



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0089321
(43) 공개일자 2011년08월05일

- (51) Int. Cl.
G06F 13/42 (2006.01) G06F 12/00 (2006.01)
- (21) 출원번호 10-2011-7012261
(22) 출원일자(국제출원일자) 2009년10월27일
심사청구일자 없음
(85) 번역문제출일자 2011년05월27일
(86) 국제출원번호 PCT/US2009/062231
(87) 국제공개번호 WO 2010/053756
국제공개일자 2010년05월14일
(30) 우선권주장
12/605,134 2009년10월23일 미국(US)
61/109,480 2008년10월29일 미국(US)
- (71) 출원인
실리콘 이미지, 인크.
미국 캘리포니아 94085 써니베일 이스트 아쿠에스
에비뉴 1060
(72) 발명자
루머그, 알렌
미국 94025 캘리포니아주 덴로 파크 엘매너 애비
뉴 1009
이, 승-중
미국 95051 캘리포니아주 산타 클라라 사라토가
애비뉴 넘버2113 141
(뒤편에 계속)
(74) 대리인
양영준, 백만기

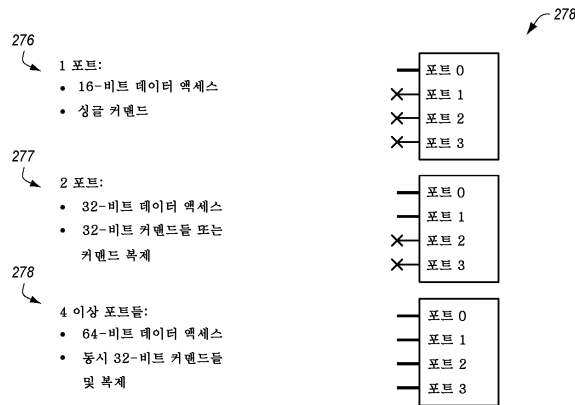
전체 청구항 수 : 총 37 항

(54) 직렬 포트 메모리 통신 레이턴시 및 신뢰성을 향상시키기 위한 방법 및 시스템

(57) 요약

메모리 레이턴시를 감소시키기 위한 방법, 장치 및 시스템이 기술된다. 일 실시예에서, 다수의 시간 간격들 동안 메모리의 포트 또는 포트들의 그룹을 통해 호스트 컴퓨터 시스템과 메모리 간에 데이터가 통신되며, 호스트 컴퓨터는 메모리에 연결된다. 또한, 단일 시간 간격 동안 포트를 통해 또는 포트들의 그룹을 통해 호스트 컴퓨터 시스템과 메모리 간에 데이터와 연관된 커맨드가 통신된다.

대표도 - 도2c



(72) 발명자

이, 형, 록

미국 94085 캘리포니아주 서니베일 레이크쇼어 드라이브 유닛 2215 1259

심, 대윤

미국 95070 캘리포니아주 사라토가 캐논 뷰 드라이브 20780

이, 동윤

미국 95129 캘리포니아주 산 호세 라쎬 애비뉴 5039

김, 성준

미국 95014 캘리포니아주 쿠파티노 이스트 이스테이즈 드라이브 10581

머티, 아누

미국 95014 캘리포니아주 쿠파티노 베트 애비뉴 889

특허청구의 범위

청구항 1

메모리 레이턴시를 감소시키기 위한 방법으로서,

다수의 시간 간격들 동안 메모리의 포트 또는 포트들의 그룹을 통해 호스트 컴퓨터 시스템과 상기 메모리 간에 데이터를 통신하는 단계 - 상기 호스트 컴퓨터는 상기 메모리에 연결되고, 상기 포트는 상기 포트들의 그룹의 멤버임 - ; 및

단일 시간 간격 동안 상기 포트 또는 상기 포트들의 그룹을 통해 상기 호스트 컴퓨터 시스템과 상기 메모리 간에 상기 데이터와 연관된 커맨드를 통신하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 포트들의 그룹 내의 상기 포트의 멤버십은 메모리 오퍼레이션 중의 임의의 시간에 변경 가능한 방법.

청구항 3

제1항에 있어서,

상기 커맨드가 차지하지 않은 상기 포트들의 그룹의 나머지 포트들을 통해 상기 커맨드에 후속적인 커맨드들을 통신하는 단계를 더 포함하는 방법.

청구항 4

제3항에 있어서,

상기 후속적인 커맨드들은 상기 커맨드처럼 상기 단일 시간 간격 내에 통신되는 방법.

청구항 5

제1항에 있어서,

상기 포트들의 그룹의 나머지 포트들을 통해 상기 커맨드의 복제 커맨드를 통신하는 단계를 더 포함하는 방법.

청구항 6

제1항에 있어서,

통신 비트들의 수 및 메모리 레이턴시를 더 감소시키기 위해 기록 데이터와 동일한 통신 프레임 내에 마스크 정보를 포함하지 않고서 상기 데이터의 기록을 금지하는 마스크 기법을 사용하는 단계를 더 포함하는 방법.

청구항 7

제6항에 있어서,

상기 마스크 기법은 데이터 스트림 내에서 변경 가능한 방법.

청구항 8

제7항에 있어서,

상기 마스크 기법은 상기 기록 커맨드와 동일한 통신 프레임 내에 마스크 정보를 포함하지 않고서 데이터의 기록을 금지하도록 변경되는 방법.

청구항 9

제7항에 있어서,

상기 마스킹 기법은 후속적인 유닛 전송들에 대해 자동으로 반복되는 방법.

청구항 10

제7항에 있어서,

상기 마스킹 기법은 단일 유닛 전송 후에 종료되는 방법.

청구항 11

제1항에 있어서,

커맨드들은 기록 데이터 스트림 내에 삽입 가능한 방법.

청구항 12

제1항에 있어서,

상태 정보는 관독 데이터 스트림 내에 삽입 가능한 방법.

청구항 13

메모리 레이턴시를 감소시키는 장치로서,

메모리에 연결된 호스트 컴퓨터 시스템 - 상기 메모리는 다수의 시간 간격들 동안 상기 메모리의 포트 또는 포트들의 그룹을 통해 상기 호스트 컴퓨터 시스템으로부터 데이터를 수신함 -

을 포함하고, 상기 포트는 상기 포트들의 그룹의 멤버이며,

상기 메모리는 단일 시간 간격 동안 상기 포트 또는 상기 포트들의 그룹을 통해 상기 호스트 컴퓨터 시스템으로부터 데이터와 연관된 커맨드를 수신하도록 더 변경되는 장치.

청구항 14

제13항에 있어서,

상기 포트들의 그룹 내의 상기 포트의 멤버십은 메모리 오퍼레이션 중의 임의의 시간에 변경 가능한 장치.

청구항 15

제14항에 있어서,

상기 메모리는 상기 커맨드가 차지하지 않은 상기 포트들의 그룹의 나머지 포트들을 통해 상기 커맨드에 후속적인 커맨드들을 수신하도록 더 변경되는 장치.

청구항 16

제15항에 있어서,

상기 후속적인 커맨드들은 상기 커맨드처럼 상기 단일 시간 간격 내에 통신되는 장치.

청구항 17

제13항에 있어서,

상기 메모리는 상기 포트들의 그룹의 나머지 포트들을 통해 상기 커맨드의 복제 커맨드를 수신하도록 변경되는 장치.

청구항 18

제17항에 있어서,

상기 메모리는 통신 비트들의 수 및 메모리 레이턴시를 더 감소시키기 위해 기록 데이터와 동일한 통신 프레임

내에 마스크 정보를 포함하지 않고서 상기 데이터의 기록을 금지하는 마스크 기법을 사용하도록 더 변경되는 장치.

청구항 19

제18항에 있어서,

상기 마스크 기법은 데이터 스트림 내에서 변경 가능한 장치.

청구항 20

제19항에 있어서,

상기 마스크 기법은 상기 기록 커맨드와 동일한 통신 프레임 내에 마스크 정보를 포함하지 않고서 데이터의 기록을 금지하도록 변경되는 장치.

청구항 21

제20항에 있어서,

상기 마스크 기법은 후속적인 유닛 전송들에 대해 자동으로 반복되는 장치.

청구항 22

제20항에 있어서,

상기 마스크 기법은 단일 유닛 전송 후에 종료되는 장치.

청구항 23

제13항에 있어서,

커맨드들은 기록 데이터 스트림 내에 삽입 가능한 장치.

청구항 24

제13항에 있어서,

상태 정보는 판독 데이터 스트림 내에 삽입 가능한 장치.

청구항 25

메모리 레이턴시를 감소시키는 시스템으로서,

메모리와 연결된 호스트 컴퓨터 시스템

을 포함하고, 상기 메모리는 포트 결합 시스템을 사용해서 상기 메모리 레이턴시를 감소시키고, 상기 포트 결합 시스템은 데이터 및 커맨드들의 통신을 위한 복수의 포트들을 가지며, 상기 복수의 포트들 중 2개 이상의 포트들은 하나 이상의 포트 그룹들로 조합될 수 있고, 상기 포트 결합 시스템은

다수의 시간 간격들 동안 상기 메모리의 포트 또는 포트들의 그룹을 통해 상기 호스트 컴퓨터 시스템과 상기 메모리 간에 데이터를 통신하고 - 상기 포트는 상기 포트들의 그룹의 멤버임 - ,

단일 시간 간격 동안 상기 포트 또는 상기 포트들의 그룹을 통해 상기 호스트 컴퓨터 시스템과 상기 메모리 간에 상기 데이터와 연관된 커맨드를 통신하는

시스템.

청구항 26

제25항에 있어서,

상기 포트들의 그룹 내의 상기 포트의 멤버십은 메모리 오퍼레이션 중의 임의의 시간에 변경 가능한 시스템.

청구항 27

제25항에 있어서,

상기 포트 결합 시스템은 상기 커맨드가 차지하지 않은 상기 포트들의 그룹의 나머지 포트들을 통해 상기 커맨드에 후속적인 커맨드들을 통신하도록 더 변경되는 시스템.

청구항 28

제25항에 있어서,

상기 후속적인 커맨드들은 상기 커맨드처럼 상기 단일 시간 간격 내에 통신되는 시스템.

청구항 29

제25항에 있어서,

상기 포트 결합 시스템은 상기 포트들의 그룹의 나머지 포트들을 통해 상기 커맨드의 복제 커맨드를 통신하도록 더 변경되는 시스템.

청구항 30

제23항에 있어서,

상기 포트 결합 시스템은 통신 비트들의 수 및 메모리 레이턴시를 더 감소시키기 위해 기록 데이터와 동일한 통신 프레임 내에 마스크 정보를 포함하지 않고서 상기 데이터의 기록을 금지하는 마스크 기법을 사용하는 시스템.

청구항 31

제30항에 있어서,

상기 마스크 기법은 데이터 스트림 내에서 변경 가능한 시스템.

청구항 32

제31항에 있어서,

상기 마스크 기법은 상기 기록 커맨드와 동일한 통신 프레임 내에 마스크 정보를 포함하면서 데이터의 기록을 금지하도록 변경되는 시스템.

청구항 33

제32항에 있어서,

상기 마스크 기법은 상기 기록 커맨드와 동일한 통신 프레임 내에 마스크 정보를 포함하지 않고서 데이터의 기록을 금지하도록 더 변경되는 시스템.

청구항 34

제33항에 있어서,

상기 마스크 기법은 후속적인 유닛 전송들에 대해 자동으로 반복되는 시스템.

청구항 35

제33항에 있어서,

상기 마스크 기법은 단일 유닛 전송 후에 종료되는 시스템.

청구항 36

제25항에 있어서,

커맨드들은 기록 데이터 스트림 내에 삽입 가능한 시스템.

청구항 37

제25항에 있어서,
상태 정보는 관독 데이터 스트림 내에 삽입 가능한 시스템.

명세서

기술분야

- [0001] 관련 출원
- [0002] 본 출원은 2008년 10월 29일에 출원된 미국 임시 출원 번호 제61/109,480호의 임시 출원이며, 그 우선권이 청구된다.
- [0003] 본 발명의 실시예들은 일반적으로 컴퓨터 메모리 분야에 관한 것으로, 특히, 직렬 포트 메모리 통신 레이턴시 및 신뢰성을 향상시키는 것에 관한 것이다.

배경기술

- [0004] 커맨드들 및 데이터가 개별 포트들을 통해 호스트(예를 들어, 시스템-온-칩, 컴퓨터, 그래픽스 컨트롤러 등) 또는 복수의 호스트들 및 메모리 간에 전송되는, 고속 직렬 인터페이스들을 사용하는 메모리 시스템들에서, 적합한 시스템 오퍼레이션을 보장하기 위해 일부 예러 검출과 조합된 최대 대역폭을 제공하는 것이 바람직하다.
- [0005] 한번에 오직 하나의 비트만이 송신되기 때문에 직렬 링크들은 고유 레이턴시를 갖는다. 또한, 직렬화(serialization) 및 역직렬화(de-serialization) 프로세스는 추가 레이턴시를 야기한다. 포트들을 개별적으로 사용해도 레이턴시를 상당히 향상시키지는 않으며, 정형화된 액세스 방법이 사용되어(예를 들어, 스트라이프드 액세스(striped accesses)에서와 같이, 각각의 포트로부터 상이한 전용 메모리 영역들에 액세스함) 대역폭을 향상시킨다. (동시에 다수의 포트들을 사용해서) 포트 결합을 가능하게 함으로써, 정형화된 액세스 방법을 필요로 하지 않은 채 대역폭을 증가시키면서, 데이터의 수개의 비트들이 한번에 전송되게 함으로써 메모리 레이턴시는 감소될 수 있다.
- [0006] 메모리는 또한 일정량의 데이터 보안을 요구한다. 예를 들어, 직렬 채널에서, 받아들이기 어려운 레이턴시를 야기하는 방법들을 사용하지 않고 검출될 수 없는 에러들이 발생할 수 있다. 결합 포트 상황에서, 커맨드 기간 중에 일부 포트들은 유휴 상태로 남는다. 이러한 비사용 대역폭은 동일한 기간들에서 커맨드들의 복제물들로 채워진다. 이 방법은 임시 복제물들을 사용하는 싱글 포트들로 확장되어, 포트 구성에 특징을 제공한다.
- [0007] 도 1은 EIA 표준 RS-232-C의 종래의 직렬 비트 할당(100)을 도시한다. 본 도면에서, 데이터의 직렬 전송은 차례대로 한번에 관찰해서 값(124) 내에서 상이한 중요도(significance)로 할당함으로써 개별 이진 값들(비트들)(102-118)이 전체 값(124)으로 어셈블되는 RS-232 링크와 유사하다. 예를 들어, 제1 비트가 값(124)의 최상위 비트에 할당된 비트(104)이면, 제2 비트(106)가 이어지는 등, 최하위 비트가 통신된 최종 비트(118)에 의해 채워질 때까지 계속된다. 이러한 경우에 어셈블된 값은 정지 및 개시 비트들(102, 120) 뿐만 아니라 값(124)을 포함하는 프레임(128)이라고 한다. 또한, 프레임(128)의 시작을 찾기 위해 또한 예상될 때 프레임들이 도착하는지를 체크하기 위해 수신기가 사용할 수 있는, 개시 비트(120) 및 정지 비트(122)를 포함하는 프레임링 비트들(126)이라고 하는, 여분의 비트들을 사용해서 프레임(128)이 묘사된다. 다른 기법들에서, 프레임링 비트들(126)은 송신기 및 수신기 간의 데이터 레이트가 약간 상이하거나 변경되더라도 수신기가 신뢰성 있게 개별 비트들을 찾도록 돕는데 사용된다.
- [0008] 개별 직렬 링크들을 통한 메모리로의 통신은 다량의 레이턴시를 야기하며, 하나 보다 많은 호스트에 의한 싱글 메모리에 대한 액세스를 제공하여 메모리 리소스 복잡성을 야기한다. 또한, 메모리는 하나 이상의 포트들을 가질 수 있으며, 각각의 포트는 직렬 송신기 및 직렬 수신기 및 레이턴시 및 대역폭을 향상시키기 위한 연관된 회로를 포함한다. 결합 포트 상황에서, 커맨드 기간 중에 일부 포트들은 유휴 상태로 남는다. 이러한 비사용 대역폭은 동일한 시간 기간들에서 커맨드들의 공간 복제물들로 채워지고, 본 방법은 상이한 시간 기간들에 임시 복제물들을 사용해서 싱글 포트들로 확장되어, 모든 포트 구성들에 특징을 제공한다. 결합 포트 상황에서, 데이터는 다수의 포트들로 송신되지만, 커맨드들은 독립적이어야만 한다. 먼저, 비사용 포트들은 커맨드 복제물들을 포함할 수 있다. 둘째로, 특정 커맨드들은 동시에 발행될 수 있다. 또한, 직렬 통신은 직렬화, 역직렬화, 에러 관리를 위한 데이터의 프레임링, 및 동기화 등의 추가 프로세스들 때문에 병렬 통신을 통해 레

이턴시를 증가시킨다.

발명의 내용

해결하려는 과제

[0009] 따라서, 메모리 레이턴시를 감소시키는 기술들을 도입 및 사용하는 것이 바람직하다.

과제의 해결 수단

[0010] 포트 메모리 통신 레이턴시 및 신뢰성을 향상시키기 위한 방법, 장치 및 시스템이 기술된다.

[0011] 일 실시예에서, 한 방법은 다수의 시간 간격들 동안 메모리의 포트들의 그룹을 통해 호스트 컴퓨터 시스템 및 메모리 간에 데이터를 통신하는 단계 - 호스트 컴퓨터는 메모리에 연결됨 - 와, 단일 시간 간격 동안 포트들의 그룹 중 한 포트를 통해 호스트 컴퓨터 시스템 및 메모리 간에 데이터와 연관된 커맨드를 통신하는 단계를 포함한다.

[0012] 일 실시예에서, 한 장치는 다수의 시간 간격들 동안 메모리의 포트들의 그룹을 통해 호스트 컴퓨터 시스템으로부터 데이터를 수신하는 메모리에 연결된 호스트 컴퓨터 시스템을 포함한다. 메모리는 단일 시간 간격 동안 포트들의 그룹 중 한 포트를 통해 호스트 컴퓨터 시스템으로부터의 데이터와 연관된 커맨드를 수신하도록 더 변경된다.

[0013] 일 실시예에서, 한 방법은 호스트 컴퓨터 시스템 및 메모리 간에 데이터를 통신하는 것을 포함하며, 대역폭 및 레이턴시를 변경하고/변경하거나 전력을 절약하기 위해 그룹 애드 혹은 포트들의 수가 선택될 수 있다.

[0014] 일 실시예에서, 한 장치는 대역폭 및 레이턴시를 변경하고/변경하거나 전력을 절약하기 위해 그룹 애드 혹은 포트들의 수를 선택하기 위해 호스트 컴퓨터 시스템 및 메모리 간에 데이터를 통신하는 것을 포함한다.

[0015] 일 실시예에서, 복수의 호스트들을 메모리에 연결하는 시스템이 기술된다. 본 시스템은 호스트 컴퓨터 시스템들 또는 메모리와 연결된 컴퓨터 시스템 내의 기능들을 포함하고, 메모리는 포트 결합 시스템을 사용해서 메모리 레이턴시를 감소시키며, 포트 결합 시스템은 데이터 및 커맨드들의 통신을 위한 복수의 포트들을 가지며, 복수의 포트들 중 2개 이상의 포트들은 하나 이상의 포트 그룹들로 애드 혹은 조합될 수 있고, 포트 결합 시스템은 다수의 시간 간격들 동안 메모리의 포트들의 그룹을 통해 호스트들 및 메모리 간에 데이터를 통신하고, 단일 시간 간격 동안 포트들의 그룹 중 한 포트를 통해 호스트들 및 메모리 간에 데이터와 연관된 커맨드를 통신한다.

도면의 간단한 설명

[0016] 본 발명의 실시예들은, 첨부 도면들에서, 일례로서 도시된 것으로, 제한적인 의미가 아니다. 도면들에서, 동일 참조 부호들은 동일 엘리먼트들을 나타낸다.

도 1은 RS-232의 종래의 직렬 비트 할당을 도시한다.

도 2a는 싱글 호스트 결합 포트 메모리의 일 실시예를 도시한다.

도 2b는 4-포트 메모리의 싱글 호스트 커넥션의 일 실시예를 도시한다.

도 2c는 싱글 호스트 인터페이스를 위한 포트 결합 선택의 일 실시예를 도시한다.

도 2d는 스마트 모바일 폰의 일 실시예를 도시한다.

도 2e는 직렬 포트 DRAM(SPDRAM)을 사용하는 스마트 폰 아키텍처의 도 2d의 다른 실시예를 도시한다.

도 2f는 멀티-호스트 결합 구성들의 실시예들을 도시한다.

도 2g는 멀티-호스트 결합 포트 메모리의 일 실시예를 도시한다.

도 2h는 최대 16개의 포트들을 위한 포트 결합 컨트롤 레지스터 및 최대 16개의 포트들을 위한 복제 커맨드 체크 레지스터의 실시예들을 도시한다.

도 2i는 결합 디멀티플렉서의 일 실시예를 도시한다.

- 도 2j는 결합 디멀티플렉서 라우팅을 도시한 표들의 실시예들을 도시한다.
- 도 2k는 결합 멀티플렉서의 일 실시예를 도시한다.
- 도 2l은 결합 멀티플렉서 라우팅의 표들의 일 실시예를 도시한다.
- 도 3은 프레임 동기화 프로세스의 일 실시예를 도시한다.
- 도 4는 전력 컨트롤 프로세스의 일 실시예를 도시한다.
- 도 5는 싱글 포트를 사용하는 복제 체크 및 커맨드 해석 프로세스의 일 실시예를 도시한다.
- 도 6은 한 포트 내에서 프레임들을 수신 및 디코딩하는 프로세스의 일 실시예를 도시한다.
- 도 7a는 17-비트 포스트-디코딩 프레임 (포맷)의 일 실시예를 도시한다.
- 도 7b는 커맨드, 상태, 및 데이터 인코딩 프레임의 일 실시예를 도시한다.
- 도 7c는 활성화 बैं크 및 활성화 커맨드들의 일 실시예를 도시한다.
- 도 7d는 기록 마스크 및 기록 커맨드의 일 실시예를 도시한다.
- 도 8a, 도 8b, 및 도 8c는 기록 마스크 모델들의 실시예를 도시한다.
- 도 9는 다수의 포트들을 사용하는 복제 체크 및 커맨드 해석 프로세스의 일 실시예를 도시한다.
- 도 10은 커맨드 복제 모델들의 일 실시예를 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0017] 본 발명의 실시예들은 일반적으로 직렬 포트 메모리 통신 레이턴시 및 신뢰성을 향상시키는 것에 관한 것이다; 그러나, 고속 병렬 등의 다른 스타일들의 인터페이스에도 동일하게 적용될 수 있다.
- [0018] 본 명세서에서 사용된, "메모리"는 컴퓨팅 프로세서 등의 임의의 "호스트" 또는 키보드, 디스플레이, 카메라, 대용량 기억 장치(자기 디스크, 광 디스크, 자기 테이프 등), 네트워크 컨트롤러, 또는 무선 네트워크 등의 주변 장치에 의해 사용되기 위해 미리 저장된 데이터를 검색할 책임이 있는 컴퓨터 시스템(예를 들어, 도 2d, 도 2e)의 컴포넌트와 관련된다. 통상, 메모리는 컴퓨터 시스템에서 데이터를 처리하기 위한 하나 이상의 마이크로-프로세서들에 연결된다. 데이터는, RAM(random access memory), SRAM(static RAM), DRAM(dynamic RAM), FLASH, PROM(programmable read-only memory), EPROM(erasable PROM), EEPROM(electrically erasable PROM), 또는 ROM(read-only memory)의 경우와 같이 선정된 메모리 등의 메모리에 호스트에 의해 저장될 수 있다. 메모리는, PCI 등의 버스를 통해, 또는 중간 메모리 컨트롤러들을 통해, 호스트에 의해 직접 액세스될 수 있다. 메모리에 대한 직렬 액세스는 (도 1에서 제시된 RS-232와 유사하게) 싱글 회로를 통해 전달되는 단일 시퀀스의 전기 신호들 및 메모리와의 의미 있는 커맨드들 및 데이터 간의 변환을 수반한다. 이러한 변환을 실행하는 회로는 "포트"라고 한다.
- [0019] 일 실시예에서, 메모리 레이턴시를 감소시키기 위해, 마스킹 기법이 사용되어서, 프레임의 비트들의 수를 감소시키고 레이턴시를 감소시키면서, 동일한 통신 프레임 내에 마스크 정보를 포함시키지 않고 기록 커맨드들 및 데이터가 기술될 수 있다. 또한, 레이턴시를 감소시키기 위해, 메모리-기반 프로토콜이 제공되어, 더 짧은 프레임 크기들로 레이턴시를 감소시키며, DRAM(Dynamic Random Access Memory) 프로토콜들에 대한 더 큰 확장성을 제공하고, 대역폭 증가에 따른 커맨드 세트 변경을 감소시킨다.
- [0020] 도 2a는 싱글 호스트 결합 포트 메모리(200)의 일 실시예를 도시한다. 결합 포트 메모리(200)의 도시된 실시예는 결합 포트 메모리(200)의 결합 포트 메모리 시스템(204)과 연관된 다수의 बैं크들(예를 들어, 8 बैं크들)을 갖는 코어 메모리(202)(예를 들어, DRAM, 플래시)를 포함한다. 메모리 코어(202)의 बैं크들은, 다수의 포트들, 예를 들어, 4 포트들(206-212)과 통신한다. 4 포트들(206-212)은 모두 함께 작동해서, 가변 대역폭을 갖는 싱글 호스트에 인터페이스를 제공한다. 예를 들어, 동시에 하나의 बैं크로부터는 판독이, 다른 बैं크로부터는 기록이 이루어지는 등, बैं크들은 독립적으로 사용된다. 메모리 코어(202)는 또한 데이터를 판독하기 위한 메모리 판독 버스 및 데이터를 기록하기 위한 메모리 기록 버스를 포함할 수 있다; 그러나, 데이터를 판독 및 기록하기 위해 메모리 코어(202) 내에 싱글 커넥션이 존재할 수도 있다. 또한, 결합 포트 메모리 시스템(204)은, 도 2k 및 도 2l에, 각각 기술된 결합 멀티플렉서(264) 및 결합 디-멀티플렉서(262)를 포함한다.

- [0021] 종래의 기술들에서, 모든 커맨드 및 데이터 비트들은 개별 병렬 와이어들을 통해서 한번에 송신되며, 동시에 도착되어, 코드가 형성된다; 그러나, 속도가 더 빨라짐에 따라, 이러한 개별 와이어들을 통한 데이터는 부정확하게 또는 잘못된 시간에 (예를 들어, 연관된 클록 신호를 사용해서) 샘플링될 수 있다. 고속 샘플링 문제점을 해결하기 위해, 자체-샘플링 직렬 신호(예를 들면, 도 1의 RS-232)가 사용된다; 그러나, 데이터가 임시로 처리되기 때문에 병렬 방법에 비해 레이턴시가 증가된다. 일 실시예에서, 메모리와의 직렬 포트 통신을 위한 통신 레이턴시 및 신뢰성은 다수의 직렬 인터페이스들, 또는 포트들의 그룹화, 및 차례대로의, 또는 일시적인, 또는 포트들 간의 동시의, 또는 공간적인 커맨드들의 복제를 이용한다. 본 기술의 일 실시예는 본 명세서에 설명되며 싱글 호스트 포트 결합 포트 메모리(200)에 의해 사용된다.
- [0022] 일 실시예에서, 송신기(Tx)는 16 비트들의 병렬 데이터를 직렬 비트 스트림으로 변환하고 싱글 비트 스트림을 송신할 수 있으며, 수신기(Rx)는 16 비트 싱글 스트림을 수신한 후 병렬 스트림으로 변환할 수 있다. 이러한 경우에, 로컬 메모리는 대략 동일한 레이트의 32-비트 폭일 수 있다. 본 포트 메모리(200)에서는, 4 포트들(206-212)이 사용되며, 따라서, 4 포트들(206-212)을 통해 각각 64-비트 데이터 스트리밍으로 데이터의 128-비트 이동이 존재한다(예를 들어, 각각, 16 비트들 × 4 포트 = 64 비트들). 128-비트 이동은 칩 내부의 필요 회로에 의해 지원된다.
- [0023] 종래의 기술들과 달리, 각각의 포트(206-212)는 직렬화기/역직렬화기(serializer/de-serializer)를 사용해서 더 빠른 레이트로 데이터 스트림들을 직렬화 및 역직렬화한다. 예를 들어, PLL은 개별 비트들을 샘플링하는데 사용되는 입력 데이터 레이트와 매치하도록 더 빠른 속도로 입력 클록을 증가시키는데 사용될 수 있다. 포트들(206-212)의 데이터 스트림들의 도착 시간이 약간 상이할 수 있더라도, 이러한 데이터 스트림들의 흐름은 더 빠르게 달성된다. 다시 말해서, 각각의 스트림은 동일한 속도로 흐르지만, 훨씬 더 빠른 속도로 흐른다. 또한, 개별 비트들의 타이밍은 완전히 완벽하지 않을 수도 있지만, 비트들의 실제 도착 시간과 상관 없이 비트들을 정렬할 필요는 없다. 따라서, 핀에서 각각의 비트를 동기화하는 대신, 비트들은, 예를 들어, '*'(232-238)로 표시된 바와 같이, 각각의 포트(206-212) 내에서 역직렬화 후에 동기화된다. 또한, 데이터 비트들이 역직렬화 후에 흐르는 속도로, 이러한 '*'(232-238)의 또는 그에 가까운 포트들(206-212)은 고속 외부 메모리 인터페이스(214-228)(예를 들어, 250 피코세컨드) 보다 20배 더 느릴 수 있다(예를 들어, 5 나노세컨드). 일 실시예에서, 포트들(206-212)은 위상 검출, 데이터 비트들의 관리, 데이터 비트들의 샘플링, 및 레인 정렬이 가능하다.
- [0024] 커맨드 인터프리터(248)는 (도 2c, 7c, 7d에 더 기술된) 레인 구성을 기반으로 커맨드들, 부가/밀접하게 관련된 커맨드들을 계속해서 처리한다.
- [0025] 도 2b는 4-포트 메모리(200)의 싱글 호스트 커넥션(270)의 일 실시예를 도시한다. 본 실시예는 호스트(271)에 대한 메모리(200)의 커넥션(270)이다. 호스트(271)는 한번에 1회 판독 및 1회 기록 오퍼레이션을 달성한다. SPMT는 와이드 데이터 통신을 형성하기 위해 포트들을 함께 그룹화하는 것을 정의하며, 결합 그룹의 포트들(206-212)의 수는 동적으로 선택 가능하다. 예를 들어, 싱글 포트가 사용될 수 있으며, 또는 임의의 수의 포트들이 제공된 포트들(206-212)의 수(예를 들어, 2의 거듭제곱)까지 결합될 수 있다. 더 적은 수의 포트들이 사용될 때, 더 적은 수의 핀들이 사용되며 따라서 전력이 덜 사용된다. 더 많은 수의 포트들이 사용될 때, 대역폭이 증가되며, 동일한 양의 데이터를 검색하기 위한 결과 레이턴시가 감소된다. 결합 포트들의 수는 임의의 시간에 변경될 수 있음이 숙고된다.
- [0026] 도 2c는 싱글 호스트 인터페이스를 위한 포트 결합 선택(275)의 일 실시예를 도시한다. 2개 이상의 포트들이 결합될 때(277, 278), 데이터는, 데이터 대역폭을 효율적으로 증가시키면서, 그룹의 모든 포트들에서 전송된다. 그러나, 개별 커맨드는 단지 하나의 포트를 필요로 할 수 있어서, 포트들의 나머지는 미사용 상태로 남겨 두어서, 대역폭 낭비를 방지하며, 메모리 오퍼레이션을 강화하고, 커맨드 대역폭을 절약하며, 밀접하게 관련된 커맨드들 또는 부가 커맨드들의 집합이 제공된다. 이러한 커맨드들은 낭비된 여분의 대역폭을 적합하게 사용하며, 하나의 커맨드가 다른 포트들에서 동시에 발행될 수 있기 전에 또는 후에 밀접하게 관련된/부가 커맨드들이 발행될 수 있다. 예를 들어, 활성화 커맨드(ACT)가 발행될 때, 활성화 बैं크 커맨드(ABNK)는 커맨드를 완료하기 위한 부가물로서 동시에 발행될 수 있다. 마찬가지로, 부가 기록 마스크 커맨드(WMSK)는 기록 커맨드(WR)를 동시에 수반할 수 있다. 모든 커맨드들은 포트 0에서 수용될 수 있지만, 관련 부가 커맨드들은 커맨드 대역폭을 절약하기 위해 다른 포트들에서 수용될 수 있다. 커맨드들 ACT, ABNK, WR 및 WMSK는 도 7b-7d를 참조해서 더 기술된다.
- [0027] 또한, 싱글 옵션은 잘못된 커맨드가 메모리 동작 상태에 오류를 야기하는 것을 방지하기 위해 강화된 에러 검출을 위한 커맨드 복제를 가능케 한다. 이러한 옵션이 가능한 경우, 싱글 포트는 제1 프레임의 커맨드를 이하의

프레임의 복제물과 비교한다. 복제물이 다른 포트에서 동시에 나타나기 때문에, 2개 이상의 포트들이 결합되어 (277, 278) 사용될 때 복제를 위해 추가 대역폭이 사용되지 않는다. 커맨드들이 복제되더라도, 본 실시예의 데이터는 복제되지 않는다. 적어도 4개의 결합 포트들(278)이 존재할 때, 복제 및 부가 커맨드들 모두 동시에 사용될 수 있다.

[0028] 도 2d는 베이스밴드 프로세서(282) 및 애플리케이션 프로세서(281), 개별 휘발성 메모리(예를 들어, DRAM(274), SRAM/DRAM(283)), 및 비휘발성 메모리(예를 들어, NAND(272) 및 NOR 플래시(273)), 및 2개의 프로세서들(281, 282) 간의 통신 채널(269)로 구성된 스마트 모바일 폰 아키텍처(280)의 일 실시예를 도시한다. 메모리들(272-274, 283)은 공유 없이 각각의 연결된 프로세서들에 사적으로 유지되는 데이터 뿐만 아니라 실행 가능 프로그램 코드들을 저장 및 검색하는데 사용된다. 임의의 공유 또는 통신은 통신 채널(269)을 통해 실행된다. 애플리케이션 프로세서(281)는 카메라(201) 및 디스플레이 디바이스(203) 등의 다른 주변 장치들에 연결될 수 있다.

[0029] 도 2e는 SPDRAM(285)을 갖는 스마트 폰 아키텍처(284)의 도 2d의 다른 실시예를 도시한다. 일 실시예에서, 메모리는 베이스밴드 프로세서(282) 및 애플리케이션 프로세서(281) 간에 공유된다. 본 실시예에서, SPDRAM(285)은 베이스밴드 프로세서 및 애플리케이션 프로세서 간의 통신에 사용될 수 있으며, 양 프로세서들을 위한 프로그램 코드들 및 데이터를 저장하고, 아키텍처를 구현하는데 필요한 메모리 컴포넌트들 또는 기술들의 수를 감소시키는데 사용된다. 또한, 전용 통신 채널의 제거를 포함해서, 메모리들 및 프로세서들 간의 커넥션들의 수는 감소된다. 일부 호스트들이 메모리의 파트들에 액세스할 수 있지만 다른 호스트들은 할 수 없는, 세그먼트이션(segmentation)이 제공된다. 이는, 베이스밴드 소프트웨어의 경우와 같이, 안전한 환경들에서 메모리 디바이스 공유가 가능하게 한다. 예를 들어, 애플리케이션 프로세서(281)는 베이스밴드 소프트웨어 이미지를 SPDRAM(285)에 로드할 수 있으며, 이미지가 준비되었음을 베이스밴드 프로세서(282)에 나타낼 수 있다. 베이스밴드 프로세서(282)는 그 후 다른 호스트들에 대한 액세스를 제거한 후 이미지의 타당성을 검사한다. 정확하면, 베이스밴드 프로세서(282)는 계속해서 애플리케이션 프로세서(281)에서 실행되는 소프트웨어로부터 중단(disruption) 없이 해당 이미지로부터 동작한다.

[0030] 도 2f는 멀티-호스트 결합 구성들(286, 287, 288)의 실시예들을 도시한다. 일 실시예에서, 다중 포트 결합은 멀티-호스트 기능과 결합될 수 있다. 예를 들어, 애플리케이션 프로세서(281) 등의 하나의 호스트가 더 많은 대역폭을 요구하면, 인터페이스를 위해 수개의 결합 포트들을 사용할 수 있으며, 다른 호스트들은 계속해서 싱글 포트를 사용할 수 있다. 본 실시예에서, 4-포트 디바이스에서 수개의 호스트들을 결합하는 소수의 조합들(286, 287, 288)이 제공된다. 예를 들어, 조합(286)에서, 호스트들 1 및 2 각각은 2개의 포트들과 인터페이스한다. 조합(287)에서, 호스트1 은 포트 0과 인터페이스하고, 호스트 2는 2개의 포트들 2, 3과 인터페이스한다. 조합(288)에서, 각각의 호스트는 싱글 포트와 인터페이스한다. 예를 들어, 싱글 호스트가 4개의 호스트들을 모두 함께 결합할 수 있는 것과 같이, 호스트-포트 결합 또는 인터페이스의 임의의 조합이 제공될 수 있음이 숙고된다. 정확하게 어떤 포트들이 어떤 호스트들에 할당되는지는 결합 포트 그룹의 길이를 제공하는 레지스터 세팅에 좌우될 수 있다.

[0031] 도 2g는 멀티-호스트 결합 포트 메모리(292)의 일 실시예를 도시한다. 4개의 포트들(290)을 갖는 본 멀티-호스트 결합 포트 메모리(멀티-호스트 메모리 시스템)(292)는 8개의 뱅크들(289)을 갖는 메모리 코어(291)와 통신중이다. 편의상 간결하게, 제한된 수의 포트들(290) 및 뱅크들(289)이 도시된다. 멀티-호스트 메모리 시스템(292)이 도 2a의 싱글-호스트 메모리 시스템(204)과 유사하더라도, 각각의 포트(290)로부터의 데이터는 개별적으로 각각의 뱅크(289)에 유효하다. 본 실시예에서, 뱅크(289)는 데이터 전송에 독립적으로 어드레싱될 수 있는 전체 멀티-호스트 메모리 시스템(292)의 파트로서 정의된다. 또한, 개별 액세스를 제공함으로써, 포트들(290) 중 싱글 포트는 다른 뱅크들에 액세스하는 다른 포트들과 충돌하지 않고 명령의 지속 기간 동안 뱅크들(289) 중 싱글 뱅크와 연관될 수 있다. 결합 멀티플렉서들(293) 및 디멀티플렉서들(294)은 복합적으로 되어서 뱅크들(289) 및 포트들(290)의 다수의 포트 그룹들 간에 데이터를 관리하기 위한 크로스바 스위치의 가능한 실시예를 생성한다.

[0032] 도 2h는 최대 16개의 포트들을 위한 포트 결합 컨트롤 레지스터(295) 및 최대 16개의 포트들을 위한 복제 커맨드 체크 레지스터(296)의 실시예들을 도시한다. 간결하게 편의상, 본 실시예는, 매칭 계수(matching modulus)(예를 들어, 4 포트들을 위한 포트 0, 2 포트들을 위한 포트들 0 또는 2, 또는 싱글 포트를 위한 임의의 포트)를 갖는 이진 배수(예를 들어, 포트들 1, 2, 또는 4)의 연속 포트들에서 결합들이 발생한다고 가정한다. 포트들은 레지스터 세팅을 기반으로 결합 그룹에서 자신의 멤버십을 결정할 수 있다. 16 포트 결합 컨트롤 레지스터(295)가 도시된다. 결합들은 계층적 패턴으로서 기술되고 제공된다. 예를 들어, 비트들이 세팅되지 않으면, 모든 포트들은 독립적으로 작동한다. 2 포트 디바이스의 경우, 단지 비트 0이 사용되며, 4 포트 디바이스의 경

우, 비트들 0-4가 4 포트들을 위한 2 포트 결합들의 나머지를 추가해서 모든 포트들을 결합함으로써 가능한 결합들을 기술한다. 8 포트 디바이스의 경우, 비트들 8-11에서 2 포트 결합들의 나머지, 비트들 13 및 14에서 4 포트 결합들, 비트 15에서 모든 포트들. 이 패턴은 무한히 계속될 수 있다.

[0033] 또한, 포트들은 결합 그룹에 속하지 않을 수도 있는데, 이러한 경우에, 스스로 동작할 수도 있다. 포트들은 결합 그룹의 파트가 아닐 수 있으며 개별적으로 동작할 수 있거나, 또는 하나 보다 많은 결합 그룹의 파트일 수 있다. 이러한 충돌을 해결하는 한가지 기술은 지정된 최대 결합 그룹을 선택하는 것이다. 포트가 이러한 레지스터(295)를 사용해서 결합 그룹에 추가되면, 다음 커맨드가 결합 그룹의 문맥 내에서 사용되고, 새로운 포트가 준비되기 전에 커맨드가 발행되지 않아야만 한다. 포트가 결합 그룹으로부터 제거될 때, 즉시 자체적으로 디스에이블되거나 사용될 수 있다.

[0034] 또한, 포트 당 1 비트가 레지스터에 할당되어 복제 커맨드 체크 레지스터(296)로서 본 명세서에 기술된 바와 같이 복제 커맨드 체크가 가능하다. 포트가 임의의 그룹에 결합되면, 커맨드 값을 연속 포트로 검사한다. 그룹에 결합되지 않으면, 연속 사이클에서 복제가 발견된다.

[0035] 도 2i는 결합 디멀티플렉서(295)의 일 실시예를 도시한다. 일 실시예에서, 결합 명령들이 주어지면, 포트 준비 레인들(295)(예를 들어, port_rdy 레인들)이 개별 포트들(290)에 의해 생성된다. 예를 들어, 4개의 포트들(290)이 결합될 때, 모든 포트 준비 레인들(295)이 표명된다. 그러나, 포트들(290)의 단지 포트들 2 및 3이 2 포트 그룹에 결합되면, 포트 준비 레인들(295)의 port_rdy[3:2]가 표명된다. 유사하게, 포트 1이 단독으로 작업중이면, 오직 port_rdy[1]만이 표명된다. 이 기술은 저장을 위한 전체 메모리 워드를 생성하는 래치들 뿐만 아니라 메모리 뱅크(289)로 향하는 포트들(290)로부터 정확한 포트 준비 레인들(295)까지의 라우팅 및 전송 크기를 결정하는데 사용된다.

[0036] 도 2j는 결합 디멀티플렉서 라우팅을 도시한 표들(296, 297)의 실시예들을 도시한다. 예를 들어, 데이터가 도착하면, 디멀티플렉서는 라우팅 함수표(296)에 따라 정확한 레인들로 포트 데이터를 라우팅한다. 그 후 디멀티플렉서 레지스터는 표(297)에 기술된 바와 같이 enable_fn(도 2i의 298)의 함수에 따라 메모리를 위해 결합된 정확한 포트 레인들로 데이터를 캡처링한다. 모든 데이터가 래치되면, 코어는 wr_strobe를 사용해서 enable_fn(298)에 의해 데이터를 저장할 것을 명령받는다.

[0037] 기록 마스크 기능 또는 선택된 데이터의 저장을 금지하는 기능은 병렬 데이터 경로를 사용해서 달성될 수 있다. 저장 사이클의 시작에서, 모든 마스크들이 enable_fn에 따라 세팅된다(예를 들어, 모든 레인들 금지). 데이터가 도착하면, 연관된 마스크들이 라우팅되고 데이터와 함께 저장된다. 모든 데이터가 도착하지 않으면(예를 들어, 중단(interrupt)되거나 또는 짧은 전송(short transfer)), 단지 도착된 데이터만이 저장되는데, 도착하지 않은 데이터 레인들이 연관된 마스크들을 제거할 기회를 갖고 있지 않기 때문이다.

[0038] 도 2k는 결합 멀티플렉서(293)의 일 실시예를 도시한다. 일 실시예에서, 포트 준비(예를 들어, port_rdy)(295) 및 판독 커맨드(예를 들어, read_cmd 신호들은 메모리의 판독 레이턴시(RL)(299)에 의해 지연되어서, 데이터는 출력 포트들(290)을 선택하기에 알맞은 시간에 메모리로부터 도착(및 래치)한다. 이러한 포트들의 선택은 간단히 지연된 입력 값들을 사용해서 실행된다. 판독 커맨드로부터의 port_rdy 레인들(295)은 디멀티플렉서(294)의 해석과 유사하게 멀티플렉서(293)에 의해 해석된다. 멀티플렉서는 도 2i의 표(279)에 도시된 함수들을 기반으로 출력 포트들(290)에 대한 레인 매핑을 선택한다.

[0039] 간결하게 편의상, 각각의 사이클에서 메모리로부터 하나의 (64-비트) 데이터 워드를 전달할 수 있으며 싱글 포트 케이스들을 커버하도록 데이터 레이트를 감소시킬 수 있는 스토리지 또는 지연이 존재한다고 가정된다. 출력 사이클들 보다 데이터를 폐치하는데 더 많은 사이클들이 요구되는 경우에, 코어는 메모리로부터 더 큰 워드들을 로드하고 연속 사이클들 동안 더 짧은 세그먼트들을 선택하는 '프리-페치 버퍼'로 생성된다. 이러한 경우에, 데이터 래치는 프리-페치 버퍼와 결합될 수 있다. 데이터를 조절하기 위해, 커맨드 인터프리터는 판독 커맨드들을 더 짧은 양들로 분할할 수 있으며 중간 커맨드들을 더 느린 속도로 발행하여 출력 레이트를 매치할 수 있다.

[0040] 도 3은 프레임 동기화 프로세스의 일 실시예를 도시한다. 초기에, 메모리 포트는 시스템 리셋(302)을 위해 파워-오프된다. 포트를 파워업하기 위해, /LPD(Link Power-Down)이 제로와 동일할 때, /LPD는 하이로 구동되고(304), 포트는 디스에이블된다. 그러나, /LPD가 1과 동일할 때, SYNC라고 하는 특정 코드 또는 비트 시퀀스에 대한 프레임 탐색이 시작된다(306). SYNC가 검출될 때, 프로세스는 동작 모드(308)로 된다. 이 프로세스는 도 4에 도시된 바와 같이 (사용되는 경우) 다수의 포트들에 대해서 계속될 수 있다.

- [0041] 호스트 및 메모리가 데이터를 직렬로 교환하기 때문에, 수신기는 프레임 내의 비트의 위치의 일치 (correspondence)를 결정하도록 동기화된다. 정확한 동기화를 결정하기 위해, 링크는 "프레임 탐색" 상태(306) 중에 특정 비트 시퀀스를 탐색한다. 예를 들어, 처음에, 직렬 링크는 2개의 동기화 비트 시퀀스들: SYNC 및 SYNC2 중 하나를 송신한다. 호스트 및 메모리 모두에 의해 사용되는, Rx-PHY는 프레임 데이터 패킷들을 검출한다. SYNC는 리셋 또는 에러 후에 링크 브링업(link bring-up)에 주요 역할을 한다. 또한, SYNC는 정상 오퍼레이션에서 임의의 비사용 프레임들 중에 메모리 Tx-PHY에 의해 송신된다. 호스트 Tx-PHY는 정상 오퍼레이션에서 비사용 프레임들 중에 SYNC 또는 SYNC2를 송신한다. 프로세스는 정상 오퍼레이션 모드(308)로 진행하여, SYNC가 메모리로부터 검출 및 인식된다. 예를 들어, 20 비트 디코딩 에러로 표시된 바와 같이, 프레임이 실패하면, SYNC가 다시 한번 검출될 때까지 메모리는 "프레임 탐색" 상태(306)로 돌아간다. 임의의 상태에서, /LPD가 0으로 되면, 포트는 "링크 다운" 상태로 돌아가서 다시 시작된다.
- [0042] SYNC2는 "링크 다운" 상태로 남겨짐으로 인해 또는 프레임 에러로 인해, 호스트 데이터를 수신하는 중의 에러를 나타내도록 메모리에 의해 송신된다. 메모리가 프레임링을 재설정하고 SYNC 송신을 개시할 때까지, 호스트는 SYNC를 배타적으로 송신함으로써 응답한다. 호스트는 적합한 에러 복구 오퍼레이션을 위해 커맨드들 간에 SYNC2를 송신한다. SYNC 및 SYNC2는 링크 프레임링을 설정 및 복구하고 호스트는 링크 설정을 조정한다.
- [0043] 도 4는 전력 컨트롤 프로세스의 일 실시예를 도시한다. /LPD(link power down)이 수신된다. 결정 블록(402)에서, 포트가 온 상태인지에 대한 결정이 이루어진다. /LPD 앞의 슬래시(/)는 역 로직, 예를 들어, /LPD가 0와 동일할 때, 링크가 파워 다운됨 - 이는 파워업 되지 않음을 의미함 - 을 나타낸다. 유사하게, 1과 동일한 /LPD는 파워 다운되지 않음을 나타내고, 이는 파워업을 의미한다. /LPD가 1과 동일하면(예를 들어, 링크가 파워업 되면), 프로세싱 블록(404)에서 트레이닝 프로세스가 실행되어, 특정 코드 또는 비트 시퀀스(예를 들어, SYNC)에 대해 프레임을 탐색한다. SYNC 탐색에 대한 트레이닝 프로세스는 SYNC가 검출될 때까지 계속되고, 그 후, 프로세스는 프로세싱 블록(406)에서 동작 모드로 들어간다. 이 프로세스는 도 3과 관련해서 더 기술된다. 결정 블록(408)에서, 포트 에러가 있는지에 대한 결정이 이루어진다. 포트 에러가 있으면, 프로세스는 결정 블록(402)으로 진행한다. 포트 에러가 없으면, 결정 블록(410)에서 더 많은 포트들이 계속해서 동작 모드로 되는지에 대한 결정이 이루어진다. 다른 포트들이 추가되지 않으면, 프로세스는 프로세싱 블록(406)에서 계속 동작 모드이다. 그러나, 추가 포트들이 검출되면, 프로세스는 프로세싱 블록(414)에서 새로운 포트들을 트레이닝한다.
- [0044] 추가 (다수의) 포트들은 프로세싱 블록(416)에서 처리된다. 다수의 포트들의 사용은 도 9와 관련해서 기술된다. 결정 블록(418)에서, 포트 에러가 결정된다. 포트 에러가 결정되면, 예를 들어, 결정 블록(430)에서 포트에 의해 야기된 것 과 같이 파워 다운(예를 들어, /LPD = 0)인지가 결정된다. 그렇다면, 모든 포트들은 프로세싱 블록(432)에서 디스에이블되고 프로세스는 결정 블록(434)에서 계속해서 싱글 포트 모드일 수 있다. 결정 블록(434)에서, /LPD가 0이 아니면(예를 들어, /LPD = 1), 모든 포트들은 프로세싱 블록(436)에서 트레이닝되고 프로세스는 프로세싱 블록(416)으로 진행한다. 다시 결정 블록(430)으로 돌아가서, /LPD가 0이 아니면(예를 들어, /LPD = 1), 프로세스는 프로세싱 블록(428)에서 계속해서 에러 포트들을 트레이닝한 후, 프로세싱 블록(416)으로 더 진행한다.
- [0045] 다시 결정 블록(418)으로 돌아가서, 포트 에러들이 발견되지 않으면, 결정 블록(420)에서 추가되는 포트들이 더 있는지에 대한 다른 결정이 이루어진다. 추가 포트들이 있으면, 프로세스는 프로세싱 블록(414)에서 새로운 포트들의 트레이닝 프로세스(예를 들어, 새로운 포트들 각각에 대한 SYNC 탐색)를 계속한다. 추가 포트들이 없으면, 결정 블록(422)에서 포트들 중에서 임의의 포트가 제거되는지에 대한 결정이 이루어진다. 그렇지 않으면, 프로세스는 프로세싱 블록(416)으로 진행한다. 그렇다면, 임의의 제거된 포트들이 프로세싱 블록(424)에서 디스에이블된다. 이 때에, 결정 블록(426)에서, 싱글 포트가 싱글 포트 모드로 돌아가는데 유효한지에 대한 결정이 이루어진다. 그렇다면, 프로세스는 프로세싱 블록(406)으로 싱글 포트 모드가 계속된다. 그렇지 않으면, 프로세스는 프로세싱 블록(416)으로 다중 포트 모드가 계속된다.
- [0046] 전력 컨트롤(238)(도 2a)은 /LPD를 전파할 책임이 있으며, 포트들(206-212)은 도 3 및 도 4의 트레이닝 및 프로세싱의 책임이 있다.
- [0047] 도 5는 싱글 포트를 사용하는 복제 체크 및 커맨드 해석 프로세스의 일 실시예를 도시한다. 다수의 포트들을 사용하는 더 복잡한 프로세스는 도 9에 도시되어 있다. 도 5에 도시된 바와 같이, 데이터의 수신, 판독 및 디코딩은 프로세싱 블록(502)에서 제1 프레임으로 시작해서 포트(예를 들어, 제1 포트 또는 마스터 포트)에서 실행된다. 결정 블록(504)에서, 포트 에러가 검출되었는지에 대한 결정이 이루어진다. 에러가 검출되면, 프로세

스는 블록(528)에서 에러 리턴으로 종료한다. 포트 에러가 검출되지 않았으면, 결정 블록(506)에서, 복제가 온 인지에 대한 결정이 이루어진다. 복제가 필요 또는 요구에 의해 턴 온 또는 오프될 수 있다고 생각된다. 복제가 온이면, 프로세스는 계속해서 프로세싱 블록(508)에서 제2 프레임으로 이제 포트에서 데이터의 수신, 판독 및 디코딩이 실행된다. 다시, 결정 블록(510)에서, 포트 에러가 검출되는지에 대한 결정이 이루어진다. 검출되면, 프로세스는 블록(528)에서 에러 리턴으로 종료한다.

[0048] 포트 에러가 검출되지 않으면(또한, 결정 블록(506)으로 다시 돌아가서, 복제가 온이 아니면), 프로세스는 프레임이 커맨드인지 또는 데이터인지에 대한 결정을 한다. 프레임이 커맨드이면, 결정 블록(516)에서 커맨드가 유효한지에 대한 결정이 이루어진다. 커맨드가 유효하지 않으면, 프로세스는 블록(528)에서 에러 리턴으로 종료한다. 커맨드가 유효하면, 결정 블록(518)에서 커맨드가 시퀀스인지 또는 적소에 있는지에 대한 결정이 이루어진다. 커맨드가 시퀀스가 아니면, 프로세스는 블록(528)에서 에러 리턴으로 종료한다. 커맨드가 시퀀스이면, 커맨드는 프로세싱 블록(520)에서 처리되어 정상 리턴이 블록(530)에서 야기된다.

[0049] 다시 결정 블록(514)으로 돌아가서, 프레임이 데이터이면, 결정 블록(522)에서 기록 오버레이션들을 위한 메모리가 준비되어 있는 지에 대한 결정이 이루어진다. 준비되어 있지 않으면, 프로세스는 블록(528)에서 에러 리턴으로 종료한다. 준비되어 있으면, 데이터는 프로세싱 블록(524)에서 메모리에 기록되고, 프로세스는 블록(530)에서 정상 리턴으로 종료한다. 본 실시예에서, 블록들(516, 518, 520)의 프로세스들은 도 2a의 커맨드 인터프리터(248)에서 실행되고, 나머지 프로세스들은 도 2a의 포트들(206-212)에서 실행된다.

[0050] 도 6은 한 포트 내에서 각종 기능들을 실행하는 프로세스의 일 실시예를 도시한다. 블록(602)에서, 데이터 스트림의 수신, 판독 및 디코딩의 프로세스가 제공된다. 예를 들어, 싱글 데이터 스트림(비트)이 Rx를 통해 포트에서 수신되어 병렬 스트림들로 형성된 후, 본 명세서에서 설명된 바와 같이 (예를 들어, 17B/20B 디코딩을 사용해서) 디코딩된다. /LPD(link power down) 신호는, 도 2a에 도시된 바와 같이(예를 들어, 도 2a의 점선은 /LPD의 전력 컨트롤을 나타냄) 싱글 호스트 결합 포트 메모리의 모든 포트들에서 들어가고 나오는 전력과 같은, 모든 포트들에 대한 전력을 (전력 컨트롤 메커니즘을 통해) 제어하는데 사용된다. 결정 블록(604)에서, /LPD가 0과 동일한지에 대한 결정이 이루어진다. 제로이면, 블록(614)에서 에러가 리턴되어 프로세스가 종료된다. 그러나, /LPD가 0과 동일하지 않으면, 프로세스는 프로세싱 블록(606)으로 진행한다.

[0051] 프로세싱 블록(606)에서, 비트 방식의(bitwise) 데이터 스트림을 수신하고 프레임들의 병렬 스트림을 생성하는 포트를 포함하는 데이터 프레임이 판독된다(예를 들어, 20 비트, 역직렬화). 프로세싱 블록(608)에서 프레임이 (예를 들어, 17B/20B 디코딩 기술을 사용해서) 디코딩된 후, 유효성(validity) 데이터가 생성된다. 결정 블록(610)에서, 프레임의 유효성이 체크된다; 예를 들어, 프레임이 정확하게 17 비트 값으로 디코딩된 20 비트 코드를 가졌는지에 대한 결정이 이루어진다. 이러한 변환이 실패하면, 모호성으로 인해 임의의 결과를 야기하지 않는 것과 같이, 유효성 검사가 실패하고 블록(614)에서 에러가 리턴된다. 그러나, 변환이 성공적이고 결과들이 생성되면, 데이터 프레임은 유효로 간주되고 블록(612)에서 정상이 리턴되며, 이는 도 9에서 더 설명된다.

[0052] 도 7a는 17-비트 포스트-디코딩 프레임 포맷(700)의 일 실시예를 도시한다. 17-비트 디코딩 프레임(700)의 본 실시예는 17-비트 데이터, 커맨드들, 및/또는 상태를 송신하고, 직렬 송신을 위한 20-비트 프레임들을 생성하기 위해 트랜지션(transition) 인코딩을 받는데 사용될 수 있다. 데이터, 커맨드, 및 상태는 20-비트 프레임들로 송신 및 수신된다. 수신될 때, 20-비트 트랜지션 인코딩 프레임이 데이터, 커맨드, 및 상태를 유지하는 17-비트 프레임(700)을 생성하기 위해 디코딩을 받는 역 프로세스가 실행된다.

[0053] 본 17-비트 포스트-디코딩 프레임(포맷)(700)은 처음 16 비트들을 페이로드 지시자(702)에 제공하고, 최종 1 비트(예를 들어, 17번째 비트)를 페이로드 지시자(704)에 제공한다. 메모리 액세스 포맷들은 기본 디코딩 포맷들 토대로 한다. 예를 들어, 비트 16(704)은 데이터, 커맨드, 또는 상태에 대해 페이로드가 1로 세팅되었는지 또는 0으로 세팅되었는지를 나타낸다. 커맨드들 및 기록 데이터는 프레임 기준(frame-by-frame basis)으로 수신 기 링크를 공유할 수 있다. 레이턴시를 감소시키기 위해, 커맨드들은 기록 커맨드의 완료를 지연시키면서, 기록 데이터 스트림에 삽입되거나 선취할 수 있다.

[0054] 도 7b는 커맨드, 상태, 및 데이터 인코딩 프레임 포맷(720)의 일 실시예를 도시한다. 본 실시예는 SPDRAM(Serial Port DRAM) 커맨드, 상태, 및 데이터 인코딩 프레임(720)의 일 실시예를 포함하지만, 이로만 제한되지는 않는다. 본 17-비트 인코딩 프레임(720)은, 예를 들어, 기술 변경 또는 요구와 같은 차후의 추가 커맨드들을 추가하기 위해 다수의 비트들을 보존하는 유연성을 제공하기에 확장 가능하다. 예를 들어, 플래그들(722) 및 서브-커맨드들(724)은 프레임(720)의 처음 7개의 비트들(비트 0 내지 7)을 차지하고, 서브-커맨드들(724)의 대부분의 엔트리들이 1이기 때문에, 플래그 영역(722)을 포함하는 영역은 프레임(720)을 확장하면서,

차후에 추가 커맨드들(예를 들어, 최대 16개의 커맨드들)을 추가하는데 사용될 수 있다.

- [0055] 유사하게, 추가 커맨드들(예를 들어, 단지 3개의 커맨드들을 갖는 모드 레지스터 그룹(726)의 서브-커맨드 영역)을 위해 사용될 수 있는, 모드 레지스터 그룹(726)과 같이, 제한된 범위를 갖는 다른 섹션들이 있다. 다른 영역은 다른 커맨드들을 추가하는데 사용될 수 있는 DRAM 커맨드 그룹(728)(예를 들어, 모두 1을 갖는 DRAM 커맨드 그룹(728)의 서브-커맨드 영역)이다.
- [0056] SYNC(730)는 링크 프레임 동기화를 제어 및 유지하고, SYNC2(732)는 특정 링크 동작 상태를 나타낸다. SYNC(730, 732)는 도 3 및 도 4와 관련해서 더 기술되어 있다. 데이터 프레임(734)은 2개의 8-비트 바이트들이 이어지는 1로 세팅된 17 비트를 갖는 도 7a에 도시된 데이터 프레임(700)에 유사한 17-비트 프레임을 포함한다. 활성화 बैं크(ABNK)(736) 및 활성화(ACT)(738)는 도 7c에서 기술된다. 기록 커맨드(WR)(740)는 지정된 बैं크 및 열에 대한 메모리 기록 사이클을 개시한다. 기록 마스크(WMSK)(742)는 진행중인 기록 커맨드를 위한 8 바이트 마스크를 세팅하고 임의의 효과를 갖도록 WR 커맨드(740)를 수행한다. WMSK(742)는 도 7d를 참조해서 더 기술된다.
- [0057] 판독(RD)(744)은 메모리 판독 사이클을 개시하는 판독 커맨드와 관련되고, 버스트 정지(BSTP)(746)는 지정된 बैं크에 따라 포트의 현재 판독 또는 기록 커맨드를 중단하라는 커맨드와 관련된다. 프리차지(PCG)(748)는 커맨드에서 지정된 बैं크를 프리차지하는 커맨드와 관련되며, PCA(precharge all)(750)는 모든 बैं크들을 동시에 프리차지하는 커맨드를 포함한다. REFB(per-bank refresh)(752)는 지정된 बैं크가 자동으로 리프레시됨을 나타내고, REFA(all-bank refresh)(754)는 내부 카운터에 따라 모든 बैं크들이 리프레시됨을 나타낸다. 모든 बैं크들은 REFA 커맨드를 발행하기 전에 프리차지 상태이다.
- [0058] 모드 레지스터 기록(MRW)(758)은 모드 레지스터에 기록을 실행하는 커맨드와 관련된다. 모드 레지스터 기록 데이터(MRD)(760)는 MRD 커맨드(760)의 형태로 포트 0으로부터 다음 중간 프레임에서 MRW 커맨드(758)를 따르는 기록 데이터를 제공한다. 모드 레지스터 판독(MRR)(756)은 모드 레지스터로부터 판독을 실행하는 커맨드와 관련된다. 자체 리프레시 파워 다운(SRPD)(762)은 메모리 코어가 즉시 자체 리프레시 상태로 되게 한다. 파워 다운 종료(PDX)(764)는 자체 리프레시 파워 다운을 종료하도록 야기된 커맨드와 관련되고, 링크 설정 후에 메모리 코어를 웨이크하는데 사용된다.
- [0059] 도 7c는 ABNK 및 ACT 커맨드들(736, 738)의 일 실시예를 도시한다. 2개 이상의 커맨드들이 동시에 송신되게 하기 위해, 서로의 기능들을 지원하거나 또는 기능적으로 직교할 수 있다. 제3 기준은 복잡성을 포함하는데, 메모리 의미 또는 구현 결정이 직교성에 있어서 실패를 야기할 수 있기 때문이다. 예를 들어, 직렬 포트 DRAM은 बैं크를 활성화하는 커맨드를 가질 수 있으며, 활성화되는 행 어드레스는 한 프레임에 대해 너무 길다. 싱글 포트 경우에, 이러한 커맨드는 2개 이상의 프레임들을 요구할 수 있지만, 결합 포트들의 경우, 2개 이상의 포트들에서 1 프레임 시간에 통신될 수 있다.
- [0060] 예를 들어, ABNK(736)는 다음 활성화(ACT) 커맨드들(738)에 의해 사용되도록 상위 5-비트들의 행 어드레스(754) 및 타겟 बैं크(752)를 세팅한다. 활성화 커맨드(738)는 최종 ABNK 커맨드(736)에서 지정된 बैं크(752)에 송신된다. 2개 이상의 포트들이 결합되면, 선택적인 ABNK(736) 커맨드가 포트 2에서 나타날 수 있다. 행 어드레스(764)의 하위 15 비트들은 ACT 커맨드(738)의 최하위 15 비트들에서 지정되며, 5개의 최상위 비트들은 포트 2에서 나타나는 ABNK(770) 또는 최종 ABNK 커맨드(736)의 하위 5 비트들에서 지정된다. 본 일례는, 각각의 커맨드(736, 738)가 임의의 시간에 다음 프레임들에서 독립적일 수 있음을 나타낸다. 이는 가변 포트 그룹 크기들, 포트 그룹 크기와 무관한 공통 컨트롤러들, 및 포트 결합에 대한 일관성 있는 의미를 가능케 한다. 또한, 커맨드들(738, 770)은 서로 보완하며, 동시에 실행될 수 있다.
- [0061] 도 7d는 WMSK 및 WR 커맨드(742, 740)의 일 실시예를 도시한다. 도 7d는 선택적인 기록을 위해 WR 커맨드(742) 및 연관된 바이트/기록 마스크(742)를 도시한다. WMSK(742)는 진행중인 WR 커맨드(740)를 위한 8 바이트 마스크(772)를 세팅하고 임의의 효과를 갖도록 WR 커맨드(740)에 이어지는 커맨드를 나타낸다. 8 바이트들의 데이터가 통신된 후에, 마스크(772)는 다음 8 바이트들에 대해 다시 시작한다. 마스크(772)의 레터 'H'는 하이 바이트(예를 들어, 비트들 15 내지 8)를 나타내고, 'L'은 워드 전송의 로우 바이트(예를 들어, 비트들 7 내지 0)를 나타낸다.
- [0062] WR 커맨드(740)는 지정된 बैं크(774) 및 열(776)에 대한 메모리 기록 사이클을 개시한다. WR 커맨드(740)가 송신되면, 기록 데이터가 이어진다. 2개 이상의 포트들이 결합되면, 선택적인 WMSK 커맨드(780)가 포트 2에서 송신되어서, 처음 8 바이트들을 커버링 또는 마스크(778)한다. 다음 WMSK 커맨드에 의해 리셋되지 않는 한, 마스크

크(778)는 모든 8 바이트들에 대해 반복한다. 2개 이상의 포트 결합의 다른 일레들은, 메모리 및 인터페이스 의미에 따라, 동시 관독 및 기록 또는 동시 활성화 및 기록의 조합들을 포함한다.

[0063] 도 8a, 도 8b, 및 도 8c는 기록 마스크 모델들(800, 850, 875)의 일 실시예를 도시한다. 직렬 통신을 사용하는 메모리의 경우, 분할할 수 없는 전송을 위한 비트들의 수의 감소가 레이턴시 감소에 사용된다. 분할 불가능 전송은 하나의 총 데이터 양(예를 들어, 바이트) 또는 예를 들어, '기록' 및 타겟 어드레스 등의 커맨드를 완료하는데 필요한 임의의 즉각적인 피연산자 데이터를 포함하는 실행 가능 커맨드를 기술하는 프레임 또는 워드 길이로서 비트들로 정의된다.

[0064] 대부분의 메모리들의 경우, 기록 오퍼레이션은 WR 커맨드, 어드레스, 오퍼레이터(이 경우 마스크), 및 기록 데이터를 동시에 포함한다. 더 빠른 메모리 디바이스들의 경우, 커맨드를 기술하는데 필요한 속도는 금지가 되어서, 버스트 전송이 사용된다. 버스트 전송은 커맨드 및 초기 데이터로 개시되지만, 다음 어드레스들이 계산되는(예를 들어, 증가) 데이터의 스트림으로 계속된다. 데이터가 전송될 때마다, 추가 기록 마스크 표시 신호들이 수반된다.

[0065] 직렬 통신의 경우, 커맨드, 어드레스, 기록 마스크, 및 데이터를 모두 한번에 코딩하는 것은 비효율적일 수 있는데, 커맨드 및 어드레스가 다음 데이터 전송에 필요하지 않을 수도 있기 때문이다. 이 정도까지, 데이터를 위탁하기 위해 버스트 전송을 사용하며, 데이터는 WR 커맨드 및 어드레스에 이어진다. 레이턴시를 감소시키기 위해, 기록 마스크, 또는 WMSK 커맨드(예를 들어, 바이트 당 1 비트)는 기록 버스트 내의 로케이션들에 대한 값들이 저장되지 않을 때 좌우되는 데이터를 오직 수반할 필요가 있다. 이러한 최적화가 직렬 인터페이스 효율성을 위해 중대할 수 있더라도, 이러한 기법은 병렬 메모리 인터페이스에서 대역폭 요구 사항들을 감소시키는데 사용될 수 있다. 직렬 인터페이스들이 멀티-호스트 메모리의 실용성(practicality)을 향상시키기 때문에, 각각의 호스트는 커맨드 스트림 내에 기록 마스크를 배치함으로써 독립적인 전송들을 위해 독립적인 기록 마스크들을 사용하는 기능을 갖는다. 레이턴시를 감소시키기 위해 데이터를 WMSK에 포함시키는 종속성을 감소시키기 위해, 버스트 내의 3개의 사용 모델들이 가정되고 도시된다.

[0066] 도 8a는 이전 메모리 콘텐츠(802), 커맨드 스트림(804), 및 이후 메모리 콘텐츠(806)를 포함하는 반복 패턴 WMSK 모델(800)의 일 실시예를 도시한다. 본 도면에서, WMSK는 각각의 전송에 대해 반복된다; 레드, 그린, 및 블루 데이터로 구성된 사각형 내의 레드 값만을 변경하는 것과 같이, 다른 2개의 컬러들은 마스크되며, WMSK는 사각형 내의 모든 RGB(레드, 그린, 블루) 데이터에 대해 반복된다. 도 8b는 이전 메모리 콘텐츠(852), 커맨드 스트림(854), 및 이후 메모리 콘텐츠(856)를 포함하는 초기 및 최종 WMSK 모델(850)의 일 실시예를 도시한다. 여기서, 기록 마스크는 전송의 초기 파트에 대해서만 사용된다. 예를 들어, 네트워크 패킷은 패킷의 데이터 구조들의 나머지에 대한 액세스(정렬)를 최적화하기 위해 외진 전송 경계(4 바이트 전송 중 제2 바이트)에서 개시할 수 있다. 초기 마스크가 소진되면, 패킷 데이터의 나머지 전체는 메모리에 기록된다. 전송을 완료하기 위해, 새로운 WMSK가 최종 2 바이트들을 트림(trim)하도록 삽입된다. 도 8c는 이전 메모리 콘텐츠(876), 커맨드 스트림(878), 및 이후 메모리 콘텐츠(880)를 포함하는 반복 패턴 WMSK 모델(875)을 위한 다수의 직렬 인터페이스들을 사용하는 일 실시예를 도시한다. 여기서, WMSK는 전송 크기 내의 하나의 데이터 구조를 선택하는 싱글 전송에 사용된다. 예를 들어, 32-비트 정수에서 제2 바이트만을 기록.

[0067] 모델들(800, 850)에 있어서, 기록 마스크들은 재사용되거나 드물게 사용된다. 예를 들어, 캐시 기록 및 대용량 저장 전송 등의 수개의 타입들의 전송은 마스크를 요구하지 않는다. 이러한 경우에, 대부분의 시간에 사용되지 않기 때문에 기록 마스크에 데이터를 포함시키는 것은 비효율적이다. 작거나 또는 유닛 전송 보다 더 작은 전송은 버스트 전송의 장점들을 수확하지 않아서, 데이터, 커맨드, 어드레스 및 기록 마스크는 모두 지정된다. 이러한 짧은 전송은 대개 캐시 메모리 내부에서 발생해서, 버스트-지향 메모리를 이러한 타입의 빈번한 오퍼레이션들로부터 구제한다.

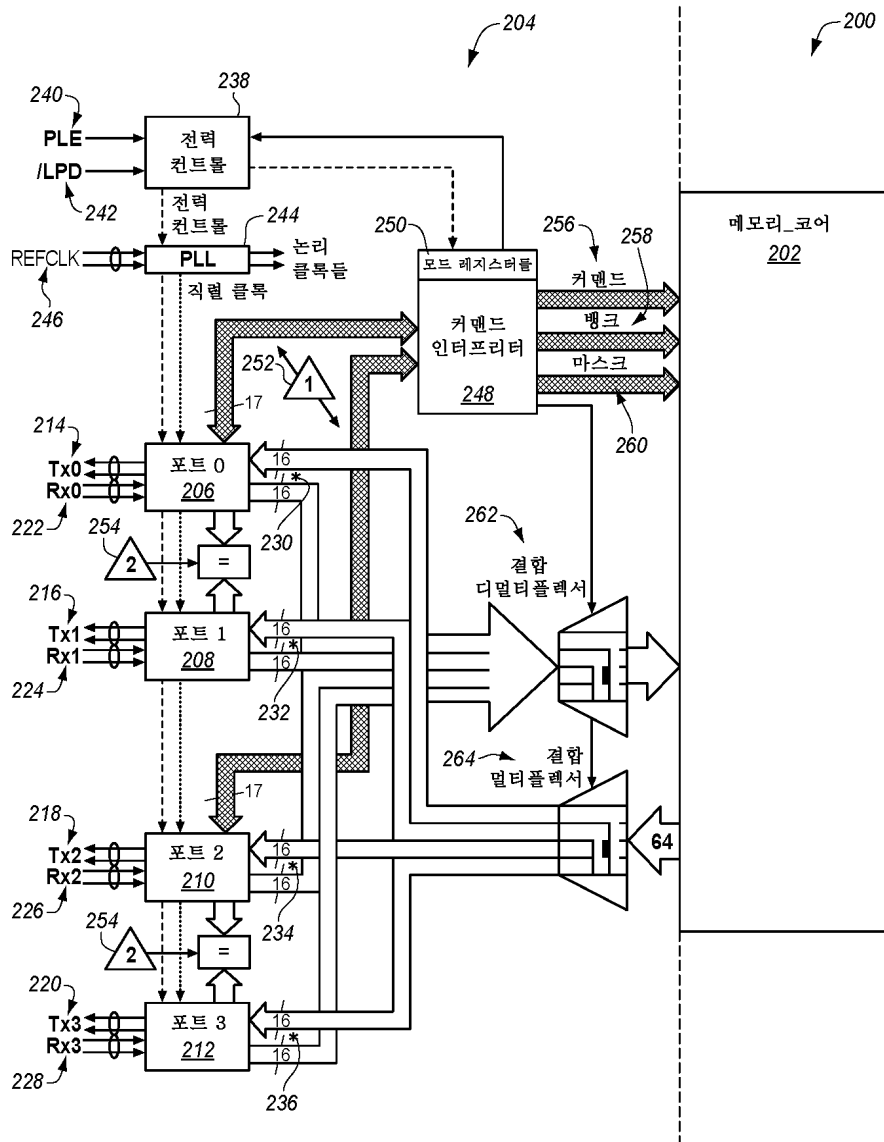
[0068] 모델들(800, 850)의 가정에 집중해서, 기록 마스크는 데이터를 포함하지만, 버스트 전송의 장점들이 수확되면, 기록 마스크를 커맨드와 결합시키는 것은 불충분하다. 이 정도까지, 커맨드 및 데이터로부터 기록 마스크 전송의 분리(decoupling)는 새로운 커맨드로서 구현되었다. 분할 불가능 전송들(프레임들)로 구성된 싱글 직렬 스트림에서, 기록 커맨드는 하나의 프레임에서의 어드레스, 및 프레임들의 시퀀스에서 계산된 메모리 어드레스들에 대한 데이터 스트림들을 발행받으며, 기록 마스크들은 개별 커맨드로서 기술되고 유닛 버스트들에 적용되며, 기록 커맨드 후에 요구된 데이터 내에서 발행된다. 유닛 버스트는 싱글 기록 마스크 비트가 적용되는 비트들의 수 \times 기록 마스크 커맨드 내의 기록 마스크 비트들의 수로서 정의된다. 기록 커맨드가 발행될 때, 모든 다음 데이터가 기록되는 등, 기록 마스크는 제거된다. 기록 마스크 커맨드가 기록 커맨드에 바로 이어지면, 제1 유

닛 버스트로부터 시작된다.

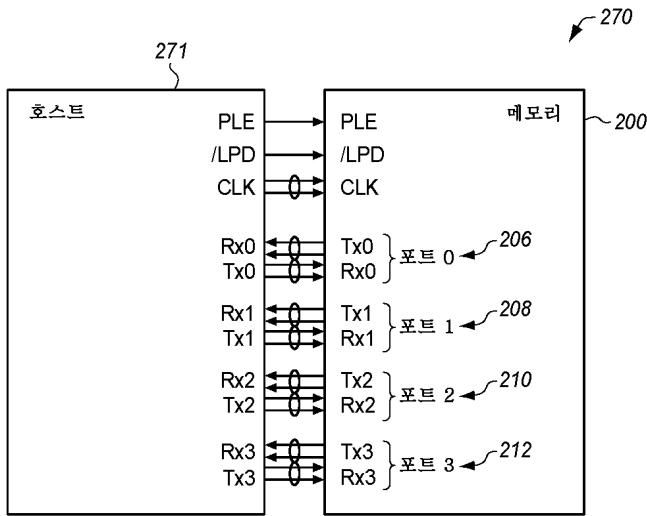
- [0069] 모델(800)에서 기술된 반복 패턴이 사용되면, 마스크는 모든 유닛 버스트들에 대해 반복한다. 패턴이 전송 내에서 변경되면, 추가 기록 마스크 커맨드가 발행되어, 새로운 기록 마스크가 모든 다음 데이터에 적용되게 한다. 모델(850)에 기술된 초기 패턴이 사용되면, 제1 유닛 버스트 후에 초기 마스크가 제거된다. (예를 들어, 최종 유닛 버스트 내에서) 추가 마스크가 요구되면, 추가 기록 마스크 커맨드가 발행되어, 다음 유닛 버스트에만 적용되며, 이 때에 기록 마스크는 제거된다.
- [0070] 모델(875)의 경우, 마스크가 반복되지만, WMSK 커맨드가 상이한 포트에서 WR 커맨드와 동시에 발생하는 모델(800)의 멀티포트 버전이 사용된다. 다수의 직렬 인터페이스들이 사용되면, 더 유연한 커맨드 레이아웃을 생성할 수 있다. 2개의 포트들이 함께 사용되면, 예를 들어, 기록 커맨드는 하나의 포트에서 다른 포트의 제1 기록 마스크와 결합되어서, 대역폭 사용을 향상시킬 수 있다.
- [0071] 도 9는 다수의 포트들을 사용하는 복제 체크 및 커맨드 해석 프로세스의 일 실시예를 도시한다. 프로세스 포트들은 프로세싱 블록(904)에서의 다수의 포트들 중 제1 포트로 블록(902)에서 개시된다. 데이터를 갖는 데이터 스트림은 프로세싱 블록(906)에서 대응 Rx를 통해 제1 포트에서 수신된다(그 후 디코딩됨). 프로세싱 블록(906)의 항 "포트 m+i"를 참조하면, "m"은 결합 그룹을 나타내고, "i"는 결합 그룹의 수를 나타낸다. 본 실시예에서, i는 0에서 시작하고, 싱글 호스트가 사용되기 때문에, m은 0과 동일하다. 결정 블록(908)에서, 제1 포트(포트 0)는 임의의 에러들에 대해 체크된다. 에러가 발견되면, 프로세스는 블록(942)에서 에러 리턴으로 종료한다. 에러가 발견되지 않으면, 프로세스는 모든 포트들이 체크될 때까지 프로세싱 블록(910)에서 다음 포트에 대한 체크를 계속한다. 예를 들어, 결정 블록(912)에서, 포트들이 더 남아있는지에 대한 결정이 이루어진다. 그렇다면, 프로세스는 계속해서 다음 포트에 대해 프로세싱 블록(906)으로 진행한다. 그렇지 않으면, 프로세스는 프로세싱 블록(914)으로 진행한다.
- [0072] 결정 블록(916)에서, 복제가 온인지에 대한 결정이 이루어진다. 온이면, 결정 블록(918)에서, 현재 포트가 결정되고, 결과에 따라, 프로세스는 블록(942)에서 에러 리턴으로 종료하거나, 또는 (다시 결정 블록(916)으로 돌아가서, 복제가 온이 아니면) 프로세스는 결정 블록(920)으로 진행하여, 포트가 데이터를 갖는지에 대한 결정이 이루어진다. 데이터가 복제중이 아니면, 비교가 되지 않는다. 데이터가 존재하면, 기록 오퍼레이션이 결정 블록(936)에서 실행된다. 기록이 진행중이 아니면, 프로세스는 블록(942)에서 에러 리턴으로 종료한다. 기록 오퍼레이션이 실행중이면, 데이터는 프로세싱 블록(938)에서 모든 포트들로부터 메모리에 기록되고 정상 리턴이 블록(940)에서 실행된다.
- [0073] 다시 결정 블록(920)으로 돌아가서, 포트가 데이터를 갖지 않으면, 결정 블록(922)에서 커맨드 유효성 검사가 실행된다. 결정 블록(922)에서, 예를 들어, 커맨드 리스트를 체크함으로써 포트 커맨드가 유효한지에 대한 결정이 이루어진다. 커맨드가 유효하지 않으면, 프로세스는 블록(942)에서 종료한다. 커맨드가 유효하면, 결정 블록(924)에서, 커맨드가 시퀀스에 있는지(예를 들어, 커맨드가 적소에 있는지)에 대한 결정이 이루어진다. 커맨드가 시퀀스에 없으면, 에러는 블록(942)에서 리턴된다. 커맨드가 시퀀스에 있으면, 커맨드는 프로세싱 블록(926)에서 처리된다.
- [0074] 결정 블록(928)에서, 다음 쌍의 포트들에서 데이터의 복제가 존재하는지를 알기 위해 다음 포트가 체크된다. 데이터의 복제는 일반적으로 한 쌍의 포트들을 수반하기 때문에, 프로세싱 블록(930)에서 포트들의 수는 2씩 증가되며 다음 두 포트들이 체크된다. 다시 결정 블록(928)으로 돌아가서, 답이 예이면, 다음 (싱글) 포트가 프로세싱 블록(934)에서 선택된다. 그 후 프로세스는 결정 블록(932)으로 진행하여, 처리될 포트들이 더 있는지에 대한 결정이 이루어진다. 그렇다면, 프로세스는 결정 블록(916)으로 진행한다. 그렇지 않으면, 정상 리턴이 블록(940)에서 야기된다.
- [0075] 일 실시예에서, 데이터는 하나의 포트에서 수신되고 커맨드는 상기 포트에서 수신된다. 커맨드는 블록들(922, 924, 926)로 표현된 (도 2a에 도시된 바와 같이) 커맨드 인터프리터(248)에서 처리되고, 복제 체크는 블록들(916, 918, 920)로 표현된 (도 2a에 도시된 바와 같이) 2 삼각형(254)에 의해 도시된 로케이션에서 실행된다.
- [0076] 도 10은 커맨드 복제 모델들(1002, 1004, 1006)의 일 실시예를 도시한다. 일 실시예에서, 커맨드 복제가 강화된 에러 검출을 위해 사용된다. 커맨드들은 두번 송신되고 고유 커맨드는 복제 커맨드와 비교된다. 1개의 또는 2개의 포트들이 사용되면(1002, 1004), 복제 커맨드(1010, 1014)는 고유 커맨드(1008, 1012) 바로 뒤의 프레임에 나타난다. 4개 이상의 포트들이 사용되면(1006), 복제 커맨드들(1016, 1018)이 다른 포트들에서 나타난다.

- [0077] (1) 결합 포트 상황에서, 복제가 비사용 대역폭을 채우기 위해 사용될 수 있기 때문에; (2) 커맨드 시퀀싱 위반 (예를 들어, 이미 활성화된 बैं크를 활성화하거나 또는 활성화되지 않은 बैं크를 기록함) 또는 현재 전송과 관련되지 않은 메모리 로케이션에 오류를 일으킴 등의 예상치 못한 결과들을 커맨드 오역이 야기할 수 있기 때문에, - 반면, 커맨드가 정확하면, 임의의 나쁜 데이터는 적어도 현재 전송으로 제한됨 -, 및 (3) 데이터 반복으로 우수한 결과들을 야기하더라도, 커맨드들에 유효한 자유 공간이 데이터 스트림에서 유효하지 않음으로 인해, 효율적인 시스템 대역폭이 절반으로 되기 때문에, 커맨드들은 구체적으로 선택된다.
- [0078] 커맨드 복제 모델들(1002, 1004, 1006)은 싱글 포트(1002) 및 결합 포트들(1004, 1006)의 복제와의 조합물들을 도시한다. 복제 및 다수의 커맨드들이 어떻게 함께 작용하는지를 더 도시한다. 예를 들어, 최대 2개의 상이한 커맨드들이 한 프레임 시간에 전송된다.
- [0079] 싱글 포트 모델(1002)에서, 커맨드들은 단독으로 발행되고 그 복제물(1010)이 커맨드(1008) 뒤에 이어진다. 2 포트 모델(1004)의 경우, 복제 커맨드(1014)는 동일한 프레임 시간에 송신된다; 그러나, 복제가 턴 오프되면, 2개의 커맨드들이 프레임 시간을 차지할 수 있다. 4개 이상의 포트들 모델(1006)의 경우, 2개(이상)의 커맨드들(1020, 1022)이 한 프레임 시간을 차지할 수 있으며, 2개의 커맨드들(1020, 1022)이 모두 복제 커맨드들(1016, 1018)로 동일한 프레임 시간에 복제될 수 있다. 동시에 송신되는 커맨드들의 수를 제한할 필요가 없으며, 그룹 내의 포트들의 수의 입도(granularity)를 제한할 필요가 없다.
- [0080] 사용 모델(1002, 1004, 1006)에 따라, 기회주의적으로 커맨드들을 실행하는 것이 만족스러울 수 있다. 이는 일부 경우들에서 레이턴시를 절약하지만, 에러 가능성 및 에러를 처리하는 비용에 대해 불리하다. 복제물들이 한 프레임 시간에 모두 유효하면, 에러 결과들이 즉시 유효하다.
- [0081] 상술된 설명에서, 설명을 위해, 다수의 특정 세부사항들이 본 발명의 철저한 이해를 제공하기 위해 기재된다. 그러나, 본 발명이 일부의 특정 세부사항들 없이 구현될 수도 있음이 당업자에게 명백하다. 다른 실례들에서, 널리 공지된 구조들 및 디바이스들이 블록도 형태로 도시되어 있다. 도시된 컴포넌트들 간의 중간 구조일 수 있다. 본 명세서에 기술 또는 도시된 컴포넌트들은 도시 또는 기술되지 않은 추가 입력들 또는 출력들을 가질 수 있다.
- [0082] 본 발명의 각종 실시예들은 각종 프로세스들을 포함할 수 있다. 이 프로세스들은 하드웨어 컴포넌트들에 의해 실행될 수 있으며, 또는 범용 또는 특별 목적 프로세서 또는 논리 회로들이 프로세스들을 실행하는 명령들로 프로그래밍되게 하는데 사용될 수 있는 컴퓨터 프로그램 또는 기계 실행 가능 명령들로 구현될 수 있다. 대안으로, 프로세스들은 하드웨어 및 소프트웨어의 조합으로 실행될 수 있다.
- [0083] 멀티-호스트 강화 메카니즘의 일 실시예와 연관된 또는 그 내부에 도시된 바와 같은, 본 문서에 기술된 하나 이상의 모듈들, 컴포넌트들, 또는 소자들은 하드웨어, 소프트웨어, 및/또는 그 조합을 포함할 수 있다. 모듈이 소프트웨어를 포함한 경우에, 소프트웨어 데이터, 명령들, 및/또는 구성은 기계/전자 장치/하드웨어에 의한 제조 물품을 통해 제공될 수 있다. 제조 물품은 명령들, 데이터 등을 제공하기 위해 콘텐츠를 갖는 기계 액세스 가능/판독 가능 매체를 포함할 수 있다. 콘텐츠는, 전자 장치, 예를 들어, 본 명세서에 기술된 파일러, 디스크, 또는 디스크 컨트롤러가 기술된 각종 오퍼레이션들 또는 실행 사항들을 실행하게 할 수 있다.
- [0084] 본 발명의 각종 실시예들의 일부는, 본 발명의 실시예들에 따라 프로세스를 실행하도록 컴퓨터(또는 다른 전자 장치들)를 프로그래밍하는데 사용될 수 있는, 컴퓨터 프로그램 명령들이 저장되어 있는 컴퓨터 판독 가능 매체를 포함할 수 있는, 컴퓨터 프로그램 제품으로서 제공될 수 있다. 기계 판독 가능 매체는, 플로피 디스켓, 광 디스크, CD ROM, 및 광자기 디스크, ROM, RAM, EPROM, EEPROM, 자석 또는 광 카드, 플래시 메모리, 또는 전자 명령들을 저장하기에 적합한 다른 타입의 미디어/기계 판독 가능 매체를 포함할 수 있는데, 이로만 제한되지는 않는다. 더욱이, 본 발명은 컴퓨터 프로그램 제품으로서 다운로드될 수 있으며, 프로그램은 원격 컴퓨터로부터 요청 컴퓨터로 전송될 수 있다.
- [0085] 여러 방법들이 가장 기본적인 형태로 기술되지만, 본 발명의 기본 범위 내에서 프로세스들은 임의의 방법들로부터 삭제되거나 추가될 수 있으며, 정보는 기술된 메시지들 중 임의의 메시지에서 감해지거나 추가될 수 있다. 다수의 다른 변경들 및 적용들이 달성될 수 있음이 당업자에게 명백하다. 특정 실시예들은 본 발명을 제한하려고 제공되는 것이 아니라 설명을 위해 제공된다. 본 발명의 실시예들의 범위는 상술된 특정 일례들에 의해 결정되지 않으며, 오직 이하의 청구항들에 의해서만 결정된다.
- [0086] 요소 "A"는 요소 "B"에 연결된다고 하면, 요소 A는 요소 B에 직접 연결될 수 있으며, 또는 예를 들어, 요소 C를 통해 간접적으로 연결될 수 있다. 명세 또는 청구항들이, 컴포넌트, 피쳐, 구조, 프로세스, 또는 특징 A가 컴

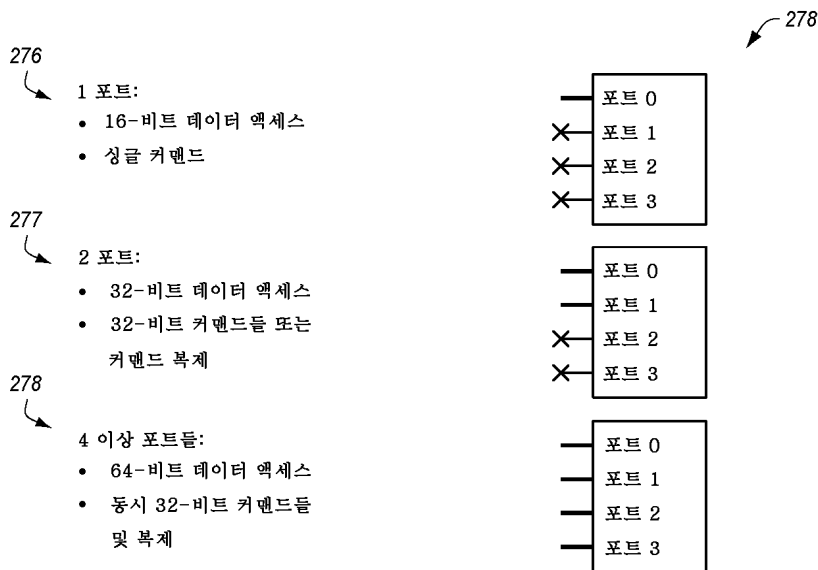
도면2a



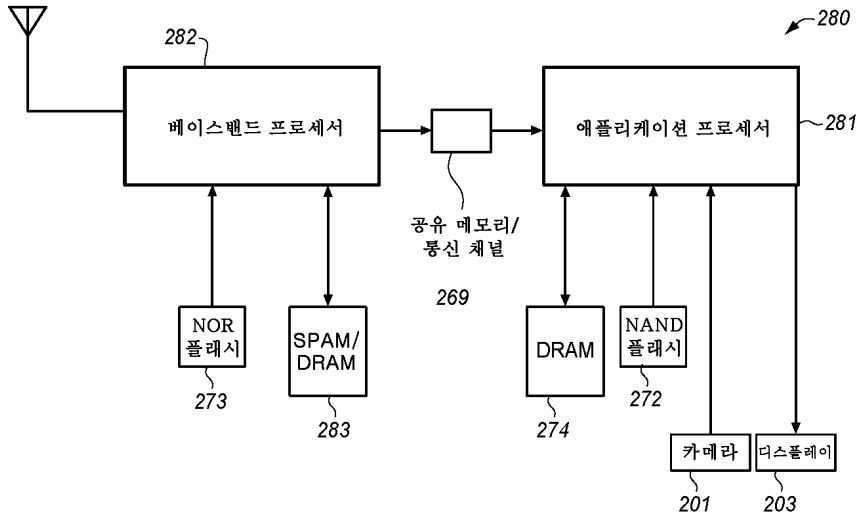
도면2b



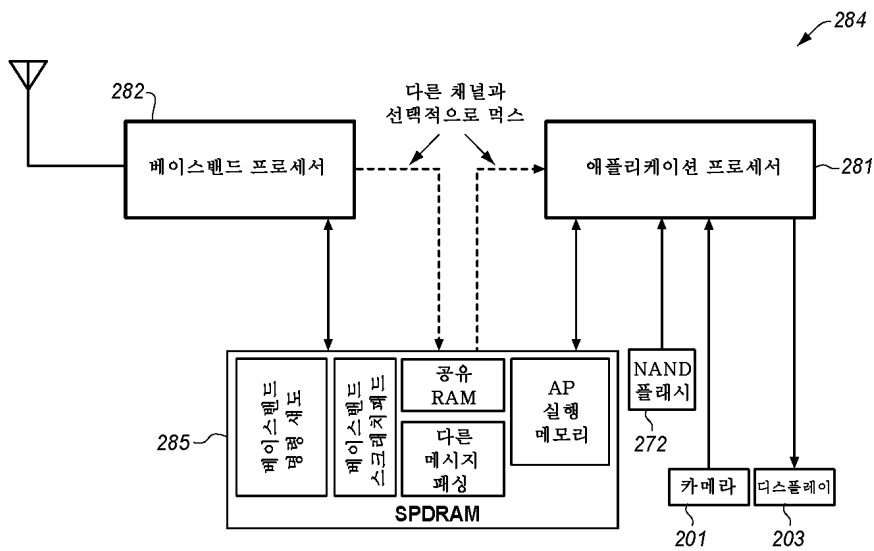
도면2c



도면2d

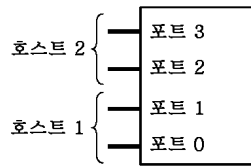


도면2e

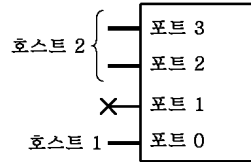


도면2f

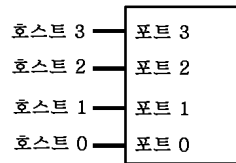
- 2 독립 호스트들
- 듀얼 포트/호스트
- ↖
286



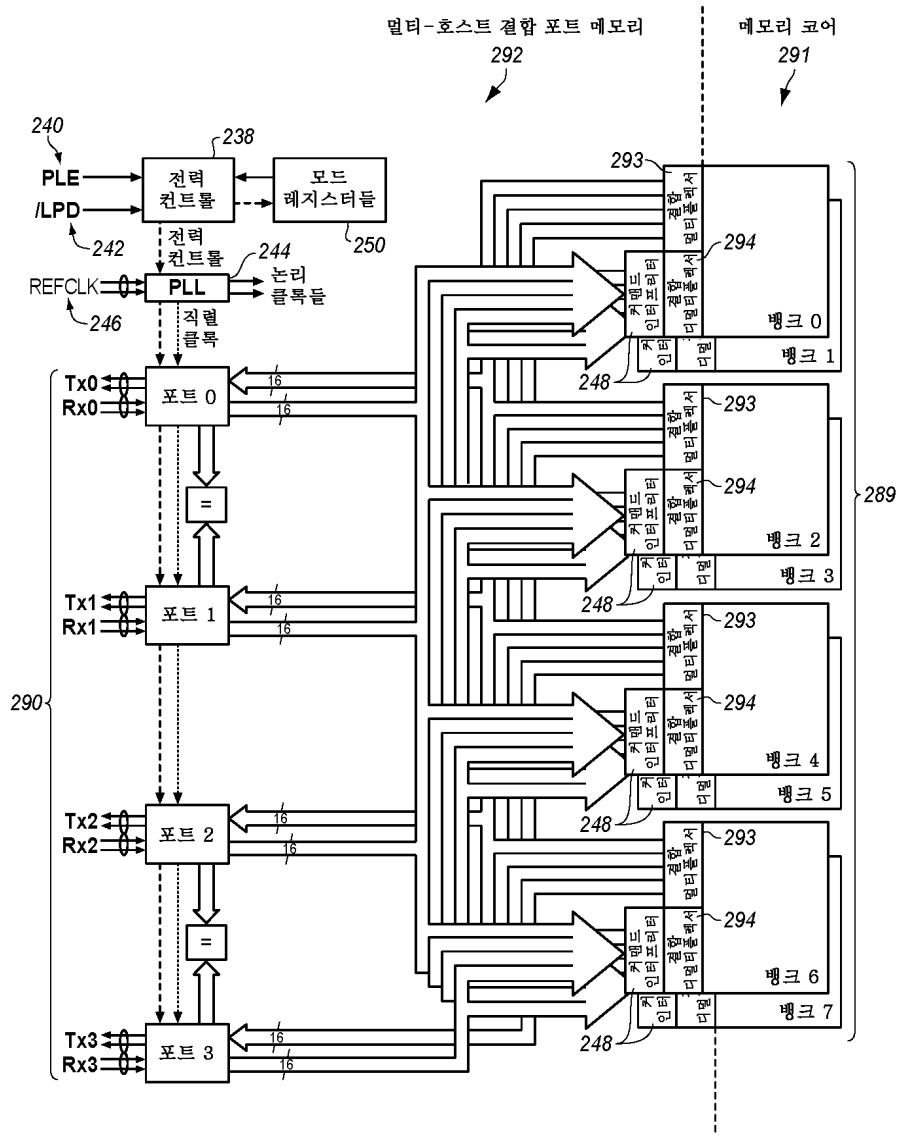
- 2 독립 호스트들
- 호스트 1: 싱글 포트
 - 호스트 2: 듀얼 포트
- ↖
287



- 4 독립 호스트들
- 포트 당 하나의 호스트
- ↖
288



도면2g



도면2h

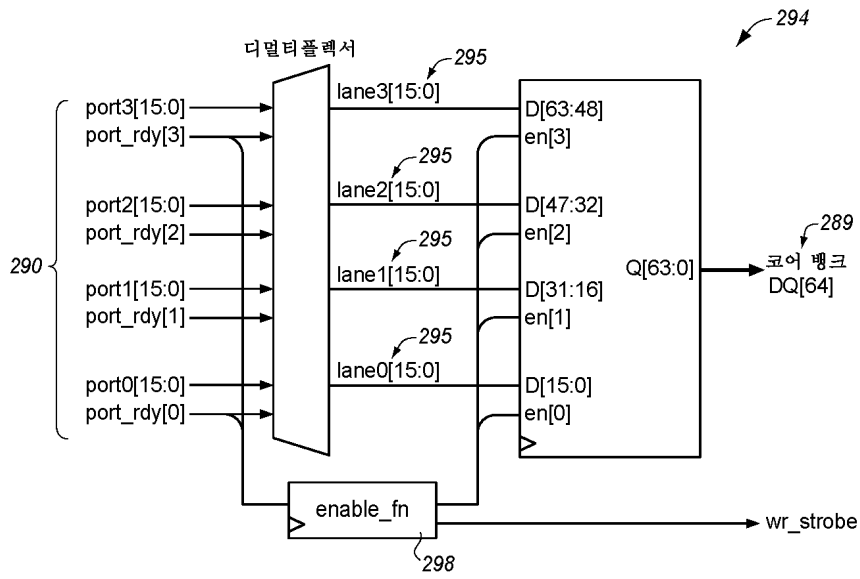
| | | | | | | | | | | | | | | | |
|-------------|-------------|--------------|-------------|--------------|--------------|--------------|------------|---|------------|------------|------------|------------|------------|------------|------------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 포트들 15-0 | 포트들 15-8 | 포트들 15-12 | 포트들 11-8 | 포트들 15-14 | 포트들 13-12 | 포트들 11-10 | 포트들 9-8 | X | 포트들 7-0 | 포트들 7-4 | 포트들 3-0 | 포트들 7-6 | 포트들 5-4 | 포트들 3-2 | 포트들 1-0 |

↖-2

| | | | | | | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 포트 15 | 포트 14 | 포트 13 | 포트 12 | 포트 11 | 포트 10 | 포트 9 | 포트 8 | 포트 7 | 포트 6 | 포트 5 | 포트 4 | 포트 3 | 포트 2 | 포트 1 | 포트 0 |

↖296

도면2i



도면2j

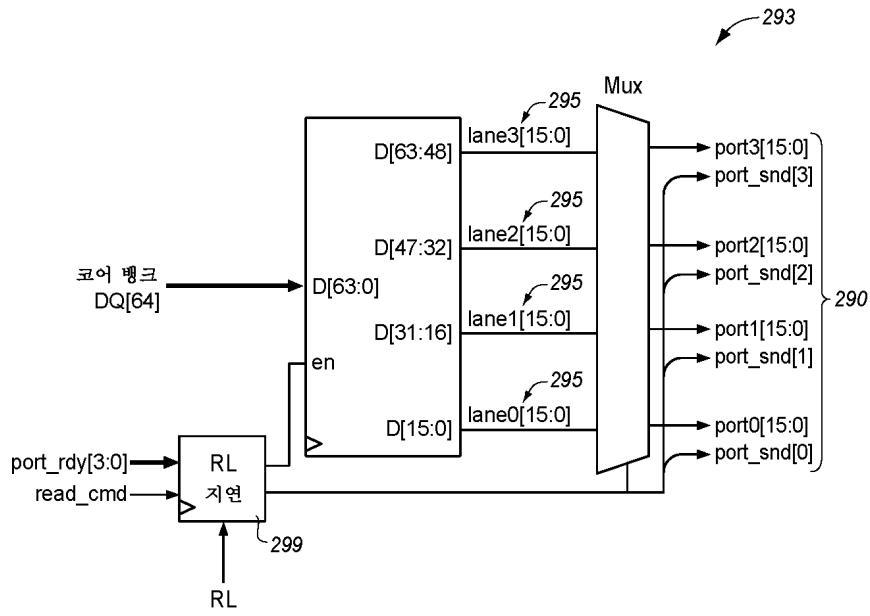
296

| port_rdy[3:0] | lane0 | lane1 | lane2 | lane3 |
|---------------|--------------|--------------|--------------|--------------|
| 0000 | Don't Care | Don't Care | Don't Care | Don't Care |
| 0001 | port0 | port0 | port0 | port0 |
| 0010 | port1 | port1 | port1 | port1 |
| 0011 | port0 | port1 | port0 | port1 |
| 0100 | port2 | port2 | port2 | port2 |
| 0101 | 무효 | 무효 | 무효 | 무효 |
| 0110 | 무효 | 무효 | 무효 | 무효 |
| 0111 | 무효 | 무효 | 무효 | 무효 |
| 1000 | port3 | port3 | port3 | port3 |
| 1001 | 무효 | 무효 | 무효 | 무효 |
| 1010 | 무효 | 무효 | 무효 | 무효 |
| 1011 | 무효 | 무효 | 무효 | 무효 |
| 1100 | port2 | port3 | port2 | port3 |
| 1101 | 무효 | 무효 | 무효 | 무효 |
| 1110 | 무효 | 무효 | 무효 | 무효 |
| 1111 | port0 | port1 | port2 | port3 |

297

| port_rdy[3:0] | 사이클 | | | |
|---------------|-------------|-------------|-------------|-------------|
| | 0 | 1 | 2 | 3 |
| 0000 | 리셋 사이클 | 리셋 사이클 | 리셋 사이클 | 리셋 사이클 |
| 0001 | 0001 | 0010 | 0100 | 1000 |
| 0010 | 0001 | 0010 | 0100 | 1000 |
| 0011 | 0011 | 1100 | 0011 | 1100 |
| 0100 | 0001 | 0010 | 0100 | 1000 |
| 0101 | 무효 | 무효 | 무효 | 무효 |
| 0110 | 무효 | 무효 | 무효 | 무효 |
| 0111 | 무효 | 무효 | 무효 | 무효 |
| 1000 | 0001 | 0010 | 0100 | 1000 |
| 1001 | 무효 | 무효 | 무효 | 무효 |
| 1010 | 무효 | 무효 | 무효 | 무효 |
| 1011 | 무효 | 무효 | 무효 | 무효 |
| 1100 | 0011 | 1100 | 0011 | 1100 |
| 1101 | 무효 | 무효 | 무효 | 무효 |
| 1110 | 무효 | 무효 | 무효 | 무효 |
| 1111 | 1111 | 1111 | 1111 | 1111 |

도면2k



도면21

279

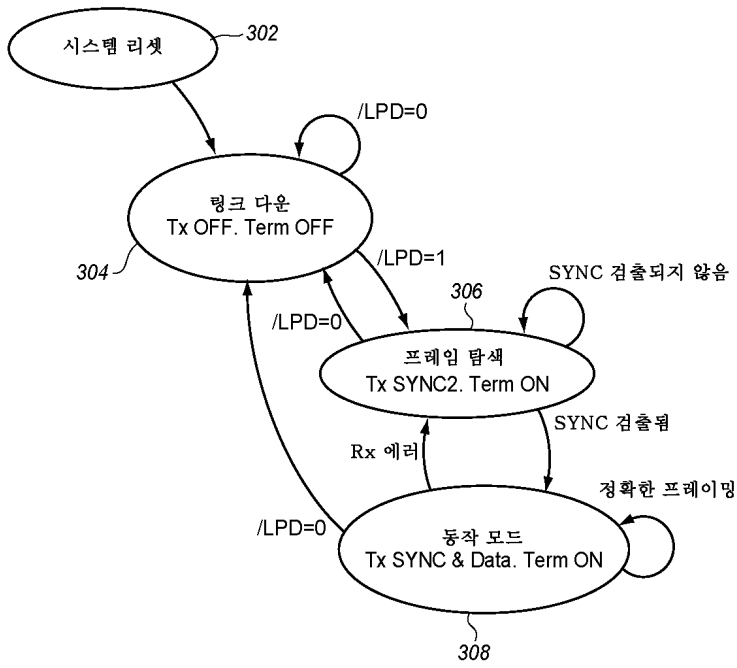
| 포트 0 | 사이클 | | | |
|---------------|-------|-------|-------|-------|
| port_snd[3:0] | 0 | 1 | 2 | 3 |
| XXX0 | None | None | None | None |
| 0001 | lane0 | lane1 | lane2 | lane3 |
| 0011 | lane0 | lane2 | lane0 | lane2 |
| 0101 | 무효 | 무효 | 무효 | 무효 |
| 0111 | 무효 | 무효 | 무효 | 무효 |
| 1001 | 무효 | 무효 | 무효 | 무효 |
| 1011 | 무효 | 무효 | 무효 | 무효 |
| 1101 | 무효 | 무효 | 무효 | 무효 |
| 1111 | lane0 | lane0 | lane0 | lane0 |

| 포트 1 | 사이클 | | | |
|---------------|-------|-------|-------|-------|
| port_snd[3:0] | 0 | 1 | 2 | 3 |
| XX0X | None | None | None | None |
| 0010 | lane0 | lane1 | lane2 | lane3 |
| 0011 | lane1 | lane3 | lane1 | lane3 |
| 0110 | 무효 | 무효 | 무효 | 무효 |
| 0111 | 무효 | 무효 | 무효 | 무효 |
| 1010 | 무효 | 무효 | 무효 | 무효 |
| 1011 | 무효 | 무효 | 무효 | 무효 |
| 1110 | 무효 | 무효 | 무효 | 무효 |
| 1111 | lane1 | lane1 | lane1 | lane1 |

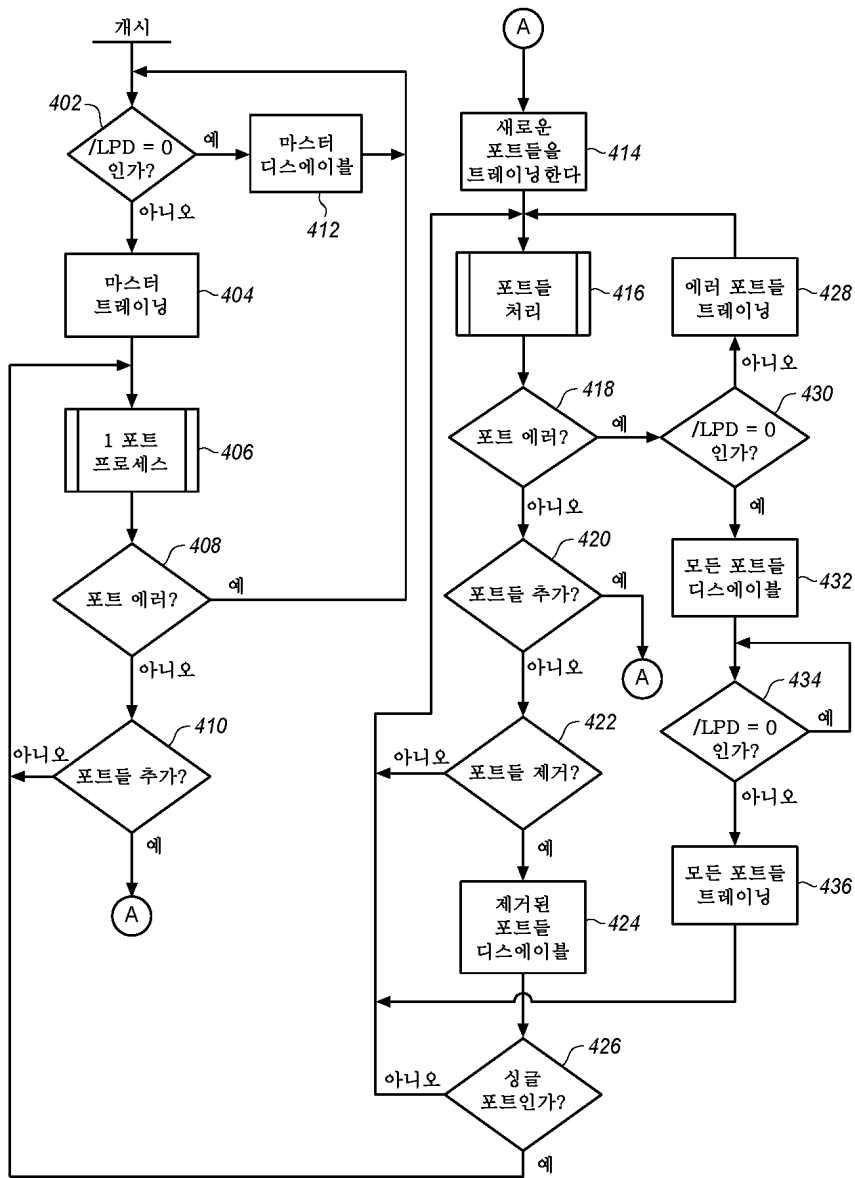
| 포트 2 | 사이클 | | | |
|---------------|-------|-------|-------|-------|
| port_snd[3:0] | 0 | 1 | 2 | 3 |
| X0XX | None | None | None | None |
| 0100 | lane0 | lane1 | lane2 | lane3 |
| 0101 | 무효 | 무효 | 무효 | 무효 |
| 0110 | 무효 | 무효 | 무효 | 무효 |
| 0111 | 무효 | 무효 | 무효 | 무효 |
| 1100 | lane0 | lane2 | lane0 | lane2 |
| 1101 | 무효 | 무효 | 무효 | 무효 |
| 1110 | 무효 | 무효 | 무효 | 무효 |
| 1111 | lane2 | lane2 | lane2 | lane2 |

| 포트 3 | 사이클 | | | |
|---------------|-------|-------|-------|-------|
| port_snd[3:0] | 0 | 1 | 2 | 3 |
| 0XXX | None | None | None | None |
| 1000 | lane0 | lane1 | lane2 | lane3 |
| 1001 | 무효 | 무효 | 무효 | 무효 |
| 1010 | 무효 | 무효 | 무효 | 무효 |
| 1011 | 무효 | 무효 | 무효 | 무효 |
| 1100 | lane1 | lane3 | lane1 | lane3 |
| 1101 | 무효 | 무효 | 무효 | 무효 |
| 1110 | 무효 | 무효 | 무효 | 무효 |
| 1111 | lane3 | lane3 | lane3 | lane3 |

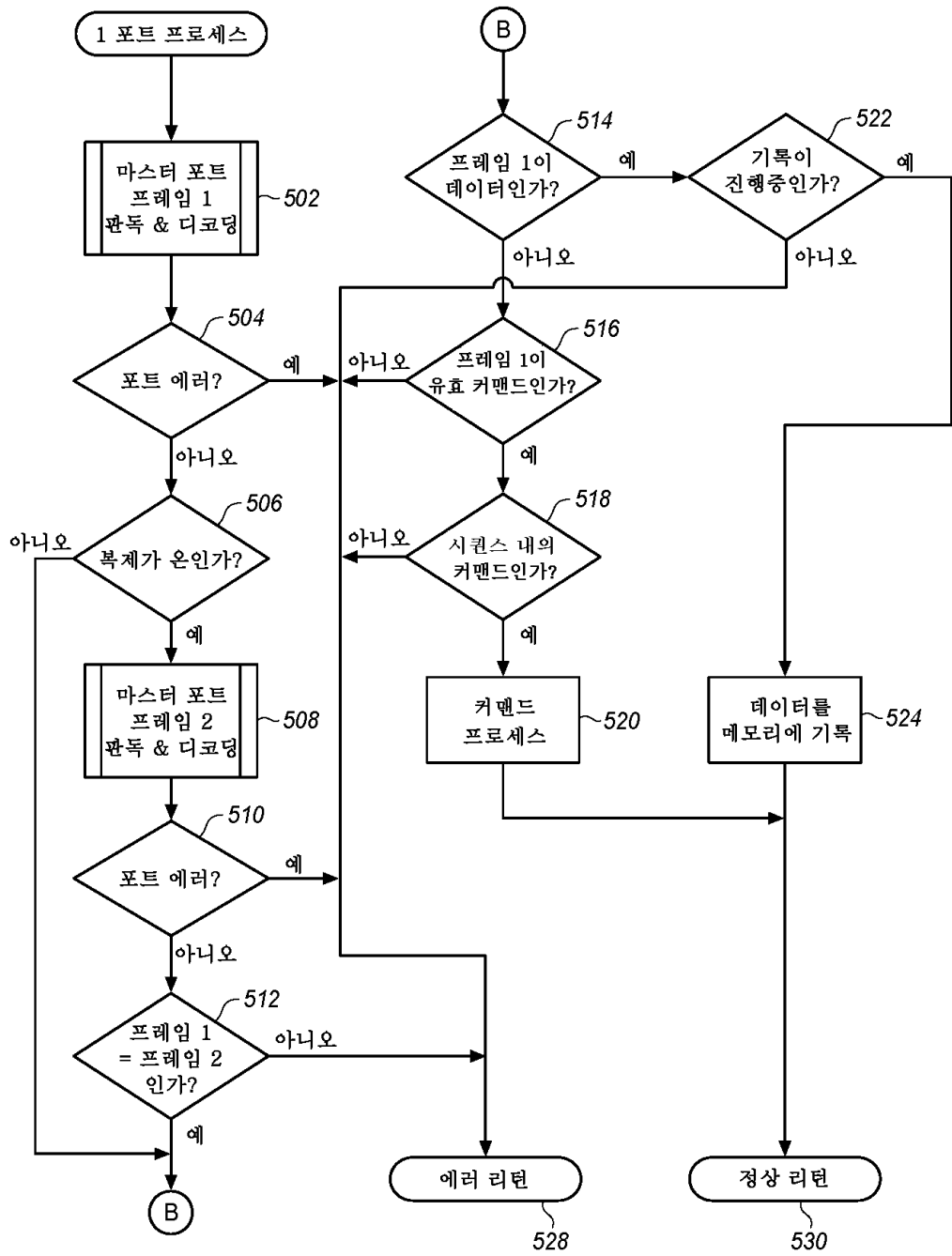
도면3



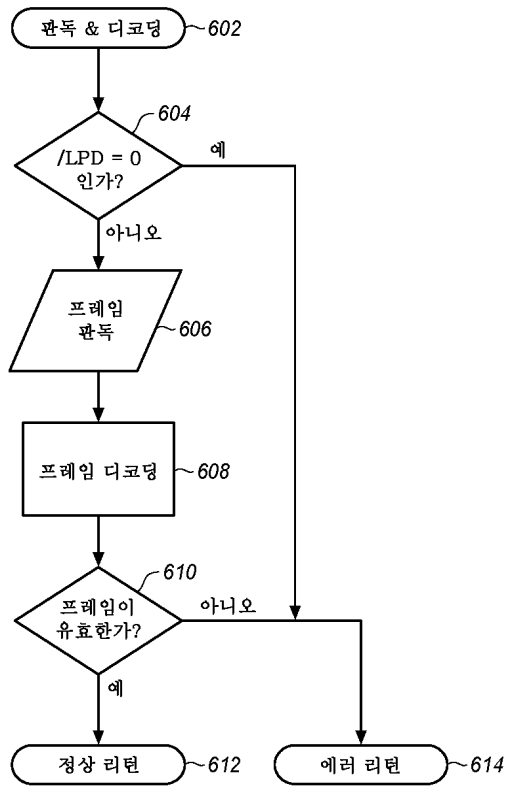
도면4



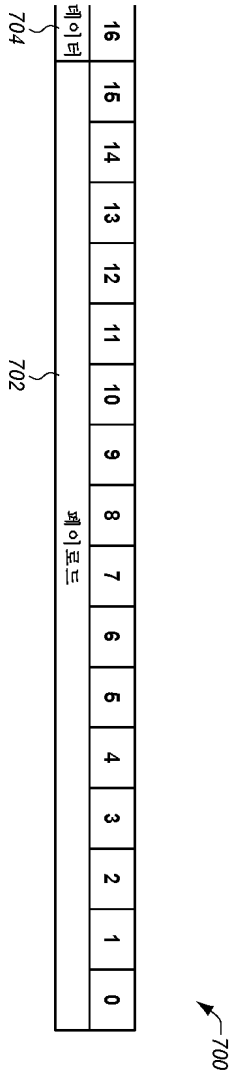
도면5



도면6



도면7a

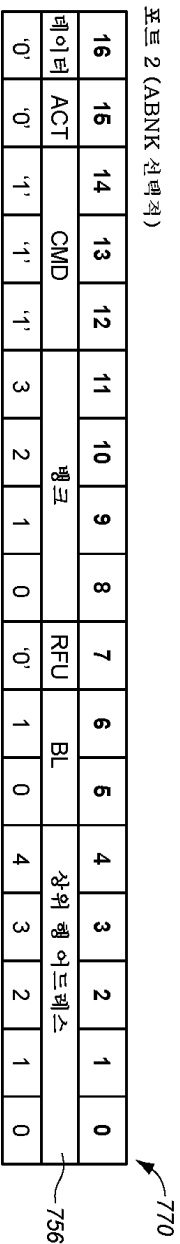
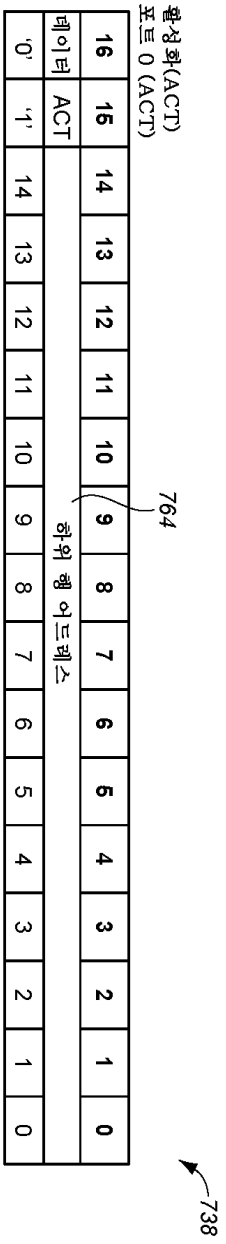
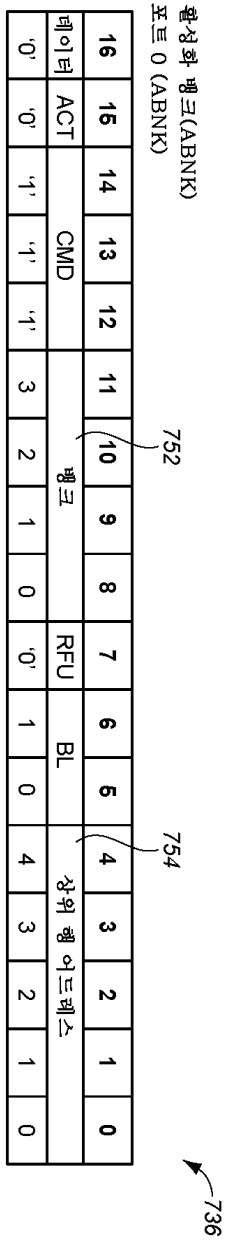


도면7b

↖ 720

| | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-------------|--------|------|----|----|---------|----|---|--------|----|------|---|---------|---|---|---|---|
| | 데이터 | ACT | CMD | | 뱅크 | | | | 서브-커맨드 | | | | 플래그 | | | | |
| | 뱅크 동기화 그룹 | | | | | | | | | | | | | | | | |
| 730 → SYNC | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 732 → SYNC2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| | DRAM 데이터 | | | | | | | | | | | | | | | | |
| 734 → DATA | 1 | 상위 바이트 | | | | | | | 하위 바이트 | | | | | | | | |
| | DRAM 커맨드 그룹 | | | | | | | | | | | | | | | | |
| 736 → ABNK | 0 | 0 | 1 | 1 | 1 | 뱅크 | | | RFU | BL | 상위 행 | | | | | | |
| 738 → ACT | 0 | 1 | 하위 행 | | | | | | | | | | | | | | |
| 740 → WR | 0 | 0 | 1 | 0 | AP | 뱅크 | | | Col | | | | | | | | |
| 742 → WMSK | 0 | 0 | 1 | 1 | 0 | 뱅크 | | | 마스크 | | | | | | | | |
| 744 → RD | 0 | 0 | 0 | 1 | AP | 뱅크 | | | Col | | | | | | | | |
| 746 → BSTP | 0 | 0 | 1 | 1 | 1 | 뱅크 | | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | |
| 748 → PCG | 0 | 0 | 1 | 1 | 1 | 뱅크 | | | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | |
| 750 → PCA | 0 | 0 | 1 | 1 | 1 | RFU | | | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | |
| 752 → REFB | 0 | 0 | 1 | 1 | 1 | 뱅크 | | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | |
| 754 → REFA | 0 | 0 | 1 | 1 | 1 | RFU | | | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | |
| | 모드 레지스터 그룹 | | | | | | | | | | | | | | | | |
| 756 → MRR | 0 | 0 | 1 | 1 | 1 | 상위 어드레스 | | | 1 | 0 | 0 | 1 | 하위 어드레스 | | | | |
| 758 → MRW | 0 | 0 | 1 | 1 | 1 | 상위 어드레스 | | | 1 | 0 | 1 | 0 | 하위 어드레스 | | | | |
| 760 → MRD | 0 | 0 | 1 | 1 | 1 | 상위 데이터 | | | 1 | 0 | 0 | 0 | 하위 데이터 | | | | |
| | 전력 관리 그룹 | | | | | | | | | | | | | | | | |
| 762 → SRPD | 0 | 0 | 1 | 1 | 1 | RFU | | | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | |
| 764 → PDX | 0 | 0 | 1 | 1 | 1 | RFU | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | |

도면7c



도면7d

기록 마스크(WMSK)
포트 0 (WMSK)

| | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|----|----|---|---|----|---------|----|----|----|----|----|----|
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 레이터 | ACT | | CMD | | | 뱅크 | | | | 마스크 비트들 | | | | | | |
| '0' | '0' | '1' | '1' | '0' | 3 | 2 | 1 | 0 | 3H | 3L | 2H | 2L | 1H | 1L | 0H | 0L |

742

772

기록(WR)

포트 0 (WR)

| | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|----|----|---|---|---|--------|---|---|---|---|---|---|
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 레이터 | ACT | | CMD | | | 뱅크 | | | | 열 어드레스 | | | | | | |
| '0' | '0' | '1' | '0' | PCG | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

774

740

776

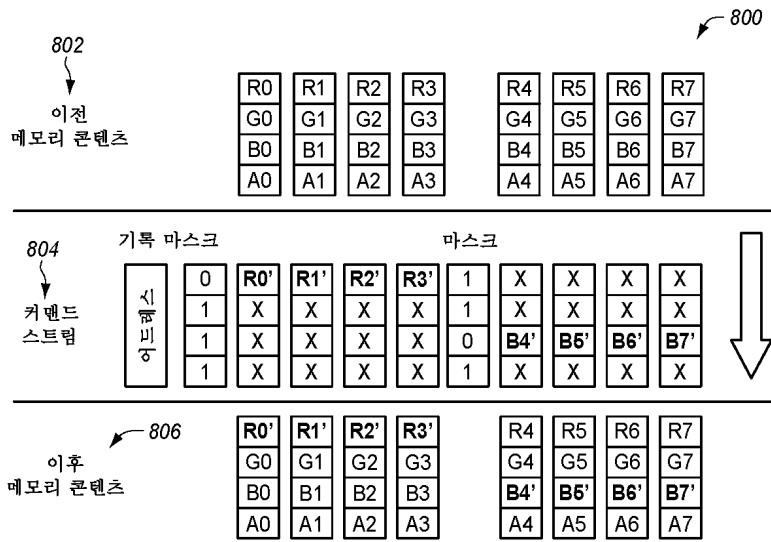
포트 2 (WMSK 선택적)

| | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|----|----|---|---|----|---------|----|----|----|----|----|----|
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 레이터 | ACT | | CMD | | | 뱅크 | | | | 마스크 비트들 | | | | | | |
| '0' | '0' | '1' | '1' | '0' | 3 | 2 | 1 | 0 | 3H | 3L | 2H | 2L | 1H | 1L | 0H | 0L |

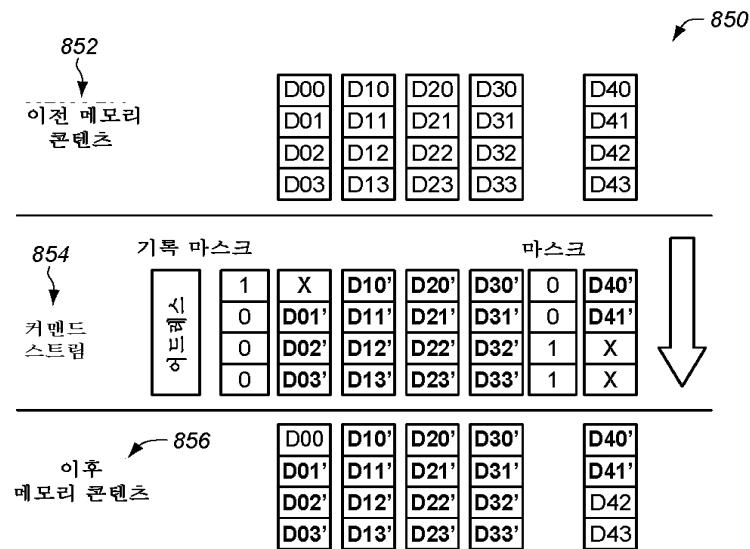
780

778

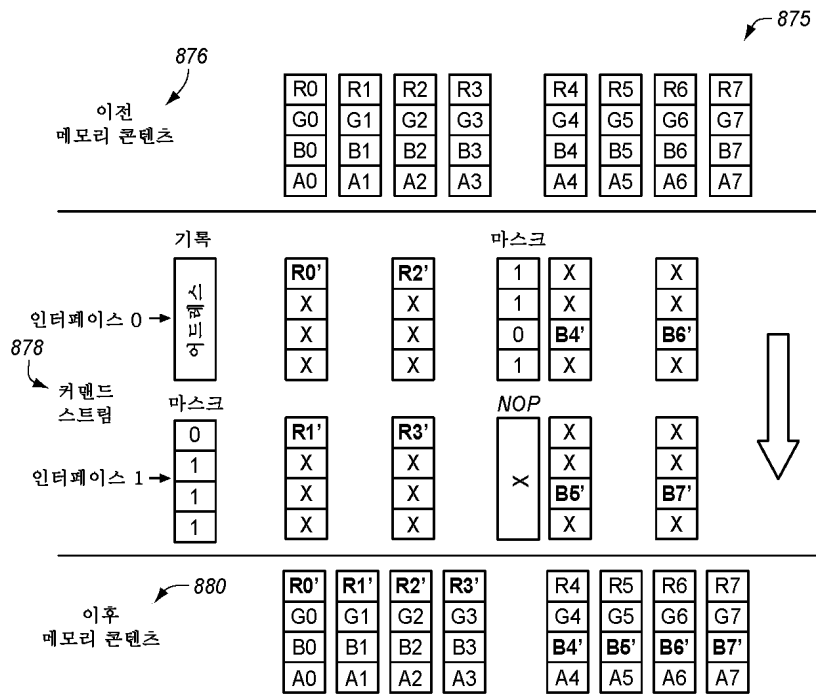
도면8a



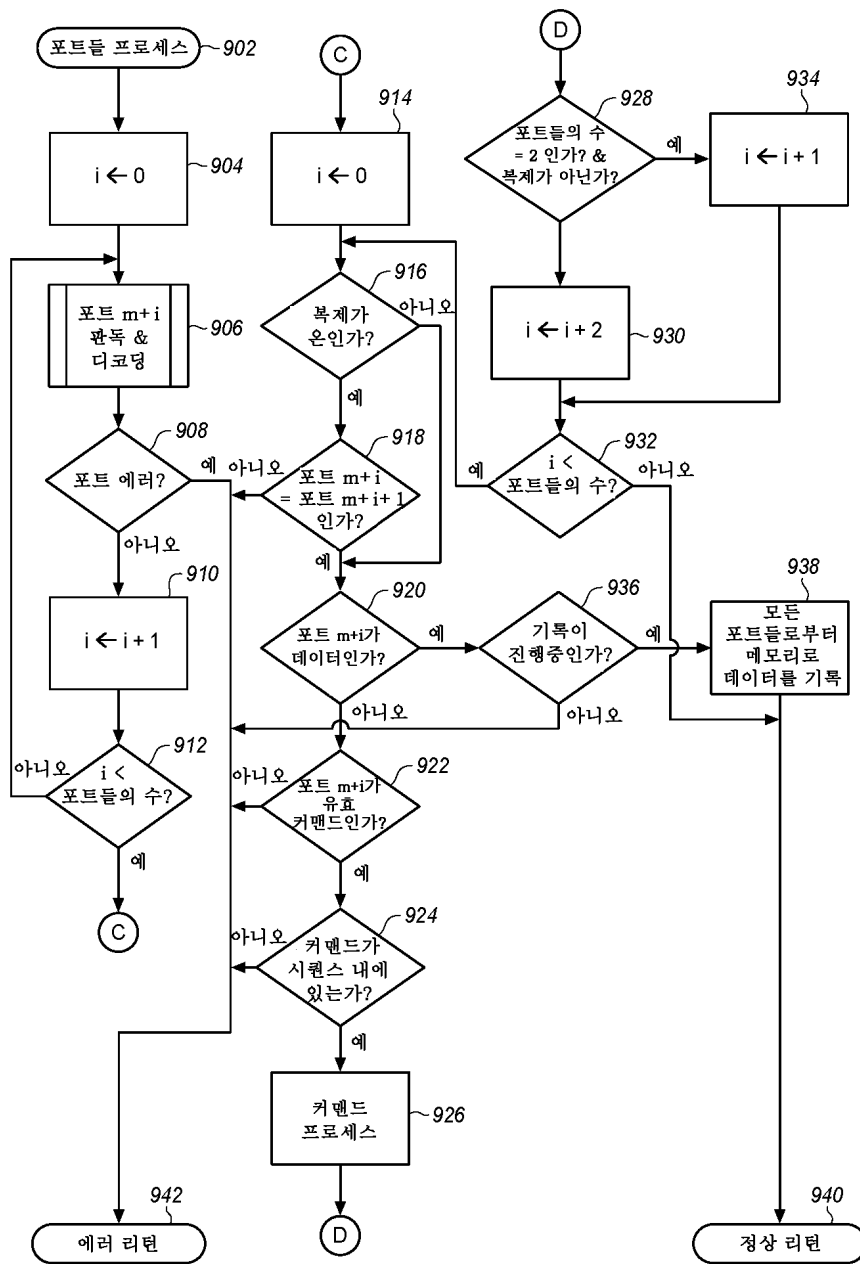
도면8b



도면8c



도면9



도면10

