



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년11월22일
(11) 등록번호 10-2329269
(24) 등록일자 2021년11월16일

- (51) 국제특허분류(Int. Cl.)
G06F 12/08 (2016.01) G06F 12/10 (2016.01)
G06F 3/06 (2006.01)
- (52) CPC특허분류
G06F 12/0811 (2013.01)
G06F 12/0804 (2013.01)
- (21) 출원번호 10-2016-7013471
- (22) 출원일자(국제) 2014년10월21일
심사청구일자 2019년06월12일
- (85) 번역문제출일자 2016년05월20일
- (65) 공개번호 10-2016-0074648
- (43) 공개일자 2016년06월28일
- (86) 국제출원번호 PCT/US2014/061603
- (87) 국제공개번호 WO 2015/061337
국제공개일자 2015년04월30일
- (30) 우선권주장
61/893,662 2013년10월21일 미국(US)
(뒷면에 계속)
- (56) 선행기술조사문헌
US07428617 B2*
W02008055272 A2*
US20080294867 A1*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
에프엘씨 글로벌 리미티드
버뮤다 해밀턴 에이치엠11 처치 스트리트 2 클래
런던 하우스
- (72) 발명자
수타르드자 세하트
미국 캘리포니아 94022 로스 알토스 엘레나 로드
27330
- (74) 대리인
박장원

전체 청구항 수 : 총 43 항

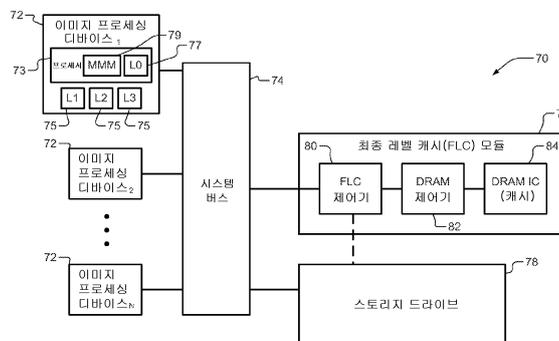
심사관 : 안지현

(54) 발명의 명칭 최종 레벨 캐시 시스템 및 이에 대응하는 방법

(57) 요약

프로세서 및 최종 레벨 캐시 모듈을 포함하는 데이터 액세스 시스템이 제공된다. 프로세서는 제 1 물리적 어드레스에 액세스하기 위한 요청을 발생시키도록 구성된다. 최종 레벨 캐시 모듈은, 동적 랜덤 액세스 메모리(DRAM), 최종 레벨 캐시 제어기, 및 DRAM 제어기를 포함한다. 최종 레벨 캐시 제어기는, (i) 프로세서로부터의 요청을 수신하도록 구성되며, 그리고 (ii) 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하도록 구성된다. DRAM 제어기는, (i) 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하도록 구성되며, 그리고 (ii) 제 2 물리적 어드레스에 근거하여 DRAM에 액세스하도록 구성된다.

대표도 - 도2



(52) CPC특허분류

G06F 12/0864 (2013.01)
G06F 12/0868 (2013.01)
G06F 12/0897 (2013.01)
G06F 12/10 (2013.01)
G06F 3/0638 (2013.01)
G06F 3/0655 (2013.01)
G06F 3/0656 (2013.01)
G06F 3/0679 (2013.01)
G06F 3/0683 (2013.01)

(30) 우선권주장

61/893,675	2013년10월21일	미국(US)
61/893,683	2013년10월21일	미국(US)
61/895,049	2013년10월24일	미국(US)
14/519,826	2014년10월21일	미국(US)

명세서

청구범위

청구항 1

데이터 액세스 시스템(data access system)으로서, 상기 데이터 액세스 시스템은,

제 1 물리적 어드레스(physical address)에 액세스(access)하기 위한 요청을 발생시키도록 되어 있는 프로세서(processor)와; 그리고

최종 레벨 캐시 모듈(final level cache module)을 포함하여 구성되며,

상기 최종 레벨 캐시 모듈은,

동적 랜덤 액세스 메모리(dynamic random access memory)와,

(i) 상기 프로세서로부터의 상기 요청을 수신하도록 되어 있으며, 그리고 (ii) 상기 제 1 물리적 어드레스를 제 1 가상 어드레스(virtual address)로 변환하도록 되어 있는 최종 레벨 캐시 제어기(final level cache controller)와, 그리고

(i) 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하도록 되어 있으며, 그리고 (ii) 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하도록 되어 있는 동적 랜덤 액세스 메모리 제어기(dynamic random access memory controller)를 포함하고,

상기 데이터 액세스 시스템은 스토리지 드라이브(storage drive)를 더 포함하고,

상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있으며,

상기 프로세서는 데이터의 전송을 위해 상기 요청을 발생시키도록 되어 있고,

상기 요청은 상기 제 1 물리적 어드레스를 표시하며,

상기 최종 레벨 캐시 제어기는,

상기 요청에 근거하여 상기 제 1 물리적 어드레스를 상기 제 1 가상 어드레스로 변환하도록 되어 있으며, 그리고

상기 최종 레벨 캐시 제어기는,

상기 제 2 물리적 어드레스에 근거하여, (i) 상기 프로세서와 상기 동적 랜덤 액세스 메모리 간의 상기 데이터의 전송을 수행할지, 혹은 (ii) 상기 프로세서와 상기 스토리지 드라이브 간의 상기 데이터의 전송을 수행할지를 결정하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 2

제1항에 있어서,

상기 최종 레벨 캐시 제어기는, 가상 어드레스로의 변환을 위해 매칭되는 것을 찾기 위해서 상기 제 1 물리적 어드레스를 상기 최종 레벨 캐시 제어기의 디렉토리(directory) 내에 저장된 모든 가상 어드레스들과 비교함으로써 상기 제 1 물리적 어드레스를 상기 제 1 가상 어드레스로 변환하는 풀 세트 연관 어드레스 변환(full set associative address translation)을 수행하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 3

데이터 액세스 시스템으로서, 상기 데이터 액세스 시스템은,

제 1 물리적 어드레스에 액세스하기 위한 요청을 발생시키도록 되어 있는 프로세서와; 그리고

최종 레벨 캐시 모듈을 포함하여 구성되며,

상기 최종 레벨 캐시 모듈은,

동적 랜덤 액세스 메모리와,

(i) 상기 프로세서로부터의 상기 요청을 수신하도록 되어 있으며, 그리고 (ii) 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하도록 되어 있는 최종 레벨 캐시 제어기와, 그리고

(i) 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하도록 되어 있으며, 그리고 (ii) 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하도록 되어 있는 동적 랜덤 액세스 메모리 제어기를 포함하고,

상기 데이터 액세스 시스템은 스토리지 드라이브를 더 포함하고,

상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있고,

상기 제 2 물리적 어드레스는 상기 동적 랜덤 액세스 메모리에서의 어드레스 혹은 상기 스토리지 드라이브에서의 어드레스이고,

상기 최종 레벨 캐시 제어기는 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 혹은 상기 스토리지 드라이브에 액세스할지를 결정하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 4

제3항에 있어서,

상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리보다 더 큰 스토리지 용량(storage capacity)을 갖는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 5

제3항에 있어서,

상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리보다 임의 차수(order)의 크기만큼 더 큰 스토리지 용량을 갖는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 6

제3항에 있어서,

상기 동적 랜덤 액세스 메모리로의 데이터의 전송율(transfer rate) 혹은 상기 동적 랜덤 액세스 메모리로부터의 데이터의 전송율은 상기 스토리지 드라이브로의 데이터의 전송율 혹은 상기 스토리지 드라이브로부터의 데이터의 전송율보다 더 빠른 것을 특징으로 하는 데이터 액세스 시스템.

청구항 7

제3항에 있어서,

상기 스토리지 드라이브는 솔리드 스테이트 메모리(solid-state memory) 혹은 회전 스토리지 매체(rotating storage medium)를 포함하는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 8

제3항에 있어서,

상기 스토리지 드라이브는 동적 랜덤 액세스 메모리를 포함하고,

상기 스토리지 드라이브의 동적 랜덤 액세스 메모리는 상기 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리보다 더 느린 데이터 전송율을 갖는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 9

제1항에 있어서,

스토리지 드라이브를 더 포함하고,

상기 동적 랜덤 액세스 메모리는 캐시로서 구현되고, 그리고 상기 스토리지 드라이브로부터 분리되어 있으며, 상기 최종 레벨 캐시 제어기는, (i) 상기 제 1 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 여부를 결정하도록 되어 있으며, 그리고 (ii) 상기 제 1 물리적 어드레스에 대해 캐시 미스(cache miss)가 일어났는지를 결정하도록 되어 있고, 그리고 (iii) 상기 캐시 미스에 근거하여, 상기 캐시 미스가 일어났다는 신호를 상기 프로세서에 보내거나 상기 스토리지 드라이브에 액세스하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 10

제1항에 있어서,

스토리지 드라이브를 더 포함하고,

상기 동적 랜덤 액세스 메모리는 캐시로서 구현되고, 그리고 상기 스토리지 드라이브로부터 분리되어 있으며,

상기 동적 랜덤 액세스 메모리 제어기는, (i) 상기 제 1 가상 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 여부를 결정하도록 되어 있으며, 그리고 (ii) 상기 제 1 가상 어드레스에 대해 캐시 미스가 일어났는지를 결정하도록 되어 있고, 그리고 (iii) 상기 캐시 미스에 근거하여, 상기 캐시 미스가 일어났다는 신호를 상기 최종 레벨 캐시 제어기에 보내도록 되어 있으며,

상기 최종 레벨 캐시 제어기는 상기 캐시 미스가 일어났음을 표시하는 상기 신호에 응답하여 상기 스토리지 드라이브에 액세스하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 11

제1항에 있어서,

상기 동적 랜덤 액세스 메모리는 캐시로서 구현되고,

상기 동적 랜덤 액세스 메모리 제어기는, (i) 상기 제 1 가상 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 여부를 결정하도록 되어 있으며, 그리고 (ii) 상기 제 1 가상 어드레스에 대해 캐시 미스가 일어났는지를 결정하도록 되어 있고, 그리고 (iii) 상기 캐시 미스에 근거하여, 상기 캐시 미스가 일어났음을 상기 최종 레벨 캐시에 표시하기 위한 제 1 신호를 발생시키도록 되어 있으며,

상기 최종 레벨 캐시 제어기는, 상기 제 1 신호에 응답하여, 상기 캐시 미스가 일어났음을 상기 프로세서에 표시하기 위한 제 2 신호를 발생시키도록 되어 있고, 그리고

상기 프로세서는 상기 제 2 신호에 응답하여 상기 스토리지 드라이브에 액세스하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 12

삭제

청구항 13

제1항에 있어서,

상기 프로세서와 상기 스토리지 드라이브 간에 상기 데이터가 전송되기 전에 상기 데이터를 암호화하도록 되어 있는 암호화 디바이스를 더 포함하는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 14

제1항에 있어서,

상기 프로세서는, (i) 제 2 가상 어드레스를 발생시키도록 되어 있으며, 그리고 (ii) 상기 제 2 가상 어드레스를 상기 제 1 물리적 어드레스로 변환하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 15

제1항에 있어서,

상기 동적 랜덤 액세스 메모리는 제 1 부분 및 제 2 부분을 포함하고,

상기 제 1 부분은 최종 레벨 캐시로서 구현되고,

상기 제 2 부분은 캐시로서 구현되지 않으며, 상기 제 2 부분의 계층적 레벨(hierarchical level)은 상기 제 1 부분의 계층적 레벨보다 더 낮은 것을 특징으로 하는 데이터 액세스 시스템.

청구항 16

제15항에 있어서,

상기 동적 랜덤 액세스 메모리 제어기는, 상기 계층적 레벨들에 근거하여 상기 제 1 부분 및 상기 제 2 부분에 액세스하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 17

제1항에 있어서,

상기 동적 랜덤 액세스 메모리는 제 1 부분 및 제 2 부분을 포함하고,

상기 제 1 부분은 가상 메모리로서 사용되고,

상기 제 2 부분은 가상 메모리로서 사용되지 않으며, 상기 제 2 부분의 계층적 레벨은 상기 제 1 부분의 계층적 레벨보다 더 낮은 것을 특징으로 하는 데이터 액세스 시스템.

청구항 18

데이터 액세스 시스템으로서, 상기 데이터 액세스 시스템은,

제 1 물리적 어드레스에 액세스하기 위한 요청을 발생시키도록 되어 있는 프로세서와; 그리고

최종 레벨 캐시 모듈을 포함하여 구성되며,

상기 최종 레벨 캐시 모듈은,

동적 랜덤 액세스 메모리와,

(i) 상기 프로세서로부터의 상기 요청을 수신하도록 되어 있으며, 그리고 (ii) 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하도록 되어 있는 최종 레벨 캐시 제어기와, 그리고

(i) 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하도록 되어 있으며, 그리고 (ii) 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하도록 되어 있는 동적 랜덤 액세스 메모리 제어기를 포함하고,

상기 데이터 액세스 시스템은 스토리지 드라이브를 더 포함하고,

상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있으며,

상기 최종 레벨 캐시 제어기는,

제 2 데이터의 전송을 위한 상기 프로세서로부터의 복수의 요청들을 수신하도록 되어 있고, 여기서 상기 복수의 요청들은 제1의 복수의 물리적 어드레스들을 표시하고,

상기 제1의 복수의 물리적 어드레스들을 복수의 가상 어드레스들로서 변환하도록 되어 있고,

상기 복수의 가상 어드레스들을 제2의 복수의 물리적 어드레스들로 변환하도록 되어 있고, 여기서 상기 제2의 복수의 물리적 어드레스들 중 일부는 상기 동적 랜덤 액세스 메모리 내에 있고, 상기 제2의 복수의 물리적 어드레스들 중 다른 것들은 상기 스토리지 드라이브 내에 있으며,

상기 제2의 복수의 물리적 어드레스들에 근거하여, (i) 상기 프로세서와 상기 동적 랜덤 액세스 메모리 간의 상기 제 2 데이터의 전송을 수행할지, 혹은 (ii) 상기 프로세서와 상기 스토리지 드라이브 간의 상기 제 2 데이터의 전송을 수행할지를 결정하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 19

제1항에 있어서,

상기 최종 레벨 캐시 모듈은 단지 하나의 집적 회로에 의해서 구현되는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 20

데이터 액세스 시스템으로서, 상기 데이터 액세스 시스템은,

제 1 물리적 어드레스에 액세스하기 위한 요청을 발생시키도록 되어 있는 프로세서와; 그리고

최종 레벨 캐시 모듈을 포함하여 구성되며,

상기 최종 레벨 캐시 모듈은,

동적 랜덤 액세스 메모리와,

(i) 상기 프로세서로부터의 상기 요청을 수신하도록 되어 있으며, 그리고 (ii) 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하도록 되어 있는 최종 레벨 캐시 제어기와, 그리고

(i) 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하도록 되어 있으며, 그리고 (ii) 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하도록 되어 있는 동적 랜덤 액세스 메모리 제어기를 포함하고,

상기 데이터 액세스 시스템은 스토리지 드라이브를 더 포함하고,

상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있으며,

상기 동적 랜덤 액세스 메모리는 제 1 엔트리(entry)들을 포함하고,

상기 스토리지 드라이브는 제 2 엔트리들을 포함하고, 여기서 상기 제 2 엔트리들은 상기 제 1 엔트리들에 맵핑(mapping)되며,

상기 최종 레벨 캐시 제어기는, 상기 제 1 엔트리들 중 하나의 엔트리에서의 잔존 스토리지 용량에 근거하여, 상기 스토리지 드라이브의 상기 제 2 엔트리들 중 하나의 엔트리와 상기 프로세서 간에 데이터를 전송하도록 되어 있으며,

상기 제 2 엔트리들 중 상기 하나의 엔트리는 상기 제 1 엔트리들 중 상기 하나의 엔트리에 맵핑되는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 21

제20항에 있어서,

상기 동적 랜덤 액세스 메모리 내의 각각의 엔트리는 상기 스토리지 드라이브 내의 하나 이상의 엔트리들에 맵핑되는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 22

제20항에 있어서,

상기 동적 랜덤 액세스 메모리 내의 하나 이상의 엔트리들은 상기 스토리지 드라이브 내의 하나 이상의 엔트리들에 맵핑되는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 23

제1항에 있어서,

상기 최종 레벨 캐시 제어기는, 상기 동적 랜덤 액세스 메모리 내에 데이터를 저장하도록 되어 있으며, 여기서 상기 최종 레벨 캐시 제어기는, (i) 상기 데이터의 이용 빈도, (ii) 상기 데이터의 우선순위 레벨(priority level), 혹은 (iii) 상기 데이터가 잠금 상태(locked state)에 있는지 여부 중 적어도 하나에 근거하여 상기 동적 랜덤 액세스 메모리 내에 상기 데이터를 저장하도록 되어 있는 것을 특징으로 하는 데이터 액세스 시스템.

청구항 24

최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법으로서, 상기 방법은,
 제 1 물리적 어드레스에 액세스하기 위한 요청을 프로세서를 통해 발생시키는 단계와;
 최종 레벨 캐시 제어기에서 상기 프로세서로부터의 상기 요청을 수신하는 단계와;
 상기 최종 레벨 캐시 제어기를 통해 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하는 단계와;
 동적 랜덤 액세스 메모리 제어기를 통해 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하는 단계와;
 그리고
 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하는 단계를 포함하고,
 상기 방법은,
 상기 최종 레벨 캐시 제어기를 통해 상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 혹은 스토리지 드라이브에 액세스할지를 결정하는 단계를 더 포함하고, 여기서,
 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있고,
 상기 제 2 물리적 어드레스는 상기 동적 랜덤 액세스 메모리에서의 어드레스 혹은 상기 스토리지 드라이브에서의 어드레스인 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 25

제24항에 있어서,
 상기 제 1 물리적 어드레스를 상기 제 1 가상 어드레스로 변환하는 단계는, 가상 어드레스로의 변환을 위해 매칭되는 것을 찾기 위해서 상기 제 1 물리적 어드레스를 상기 최종 레벨 캐시 제어기의 디렉토리 내에 저장된 모든 가상 어드레스들과 비교함으로써 상기 제 1 물리적 어드레스를 상기 제 1 가상 어드레스로 변환하는 풀 세트 연관 어드레스 변환을 수행하는 것을 포함하는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 26

삭제

청구항 27

제24항에 있어서,
 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리보다 더 큰 스토리지 용량을 갖는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 28

제24항에 있어서,
 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리보다 임의 차수의 크기만큼 더 큰 스토리지 용량을 갖는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 29

제24항에 있어서,
 상기 동적 랜덤 액세스 메모리의 데이터의 전송율 혹은 상기 동적 랜덤 액세스 메모리로부터의 데이터의 전송율은 상기 스토리지 드라이브로의 데이터의 전송율 혹은 상기 스토리지 드라이브로부터의 데이터의 전송율보다 더 빠른 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 30

제24항에 있어서,

상기 스토리지 드라이브는 솔리드 스테이트 메모리 혹은 회전 스토리지 매체를 포함하는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 31

제24항에 있어서,

상기 스토리지 드라이브는 동적 랜덤 액세스 메모리를 포함하고,

상기 스토리지 드라이브의 동적 랜덤 액세스 메모리는 상기 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리보다 더 느린 데이터 전송율을 갖는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 32

제24항에 있어서,

상기 제 1 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 여부를 결정하는 단계와, 여기서 상기 동적 랜덤 액세스 메모리는 캐시로서 구현되고;

상기 제 1 물리적 어드레스에 대해 캐시 미스가 일어났는지를 결정하는 단계와; 그리고

상기 캐시 미스에 근거하여, 상기 캐시 미스가 일어났다는 신호를 상기 프로세서에 보내거나 스토리지 드라이브에 액세스하는 단계를 더 포함하며,

여기서, 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 33

제24항에 있어서,

상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 여부를 결정하는 단계와, 여기서 상기 동적 랜덤 액세스 메모리는 캐시로서 구현되고;

상기 제 1 가상 어드레스에 대해 캐시 미스가 일어났는지를 결정하는 단계와;

상기 캐시 미스에 근거하여, 상기 캐시 미스가 일어났다는 신호를 상기 최종 레벨 캐시 제어기에 보내는 단계와; 그리고

상기 캐시 미스가 일어났음을 표시하는 상기 신호에 응답하여 스토리지 드라이브에 액세스하는 단계를 더 포함하고,

여기서, 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 34

제24항에 있어서,

상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스할지 여부를 결정하는 단계와, 여기서 상기 동적 랜덤 액세스 메모리는 캐시로서 구현되고;

상기 동적 랜덤 액세스 메모리 제어기를 통해 상기 제 1 가상 어드레스에 대해 캐시 미스가 일어났는지를 결정하는 단계와;

상기 캐시 미스에 근거하여, 상기 캐시 미스가 일어났음을 상기 최종 레벨 캐시에 표시하기 위한 제 1 신호를 발생시키는 단계와;

상기 제 1 신호에 응답하여, 상기 캐시 미스가 일어났음을 상기 프로세서에 표시하기 위한 제 2 신호를 상기 최종 레벨 캐시 제어기에서 발생시키는 단계와; 그리고

상기 제 2 신호에 응답하여 상기 스토리지 드라이브에 액세스하는 단계를 더 포함하며,

여기서, 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 35

최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법으로서, 상기 방법은,

제 1 물리적 어드레스에 액세스하기 위한 요청을 프로세서를 통해 발생시키는 단계와;

최종 레벨 캐시 제어기에서 상기 프로세서로부터의 상기 요청을 수신하는 단계와;

상기 최종 레벨 캐시 제어기를 통해 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하는 단계와;

동적 랜덤 액세스 메모리 제어기를 통해 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하는 단계와;
그리고

상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하는 단계를 포함하고,

상기 방법은,

데이터의 전송을 위해 상기 요청을 발생시키는 단계와, 여기서 상기 요청은 상기 제 1 물리적 어드레스를 표시하며;

상기 요청에 근거하여 상기 제 1 물리적 어드레스를 상기 제 1 가상 어드레스로 변환하는 단계와;

상기 제 2 물리적 어드레스에 근거하여, (i) 상기 프로세서와 상기 동적 랜덤 액세스 메모리 간의 상기 데이터의 전송을 수행할지, 혹은 (ii) 상기 프로세서와 스토리지 드라이브 간의 상기 데이터의 전송을 수행할지를 결정하는 단계를 더 포함하며,

여기서, 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 36

제35항에 있어서,

상기 프로세서와 상기 스토리지 드라이브 간에 상기 데이터가 전송되기 전에 상기 데이터를 암호화하는 단계를 더 포함하는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 37

제24항에 있어서,

상기 프로세서에서 제 2 가상 어드레스를 발생시키는 단계와; 그리고

상기 제 2 가상 어드레스를 상기 제 1 물리적 어드레스로 변환하는 단계를 더 포함하는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 38

제24항에 있어서,

상기 동적 랜덤 액세스 메모리의 일부분들의 계층적 레벨들에 근거하여 상기 동적 랜덤 액세스 메모리의 상기 일부분들에 액세스하는 단계를 더 포함하고,

여기서, 상기 동적 랜덤 액세스 메모리는 제 1 부분 및 제 2 부분을 포함하고,

상기 제 1 부분은 최종 레벨 캐시로서 구현되고,

상기 제 2 부분은 캐시로서 구현되지 않으며, 상기 제 2 부분의 계층적 레벨은 상기 제 1 부분의 계층적 레벨보다 더 낮은 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 39

제24항에 있어서,

상기 동적 랜덤 액세스 메모리의 일부분들의 계층적 레벨들에 근거하여 상기 동적 랜덤 액세스 메모리의 상기 일부분들에 액세스하는 단계를 더 포함하고,

여기서, 상기 동적 랜덤 액세스 메모리는 제 1 부분 및 제 2 부분을 포함하고,

상기 제 1 부분은 가상 메모리로서 사용되고,

상기 제 2 부분은 가상 메모리로서 사용되지 않으며, 상기 제 2 부분의 계층적 레벨은 상기 제 1 부분의 계층적 레벨보다 더 낮은 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 40

최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법으로서, 상기 방법은,

제 1 물리적 어드레스에 액세스하기 위한 요청을 프로세서를 통해 발생시키는 단계와;

최종 레벨 캐시 제어기에서 상기 프로세서로부터의 상기 요청을 수신하는 단계와;

상기 최종 레벨 캐시 제어기를 통해 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하는 단계와;

동적 랜덤 액세스 메모리 제어기를 통해 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하는 단계와;
그리고

상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하는 단계를 포함하고,

상기 방법은,

제 2 데이터의 전송을 위한 상기 프로세서로부터의 복수의 요청들을 수신하는 단계와, 여기서 상기 복수의 요청들은 제1의 복수의 물리적 어드레스들을 표시하고;

상기 제1의 복수의 물리적 어드레스들을 복수의 가상 어드레스들로서 변환하는 단계와;

상기 복수의 가상 어드레스들을 제2의 복수의 물리적 어드레스들로 변환하는 단계와, 여기서 상기 제2의 복수의 물리적 어드레스들 중 일부는 상기 동적 랜덤 액세스 메모리 내에 있고, 상기 제2의 복수의 물리적 어드레스들 중 다른 것들은 스토리지 드라이브 내에 있으며, 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있고; 그리고

상기 제2의 복수의 물리적 어드레스들에 근거하여, (i) 상기 프로세서와 상기 동적 랜덤 액세스 메모리 간의 상기 제 2 데이터의 전송을 수행할지, 혹은 (ii) 상기 프로세서와 상기 스토리지 드라이브 간의 상기 제 2 데이터의 전송을 수행할지를 결정하는 단계를 더 포함하는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 41

제24항에 있어서,

상기 최종 레벨 캐시 모듈은 단지 하나의 집적 회로에 의해서 구현되는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 42

최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법으로서, 상기 방법은,

제 1 물리적 어드레스에 액세스하기 위한 요청을 프로세서를 통해 발생시키는 단계와;

최종 레벨 캐시 제어기에서 상기 프로세서로부터의 상기 요청을 수신하는 단계와;

상기 최종 레벨 캐시 제어기를 통해 상기 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하는 단계와;

동적 랜덤 액세스 메모리 제어기를 통해 상기 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하는 단계와;
그리고

상기 제 2 물리적 어드레스에 근거하여 상기 동적 랜덤 액세스 메모리에 액세스하는 단계를 포함하고, 상기 방법은,

상기 동적 랜덤 액세스 메모리의 제 1 엔트리들 중 하나의 엔트리에서의 잔존 스토리지 용량에 근거하여, 스토리지 드라이브의 제 2 엔트리들 중 하나의 엔트리와 상기 프로세서와 간에 데이터를 전송하는 단계를 더 포함하고, 여기서 상기 스토리지 드라이브는 상기 동적 랜덤 액세스 메모리로부터 분리되어 있으며, 상기 제 2 엔트리들은 상기 제 1 엔트리들에 맵핑되고; 그리고

상기 제 2 엔트리들 중 상기 하나의 엔트리는 상기 제 1 엔트리들 중 상기 하나의 엔트리에 맵핑되는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 43

제42항에 있어서,

상기 동적 랜덤 액세스 메모리 내의 각각의 엔트리는 상기 스토리지 드라이브 내의 하나 이상의 엔트리들에 맵핑되는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 44

제42항에 있어서,

상기 동적 랜덤 액세스 메모리 내의 하나 이상의 엔트리들은 상기 스토리지 드라이브 내의 하나 이상의 엔트리들에 맵핑되는 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

청구항 45

제24항에 있어서,

상기 동적 랜덤 액세스 메모리 내에 데이터를 저장하는 단계를 더 포함하고, 여기서 상기 저장하는 단계는, (i) 상기 데이터의 이용 빈도, (ii) 상기 데이터의 우선순위 레벨, 혹은 (iii) 상기 데이터가 잠금 상태에 있는지 여부 중 적어도 하나에 근거하여 상기 동적 랜덤 액세스 메모리 내에 상기 데이터를 저장하는 것인 것을 특징으로 하는 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법.

발명의 설명

기술 분야

- [0001] 관련 출원들에 대한 상호 참조
- [0002] 본 출원은 미국 특허 출원 번호 제14/519,826호(출원일: 2014년 10월 21일)에 대해 우선권을 주장하며, 아울러 미국 가출원 번호 제61/893,662호(출원일: 2013년 10월 21일), 미국 가출원 번호 제61/893,675호(출원일: 2013년 10월 21일), 미국 가출원 번호 제61/893,683호(출원일: 2013년 10월 21일), 그리고 미국 가출원 번호 제61/895,049호(출원일: 2013년 10월 24일)의 우선권 혜택을 또한 주장한다. 앞서 언급된 출원들의 전체 개시내용들은 참조로 본 명세서에 통합된다.
- [0003] 본 개시내용은 집적 회로들에 관한 것으로, 보다 구체적으로는 시스템-온-칩(system-on-chip)과 관련된 캐시(cache)에 관한 것이다.

배경 기술

- [0004] 본 명세서에서 제공되는 배경기술은 본 개시내용의 배경 상황을 전반적으로 제시하기 위한 것이다. 현재 본 발명의 발명자로 명명된 사람들이 행한 작업은, 이러한 작업이 본 배경기술 부분에서 설명되는 한, 뿐만 아니라 (만약 본 배경기술 부분에서 설명되지 않은 경우 본 출원의 출원시 종래 기술로서의 자격이 없는) 여러 양상의 설명으로 제공되는 한, 본 개시내용에 대한 종래 기술로서 명백하게 인정되는 것이 아니며 암시적으로 인정되는 것도 아니다.
- [0005] 컴퓨터들, 모바일 폰들, 태블릿들 등과 같은 디바이스들은 전형적으로 시스템-온-칩(System-on-Chip, SoC)을 포함한다. 도 1은 SoC(12) 및 하나 이상의 동적 랜덤 액세스 메모리(Dynamic Random Access Memory, DRAM)들(1

4)을 포함하는 디바이스(10)의 예를 보여준다. DRAM들(14)은 SoC(12)에 연결되는(하지만 SoC(12)로부터 분리되어 있는) 하나 이상의 집적 회로들로서 구현될 수 있다. 디바이스(10)는 또한, SoC(12)의 포트(port)들(17)에 연결되는 하나 이상의 스토리지 드라이브(storage drive)들(16)을 포함할 수 있다. 스토리지 드라이브들(16)은 플래시 메모리(flash memory), 솔리드-스테이트 드라이브(solid-state drive)들, 하드 디스크 드라이브(hard disk drive)들, 및/또는 하이브리드 드라이브(hybrid drive)들을 포함할 수 있다. 하이브리드 드라이브는 솔리드-스테이트 메모리를 갖는 솔리드-스테이트 드라이브 및 회전 스토리지 매체들(rotating storage media)을 갖는 하드 디스크 드라이브를 포함한다.

[0006] SoC(12)는 하나 이상의 이미지 프로세싱 디바이스(image processing device)들(20), 시스템 버스(system bus)(22), 및 메모리 제어기(memory controller)(24)를 포함할 수 있다. 이미지 프로세싱 디바이스들(20) 각각은 예를 들어, 중앙 프로세서(혹은 중앙 프로세싱 유닛(Central Processing Unit, CPU))(28)를 갖는 제어 모듈(control module)(26); 그래픽 프로세서(혹은 그래픽 프로세싱 유닛(Graphics Processing Unit, GPU))(30); 비디오 리코더(video recorder)(32); 카메라 이미지 신호 프로세서(Image Signal Processor, ISP)(34); 이더넷 인터페이스(Ethernet interface)(예컨대, 기가비트(Gigabit)(Gb) 이더넷 인터페이스(36)); 직렬 인터페이스(serial interface)(예컨대, 범용 직렬 버스(Universal Serial Bus, USB) 인터페이스(38) 및 직렬 고급 기술 결합(Serial Advanced Technology Attachment, SATA) 인터페이스(40)); 그리고 주변 컴포넌트 상호연결 익스프레스(Peripheral Component Interconnect express, PCIe) 인터페이스(42)를 포함할 수 있다. 이미지 프로세싱 디바이스들(20)은 시스템 버스(22) 및 메모리 제어기(24)를 통해 DRAM들(14)에 액세스한다. DRAM들(14)은 메인 메모리(main memory)로서 사용된다. 예를 들어, 이미지 프로세싱 디바이스들(20) 중 하나는 메모리 제어기(24)에 물리적 어드레스(physical address)를 제공하는데, 이러한 것은 DRAM들(14) 중 하나의 DRAM 내의 대응하는 물리적 위치에 액세스할 때 일어난다. 이미지 프로세싱 디바이스들(20)은 또한 시스템 버스(22)를 통해 스토리지 드라이브들(16)에 액세스할 수 있다.

[0007] SoC(12) 및/또는 메모리 제어기(24)는 SoC(12)의 하나 이상의 액세스 포트들(44)을 통해 DRAM들(14)에 연결될 수 있다. DRAM들(14)은 사용자 데이터, 시스템 데이터, 및/또는 프로그램들을 저장한다. SoC(12)는 제 2 데이터를 발생시키기 위해 제 1 데이터를 사용하여 프로그램들을 실행할 수 있다. 제 1 데이터는 프로그램들의 실행 이전에 DRAM들(14) 내에 저장될 수 있다. SoC(12)는 프로그램들의 실행 동안에 그리고/또는 프로그램들의 실행 이후에 DRAM들(14) 내에 제 2 데이터를 저장할 수 있다. DRAM들(14)은 고-대역폭 인터페이스 및 비트-당-저-비용 메모리 스토리지 용량을 가지며, 다양한 범위의 애플리케이션들을 처리할 수 있다.

[0008] SoC(12)는 캐시 메모리(cache memory)를 포함하고, 캐시 메모리는, 레벨 0(Level zero, L0) 캐시, 레벨 1(Level one, L1) 캐시, 레벨 2(Level two, L2) 캐시, 혹은 레벨 3(Level three, L3) 캐시 중 하나 이상을 포함할 수 있다. L0 내지 L3 캐시들은 이미지 프로세싱 디바이스들(20) 중 대응하는 이미지 프로세싱 디바이스들에 가까이 근접하여 SoC(12) 상에 정렬된다. 제시된 예에서, 제어 모듈(26)은 중앙 프로세서(28) 및 L1 내지 L3 캐시들(50)을 포함한다. 중앙 프로세서(28)는 L0 캐시(52)를 포함한다. 중앙 프로세서(28)는 또한, 메모리 관리 유닛(Memory Management Unit, MMU)(54)을 포함하고, 메모리 관리 유닛(MMU)(54)은 캐시들(50, 52)에 대한 액세스를 제어할 수 있다.

[0009] 캐시의 레벨이 증가함에 따라, 캐시의 스토리지 용량 및 액세스 레이턴시(access latency)가 증가한다. 예를 들어, L1 캐시는 전형적으로 L2 캐시 및 L3 캐시보다 더 적은 스토리지 용량을 갖는다. 그러나, L1 캐시는 L2 캐시 및 L3 캐시보다 더 낮은 레이턴시를 갖는다.

[0010] SoC(12) 내의 캐시들은 전형적으로 정적 랜덤 액세스 메모리(Static Random Access Memory, SRAM)들로서 구현된다. 캐시들이 이미지 프로세싱 디바이스들(20)에 가까이 근접해 있기 때문에, 이들 캐시들은 이미지 프로세싱 디바이스들(20)과 동일한 클럭 주파수(clock frequency)들에서 동작할 수 있다. 따라서, 캐시들은 DRAM들(14)보다 더 짧은 레이턴시 기간(latency period)들을 보여준다.

[0011] SoC(12) 내의 캐시들의 개수 및 크기는 애플리케이션에 따라 달라진다. 예를 들어, 핸드셋(handset)(혹은 모바일 폰)은 L3 캐시를 포함하지 않을 수 있고, 개인용 컴퓨터보다 더 작은 크기의 L1 캐시 및 L2 캐시를 가질 수 있다. 유사하게, DRAM들(14) 각각의 개수 및 크기는 애플리케이션에 따라 달라진다. 예를 들어, 모바일 폰들은 현재 1 내지 4 기가바이트(gigabytes)(GB)의 DRAM을 가지고 있으며, 개인용 컴퓨터들은 현재 4 내지 16 GB의 DRAM을 가지고 있고, 그리고 서버들은 현재 32GB 내지 512GB의 DRAM을 가지고 있다. 일반적으로, DRAM의 양이 증가함에 따라 비용이 증가한다.

[0012] DRAM의 비용에 추가하여, 동일한 양의 스토리지 용량에 대해 DRAM의 패키지 크기를 감소시키는 것은 점점 더 어

려워지고 있다. 또한, 디바이스 내에 포함되는 DRAM들의 크기 및 개수가 증가함에 따라, DRAM들의 커패시턴스들이 증가하고, DRAM들과 관련된 전도성 요소들의 개수 및/또는 길이들이 증가하며, 그리고 DRAM들과 관련된 버퍼링(buffering)이 증가한다. 추가적으로, DRAM들의 커패시턴스들이 증가함에 따라, DRAM들의 동작 주파수들이 감소하고, DRAM들의 레이턴시 기간들이 증가한다.

[0013] 동작 동안, 프로그램들 및/또는 데이터는 필요에 따라 DRAM들(14)로부터 SoC(12) 내의 캐시들로 전송된다. 이러한 전송은, (i) 캐시들과 (ii) 대응하는 프로세서들 및/또는 이미지 프로세싱 디바이스들 간의 데이터 교환과 비교해 더 큰 레이턴시를 갖는다. 이러한 이유 때문에, DRAM들(14)에 대한 액세스들은 더 긴 레이턴시 기간들로 인해 전형적으로 피해지고 있다.

[0014] 부트업(boot up) 동안, 프로그램들이 스토리지 드라이브들(16)로부터 DRAM들(14)로 전송될 수 있다. 예를 들어, 중앙 프로세서(28)는 부트업 동안 프로그램들을 스토리지 드라이브(16)로부터 DRAM들(14)로 전송할 수 있다. 부트업 동안, 중앙 프로세서(28)는 DRAM들(14) 내에 저장된 데이터에 액세스하려는 시도를 행할 수 있다. 이러한 액세스 시도와 관련된 히트(hit)들의 퍼센티지는 초기에 거의 0%이거나 0%일 수 있다. 그러나, 부트업의 끝날 즈음에 히트들의 퍼센티지는 100%에 접근한다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0015] 데이터 액세스 시스템(data access system)이 제공되고, 여기서 데이터 액세스 시스템은 프로세서 및 최종 레벨 캐시 모듈(final level cache module)을 포함한다. 프로세서는 제 1 물리적 어드레스에 액세스하기 위한 요청을 발생시키도록 구성된다. 최종 레벨 캐시 모듈은, 동적 랜덤 액세스 메모리(DRAM), 최종 레벨 캐시 제어기(final level cache controller), 그리고 DRAM 제어기를 포함한다. 최종 레벨 캐시 제어기는, (i) 프로세서로부터의 요청을 수신하도록 구성되며, 그리고 (ii) 제 1 물리적 어드레스를 제 1 가상 어드레스(virtual address)로 변환하도록 구성된다. DRAM 제어기는, (i) 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하도록 구성되며, 그리고 (ii) 제 2 물리적 어드레스에 근거하여 DRAM에 액세스하도록 구성된다.

[0016] 최종 레벨 캐시 모듈의 동적 랜덤 액세스 메모리에 액세스하기 위한 방법이 제공된다. 본 방법은, 제 1 물리적 어드레스에 액세스하기 위한 요청을 프로세서를 통해 발생시키는 것과; 최종 레벨 캐시 제어기에서 프로세서로부터의 요청을 수신하는 것과; 최종 레벨 캐시 제어기를 통해 제 1 물리적 어드레스를 제 1 가상 어드레스로 변환하는 것과; 동적 랜덤 액세스 메모리 제어기를 통해 제 1 가상 어드레스를 제 2 물리적 어드레스로 변환하는 것과; 그리고 제 2 물리적 어드레스에 근거하여 동적 랜덤 액세스 메모리에 액세스하는 것을 포함한다.

[0017] 본 개시내용의 다른 이용가능한 분야는 아래의 상세한 설명, 특허청구범위, 및 도면으로부터 명백하게 될 것이다. 상세한 설명 및 특정 예들은 본 개시내용의 범위를 한정할 의도로 제공되는 것이 아니고 오로지 예시적 목적을 갖도록 의도된 것이다.

도면의 간단한 설명

- [0018] 도 1은 종래 기술에 따른 디바이스의 기능 블록도이다.
 - 도 2는 본 개시내용의 실시예에 따른 데이터 액세스 시스템의 기능 블록도이다.
 - 도 3은 도 2의 데이터 액세스 시스템의 DRAM 및 스토리지 드라이브의 엔트리들을 예시하는 기능 블록도이다.
 - 도 4는 도 2의 데이터 액세스 시스템을 동작시키는 방법을 예시한다.
- 도면에서는, 유사한 요소들 및/또는 동일한 요소들을 식별시키기 위해 참조 번호들이 재사용될 수 있다.

발명을 실시하기 위한 구체적인 내용

[0019] 다음의 예들 중 적어도 일부는 최종 레벨 캐시(Final Level Cache, FLC) 모듈들 및 스토리지 드라이브들을 포함한다. FLC 모듈들은 가상 스토리지로서 사용되고, 스토리지 드라이브들은 물리적 스토리지로서 사용된다. 물리적 스토리지로부터의 판독 혹은 물리적 스토리지로의 기입 이전에, 데이터는 먼저 가상 스토리지로부터 판독되

거나 가상 스토리지에 기입된다. FLC 모듈들 및 스토리지 드라이브들 내의 메모리는 본 명세서에서 콘텐츠 어드레싱가능 메모리(Content Addressable Memory, CAM)로서 지칭된다. FLC 모듈들의 제어 모듈들은 아래에서 설명되는 다양한 CAM 기법들을 사용하여 FLC 모듈들 및 스토리지 드라이브들 내의 메모리에 대한 액세스를 제어한다. CAM 기법들 및 다른 개시되는 특징들은 메모리 액세스율(memory access rates)을 최대화시킴과 아울러 전력 소비를 최소화시키면서 디바이스 내의 DRAM의 스토리지 능력 요구는 감소시킨다. 디바이스는 네트워크 디바이스 혹은 무선 네트워크 디바이스일 수 있다. 디바이스들의 예들은, 컴퓨터, 모바일 폰, 태블릿, 카메라 등을 포함하지만 이러한 것으로만 한정되는 것은 아니다. 아래의 예들에서의 DRAM은 일반적으로 메인 메모리로서 사용되지 않으며, 오히려 가상 메모리로서 사용되고 그리고/또는 캐시의 최종 레벨로서 사용된다.

[0020] 도 2는 데이터 액세스 시스템(70)을 보여주고 있으며, 여기서 데이터 액세스 시스템(70)은 이미지 프로세싱 디바이스들(72), 시스템 버스(74), FLC 모듈(76), 그리고 스토리지 드라이브(78)를 포함한다. 데이터 액세스 시스템(70)은 예를 들어, 컴퓨터, 모바일 폰, 태블릿, 서버 및/또는 다른 디바이스 내에서 구현될 수 있다. 이미지 프로세싱 디바이스들(72)은 예를 들어, 중앙 프로세서(혹은 중앙 프로세싱 유닛(CPU)); 그래픽 프로세서(혹은 그래픽 프로세싱 유닛(GPU)); 비디오 리코더; 카메라 이미지 신호 프로세서(ISP); 이더넷 인터페이스(예컨대, 기가비트(Gb) 이더넷 인터페이스); 직렬 인터페이스(예컨대, 범용 직렬 버스(USB) 인터페이스 및 직렬 고급 기술 결합(SATA) 인터페이스); 및 주변 컴포넌트 상호연결 익스프레스(PCIe) 인터페이스; 그리고/또는 다른 이미지 프로세싱 디바이스들을 포함할 수 있다. 이미지 프로세싱 디바이스들(72)은 하나 이상의 모듈들 내에 구현될 수 있다. 하나의 예로서, 이미지 프로세싱 모듈들(72) 중 첫 번째 것은, 레벨 0(L0) 캐시, 레벨 1(L1) 캐시, 레벨 2(L2) 캐시, 혹은 레벨 3(L3) 캐시 중 하나 이상과 같은 그러한 캐시 메모리를 포함하는 것으로서 제시되어 있다. 제시된 예에서, 첫 번째 이미지 프로세싱 디바이스는 중앙 프로세서(73) 및 L1 내지 L3 캐시들(75)을 포함할 수 있다. 중앙 프로세서(73)는 L0 캐시(77)를 포함할 수 있다. 중앙 프로세서(73)는 또한 메모리 관리 모듈(Memory Management Module, MMM)(79)을 포함할 수 있고, 메모리 관리 모듈(MMM)(79)은 캐시들(75, 77)에 대한 액세스를 제어할 수 있다. 아래에서 이미지 프로세싱 디바이스에 의해 수행되는 것으로서 설명되는 태스크(task)들은 예를 들어, 중앙 프로세서(73) 및/또는 MMM(79)에 의해 수행될 수 있다.

[0021] 이미지 프로세싱 디바이스들(72)은 시스템 버스(74)를 통해 FLC 모듈(76)에 연결된다. 이미지 프로세싱 디바이스들(72)은 스토리지 드라이브(78)에 연결되는데, 이러한 연결은 (i) 이미지 프로세싱 디바이스들(72)과 시스템 버스(74) 간의 인터페이스들 및 (ii) 시스템 버스(74)와 스토리지 드라이브(78) 간의 인터페이스들 그리고 버스를 통해 이루어진다. 인터페이스들은 예를 들어, 이더넷 인터페이스들, 직렬 인터페이스들, PCIe 인터페이스들, 및/또는 임베디드 멀티-미디어 제어기(embedded Multi-Media Controller, eMMC) 인터페이스들을 포함할 수 있다. 스토리지 드라이브(78)는 이미지 프로세싱 디바이스들(72) 및/또는 FLC 제어기(80)로부터 떨어져 어느 곳에든 위치할 수 있다. 스토리지 드라이브(78)는 하나 이상의 네트워크들(예를 들어, WLAN, 인터넷 네트워크, 또는 원격 스토리지 네트워크(혹은 클라우드(cloud))를 통해 이미지 프로세싱 디바이스들(72) 및/또는 FLC 제어기(80)와 통신할 수 있다.

[0022] FLC 모듈(76)은 FLC 제어기(80), DRAM 제어기(82), 및 DRAM IC(84)를 포함한다. DRAM IC(84)은 주로 가상 스토리지로서 사용되는 반면, 스토리지 드라이브(78)는 물리적 스토리지로서 사용된다. 이미지 프로세싱 디바이스들(72)은 DRAM IC(84) 및 스토리지 드라이브(78)에 어드레싱을 행하는데 마치 이들이 단일 스토리지 디바이스인 것처럼 어드레싱을 행한다. 데이터의 판독 및/또는 기입을 행하는 동안, 이미지 프로세싱 디바이스들(72)은 FLC 제어기(80)에 액세스 요청들을 보낸다. FLC 제어기(80)는 스토리지 드라이브(78)에 액세스하고, 그리고/또는 DRAM 제어기(82)를 통해 DRAM IC(84)에 액세스한다. FLC 제어기(80)는 시스템 버스(74)를 통해 혹은 (점선으로 표시된 바와 같이) 직접적으로 스토리지 드라이브에 액세스할 수 있다.

[0023] 데이터 액세스 시스템(70)의 다양한 예들이 아래에서 설명된다. 제 1 예에서, FLC 모듈(76)은 이미지 프로세싱 디바이스들(72), 시스템 버스(74) 및 스토리지 드라이브(78)로부터 분리되어 SoC 내에 구현된다. 제 2 예에서, 이미지 프로세싱 디바이스들(72) 중 하나의 이미지 프로세싱 디바이스는 CPU로 구현된 이미지 프로세싱 디바이스이다. 이미지 프로세싱 디바이스들(72) 중 이러한 하나의 이미지 프로세싱 디바이스는 FLC 모듈(76) 및 스토리지 드라이브(78)로부터 분리되어 SoC 내에 구현될 수 있다. 또 하나의 다른 예로서, 이미지 프로세싱 디바이스들(72) 및 시스템 버스(74)는 FLC 모듈(76) 및 스토리지 드라이브(78)로부터 분리되어 SoC 내에 구현된다. 또 하나의 다른 예에서, 이미지 프로세싱 디바이스들(72), 시스템 버스(74) 및 FLC 모듈(76)은 스토리지 드라이브(78)로부터 분리되어 SoC 내에 구현된다. 데이터 액세스 시스템(70)의 다른 예들이 아래에서 개시된다.

[0024] DRAM IC(84)는 캐시의 최종 레벨로서 사용될 수 있다. DRAM IC(84)는 다양한 스토리지 용량을 가질 수 있다. 예를 들어, DRAM IC(84)는 모바일 폰 애플리케이션들에 대해 1 내지 4 GB의 스토리지 용량을 가질 수 있고, 개인

용 컴퓨터 애플리케이션들에 대해 4 내지 32 GB의 스토리지 용량을 가질 수 있으며, 그리고 서버 애플리케이션들에 대해 32 GB 내지 1 테라바이트(terabyte)(T)의 스토리지 용량을 가질 수 있다.

[0025] 스토리지 드라이브(78)는 플래시 메모리(예컨대, NAND 플래시 메모리), 솔리드-스테이트 메모리, 솔리드-스테이트 드라이브(Solid-State Drive, SSD), 하드 디스크 드라이브(Hard Disk Drive, HDD) 및/또는 다른 비휘발성 메모리를 포함할 수 있다. 스토리지 드라이브(78)는 하이브리드 드라이브일 수 있고, 예를 들어, 솔리드-스테이트 메모리 및 회전 스토리지 매체들을 포함할 수 있다. 스토리지 드라이브(78)는 DRAM IC(84)보다 더 큰 스토리지 용량을 가질 수 있다. 예를 들어, 스토리지 드라이브(78)는 DRAM IC(84)보다 4배 내지 16배 더 많은 스토리지를 포함할 수 있다. 또 하나의 다른 예로서, 스토리지 드라이브(78)는 DRAM IC(84)보다 임의 차수(order)의 크기만큼 더 큰 스토리지 용량을 가질 수 있다. DRAM IC(84)는 고속 DRAM을 포함할 수 있고, 스토리지 드라이브(78)는 저속 DRAM을 포함할 수 있다. 달리 말하면, DRAM IC(84)의 속도는 스토리지 드라이브(78) 내의 DRAM의 속도보다 더 빠르다.

[0026] DRAM IC(84) 내에 저장된 데이터는, 가장 최근에 액세스된 데이터, 및/또는 가장 빈번하게 액세스된 데이터, 및/또는 관련 우선순위 레벨(priority level)이 가장 높은 데이터일 수 있다. DRAM IC(84) 내에 저장된 데이터는 잠금 상태에 있는 데이터를 포함할 수 있다. 잠금 상태에 있는 데이터는 DRAM IC(84) 내에 항상 저장되어 있는 데이터를 지칭한다. 이러한 잠금 상태의 데이터는 업데이트될 수 있다. DRAM IC(84) 내에 저장된 데이터는 스토리지 드라이브(78) 내에 저장된 데이터보다, 더 빈번하게 액세스될 수 있고 그리고/또는 더 높은 우선순위 레벨들을 가질 수 있다.

[0027] 일부 예들에서, DRAM IC(84) 내에 저장된 데이터는, 빈번하게 사용되지 않는(예를 들어, 소정의 기간에 걸쳐 소정의 횟수보다 더 많이 액세스되지 않은) 상이한 음성 언어들 등을 지원할 소프트웨어 애플리케이션들, 폰트(font)들, 소프트웨어 코드, 대체 코드(alternate code) 및 데이터를 포함하지 않는다. 이것은 DRAM IC(84)의 크기 요건들을 최소화시킴에 있어 도움을 줄 수 있다. 빈번하게 사용되지 않는 소프트웨어 코드는 "가비지 코드(garbage code)"로서 지칭될 수 있으며, 부트업 프로세스 동안 스토리지 드라이브(78)로부터 DRAM IC(84)로 로딩(loading)되지 않을 수 있다. 부트업 프로세스는 빈번하게 사용되는 데이터만을 로딩하는 것을 포함할 수 있다. DRAM IC(84)의 크기가 감소함에 따라, 성능은 증가하고, 전력 소비, 커패시턴스 및 버퍼링은 감소한다. 커패시턴스 및 버퍼링이 감소함에 따라, 레이턴시가 감소한다. 또한, 더 적은 전력을 소비함으로써, 대응하는 디바이스의 배터리 수명이 증가한다.

[0028] FLC 제어기(80)는 이미지 프로세싱 디바이스들(72)로부터의 요청들을 수신함에 응답하여 CAM 기법들을 수행한다. CAM 기법들은 이미지 프로세싱 디바이스들(72)에 의해 제공된 요청들의 제 1 물리적 어드레스를 가상 어드레스들로 변환하는 것을 포함한다. 이러한 가상 어드레스들은, 이미지 프로세싱 디바이스들(72)에 의해 본래 발생하는 가상 어드레스들 아울러 이미지 프로세싱 디바이스들(72)에 의해 제 1 물리적 어드레스들에 맵핑(mapping)되는 그러한 가상 어드레스들과는 다른 것이며 이들로부터 독립되어 있다. DRAM 제어기(82)는 FLC 제어기(80)에 의해 발생된 가상 어드레스들을 제 2 물리적 어드레스들로 변환(혹은 맵핑)한다. 만약 제 2 물리적 어드레스들이 DRAM IC(84) 내에 존재하지 않는다면, FLC 제어기(80)는, (i) 스토리지 드라이브(78)로부터 데이터를 페치(fetch)할 수 있거나, 혹은 (ii) 캐시 미스(cache miss)가 발생했다고 이미지 프로세싱 디바이스들(72) 중 대응하는 이미지 프로세싱 디바이스에 표시할 수 있다(혹은 신호를 보낼 수 있다). 스토리지 드라이브(78)로부터 데이터를 페치하는 것은, FLC 제어기(80)에 의해 발생된 가상 어드레스들을 스토리지 드라이브(78) 내의 데이터에 액세스하기 위한 제 3 물리적 어드레스에 맵핑시키는 것을 포함할 수 있다. 캐시 미스는 물리적 어드레스를 가상 어드레스로 변환하는 동안 FLC 제어기(80)에 의해 검출될 수 있고, 그리고/또는 가상 어드레스를 제 2 물리적 어드레스로 변환하는 동안 DRAM 제어기(82)에 의해 검출될 수 있다. DRAM 제어기(82)는 캐시 미스가 발생했다는 신호를 FLC 제어기(80)에 보낼 수 있고, 그 다음에 FLC 제어기(80)는 캐시 미스에 관한 신호를 이미지 프로세싱 디바이스들(72) 중 하나에 보낼 수 있거나, 혹은 해당 데이터에 대해 스토리지 드라이브(78)에 액세스할 수 있다.

[0029] 만약 FLC 제어기(80)가 캐시 미스가 발생했다는 신호를 이미지 프로세싱 디바이스들(72) 중 대응하는 이미지 프로세싱 디바이스에게 보낸다면, 이미지 프로세싱 디바이스들(72) 중 대응하는 그 이미지 프로세싱 디바이스는 스토리지 드라이브(78) 내의 데이터에 액세스할 수 있다. 이것은, (i) 제 1 물리적 어드레스들에 근거하여 스토리지 드라이브(78) 내의 데이터에 액세스하는 것, (ii) 본래 발생된 가상 어드레스들을 제 3 물리적 어드레스들에 맵핑시키고, 그 다음에 제 3 물리적 어드레스들에 근거하여 스토리지 드라이브(78)에 액세스하는 것, 또는 (iii) 제 1 물리적 어드레스들을 제 3 물리적 어드레스들에 맵핑시키고, 그 다음에 제 3 물리적 어드레스들에 근거하여 스토리지 드라이브(78)에 액세스하는 것을 포함할 수 있다. CAM 기법들은 DRAM IC(84)와 스토리지 드라

이브(78) 간의 어드레스 매핑을 포함한다.

[0030] 예를 들어, CAM 기법들은 풀 세트 연관 어드레스 변환(full set associative address translation)을 제공할 수 있다. 이것은 제 1 물리적 어드레스들을 FLC 제어기(80)의 디렉토리(directory) 내에 저장된 모든 가상 어드레스들과 비교하는 것을 포함할 수 있다. 풀 세트 연관 어드레스 변환은 대응하는 데이터 요청들의 제 1 물리적 어드레스들을 제 2 가상 어드레스들(혹은 FLC 제어기(80)에 의해 발생된 가상 어드레스들)로 변환하기 위해 FLC 제어기(80)에 의해 사용될 수 있다. 이것은 가상 어드레스들로의 변환을 위해 매칭되는 것들을 찾기 위해서 제 1 물리적 어드레스들을 제 1 디렉토리 내의 모든 어드레스들과 비교하는 것을 포함할 수 있다. 풀 세트 연관 어드레스 변환은 또한, 제 2 물리적 어드레스들로의 변환을 위해 매칭되는 것들을 찾기 위해서 제 2 가상 어드레스들을 제 2 디렉토리, DRAM IC(84), 및/또는 스토리지 드라이브(78) 내의 모든 엔트리(entry)들의 모든 어드레스들과 비교하기 위해 DRAM 제어기(82)에 의해 사용될 수 있다. 초기 부트업 이후 DRAM IC(84) 내에 데이터가 위치하고 있을 히트율(hit rate)은 DRAM IC(84)의 크기에 따라 100%만큼 높을 수 있다. DRAM IC(84)는 프로세서 및/또는 이미지 프로세싱 디바이스의 최소의 아이들 타임(idle time)으로 거의 100%의 히트율을 보장하기 위해 크기조정될 수 있다. 예를 들어, 이것은 모바일 폰 애플리케이션들에 대해 1 내지 4 GB의 DRAM IC를 사용하여 달성될 수 있고, 개인용 컴퓨터 애플리케이션들에 대해 4 내지 32 GB의 DRAM IC들을 사용하여 달성될 수 있으며, 그리고 서버 애플리케이션들에 대해 32 내지 500 GB의 DRAM IC들을 사용하여 달성될 수 있다.

[0031] 도 2에 추가하여, 도 3은 데이터 액세스 시스템(70)의 DRAM IC(84) 및 스토리지 드라이브(78)의 엔트리들을 보여준다. DRAM IC(84)는 DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN}를 포함할 수 있다. 스토리지 드라이브(78)는 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN}를 가질 수 있다. DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 각각의 어드레스들은 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN}의 하나 이상의 어드레스들에 매핑될 수 있다. 결과적으로, DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 중 하나 이상의 DRAM 엔트리 내에 저장된 데이터의 제 1 부분들은 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN} 내에 저장된 데이터의 제 2 부분들에 대응할 수 있다. DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 내에 저장된 데이터는 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN} 내에 저장된 데이터에 관한 정보를 제공하는 메타데이터(metadata)를 포함할 수 있다. 이러한 정보는 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN} 내에 저장된 데이터의 데이터 타입들 및/또는 길이들을 포함할 수 있다.

[0032] 하나의 예로서, DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 각각은 예를 들어, 16 KB의 스토리지 용량을 가질 수 있다. 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN} 각각은 16 GB의 스토리지 용량을 가질 수 있다. 만약 DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 중 하나의 DRAM 엔트리로부터 데이터가 관독돼야 하거나 DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 중 하나의 DRAM 엔트리에 데이터가 기입돼야 하고, DRAM 엔트리₀₀ 내지 DRAM 엔트리_{MN} 중 그 하나의 DRAM 엔트리가 풀(full) 상태에 있다면 그리고/또는 요청과 관련된 데이터를 모두 가지고 있는 것이 아니라면, 드라이브 엔트리₀₀ 내지 드라이브 엔트리_{MN} 중 대응하는 하나의 드라이브 엔트리에 대한 액세스가 일어난다. 따라서, DRAM IC(84) 및 스토리지 드라이브(78)는 메모리의 블록(block)들로 분할될 수 있다. DRAM IC(84)에서의 메모리의 각각의 블록은 스토리지 드라이브(78)에서의 메모리의 각각의 하나 이상의 블록들을 가질 수 있다. 메모리의 이러한 매핑 및 분할은 도 2의 이미지 프로세싱 디바이스들(72)에 대해 투명한 것일 수 있다.

[0033] 동작 동안, 이미지 프로세싱 디바이스들(72) 중 하나는 데이터의 블록에 대한 요청 신호를 발생시킬 수 있다. 만약 데이터의 블록이 DRAM IC(84) 내에 위치하고 있지 않다면, FLC 제어기(80)는 스토리지 드라이브(78) 내의 데이터의 블록에 액세스할 수 있다. FLC 제어기(80)가 스토리지 드라이브(78)로부터의 데이터에 액세스하고 있는 동안, FLC 제어기(80)는 데이터를 요청한 이미지 프로세싱 디바이스에게 역으로 버스 에러 신호(bus error signal)(혹은 경고 신호(alert signal))를 보낼 수 있다. 버스 에러 신호는, FLC 제어기(80)가 데이터에 액세스하고 있고 그 결과로서 시스템 버스(74)는 이미지 프로세싱 디바이스(72)로 데이터를 전송할 준비가 되어 있지 않음을 표시할 수 있다. 이러한 버스 에러 신호의 전송은, FLC 모듈(76)로부터 이미지 프로세싱 디바이스 및/또는 이미지 프로세싱 디바이스(72)의 SoC로의 "버스 중단(bus abort)"으로서 지칭될 수 있다. 이미지 프로세싱 디바이스(72)는 버스가 준비되기를 기다리는 동안 다른 태스크들을 수행할 수 있다. 다른 태스크들은, 예를 들어, 이미지 프로세싱 디바이스의 SoC 내의 하나 이상의 캐시들(예컨대, L0 내지 L3 캐시들)에 이미 저장된 데이터를 사용하는 것을 포함할 수 있다. 이것은 또한, 프로세서 및/또는 이미지 프로세싱 디바이스의 아이들 타임을 최소화시킨다.

- [0034] 만약 순차적 액세스가 수행된다면, FLC 제어기(80) 및/또는 DRAM 제어기(82)는 장래에 액세스될 것으로 예상되는 어드레스들에 저장된 데이터의 예측 페칭(predictive fetching)을 수행할 수 있다. 이것은 부트업 동안 그리고/또는 부트업 이후에 일어날 수 있다. FLC 제어기(80) 및/또는 DRAM 제어기(82)는, 데이터 및/또는 소프트웨어 사용을 추적할 수 있고; 실행될 코드의 곧 나타날 라인들을 평가할 수 있고; 메모리 액세스 패턴들을 추적할 수 있고; 그리고 이러한 정보에 근거하여 액세스될 것으로 예상되는 데이터의 다음 어드레스들을 예측할 수 있다. 다음 어드레스들은 DRAM IC(84) 및/또는 스토리지 드라이브(78)의 어드레스들일 수 있다. 하나의 예로서, FLC 제어기(80) 및/또는 DRAM 제어기(82)는, 데이터에 대한 요청을 이전에 수신함이 없이 그리고/또는 이와는 독립적으로, 스토리지 드라이브(78) 내에 저장된 데이터에 액세스할 수 있고, 이 데이터를 DRAM IC(84)로 전송할 수 있다. 또 하나의 다른 예로서, FLC 제어기(80) 및/또는 DRAM 제어기(82)는, 데이터에 대한 요청을 이전에 수신함이 없이 그리고/또는 이와는 독립적으로, DRAM IC(84) 내에 저장된 데이터에 액세스할 수 있고, 이 데이터를 이미지 프로세싱 디바이스들(72) 중 하나 및/또는 시스템 버스(74)로 전송할 수 있다.
- [0035] 데이터 액세스 시스템(70)의 동작이 도 4의 방법과 관련하여 더 설명된다. 비록 아래에서 설명되는 TASK들이 주로 도 2 내지 도 4에서의 예들과 관련하여 설명되지만, 이러한 TASK들은 본 개시내용에서의 다른 예들에 적용되도록 용이하게 수정될 수 있다. TASK들은 반복적으로 수행될 수 있다.
- [0036] 본 방법은 도면번호 "100"에서 시작한다. 도면번호 "102"에서, FLC 제어기(80)는 이미지 프로세싱 디바이스들(72) 중 하나로부터의 요청을 수신한다. 이러한 요청은 판독 요청 혹은 기입 요청일 수 있다.
- [0037] 도면번호 "104"에서, FLC 제어기(80)는 수신된 요청이 판독 요청인지 여부를 결정한다. 만약 요청이 판독 요청이면 TASK(106)가 수행되고, 그렇지 않다면 TASK(122)가 수행된다. 도면번호 "106"에서, FLC 제어기(80)는 수신된 요청에서 제공된 제 1 물리적 어드레스를 가상 어드레스로 변환한다. 가상 어드레스는 DRAM IC(84) 내의 물리적 어드레스 및/또는 스토리지 드라이브(78) 내의 물리적 어드레스와 관련될 수 있다.
- [0038] 도면번호 "108"에서, DRAM 제어기(82)는 가상 어드레스를 제 2 물리적 어드레스로 변환한다. FLC 제어기(80) 및/또는 DRAM 제어기(82)는, 제 1 물리적 어드레스들을 가상 어드레스들에 맵핑시키기 위한 아올러 가상 어드레스들을 제 2 물리적 어드레스들에 맵핑시키기 위한 하나 이상의 변환 맵핑 테이블(translation mapping table)들을 포함할 수 있다. 이러한 테이블들은 FLC 제어기(80)와 DRAM 제어기(82) 간에 공유될 수 있다.
- [0039] 도면번호 "110"에서, DRAM 제어기(82)는 제 2 물리적 어드레스가 DRAM IC(84) 내에 존재하는지 여부, 이에 따라 데이터의 대응하는 블록이 DRAM IC(84)의 할당된 공간에 있는지 여부를 결정한다. 만약 제 2 물리적 어드레스가 DRAM IC(84) 내에 있다면, TASK(112)가 수행되고, 그렇지 않다면 TASK(114)가 수행된다.
- [0040] 도면번호 "112"에서는, 제 2 물리적 어드레스에 근거하여, DRAM 제어기(82)가 그 수신된 요청과 관련된 데이터의 하나 이상의 제 1 블록들을 DRAM IC(84)로부터 판독한다. 데이터의 제 1 블록들 각각은 DRAM IC(84) 내의 대응하는 물리적 어드레스를 가질 수 있다.
- [0041] 도면번호 "114"에서는, 제 2 물리적 어드레스에 근거하여, FLC 제어기(80)가 그 수신된 요청과 관련된 데이터의 하나 이상의 제 2 블록들을 스토리지 드라이브(78)로부터 판독한다. 데이터의 하나 이상의 제 2 블록들은 데이터의 제 1 블록들을 포함할 수 있다. 데이터의 제 2 블록들 각각은 스토리지 드라이브(78) 내의 대응하는 물리적 어드레스를 가질 수 있다. 스토리지 드라이브(78)의 데이터의 제 2 블록들 및 대응하는 어드레스들은 DRAM IC(84)의 데이터의 제 1 블록들 및 대응하는 어드레스들에 각각 대응할 수 있다.
- [0042] 도면번호 "116"에서, DRAM IC(84) 혹은 스토리지 드라이브(78)로부터 판독된 데이터는 수신된 요청을 보낸 이미지 프로세싱 디바이스로 포워딩(forwarding)된다. 도면번호 "118"에서는, 도면번호 "102"에서 수신된 요청에 대해, FLC 제어기(80) 및/또는 DRAM 제어기(82)가 DRAM IC(84) 및/또는 스토리지 드라이브(78)로부터 판독될 추가 데이터가 존재하는지 여부를 결정할 수 있다. 만약 판독될 추가 데이터가 존재한다면, TASK(110)가 수행되고, 그렇지 않다면, 본 방법은 도면번호 "120"에서 끝날 수 있다. TASK(110)로 되돌아가는 것에 대한 대안예로서, 도면번호 "102"에서 수신된 요청에 대해, 만약 판독될 추가 데이터가 존재한다면, 그리고 변환될 또 하나의 다른 물리적 어드레스가 존재한다면, TASK(118) 이후에 TASK(106)가 수행될 수 있다.
- [0043] 도면번호 "122"에서, FLC 제어기(80)는 수신된 요청이 기입 요청인지 여부를 결정한다. 만약 수신된 요청이 기입 요청이면, TASK(126)가 수행되고, 그렇지 않다면, 본 방법은 도면번호 "124"에서 끝날 수 있다. 도면번호 "126"에서, FLC 제어기(80)는 수신된 요청에서 제공된 제 1 물리적 어드레스를 가상 어드레스로 변환한다. 가상 어드레스는 DRAM IC(84) 내의 물리적 어드레스 및/또는 스토리지 드라이브(78) 내의 물리적 어드레스와 관련될 수 있다. 도면번호 "128"에서, DRAM 제어기(82)는 가상 어드레스를 제 2 물리적 어드레스로 변환한다. TASK

(126)와 태스크(128)는 앞서-언급된 맵핑 테이블들을 이용하여 달성될 수 있다.

- [0044] 도면번호 "130"에서, DRAM 제어기(82)는 제 2 물리적 어드레스가 DRAM IC(84) 내에 존재하는지 여부, 이에 따라 데이터의 대응하는 블록이 DRAM IC(84) 내의 할당된 공간에 기입될 것인지 여부를 결정한다. 만약 제 2 물리적 어드레스가 DRAM IC(84) 내에 있다면, 태스크(132)가 수행되고, 그렇지 않다면 태스크(134)가 수행된다.
- [0045] 도면번호 "132"에서는, 제 2 물리적 어드레스에 근거하여, DRAM 제어기(82)가 그 수신된 요청과 관련된 데이터의 하나 이상의 제 1 블록들을 DRAM IC(84)에 기입한다. 데이터의 제 1 블록들 각각은 DRAM IC(84) 내의 대응하는 물리적 어드레스를 가질 수 있다.
- [0046] 도면번호 "134"에서는, 제 2 물리적 어드레스에 근거하여, FLC 제어기(80)가 그 수신된 요청과 관련된 데이터의 하나 이상의 제 2 블록들을 스토리지 드라이브(78)에 기입한다. 데이터의 하나 이상의 제 2 블록들은 데이터의 제 1 블록들을 포함할 수 있다. 데이터의 제 2 블록들 각각은 스토리지 드라이브(78) 내의 대응하는 물리적 어드레스를 가질 수 있다. 스토리지 드라이브(78)의 데이터의 제 2 블록들 및 대응하는 어드레스들은 DRAM IC(84)의 데이터의 제 1 블록들 및 대응하는 어드레스들에 각각 대응할 수 있다.
- [0047] 도면번호 "136"에서는, 도면번호 "102"에서 수신된 요청에 대해, FLC 제어기(80) 및/또는 DRAM 제어기(82)가 DRAM IC(84) 및/또는 스토리지 드라이브(78)에 기입될 추가 데이터가 존재하는지 여부를 결정할 수 있다. 만약 기입될 추가 데이터가 존재한다면, 태스크(130)가 수행되고, 그렇지 않다면, 본 방법은 도면번호 "138"에서 끝날 수 있다. 태스크(130)로 되돌아가는 것에 대한 대안으로서, 도면번호 "102"에서 수신된 요청에 대해, 만약 기입될 추가 데이터가 존재한다면, 그리고 변환될 또 하나의 다른 물리적 어드레스가 존재한다면, 태스크(136) 이후에 태스크(126)가 수행될 수 있다.
- [0048] 도 4의 앞서-설명된 태스크들은 예시적 사례들이 되도록 의도되었고, 여기서 태스크들은 애플리케이션에 따라 상이한 순서로 혹은 중복되는 기간 동안 순차적으로, 동기화되어, 동시에, 연속적으로 수행될 수 있다. 또한, 태스크들 중 임의의 태스크는 이벤트들의 시퀀스 및/또는 사례에 따라 수행되지 않을 수 있거나 혹은 스킵(skip)될 수 있다.
- [0049] 앞서-설명된 예들은 ("클라우드(cloud)"로서 지칭될 수 있는) 네트워크 내의 서버들을 통해 구현될 수 있다. 서버들 각각은 FLC 모듈(예컨대, FLC 모듈(76))을 포함할 수 있고, 서로 통신할 수 있다. 서버들은 DRAM IC들 및 스토리지 드라이브들 내에 저장된 DRAM 및/또는 메모리를 공유할 수 있다. 서버들 각각은 네트워크를 통해 다른 서버들 내의 DRAM들 및/또는 스토리지 드라이브들에 액세스할 수 있다. FLC 모듈들 각각은 도 2의 FLC 모듈과 유사하게 동작할 수 있으며, 하지만 클라우드를 통해 다른 서버들 각각 내의 DRAM 및/또는 메모리에 액세스할 수 있다. 서버들과 클라우드 간에 전송된 신호들은 전송이 일어나기 전에 암호화될 수 있으며, 클라우드의 네트워크 디바이스 및/또는 서버에 도달하면 복호화될 수 있다. 서버들은 또한, 클라우드 내에서 메모리를 공유할 수 있고 그리고/또는 메모리에 액세스할 수 있다. 하나의 예로서, 서버들 중 하나의 서버의 FLC 제어기에 의해 발생된 가상 어드레스는, FLC 제어기의 FLC 모듈의 DRAM; 서버들 중 하나의 서버의 스토리지 드라이브; 다른 서버들 중 하나의 서버의 FLC 모듈의 DRAM; 다른 서버들 중 하나의 서버의 스토리지 드라이브; 혹은 클라우드의 스토리지 디바이스 내의 물리적 어드레스에 대응할 수 있다. 서버들 중 하나의 서버의 이미지 프로세싱 디바이스 및/또는 FLC 제어기는, 캐시 미스가 일어나면, 다른 FLC 모듈들, 스토리지 드라이브들, 및/또는 스토리지 디바이스들 내의 DRAM 및/또는 메모리에 액세스할 수 있다.
- [0050] 앞서-설명된 예들은 또한, 복수의 칩들을 갖는 복수-칩 모듈; 스위치; 그리고 1차 FLC 모듈을 갖는 1차 칩을 포함하는 그러한 데이터 액세스 시스템 내에서 구현될 수 있다. 복수-칩 모듈은 스위치를 통해 1차 칩 모듈에 연결된다. FLC 모듈들 각각은 도 2의 FLC 모듈과 유사하게 동작할 수 있으며, 하지만 스위치를 통해 다른 칩들 각각 내의 DRAM 및/또는 메모리에 액세스할 수 있다. 하나의 예로서, 칩들 중 하나의 칩의 FLC 제어기에 의해 발생된 가상 어드레스는, FLC 제어기의 FLC 모듈의 DRAM; 칩들 중 하나의 칩의 스토리지 드라이브; 다른 칩들 중 하나의 칩의 FLC 모듈의 DRAM; 다른 칩들 중 하나의 칩의 스토리지 드라이브; 또는 클라우드의 스토리지 디바이스 내의 물리적 어드레스에 대응할 수 있다. 칩들 중 하나의 칩의 이미지 프로세싱 디바이스 및/또는 FLC 제어기는, 캐시 미스가 일어나면, 다른 FLC 모듈들, 스토리지 드라이브들, 및/또는 스토리지 디바이스들 내의 DRAM 및/또는 메모리에 액세스할 수 있다.
- [0051] 하나의 예로서, 복수-칩 모듈 내의 2차 DRAM들 각각과 1차 칩 내의 1 차 DRAM은 1 GB의 스토리지 용량을 가질 수 있다. 1차 칩 내의 스토리지 드라이브는 예를 들어, 64 GB의 스토리지 용량을 가질 수 있다. 또 하나의 다른 예로서, 데이터 액세스 시스템은 자동차에서 사용될 수 있다. 1차 칩은 예를 들어, 중앙 제어기, 모듈, 프로세

서, 엔진 제어 모듈, 트랜스미션 제어 모듈(transmission control module), 및/또는 하이브리드 제어 모듈일 수 있다. 1차 칩은 스로틀 위치(throttle position), 스파크 타이밍(spark timing), 연료 타이밍(fuel timing), 트랜스미션 기어들 간의 전이(transition)들 등과 같은 관련 시스템들의 대응하는 양상들을 제어하기 위해 사용될 수 있다. 복수-칩 모듈 내의 2차 칩들은 각각, 점등 시스템(lighting system), 엔터테인먼트 시스템(entertainment system), 공조 시스템(air-conditioning system), 배기 시스템(exhaust system), 네비게이션 시스템(navigation system), 오디오 시스템(audio system), 비디오 시스템(video system), 제동 시스템(braking system), 조향 시스템(steering system) 등과 같은 특정 차량 시스템과 관련될 수 있고, 그리고 대응하는 시스템들의 양상들을 제어하기 위해 사용될 수 있다.

[0052] 또 하나의 다른 예로서, 앞서-설명된 예들은 또한, 호스트(혹은 SoC) 및 하이브리드 드라이브를 포함하는 그러한 데이터 액세스 시스템 내에서 구현될 수 있다. 호스트는 중앙 프로세서 혹은 다른 이미지 프로세싱 디바이스를 포함할 수 있고, 그리고 인터페이스를 통해 하이브리드 드라이브와 통신할 수 있다. 인터페이스는 예를 들어, GE 인터페이스, USB 인터페이스, SATA 인터페이스, PCIe 인터페이스, 혹은 다른 적절한 인터페이스일 수 있다. 하이브리드 드라이브는 제 1 스토리지 드라이브 및 제 2 스토리지 드라이브를 포함한다. 제 1 스토리지 드라이브는 FLC 모듈(예를 들어, 도 2의 FLC 모듈(76))을 포함한다. FLC 모듈의 FLC 제어기는, 제 2 스토리지 드라이브 및 FLC 모듈의 DRAM으로부터 데이터를 판독할 것인지 여부 그리고/또는 제 2 스토리지 드라이브 및 FLC 모듈의 DRAM에 데이터를 기입할 것인지 여부를 결정할 때, CAM 기법들을 수행한다

[0053] 다른 예로서, 앞서-설명된 예들은 또한, SoC, 제 1 DRAM(혹은 캐시), 제 2 DRAM, 및 비휘발성 메모리를 포함하는 그러한 스토리지 시스템 내에서 구현될 수 있다. SoC는 제 1 DRAM, 별개의 DRAM, 및 비휘발성 메모리로부터 분리되어 있다. 제 1 DRAM은 우선순위가 높고 그리고/또는 빈번하게 액세스되는 데이터를 저장할 수 있다. 데이터 액세스 요청들 중 퍼센티지가 높은 데이터 액세스 요청들이 제 1 DRAM 내에 저장되는 데이터로 지향될 수 있다. 하나의 예로서, 데이터 액세스 요청들 중 99% 혹은 그 이상의 데이터 액세스 요청들이 제 1 DRAM 내에 저장되는 데이터로 지향될 수 있고, 데이터 액세스 요청들 중 잔존하는 1% 혹은 그 이하의 데이터 액세스 요청들이 제 2 DRAM 및/또는 비휘발성 메모리 내에 저장되는 데이터로 지향될 수 있다. 우선순위가 낮고 그리고/또는 덜 빈번하게 액세스되는 데이터는 제 2 DRAM 및/또는 비휘발성 메모리 내에 저장될 수 있다. 하나의 예로서, 사용자는 개방된 상태의 복수의 웹 브라우저들을 가질 수 있다. 제 2 DRAM은 고-대역폭 인터페이스 및 비트-당-저-비용 메모리 스토리지 용량을 가지며, 다양한 범위의 애플리케이션들을 처리할 수 있다. 제 2 DRAM은 전형적으로 캐시 혹은 제 1 DRAM보다 더 긴 레이턴시 기간을 보여준다. 현재 사용되거나 혹은 가장 빈번하게 사용되는 웹 브라우저에 대한 콘텐츠가 제 1 DRAM 내에 저장될 수 있고, 반면 다른 웹 브라우저들에 대한 콘텐츠는 제 2 DRAM 및/또는 비휘발성 메모리 내에 저장될 수 있다.

[0054] SoC는 하나 이상의 제어 모듈들, 인터페이스 모듈, 캐시(혹은 FLC) 모듈, 그리고 그래픽 모듈을 포함할 수 있다. 캐시 모듈은 도 2의 FLC 모듈과 유사하게 동작할 수 있다. 제어 모듈들은 인터페이스 모듈을 통해 캐시 모듈에 연결된다. 캐시 모듈은 각각의 계층적 레벨들에 근거하여 제 1 DRAM, 제 2 DRAM, 및 비-휘발성 메모리에 액세스하도록 구성된다. 제어 모듈들 각각은 각각의 L1, L2, 및 L3 캐시들을 포함할 수 있다. 제어 모듈들 각각은 또한, 하나 이상의 추가적인 캐시들을 포함할 수 있는데, 예컨대, L4 캐시 혹은 다른 가장 높은 레벨의 캐시를 포함할 수 있다. 많은 신호 라인들(혹은 전도성 요소들)이 SoC와 제 1 DRAM 사이에 존재할 수 있다. 이것은 SoC와 제 1 DRAM 간에 빠른 병렬 및/또는 직렬 데이터 전송이 일어나게 할 수 있다. SoC와 제 1 DRAM 간의 데이터 전송은, (i) SoC와 제 2 DRAM 간의 데이터 전송, 및 (ii) SoC와 비휘발성 메모리 간의 데이터 전송보다 더 빠르다.

[0055] 제 1 DRAM은 L3 캐시(622), L4 캐시 및/또는 가장 높은 레벨의 캐시와 비교하여 동일한 혹은 더 높은 계층적 레벨을 갖는 제 1 부분을 구비할 수 있다. 제 1 DRAM의 제 2 부분은 제 2 DRAM 및/또는 비휘발성 메모리와 비교하여 동일한 혹은 더 낮은 계층적 레벨을 가질 수 있다. 제 2 DRAM은 제 1 DRAM보다 더 높은 계층적 레벨을 가질 수 있다. 비휘발성 메모리는 제 2 DRAM과 비교하여 동일한 혹은 더 높은 계층적 레벨을 가질 수 있다. 제어 모듈들은 예를 들어, 캐시의 필요에 근거하여, 제 1 DRAM, 제 2 DRAM, 및/또는 비휘발성 메모리 각각의 일부분들 혹은 모두의 계층적 레벨들을 변경시킬 수 있다.

[0056] 제어 모듈들, 인터페이스 모듈에 연결된 그래픽 모듈, 및/또는 인터페이스 모듈에 연결된 (Soc 내부 혹은 외부에 있는) 다른 디바이스들은, 제 1 DRAM, 제 2 DRAM, 및/또는 비휘발성 메모리 내에 데이터를 저장하기 위해 그리고/또는 제 1 DRAM, 제 2 DRAM, 및/또는 비휘발성 메모리 내의 데이터에 액세스하기 위해, 캐시 모듈에 요청 신호들을 보낼 수 있다. 캐시 모듈은 제 1 DRAM, 제 2 DRAM, 및 비휘발성 메모리에 대한 액세스를 제어할 수 있다. 하나의 예로서, 제어 모듈들, 그래픽 모듈, 및/또는 (인터페이스 모듈에 연결된) 다른 디바이스들은 SoC에

연결된 DRAM들의 개수 및/또는 크기를 모르고 있을 수 있다.

- [0057] 캐시 모듈은, 제어 모듈들, 그래픽 모듈, 및/또는 (인터페이스 모듈에 연결된) 다른 디바이스들로부터 수신된 제 1 논리적 블록 어드레스(logical block address)들 및 요청들을, (i) 제 1 DRAM, 제 2 DRAM, 및/또는 비휘발성 메모리의 물리적 블록 어드레스들 혹은 (ii) 제 2 논리적 블록 어드레스들로 변환할 수 있다. 제 2 논리적 블록 어드레스들은 제 1 DRAM, 제 2 DRAM, 및/또는 비휘발성 메모리에 의해 물리적 블록 어드레스들로 변환될 수 있다. 캐시 모듈은, 제 1 논리적 블록 어드레스들의 제 2 논리적 블록 어드레스들로의 변환 그리고/또는 제 1 논리적 블록 어드레스들의 물리적 블록 어드레스들로의 변환을 위한 하나 이상의 룩업 테이블(lookup table)들(예를 들어, 풀 세트 연관 룩업 테이블들)을 저장할 수 있다. 결과적으로, 제 1 DRAM, 제 2 DRAM, 및 비휘발성 메모리 중 하나 이상의 것과 캐시 모듈은, 제어 모듈들, 그래픽 모듈, 및/또는 (인터페이스 모듈에 연결된) 다른 디바이스들에 대해 단일 메모리로서 동작할 수 있다. 그래픽 모듈은 제어 모듈들 및/또는 SoC로부터의 비디오 데이터를 디스플레이 및/또는 다른 비디오 디바이스에 출력하는 것을 제어할 수 있다.
- [0058] 제어 모듈들은, (i) 캐시 모듈과 (ii) L1 캐시, L2 캐시, 및 L3 캐시 간에 데이터, 데이터 세트들, 프로그램들, 및/또는 그 일부분들을 스와핑(swapping)(혹은 전송)할 수 있다. 캐시 모듈은, 제 1 DRAM, 제 2 DRAM 및 비휘발성 메모리 중 둘 이상의 것들 간에 데이터, 데이터 세트들, 프로그램들, 및/또는 그 일부분들을 스와핑(혹은 전송)할 수 있다. 이것은 전송을 수행하기 위해 제어 모듈들로부터 제어 신호들을 수신함이 없이 그리고/또는 제어 모듈들과는 독립적으로 수행될 수 있다. 제 1 DRAM, 제 2 DRAM, 및 비휘발성 메모리 중 하나 이상의 것 내에서의 데이터, 데이터 세트들, 프로그램들 및/또는 그 일부분들의 저장 위치는, 대응하는 우선순위 레벨들, 사용 빈도, 액세스 빈도, 및/또는 (데이터, 데이터 세트들, 프로그램들 및/또는 그 일부분들과 관련된) 다른 파라미터들에 근거할 수 있다. 데이터, 데이터 세트들, 프로그램들 및/또는 그 일부분들의 전송은 데이터의 블록들을 전송하는 것을 포함할 수 있다. 데이터의 블록들 각각은 미리결정된 크기를 가질 수 있다. 하나의 예로서, 제 2 DRAM으로부터 제 1 DRAM으로의 데이터의 스와핑은 복수의 전송 이벤트들을 포함할 수 있고, 여기서 각각의 전송 이벤트는 데이터의 블록(예를 들어, 16 KB의 데이터)을 전송하는 것을 포함한다.
- [0059] 캐시 모듈은, 제 1 DRAM을 세트 연관 캐시(set associative cache), 직접 맵핑 캐시(directed mapped cache), 및/또는 완전 연관 캐시(fully associative cache)로서 사용할 수 있고, 대응하는 캐시 프로토콜(cache protocol)들을 사용하여 제 1 DRAM에 액세스할 수 있다. 캐시 모듈은 마지막(혹은 가장 높은) 레벨의 캐시(Last Level Cache, LLC) 제어기로서 지칭될 수 있다.
- [0060] 제 1 DRAM은 제1의 미리결정된 양의 스토리지 용량(예를 들어, 0.25 GB, 0.5 GB, 1 GB, 혹은 8 GB)을 가질 수 있다. 0.5 GB의 제 1 DRAM은 전형적인 L2 캐시와 비교해 512배만큼 크다. 제 2 DRAM은 제2의 미리결정된 양의 스토리지 용량을 가질 수 있다(예를 들어, 비-서버 기반의 시스템들에 대해서는 1 GB 혹은 그 이상, 또는 서버 기반 시스템들에 대해서는 256 GB 혹은 그 이상). 비휘발성 메모리는 제3의 미리결정된 양의 스토리지 용량(예를 들어, 256 GB 혹은 그 이상)을 가질 수 있다. 비휘발성 메모리는 솔리드-스테이트 메모리를 포함할 수 있는 바, 예컨대, 플래시 메모리 혹은 자기저항식 랜덤 액세스 메모리(Magnetoresistive Random Access Memory, MRAM), 및/또는 회전 자기 매체들(rotating magnetic media)을 포함할 수 있다. 비휘발성 메모리는 SSD 및 HDD를 포함할 수 있다. 스토리지 시스템이 제 2 DRAM 및 비휘발성 메모리를 가지고 있지만, 제 2 DRAM과 비휘발성 메모리 중 어느 하나는 스토리지 시스템 내에 포함되지 않을 수 있다.
- [0061] 다른 예로서, 앞서-설명된 예들은 또한, SoC 및 DRAM IC를 포함하는 그러한 스토리지 시스템 내에서 구현될 수 있다. SoC는 링 버스(ring bus)를 통해 DRAM IC에 액세스하는 복수의 제어 모듈들(혹은 프로세서들)을 포함할 수 있다. 링 버스는 액세스 레이턴시들을 최소화시키는 쌍-방향 버스일 수 있다. 만약 성능보다 비용이 더 중요하다면, 링 버스는 단방향 버스일 수 있다. 제어 모듈들과 링 버스 사이 그리고/또는 링 버스와 DRAM IC 사이에는 중간 디바이스들이 위치할 수 있다. 예를 들어, 제어 모듈들과 링 버스 사이 또는 링 버스와 DRAM IC 사이에는 앞서-설명된 캐시 모듈이 위치할 수 있다.
- [0062] DRAM IC는 L3, L4 및/또는 가장 높은 레벨의 캐시일 수 있다. 제어 모듈들은 DRAM IC를 공유할 수 있고, 그리고/또는 DRAM IC의 지정된 부분들을 가질 수 있다. 예를 들어, DRAM IC의 제 1 부분은 제 1 제어 모듈에 대한 캐시로서 할당될 수 있다. DRAM IC의 제 2 부분은 제 2 제어 모듈에 대한 캐시로서 할당될 수 있다. DRAM IC의 제 3 부분은 제 3 제어 모듈에 대한 캐시로서 할당될 수 있다. DRAM IC의 제 4 부분은 캐시로서 할당되지 않을 수 있다.
- [0063] 다른 예로서, 앞서-설명된 예들은 또한, 서버 시스템 내에서 구현될 수 있다. 서버 시스템은 스토리지 시스템으로서 지칭될 수 있고, 복수의 서버들을 포함할 수 있다. 서버들은 네트워크(혹은 클라우드)를 통해 서로 통신하

는 각각의 스토리지 시스템들을 포함한다. 스토리지 시스템들 중 하나 이상은 클라우드 내에 위치할 수 있다. 스토리지 시스템들 각각은 각각의 SoC들을 포함할 수 있다.

- [0064] SoC들은 각각의 제 1 DRAM들, 제 2 DRAM들, 솔리드-스테이트 비휘발성 메모리들, 비휘발성 메모리들 및 I/O 포트들을 가질 수 있다. I/O 포트들은, 주변 컴포넌트 상호연결 익스프레스(PCIe) 채널들과 같은 각각의 I/O 채널들, 그리고 주변 컴포넌트 상호연결 익스프레스(PCIe) 채널들과 같은 각각의 네트워크 인터페이스들을 통해 클라우드와 통신할 수 있다. I/O 포트들, I/O 채널들, 및 네트워크 인터페이스들은 이더넷 포트들, 채널들, 네트워크 인터페이스들일 수 있고, 미리결정된 속도(예를 들어, 초당 1 기가비트(1 Gb/s), 10 Gb/s, 50 Gb/s 등)에서 데이터를 전송할 수 있다. 네트워크 인터페이스들 중 일부는 클라우드 내에 위치할 수 있다. 복수의 스토리지 시스템들의 연결은 저-비용의 분산형 그리고 스케일링가능한 서버 시스템을 제공한다. 개시되는 스토리지 시스템들 및/또는 서버 시스템들 중 다수는 서로 통신할 수 있으며, 네트워크(혹은 클라우드) 내에 포함될 수 있다.
- [0065] 솔리드-스테이트 비휘발성 메모리들 각각은 예를 들어, NAND 플래시 메모리 및/또는 다른 솔리드-스테이트 메모리를 포함할 수 있다. 비휘발성 메모리들 각각은 솔리드-스테이트 메모리 및/또는 회전 자기 매체들을 포함할 수 있다. 비휘발성 메모리들 각각은 SSD 및/또는 HDD를 포함할 수 있다.
- [0066] 서버 시스템의 아키텍처는 DRAM들을 캐시들로서 제공한다. DRAM들은 각각의 SoC들에 대해 L3, L4, 및/또는 가장 높은 레벨의 캐시들로서 할당될 수 있고, 고-대역폭 및 큰 스토리지 용량을 가질 수 있다. 적층형 DRAM들은 예를 들어, DDR3 메모리, DDR4 메모리, 저전력 더블 데이터 레이트 타입 4(Low Power Double Data Rate type four)(LPDDR4) 메모리, 와이드-I/O2(wide-I/O2) 메모리, HMC 메모리, 및/또는 다른 적절한 DRAM을 포함할 수 있다. SoC들 각각은 하나 이상의 제어 모듈들을 가질 수 있다. 제어 모듈들은 각각의 링 버스들을 통해 그 대응하는 DRAM들과 통신한다. 링 버스들은 쌍-방향 버스들일 수 있다. 이것은 제어 모듈들과 그 대응하는 DRAM들 간에 고-대역폭 및 최소 레이턴시를 제공한다.
- [0067] 제어 모듈들 각각은, 동일한 혹은 상이한 SoC의 제어 모듈들 내에; DRAM들 중 임의의 DRAM 내에; 솔리드-스테이트 비휘발성 메모리들 중 임의의 메모리 내에; 그리고/또는 비휘발성 메모리들 중 임의의 메모리 내에 저장된 데이터 및/또는 프로그램들에 액세스할 수 있다.
- [0068] SoC들 및/또는 SoC들의 포트들은 매체 액세스 제어기(Medium Access Controller, MAC) 어드레스들을 가질 수 있다. SoC들의 제어 모듈들(혹은 프로세서들)은 각각의 프로세서 클러스터 어드레스(processor cluster address)들을 가질 수 있다. 제어 모듈들 각각은 동일한 SoC 내에 있는 다른 제어 모듈들 혹은 또 하나의 다른 SoC 내에 있는 다른 제어 모듈들에 그 대응하는 MAC 어드레스 및 프로세서 클러스터 어드레스를 사용하여 액세스할 수 있다. SoC들의 제어 모듈들 각각은 DRAM들에 액세스할 수 있다. 제 1 SoC의 제어 모듈은, 제 2 SoC의 MAC 어드레스, 그리고 제 2 SoC 내의 제 2 제어 모듈의 프로세서 클러스터 어드레스를 갖는 요청 신호를 보냄으로써, 제 2 SoC에 연결된 DRAM 내에 저장된 데이터 및/또는 프로그램들을 요청할 수 있다.
- [0069] SoC들 및/또는 SoC들 내의 제어 모듈들 각각은 하나 이상의 어드레스 변환 테이블들을 저장할 수 있다. 어드레스 변환 테이블들은, SoC들의 MAC 어드레스들; 제어 모듈들의 프로세서 클러스터 어드레스들; DRAM들, 솔리드-스테이트 비휘발성 메모리들, 및 비휘발성 메모리들 내의 메모리 셀들의 논리적 블록 어드레스들; 그리고/또는 DRAM들, 솔리드-스테이트 비휘발성 메모리들, 및 비휘발성 메모리들 내의 메모리 셀들의 물리적 블록 어드레스들에 대한 변환들을 포함 및/또는 제공할 수 있다.
- [0070] 하나의 예로서, 데이터 및 프로그램들은 솔리드-스테이트 비휘발성 메모리들 및/또는 비휘발성 메모리들 내에 저장될 수 있다. 데이터 및 프로그램들 및/또는 그 일부분들은 네트워크를 통해 SoC들 및 제어 모듈들에 분산될 수 있다. 제어 모듈에 의한 실행을 위해 필요한 프로그램들 및/또는 데이터는 제어 모듈이 위치하고 있는 SoC의 DRAM들, 솔리드-스테이트 비휘발성 메모리, 및/또는 비휘발성 메모리 내에 국지적으로 저장될 수 있다. 그 다음에, 제어 모듈은 실행을 위해 필요한 프로그램들 및/또는 데이터에 액세스할 수 있고, 그 프로그램들 및/또는 데이터를 DRAM들, 솔리드-스테이트 비휘발성 메모리, 및/또는 비휘발성 메모리로부터 제어 모듈 내의 캐시들로 전송할 수 있다. SoC들과 네트워크 간의 통신 및/또는 SoC들 간의 통신은 무선 통신을 포함할 수 있다.
- [0071] 다른 예로서, 앞서-설명된 예들은 또한, SoC들을 포함하는 서버 시스템 내에서 구현될 수 있다. SoC들 중 일부는 각각의 서버들 내에 통합될 수 있고, 서버 SoC(server SoC)들로서 지칭될 수 있다. (동반 SoC(companion SoC)들로서 지칭되는) SoC들 중 일부는 제 1 SoC의 서버 내에 통합될 수 있고, 또는 제 1 SoC의 서버로부터 분리되어 있을 수 있다. 서버 SoC들은 각각의: 제어 모듈들(예를 들어, 중앙 프로세싱 모듈들)의 클러스터들; 클

러스터-내 링 버스(intra-cluster ring bus)들, FLC 모듈들, 메모리 제어 모듈들, FLC 링 버스들, 및 하나 이상의 호핑 버스(hopping bus)들을 포함한다. 호핑 버스들은 (i) 서버 SoC들과 동반 SoC들 사이에서 칩-간 버스 멤버(inter-chip bus member)들 및 대응하는 포트들을 통해 연장되고 그리고 (ii) 동반 SoC들을 통해 연장된다. 호핑 버스는 호핑 버스 스톱(hopping bus stop)들, 어댑터(adaptor)들, 혹은 노드(node)들 그리고 하나 이상의 SoC들의 대응하는 포트들까지 연장되는 버스, 그리고 호핑 버스 스톱들, 어댑터들, 혹은 노드들 그리고 하나 이상의 SoC들의 대응하는 포트들로부터 연장되는 버스를 지칭할 수 있다. 호핑 버스는 호핑 버스 스톱들 및/또는 하나 이상의 SoC들을 통해 연장될 수 있다. 호핑 버스 스톱으로의 데이터의 단일 전송 혹은 호핑 버스 스톱으로부터의 데이터의 단일 전송은 단일 홉(hop)으로서 지칭될 수 있다. 전송 디바이스와 수신 디바이스 간에 데이터를 전송할 때 복수의 홉들이 수행될 수 있다. 데이터는 데이터가 목적지에 도달할 때까지 각각의 클럭 사이클에서 버스 스톱들 간을 이동할 수 있다. 본 명세서에서 개시되는 각각의 버스 스톱은 모듈로서 구현될 수 있고, 그리고 클럭 신호에 근거하여 디바이스들 간에 데이터를 전송하기 위한 로직(logic)을 포함할 수 있다. 또한, 본 명세서에서 개시되는 각각의 버스는 데이터의 직렬 및/또는 병렬 전송을 위한 임의의 개수의 채널들을 가질 수 있다.

[0072] 제어 모듈들의 클러스터들 각각은 클러스터-내 링 버스들 중 대응하는 하나를 갖는다. 클러스터-내 링 버스들은 쌍-방향성이고, 클러스터들 각각 내의 제어 모듈들 간의 통신을 제공한다. 클러스터-내 링 버스들은, 클러스터-내 링 버스들을 통해 전송된 데이터 신호들에 대해서 제어 모듈들에 의한 액세스를 위해 링 버스 스톱들을 가질 수 있다. 링 버스 스톱들은 신호 리피터(signal repeater)들 및/또는 액세스 노드들로서 실행될 수 있다. 제어 모듈들은 링 버스 스톱들을 통해 클러스터-내 링 버스들에 연결될 수 있고 그리고 클러스터-내 링 버스들에 액세스할 수 있다. 데이터는, 링 버스 스톱들 중 제 1 링 버스 스톱에서의 제 1 제어 모듈로부터 링 버스 스톱들 중 제 2 링 버스 스톱에서의 제 2 제어 모듈로, 클러스터-내 링 버스들을 통해 전송될 수 있다. 제어 모듈들 각각은 중앙 프로세싱 유닛 혹은 프로세서일 수 있다.

[0073] 메모리 제어 모듈들 각각은 FLC 모듈들 각각에 대한 액세스를 제어할 수 있다. FLC 모듈들은 서버 SoC 상에 적층될 수 있다. FLC 모듈들 각각은 FLC(혹은 DRAM)를 포함하고, 본 명세서에서 개시되는 FLC 모듈들 중 임의의 FLC 모듈로서 구현될 수 있으며 이러한 FLC 모듈과 유사하게 동작할 수 있다. 메모리 제어 모듈들은 FLC 링 버스들 상의 각각의 링 버스 스톱들에서 FLC 링 버스들에 액세스할 수 있고, 링 버스 스톱들과 FLC 모듈들 간에 데이터를 전송할 수 있다. 대안적으로, FLC 모듈들은 각각의 링 버스 스톱들에서 FLC 링 버스들에 직접적으로 액세스할 수 있다. 메모리 제어 모듈들 각각은, FLC 모듈들 각각에 대해 그리고/또는 링 버스들 및/또는 호핑 버스들의 버스 스톱들에 대해 메모리 클럭 신호(memory clock signal)들을 발생시키는 메모리 클럭들을 포함할 수 있다. 버스 스톱들은 메모리 클럭 신호들을 링 버스들 및/또는 호핑 버스들을 통해 간접적으로 수신할 수 있거나, 혹은 메모리 제어 모듈들로부터 직접적으로 수신할 수 있다. 이러한 메모리 클럭 신호에 근거하여 데이터는 버스 스톱들을 통해 순환(cycle)될 수 있다.

[0074] FLC 링 버스들은 쌍-방향성 버스들일 수 있고 두 가지 타입의 링 버스 스톱들(S_{RB} 및 S_{RH})을 갖는다. 링 버스 스톱들 각각은 신호 리피터로서 실행될 수 있고 그리고/또는 액세스 노드로서 실행될 수 있다. 링 버스 스톱들(S_{RB})은 호핑 버스들과는 다른 디바이스들에 연결된다. 이러한 디바이스들은, 클러스터-간 링 버스0; FLC 모듈 및/또는 메모리 제어 모듈들; 그리고 그래픽 프로세싱 모듈들을 포함할 수 있다. 클러스터-간 링 버스는 (i) 클러스터들 간의 연결들, 그리고 (ii) 인터섹션 링 버스 스톱(intersection ring bus stop)들 간의 연결들을 제공한다. 인터섹션 링 버스 스톱들은 클러스터-간 링 버스에 대한 액세스를 제공하고 클러스터-간 링 버스를 (i) 클러스터들과 (ii) 링 버스 스톱들 사이에서 연장되는 링 버스 연장들에 연결할 수 있다. 링 버스 스톱들은 FLC 링 버스들 상에 있다. 클러스터-간 링 버스 및 인터섹션 링 버스 스톱들은, (iii) 제 2 FLC 링 버스의 링 버스 스톱과 제 1 클러스터 간의 연결들 및 (iv) 제 1 FLC 링 버스의 링 버스 스톱과 제 2 클러스터 간의 연결들을 제공한다. 이것은 제어 모듈들이 제 2 FLC 모듈의 FLC에 액세스할 수 있게 하고, 그리고 제어 모듈들이 제 1 FLC 모듈의 FLC에 액세스할 수 있게 한다.

[0075] 클러스터-간 링 버스는 칩-내 트레이스(intra-chip trace)들 및 칩-간 트레이스(inter-chip trace)들을 포함할 수 있다. 칩-내 트레이스들은 서버 SoC들 내부에서 연장되고, 그리고 (i) 링 버스 스톱들 중 하나와 (ii) 포트들 중 하나 사이에서 연장된다. 칩-간 트레이스들은 서버 SoC들 외부에서 연장되고, 그리고 포트들의 각각의 쌍(pair)들 사이에서 연장된다.

[0076] 서버 SoC들 각각의 링 버스 스톱들(S_{RH})은 호핑 버스들 및 FLC 링 버스들 중 대응하는 것들에 연결된다. 호핑 버스들 각각은 복수의 호핑 버스 스톱들(S_{HB})을 가지며, 여기서 복수의 호핑 버스 스톱들(S_{HB})은 호핑 버스들 중 대

응하는 것에 대한 각각의 인터페이스 액세스를 제공한다. 호핑 버스 스톱들(S_{HB})은 신호 리피터들로서 실행될 수 있고 그리고/또는 액세스 노드들로서 실행될 수 있다.

[0077] 제 1 호핑 버스, 링 버스 스톱, 및 제 1 호핑 버스 스톱들은, (i) FLC 링 버스와 (ii) 서버 SoC 내의 액정 디스플레이(Liquid Crystal Display, LCD) 인터페이스 및 동반 SoC들의 인터페이스들 간의 연결들을 제공한다. LCD 인터페이스는 디스플레이에 연결될 수 있고 GPM을 통해 제어될 수 있다. 동반 SoC의 인터페이스들은 SAS(Serial Attached Small Computer System Interface) 인터페이스 및 PCIe 인터페이스를 포함한다. 동반 SoC의 인터페이스들은 이미지 프로세서(Image Processor, IP) 인터페이스들이다.

[0078] 이러한 인터페이스들은 각각의 포트들에 연결되고, 포트들은 주변 디바이스들과 같은 디바이스들에 연결될 수 있다. SAS 인터페이스 및 PCIe 인터페이스는 포트들을 통해 SAS 호환가능 디바이스 및 PCIe 호환가능 디바이스에 각각 연결될 수 있다. 하나의 예로서, 스토리지 드라이브가 포트에 연결될 수 있다. 스토리지 드라이브는 하드 디스크 드라이브, 솔리드-스테이트 드라이브, 혹은 하이브리드 드라이브일 수 있다. 포트들은 이미지 프로세서 디바이스들에 연결될 수 있다. 이미지 프로세서 디바이스들의 예들은 위에 개시되어 있다. 제 4 SoC는(데이지 체인 멤버(daisy chain member)로서 지칭되기도 하는) 칩-간 버스 멤버를 통해 제 3 SoC에 데이지 체인(daisy chain)으로 연결될 수 있다. 칩-간 버스 멤버는 제 1 호핑 버스의 멤버이다. 추가적인 SoC들이 포트들 통해 제 4 SoC에 데이지 체인으로 연결될 수 있고, 여기서 포트는 제 1 호핑 버스에 연결된다. 서버 SoC, 제어 모듈들, 및 FLC 모듈은 FLC 링 버스, 제 1 호핑 버스, 및/또는 제 3 SoC를 통해 제 4 SoC와 통신할 수 있다. 하나의 예로서, SoC들은 사우스브리지 칩(southbridge chip)들일 수 있고, 그리고 (i) 서버 SoC와 (ii) 포트들에 연결된 주변 디바이스들 간의 인터럽트(interrupt)들의 전달 및 통신을 제어할 수 있다.

[0079] 제 2 호핑 버스는, 링 버스 스톱 및 제 2 호핑 버스 스톱들을 통해, (i) FLC 링 버스와 (ii) 서버 SoC 내의 인터페이스들 간의 연결들을 제공한다. 서버 SoC 내의 인터페이스들은 이더넷 인터페이스, 하나 이상의 PCIe 인터페이스들, 및 하이브리드(혹은 결합형) 인터페이스를 포함할 수 있다. 이더넷 인터페이스는 10GE 인터페이스일 수 있고 제 1 이더넷 버스를 통해 네트워크에 연결된다. 이더넷 인터페이스는 제 1 이더넷 버스, 네트워크, 및 제 2 이더넷 버스를 통해 제 2 SoC와 통신할 수 있다. 네트워크는 이더넷 네트워크, 클라우드 네트워크, 및/또는 다른 이더넷 호환가능 네트워크일 수 있다. 하나 이상의 PCIe 인터페이스들은 예를 들어, 3세대 PCIe 인터페이스(third generation PCIe interface)(PCIe3) 및 미니 PCIe 인터페이스(mini PCIe interface)(mPCIe)를 포함할 수 있다. PCIe 인터페이스들은 솔리드-스테이트 드라이브들에 연결될 수 있다. 하이브리드 인터페이스는, SATA 및/또는 PCIe 프로토콜들에 따라 데이터를 SATA 호환가능 디바이스들 및/또는 PCIe 호환가능 디바이스들에 전송하고 이들 디바이스들로부터의 데이터를 전송하기 위한 SATA 및 PCIe 호환가능형일 수 있다. 하나의 예로서, PCIe 인터페이스는 솔리드-스테이트 드라이브 혹은 하이브리드 드라이브와 같은 스토리지 드라이브에 연결될 수 있다. 이러한 인터페이스들은 서버 SoC 외부에 있는 디바이스들로의 연결을 위한 각각의 포트들을 갖는다.

[0080] 제 3 호핑 버스는 링 버스 스톱을 통해 링 버스에 연결될 수 있고, 호핑 버스 스톱을 통해 LCD 인터페이스 및 포트에 연결될 수 있다. LCD 인터페이스는 디스플레이에 연결될 수 있고, GPM을 통해 제어될 수 있다. 포트는 하나 이상의 동반 SoC들에 연결될 수 있다. 제 4 호핑 버스는 (i) 링 버스 스톱을 통해 링 버스에 연결될 수 있고, 그리고 (ii) 호핑 버스 스톱들을 통해 인터페이스들에 연결될 수 있다. 이러한 인터페이스들은 이더넷, PCIe 및 하이브리드 인터페이스들일 수 있다. 인터페이스들은 각각의 포트들을 갖는다.

[0081] 서버 SoC들 및/또는 다른 서버 SoC들은 클러스터-간 링 버스를 통해 서로 통신할 수 있다. 서버 SoC들 및/또는 다른 서버 SoC들은 각각의 이더넷 인터페이스들 및 네트워크를 통해 서로 통신할 수 있다.

[0082] 동반 SoC들은 각각의 제어 모듈들을 포함할 수 있다. 제어 모듈들은 호핑 버스 스톱들을 통해 인터페이스들에 액세스할 수 있고 그리고/또는 이들 인터페이스들에 대한 액세스를 제어할 수 있다. 일 실시예에서, 제어 모듈들은 포함되지 않는다. 제어 모듈들은 호핑 버스 스톱들 중 대응하는 것들 및/또는 인터페이스들 중 대응하는 것들에 연결될 수 있고 이들과 통신할 수 있다.

[0083] 다른 예로서, 앞서-설명된 예들은 또한, 모바일 디바이스의 회로 내에서 구현될 수 있다. 모바일 디바이스는 컴퓨터, 셀룰러 폰, 또는 다른 무선 네트워크 디바이스일 수 있다. 회로는 SoC들을 포함한다. SoC는 모바일 SoC로서 지칭될 수 있다. SoC는 동반 SoC로서 지칭될 수 있다. 모바일 SoC는, 제어 모듈들의 클러스터들; 클러스터-내 링 버스, FLC 모듈, 메모리 제어 모듈, FLC 링 버스, 및 하나 이상의 호핑 버스들을 포함한다. 호핑 버스는 (i) 모바일 SoC와 동반 SoC 사이에서 칩-간 버스 멤버들 및 대응하는 포트들을 통해 연장되고 그리고 (ii) 동

반 SoC를 통해 연장된다.

- [0084] 클러스터-내 링 버스는 쌍-방향성이고, 제어 모듈들 간의 통신을 제공한다. 클러스터-내 링 버스는, 클러스터-내 링 버스를 통해 전송된 데이터 신호들에 대해서 제어 모듈들에 의한 액세스를 위해 링 버스 스톱들을 가질 수 있다. 링 버스 스톱들은 신호 리피터들 및/또는 액세스 노드들로서 실행될 수 있다. 제어 모듈들은 링 버스 스톱들을 통해 클러스터-내 링 버스에 연결될 수 있고 그리고 클러스터-내 링 버스에 액세스할 수 있다. 데이터는, 링 버스 스톱들 중 제 1 링 버스 스톱에서의 제 1 제어 모듈로부터 링 버스 스톱들 중 제 2 링 버스 스톱에서의 제 2 제어 모듈로, 클러스터-내 링 버스를 통해 전송될 수 있다. 데이터는 데이터가 목적지에 도달할 때까지 각각의 클럭 사이클에서 버스 스톱들 간을 이동할 수 있다. 제어 모듈들 각각은 중앙 프로세싱 유닛 혹은 프로세서일 수 있다.
- [0085] 메모리 제어 모듈은 FLC 모듈에 대한 액세스를 제어할 수 있다. 일 실시예에서, 메모리 제어 모듈은 포함되지 않는다. FLC 모듈은 모바일 SoC 상에 적층될 수 있다. FLC 모듈은 FLC 혹은 DRAM일 수 있고, 본 명세서에서 개시되는 FLC 모듈들 중 임의의 FLC 모듈로서 구현될 수 있으며 이러한 FLC 모듈과 유사하게 동작할 수 있다. 메모리 제어 모듈은 FLC 링 버스 상의 각각의 링 버스 스톱에서 FLC 링 버스에 액세스할 수 있고, 링 버스 스톱과 FLC 모듈 간에 데이터를 전송할 수 있다. 대안적으로, FLC 모듈은 각각의 링 버스 스톱에서 FLC 링 버스에 직접적으로 액세스할 수 있다. 메모리 제어 모듈은, FLC 모듈에 대해, 그리고 링 버스 및/또는 호핑 버스들의 버스 스톱들에 대해 메모리 클럭 신호를 발생시키는 메모리 클럭을 포함할 수 있다. 버스 스톱들은 메모리 클럭 신호를 링 버스 및/또는 호핑 버스들을 통해 간접적으로 수신할 수 있거나, 혹은 메모리 제어 모듈들로부터 직접적으로 수신할 수 있다. 이러한 메모리 클럭 신호에 근거하여 데이터는 버스 스톱들을 통해 순환될 수 있다.
- [0086] FLC 링 버스는 쌍-방향성 버스일 수 있고 두 가지 타입의 링 버스 스톱들(S_{RB} 및 S_{RH})을 갖는다. 링 버스 스톱들 각각은 신호 리피터로서 실행될 수 있고 그리고/또는 액세스 노드로서 실행될 수 있다. 링 버스 스톱들(S_{RB})은 호핑 버스들과는 다른 디바이스들에 연결된다. 이러한 디바이스들은, 클러스터; FLC 모듈 및/또는 메모리 제어 모듈; 그리고 그래픽 프로세싱 모듈을 포함할 수 있다.
- [0087] 모바일 SoC의 링 버스 스톱들(S_{RH})은 호핑 버스들 중 대응하는 것 및 FLC 링 버스에 연결된다. 호핑 버스들 각각은 복수의 호핑 버스 스톱들(S_{HB})을 가지며, 여기서 복수의 호핑 버스 스톱들(S_{HB})은 호핑 버스들 중 대응하는 것에 대한 각각의 인터페이스 액세스를 제공한다. 호핑 버스 스톱들(S_{HB})은 신호 리피터들로서 실행될 수 있고 그리고/또는 액세스 노드들로서 실행될 수 있다.
- [0088] 제 1 호핑 버스, 링 버스 스톱, 및 제 1 호핑 버스 스톱들은, (i) FLC 링 버스와 (ii) 액정 디스플레이(LCD) 인터페이스, 비디오 프로세싱 모듈(Video Processing Module, VPM) 및 동반 SoC의 인터페이스들 간에 연결된다. LCD 인터페이스는 서버 SoC 내에 있고, 디스플레이에 연결될 수 있으며, GPM을 통해 제어될 수 있다. 동반 SoC의 인터페이스들은 셀룰러 인터페이스, 무선 로컬 영역 네트워크(Wireless Local Area Network, WLAN) 인터페이스, 및 이미지 신호 프로세서 인터페이스를 포함한다. 셀룰러 인터페이스는 다른 모바일 및/또는 무선 디바이스들과의 무선 통신을 위한 물리 계층 디바이스를 포함할 수 있다. 물리 계층 디바이스는 롱-텀 에볼루션(Long-Term Evolution, LTE) 표준들 그리고/또는 3세대(3G), 4세대(4G), 및/또는 5세대(5G) 모바일 원격통신 표준들에 따라 동작할 수 있고 그리고/또는 신호들을 전송 및 수신할 수 있다. WLAN 인터페이스는 블루투스(Bluetooth[®]) 와이-파이(Wi-Fi[®]) 및/또는 다른 WLAN 프로토콜들에 따라 동작할 수 있고, 모바일 디바이스의 WLAN 내의 다른 네트워크 디바이스들과 통신할 수 있다. ISP 인터페이스는 스토리지 드라이브와 같은 동반 SoC 외부에 있는 이미지 프로세싱 디바이스(혹은 이미지 신호 프로세싱 디바이스들), 또는 다른 이미지 프로세싱 디바이스에 연결될 수 있다. 이러한 인터페이스들은 각각의 포트들을 통해 동반 SoC 외부에 있는 디바이스들에 연결될 수 있다. ISP 인터페이스는 모바일 디바이스 외부에 있는 디바이스들에 연결될 수 있다.
- [0089] 동반 SoC는 칩-간 버스 멤버를 통해 모바일 SoC에 연결될 수 있다. 칩-간 버스 멤버는 제 1 호핑 버스의 멤버이다. 추가적인 SoC들이 포트를 통해 동반 SoC에 데이지 체인으로 연결될 수 있고, 여기서 포트는 제 1 호핑 버스에 연결된다. 모바일 SoC, 제어 모듈들, 및 FLC 모듈은 FLC 링 버스 및 제 1 호핑 버스를 통해 동반 SoC와 통신할 수 있다.
- [0090] 제 2 호핑 버스는, 링 버스 스톱 및 제 2 호핑 버스 스톱들을 통해, (i) FLC 링 버스와 (ii) 모바일 SoC 내의 인터페이스들 간의 연결들을 제공한다. 모바일 SoC 내의 인터페이스들은 이더넷 인터페이스, 하나 이상의 PCIe 인터페이스들, 및 하이브리드(혹은 결합형) 인터페이스를 포함할 수 있다. 이더넷 인터페이스는 10GE 인터페이

스일 수 있고 포트를 통해 이더넷 네트워크에 연결된다. 하나 이상의 PCIe 인터페이스들은 예를 들어, 3세대 PCIe 인터페이스(PCIe3) 및 미니 PCIe 인터페이스(mPCIe)를 포함할 수 있다. PCIe 인터페이스들은 솔리드-스테이트 드라이브들에 연결될 수 있다. 하이브리드 인터페이스는, SATA 및/또는 PCIe 프로토콜들에 따라 데이터를 SATA 호환가능 디바이스들 및/또는 PCIe 호환가능 디바이스들에 전송하고 이들 디바이스들로부터의 데이터를 전송하기 위한 SATA 및 PCIe 호환가능형일 수 있다. 하나의 예로서, PCIe 인터페이스는 포트를 통해 스토리지 드라이브에 연결될 수 있다. 스토리지 드라이브는 솔리드-스테이트 드라이브 혹은 하이브리드 드라이브일 수 있다. 이러한 인터페이스들은 모바일 SoC 외부에 있는 디바이스들로의 연결을 위한 각각의 포트들을 갖는다.

[0091] 동반 SoC는 제어 모듈을 포함할 수 있다. 제어 모듈은 호핑 버스 스톱들을 통해 VPM 및 인터페이스들에 액세스할 수 있고 그리고/또는 VPM 및 인터페이스들에 대한 액세스를 제어할 수 있다. 일 실시예에서, 제어 모듈은 포함되지 않는다. 제어 모듈은 호핑 버스 스톱들, VPM(1198) 및/또는 인터페이스들에 연결될 수 있고 이들과 통신할 수 있다.

[0092] 본 개시내용에서 설명되는 무선 통신들은 전체적으로 혹은 부분적으로, IEEE 표준 802.11-2012, IEEE 표준 802.16-2009, IEEE 표준 802.20-2008, 및/또는 블루투스 코어 사양(Bluetooth Core Specification) v4.0에 따라 수행될 수 있다. 다양한 구현예들에서, 블루투스 코어 사양 v4.0은, 블루투스 코어 사양 부록(Bluetooth Core Specification Addendums) 2, 3 혹은 4 중 하나 이상의 것에 의해 수정될 수 있다. 다양한 구현예들에서, IEEE 802.11-2012는, 입안된 IEEE 표준 802.11ac, 입안된 IEEE 표준 802.11ad, 및/또는 입안된 IEEE 표준 802.11ah에 의해 보충될 수 있다.

[0093] 다양한 칩들, 모듈들, 신호들, 요소들, 및/또는 컴포넌트들을 설명하기 위해 본 명세서에서 "제1", "제2", "제3" 등의 용어들이 사용될 수 있지만, 이들 아이템들이 이러한 용어들에 의해 한정되는는 안 된다. 이러한 용어들은 하나의 아이템을 또 하나의 다른 아이템과 구분하기 위해 단지 사용될 수 있다. "제1", "제2", 및 다른 수치적 용어들과 같은 그러한 용어들이 본 명세서에서 사용되는 경우, 이것은 문맥상 명확히 표시되지 않는다면 임의의 시퀀스 혹은 순서를 암시하는 것이 아니다. 따라서, 아래에서 논의되는 제 1 아이템은 예시적 사례들에 관한 가르침으로부터 벗어남이 없이 제 2 아이템으로 지칭될 수 있다.

[0094] 또한, 컴포넌트들 간의 물리적 관계를 설명하기 위해 다양한 용어들이 사용된다. 제 1 요소가 제 2 요소에 "연결", "관련", 혹은 "결합"된 것으로서 기재되는 경우, 제 1 요소는 제 2 요소에 직접적으로 결합, 관련, 배치, 적용, 혹은 결합될 수 있거나, 또는 중간에 개입되는 요소들이 존재할 수 있다. 대조적으로, 임의의 요소가 또 하나의 다른 요소에 "직접적으로 연결", "직접적으로 관련", 혹은 "직접적으로 결합"된 것으로서 기재되는 경우, 중간에 개입되는 요소들은 없을 수 있다. 제 1 요소가 제 2 요소에 "연결", "관련", 혹은 "결합"된다는 기재는 제 1 요소가 제 2 요소에 "직접적으로 결합", "직접적으로 관련", 혹은 "직접적으로 결합"될 수 있음을 암시한다. 요소들 간의 관계를 설명하기 위해 사용되는 다른 용어들도 유사한 방식으로 해석되어야만 한다(예를 들어, "사이에" 대 "사이에 직접적으로", "인접하는" 대 "직접적으로 인접하는" 등).

[0095] 앞서의 설명은 본질적으로 단지 예시적인 것이며 본 개시내용과 그 응용 혹은 용도를 제한할 의도를 전혀 갖고 있지 않다. 본 개시내용의 광범위한 가르침은 다양한 형태로 구현될 수 있다. 따라서, 본 개시내용이 특정 예들을 포함하고 있지만, 도면, 명세서 및 다음의 청구범위를 연구함으로써 다른 수정들이 가능성이 명백해지기 때문에, 본 개시내용의 진정한 범위가 이러한 예들로부터 한정되는 것은 아니다. 본 명세서에서 사용되는 바와 같은, 문구 "A, B 및 C 중 적어도 하나"의 의미는, 비-배타적 논리 연산자인 OR를 사용하는 논리연산 (A OR B OR C)를 의미하는 것으로 해석되어야 하며, "적어도 하나의 A, 적어도 하나의 B, 그리고 적어도 하나의 C"를 의미하는 것으로 해석되는는 안 된다. 본 발명의 방법에서의 하나 이상의 단계들은 본 개시내용의 원리를 변경시킴 없이 서로 다른 순서로(혹은 동시에) 실행될 수 있음을 이해해야 한다.

[0096] 아래의 정의들을 포함하는 본 명세서에서, 용어 "모듈" 혹은 용어 "제어기"는 용어 "회로"로 대체될 수 있다. 용어 "모듈" 및 용어 "제어기"는, 예컨대 시스템-온-칩에서와 같은, 애플리케이션 특정 집적 회로(Application Specific Integrated Circuit, ASIC); 디지털, 아날로그, 혹은 아날로그/디지털 혼합형 이산 회로; 디지털, 아날로그, 혹은 아날로그/디지털 혼합형 집적 회로; 조합형 로직 회로; 현장 프로그래밍가능 게이트 어레이(Field Programmable Gate Array, FPGA); 코드를 실행하는 (공유된 혹은 전용의 혹은 그룹) 프로세서 회로; 프로세서 회로에 의해 실행되는 코드를 저장하는 (공유된 혹은 전용의 혹은 그룹) 메모리 회로; 설명된 기능을 제공하는 다른 적절한 하드웨어 컴포넌트들; 또는 앞서의 것들의 일부 혹은 모두의 임의의 조합을 지칭할 수 있거나, 이들의 일부일 수 있거나, 혹은 이들을 포함할 수 있다.

[0097] 모듈 혹은 제어기는 하나 이상의 인터페이스 회로들을 포함할 수 있다. 일부 예들에서, 인터페이스 회로들은 로

컬 영역 네트워크(Local Area Network, LAN), 인터넷, 와이드 영역 네트워크(Wide Area Network, WAN), 혹은 이들의 조합들에 연결되는 유선 또는 무선 인터페이스들을 포함할 수 있다. 본 개시내용의 임의의 주어진 모듈 혹은 제어기의 기능은 인터페이스 회로들을 통해 연결되는 복수의 모듈들 및/또는 제어기들 간에 분산될 수 있다. 예를 들어, 복수의 모듈들 및/또는 제어기들은 로드 발란싱(load balancing)을 허용할 수 있다. 다른 예에서, 서버(이것은 또한 원격 혹은 클라우드로서 알려져 있음) 모듈 또는 (원격 혹은 클라우드) 제어기는 클라이언트 모듈 및/또는 클라이언트 제어기를 위한 일부 기능을 달성할 수 있다.

[0098] 앞에서 사용된 바와 같은 용어 "코드"는 소프트웨어, 펌웨어 및/또는 마이크로코드(microcode)를 포함할 수 있고, 프로그램들, 루틴들(routines), 기능들(functions), 클래스들(classes), 데이터 구조들 및/또는 오브젝트들(objects)을 지칭할 수 있다. 용어 "공유된 프로세서 회로"는 복수의 모듈들 및/또는 제어기들로부터의 일부 혹은 모든 코드를 실행하는 단일 프로세서 회로를 포괄한다. 용어 "그룹 프로세서 회로"는 하나 이상의 모듈들 및/또는 제어기들로부터의 일부 혹은 모든 코드를 추가적인 프로세서 회로들과 함께 실행하는 프로세서 회로를 포괄한다. 복수의 프로세서 회로들로 지칭되는 것은, 개별 다이들 상의 복수의 프로세서 회로들, 단일 다이 상의 복수의 프로세서 회로들, 단일 프로세서 회로의 복수의 코어들(cores), 단일 프로세서 회로의 복수의 스레드들(threads), 혹은 앞서의 것들의 임의의 조합을 포괄한다. 용어 "공유된 메모리 회로"는 복수의 모듈들 및/또는 제어기들로부터의 일부 혹은 모든 코드를 저장하는 단일 메모리 회로를 포괄한다. 용어 "그룹 메모리 회로"는 하나 이상의 모듈들 및/또는 제어기들로부터의 일부 혹은 모든 코드를 추가적인 메모리들과 함께 저장하는 메모리 회로를 포괄한다.

[0099] 용어 "메모리 회로"는 용어 "컴퓨터-판독가능 매체"의 서브세트이다. 본 명세서에서 사용되는 바와 같은 용어 "컴퓨터-판독가능 매체"는 임의의 매체를 통해서 (예컨대, 반송파를 통해) 전파되는 일시적인 전기적 혹은 전자적 신호들을 포괄하지 않으며, 따라서, 용어 "컴퓨터-판독가능 매체"는 형태가 있는 유형의 일시적이지 않은 것으로 고려될 수 있다. 비-일시적인 유형의 컴퓨터-판독가능 매체의 비-한정적 예들은, 비휘발성 메모리 회로들(예컨대, 플래시 메모리 회로, 소거가능하고 프로그래밍가능한 판독-전용 메모리 회로, 혹은 마스크 판독-전용 메모리 회로), 휘발성 메모리 회로들(예컨대, 정적 랜덤 액세스 메모리 회로 혹은 동적 랜덤 액세스 메모리 회로), 자기적 스토리지 매체들(예컨대, 아날로그 혹은 디지털 자기 테이프 또는 하드 디스크 드라이브), 그리고 광학적 스토리지 매체들(예컨대, CD, DVD, 혹은 블루-레이 디스크)이다.

[0100] 본 명세서에서 설명되는 장치들 및 방법들은, 컴퓨터 프로그램들로 구현되는 하나 이상의 특정 기능들을 실행하도록 범용 컴퓨터를 구성함으로써 생성되는 특수 목적 컴퓨터에 의해 부분적으로 혹은 전체적으로 구현될 수 있다. 앞서 설명된 기능 블록들 및 흐름도의 요소들은 소프트웨어 사양들로서의 역할을 하며, 이것은 숙련된 기술자 혹은 프로그래머의 일상적 작업에 의해 컴퓨터 프로그램들로 변환될 수 있다.

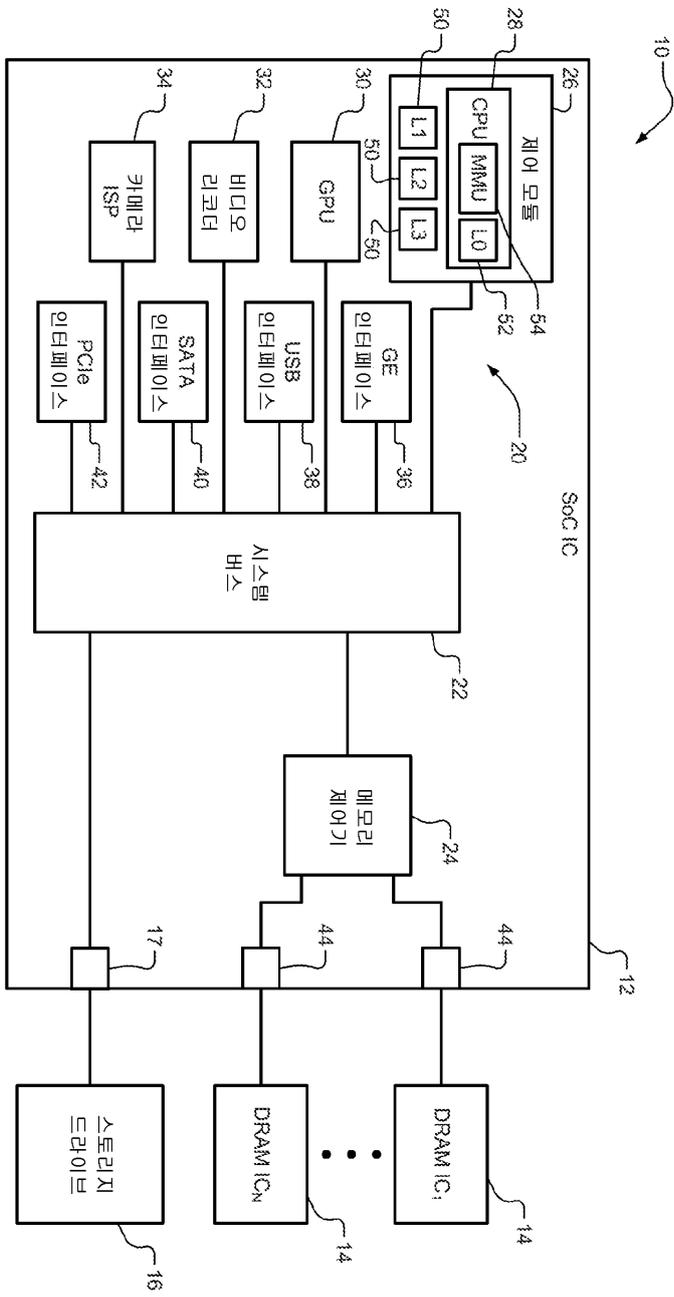
[0101] 컴퓨터 프로그램들은 적어도 하나의 비-일시적인 유형의 컴퓨터-판독가능 매체 상에 저장되는 프로세서-실행가능 명령들을 포함한다. 컴퓨터 프로그램들은 또한 저장된 데이터를 포함할 수 있거나 저장된 데이터에 의존할 수 있다. 컴퓨터 프로그램들은, 특수 목적 컴퓨터의 하드웨어와 상호작용하는 기본 입력/출력 시스템(Basic Input/Output System, BIOS), 특수 목적 컴퓨터의 특정 디바이스들과 상호작용하는 디바이스 드라이버들, 하나 이상의 오퍼레이팅 시스템들, 사용자 애플리케이션들, 백그라운드 서비스들, 백그라운드 애플리케이션들, 등을 포괄할 수 있다.

[0102] 컴퓨터 프로그램들은, (i) HTML(HyperText Markup Language) 혹은 XML(eXtensible Markup Language)와 같은 파싱(parsing)될 서술적 텍스트(descriptive text), (ii) 어셈블리 코드(assembly code), (iii) 컴파일러에 의해 소스 코드로부터 발생하는 오브젝트 코드, (iv) 인터프리터(interpreter)에 의한 실행을 위한 소스 코드, (v) 적시 컴파일러(just-in-time compiler)에 의한 컴파일 및 실행을 위한 소스 코드 등을 포함할 수 있다. 단지 예시적인 것으로서, 소스 코드는, C, C++, C#, 오브젝티브 C(Objective C), 하스켈(Haskell), 고(Go), SQL, R, 리스프(Lisp), 자바(Java[®]), 포트란(Fortran), 펄(Perl), 파스칼(Pascal), 컬(Curl), 오페르(OCaml), 자바스크립트(Javascript[®]), HTML5, 에이다(Ada), ASP(Active Server Pages), PHP, 스칼라(Scala), 에펠(Eiffel), 스몰토크(Smalltalk), 얼랭(Erlang), 루비(Ruby), 플래시(Flash[®]), 비주얼 베이직(Visual Basic[®]), 루아(Lua), 및 파이썬(Python[®])을 포함하는 언어들로부터의 선택스(syntax)를 사용하여 기입될 수 있다.

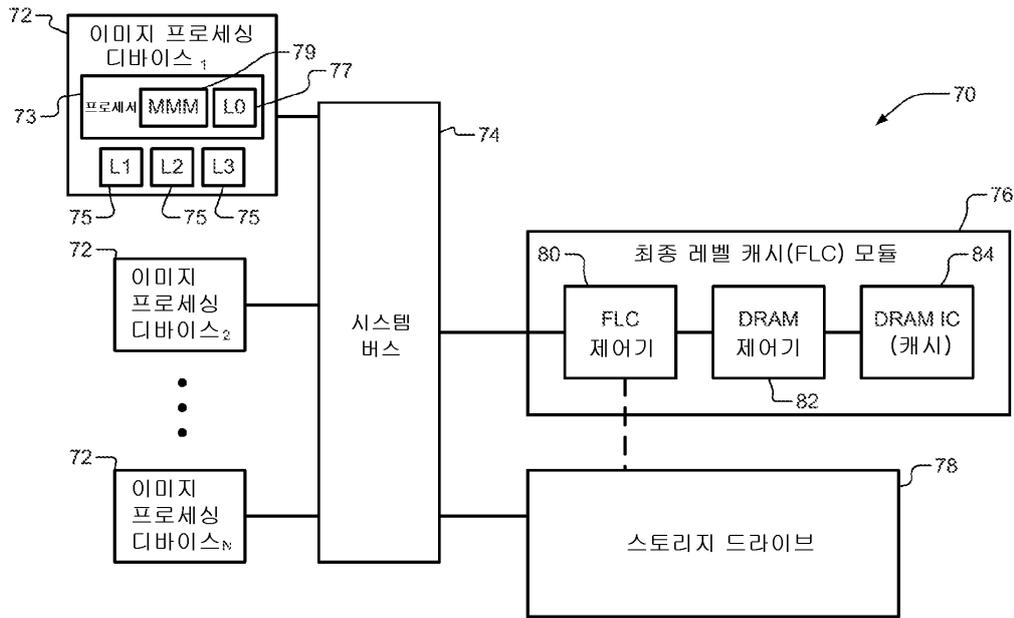
[0103] 청구항들에 기재되는 요소들 그 어떤 것도, 만약 임의의 요소가 어구 "~하기 위한 수단"을 사용하여 명시적으로 기재되지 않는다면 또는 방법 청구항의 경우에는 어구들 "~을 위한 동작" 혹은 "~을 위한 단계"를 사용하여 명시적으로 기재되지 않는다면, 35 U.S.C. § 112(f)의 의미 내에서 기능적 요소가 되도록 의도된 것이 아니다.

도면

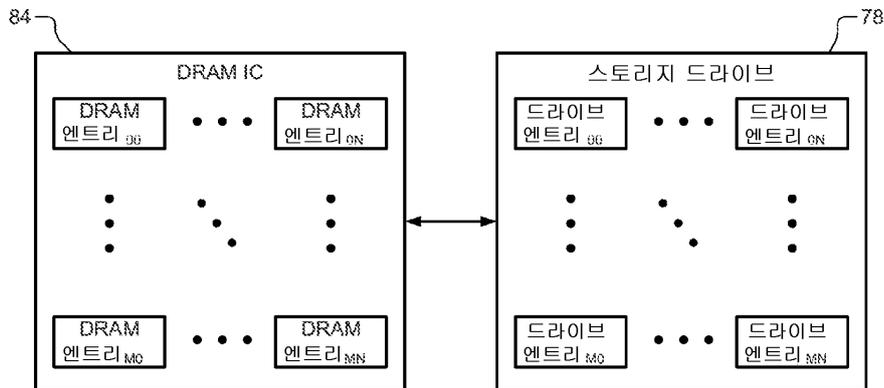
도면1



도면2



도면3



도면4

