



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2018년07월18일

(11) 등록번호 10-1879707

(24) 등록일자 2018년07월12일

- (51) 국제특허분류(Int. Cl.)  
*G06F 13/40* (2006.01) *G06F 12/00* (2016.01)  
*G06F 13/42* (2006.01)
- (21) 출원번호 10-2013-7009085
- (22) 출원일자(국제) 2011년09월13일  
 심사청구일자 2016년09월09일
- (85) 번역문제출일자 2013년04월09일
- (65) 공개번호 10-2014-0007331
- (43) 공개일자 2014년01월17일
- (86) 국제출원번호 PCT/US2011/051345
- (87) 국제공개번호 WO 2012/037086  
 국제공개일자 2012년03월22일
- (30) 우선권주장  
 12/910,412 2010년10월22일 미국(US)  
 61/382,089 2010년09월13일 미국(US)
- (56) 선행기술조사문헌  
 KR100403347 B1\*  
 US20090244995 A  
 KR100673904 B1  
 KR1020010085620 A
- \*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
**어드밴스드 마이크로 디바이시스, 인코포레이티드**  
 미국 캘리포니아 95054 산타 클라라 어거스틴 드  
 라이브 2485
- (72) 발명자  
**셜즈 산**  
 미국 78737 텍사스 오스틴 트와일라이트 메사 드  
 라이브 6912
- 험프리즈 니콜라스 디.**  
 미국 78757 텍사스 오스틴 피드몬트 애비뉴 1717  
 (뒷면에 계속)
- (74) 대리인  
**박장원**

전체 청구항 수 : 총 19 항

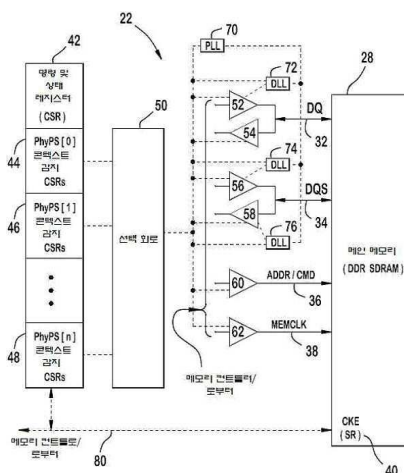
심사관 : 김병균

(54) 발명의 명칭 설정가능한 파워 상태를 갖는 DRAM 파이 인터페이스

(57) 요약

물리 메모리 인터페이스(Phy: 파이) 및 그 작동 방법이 개시된다. 파이 인터페이스는 제 1 파워 콘텍스트 및 제 2 파워 콘텍스트를 수신하도록 구성되는 명령 및 상태 레지스터(CSR)를 포함한다. 선택 회로는 제 1 및 제 2 파워 콘텍스트 사이에서 스위칭하도록 구성된다. 조정가능한 복수의 지연 요소가 제공되고, 각각의 지연 요소는 선택 (뒷면에 계속)

대표도 - 도2



택된 파워 콘텍스트에 대응하는 지연 시간을 갖는다. 설정되는 제 1 세트의 CSR은 제 1 파워 콘텍스트를 저장하도록 구성되고, 설정되는 제 2 세트의 CSR은 제 2 파워 콘텍스트를 저장하도록 구성된다. 파이 인터페이스는 복수의 드라이버를 또한 포함할 수 있고, 각각의 드라이버는 선택된 파워 콘텍스트에 따라 선택가능한 구동 강도를 갖는다. 파이 인터페이스는 복수의 수신기를 또한 포함할 수 있고, 각각의 수신기는 선택된 파워 콘텍스트에 따라 선택가능한 차단 임피던스를 갖는다. 파워 콘텍스트 사이의 스위칭은 하나 이상의 드라이버/수신기의 구동 강도 및/또는 차단 임피던스 조정, 지연 요소 조정으로 나타날 수 있다.

(72) 발명자

**아믹 브라이언 더블유.**

미국 01730 매사추세츠 베드포드 힐크레스트 로드  
4

**리브스 리차드 더블유.**

미국 01581 매사추세츠 웨스트버러 벨크넵 스트리트  
9

**조 한우**

미국 01720 매사추세츠 액튼 포크너 힐 로드 38

**페티존 로날드 엘.**

미국 01742 매사추세츠 콩코드 미니스티리얼 드라이브  
143

## 명세서

### 청구범위

#### 청구항 1

메모리 소자용 물리 메모리 인터페이스를 제어하기 위한 방법에 있어서, 상기 방법은,

제 1 및 제 2 파워 콘텍스트를 저장하는 단계;

상기 메모리 소자로부터 데이터를 읽고 상기 메모리 소자에 데이터를 쓰기 위한 시간 지연을 제공하도록 구성되는, 조정가능한 복수의 지연 요소를 제공하는 단계, 각각의 조정가능한 지연 요소는 제 1 및 제 2 파워 콘텍스트 중 선택된 하나에 대응하는 지연 시간을 가지며;

파워 콘텍스트 변경 요청을 수신하는 단계;

파워 콘텍스트 변경 요청에 기초하여 제 1 및 제 2 파워 콘텍스트 중 하나를 선택하는 단계;

상기 메모리 소자와 관련된 셀프 리프레시 모드를 선택하도록 구성된 셀프 리프레시 출력을 생성하는 단계;

상기 파워 콘텍스트 변경 요청에 기초하여 상기 물리 메모리 인터페이스에 대한 콘텍스트 변경을 개시하는 단계;

상기 셀프 리프레시 모드를 빠져나오는 단계; 및

상기 물리 메모리 인터페이스의 지연 회로를 리로킹(relocking)하는 단계

를 포함하는, 메모리 소자용 물리 메모리 인터페이스 제어 방법.

#### 청구항 2

삭제

#### 청구항 3

제 1 항에 있어서,

상기 제 1 파워 콘텍스트를 저장하도록 구성되는 제 1 세트의 레지스터를 제공하는 단계와,

상기 제 2 파워 콘텍스트를 저장하도록 구성되는 제 2 세트의 레지스터를 제공하는 단계와,

상기 파워 콘텍스트 변경 요청에 따라 상기 제 1 및 제 2 세트의 레지스터 중 하나를 선택하는 단계

를 더 포함하는, 메모리 소자용 물리 메모리 인터페이스 제어 방법.

#### 청구항 4

제 1 항에 있어서,

상기 파워 콘텍스트 변경 요청에 따라 물리 인터페이스의 적어도 하나의 드라이버에 대한 선택가능한 구동 강도를 조정하는 단계를 더 포함하는

메모리 소자용 물리 메모리 인터페이스 제어 방법.

#### 청구항 5

제 1 항에 있어서,

상기 파워 콘텍스트 변경 요청에 따라 물리 인터페이스의 적어도 하나의 수신기에 대한 선택가능한 차단 임피던스를 조정하는 단계를 더 포함하는

메모리 소자용 물리 메모리 인터페이스 제어 방법.

## 청구항 6

제 1 항에 있어서,

제 1 메모리 트레이닝 페이지를 통해 제 1 파워 콘텍스트를 발생시키고 제 2 메모리 트레이닝 페이지를 통해 제 2 파워 콘텍스트를 발생시키는 단계를 더 포함하는

메모리 소자용 물리 메모리 인터페이스 제어 방법.

## 청구항 7

제 1 항에 있어서,

고급 컨피규레이션 및 파워 인터페이스(ACPI) 파워 상태 S3로부터 재개될 때 제 1 및 제 2 파워 콘텍스트 중 적어도 하나를 복원하는 단계를 더 포함하는

메모리 소자용 물리 메모리 인터페이스 제어 방법.

## 청구항 8

제 1 항에 있어서,

사우스 브리지 상의 메모리 위치로부터 제 1 및 제 2 파워 콘텍스트를 불러들이는 단계를 더 포함하는

메모리 소자용 물리 메모리 인터페이스 제어 방법.

## 청구항 9

제 1 항에 있어서,

노스 브리지 상의 메모리 위치로부터 제 1 및 제 2 파워 콘텍스트를 불러들이는 단계를 더 포함하는

메모리 소자용 물리 메모리 인터페이스 제어 방법.

## 청구항 10

메모리 소자용 물리 메모리 인터페이스에 있어서, 상기 물리 메모리 인터페이스는,

제 1 파워 콘텍스트 및 제 2 파워 콘텍스트를 수신하도록 구성되는 복수의 레지스터;

상기 제 1 및 제 2 파워 콘텍스트 중 하나를 선택하도록 구성되는 선택 회로; 및

메모리 소자로부터 데이터를 읽어들이고 메모리 소자에 데이터를 쓰기 위한 타이밍 지원을 제공하도록 구성되는 조정가능한 복수의 지연 요소를 포함하며,

각각의 조정가능한 지연 요소는 제 1 및 제 2 파워 콘텍스트 중 선택된 파워 콘텍스트에 대응하는 지연 시간을 가지며,

상기 물리 메모리 인터페이스는 상기 메모리 소자와 관련된 셀프 리프레시 모드를 선택하도록 구성된 셀프 리프레시 출력을 생성하도록 구성되고,

상기 물리 메모리 인터페이스는 파워 콘텍스트 변경 요청에 기초하여 상기 물리 메모리 인터페이스에 대한 콘텍스트 변경을 개시하도록 구성되고,

상기 물리 메모리 인터페이스는 상기 셀프 리프레시 모드를 빠져나오게 하도록 구성되며, 그리고

상기 물리 메모리 인터페이스는 상기 물리 메모리 인터페이스의 지연 회로를 리로킹하도록 구성되는 것을 특징으로 하는 메모리 소자용 물리 메모리 인터페이스.

## 청구항 11

제 10 항에 있어서,

상기 선택 회로는 메모리 위치로부터 상기 제 1 및 제 2 파워 콘텍스트 중 선택된 파워 콘텍스트를 불러들이는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 12

제 10 항에 있어서,

상기 제 1 파워 콘텍스트를 저장하도록 구성되는 제 1 세트의 레지스터와,

상기 제 2 파워 콘텍스트를 저장하도록 구성되는 제 2 세트의 레지스터

를 더 포함하며, 상기 선택 회로는 상기 제 1 및 제 2 세트의 레지스터 사이에서 선택하도록 구성되는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 13

제 10 항에 있어서,

복수의 드라이버를 더 포함하며,

각각의 드라이버는 상기 제 1 및 제 2 파워 콘텍스트 중 선택된 파워 콘텍스트에 따라 선택가능한 구동 강도를 갖는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 14

제 10 항에 있어서,

복수의 수신기를 더 포함하며,

각각의 수신기는 제 1 및 제 2 파워 콘텍스트 중 선택된 파워 콘텍스트에 따라 선택가능한 차단 임피던스를 갖는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 15

제 10 항에 있어서,

상기 제 1 파워 콘텍스트는 제 1 메모리 트레이닝 페이즈를 통해 결정되고, 상기 제 2 파워 콘텍스트는 제 2 메모리 트레이닝 페이즈를 통해 결정되는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 16

제 10 항에 있어서,

물리 메모리 소자의 복수의 채널에 대한 인터페이스를 더 포함하는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 17

제 10 항에 있어서,

상기 물리 메모리 인터페이스가 중앙 처리 유닛(CPU)의 다이 상에 위치하는

메모리 소자용 물리 메모리 인터페이스.

#### 청구항 18

제 10 항에 있어서,

고급 컨피규레이션 및 파워 인터페이스(ACPI) S3 파워 상태에서부터 재개될 때 상기 제 1 및 제 2 파워 콘텍스트

중 적어도 하나를 불러들이도록 구성되는 메모리 인터페이스를 더 포함하는  
메모리 소자용 물리 메모리 인터페이스.

#### 청구항 19

제 10 항에 있어서,  
메모리 위치로부터 상기 제 1 및 제 2 파워 콘텍스트를 로딩하도록 구성되는 메모리 인터페이스를 더 포함하는  
메모리 소자용 물리 메모리 인터페이스.

#### 청구항 20

제 10 항에 있어서,  
상기 물리 메모리 인터페이스에 연결되고, 그리고, 상기 제 1 및 제 2 파워 콘텍스트 중 선택된 파워 콘텍스트  
와 연계된 타이밍 지연을 이용하여 데이터를 읽고 쓰도록 구성되는, 메모리 소자를 더 포함하는  
메모리 소자용 물리 메모리 인터페이스.

### 발명의 설명

#### 기술 분야

[0001] 본 발명은 DRAM과 직접 인터페이싱하는 물리 계층을 포함하는 메모리 서브시스템에 관한 것이다.

[0002] 관련 출원의 상호 참조

[0003] 본 출원은 2010년 9월 13일자 미국특허가출원 제61/382,089호와, 2010년 10월 22일자 미국특허출원 제 12/910,412호에 대한 우선권을 주장하고, 그 내용은 여기서 완전하게 제시된 것처럼 참고자료로 포함된다.

#### 배경 기술

[0004] 전형적인 메모리 시스템은 메모리 컨트롤러와 메모리 소자 사이에서 데이터를 송신하기 위해 비동기식 또는 동기식 클러킹 기법을 이용한다. 동기식 클러킹은 메모리 소자가 제어 입력에 응답하기 전에 클럭 신호를 기다리고 따라서 컴퓨터의 시스템 버스와 동기화됨을 의미한다. 동기식 DRAM(SDRAM)은 이러한 소자가 통상적으로 비동기식 메모리 소자보다 높은 클럭 속도를 지원하기 때문에 널리 사용되고 있다.

[0005] 더블 데이터 레이트(DDR) SDRAM은 클럭 신호의 상승 및 하강 에지 모두에서 데이터를 전송한다. 이러한 메모리 소자는 낮은 클럭 주파수를 이용하지만, 전기 데이터 및 클럭 신호의 타이밍의 엄격한 제어를 요한다. 이러한 소자의 제 1 버전(DDR1)은 동일 클럭 주파수로 구동되는 단일 데이터 레이트(SDR) SDRAM의 대역폭의 거의 두배를 달성하였다. DDR2 및 DDR3 SDRAM 소자는 DDR1 소자에 이은 개선된 버전이다. 어느 타입의 DDR 메모리(DDR1/DDR2/DDR3)가 사용되는지에 관계없이, 물리 인터페이스(Phy)가 메모리 컨트롤러와 DDR SDRAM 소자 사이에 직접 연결된다. 파이 인터페이스는 일반적으로 DDR SDRAM 데이터 스트로브의 타이밍 요건을 취급하기 위한 회로를 포함한다. 전형적인 파이 인터페이스 구현에는 메모리 성능 레벨 또는 요청 파워를 신속하게 조정하기 위한 메커니즘을 제공하지 않는다.

#### 발명의 내용

[0006] 물리 메모리 인터페이스(Phy: 파이)가 제공된다. 파이는 메모리 컨트롤러와 물리 메모리 소자 사이에서 인터페이스된다. 파이 인터페이스는 제 1 파워 콘텍스트 및 제 2 파워 콘텍스트를 수신하도록 구성되는 명령 및 상태 레지스터(CSR)를 포함한다. 선택 회로가 제공된다. 선택 회로는 제 1 및 제 2 파워 콘텍스트 사이에서 스위칭하도록 구성된다. 파이 인터페이스는 조정가능한 복수의 지연 요소를 또한 포함할 수 있고, 각각의 지연 요소는 선택된 파워 콘텍스트에 대응하는 지연 시간을 갖는다. 파워 콘텍스트 사이의 스위칭은 조정가능한 지연 요소들 중 하나 이상의 조정으로 나타난다.

[0007] 다른 실시예에서, 파이 인터페이스는 제 1 파워 콘텍스트를 저장하도록 구성된 제 1 세트의 CSR과, 제 2 파워 콘텍스트를 저장하도록 구성된 제 2 세트의 CSR을 포함한다. 파이 인터페이스는 복수의 드라이버를 포함하고, 각각의 드라이버는 선택된 파워 콘텍스트에 따라 선택가능한 구동 강도를 갖는다. 파이 인터페이스는 복수의 수

신기를 또한 포함할 수 있고, 각각의 수신기는 선택된 파워 콘텍스트에 따라 선택가능한 차단 임피던스를 갖는다. 파워 콘텍스트 사이의 스위칭은 하나 이상의 드라이버/수신기의 구동 강도 및/또는 차단 임피던스 조정으로 나타날 수 있다.

[0008] 제 1 및 제 2 파워 콘텍스트는 BIOS 트레이닝 과정을 통해 결정될 수 있다. 이러한 과정은 복수의 페이지를 가질 수 있다. 예를 들어, 제 1 파워 콘텍스트는 제 1 메모리 트레이닝 페이지를 통해 결정될 수 있고, 제 2 파워 콘텍스트는 제 2 메모리 트레이닝 페이지를 통해 결정될 수 있다.

[0009] 파이 인터페이스는 CSR에 읽기/쓰기 액세스할 수 있도록 구성되는 컨피규레이션 버스를 포함할 수 있다. 파이 인터페이스는 복수 채널의 물리 메모리 소자를 지원하도록 또한 구성될 수 있다. 파이 인터페이스는 중앙 처리 유닛(CPU)의 다이 상을 포함한 다양한 위치에 놓일 수 있다.

### 도면의 간단한 설명

[0010] 도 1은 전형적인 컴퓨터 시스템을 도시하고,

도 2는 파이의 하이 레벨 구조를 도시하며,

도 3은 파이 인터페이스 컨피규레이션 서브 상의 어드레스, 명령(가령, 읽기, 쓰기) 및 데이터를 보여주는 타이밍도이고,

도 4는 명세 CSR 어드레싱(breakdwon CSR addressing)을 보여준다.

### 발명을 실시하기 위한 구체적인 내용

[0011] 도 1은 중앙 처리 유닛(CPU)(12), 코어 로직 칩셋(14, 16), 및 메인 메모리(28)에 대한 메모리 인터페이스(18)를 포함하는 컴퓨터 시스템(100)의 일부분을 도시한다. 코어 로직 칩셋은 노스 브리지(14)(또는 통합 메모리 컨트롤러)와 사우스 브리지(또는 입/출력 컨트롤러 허브)(16) 사이에서 나누어질 수 있다. 메모리 컨트롤러(18)는 노스 브리지(14)에 자주 위치한다. 이러한 회로는 단일 칩과 같이 다양한 위치에, 또는 CPU에 물리적으로 위치할 수 있다. 메모리 컨트롤러(18)는 일반적으로, 메인 메모리(30) 내외로 드나드는 데이터의 논리적 흐름을 관리한다. DRAM은 다양한 메모리 스토리지 셀 내에 전하를 유지하기 위해 주기적인 리프레시 신호를 또한 요구한다. 메모리 컨트롤러(18)는 통상적으로 이러한 리프레시 작동을 관리한다. CPU 구조 및/또는 운영 체제에 따라, 메인 메모리(30)는 32 또는 64 비트 데이터 유닛으로 데이터를 전송할 수 있다. 다른 크기의 데이터 유닛도 지원될 수 있다. 일부 메모리 시스템은 블록(20, 24, 30)에 의해 반영되는 바와 같이 복수의 채널(가령, 2개 이상의 독립형 메모리 컨트롤러)을 포함한다. 명료성을 위해, 여기서 제시되는 예는 단일 메모리 채널만을 이용할 수 있다. 본 개시내용의 범위로부터 벗어나지 않으면서 복수의 채널이 사용될 수 있다.

[0012] 파이 인터페이스(22)는 메모리 컨트롤러(18)와 물리 메모리 소자 사이에 놓인다. 파이 인터페이스는 통상적으로 중앙 처리 유닛에 위치하지만, 그외 다른 곳에 위치할 수도 있다. 명료성을 위해, 파이 인터페이스(22)는 도 1에 별도의 블록으로 도시된다. 파이 인터페이스는 메인 메모리 데이터 스트로브의 타이밍 요건을 취급하기 위한 회로를 일반적으로 포함한다. 이 개시내용의 용도를 위해, 더블 데이터 레이트(DDR)라는 용어는 DDR1/DDR2/DDR3 및/또는 이러한 메모리 소자의 후속 버전을 포괄한다. DDR 메모리 소자는 통상적으로, JEDEC(Joint Electron Devices Engineering Councils) 표준을 따른다. 이러한 표준은 DDR 메모리 데이터가 어떻게 액세스되고(읽히고) 저장되는지(쓰여지는지)를 규정한다. 파이 인터페이스는 다른 메모리 타입 및/또는 다른 메모리 표준과 함께 작동하도록 구성될 수 있다. 전형적인 DDR SDRAM 메모리 소자에 대한 인터페이스는 주로 2개의 신호 클래스 DQ(데이터)(32) 및 DQS(데이터 스트로브)(34)를 통해 달성된다. 다른 신호는 메모리 클럭(MEMCLK)(38)과, 어드레스 및 명령 신호(일반적으로 ADDR/CMD(36)으로 도시됨)를 포함한다. 전형적인 메모리 소자는 도면에 도시되지 않은 추가적인 신호들을 이용할 수 있다. 이러한 신호는 명료성을 위해 생략되었다. 예를 들어, 파워 및 접지 신호가 도시되지 않는다. 이러한 신호는 전형적인 구현예에 포함될 것이다.

[0013] 읽기 작동 중, DDR SDRAM은 DQ 및 DQS를 흔히 "에지 정렬"로 불리는 방식으로 동시에 발급한다. 메모리 컨트롤러가 DDR SDRAM으로부터 전송되는 데이터를 정확하게 획득하기 위해, 파이 인터페이스(22)는 지연-동기 루프(DLL)와 같은 지연 회로를 이용하여 DQS 신호를 지연시켜서, 유효 데이터 윈도우 또는 "데이터 아이"(data eye) 중 DQ 신호를 정확하게 래칭할 수 있게 된다. 마찬가지로, 파이 인터페이스(22)는 DDR DRAM에 대한 데이터 쓰기를 지원하기 위해 지연 회로를 또한 이용한다. 데이터를 읽기 위해, DQS(34)가 지연되어야만 한다. 데이터를 쓰기 위해, DQS 및 DQ(34, 32)가 지연되어야만 한다. 파이는 DQS(34)를 에지 정렬보다는 DQ(32) 데이터 아이의 중

간과 정렬시킨다. DQS(34)는 쓰기 레벨링(write leveling)을 위해, 그리고, 데이터 아이 중간 요건(middle-of-data-eye requirements)을 충족시키기 위해, 지연된다. 다른 지연이 (예를 들어, 데이터 버스의 읽기/쓰기 3-상태(tri-state) 제어를 위해) 또한 사용될 수 있다. 아래에서 더욱 상세하게 설명되는 바와 같이 지연 타이밍, 구동 강도, 및 다양한 다른 파라미터를 제어하는 데 사용되는 복수의 명령 및 상태 레지스터(CSR)(42)를 포함한다. 이러한 회로가 채널 당 원칙으로 또한 복제될 수 있다.

[0014] 파이 인터페이스는 또한 송신기 구동 강도 및 수신기 차단 임피던스를 조정 또는 선택할 수 있다. 고정된 타이밍 지연, 송신기 구동 강도, 및 수신기 차단 임피던스를 이용하기보다는, 이러한 파라미터는 컴퓨터 시스템이 켜질 때마다 조정될 수 있다. 이는 통상적으로 트레이닝 프로그램의 도움으로 통상적으로 달성된다. 트레이닝 프로그램은 통상적으로 기본 입/출력 시스템(BIOS) 메모리 소자(26)에 저장되지만, 소자 하드웨어 내에서 구현될 수도 있다. 트레이닝 프로그램은, 메모리 인터페이스 신호들 중 많은 신호들과 상관된 적절한 타이밍 지연, 구동 강도, 및 차단 임피던스를 결정하는, 파워-온 셀프-테스트(POST) 중 알고리즘을 실행한다. 이러한 파라미터는 파이 내외로 드나드는 다양한 신호 경로의 전체 타이밍을 규정하는 복수의 레지스터 내 파이 인터페이스 내에 저장된다. 대안으로서, 이러한 파라미터는 그외 다른 곳에(가령, 노스 브리지(14) 또는 사우스 브리지(16)에) 저장될 수 있다.

[0015] 도 2는 파이 인터페이스(22)의 하이 레벨 구조를 도시한다. 파이 인터페이스는 일반적으로, 명령 및 상태 레지스터(CSR)(42)를 포함하는 제어 인터페이스 및 클럭 소스(가령, PLL(70))를 포함한다. 파이 인터페이스(20)는 DQ(32), DQS(34), 어드레스/명령 라인(36), 및 클럭 입력(38)을 포함하는 듀얼 인-라인 메모리 모듈(DIMM)과 같이, 전형적인 메모리 소자 상의 주종의 핀들에 대한 물리적 연결을 제공한다.

[0016] 전형적인 메모리 소자에는 클럭 이네이블(clock enable)(셀프 리프레시) 입력(40)이 또한 제공된다. 클럭 이네이블 입력(40)은 셀프 리프레시 모드로 메모리 소자를 배치하는 데 사용된다. 이 모드에서, 메모리 소자는 온-칩 타이머를 이용하여 내부 리프레시 사이클을 필요에 따라 발생시킨다. 외부 클럭은 또한 이 시간동안 중단될 수 있다. 이러한 입력은 메인 메모리 데이터의 손실없이 메모리 컨트롤러를 디스에이블(disable)시킬 수 있기 때문에 통상적으로 파워 다운 모드와 연계하여 사용된다.

[0017] 도 2에 도시되는 바와 같이, DQ(32)(데이터) 및 DQS(34)(스트로브) 라인은 양방향성이다. 각각의 DIMM은 복수의 DQ 라인(가령, 64 데이터 비트) 및 복수의 DQS 라인을 가질 것이다. 이러한 라인들 각각은 파이 인터페이스(22) 내에 관련 드라이버(52, 56) 및 수신기(54, 58)를 갖는다. 단일 드라이버/수신기 쌍이 도 2에 명료성을 위해 도시된다. 각각의 드라이버는 선택가능한 또는 조정가능한 구동 출력을 갖는다. 마찬가지로, 각각의 수신기는 (가령, 다이 터미네이션 상에서) 선택가능한 또는 조정가능한 차단 임피던스를 갖는다. DQ 및 DQS 라인(32, 34)은 지연 동기 루프(DLL)(72, 74, 76)와 같은 지연 요소와 또한 연계된다. 지연 요소(가령, 70-76), 구동 강도, 또는 차단 임피던스 값에 대한 조정은, 아래에서 더욱 상세하게 설명되는 바와 같이, 관련 CSR(42)의 적정 필드 내로 적절한 값으로 프로그래밍함으로써 달성된다. 지연 요소, 구동 강도, 또는 차단 임피던스의 조정과 CSR 사이의 논리적 연결은 대체로 점선으로 도시된다.

[0018] 본 예에서, 드라이버(52, 56)는 각각 DLL(72, 74)과 연계된다. 수신기(58)는 DLL(76)과 연계된다. 상술한 바와 같이, DLL은 읽기 및 쓰기 작동에 적절한 타이밍 지연을 제공하도록 조정된다. 파이 인터페이스는 레벨링과 함께, 또는 레벨링없이, 읽기 및 쓰기 작동을 실행하도록 또한 구성될 수 있다. 레벨링과 함께 DDR3 DIMM에 대한 메모리 쓰기 작동 중, 파이 인터페이스는 DIMM으로 들어가는 각각의 DQS의 론치(launch)를 지연시켜서, 각각의 DRAM 칩에서 DQS가 메모리 클럭(58)과 합쳐지는 것으로 보이게 된다. 레벨링과 함께 읽기 작동 중, 파이 인터페이스는 플라이-바이토폴로지(fly-by topology)에 의해 야기되는 지연을 또한 보상할 수 있다.

[0019] 높은 데이터 속도에서 작동의 신호 무결성 문제로 인해, 파이 인터페이스는 버스트 단위(또는 거래별) 원칙으로 DLL 세팅을 동적으로 변화시킬 수 있다. 파이 인터페이스는 시스템 내 각각의 DIMM에 대한 DQ 및 DQS의 지연 세팅의 최적 튜플(optimum tuple)을 저장할 수 있다. DIMM의 액세스에 따라, 파이 인터페이스는 적절한 DLL 세팅을 불러들여, 이를 적용한다.

[0020] 파이 인터페이스는 요망 레벨의 성능에 기초하여 요청받은 파워를 재단할 수 있다. 파워 컨텍스트를 변경할 시기에 관한 결정은 다양한 소스로부터 나타날 수 있다. 예를 들어, 운영체제는 (가령, 세팅된 비활동 주기 이후, 사용자 명령에 의해, 시간 스케줄에 의해, 등에 의해) 컨텍스트 변화가 요망됨을 결정할 수 있다. 대안으로서, 컨텍스트 변화가 요망될 때를 결정하기 위해 하드웨어가 사용될 수 있다. 컨텍스트 변화는 서로 다른 파워 상태와 관련된, 서로 다른 파이 인터페이스 파라미터 세트 사이에서의 스위칭에 의해 달성된다. 예를 들어, 높은 파워 상태(가령, 높은 메모리 속도) 및 낮은 파워 상태(가령, 낮은 메모리 속도). 각각의 파워 상태는 관련된 파



이 인터페이스 파라미터 또는 컨텍스트 세트를 갖는다(즉, 각각의 신호 라인에 대한 지연 요소 세팅, 구동 강도, 및 차단 임피던스). 이후 더욱 상세히 설명되는 바와 같이, 파워 상태 사이의 스위칭은 여러 가지 방식으로 달성될 수 있다. 여기서 개시되는 바와 같이 복수의 파워 상태 사이의 스위칭은 임의의 메모리 타입에 적용될 수 있고, 아래 예시에 사용되는 바와 같이 DDR 메모리와 함께 이용하는 데 제한되지 않는다.

[0021] 본 예에서, 메모리 컨트롤러(18)는 32비트, 시간-인터리빙된, 일방향 컨피규레이션 버스(80)를 통해 CSR(42)에 액세스할 수 있다. 도 3에 도시되는 바와 같이, 어드레스 및 명령(가령, 읽기, 쓰기, 두 나중(do nothing))은 제 1 파이프스테이지로 보내지고, 이어서 데이터가 제 2 파이프스테이지로 전달된다. CSR은 다른 통신 링크를 통해 액세스될 수 있다.

[0022] 본 예에서, CSR 어드레스 공간은 16비트 폭이어서, 65,536개의 고유 16-비트 레지스터의 공간을 가능하게 한다. 이러한 큰 공간을 허용하는 대신에, 어드레스가 매핑되어 다음의 기능 - 칩셋 식별, 인트라-칩셋 송출, 보상 송출, 칩셋 인스턴스 식별을 가능하게 한다(D3DBYTE, D3CLK, D3CMP는 두 번 이상 배치되는 칩셋임). 도 4는 CSR 어드레싱의 명세(breakdown)를 보여준다.

[0023] CSR의 일부분만이, 주어진 파워 상태에 관련된 값들을 지닌다. 파워 상태 사이의 낮은 대기시간 스위칭을 촉진시키기 위해, 한 세트의 파워 컨텍스트 감지 CSR이 각각의 파워 상태에 대해 제공된다. 도 2로 되돌아가면, 제 1 세트의 CSR이 제 1 파워 컨텍스트 - PhyPS[0](44)와 연계된다. 다른 한 세트의 CSR은 제 2 파워 컨텍스트 - PhyPS[1](46)에 연계된다. PhyPS[n](48)에 의해 도시되는 바와 같이 3개 이상의 파워 컨텍스트를 지원하기 위해 추가적인 세트의 CSR이 제공될 수 있다. 서로 다른 세트의 CSR 사이에서 선택을 위해 멀티플렉서 또는 선택 회로(50)가 제공된다. 파이 인터페이스는 파워 컨텍스트에 민감하지 않은 CSR을 또한 지닌다. 이러한 CSR은 모든 파워 상태에 적용가능하다.

[0024] 각각의 PhyPS 내 각각의 CSR에 대한 프로그래머블 필드의 요약이 아래 표1에서 제시된다.

표 1

[0025]

번호	설명
1	DDR PLL 레이트
2	DLL(지연 동기 루프) 주파수-특정 프로그래밍
3	DQS(스트로브 비트) 구동 강도(DS)
4	DQ(데이터 비트) DS
5	MemClk DS
6	어드레스 및 명령(Addr, Bank, RAS, CAS, WE, AddrParity) DS
7	칩 선택(Cs) 및 온-다이 차단(Odt) DS
8	클럭 이네이블(Cke) DS
9	ProcOdt 강도
10	ProcOdtOff
11	바이트당, DIMM 당, 채널 당 TxDqsD11 제어
12	바이트당, DIMM 당, 채널 당 TxDqD11 제어
13	바이트당, DIMM 당, 채널 당 RxEnD11 제어
14	(x4 DIMM 타입에 사용되는) 니블 당 DIMM당 채널 당 RxDqsLD11 제어
15	(x4 DIMM 타입에 사용되는) 니블 당 DIMM당 채널 당 RxDqsUD11 제어
16	채널 당 AddrCmdD11 제어
17	채널 당 CsOdtD11 제어
18	채널 당 CkeD11 제어
19	DIMM 쌍 당 채널 당 MemClkD11 제어

[0026] 본 예에서, 파이 인터페이스의 관련 필드의 프로그래밍은 컨피규레이션 버스(80)를 통해 CSR의 특정 필드를 프로그래밍하거나 명령을 발급함으로써 달성된다. 예를 들어, 하나의 PhyPS로부터 다른 하나의 PhyPS로 변화하는 것은 DDR 파이 인터페이스 간접 레지스터 공간에 발급된 단일 명령을 이용하여 달성될 수 있다. PhyPS 컨텍스트 감지 CSR의 프로그래밍은 적절한 PhyPS 컨텍스트를 세팅함으로써, 그리고 그 후 정규 간접 CSR 쓰기 또는 읽기를 수행함으로써, 달성될 수 있다. 대안으로서, 직접 CSR 쓰기 또는 읽기가 PhyPS 컨텍스트와 관련없이 임의의 CSR에 도달할 수 있다.

[0027] 파이 인터페이스는 Master\_0x08[12]PhyPS, Master\_0x08[8] - PstateToAccess 및 Master\_0x18[8] -

PhyPSMasterChannel 을 포함한 일련의 명령을 통해 제어될 수 있다. [12], [8], 등의 표기는 명령 내의 비트 위치를 언급한다. Master\_0x08[12] 명령은 현재의 파이 인터페이스 P-상태(0 또는 1)에 대응한다. 이 명령은 파워 콘텍스트(어느 CSR 세트)가 현재 액티브한지를 제어한다. Master\_0x08[8] 명령은 CSR 액세스 중 읽거나 쓸 P-상태를 선택한다. BIOS는 실제 P-상태 변화를 할 필요없이 쓰기를 해야할 P-상태를 제어하기 위해 이 방법을 이용할 수 있다. 3개 이상의 파워 콘텍스트를 지원하기 위해 추가적인 비트가 추가될 수 있다. Master\_0x18[8] 명령이 마스터 채널을 선택한다. 본 실시예에서, 이 비트에 의해 지정되는 채널(마스터 채널)만이 0x0B[PhyPS 변화] 명령을 발급하도록 기능한다. 다른 채널로부터 발급된 0x0B[PhyPS 변화] 명령은 무시될 것이다. 3개 이상의 채널을 지원하기 위해 추가적인 비트가 추가될 수 있다. 본 실시예에서, 다음의 0x0B 명령이 규정된다:

표 2

번호	설명
1	PhyPS 변화
2	PhySR 입/출 (셀프 리프레시)
3	DDR 레이트 변화(즉, DDR PLL 주파수 변화)
4	(서브 명령을 통해) 다른 파이 인터페이스 서브모드 업데이트
5	~PwrOk 디폴트 세팅

풀 0x0B 데이터 패킷이 아래 표 3에 도시된다.

표 3

비트 위치	설명
31	PhySRRequest (셀프 리프레시)
30	PhyPSRequest
29-27	예비
26	PhyPS = <요망 파이 인터페이스 파워 상태> PhyPS 0 또는 1과 저파워 또는 고파워 모드(각각 낮은 DDR 레이트 또는 높은 DDR 레이트)의 상관은, 파이의 관점으로부터 임의적이다. 본 예에서, PhyPS1은 저파워 상태로 규정되고 PhyPS0는 고파워 상태로 규정된다.
25-24	PhySR P11 제어
23	PhySR
22-18	예비
17-16	DdrRate
15-4	예비
3	DdrRateRequest
2-0	DdrRate

본 예에서, 파워 콘텍스트는 단일 비트(즉, PhyPS[0] 및 PhyPS[1] - 비트 위치(26))를 통해 선택된다. 3개 이상의 콘텍스트를 지원하기 위해 추가 비트가 추가될 수 있다. PhyPSRequest 비트(비트 위치(30))를 이용하여, 명령이 콘텍스트 변화를 포함함을 표시할 수 있다. 파워 콘텍스트는 일반적으로 다음과 같이 변경될 수 있다. 콘텍스트 변화 요청의 수신 시, 각각의 액티브 채널은 셀프 리프레시(SR) 모드에 놓인다(0x0B[PhySR=1]). 본 "안전 모드"에서, 메모리 소자는 온-칩 타이머를 이용하여 필요에 따라 내부 리프레시 사이클을 발생시킬 수 있다. 선택된 SR 모드에 따라, 외부 클럭이 이 시간 중 중지될 수 있다. SR 모드로의 진입은 서로 다른 시기에 이루어질 수 있지만, 모든 채널은 콘텍스트 변화 전에 SR 모드에 있어야 한다. 파워 콘텍스트 변화가 개시된다(0x0B[PhyPS=X], [PhyPSRequest=1]). 각각의 액티브 채널은 SR 모드로부터 정규 모드로 스위칭된다(0x0B[PhySR=0]).

콘텍스트 변화 명령 수신 시, 파이 인터페이스는 PhyPS 콘텍스트를 변경시킬 것이고, DDR PLL 멀티플라이어 및 디바이더는 업데이트될 것이며, PLL은 리로킹(relocking)될 것이다. 이 모두가 완료될 때, CfgDone이 세팅될 것이다. 이어서 SR 모드를 빠져나올 때, DLL이 리로킹될 것이다. PhyPS 변화가 발생하고 있는 동안, 메모리 컨트롤러는 파이 인터페이스에 대한 모든 입력에 대해 제어를 유지한다(구체적으로, CKE, MemReset, ReadPending, WritePending, 모든 다른 3-상태 제어).

PhyPS를 가능한 빨리 변경시키는 것이 의도이기 때문에 PhyPS 변경은 PLL 파워 업을 남길 것으로 대체로 예상된

다. 따라서, PhyPS 변경은 PLL 리로킹을 즉시 트리거링할 것이고(그리고 리로킹을 대기), DLL 리로킹은 SR을 빠져나간 후 일어난다. 그러나, PLL(VCO 또는 레귤레이터)을 파워 다운시킨 SR 모드에 놓이는 것이 가능하고, 그리고, 본 SR 모드에 있을 때 PhyPS를 변경하는 것도 가능하다. 이 경우에, 파이 인터페이스는 PhyPS 변경 직후 PLL 리로킹 시간을 여전히 기다릴 것이지만, 후속 SR 탈출 후, 파이 인터페이스는 (PLL이 파워 백업 프로세스에 있을 것이기 때문에) PLL 리로킹 및 DLL 리로킹 모두를 기다릴 것이다.

[0034] 파이 인터페이스 셀프 리프레시(PhySR)의 입/출을 위한 0x0B 명령은 채널 0 또는 1로 발급될 수 있고, 하나의 DRAM 채널을 SR 모드로, 다른 하나는 SR 모드가 아니도록 하는 것이 가능하다. 파이 내 PC1k 전역 그리드를 완전히 파워 다운시키기 위해, 채널 0 과 채널 1 모두가 PhySR 모드에 있어야 한다. PhySR 변경을 위한 0x0B가 아래 표 4에 도시된다:

표 4

비트 위치	설명
31	PhySRRequest (셀프 리프레시)
30	PhyPSRequest = 0
29-27	해당 무
26	PhyPS = <요망 파이 인터페이스 파워 상태> = X
25-24	PhySR P11 제어 {정규 PD, VCO PD}
23	PhySR = 1 (PhySR 진입시), 0 (나올때)
22-0	해당 무

[0036] 0x0B[23=PhySR]과 함께 0x0B[31=PhySRRequest]을 세팅하면, 파이 인터페이스가 셀프 리프레시에 진입하거나 셀프 리프레시를 빠져나올 수 있다. 0x0B[25:24]는 SR 모드에 있을 때 PLL의 거동을 제어한다. SR 모드로의 진입은 매우 빨라서, 0x0B 명령의 수신으로부터 CfgDone 시간까지 ~300ns 소요가 주장된다. 메모리 컨트롤러를 가능한 빨리 순방향으로 이동시키기 위해, 메모리 컨트롤러는 1로부터 0까지 천이에 대해 CfgDone을 모니터링할 수 있다. 이는, 파이 인터페이스가 메모리 컨트롤러로부터 입력을 차단하여, 모든 CKE를 로우 상태로 구동하고, MemReset을 적절히 구동시키며, 모든 나머지 DDR 버스 핀을 3-상태로 배치함을 표시한다. 이 시점에서, 메모리 컨트롤러가 미치는 것(go insane)이 안전하다. 필요한 모든 CSR 거래가 완료됨을 표시하는 CfgDone이 주장될 때까지 Vddr을 강하시키는 것은 안전하지 않다.

[0037] PhySR로부터 빠져나오는 것은 훨씬 더 잠재적인 사항으로서, 파이 인터페이스 클럭 그리드(PC1k)가 다시 턴온되고 DLL이 리로킹됨을 요구한다. PLL은 SR 모드 중 온 상태로 남겨나 턴오프될 수 있다. 본 실시예에서, 0x0B[30=PhyPSRequest]는 PhySR 변경을 실행할 때 0이어야 한다. 0x0B[31 = PhySRRequest] 또는 0x0B[30 = PhyPSRequest]를 세팅하면, 0x0B[22:0]이 디스에이블되어, DDR 레이트 필드를 업데이트하기 위해 0x0B[3 = DdrRateRequest]를 세팅하는 것이 불가능함을 의미한다. DDR 레이트는 파이 인터페이스 PLL 멀티플라이어 및 디바이더를 지닌 CSR을 먼저 로딩하도록 세팅된다. 0x0B[PhyPS 변경] 전송은 DDR PLL 멀티플라이어 및 디바이더를 로딩시킬 것이다. 다른 실시예에서, 한번에 복수의 하이 레벨 0x0B 명령을 실행하는 것이 가능할 수 있다. 0x0B[31 = PhySRRequest] = 1 이고 0x0B[23 = PhySR]가PhySR 상태에 어떤 변화도 주지 않을 경우, DLL 로킹 시간은 CfgDone 주장 이전에 여전히 준수된다.

[0038] 본 실시예에서, DDR 버스 상의 DRAM 데이터 레이트에 대한 0x0B 명령은 채널 0에만 BIOS에 의해 세팅된다. 채널 1에서 DRAM 데이터 레이트를 세팅하기 위해 0x0B 명령을 전송하는 것은 어떤 효과도 없다. 다른 실시예에서는 각각의 채널 상에서 독립적인 DDR 데이터 레이트를 지원할 수 있다. 0x0B DDR 레이트 명령이 아래 표 5에 도시된다:

표 5

비트 위치	설명
31	PhySRRequest = 0
30	PhyPSRequest = 0
29-27	예비
26	PhyPS = <요망 파이 인터페이스 파워 상태> = X
25-24	PhySR P11 제어 = X
23	PhySR = X
22-18	예비

17-16	DdrRate [4:3]
15-4	예비
3	DdrRateRequest = 1
2-0	DdrRate [2:0] = 요망 레이트로 세팅

[0040] 이러한 0x0B[DdrRate] 명령을 통한 실제 DDR 레이트 및 PLL 주파수의 변화가 레거시 BIOS용으로 포함된다. 파이 인터페이스 P-상태를 이해하는 BIOS는 대신에, 직접 CSR Master\_0x00[DdrRate] 및 Master\_0x40[DdrRate]를 통한 두 파이 인터페이스 P-상태 모두에 대한 DDR 레이트를 프로그래밍하여야 한다. (직접 CSR을 통한) 이러한 레이트 변화는 후속 0x0B[PhyPS 변화] 이후에만 영향을 끼칠 것이다.

[0041] 본 실시예에서, 0x0B[31 = PhySRRequest] 및 0x0B[30 = PhyPSRequest]는 DDR 레이트에 대한 업데이트를 실행할 때 모두 0이어야 한다. 실시예는 한번에 복수의 하이 레벨 0x0B 명령을 지원할 수 있다.

[0042] 콜드 리셋(cold reset), 워م 리셋(warm reset) 후의 PhyPS 상태, 또는 고급 컨피규레이션 및 파워 인터페이스 (ACPI) 파워 상태 S3(흔히, 스탠바이, 슬립, 또는 RAM에 서스펜드로 불림)가 아래 표 6에 도시된다. 모든 상태들은 워م 리셋을 통해 지속된다. 따라서, 아래의 워م 리셋 엔트리에서, 값들은 워م 리셋 이전인 것처럼 유지된다.

표 6

채널 분포	이벤트	두 PhyPS 모두가 지원될 경우 디폴트	하나의 PhyPS만이 지원될 경우 디폴트
채널 0 만	콜드 리셋	PhyPS = 0	PhyPS = 0
		PStateToAccess = 0	PStateToAccess = 0
		PhyPSMasterChannel = 0	PhyPSMasterChannel = 0
		PhySR = 0 (모든 채널)	PhySR = 0 (모든 채널)
	워م 리셋	PhyPS = 0 또는 1	PhyPS = 0
		PStateToAccess = 0 또는 1	PStateToAccess = 0
		PhyPSMasterChannel = 0	PhyPSMasterChannel = 0
		PhySR = 0 또는 1 (채널 0)	PhySR = 0 또는 1 (채널 0)
		PhySR = 1 (채널 1)	PhySR = 1 (채널 1)
채널 1 만	콜드 리셋	PhyPS = 0	PhyPS = 0
		PStateToAccess = 0	PStateToAccess = 0
		PhyPSMasterChannel = 0	PhyPSMasterChannel = 0
		PhySR = 0 (모든 채널)	PhySR = 0 (모든 채널)
	워م 리셋	PhyPS = 0 또는 1	PhyPS = 0
		PStateToAccess = 0 또는 1	PStateToAccess = 0
		PhyPSMasterChannel = 0	PhyPSMasterChannel = 0
		PhySR = 1 (채널 0)	PhySR = 1 (채널 0)
		PhySR = 0 or 1 (채널 0)	PhySR = 0 or 1 (채널 0)
두 채널 모두	콜드 리셋	PhyPS = 0	PhyPS = 0
		PStateToAccess = 0	PStateToAccess = 0
		PhyPSMasterChannel = 0	PhyPSMasterChannel = 0
		PhySR = 0 (모든 채널)	PhySR = 0 (모든 채널)
	워م 리셋	PhyPS = 0 또는 1	PhyPS = 0
		PStateToAccess = 0 또는 1	PStateToAccess = 0
		PhyPSMasterChannel = 0	PhyPSMasterChannel = 0
		PhySR = 0 또는 1 (채널 0)	PhySR = 0 또는 1 (채널 0)
		PhySR = 1 (채널 1)	PhySR = 1 (채널 1)
	S3	콜드 리셋과 동일	콜드 리셋과 동일

[0044] 전형적인 컴퓨터 전력 공급원이 먼저 여기될 때, 다양한 전압을 안정화시키는 데 소정의 시간이 걸린다. 전압이 안정화되기 전에, 컴퓨터가 부팅 시도될 경우, 예측불가능한 결과가 발생할 수 있다. 컴퓨터가 조기에 스타트업되는 것을 방지하기 위해, 전력 공급원이 사용 대기 상태가 될 때 전력 공급원은 PwrOK 신호를 출력한다. 이 신호가 전송될 때까지, 마더보드는 컴퓨터의 스타트업을 거부할 것이다.

[0045] 콜드 부팅시(~PwrOK 주장 및 Reset 주장됨), PhyPS는 자동적으로 0으로 세팅될 것이고, 파이 인터페이스는 PhySR 상태에 있지 않을 것이다(모든 DLL이 파워업될 것이다). 워 부팅시(PwrOk가 연속적으로 주장되고 Reset 만 이 주장됨), PhyPS 및 PhySR 상태는 워 리셋 직전의 상태에 의해 결정될 것이다. 워 리셋 상태로부터의 재개는 BIOS로 하여금 적절한 PhyPS 및 PhySR 상태를 세팅할 것을 요구한다(시간 최적화 솔루션은 이 상태들의 직접 CSR 읽기를 수행하고, 그 후 필요할 경우 쓰기를 수행하는 것이다). BIOS는 파이 인터페이스가 워 리셋 이후 SR 모드에 있지 않음을 보장하여야 한다.

[0046] 쓰기 레벨화(WL) 및 RxEn의 명확한 수렴을 갖기 위해 DDR3가 두 패스 트레이닝 과정을 요하기 때문에, 하이 파워 P상태에 대한 트레이닝 프로세스의 일부분으로 저주파수 P상태동안 트레이닝이 가능할 수 있다. 이 과정은 워 리셋 또는 S3 엑시트에 반해, 콜드 리셋임(따라서 트레이닝을 필요로 함을) BIOS가 이미 결정하였다고 또한 가정한다.

[0047] 메모리 트레이닝의 제 1 패스는, 아래 표 7에 도시되는 바와 같이 명백한 트레이닝에 요구되는 초기 DDR 레이트를 이용한다. 아래 표 전체에 제시되는 특정 순서의 단계들은 본 개시내용의 범위로부터 벗어나지 않으면서 변경될 수 있다.

표 7

[0048]

	설명
1	메모리를 이용하여 채널에 대한 다음의 과정을 수행한다. 메모리없이 임의의 채널을 SR로 배치하고 이 채널에 대해 아무것도 하지 않는다.
2	관련 메모리 컨트롤러 레지스터 공간을 구성한다.
3	모든 액티브 채널을 SR로 배치한다(선택사항).
4	DDR 레이트를 목표 레이트로 프로그래밍한다.
5	모든 액티브 채널을 정규 모드로 세팅한다(즉, 채널을 SR로부터 빼낸다). DLL이 리로킹될 것이다.

[0049] 이 시점에서, DDR PLL 및 DLL은 사용을 위해 적절하게 구성된다. PhyPS 콘텍스트는 PhyPS0이다. 두 채널 모두, 메모리가 존재할 경우, 트레이닝 시작 준비가 되어 있다. 트레이닝은, 표 8에 일반적으로 도시되는 바와 같이, AGESA(AMD Generic Encapsulated Software Architecture) 부트스트랩 프로토콜과 같은 공지된 프로토콜을 따른다:

표 8

[0050]

	설명
6	모든 구동 강도 필드 세팅.
7	NB_TxChnClkEn을 세팅하고, 하드웨어-기반 펜스 트레이닝을 수행하며, 최종 FenceValue를 표시 및 세팅.
8	Address/Bank/RAS/CAS/WE/MemClk D11 론치 지연을 세팅.
9	Cs 및 Odt D11 론치 지연을 세팅.
10	Cke D11 론치 지연을 세팅.
11	DRAM init 수행.
12	디폴트 상수를 이용하여 하드웨어-기반 WL 트레이닝 수행.
13	디폴트 상수를 이용하여 하드웨어-기반 RxEn 트레이닝 수행.
14	TxDq/RxDqs 쓰기/읽기 트레이닝 수행.
15	MRL 트레이닝이 여기서 발생할 수도 있고, 나중에 발생할 수도 있음. NCLK는 이러한 DDR 레이트를 위해 결과적인 목표 속도로 구동되어야 함.

[0051] 이 시점에서, 명백한 트레이닝을 위한 초기 DDR 레이트의 트레이닝 및 프로그래밍이 완료된다. 이러한 초기 트레이닝 프로토콜 중 구축된 이러한 DDR 레이트 및 다른 파라미터는 제 1 파워 콘텍스트로 사용될 수 있다(가령,

저속 - PhyPS[1]). PhyPS[1]는 다른 DDR 레이트에 기초하여 세팅될 수 있다(즉, 이 주파수에 대해 단계 4 내지 15를 반복). 다음 페이지는 더 높은 (PhyPS[0]) DDR 레이트를 위해 트레이닝된다. 위의 패스는 쓰기 레벨화 및 RxEn 하드웨어 트레이닝 알고리즘에일리어싱을 해결하였다. 쓰기 레벨화 및 RxEn 트레이닝 값은 MemC1k 데이터 레이트 비 PhyPS0/PhyPS1에 의해 스케일링되고, PhyPS[0]의 제 2 트레이닝 페이지에 대한 시드(seed)로 사용된다. 이 과정은 표 9에 도시되는 것과 같이 개시된다.

표 9

[0052]

	설명
16	모든 액티브 채널을 SR로 배치 (선택사항)
17	DDR 레이트를 요망 PhyPS[0] DDR 레이트로 프로그래밍
18	모든 액티브 채널을 정규 모드로 세팅(즉, SR로부터 채널을 빼냄). DLL이 리로킹됨.

[0053]

이 시점에서, DDR PLL 및 DLL 은 이용을 위해 적절하게 구성된다. PhyPS 콘텍스트는 여전히 PhyPS[0]이고, 파이 인터페이스는 PhyPS[0] DDR 레이트로 구동된다. 두 채널은 모두, 메모리가 존재할 경우, 표 8에 대체적으로 도시되는 바와 같이 다음 트레이닝 페이지를 시작할 준비가 되어 있다:

표 10

[0054]

	설명
19	모든 구동 강도 필드 세팅.
20	NB_TxChnClkEn를 세팅하고, 하드웨어-기반 펜스 트레이닝을 수행하며, 최종 FenseValue를 표시 및 세팅.
21	Address/Bank/RAS/CAS/WE/MemC1k D11 론치 지연을 세팅.
22	Cs & Odt D11 론치 지연을 세팅.
23	Cke D11 론치 지연을 세팅.
24	DRAM init를 수행.
25	디폴트 상수를 이용하여 하드웨어-기반 WL 트레이닝을 수행.
26	디폴트 상수를 이용하여 하드웨어-기반 RxEn 트레이닝을 수행.
27	TxDq/RxDqs 쓰기/읽기 트레이닝을 수행.
28	MRL 트레이닝이 여기서 발생할 수도 있고 나중에 발생할 수도 있다. NCLK는 이러한 DDR 레이트 동안 결과적인 목표 속도로 구동되어야 한다.

[0055]

이는 제 2 페이지 트레이닝 및 파워 콘텍스트 PhyPS[0]의 프로그래밍을 완료시킨다. 이 스테이지에서, 두 PhyPS 공간 모두에 대한 레이트는 이미 트레이닝되고 있고, 트레이닝된 값들은 트레이닝의 일부분으로 파이 인터페이스 PhyPS[0] CSR에 이미 쓰기되어 있다. 파이 인터페이스는 현재 PhyPS[0] 콘텍스트에 있다. PhyPS[1] 트레이닝된 값은 트레이닝되어 있지만 아직 PhyPS[1] CSR에 쓰기되어 있지 않다. PhyPS[0] CSR은 표 11에 대체로 도시되는 바와 같이 업데이트된다:

표 11

[0056]

	설명
29	1 내지 Master_0x08[PStateToAccess] - PhyPS[1]를 쓰기
30	저장된 PhyPS1 트레이닝된 값을 PhyPS1 CSR에 쓰기
31	0 내지 Master_0x08[PStateToAccess] - PhyPS[0]을 쓰기

[0057]

하나의 PhyPS만이 요구되는 경우에, BIOS는 어느 PhyPS 콘텍스트가 사용되어야하는 지를 선택할 수 있다. 역사적 의미를 보존하기 위해, BIOS는 PhyPS[0]에 대한 파이 인터페이스를 구성할 수 있다.

[0058]

S3로부터의 재개는 DRAM 트레이닝을 포함하지 않으며, 단지, (일반적으로 사우스 브리지에서) 비휘발성 상태에서부터 트레이닝된 값을 복원할 뿐이다. S3로부터의 재개는 PhyPS 콘텍스트가 PhyPS0이고 Master\_0x08[PStateToAccess]는 0이며 두 채널 모두 PhySR로부터 빠져나와 있고, (메모리가 채널 0 상에 존재하지 않을 경우에도) 마스터 채널은 채널 0임을 (~PwrOk 때문에) 통상적으로 보장할 것이다. S3로부터 재개를 위한 과정은 표 12에 일반적으로 도시된다:



## 표 12

[0059]

	설명
1	모든 액티브 채널을 SR 모드로 배치
2	채널 0 상에 메모리가 없을 경우, Master_0x18[PhyPSMasterChannel]을 세팅하여(3.7 참조) 액티브 DCT만에 의해 채널 1 상에서 발급된 PhyPS 변화가 효과를 나타냄. 메모리가 채널 1 상에 존재하지 않을 경우, Master_18[PhyPSMasterChannel]이 채널 0의 디폴트에서 남아야하기 때문에 어떤 액션도 요구되지 않음.
3	~PwrOk로 인해 PhyPS0의 파이 인터페이스가 S3에서 발급되면서, 모든 PhyPS0 트레이닝 및 도출 값들이 복원됨.
4	DDR 레이트를 PhyPS0 레이트로 업데이트.
5	마스터 채널에 1 내지 Master_0x08[PStateToAccess]를 쓰기. 이는 모든 PCI 액세스가 이제 PhyPS=1을 지향함을 의미.
6	모든 PhyPS1 트레이닝 및 도출 값을 복원.
7	DDR 레이트를 PhyPS1 레이트로 업데이트. 이제 두 세트의 PhyPS CSR 모두가 복원되고 준비됨.
8	마스터 채널에서 0x0B[PhyPS가 요망 PhyPS로 변경됨]를 발급. 이는 DDR PLL을 요망 PhyPS 레이트로 업데이트함. CfgDone 주장을 기다림.
9	SR로부터 모든 액티브 채널을 빼냄. SR에 실장되지 않은 채널은 남겨둠. DLL은 리로킹됨. 파이 인터페이스는 어느 한 PhyPS로 작동할 준비가 되어 있고, 현재 요청받은 PhyPS에 있음.

[0060]

웜 리셋 재개는 S3로부터의 재개와 거의 동일하다. S3로부터의 재개는 ~PwrOk를 가져서 PhyPS 콘텍스트를 PhyPS0로 세팅시키고 채널 0을 마스터 채널로 지정하여 PhyPS 변경을 통신하며, 두 채널 모두 PhySR로부터 파이 인터페이스를 취하게 한다. 이에 반해, 웜 리셋은 PhyPS 및 PhySR은 물론 마스터 채널을 알려지지 않은 상태에서 떠난다. 구조적 구멍이 웜 리셋 재개와 함께 존재한다. 시스템이 콜드 부트하고, 메모리 트레이닝을 완성하며, 그리고 비휘발성 메모리에 모든 트레이닝 값을 저장하기 전에, 웜 리셋 재개가 발급될 경우, 재개는 실패할 것이다. 이러한 발급을 피하기 위해, BIOS는 트레이닝 값이 연산되고 성공적으로 저장되었는 지 여부를 표시하기 위해 플래그(콜드 리셋 상에서 리셋되지만 웜 리셋을 통해 지속됨)를 이용하여야 한다 - 웜 리셋 중, BIOS는 이 플래그의 세팅을 볼 경우, 트레이닝된 상태를 복원함으로써 재개가 이루어질 수 있다. 트레이닝된 상태는 Master\_0x18[PhyPSMasterChannel]를 포함하고, SR 모드로 남아 있는 비-실장 채널(unpopulated channel)도 포함한다. 이 플래그가 세팅되지 않은 경우, BIOS는 파이를 보유하여야만 한다. 다음의 개시내용의 용도를 위해, 이 플래그는 세팅되어 있고 트레이닝 값이 저장되어 있다고 가정한다. 따라서, 웜 리셋 재개는 DRAM 트레이닝을 포함하지 않고, (일반적으로 사우스 브리지에서) 비휘발성 상태로부터 트레이닝 값을 복원할 뿐이다. 웜 리셋 재개를 수행하기 위한 과정은 일반적으로 표 13에서 도시된다:

## 표 13

[0061]

	설명
1	모든 액티브 채널을 SR로 배치.
2	BIOS는 어느 채널이 PhyPS 변화를 수용하는 마스터 채널인지를 결정. BIOS에 의한 Master_0x08[PStateToAccess] 또는 0x0B[PhyPS] 명령에 대한 후속 액세스는 마스터 채널로 지향되어야 함.
3	PhyPS[0] (Master_0x08[PStateToAccess] = 0) 선택
4	모든 PhyPS0 트레이닝 및 도출 값들을 복원
5	DDR 레이트를 PhyPS[0] 레이트로 업데이트
6	PhyPS[1] (Master_0x08[PStateToAccess] = 1) 선택
7	모든 PhyPS1 트레이닝 및 도출 값들을 복원
8	DDR 레이트를 PhyPS1 레이트로 업데이트
9	마스터 채널에서 0x0B[PhyPS가 요망 PhyPS로 변경됨] 발급. 이는 DDR PLL을 요망 PhyPS 레이트로 업데이트. CfgDone의 주장을 기다림.
10	SR로부터 모든 액티브 채널을 취함. 이는 DLL을 리로킹시킴.

[0062]

표 14는 각각의 파워 콘텍스트에 대해 중복되는 모든 PhyPS CSR의 리스트를 보여준다:

## 표 14

[0063]

	설명
1	PhyPS0에 대한 Master_0x00[Config0]
2	PhyPS1에 대한 Master_0x40[Config0]
3	PhyPS0에 대한 Master_0x01[Config1]
4	PhyPS1에 대한 Master_0x41[Config1]
5	PhyPS0에 대한 Master_0x02[AddrCmdCke]
6	PhyPS1에 대한 Master_0x42[AddrCmdCke]
7	Master_0x03[PllDiv, PllMult, PllDiv2Bypass, PllUpdate] 가 이제 PhyPS0 PLL 제어용이 됨
8	Master_0x43[PllDiv, PllMult, PllDiv2Bypass, PllUpdate] 가 이제 PhyPS1 PLL 제어용이 됨
9	8:0.DByte_0x{00,04,08} 는 PhyPS0 DS 및 P0dt 제어
10	8:0.DByte_0x{40,44,48} 는 PhyPS1 DS 및 P0dt 제어
11	8:0.DByte_0x0C 는 PhyPS0 D11 & Pre 컨피그
12	8:0.DByte_0x0C 는 PhyPS1 D11 & Pre 컨피그
13	8:0.DByte_0x{10,11} 은 PhyPS0 D11 파워 관리
14	8:0.DByte_0x{50,51} 은 PhyPS1 D11 파워 관리
15	8:0.DByte_0x13 은 PhyPS0 D11 DByte 컨피그
16	8:0.DByte_0x53 은 PhyPS1 D11 DByte 컨피그
17	PhyPS0에 대한 8:0.DByte_0x{14,15,16,17}[RxDQS 얼리/레이트 선택]
18	PhyPS1에 대한 8:0.DByte_0x{54,55,56,57}[RxDQS 얼리/레이트 선택]
19	PhyPS0에 대한 8:0.DByte_0x{18,19,1A,1B}[RxDQS 얼리 지연]
20	PhyPS1에 대한 20. 8:0.DByte_0x{58,59,5A,5B}[RxDQS 얼리 지연]
21	PhyPS0에 대한 8:0.DByte_0x1E [바이어스 제어]
22	PhyPS1에 대한 8:0.DByte_0x5E [바이어스 제어]
23	PhyPS0에 대한 8:0.DByte_0x1F [수신기 제어]
24	PhyPS1에 대한 8:0.DByte_0x5F [수신기 제어]
25	PhyPS0에 대한 RxDqs 레이트 지연{0,1,2,3}에 대해 8:0.DByte_0x{20,21,22,23}
26	PhyPS1에 대한 RxDqs 레이트 지연{0,1,2,3}에 대해 8:0.DByte_0x{60,61,62,63}
27	PhyPS0에 대한 D11TxDqs에 대해 8:0.DByte_0x{24,25}
28	PhyPS1에 대한 D11TxDqs에 대해 8:0.DByte_0x{64,65}
29	PhyPS0에 대한 D11RxEn에 대해 8:0.DByte_0x{26,27}
30	PhyPS1에 대한 D11RxEn에 대해 8:0.DByte_0x{66,67}
31	PhyPS0에 대한 D11TxDq에 대해 8:0.DByte_0x{28,29}
32	PhyPS1에 대한 D11TxDq에 대해 8:0.DByte_0x{68,69}
33	PhyPS0에 대한 D11TxUDQS에 대해 8:0.DByte_0x{2A,2B}
34	PhyPS1에 대한 D11TxUDQS에 대해 8:0.DByte_0x{6A,6B}
35	PhyPS0에 대한 D11TxUDq에 대해 8:0.DByte_0x{2C,2D}
36	PhyPS1에 대한 D11TxUDq에 대해 8:0.DByte_0x{6C,6D}
37	3:0.MemCk_0x00[TX 컨피그/DS] 는 PhyPS0 DS
38	3:0.MemCk_0x40[TX 컨피그/DS] 는 PhyPS1 DS
39	3:0.MemCk_0x1E[바이어스 제어] 는 PhyPS0 DS
40	3:0.MemCk_0x5E[바이어스 제어] 는 PhyPS1 DS
41	PhyPS0에 대한 3:0.MemCk_0x20 [D11 지연/컨피그]
42	PhyPS1에 대한 3:0.MemCk_0x60 [D11 지연/컨피그]
43	0:1.CsOdt_0x{00,04,08}[OdtCfg,CsCfg0145,CsCfg2367] 은 PhyPS0 DS
44	0:1.CsOdt_0x{40,44,48}[OdtCfg,CsCfg0145,CsCfg2367] 은 PhyPS1 DS
45	PhyPS0에 대한 0:1.CsOdt_0x1E [바이어스 제어]
46	PhyPS1에 대한 0:1.CsOdt_0x5E [바이어스 제어]
47	PhyPS0에 대한 0:1.CsOdt_0x{20,21} [D11 지연/컨피그]
48	PhyPS1에 대한 0:1.CsOdt_0x{60,61} [D11 지연/컨피그]
49	A2Byt_0x{00,04,08,0C,10}[A2CfgCke,A2Cfg0189,A2Cfg23AB,A2Cfg45CD,A2Cfg67EF] 는 PhyPS0 DS
50	A2Byt_0x{40,44,48,4C,50}[A2CfgCke,A2Cfg0189,A2Cfg23AB,A2Cfg45CD,A2Cfg67EF] 는 PhyPS1 DS
51	PhyPS0에 대한 A2Byt_0x1E [바이어스 제어]
52	PhyPS1에 대한 A2Byt_0x5E [바이어스 제어]
53	PhyPS0에 대한 A2Byt_0x{20,21} [D11 지연/컨피그]
54	PhyPS1에 대한 A2Byt_0x{60,61} [D11 지연/컨피그]
55	PhyPS0에 대한 D3Cmp_0x02 [Predriver slew rate(프리다이버 슬루 레이트)]



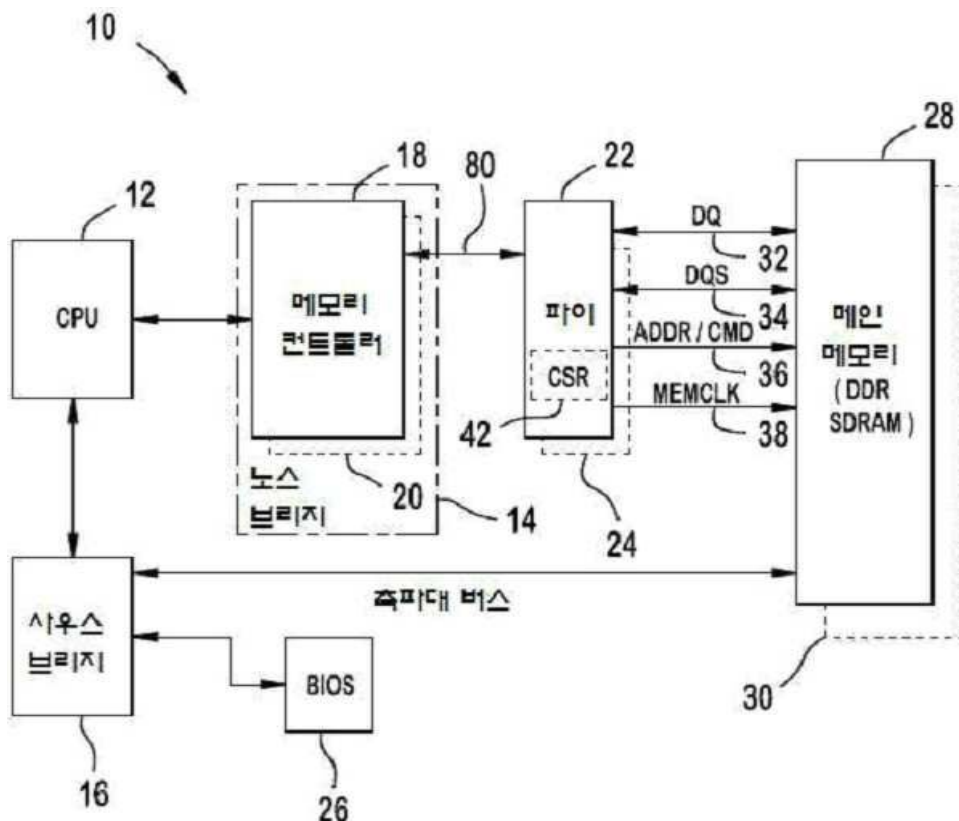
[0064] 여기서의 개시내용에 기초하여 많은 변화들이 가능하다. 예를 들어, 복수의 파워 콘텍스트가 (가령, 노스 브리지(14) 또는 사우스 브리지(16)의) 다른 메모리 위치에 저장될 수 있다. 이러한 시나리오에서, 표준 Phy 인터페이스는 전용 세트의 콘텍스트 감지 CSR의 필요없이, 파워 콘텍스트를 스위칭할 수 있다. 이러한 시나리오는 복수 세트의 CSR에 대한 필요성을 제거할 수 있지만, 콘텍스트 변화에 대한 대기 시간을 증가시킬 수 있다.

[0065] 여러 특징 및 조합들이 여기서 특징 조합으로 설명되지만, 각각의 특징 또는 요소가 타 특징 및 요소없이 단독으로 사용될 수 있고, 또는 타 특징 및 요소와 함께, 또는 없이, 다양한 조합으로 사용될 수 있다. 여기서 제공되는 방법 또는 순서도는 범용 컴퓨터 또는 프로세서에 의해 실행하기 위한 컴퓨터-판독가능 기록 매체에 통합된 컴퓨터 프로그램, 소프트웨어, 또는 펌웨어로 구현될 수 있다. 컴퓨터-판독가능 기록 매체의 예는 ROM, RAM, 레지스터, 캐시 메모리, 반도체 메모리 소자, 자기 매체(가령, 내장형 하드 디스크 및 탈착형 디스크, 자기-광학 매체, 및 광학 매체(가령, CD-ROM 디스크 및 DVD)를 포함한다.

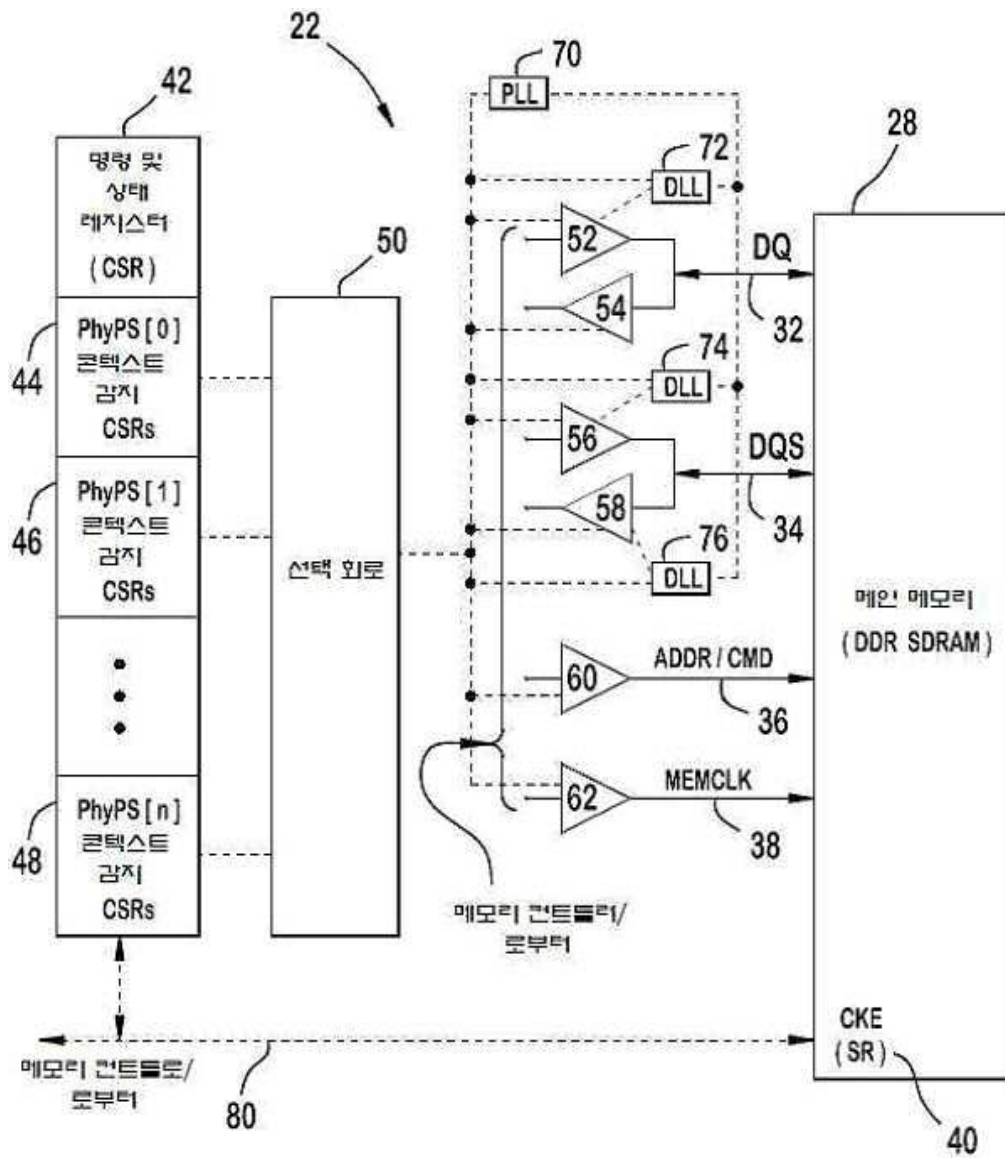
[0066] 적절한 프로세서는 예를 들어, 범용 프로세서, 전용 프로세서, 종래의 프로세서, 디지털 신호 프로세서(DSP), 복수의 마이크로프로세서, DSP 코어와 연계된 하나 이상의 마이크로프로세서, 컨트롤러, 마이크로컨트롤러, ASIC, 필드 프로그래머블 게이트 어레이(FPGA) 회로, 그외 다른 타입의 집적 회로(IC), 및/또는 상태 기계를 포함한다. 이러한 프로세서는 처리되는 하드웨어 설명 언어(HDL) 명령어의 결과를 이용하여 제작 프로세스를 구성함으로써 제작될 수 있다(이러한 명령어는 컴퓨터 판독가능 매체 상에 저장될 수 있음). 이러한 처리의 결과는 본 발명의 형태들을 구현하는 프로세서를 제작하기 위한 반도체 제조 프로세스에 사용되는 마스크웍일 수 있다.

## 도면

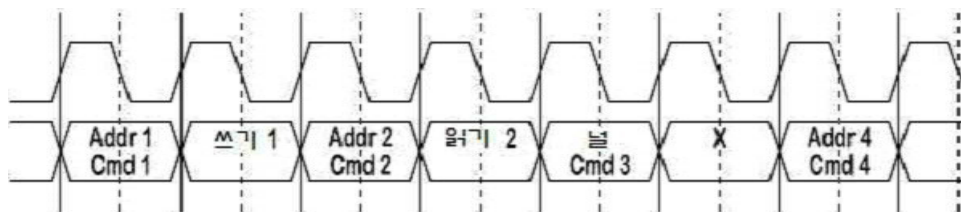
도면1



도면2



도면3



도면4

어드레스	값	임수
[15:13]	000	칩셋 타입 : D3Dbyte (9 인스턴스)
	001	칩셋 타입 : D3Clk (4 인스턴스)
	010	칩셋 타입 : D3Cmp (3 인스턴스)
	011	칩셋 타입 : D3MemVref (2 인스턴스)
	100	칩셋 타입 : D3CsOdt (1 인스턴스)
	101	칩셋 타입 : D3AByte1 (1 인스턴스)
	110	칩셋 타입 : D3AByte (1 인스턴스)
[12]	0/1	1일 때, 어드레스 비트[5,1,0]과 일치하는 칩셋 내의 모든 레지스터를 업데이트; 0일 때, 모든 어드레스 비트[5:0]을 디코딩
[11:8]	1011-0000	칩셋 타입의 인스턴스 수, 최대 12개의 인스턴스 허용.
	1100	일치하는 모든 ODT 보상 레지스터로 승출 [7:6]
	1101	일치하는 모든 드라이버 보상 레지스터로 승출 [7:6]
	1110	일치하는 모든 프리-드라이버 보상 레지스터로 승출 [7:6]
	1111	타입 필드가 일치하는 모든 칩셋으로 승출 [15:13]
[7:6]	11-00	보상 정보를 승출할 때 소스를 식별하는 최대 4개의 코드. 각각의 칩셋은 상관된 D3CMP 블록을 식별하는 2-비트 금속 스트랩 코드를 가질 것이다. 보상 승출이 이루어질 때, [7:6]은 데이터를 소싱한 D3CMP를 식별한다.
[5:0]	1x_xxxx	DLL 레지스터
	0x_xx00	구동 강도 레지스터
	0x_xx01	드라이버 보상 레지스터
	0x_xx10	프리-드라이버 보상 레지스터
	0x_xx11	ODT 보상 레지스터 (D3DBYTE 만 )