력(GPIO) 인터페이스(1202) 및 그 레지스터(1204)를 포함할 수 있다. 호스트 프로세서는, GPIO 레지스터들(1204) 내의 비트들을 구성함으로써 GPIO(1202)의 동작을 제어할 수 있다. GPIO(1202)는, I/O 인터페이스(1104) 내의 핀들 중 일부를 제어하여, 가속도계, 주변 광 센서 또는 오디오 센서 등의 외부 디바이스와 통신할 수 있다.

[0193] 소프트웨어 정의된 인터페이스(1102)는 또한, IPE 모듈(1106) 및 그 레지스터(1206)를 포함할 수 있다. 호스트 프로세서는, IPE 레지스터(1206) 내의 비트들을 구성함으로써 IPE 모듈(1106)의 동작을 제어할 수 있다. IPE 모듈(1106)은, (1) 소프트웨어 정의된 인터페이스(1102)에 의해 구현될 인터페이스 프로토콜, 및 (2) 인터페이스 프로토콜을 구현하는데 이용되는 I/O 인터페이스 핀을 결정하도록 구성될 수 있다. 일단 IPE 모듈(1106)이 인터페이스 프로토콜을 구현하는데 이용되는 I/O 인터페이스 핀을 결정하고 나면, IPE 모듈(1106)은, 선택된 I/O 인터페이스 핀을 IPE 모듈(1106)에 멀티플렉싱하기 위해 제어 신호를 멀티플렉서(1208)에 전송할 수 있다. IPE 모듈(1106)은, I/O 인터페이스 핀이 인터페이스 프로토콜에 따라 제어 신호 및 데이터를 전송하게 함으로써 I/O 인터페이스 핀이 인터페이스 프로토콜을 에뮬레이트하게 할 수 있다.

[0194] 일부 실시예에서, 타이머(1214) 및/또는 프리스케일러(1216)는, IPE에 적절한 클록 신호를 제공하기 위해 고주파 기준 클록(예를 들어, 수백 메가-헤르츠의 범위)을 저주파수 클록(예를 들어, 수백 킬로-헤르쯔의 범위)으로 변환하는데 이용될 수 있다.

[0195] 일부 실시예에서, 프리스케일러(1216)로부터의 출력 클록의 주파수는 정수 값으로 곱해져 소정 인터페이스를 에뮬레이트할 수 있다. 예를 들어, 프리스케일러(1216)의 출력 클록이 500kHz에서 동작할 때, 프리스케일러(1216)로부터의 출력 클록의 주파수는 I2C 인터페이스를 에뮬레이트하기 위해 3으로 곱해질 수 있다. 이런 방식으로, IPE 로직을 동작시키고 I/O 핀에 접속된 출력 레지스터들을 샘플링하기 위해 500kHz 클록이 이용될 수 있다.

[0196] 일부 실시예에서, 주변 디바이스(108) 내의 IPE 모듈(1106)은, I/O 인터페이스(1104)의 입력 핀과 출력 핀 사이에서 바이패스를 수행함으로써, 처리 유닛들을 실제로 파워-업하지 않고도 컴퓨팅 디바이스(100)의 한 측 상의 입력과 컴퓨팅 디바이스(100)의 다른 측 상의 출력을 에뮬레이트하도록 구성될 수 있다. 이것은, I2C를 통해 컴퓨팅 디바이스(100)에 결합된 가속도계 등의 제1 외부 디바이스가 컴퓨팅 디바이스(100)의 처리 유닛들을 깨

우지 않고 애플리케이션 프로세서 SoC 등의 제2 외부 디바이스와 통신하는 것을 허용한다.

[0197] 소프트웨어 정의된 인터페이스(1102)는 또한, 이벤트 프로세서(1210) 및 그 레지스터(1212)를 포함할 수 있다. 이벤트 프로세서(1210)는, 외부 신호를 수신하고 컴퓨팅 디바이스(100)가 응답해야 하는 임의의 이벤트를 검출하도록 구성될 수 있다. 이벤트 프로세서(1210)의 기능은 EP 레지스터(1212)를 이용하여 구성될 수 있다. 일부 실시예에서, 일단 이벤트 프로세서(1210)가 응답할 이벤트를 검출하고 나면, 이벤트 프로세서(1210)는 이벤트에 응답하는데 필요한 벡터 프로세서(102), 하드웨어 가속기(104), 및/또는 메모리 패브릭(106)을 결정할 수 있고, 결정된 벡터 프로세서(102), 하드웨어 가속기(104), 및/또는 메모리 패브릭(106)과 연관된 전력 아일랜드에 신호를 전송할 수 있다.

[0198] 도 13은 일부 실시예에 따른 이벤트 프로세서를 나타낸다. 전술된 바와 같이, 이벤트 프로세서(1210)는 외부 디바이스와 통신하고 외부 디바이스로부터 신호를 수신할 수 있다. 신호는, 오디오 샘플, 가속도계 값, 주변 광 센서 값, 또는 GPIO 등의 통신 인터페이스를 통해 제공될 수 있는 기타 임의의 입력을 포함할 수 있다. 이벤트 프로세서(1210)는, 이벤트 또는 이벤트 시퀀스를 인식하기 위해 수신된 신호를 특정 구성과 비교하도록 구성될 수 있다. 일단 이벤트 프로세서(1210)가 이벤트 또는 이벤트 시퀀스를 인식하고 나면, 이벤트 프로세서(1210)는 컴퓨팅 디바이스(100) 내의 하나 이상의 컴포넌트로 하여금 저전력 모드로부터 깨어나 동작을 개시하게 할 수 있다.

[0199] 일부 실시예에서, 이벤트 프로세서(1210)는 하나 이상의 이벤트 필터(1302A-1302N)를 포함할 수 있다. 이벤트 필터(1302)는, 인터페이스(1104)로부터 입력 신호를 수신하고 특정 이벤트가 발생했는지를 결정하도록 구성된다. 특정 이벤트가 발생했다면, 이벤트 필터(1302)는 컴퓨팅 디바이스(100) 내의 복수의 전력 아일랜드 중 하나에 제어 신호 및/또는 전력 아일랜드 인레이블을 전송할 수 있다.

[0200] 도 14는 일부 실시예에 따른 이벤트 필터의 구현을 도시한다. 이벤트 필터(1302)는, 레지스터(1402), 비교기(1404) 및 불리언(Boolean) 연산자(1406)를 포함할 수 있다. 이벤트 필터(1302)는 이벤트 프로세서 제어 레지스터(1212) 및 타이머(1214)를 통해 제어될 수 있다.

[0201] 입력 레지스터(1402)는, 하나 이상의 외부 디바이스로부터 입력 신호를 수신하고 수신된 입력 신호를 비교기(1404)들의 뱅크에 제공하도록 구성될 수 있다. 비교기(1404)는, 부울, 정수, 고정 소수점 및 부동 소수점 표현을 포함한, 광범위한 입력 신호 표현을 지원하도록 구성될 수 있다.

[0202] 후속해서, 비교기(1404)들로부터의 출력들은, 특정 이벤트 또는 이벤트 시퀀스가 발생했는지를 결정하기 위하여 EP 타이머(1214)로부터의 타이머 값에 기초하여 논리적으로 결합될 수 있다. 일부 경우에, 비교기 출력들 사이의 특정 관계가 미리결정된 기간 동안 지속될 때 특정 이벤트 또는 이벤트 시퀀스가 발생했다고 간주된다. 일단 이벤트 필터(1302)가 특정 이벤트 또는 이벤트 시퀀스가 발생했다는 것을 결정하고 나면, 이벤트 필터(1302)는, 벡터 프로세서(102) 또는 하드웨어 가속기(104) 등의 컴퓨팅 디바이스(100) 내의 다른 컴포넌트들, 또는 주변 디바이스(108)에 결합된 외부 디바이스를 제어하기 위한 제어 신호를 출력할 수 있다.

[0203] 이벤트 프로세서(1210)는, 사용자가 전자 디바이스의 이용을 시작하는 이벤트를 검출하도록 구성될 수 있다. 이벤트 프로세서(1210)는 후속해서, 시작 이벤트에 응답하도록 컴퓨팅 디바이스(100) 내의 컴포넌트들을 터온할 수 있다. 예를 들어, 이벤트 프로세서(1210)는, 전자 디바이스가 주머니로부터 제거되었음을 나타낼 수 있는, 주변 광의 변화를 검출하도록 구성될 수 있다. 주변 광이 수 밀리 초 이상 동안 높은 레벨로 유지될 때, 이벤트 프로세서(1210)는 입력 오디오 신호에서 변화가 있는지를 결정하기 위해 오디오 입력을 체크할 수 있다. 이벤트 프로세서(1210)가 입력 오디오 신호의 변화를 검출할 때, 이벤트 프로세서(1210)는 컴퓨팅 디바이스(100) 내의 디지털 신호 프로세서가 음성 명령을 검출할 수 있게 한다. 이러한 방식으로, 이벤트 프로세서(1210)는, 컴퓨팅 디바이스(100) 내의 컴포넌트들이 저전력 모드로 유지되고 이벤트 또는 이벤트 시퀀스가 발생했을 때만 동작을 수행하는 것을 허용한다. 따라서, 이벤트 프로세서(1210)는 컴퓨팅 디바이스(100)의 평균 대기 전력을 상당히 감소시킬 수 있다.

[0204] 도 15는 일부 실시예에 따른 주변 디바이스의 바이패스 모드를 도시한다. 도 15에서, 컴퓨팅 디바이스(100)는 하나 이상의 전력 아일랜드가 저전력 모드(예를 들어, 전원 전압이 하나 이상의 전력 아일랜드에 인가되지 않음)에 있는 저전력 동작 모드에 있을 수 있다. 이 경우, IPE 모듈(1106)은, 입력 MIPI 레인(1502) 및 출력 MIPI 레인(1504) 등의 I/O 인터페이스(1104)의 입력 핀과 출력 핀 사이에서 바이패스를 수행하도록 구성될 수 있다. 이 예에서, 입력 MIPI 레인(1502)은 카메라 모듈에 결합되고 출력 MIPI 레인(1504)은 애플리케이션 프로세서에 결합된다. 따라서, 카메라 모듈은, 저전력 모드에 있는 하나 이상의 전력 아일랜드를 실제로 깨우지 않

고도 애플리케이션 프로세서에 결합될 수 있다.

[0205] 일부 실시예에서, 상이한 인터페이스 프로토콜들을 위한 주변 디바이스(108)들은, 컴퓨팅 디바이스(100)의 물리적 핀(또는 패드)을 공유할 수 있다. 예를 들어, 주변 디바이스(108)들은 제1 통신 프로토콜을 위한 제1 인터페이스 및 제2 통신 프로토콜을 위한 제2 인터페이스를 포함할 수 있다. 제1 인터페이스 및 제2 인터페이스는, 주변 디바이스(108)에 전용된 I/O 핀의 수를 감소시킬 수 있도록 물리적 I/O 핀을 시간-멀티플렉싱하도록 구성될 수 있다. 일부 경우에, 주변 디바이스(108)들은, 제1 및 제2 인터페이스들 내의 신호들과 물리적 핀들 사이의 맵핑을 포함하는 테이블을 포함할 수 있다.

[0206] 컴퓨팅 디바이스(100)가 카메라 및 디스플레이 등의 MIPI 디바이스의 범위에 접속되거나, 컴퓨팅 디바이스(100)가 카메라로서 "보이는" 애플리케이션 프로세서 또는 다른 디바이스들에 접속되는 응용에서, 컴퓨팅의 디바이스(100)의 구성은, MIPI 인터페이스 블록 및 연관된 핀의 개수의 관점에서, 설계시에 알려지지 않을 수도 있다. 이러한 이유로, MIPI I/O 핀 세트를 복수의 프로그램가능한 MIPI I/O 프로토콜 제어 블록에 접속하여, 특정 MIPI 이용 사례를 지원하는데 요구되는 MIPI 입력 및 출력의 수를 소프트웨어를 통해 런타임시에 구성될 수 있도록 하는 것이 유익하다.

[0207] 도 16은 일부 실시예에 따른 프로그램가능한 MIPI 인터페이스를 도시한다. 프로그램가능한 MIPI 인터페이스(1600)는, MIPI 매체 액세스 제어(MAC) 프로토콜 블록(1602), MIPI 전송기(1604), MIPI 수신기(1606), MIPI 전송기(1604) 또는 MIPI 수신기(1606) 중 하나로부터 신호들을 채널화하도록 구성된 멀티플렉서(1608), 차동 MIPI I/O 패드(1612)의 극성을 변경하도록 구성된 MIPI 극성 스위치(1610), 및 도 15에 관하여 예시된 바와 같이 I/O 인터페이스(1104)의 입력 핀과 출력 핀 사이에서 바이패스를 수행하기 위한 바이패스 멀티플렉서(1614) 및 바이패스 버퍼(1616)를 포함할 수 있다.

[0208] 일부 실시예에서, MIPI MAC 프로토콜 블록(1602)은, MIPI 전송기(1604) 및/또는 MIPI 수신기(1606)의 동작을 제어하여 MIPI 전송기(1604) 및/또는 MIPI 수신기(1606)의 동작이 MIPI 프로토콜을 따르게 하도록 설계된다.

[0209] 일부 실시예에서, 프로그램가능한 MIPI 인터페이스(1600)는, MIPI 전송기(1604) 또는 MIPI 수신기(1606) 중 하나만이 특정한 시간 인스턴스에서 MIPI I/O 패드(1612)를 통해 통신하는 것을 허용할 수 있다. 예를 들어, 프로그램가능한 MIPI 인터페이스(1600)는, 멀티플렉서(1608)를 통해 MIPI 전송기(1604) 또는 MIPI 수신기(1606) 중 하나만을 MIPI I/O 패드(1612)와 결합할 수 있다. 이런 방식으로, 외부 디바이스에 대해, MIPI I/O 패드(1612)는 양방향 MIPI 인터페이스로서 간주될 수 있다.

[0210] 일부 실시예에서, 프로그램가능한 MIPI 인터페이스(1600)는 MIPI 극성 스위치(1610)를 이용하여 차동 MIPI I/O 패드의 극성을 반전시킴으로써, 더 나은 임피던스 정합을 달성하거나 재작업없이 외부 PCB 설계의 오류를 정정하기 위하여 차동 MIPI I/O 패드의 극성이 런타임시에 역전될 수 있게 한다. 도 17은 일부 실시예에 따른 입/출력 인터페이스를 위한 극성 반전 메커니즘의 적용을 나타낸다. 도 17은 MIPI I/O 패드에 대한 극성 반전 메커니즘의 적용을 나타내지만, 극성 반전 메커니즘은 신호 라인의 차동 쌍을 이용하는 다양한 다른 인터페이스에서 이용될 수 있다.

[0211] 일부 실시예에서, 도 15와 관련하여 개괄적으로 설명된 바와 같이, 프로그램가능한 MIPI 인터페이스(1600)는, 컴퓨팅 디바이스(100)의 처리 유닛들이 파워-업되는 것을 요구하지 않고도 MIPI I/O 패드들(1612)이 출력들에 접속되는 것을 허용하는 MIPI 멀티플렉서(1614) 및 버퍼(1616)를 제공함으로써 저전력 MIPI 바이패스 모드를 제공할 수 있다. 이러한 피쳐는, 복수의 카메라 센서가 컴퓨팅 디바이스(100)에 접속되어 컴퓨터 비전 태스크를 수행하는 모드에서 바람직한 반면, 다른 이용 사례에서는 컴퓨팅 디바이스(100)가 요구되지 않고 애플리케이션 프로세서는 동일한 센서 세트를 이용하여 정지 이미지 또는 비디오 이미지 포착을 수행한다. 내부 MIPI 멀티플렉서(1614)의 제공에 의해, 이러한 이용 사례는 외부 컴포넌트를 이용하는 것이 아니라 내부 바이패스 멀티플렉서(1614)를 통해 지원될 수 있고, 칩들이 결합되는 PCB의 비용 및 복잡성을 크게 간소화한다.

[0212] 일부 실시예에서, 메모리 패브릭(106)은, 공간적 및 시간적 지역성의 관점에서 데이터 지역성을 활용하도록 설계된 캐시 메모리를 포함할 수 있다. 컴퓨팅 디바이스(100)가 외부 메모리 디바이스에 결합되지 않을 때, 메모리 패브릭(106)은, 벡터 프로세서(102) 및 하드웨어 가속기(104)가 캐시 메모리를 일반적인 메모리 디바이스로서 이용하는 것을 허용할 수 있다. 일부 실시예에서, 캐시 메모리는 섹션들로 분할되며, 각각의 섹션이 벡터 프로세서들 중 하나 또는 하드웨어 가속기들 중 하나에 의해 배타적으로 이용되도록 분할될 수 있다.

[0213] 일부 실시예에서, 메모리 패브릭(106)은, 컴퓨팅 디바이스(100)가 절전 모드에 있을 때 컴퓨팅 디바이스(100)의 상태 정보를 유지하도록 구성된다. 이러한 방식으로, 컴퓨팅 디바이스(100)가 다시 스위치 온될 때, 컴퓨팅 디

바이스(100)는, "웨이크-업" 절차와 연관된 지연이 감소될 수 있도록 적절한 디바이스들에 상태 정보를 재분배 할 수 있다.

[0214] 일부 경우에는, 상태 정보는 캐시 메모리에 유지된다. 이러한 경우, 상태 정보를 저장하는 캐시 메모리는 컴퓨팅 디바이스(100)가 절전 모드에 진입하더라도 전력이 파워 온될 수 있다. 상태 정보는, 부팅시 또는 런타임 동안에 로드되는 애플리케이션(들)의 바이너리를 포함할 수 있다. 상태 정보는 또한, 레지스터 설정, 동작 모드, 파이프라인 구성, 및 보통은 외부 비휘발성 메모리에 저장되고 파워다운으로부터 파워-업까지의 시퀀스의 이벤트시에 회수되는 부팅시 로드되고 런타임 동안에 수정되는 런타임 환경 설정 등의 구성 정보를 포함할 수 있다. 상태 정보는 또한, 이미지 데이터 등의 데이터, 및 다른 센서들로부터의 값들을 포함할 수 있다. 상태 정보는 또한, 보통은 파워다운으로부터 파워-업까지의 시퀀스의 이벤트에서 외부의 비휘발성 메모리로부터 회수 및 저장될 필요가 있는, 컴퓨팅 디바이스(100)와 다른 시스템 컴포넌트들 사이의 통신 프로토콜의 상태를 포함 할 수 있다.

[0215] 일부 실시예에서, 메모리 패브릭(106)은 하드웨어-기반 상호-배타(뮤텍스) 제어기(206)를 포함할 수 있다. 도 18은 일부 실시예에 따른 하드웨어-기반 뮤텍스 제어기를 갖는 메모리 패브릭을 나타낸다. 도 18은, 복수의 처리 유닛(1802A-1802P), 메모리 패브릭(106), 및 뮤텍스 제어기(206)를 도시한다. 처리 유닛(1802)은 벡터 프로세서(102) 또는 하드웨어 가속기(104)를 포함할 수 있다. 뮤텍스 제어기(206)는, 데이터 요소를 공유하는 처리 유닛(1802)들의 멀티태스킹을 조율하도록 구성된 하나 이상의 독립적으로 어드레싱 가능한 뮤텍스 요소를 포함할 수 있다. 더 구체적으로는, 뮤텍스 요소는, 메모리 패브릭(106) 또는 컴퓨팅 디바이스(100)의 다른 부분에 저장된 공유 데이터 요소를 제1 처리 유닛(1802A)에 잠금처리(lock)하여, 공유 데이터 요소를 역시 이용하는 다른 처리 유닛(1802P)이 제1 처리 유닛(1802A)이 공유 데이터 요소를 해제할 때까지 대기할 수 있게 하도록 구성될 수 있다. 뮤텍스 제어기(206)는 메모리 패브릭(106) 내에 존재하기 때문에, 공유된 버스 또는 다른 수단을 이용하는 것과 비교할 때, 공유 자원을 해제하거나 잠금처리하는 시간이 감소된다.

[0216] 전통적으로, 뮤텍스 제어기가 공유 자원에 대한 배타적 액세스 요청을 수신할 때, 뮤텍스 제어기는 그 요청에 즉시 응답하여, 요청측 처리 유닛이 공유 자원에 대한 배타적 액세스를 얻을 수 있는지를 표시한다. 따라서, 요청측 처리 유닛이 배타적 액세스를 얻지 못한다면, 요청측 처리 유닛은, 요청측 처리 유닛이 뮤텍스 제어기로부터 배타적 액세스를 수신할 때까지, 계속해서 뮤텍스 제어기에게 요청해야 한다. 이것은, 전통적인 뮤텍스 제어기와 처리 유닛 사이의 버스 상의 트래픽을 증가시킬 수 있다.

[0217] 이 문제를 해결하기 위해, 일부 실시예에서, 처리 유닛(1802A)이 공유 자원에 대한 배타적 액세스를 요청하는 배타적 액세스 요청을 전송할 때, 뮤텍스 제어기(206)는 자체적으로 요청 상태를 모니터링할 수 있다. 일단 뮤텍스 제어기(206)가 처리 유닛(1802A)이 배타적 액세스를 허가받았다고 결정하고 나면, 뮤텍스 제어기(206)는, 처리 유닛(1802A)이 공유 자원에 대한 배타적 액세스를 갖는다는 것을 나타내는 확인응답 메시지를 처리 유닛(1802A)에 전송할 수 있다. 이러한 방식으로, 처리 유닛(1802A)은 처리 유닛(1802A)이 배타적 액세스를 수신할 때까지 배타적 액세스 요청을 복수회 전송할 것이 요구되지 않는다; 처리 유닛(1802A)은 배타 액세스 요청을 한번만 전송할 수 있고 뮤텍스 제어기(206)로부터 배타적 액세스를 수신하기를 기다릴 수 있다. 이러한 메시징 메커니즘은 메모리 패브릭(106) 상의 통신 부하를 감소시킬 수 있다.

[0218] 일부 실시예에서, 메모리 패브릭(106)은, 처리 유닛들 사이에서 통신을 제공하는 유연한 버스 아키텍처를 포함 할 수 있다. 종종, 처리 유닛들 사이의 통신을 위한 인터페이스는, FIFO(First-In-First-Out) 등의 버퍼를 포함한다.

[0219] 예를 들어, 제1 처리 유닛이 제2 처리 유닛에 메시지를 전송할 준비가 되면, 제1 처리 유닛은, 제2 처리 유닛에 할당된 버퍼에 메시지를 전송할 수 있다. 제2 처리 유닛이 메시지를 수신할 준비가 되면, 제2 처리 유닛은 버퍼로부터 메시지를 회수할 수 있다.

[0220] 그러나, 종래의 인터페이스 내의 버퍼는 제한된 저장 용량을 갖는다. 따라서, 전통적인 인터페이스 내의 버퍼는 종종 제어 메시지를 저장하는 것으로 제한되며, 이미지 및 비디오 데이터 등의 많은 양의 데이터를 수용할 수 없다. 또한, 각각의 버퍼는 처리 유닛들 중 하나에 영구적으로 할당된다. 따라서, 제1 처리 유닛에 할당된 제1 버퍼는 오버플로우되는 반면, 제2 처리 유닛에 할당된 제2 버퍼는 비어있을 수 있다. 따라서 버퍼의 용량은 시스템 레벨에서 완전히 활용되지 않을 수도 있다.

[0221] 메모리 패브릭(106)은, 버퍼의 용량을 증가시키고 통신을 위한 실시간 필요성에 기초하여 처리 유닛에 버퍼를 동적으로 할당함으로써 종래의 인터페이스의 이러한 단점을 해결한다. 메모리 패브릭(106)은, 버퍼를 생성, 관

리 및 해제하기 위한 유연한 메커니즘을 제공한다. 베피는 프로세스의 지속 기간 동안 생성될 수 있고, 일단 프로세스가 완료되면 해제될 수 있다. 해제된 베피는, 소프트웨어 프로그램 제어하에 다른 애플리케이션 또는 처리 유닛들을 위해 이용될 수 있다.

[0222] 도 19는 일부 실시예에 따른 베피의 동적 할당을 나타낸다. 메모리 패브릭(106)은, 복수의 베피(1902A-1902P)를 포함할 수 있고, 그 각각은 베터 프로세서 또는 하드웨어 가속기 등의 처리 유닛들 중 하나에 배타적으로 할당될 수 있다. 어떤 경우에는, 복수의 베피(1902)가 동일한 처리 유닛에 할당될 수 있다.

[0223] 일부 실시예에서, 복수의 베피(1902)는, 분할되어 처리 유닛들 중 하나에 배타적으로 할당될 수 있는 베피들의 저장소의 일부일 수 있다. 저장소는 메모리 패브릭(106)으로부터의 메모리 슬라이스를 포함할 수 있다. 일부 실시예에서, 복수의 베피(1902) 각각은 동일한 용량을 가질 수 있다. 다른 실시예에서, 베피(1902)들 중 하나 이상은 가변 용량을 가질 수 있다. 예를 들어, 제1 처리 유닛(1802N)이 작은 수의 제어 메시지를 제2 처리 유닛(1802C)에 전송하려고 시도할 때, 메모리 패브릭(106)은 작은 베피(1902C)를 제2 처리 유닛(1802C)에 할당하여 제2 처리 유닛 작은 수의 제어 메시지를 수신할 수 있게 할 수 있다. 그러나, 제1 처리부(1802N)가 대량의 비디오 데이터를 제2 처리부(1802M)에 전송하려고 시도할 때, 메모리 패브릭(106)은 큰 용량을 갖는 베피를 제2 처리 유닛(1802M)에 할당하여 제2 처리 유닛(1802M)이 대량의 비디오를 수신할 수 있게 할 수 있다.

[0224] 일부 실시예에서, 복수의 베피(1902)들 중 하나 이상은, 디바이스(시스템-온-칩) 설계시에 예측될 수 있는 USB, MIPI 또는 Ethernet을 포함하는 통신 인터페이스 등의 특정 애플리케이션들과 연관될 수 있다.

[0225] 일부 실시예에서, 전력 관리 모듈(110)은, 로직 회로 및 메모리 디바이스에 상이한 전원 전압을 제공하도록 구성될 수 있다. 도 20은 일부 실시예에 따른 로직 회로 메모리 디바이스에 상이한 전압들을 제공하는 전력 관리 메커니즘을 나타낸다. 단일 전력 아일랜드(2002A)는 로직 회로 영역(2004) 및 메모리 영역(2006)을 포함할 수 있다. 전력 관리 모듈(110)은, 로직 회로 영역(2004)에 제1 전압 V₁를 제공하고 메모리 영역(2006)에 제2 전압 V₂를 제공하도록 구성될 수 있다. 일부 실시예에서, 제1 전압 및 제2 전압은 상이한 전력 조정기에 의해 제공될 수 있다. 따라서, 제1 전압과 제2 전압은 독립적으로 제어될 수 있다.

[0226] 일부 실시예에서, 로직 회로 영역(2004) 및 메모리 영역(2006)은 독립적으로 저전력 모드로 진입할 수 있다. 예를 들어, 전력 관리 모듈(110)은 로컬 스위치(2008, 2010)를 이용하여 각각 로직 회로 영역(2004) 및 메모리 영역(2006)에 대한 전원 전압을 차단할 수 있다. 일부 실시예에서, 전력 관리 모듈(110)은 전역 스위치(2012)를 이용하여 하나 이상의 전력 아일랜드(2002A, ..., 2002N) 내의 메모리 영역(2006)으로의 전원 전압을 차단할 수 있다.

[0227] 일부 실시예에서, 메모리 패브릭(106)은 직접 메모리 액세스(DMA) 엔진을 포함할 수 있다. DMA 엔진은, DMA 데이터 구조의 이중 연결 리스트를 포함하는 동작 리스트를 유지할 수 있다. 각각의 DMA 데이터 구조는 DMA 엔진에 의해 수행될 특정한 동작을 지시한다. DMA 데이터 구조는 DMA 엔진이 DMA 데이터 구조와 연관된 동작들을 수행해야 하는 순서로 유지된다.

[0228] 동작 리스트는 DMA 데이터 구조들의 이중 링크된 리스트를 포함하기 때문에, 이중 링크된 리스트에 의해 표현된 동작들의 시퀀스에 대한 DMA 동작을 제거하는데 상당한 시간이 걸린다. 일부 실시예에서, DMA 엔진은 DMA 데이터 구조가 실행되어야 하는지를 나타내는 베피를 유지함으로써 이 문제를 해결할 수 있다. 베피 내의 각각의 비트는 연관된 DMA 데이터 구조에 대한 인에이블 신호로서 간주될 수 있다.

[0229] 도 21은 일부 실시예에 따른 베피-기반 DMA 데이터 구조 인에이블 메커니즘을 구현하는 DMA 엔진을 나타낸다. DMA 엔진은 복수의 DMA 데이터 구조(2104)를 갖는 동작 리스트(2102)를 포함한다. 복수의 DMA 데이터 구조(2104)는 이중 링크된 리스트로서 서로 결합될 수 있다. DMA 엔진은 또한, 인에이블 베피(2106)를 포함한다. 인에이블 베피(2106)는 복수의 비트를 포함할 수 있다. 인에이블 베피(2106) 내의 비트 수는 동작 리스트(2102) 내의 DMA 데이터 구조의 수와 동일할 수 있다. 인에이블 베피(2106) 내의 각각의 비트는 그 비트와 관련된 DMA 데이터 구조가 인에이블되는지를 나타낼 수 있다. 예를 들어, 베피의 제1 비트가 "1"일 때, DMA 엔진은 제1 DMA 데이터 구조가 인에이블되는 것을 결정하고 제1 DMA 데이터 구조를 실행할 수 있다. 베피 내의 제2 비트가 "0"일 때, DMA 엔진은 제2 DMA 데이터 구조가 인에이블되는 것을 결정하고 제2 DMA 데이터 구조를 실행하지 않을 수 있다. 이러한 방식으로, DMA 엔진은, 동작 리스트로부터 DMA 데이터 구조를 실제로 제거하지 않고 동작 리스트 내의 DMA 데이터 구조들의 서브셋을 선택적으로 실행할 수 있다. DMA 엔진은 DMA 데이터 구조를 제거할 필요가 없으므로, 하나 이상의 DMA 데이터 구조를 디스에이블하는 것과 연관된 지연이 작을 수 있다.

- [0230] 일부 실시예에서, 병렬 컴퓨팅 디바이스(100)는 전자 디바이스 내에 존재할 수 있다. 도 22는 일부 실시예에 따른 컴퓨팅 디바이스를 포함하는 전자 디바이스를 나타낸다. 전자 디바이스(2200)는, 프로세서(2202), 메모리(2204), 하나 이상의 인터페이스(2206), 및 컴퓨팅 디바이스(100)를 포함할 수 있다.
- [0231] 전자 디바이스(2200)는, 컴퓨터 판독 가능한 매체, 플래시 메모리, 자기 디스크 드라이브, 광 드라이브, 프로그램가능한 판독 전용 메모리(PROM), 및/또는 판독 전용 메모리(ROM) 등의 메모리(2204)를 가질 수 있다. 전자 디바이스(2200)는, 명령어들을 처리하고 메모리(2204)에 저장될 수 있는 소프트웨어를 실행하는 하나 이상의 프로세서들(2202)로 구성될 수 있다. 프로세서(2202)는 또한, 메모리(2204) 및 인터페이스(2206)와 통신하여 다른 디바이스와 통신할 수 있다. 프로세서(2202)는, CPU, 애플리케이션 프로세서, 및 플래시 메모리, 또는 RISC(reduced instruction set computing) 프로세서를 결합하는 시스템 온 칩(system-on-a-chip) 등의 임의의 적용가능한 프로세서일 수 있다.
- [0232] 메모리(2204)는, 비일시적 컴퓨터 판독가능한 매체, 플래시 메모리, 자기 디스크 드라이브, 광학 드라이브, 프로그램가능한 판독 전용 메모리(PROM), 판독 전용 메모리(ROM), 또는 기타 임의의 메모리 또는 메모리들의 조합일 수 있다. 소프트웨어는, 컴퓨터 명령어 또는 컴퓨터 코드를 실행할 수 있는 프로세서에서 실행될 수 있다. 프로세서는 또한, ASIC(Application Specific Integrated Circuit), PLA(Programmable Logic Array), FPGA(Field Programmable Gate Array) 또는 기타 임의의 집적 회로를 이용하여 하드웨어로 구현될 수도 있다.
- [0233] 인터페이스(2206)는 하드웨어 또는 소프트웨어로 구현될 수 있다. 인터페이스(2206)는, 텔레비전에 대한 원격 제어 디바이스 등의 로컬 소스뿐만 아니라 네트워크로부터 데이터 및 제어 정보 양쪽 모두를 수신하는데 이용될 수 있다. 전자 디바이스는 또한, 키보드, 터치 스크린, 트랙볼, 터치 패드 및/또는 마우스 등의 다양한 사용자 인터페이스를 제공할 수 있다. 전자 디바이스는 또한, 일부 실시예에서 스피커 및 디스플레이 디바이스를 포함할 수 있다.
- [0234] 일부 실시예에서, 컴퓨팅 디바이스(100) 내의 벡터 프로세서(102) 및 하드웨어 가속기(104) 등의 처리 유닛은, 컴퓨터 명령어 또는 컴퓨터 코드를 실행할 수 있는 집적 칩을 포함할 수 있다. 프로세서는 또한, ASIC(Application Specific Integrated Circuit), PLA(Programmable Logic Array), FPGA(Field Programmable Gate Array) 또는 기타 임의의 집적 회로를 이용하여 하드웨어로 구현될 수도 있다.
- [0235] 일부 실시예에서, 컴퓨팅 디바이스(100)는 시스템 온 칩(SOC)으로서 구현될 수 있다. 다른 실시예에서, 병렬 컴퓨팅 디바이스 내의 하나 이상의 블록은 별개의 칩으로서 구현될 수 있고, 병렬 컴퓨팅 디바이스는 시스템 인 패키지(SIP; system in package)로 팩키징될 수 있다. 일부 실시예에서, 병렬 컴퓨팅 디바이스(400)는 데이터 처리 애플리케이션에 이용될 수 있다. 데이터 처리 애플리케이션은 이미지 처리 애플리케이션 및/또는 비디오 처리 애플리케이션을 포함할 수 있다. 이미지 처리 애플리케이션은 이미지 필터링 연산을 포함하는 이미지 처리 프로세스를 포함할 수 있다; 비디오 처리 애플리케이션은, 비디오 디코딩 동작, 비디오 인코딩 동작, 비디오 내의 움직임 또는 객체를 검출하기 위한 비디오 분석 동작을 포함할 수 있다. 본 발명의 추가 애플리케이션은, 이미지, 객체 또는 비디오의 시퀀스에 기초한 머신 학습 및 분류와, 깊이 인에이블형 카메라를 포함하는 복수의 카메라 뷔로부터 지오메트리를 추출하고, GPU에 의한 후속 버텍스 쉐이딩을 위해 (예를 들어, 포인트-클라우드를 통해) 와이어 프레임 지오메트리를 추출하는 대상으로서의 복수의 뷔로부터 피쳐들을 추출하는 증강 현실 애플리케이션을 포함한다.
- [0236] 전자 디바이스(2200)는 셀룰러 전화 등의 모바일 디바이스를 포함할 수 있다. 모바일 디바이스는, 복수의 액세스 기술 및 유선 통신 네트워크를 이용하여 복수의 무선 액세스 네트워크와 통신할 수 있다. 모바일 디바이스는, 워드 처리, 웹 브라우징, 게이밍, 전자-서적 능력, 운영 체제, 및 전체 키보드 등의, 고급 능력을 제공하는 스마트 폰일 수 있다. 모바일 디바이스는, Symbian OS, iPhone OS, RIM사의 Blackberry, Windows Mobile, Linux, Palm WebOS 및 Android 등의 운영 체제를 실행할 수 있다. 스크린은 모바일 디바이스에 데이터를 입력하는데 이용할 수 있는 터치 스크린일 수 있고, 스크린은 전체 키보드 대신에 이용될 수 있다. 모바일 디바이스는, 통신 네트워크 내의 서버에 의해 제공되는 애플리케이션을 실행하거나 애플리케이션과 통신하는 능력을 가질 수 있다. 모바일 디바이스는 네트워크 상의 이들 애플리케이션으로부터 업데이트 및 기타의 정보를 수신 할 수 있다.
- [0237] 전자 디바이스(2200)는 또한, 텔레비전(TV), 비디오 프로젝터, 셋탑 박스 또는 셋탑 유닛, 디지털 비디오 레코더(DVR), 컴퓨터, 넷북, 랩탑, 태블릿 컴퓨터, 및 네트워크와 통신할 수 있는 기타 임의의 청각적/시각적 디바이스를 포함할 수 있다. 전자 디바이스는 또한, 글로벌 위치결정 좌표, 프로파일 정보, 또는 기타의 위치 정보

를 그 스택 또는 메모리에 유지할 수 있다.

[0238] 본 명세서에서 수 개의 상이한 배열들이 설명되었지만, 각각의 피쳐들은 이점을 달성하기 위해 다양한 형태로 함께 유익하게 결합될 수 있다는 것을 이해할 것이다.

[0239] 상기 명세서에서, 애플리케이션이 특정 예들을 참조하여 설명되었다. 그러나, 첨부된 특허청구범위에 개시된 본 발명의 더 넓은 사상과 범위로부터 벗어나지 않고 다양한 수정 및 변경이 이루어질 수 있다는 것은 명백할 것이다. 예를 들어, 접속은, 예를 들어 중간 디바이스를 통해, 각각의 노드, 유닛 또는 디바이스로부터 또는 이들에 신호를 전송하기에 적합한 임의의 타입의 접속일 수 있다.

[0240] 따라서, 목시적으로 또는 달리 언급되지 않는 한, 접속은 예를 들어 직접 접속 또는 간접 접속일 수 있다.

[0241] 본 명세서에 도시된 아키텍쳐는 단지 예시적인 것이며, 사실상 동일한 기능을 달성하는 많은 다른 아키텍쳐가 구현될 수 있다는 것을 이해해야 한다. 추상적이지만 여전히 명확한 의미에서, 동일한 기능을 달성하기 위한 컴포넌트들의 임의의 배열은 원하는 기능이 달성되도록 사실상 "연관"되어 있다. 따라서, 특정한 기능을 달성하도록 여기서 결합된 임의의 2개의 컴포넌트는, 아키텍쳐 또는 중간 컴포넌트와는 관계없이, 원하는 기능이 달성되도록 서로 "연관"되어 있는 것으로 간주될 수 있다. 마찬가지로, 이와 같이 연관된 임의의 2개의 컴포넌트는 또한, 원하는 기능을 달성하도록 "동작적으로 접속된" 또는 "동작적으로 결합된" 것으로 볼 수 있다.

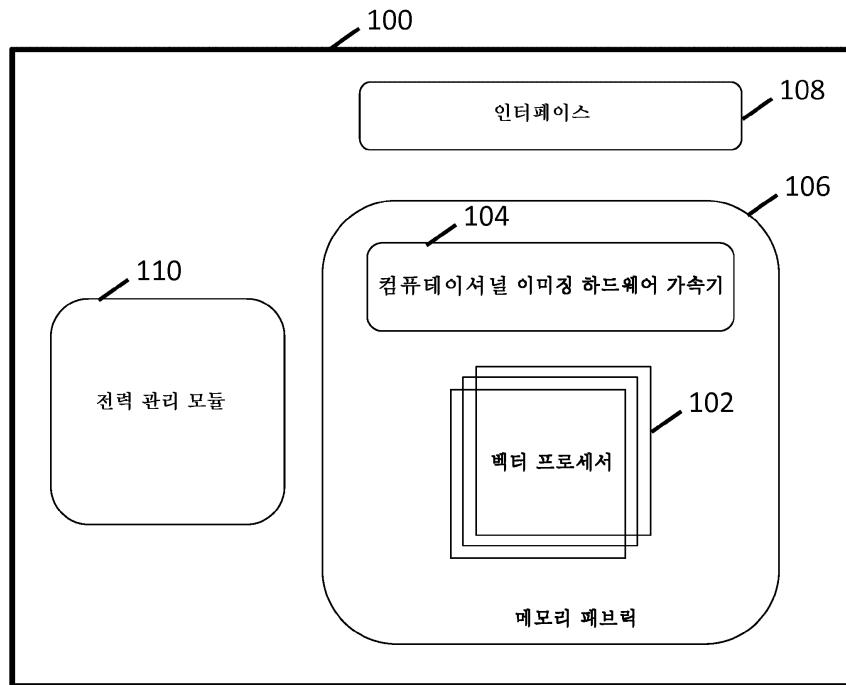
[0242] 또한, 본 기술분야의 통상의 기술자라면, 전술된 동작들의 기능 사이의 경계는 단지 예시적인 것이라는 것을 인식할 것이다. 복수의 동작들의 기능은 단일의 동작으로 결합될 수 있고, 및/또는 단일의 동작의 기능은 추가적인 동작들로 분산될 수 있다. 또한, 대안적 실시예들은 특정 동작의 복수의 사례를 포함할 수 있고, 동작들의 순서는 다양한 다른 실시예에서 변경될 수 있다.

[0243] 그러나, 다른 수정, 변형 및 대안도 역시 가능하다. 따라서, 본 명세서 및 도면은 제한적 의미라기보다는 예시적인 의미로 간주되어야 한다.

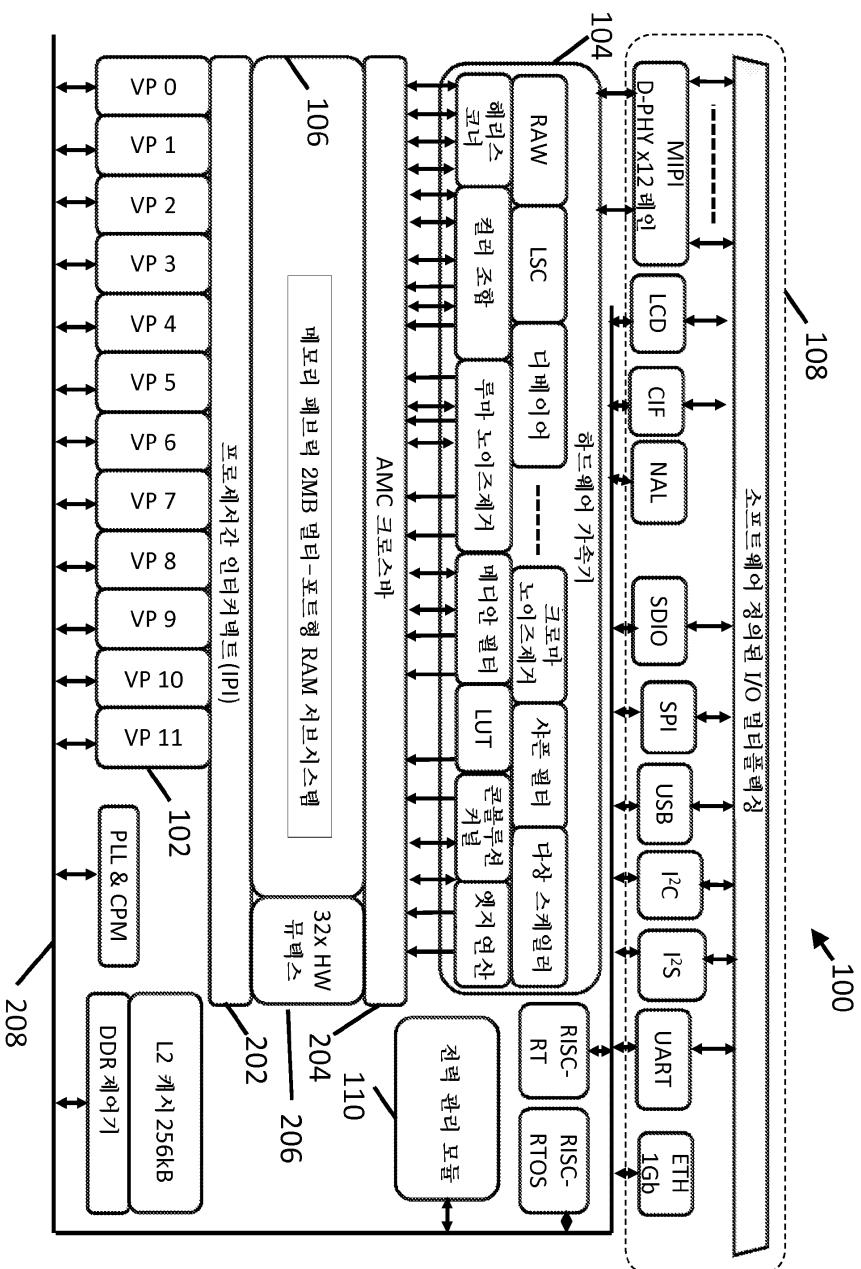
[0244] 청구항에서, 괄호안의 임의의 참조 부호는 청구항을 제한하는 것으로 해석되어서는 안된다. 용어 "~을 포함하는"은 청구항에 열거된 것들 이외의 다른 요소들이나 단계들의 존재를 배제하지 않는다. 또한, 본 명세서에서 사용되는 용어 "한(a)" 또는 "한(an)"은, 하나 또는 그 초과로서 정의된다. 또한, 청구항에서 "적어도 하나" 및 "하나 이상" 등의 도입 문구의 사용은, 부정 판사 "a" 또는 "an"에 의한 또 다른 청구항 요소의 도입이, 이러한 도입된 청구항 요소를 포함하는 임의의 특정한 청구항을, 그 청구항이 "하나 이상" 또는 "적어도 하나"라는 도입 문구와 "a" 또는 "an" 등의 부정 판사를 포함하는 경우에도, 단 하나의 이러한 요소만을 포함하는 발명으로 제한한다는 것을 암시하는 것으로 해석되어서는 안된다. 이것은 정판사의 사용에 대해서도 마찬가지로 적용된다. 달리 언급되지 않는 한, "제1" 및 "제2" 등의 용어는, 이러한 용어들이 기술하는 요소들간을 임의적으로 구분하기 위해 사용되는 것이다. 따라서, 이를 용어들은 이와 같은 요소들의 시간적 또는 기타의 우선순위를 나타내려고 의도한 것은 아니다. 소정의 수단들이 서로 다른 청구항들에서 인용되고 있다는 단순한 사실이, 이점을 얻기 위해 이를 수단들의 조합이 사용될 수 없다는 것을 나타내는 것은 아니다.

도면

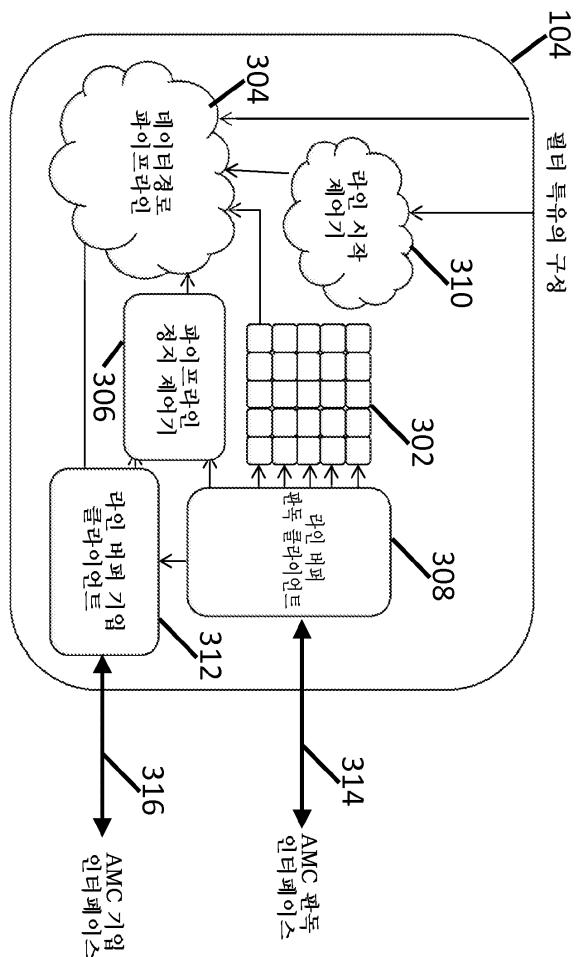
도면1



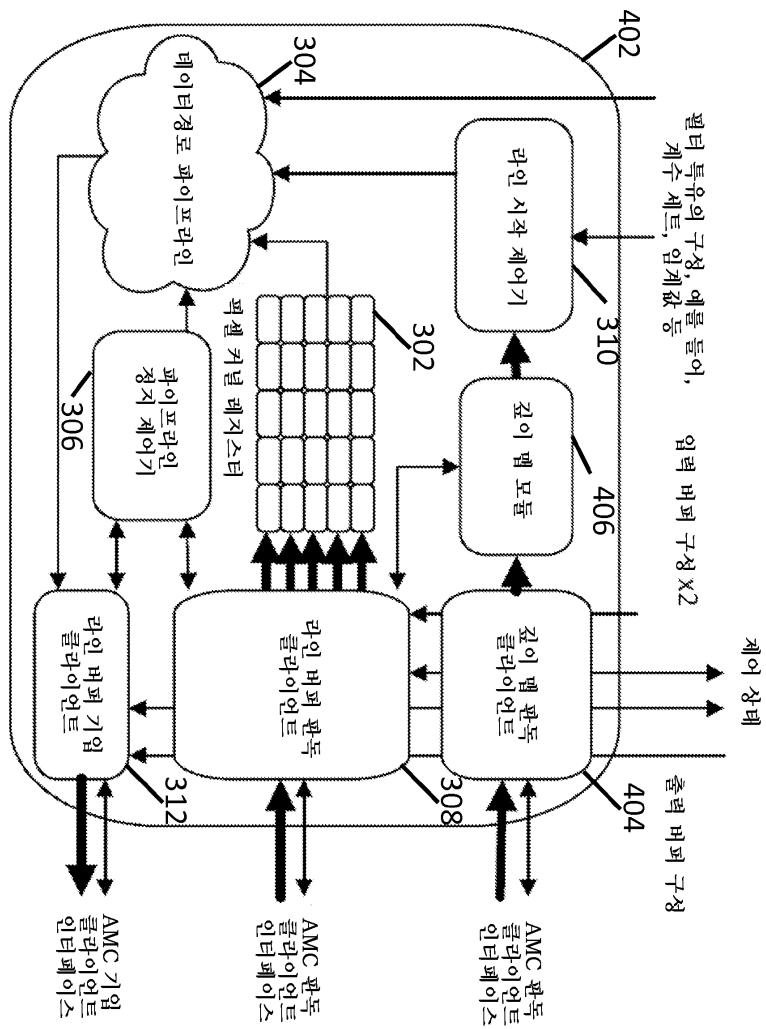
도면2



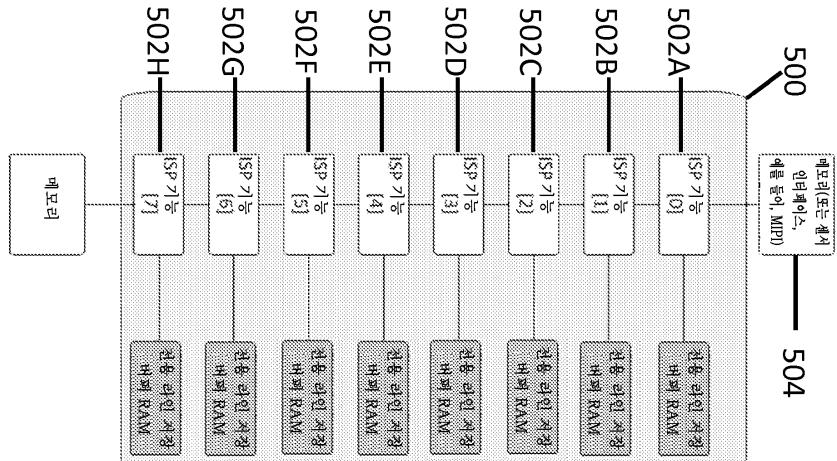
도면3



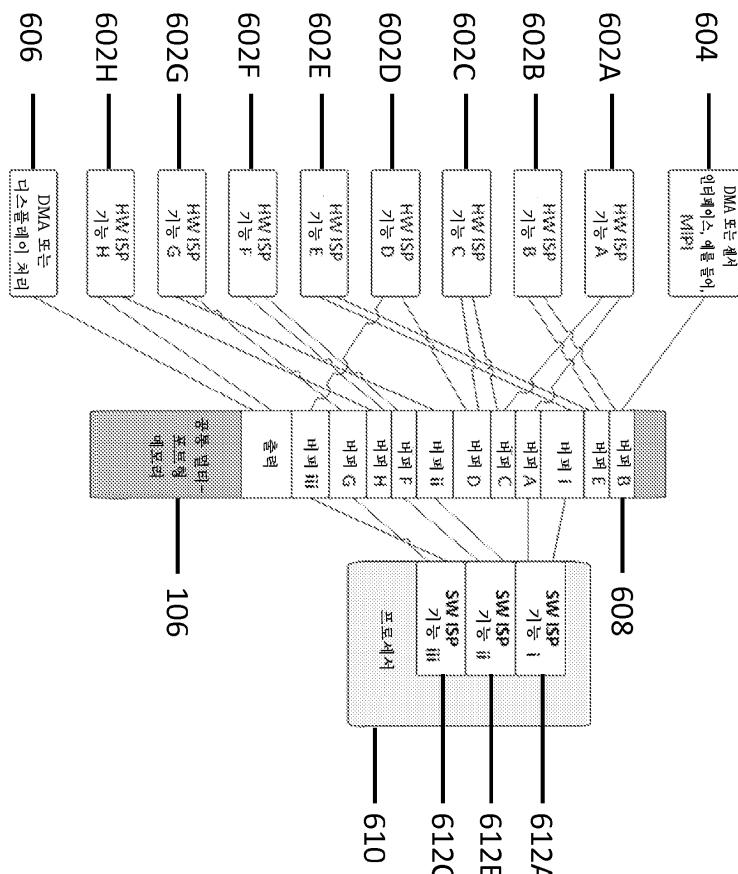
도면4



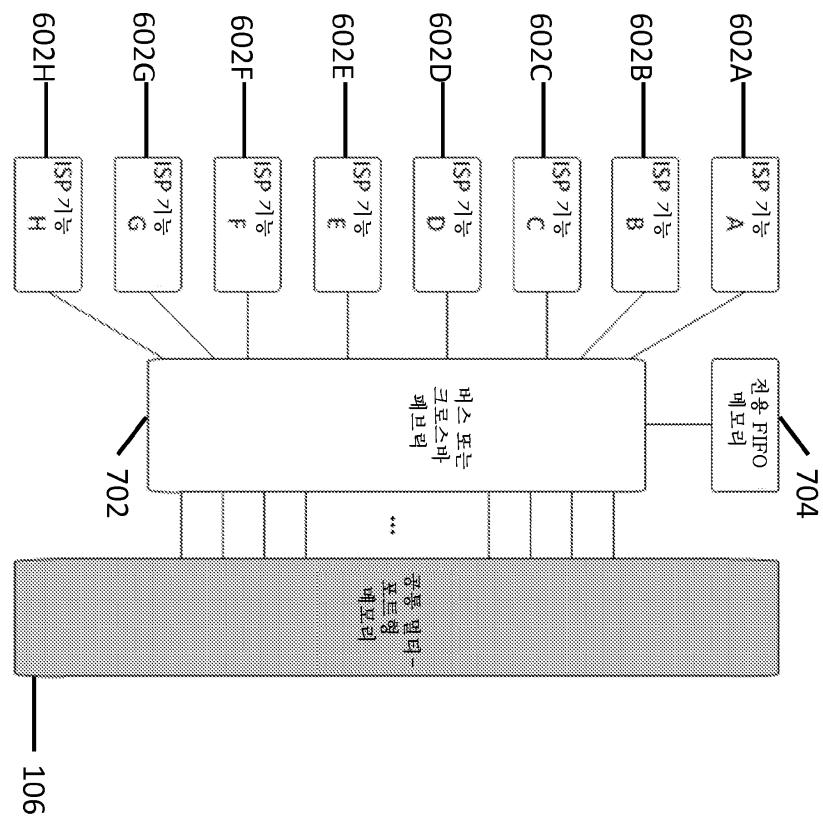
도면5



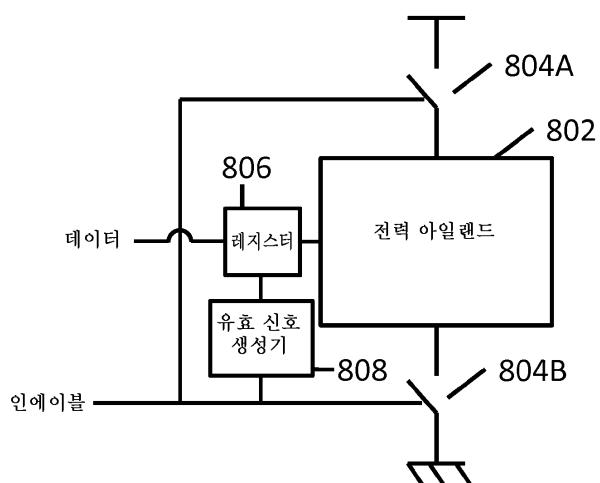
도면6



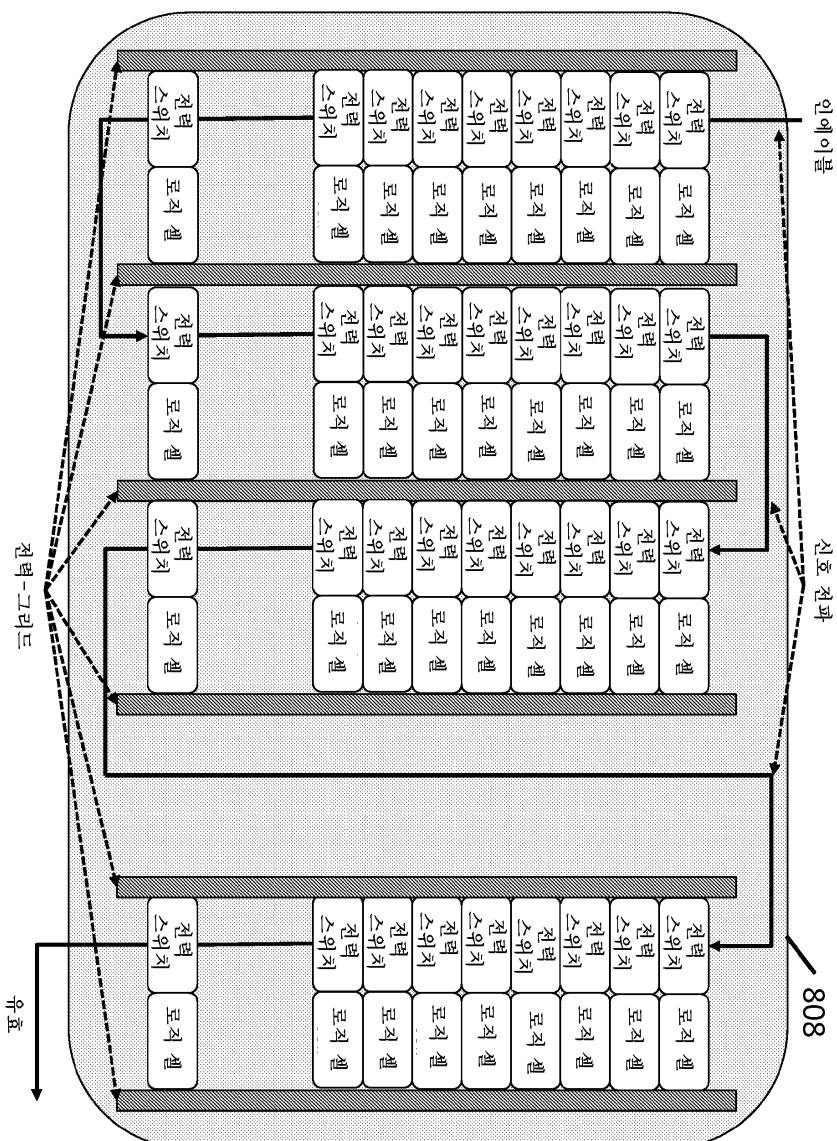
도면7



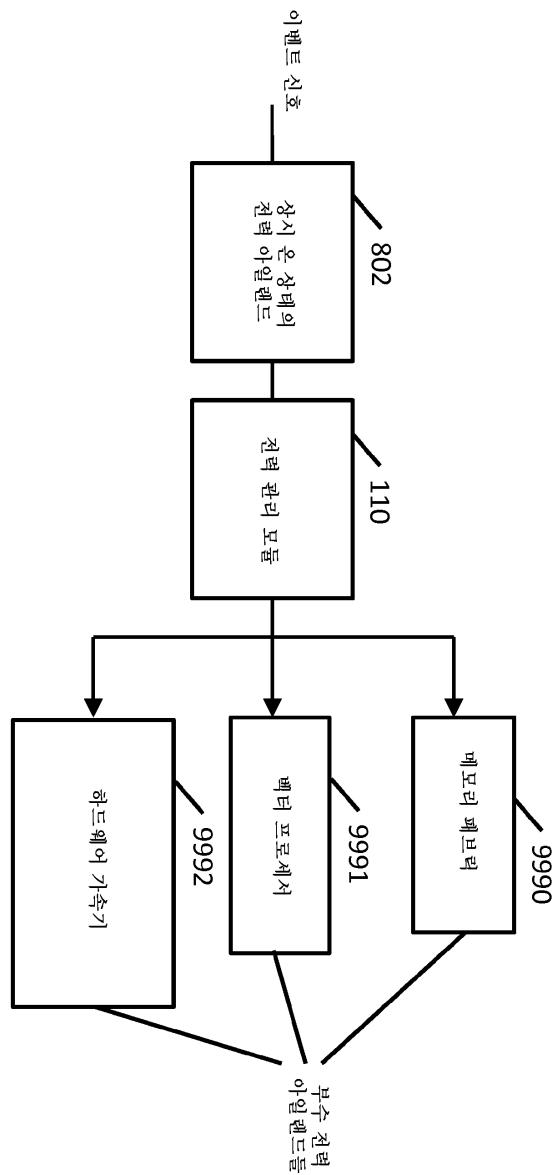
도면8



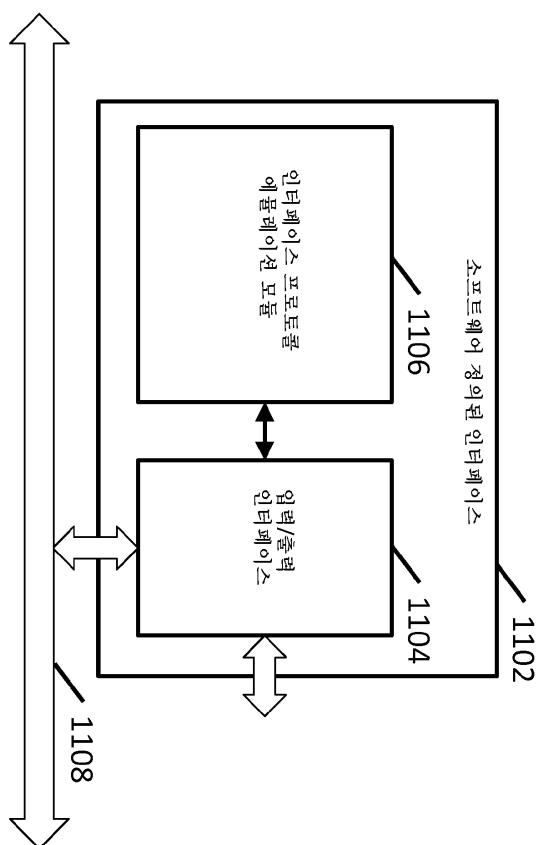
도면9



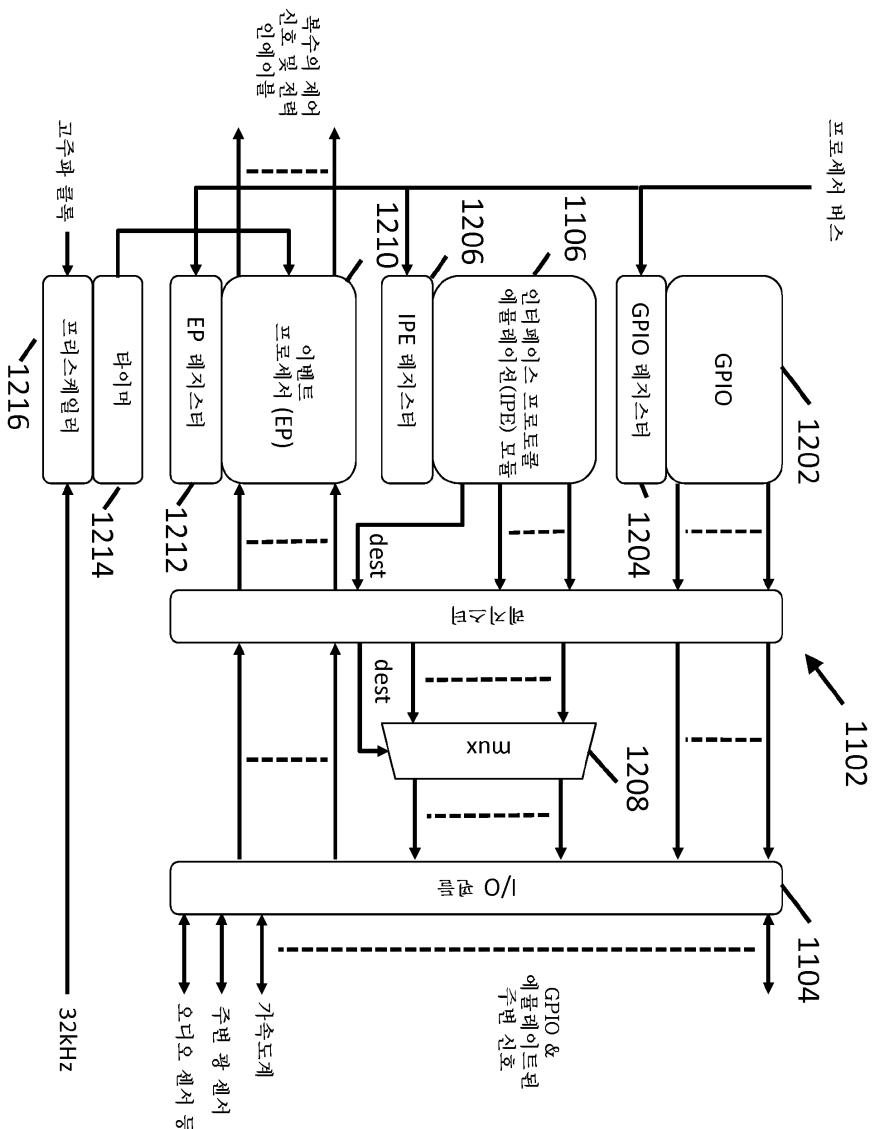
도면10



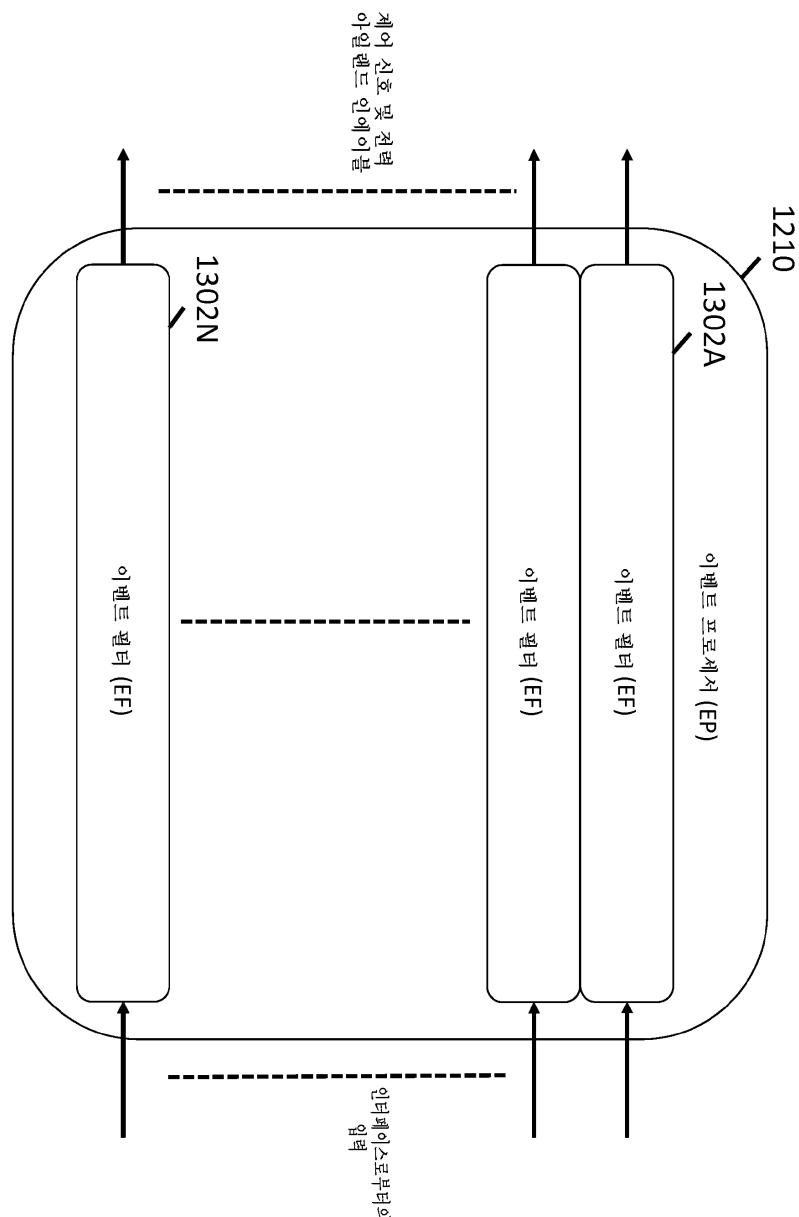
도면11



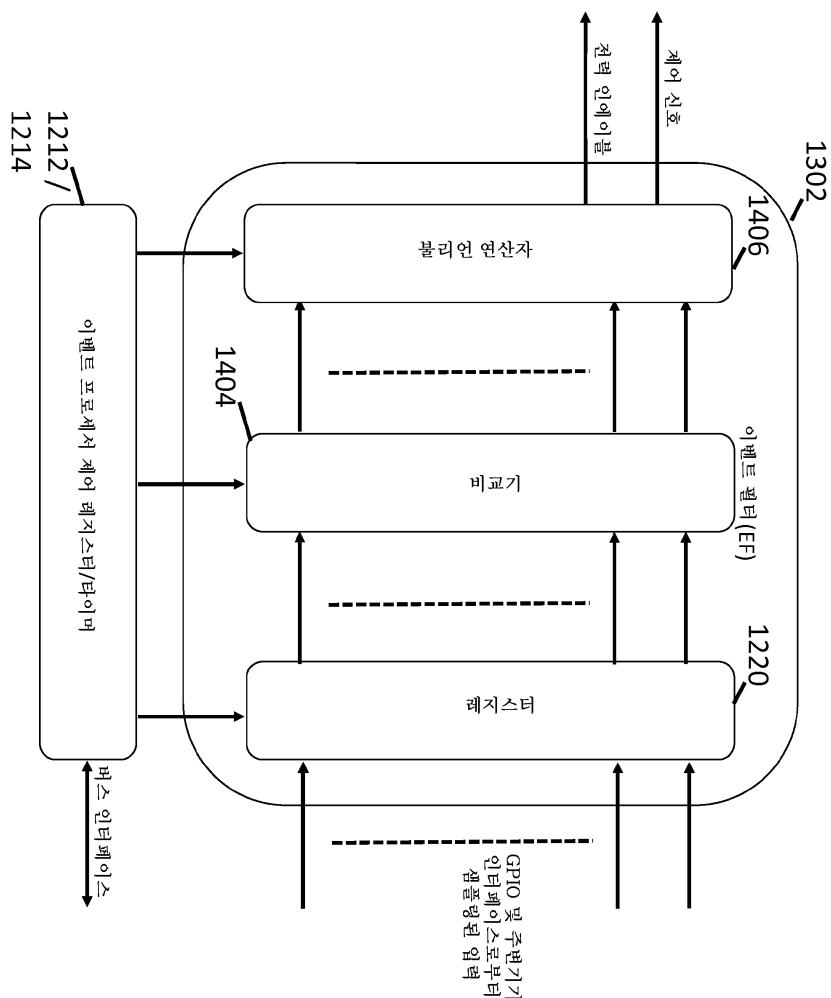
도면12



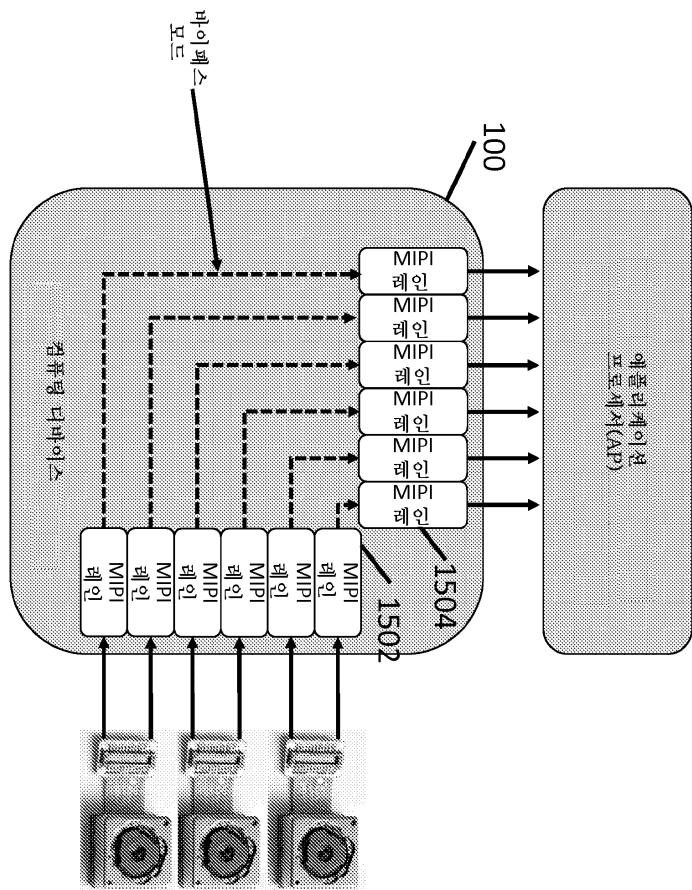
도면13



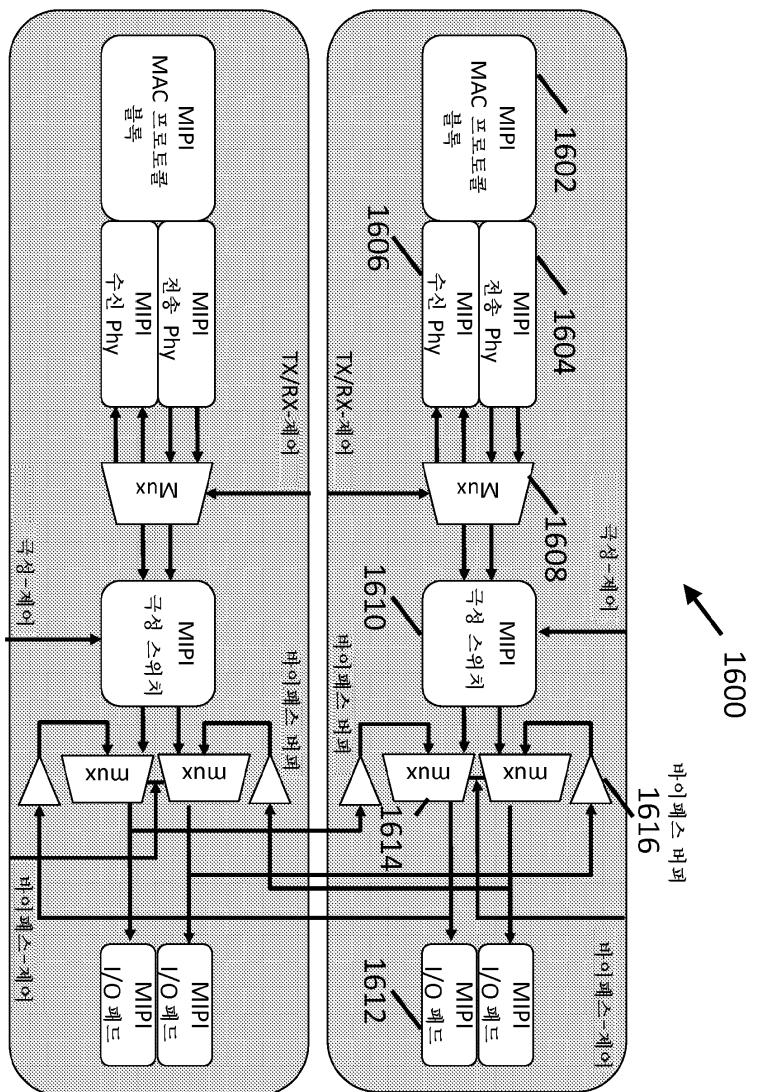
도면14



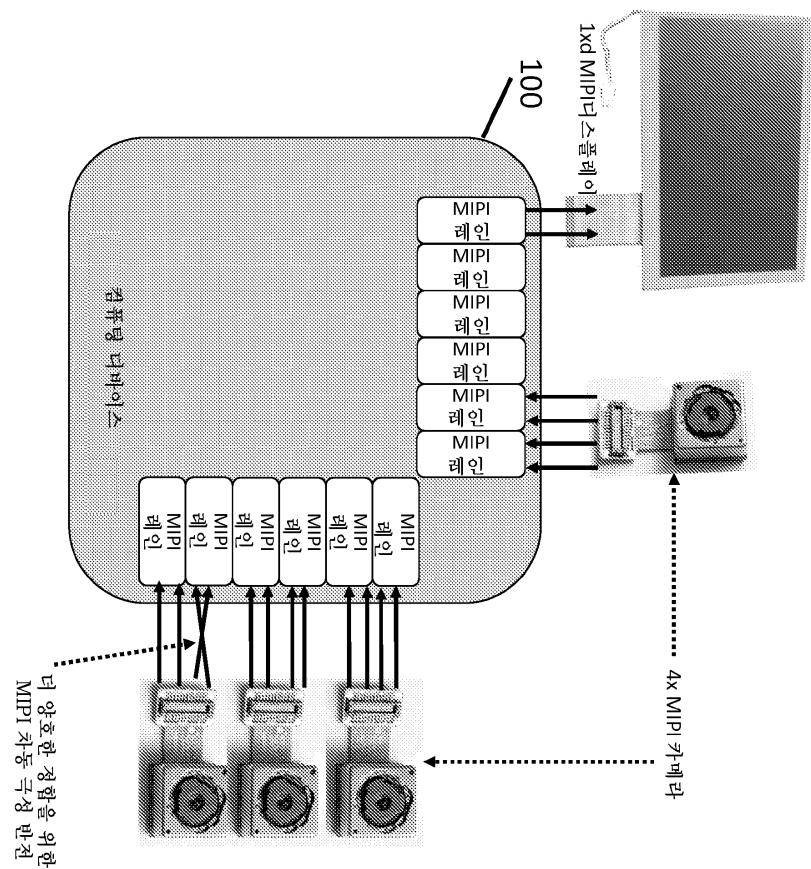
도면15



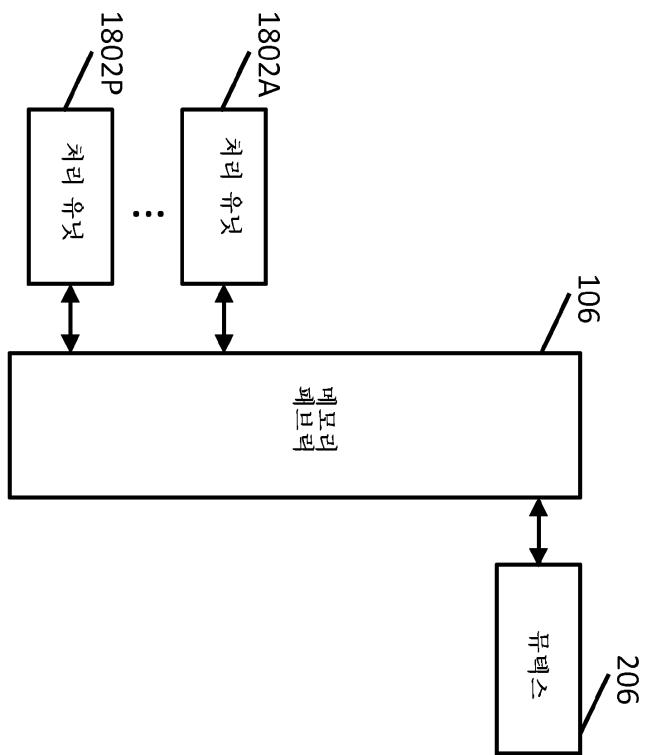
도면16



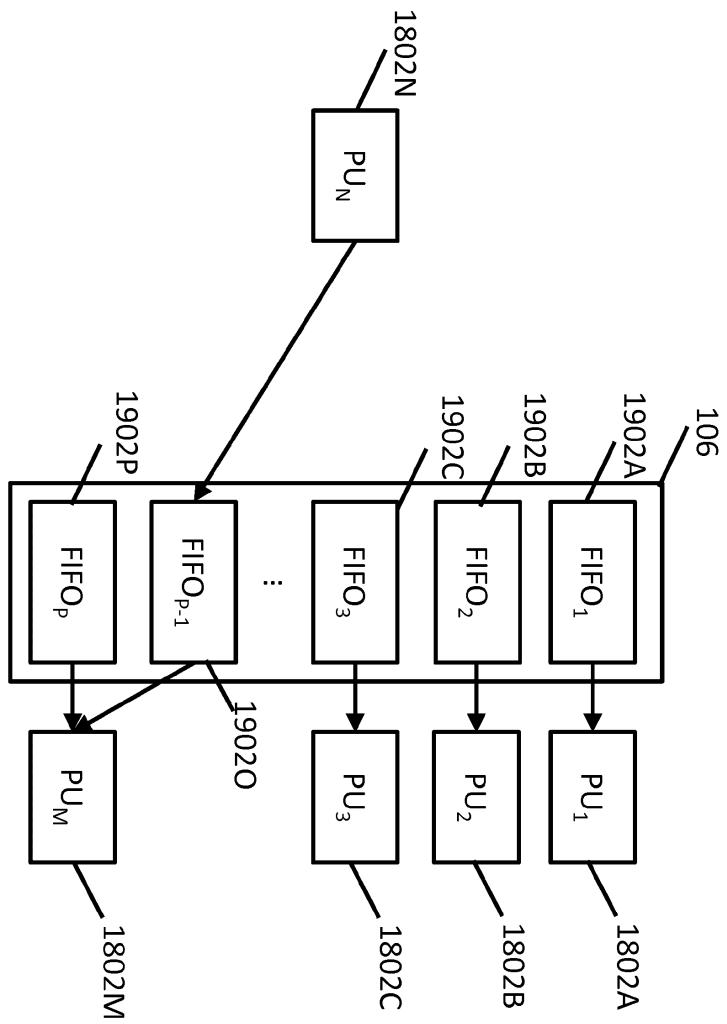
도면17



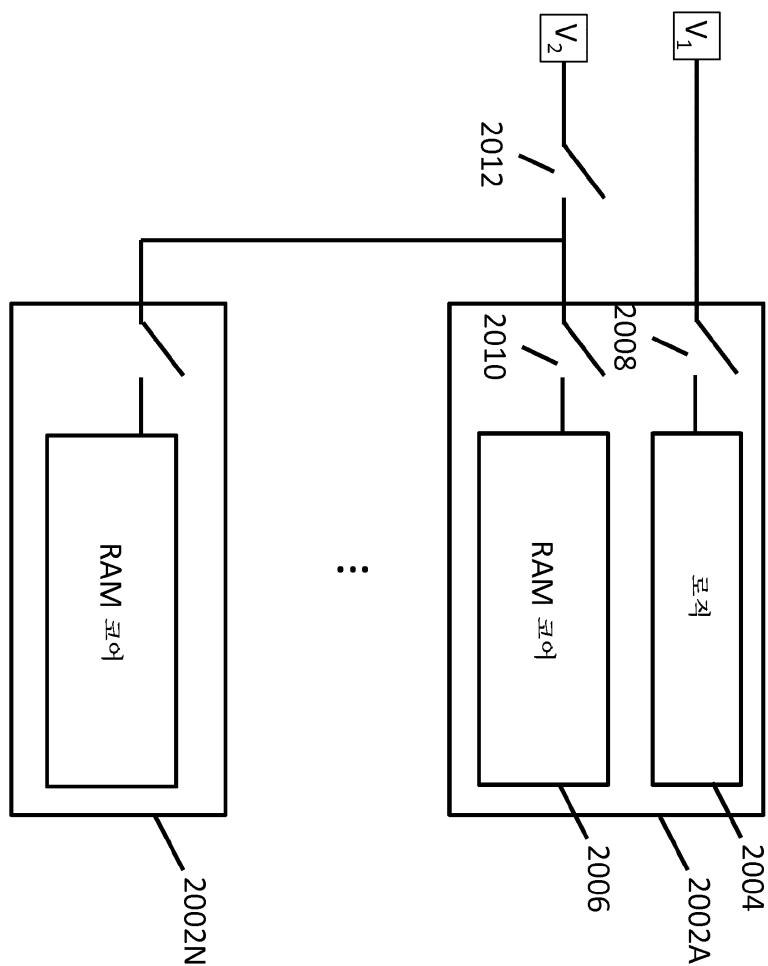
도면18



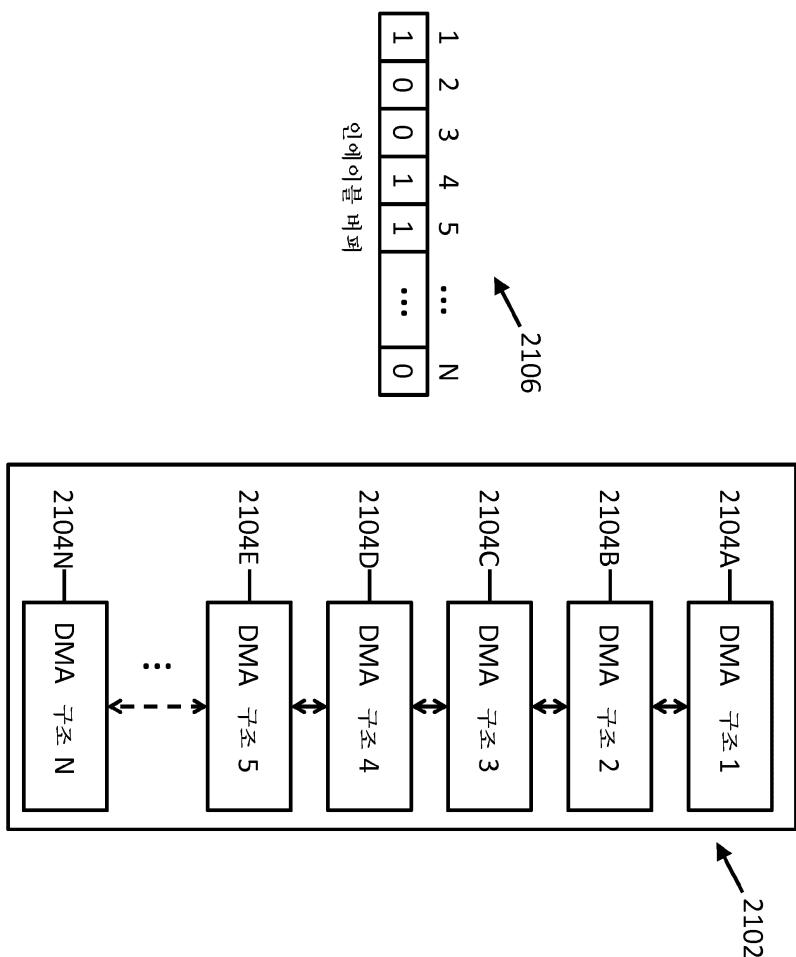
도면19



도면20



도면21



도면22

