



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2022년04월01일

(11) 등록번호 10-2380776

(24) 등록일자 2022년03월25일

(51) 국제특허분류(Int. Cl.)  
G06F 12/08 (2016.01) G06F 9/30 (2018.01)(52) CPC특허분류  
G06F 12/08 (2013.01)  
G06F 9/30043 (2013.01)

(21) 출원번호 10-2016-0097039

(22) 출원일자 2016년07월29일

심사청구일자 2021년06월24일

(65) 공개번호 10-2017-0063334

(43) 공개일자 2017년06월08일

(30) 우선권주장

62/261,306 2015년11월30일 미국(US)

15/086,010 2016년03월30일 미국(US)

(56) 선행기술조사문헌

JP2000010770 A\*

- Peter Gadfort et al., "A Power Efficient  
Reconfigurable System-in-Stack: 3D  
Integration of Accelerators, FPGAs, and DRAM  
" (2004)\*

\*는 심사관에 의하여 인용된 문헌

(73) 특허권자

삼성전자주식회사

경기도 수원시 영통구 삼성로 129 (매탄동)

(72) 발명자

정, 홍중

미국 캘리포니아주 94087 서니배일 파킹톤 애비뉴  
1295

창, 무-티엔

미국 캘리포니아주 95134 산호세 팔밀라 드라이브  
3507 4103호

(74) 대리인

특허법인 고려

전체 청구항 수 : 총 20 항

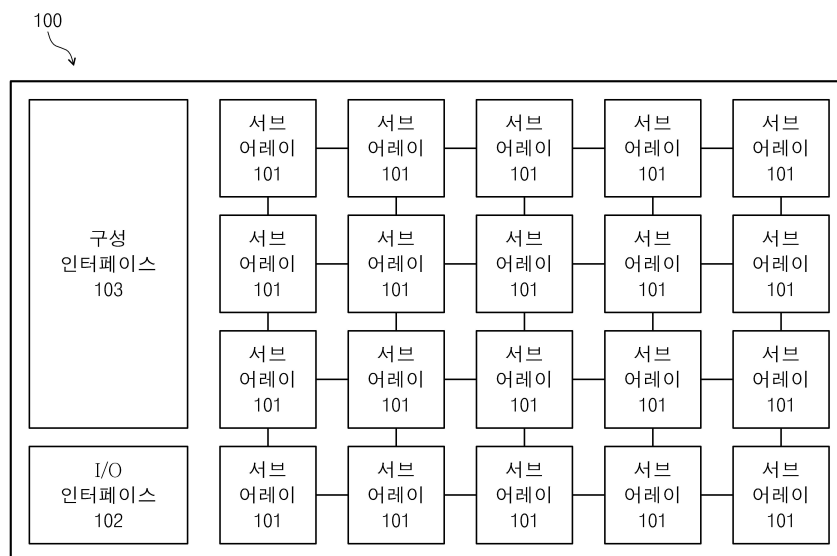
심사관 : 안지현

(54) 발명의 명칭 가속기 컨트롤러 및 그것의 가속기 로직 로딩 방법

## (57) 요약

본 발명의 실시 예에 따른 가속기 컨트롤러는 검출기와 로더를 포함할 수 있다. 검출기는 가상 머신(Virtual Machine) 또는 어플리케이션의 런타임(Runtime) 특징들을 검출하고, 검출된 런타임 특징들에 대응되는 어플리케이션 또는 가상 머신과 관련된 가속기 로직을 확인할 수 있다. 로더는 확인된 가속기 로직을 적어도 하나의 DRAM(Dynamic Random Access Memory)에 로드(Load)할 수 있다. 적어도 하나의 DRAM의 어레이(Array)는 확인된 가속기 로직에 기초하여 선택적으로 LUT(Look-Up Table)처럼 동작하도록 재구성 가능할 수 있다.

## 대표도



## 명세서

### 청구범위

#### 청구항 1

가상 머신(Virtual Machine) 또는 어플리케이션의 런타임(Runtime) 특징들을 검출하고, 상기 검출된 런타임 특징들에 대응되는 상기 어플리케이션 또는 상기 가상 머신과 관련된 가속기 로직을 확인하는 검출기; 그리고

상기 확인된 가속기 로직을 적어도 하나의 DRAM(Dynamic Random Access Memory) 어레이에 로드(Load)하는 로더를 포함하되,

상기 적어도 하나의 DRAM 어레이(Array)는 상기 확인된 가속기 로직에 기초하여 LUT(Look-Up Table)처럼 동작하도록 선택적으로 재구성 가능한 가속기 컨트롤러.

#### 청구항 2

제 1 항에 있어,

상기 적어도 하나의 DRAM을 더 포함하는 가속기 컨트롤러.

#### 청구항 3

제 1 항에 있어,

상기 적어도 하나의 DRAM 어레이는 운영 체제 환경의 캐시 일관성(Cache-Coherent) 주소 공간 내에 있는 가속기 컨트롤러.

#### 청구항 4

제 1 항에 있어,

상기 런타임 특징들은 상기 어플리케이션 또는 상기 가상 머신의 미리 정의된 아이디(Identification)를 기반으로 하는 가속기 컨트롤러.

#### 청구항 5

제 1 항에 있어,

상기 런타임 특징들은 기능 활용도(Function Utilization), 중앙 처리(Central Processing) 활용도, 메모리 활용도, 그리고 상기 어플리케이션 또는 상기 가상 머신과 관련된 레이턴시(Latency) 중 적어도 하나를 기반으로 하는 가속기 컨트롤러.

#### 청구항 6

제 1 항에 있어,

상기 가속기 로직은 LUT(Look-Up Table)를 포함하는 가속기 컨트롤러.

#### 청구항 7

제 1 항에 있어,

상기 적어도 하나의 DRAM 어레이는 DIMM(Dual In-line Memory Module)의 일부인 가속기 컨트롤러.

#### 청구항 8

제 7 항에 있어,

상기 DIMM은 LUT 또는 DRAM 메모리 어레이처럼 동작하도록 선택적으로 재구성 가능하지 않는 적어도 하나의 DRAM 어레이를 더 포함하는 가속기 컨트롤러.

## 청구항 9

제 1 항에 있어,

상기 확인된 가속기 로직은 로드(Load) 및 저장 커맨드들을 사용하여 상기 적어도 하나의 DRAM 어레이 내에 로드(Load)되는 가속기 컨트롤러.

## 청구항 10

어플리케이션 또는 가상 머신의 런타임 특징들을 검출하는 검출기; 그리고

상기 검출된 런타임 특징들에 대응하는 가속기 로직을 적어도 하나의 DRAM 어레이에 로드(Load)하는 로더를 포함하되,

상기 런타임 특징들은 상기 어플리케이션 또는 상기 가상 머신의 미리 정의된 아이디(Identification), 기능 활용도, 중앙 처리 활용도, 메모리 활용도, 그리고 상기 어플리케이션 또는 상기 가상 머신과 관련된 레이턴시 중 적어도 하나를 기반으로 하고,

상기 적어도 하나의 DRAM 어레이(Array)는 상기 가속기 로직에 기초하여 LUT(Look-Up Table)처럼 동작하도록 선택적으로 재구성 가능한 가속기 컨트롤러.

## 청구항 11

제 10 항에 있어,

상기 검출기는 상기 검출된 런타임 특징들에 대응되는 상기 어플리케이션 또는 상기 가상 머신과 관련된 가속기 로직을 확인하는 가속기 컨트롤러.

## 청구항 12

제 10 항에 있어,

상기 적어도 하나의 DRAM 어레이는 운영 체제 환경의 캐시 일관성 주소 공간 내에 있는 가속기 컨트롤러.

## 청구항 13

제 10 항에 있어,

상기 가속기 로직은 LUT를 포함하는 가속기 컨트롤러.

## 청구항 14

제 10 항에 있어,

상기 적어도 하나의 DRAM 어레이는 DIMM(Dual In-line Memory Module)의 일부인 가속기 컨트롤러.

## 청구항 15

제 14 항에 있어,

상기 DIMM은 LUT 또는 DRAM 메모리 어레이처럼 동작하도록 선택적으로 재구성 가능하지 않는 적어도 하나의 DRAM 어레이를 더 포함하는 가속기 컨트롤러.

## 청구항 16

검출기 및 로더를 포함하는 가속기 컨트롤러의 로딩 방법에 있어서,

상기 검출기에 의해, 운영 체제(OS) 환경 내에서 실행되는 어플리케이션 또는 가상 머신의 런타임 특징들을 검출하는 단계;

상기 검출기에 의해, 상기 검출된 런타임 특징들에 대응하는 가속기 로직을 확인하는 단계; 그리고

상기 로더에 의해, 로드(Load) 및 저장 커맨드들을 사용하여 상기 확인된 가속기 로직을 적어도 하나의 DRAM 서브 어레이(Sub-array) 내에 로드(Load)하는 단계를 포함하고,

상기 적어도 하나의 DRAM 서브 어레이는 선택적으로 LUT(Look-Up Table)처럼 동작하도록 재구성 가능한 방법.

#### 청구항 17

제 16 항에 있어,

상기 적어도 하나의 DRAM 서브 어레이는 LUT처럼 동작하도록 구성되고,

상기 적어도 하나의 DRAM 서브 어레이는 상기 운영 체제 환경의 캐시 일관성(Cache-Coherent) 주소 공간 내에 있고,

상기 적어도 하나의 DRAM 서브 어레이는 DIMM의 일부이고,

상기 검출된 런타임 특징들은 상기 어플리케이션 또는 상기 가상 머신의 미리 정의된 아이디(Identification), 기능 활용도, 중앙 처리 활용도, 메모리 활용도, 그리고 상기 어플리케이션 또는 상기 가상 머신과 관련된 레이턴시 중 적어도 하나를 기반으로 하는 방법.

#### 청구항 18

제 16 항에 있어,

상기 가속기 로직은 LUT를 포함하는 방법.

#### 청구항 19

제 16 항에 있어,

상기 확인된 가속기 로직은 가속기 컨트롤러에 의해 상기 적어도 하나의 DRAM 서브 어레이에 로드되는 방법.

#### 청구항 20

제 16 항에 있어,

상기 확인된 가속기 로직은 상기 어플리케이션 또는 상기 가상 머신에 의해 상기 적어도 하나의 DRAM에 로드되는 방법.

### 발명의 설명

### 기술 분야

[0001] 본 발명은 시스템 구조에 관한 것으로, 좀 더 상세하게는 가속기 컨트롤러 및 그것의 가속기 로직 로딩 방법에 관한 것이다.

### 배경 기술

[0002] 종래의 프로세싱 가속기(Processing Accelerator)는 효율적인 가속기 로직을 제공하기 위하여 FPGA(Field Programmable Gate Array) 또는 GPU(Graphics Processing Unit)와 같은 PLD(Programmable Logic Device)에 의해 구현될 수 있다. 데이터 센터와 모바일 장치 어플리케이션 등은 점차적으로 데이터 집적화되고 있고, 프로세싱 가속기를 사용하는 어플리케이션에 대해, 종래의 메모리 계층(Hierarchy)은 좋은 성능과 에너지 효율을 제공할 수 없다. 종래의 메모리 계층 내의 프로세싱 가속기의 최대 효율은 오프 칩(즉, 비 CPU) 메모리 액세스 등에 의해 저하된다. 높은 비용과 차선의 성능 때문에, 효율을 개선하기 위해 메모리 근처에 가속기 로직을 구현하는 것은 지금까지 제한적인 성공 만을 가져왔다. 더욱이, 만약 가속기 로직이 DRAM(Dynamic Random Access Memory)과 같은 메모리 근처에 구현되는 경우, DRAM 반도체 다이(Die) 상에 집적된 비 DRAM인 가속기 로직에 의해 DRAM 용량이 몰수되어 줄어든다.

[0003] 일반적으로, PLD(Programmable Logic Device)는 재구성 가능한 디지털 회로들을 구성하는데 사용되는 전자적 구성요소이다. 일반적으로 기능이 고정되는 로직 게이트 또는 로직 회로와 달리, PLD는 전통적으로 제조 시에는 정의되지 않는 기능을 갖는다. 또한, 종종 PLD가 회로에서 사용될 수 있기 전에, PLD는 원하는 기능을 수행하기 위해 프로그램 되거나 재구성 되어야 한다.

[0004] 전통적으로, PLD는 로직 장치 및 메모리 장치의 조합을 포함할 수 있다. 메모리는 일반적으로 원하는 기능을 정

의하는 프로그래밍 패턴을 저장하는데 사용된다. 집적 회로 내에서 데이터를 저장하는데 사용되는 대부분의 테크닉들은 실리콘 안티-퓨즈들(Silicon Anti-Fuses), SRAM(Static Random Access Memory), EPROM(Erasable Programmable Read Only Memory), EEPROM(Electronically EPROM), 그리고 불휘발성 RAM(Non-Volatile RAM) 등과 같은 PLD 내에서 사용되기 위하여 변형되어 적용되어왔다. 대부분의 PLD들은 일반적으로 PLD 내부의 변형된(Modified) 실리콘 영역을 가로질러 특별한 전압(비 동작상의 전압 또는 높은 전압)을 인가하여 프로그램되는 구성 요소들을 포함한다. 이러한 특별한 전압은 기술에 따라 전자적 연결들을 끊거나 설정(Set)하거나, 또는 PLD의 전기 회로의 레이아웃(Layout)을 변경한다.

[0005] PLD들의 가장 공통적인 타입들의 하나는 FPGA(Field-Programmable Gate Array)이다. FPGA는 생산된 이후에 고객 또는 설계자에 의해 구성되도록 디자인된 집적회로이다. 이러한 의미에서 “Field-Programmable” 라 한다. FPGA는 프로그램 가능한 로직 블록들의 어레이, 및 블록들을 서로 연결하도록 하는 재구성 가능한 인터커넥트(Interconnect)들의 계층을 포함한다. FPGA의 로직 블록들은 복잡하고 조합 가능한 함수들 또는 AND, XOR 등과 같이 단순히 간단한 로직을 구현하도록 구성될 수 있다.

## 발명의 내용

### 해결하려는 과제

[0006] 본 발명은 위에서 설명한 기술적 과제를 해결하기 위한 것으로, 본 발명은 검출기와 로더를 포함하는 가속기 컨트롤러 및 그것의 가속기 로직 로딩 방법을 제공할 수 있다.

### 과제의 해결 수단

[0007] 본 발명의 실시 예에 따른 가속기 컨트롤러는 검출기와 로더를 포함할 수 있다. 검출기는 가상 머신(Virtual Machine) 또는 어플리케이션의 런타임(Runtime) 특징들을 검출하고, 검출된 런타임 특징들에 대응되는 어플리케이션 또는 가상 머신과 관련된 가속기 로직을 확인할 수 있다. 로더는 확인된 가속기 로직을 적어도 하나의 DRAM(Dynamic Random Access Memory)에 로드(Load)할 수 있다. 적어도 하나의 DRAM의 어레이(Array)는 확인된 가속기 로직에 기초하여 선택적으로 LUT(Look-Up Table)처럼 동작하도록 재구성 가능할 수 있다.

[0008] 본 발명의 다른 실시 예에 따른 가속기 컨트롤러는 검출기와 로더를 포함할 수 있다. 검출기는 어플리케이션 또는 가상 머신의 런타임 특징들을 검출할 수 있다. 로더는 검출된 런타임 특징들에 대응하는 가속기 로직을 적어도 하나의 DRAM에 로드(Load)할 수 있다. 런타임 특징들은 어플리케이션 또는 가상 머신의 미리 정의된 아이디(Identification), 기능 활용도, 중앙 처리 활용도, 메모리 활용도, 그리고 어플리케이션 또는 가상 머신과 관련된 레이턴시 중 적어도 하나를 기반으로 할 수 있다. 적어도 하나의 DRAM의 어레이(Array)는 가속기 로직에 기초하여 선택적으로 LUT(Look-Up Table)처럼 동작하도록 재구성 가능할 수 있다.

[0009] 본 발명의 다른 실시 예에 따른 방법은 OS 환경 내에서 실행되는 어플리케이션 또는 가상 머신의 런타임 특징들을 검출하는 단계, 검출된 런타임 특징들에 대응하는 가속기 로직을 확인하는 단계, 그리고 로드(Load) 및 저장 커맨드들을 사용하여 확인된 가속기 로직을 적어도 하나의 DRAM의 서브 어레이(Sub-array) 내에 로드(Load)하는 단계를 포함할 수 있다.

## 발명의 효과

[0010] 본 발명의 실시 예에 따르면, DRAM 기반의 재구성 가능한 로직이 제공될 수 있다. 제공된 로직에 의해, 종래의 FPGA에 비하여 높은 밀도, 낮은 비용, 그리고 좀 더 좋은 유연성(Flexibility)이 제공될 수 있다.

### 도면의 간단한 설명

[0011] 도 1은 본 발명의 실시 예에 따른 재구성 가능한 메모리 로직 장치를 보여주는 그림이다.

도 2는 본 발명의 다른 실시 예에 따른 재구성 가능한 메모리 로직 장치를 보여주는 블록 다이어그램이다.

도 3a 내지 도 3d는 본 발명에 따른 하나 또는 하나 이상의 DRAM-FPGA를 포함하는 DIMM을 보여주는 블록도이다.

도 4는 본 발명에 따른 메모리 채널 내에서 하나 또는 하나 이상의 DRAM-FPGA를 사용하는 시스템 구조를 보여주는 기능 블록도이다.

도 5는 본 발명에 따른 가상 머신 또는 어플리케이션의 런타임 특징을 검출하고 재구성 가능한 DRAM 서브 어레이

이 내의 가상 머신 또는 어플리케이션에 해당하는 가속기 로직을 저장하는 방법을 보여주는 순서도이다.

도 6은 본 발명에 따른 메모리 채널 내에서 하나 또는 하나 이상의 DRAM-FPGA를 사용하는 QPI 기반의 시스템 구조를 보여주는 블록도이다.

도 7은 본 발명에 따른 메모리 채널 내에서 하나 또는 하나 이상의 DRAM-FPGA를 포함하는 시스템 구조를 사용할 수 있는 정보 처리 시스템을 보여주는 기능 블록도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0012] 아래에서는, 본 발명의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 용이하게 실시할 수 있을 정도로, 본 발명의 실시 예들이 명확하고 상세하게 기재될 것이다.
- [0013] 본 명세서에 기재된 기술적 특징은 메모리 채널 내에 하나 또는 하나 이상의 DRAM-FPGA(Dynamic Random Access Memory Field Programmable Gate array)들을 사용하는 시스템 구조에 관한 것이다. DRAM-FPGA는 선택적으로 재구성 가능한 DRAM 셀 어레이를 LUT(Lookup Table)로 사용하고, 재구성 가능한 로직을 메모리 근처에 구현한다. 또한, DRAM-FPGA는 임의의 로직 디자인들로 재구성 가능하다.
- [0014] 더욱이, 이러한 DRAM 기반의 재구성 가능한 로직은 DRAM 제조 공정들과 호환되고, 인-메모리(In-Memory) 프로세싱에 대한 하나의 좋은 후보이다. 종래의 FPGA들과 비교하여, DRAM 기반의 재구성 가능한 로직은 보다 높은 밀도(Density), 낮은 비용, 그리고 좀 더 좋은 유연성(Flexibility)을 제공한다. 즉, 좀 더 좋은 유연성은 웨어하우스 스케일(Warehouse-Scale)의 데이터 센터들 내의 다른 연산 노드(Node)들 사이에서의 빠른 이주(Migration)를 제공하는 자체 변형(Self-Modify) 및 빠른 동적 재구성을 의미한다.
- [0015] FPGA의 형태를 갖는 PLD들을 사용하는 종래의 시스템 구조들은 일반적으로 PCIe(Peripheral Component Interconnect- express) 슬롯 내에 FPGA를 위치시킨다. PCIe 슬롯은 캐시 일관성(Cach-Coherent) 주소 공간에 있지 않고, 낮은 통신 밴드위스(Bandwidth)를 제공한다. 또한, PCIe 슬롯 내의 FPGA 내의 가속기 로직의 동작은 드라이버에 의해 제공되는 명시적인 명령에 의해 수행된다.
- [0016] 도 1은 본 발명의 실시 예에 따른 재구성 가능한 메모리 로직 장치(100)를 보여주는 그림이다. 재구성 가능한 메모리 로직 장치(100)는 LUT(Look-up Table) 모드 또는 정규(Regular) DRAM(Dynamic Random Access Memory) 모드의 두 가지 모드에서 동작하도록 구성되고 연결될 수 있다. 재구성 가능한 메모리 로직 장치(100)는 하나 또는 하나 이상의 서브 어레이들(101), I/O 인터페이스(102), 및 구성 인터페이스(103)를 포함할 수 있다. 서브 어레이들(101)은 복수의 행 및 열로 정렬된다. 서브 어레이들(101) 각각은 다수의 메모리 셀들(미도시)을 포함할 수 있고, 다수의 메모리 셀들(미도시)은 또한 복수의 행 및 열 어레이로 정렬된다. 재구성 가능한 메모리 로직 장치(100)는 도 1에 도시되지 않은 다른 구성 요소들 및/또는 부품들을 포함할 수 있다.
- [0017] 일 실시 예로서, 서브 어레이들(101)은 DRAM을 포함할 수 있다. LUT 또는 종래의 DRAM 서브 어레이처럼 동작하기 위해, 하나 또는 하나 이상의 서브 어레이들(101)은 일반적인 메모리 액세스 동작들을 이용하여 선택적으로 재구성되거나 다시 프로그램 될 수 있다. 본 명세서에서 사용된 것처럼, 재구성 가능한 메모리 로직 장치(100)는 여기서 FPGA로서 나타날 수 있다. DRAM을 포함하는 서브 어레이들(101)을 포함하는 재구성 가능한 메모리 로직 장치(100)의 실시 예들에 대해, 재구성 가능한 메모리 로직 장치(100)는 본 명세서에서 “DRAM-FPGA”로 칭해질 수 있다.
- [0018] 일 실시 예로서, LUT들 또는 RAM(Random Access Memory)들로 구성된 서브 어레이들(101)의 수는 필요에 따라 동적으로 조정될 수 있다. 다른 실시 예로서, 하나 또는 하나 이상의 서브 어레이들(101)이 재구성이 불가능하거나 종래의 메모리 서브 어레이처럼 구성될 수 있는 반면에, 하나 또는 하나 이상의 서브 어레이들(101)은 LUT 또는 종래의 메모리 서브 어레이처럼 동작하기 위해 재구성될 수 있다. 본 명세서에서 사용된 것처럼, LUT 또는 종래의 메모리 서브 어레이처럼 동작하도록 재구성 가능한 서브 어레이는 RLUT(Reconfigurable LUT) 서브 어레이로 칭해진다.
- [0019] 만약 서브 어레이(101)가 LUT처럼 동작하도록 구성되는 경우, LUT 동작은 산술 논리 기능(더하기, 곱하기 등), 로직 기능(AND, OR, XOR 등), 또는 이들의 조합 등과 같은 로직 기능을 구현할 수 있다. 다만, 본 발명은 이에 제한되지 않는다. 로직 기능은 서브 어레이 상에 메모리 쓰기 동작 또는 특별한 쓰기 동작을 단지 수행함에 따라 변경될 수 있다. 이에 따라, 재구성 가능한 메모리 로직 장치(100)의 동작 중에, 로직 기능들은 동적으로 재구성되거나 변경될 수 있다. 특별한 쓰기 동작은 쓰기 동작이 특정 서브 어레이와 연관됨을 표시하는 식별자를 포함할 수 있다. 종래의 쓰기 동작(또는 유사한 동작)의 사용은 EEPROM 등의 프로그램을 위해 사용되는 것과 같



은 특별한 전압(비 동작 전압 또는 높은 전압)의 필요성 없이도 선택된 서브 어레이(101)가 재 프로그래밍 되도록 할 수 있다.

[0020] RLUT 서브 어레이들 및 RAM 서브 어레이들의 혼합(Co-mingling)은 몇 가지 장점들을 제공할 수 있다. 예를 들어, RAM 서브 어레이 내에 저장된 데이터에 인접한 경우, RLUT 서브 어레이에 의해 수행되는 계산이 빨라질 수 있고 전력 요구량이 줄어들 수 있다. 이는 데이터가 개별 구성 요소들 사이의 버스들을 통해 이동될 필요가 없기 때문이다. 더욱이, 일 실시 예로서, 하나 또는 하나 이상의 서브 어레이들(101)은 한번에 액세스될 수 있다. 이에 따라, 파워 소모 및 재구성 가능한 메모리 로직 장치(100)의 계산 복잡도가 감소할 수 있다. 이와 같이, DRAM RLUT에 의해 제공되는 근거리 데이터 컴퓨팅은 좀 더 빠르고 좀 더 효율적일 수 있다.

[0021] RLUT 서브 어레이의 근거를 프로세서(미도시)와 메모리(미도시)를 제조하기 위해 사용된 것과 유사한 DRAM 기술에 두는 것에 의해, RLUT 서브 어레이는 프로세서 및/또는 메모리로서 동일 다이(Die) 또는 패키지 내에 존재할 수 있다. 동일한 제조 공정을 사용한다면, 본 발명에 따른 하나의 메모리 채널 내에 하나 또는 하나 이상의 RLUT들을 사용하는 시스템의 제조 비용이 낮아질 수 있다. 더욱이, RLUT 서브 어레이들의 근거를 DRAM 서브 어레이들에 두는 것으로 인해, SRAM 기반의 FPGA의 밀도와 비교하여 상대적으로 고 밀도가 달성될 수 있다. 예를 들어, SRAM 기반의 FPGA가 메모리 셀 또는 비트 단위의 정보 당(Per) 여섯 개의 트랜지스터(6T)를 요구하는 반면에, DRAM 기반의 RLUT는 메모리 셀 또는 비트 단위의 정보 당(Per) 단지 하나의 트랜지스터 및 하나의 캐패시터(1T1C)를 요구할 수 있다. 게다가, DRAM 기반의 RLUT는 SRAM 또는 플래시 기반의 FPGA들과 비교하여 비용을 낮출 수 있다.

[0022] I/O 인터페이스(102)는 선택된 서브 어레이(101)에 읽거나 쓰도록 구성될 수 있다. 쓰기 액세스는 서브 어레이(101)에 대한 쓰기 동작을 포함할 수 있다. 이는 서브 어레이(101)가 LUT로서 동작할지 여부 또는 종래의 DRAM 서브 어레이로서 동작할지 여부를 정의하기 위함이다. 일부 실시 예들에서, 모든 메모리 액세스들 또는 동작들은 I/O 인터페이스(102)를 통과할 수 있다. 만약 메모리 액세스가 RAM 서브 어레이 또는 RAM 서브 어레이로서 구성된 RLUT 서브 어레이의 리바이벌(Revival)을 위해 데이터를 저장하는 서브 어레이에 대한 액세스인 경우, I/O 인터페이스(102)는 단순히 종래의 메모리 액세스로서 읽기/쓰기 요청을 처리할 수 있다. 하지만, 만약 메모리 액세스가 LUT로서 구성된 RLUT 서브 어레이에 대한 액세스인 경우, I/O 인터페이스(102)는 프로세싱하기 위해 메모리 액세스를 구성 인터페이스(103)에 통과시킬 수 있다.

[0023] 구성 인터페이스(103)는 전체적으로 재구성 가능한 메모리 로직 장치(100) 내의 신호들의 라우팅(Routing)을 조정하거나, 또는 각각의 RLUT 서브 어레이(101) 내의 신호들의 라우팅을 조정하도록 구성될 수 있다. 예를 들어, 구성 인터페이스(103)는 RLUT 및/또는 RAM으로 구성된 복수의 서브 어레이들 사이의 신호들의 라우팅을 조정하도록 구성될 수 있다. 일 실시 예로서, I/O 인터페이스(102)는 서브 어레이들(101)에 대한 데이터 액세스를 관리하도록 구성될 수 있다. 예를 들어, 일 실시 예로서, I/O 인터페이스(102)는 프로세서(미도시)로부터 구성 정보를 수신할 수 있다. 구성 정보는 서브 어레이들(101)에 대한 I/O 액세스를 제어하기 위해 I/O 인터페이스(102)를 구성하는데 사용되며, 이에 서브 어레이들(101) 내의 메모리의 위치들은 주소화 가능한(Addressable) 레지스터들로서 보일 수 있다. 구성 인터페이스(103)는 각각의 서브 어레이들(101) 사이의 인터커넥트들(Interconnects)과 신호 라우팅을 관리하도록 구성될 수 있다. 예를 들어, 이러한 인터커넥트들과 신호 라우팅은 포인트 투 포인트(Point-to-Point) 라우팅, 주소 기반의 라우팅, 또는 이들의 조합 등에 의해 이루어질 수 있다. 예를 들어, 일 실시 예로서, 구성 정보는 구성 인터페이스(103)를 구성하는데 사용되는 I/O 인터페이스(102)에 의해 수신될 수 있고, 이에 서브 어레이들(101)은 복수의 서브 어레이들(101)에 걸쳐서 복잡한 논리 동작들을 수행할 수 있다. 일 실시 예로서, I/O 인터페이스(102)는 구성 인터페이스(103)를 포함할 수 있다.

[0024] 메모리 액세스는 특정 RLUT 서브 어레이(101) 내의 룩 업 테이블(Look-up Table)을 저장하는 쓰기 동작을 포함할 수 있다. 또한, 메모리 액세스는 LUT의 크기에 따라 일련의 메모리 액세스들을 포함할 수 있다. 일부 실시 예들에서, 특정 메모리 액세스들은 LUT에 대한 입력의 수와 LUT로부터의 출력의 수를 나타낼 수 있다. 또한, 추가적인 메모리 액세스들은 RLUT 서브 어레이(101)에 대한 신호 라우팅 정보 및/또는 둘 또는 둘 이상의 RLUT 서브 어레이들이 로직 기능(덧셈기 등)을 수행하기 위해 캐스케이드(Cascade)되어 있거나 함께 라우팅 되어 있는지 여부를 나타낼 수 있다. 서브 어레이(101)에 의해 제공될 수 있는 논리 기능들의 상세 설명들은 본 명세서의 참고로 인용된 미국 출원 (출원 번호: 14/814,503)(발명의 명칭: DRAM-Based Reconfigurable Logic)의 개시 내용에 명시되어 있다. 또는, 메모리 액세스는 RLUT 서브 어레이(101)를 종래의 RAM 서브 어레이로서 동작하도록 재구성하는 쓰기 동작을 포함할 수 있다.

[0025] 도 2는 본 발명의 다른 실시 예에 따른 재구성 가능한 메모리 로직 장치(200)를 보여주는 블록 다이어그램이다.

도 2를 참조하면, 재구성 가능한 메모리 로직 장치(200)는 독립된 다이(Die)나 단일 다이의 일부에 구현될 수 있다. 단일 다이의 일부에 구현되는 경우, 재구성 가능한 메모리 로직 장치(200)는 공유 다이(시스템 온 칩, 프로세서 캐시 등) 상에서 다른 구성 요소들(미도시)과 함께 집적될 수 있다. 다만, 공유 다이의 종류는 이에 한정되는 것이 아니다. 재구성 가능한 메모리 로직 장치(200)는 하나 또는 하나 이상의 서브 어레이들(201), 복수의 통신 버스들(202), 입력 신호 패드들(203), 그리고 출력 신호 패드들(204)을 포함할 수 있다. 재구성 가능한 메모리 로직 장치(200)는 버스 기반의 인터커넥션과 라우팅 방식으로 구현될 수 있다. 통신 버스(202)는 서브 어레이들(201) 사이의 라우팅 방식이 서브 어레이들(201) 사이의 신호를 다시 라우팅하기 위해 동적으로 변경되도록 할 수 있다.

[0026] 만약 서브 어레이(201)가 LUT처럼 동작하도록 구성되는 경우, LUT 동작은 산술 논리 기능(더하기, 곱하기 등), 로직 기능(AND, OR, XOR 등), 또는 이들의 조합 등과 같은 로직 기능을 구현할 수 있다. 서브 어레이(201)에 의해 제공될 수 있는 논리 기능들의 상세 설명들은 본 명세서의 참고로 인용된 미국 출원 (출원 번호: 14/814,503)(발명의 명칭: DRAM-Based Reconfigurable Logic)의 개시 내용에 명시되어 있다. 로직 기능은 서브 어레이 상에 메모리 쓰기 동작 또는 특별한 쓰기 동작을 단지 수행하여 변경될 수 있다. 이에 따라, 이는 재구성 가능한 메모리 로직 장치(200)의 동작 중에 로직 기능들을 동적으로 재구성되거나 변경되도록 할 수 있다. 특별한 쓰기 동작은 쓰기 동작이 특정 서브 어레이와 연관됨을 표시하는 식별자를 포함할 수 있다.

[0027] 일 실시 예에 따라, 도 1의 하나 또는 하나 이상의 서브 어레이들(101) 및/또는 도 2의 하나 또는 하나 이상의 재구성 가능한 메모리 로직 장치(200)는 DIMM(Dual In-line Memory Module) 내에서 사용될 수 있다. 즉, 하나 또는 하나 이상의 정규 DRAM 모듈들과 함께, DIMM은 하나 또는 하나 이상의 DRAM-FPGA 모듈들, 하나 또는 하나 이상의 RLUT 모듈들, 하나 또는 하나 이상의 DRAM-FPGA 모듈들과 하나 또는 하나 이상의 RLUT 모듈들의 조합, 또는 하나 또는 하나 이상의 DRAM-FPGA 모듈들 및/또는 하나 또는 하나 이상의 RLUT 모듈들의 조합을 포함하도록 구성될 수 있다.

[0028] 도 3a 내지 도 3d는 본 발명에 따른 하나 또는 하나 이상의 DRAM-FPGA를 포함하는 DIMM을 보여주는 블록도이다.

[0029] 3a를 참조하면, DIMM(300)은 복수의 DRAM-FPGA 모듈들(301)을 포함할 수 있다. 여기서, 네 개의 DRAM-FPGA 모듈들(301)이 도시되었다. 그러므로, DIMM(300)의 특정 실시 예에 대해, 모든 모듈의 자리는 DRAM-FPGA 모듈들(301)에 의해 채워질 수 있다. 또는, 모든 모듈의 자리는 RLUT 모듈들에 의해 채워질 수 있고, RLUT 모듈들 각각은 오직 RLUT(Reconfigurable Look-Up Table)들을 포함할 수 있다.

[0030] 도 3b를 참조하면, DIMM(310)은 하나 또는 하나 이상의 DRAM-FPGA 모듈들(301)과 하나 또는 하나 이상의 정규 DRAM 모듈들(302)을 포함할 수 있다. 여기서, 두 개의 DRAM-FPGA 모듈들(301)과 두 개의 정규 DRAM 모듈들(302)이 도시되었다. DIMM(310)의 특정 실시 예에 대해, DRAM-FPGA 모듈들(301)과 정규 DRAM 모듈들(302)의 물리적인 배치는 어떠한 순서도 포함할 수 있다. 또는 DIMM(301)의 구성은 하나 또는 하나 이상의 RLUT 모듈들과 하나 또는 하나 이상의 정규 DRAM 모듈들(302)일 수 있다.

[0031] 도 3c를 참조하면, DIMM(320)은 복수의 하이브리드 DRAM-FPGA 모듈들(303)을 포함할 수 있다. 여기서, 네 개의 하이브리드 DRAM-FPGA 모듈들(303)이 도시되었다. 하이브리드 DRAM-FPGA 모듈(303)은 LUT 또는 종래의 DRAM 서브 어레이처럼 동작하도록 재구성 가능한 하나 또는 하나 이상의 서브 어레이들을 포함하고, 종래의 DRAM 서브 어레이처럼 동작하는 재구성이 불가능한 하나 또는 하나 이상의 서브 어레이들을 포함할 수 있다. 즉, DIMM(320)의 구성은 오직 하이브리드 DRAM-FPGA 모듈들(303)을 포함할 수 있다.

[0032] 도 3d를 참조하면, DIMM(330)은 하나 또는 하나 이상의 하이브리드 DRAM-FPGA 모듈들(303)과 하나 또는 하나 이상의 정규 DRAM 모듈들(302)을 포함할 수 있다. 여기서, 두 개의 하이브리드 DRAM-FPGA 모듈들(303)과 두 개의 정규 DRAM 모듈들(302)이 도시되었다. DIMM(330)의 특정 실시 예에 대하여, 하이브리드 DRAM-FPGA 모듈들(303)과 정규 DRAM 모듈들(302)의 물리적 배치는 어떠한 순서도 포함할 수 있다.

[0033] DIMM은 다른 구성도 포함할 수 있다. 즉, 본 발명은 하나 또는 하나 이상의 DRAM-FPGA 모듈들, 하나 또는 하나 이상의 정규 DRAM 모듈들, 하나 또는 하나 이상의 하이브리드 DRAM-FPGA 모듈들 및/또는 하나 또는 하나 이상의 RLUT 모듈들을 포함하는 DIMM 구성이 가능하며, 이에 제한되지는 않는다.

[0034] 도 4는 본 발명에 따른 메모리 채널 내에서 하나 또는 하나 이상의 DRAM-FPGA를 사용하는 시스템 구조(400)를 보여주는 기능 블록도이다. 도 5는 본 발명에 따른 가상 머신 또는 어플리케이션의 런타임 특징을 검출하고 재구성 가능한 DRAM 서브 어레이 내의 가상 머신 또는 어플리케이션에 대응하는 가속기 로직을 저장하는 방법(500)을 보여주는 순서도이다. 도 4에 도시된 순서도(500)의 S501 내지 S504의 단계는 도 5를 참조하여 설명될



것이다.

- [0035] 도 4를 참조하면, 시스템 구조(400)는 RAM(Random Access Memory)과 같은 시스템 메모리 및 하나 또는 하나 이상의 대용량 스토리지 장치들(403)에 연결된 CPU(401)(Central Processing Unit)를 포함할 수 있다. 예를 들어, CPU(401)는 메모리 채널(402)를 통해 시스템 메모리와 연결될 수 있다. 시스템 메모리의 RAM은 SRAM(Static RAM) 및/또는 DRAM(Dynamic RAM)을 포함할 수 있다. 일 실시 예로서, 도 4에 도시된 바와 같이, 시스템 메모리는 하나 또는 하나 이상의 DIMM 내의 DRAM을 포함할 수 있다. 일 실시 예로서, 하나 또는 하나 이상의 정규 DRAM 모듈들과 함께, DIMM들(404)은 하나 또는 하나 이상의 DRAM-FPGA 모듈들, 하나 또는 하나 이상의 RLUT 모듈들, 하나 또는 하나 이상의 DRAM-FPGA 모듈들과 하나 또는 하나 이상의 RLUT 모듈들의 조합, 또는 하나 또는 하나 이상의 DRAM-FPGA 모듈들 및/또는 하나 또는 하나 이상의 RLUT 모듈들의 조합을 포함하도록 구성될 수 있다. 메모리 채널 내의 DRAM-FPGA들은 하나 또는 하나 이상의 DIMM들로 구현된다.
- [0036] OS(405)(Operating System)는 CPU(401) 상에서 실행된다. OS(405)는 Unix 기반의 OS, Unix와 유사한 OS, 리눅스 기반의 OS, 크롬 기반의 OS 또는 윈도우 기반의 OS 등일 수 있다. 다만, 본 발명은 이에 제한되는 것이 아니다. 하나 또는 하나 이상의 가상 머신(VM: Virtual Machine)들은 OS(405) 상에서 실행될 수 있다. 도 4에 도시된 바와 같이, 가상 머신들(VM1~VMn)은 OS(405) 상에서 실행되고 있다. 시스템 구조(400)가 데이터 센터 환경에서 구현되는 상황에서, 가상 머신들(VM1~VMn)은 데이터 센터 어플리케이션들을 의미할 수 있다. 시스템 구조(400)가 모바일 장치 환경에서 구현되는 상황에서, 가상 머신들(VM1~VMn)은 모바일 장치 어플리케이션들을 의미할 수 있다.
- [0037] OS(405)는 가속기 컨트롤러(406)를 포함할 수 있다. 가속기 컨트롤러(406)는 실체화된(instantiated) 가상 머신(VM)과 관련된 런타임 특징들을 검출(407)하여 가상 머신(VM)의 실체화를 검출한다. 여기서 사용된 것과 같이, 가상 머신(VM)의 “런타임 특징들”이라는 용어는 동작 특성들을 의미한다. 예를 들어, 동작 특성들은 가상 머신(VM)의 ID(Identity) 및/또는 특정 설정 구성일 수 있다. 다만, 본 발명은 이에 제한되는 것이 아니다. 일 실시 예로서, 가속기 컨트롤러(406)는 소프트웨어 기반, 하드웨어 기반, 또는 이들의 조합에 의해 구현될 수 있다. 일 실시 예로서, 가속기 컨트롤러(406)는 OS(405) 내의 소프트웨어 모듈을 포함할 수 있다. 다른 실시 예로서, 가속기 컨트롤러(406)는 메모리 컨트롤러(미도시) 내의 하드웨어 모듈을 포함할 수 있다.
- [0038] 일 실시 예로서, 가속기 컨트롤러(406)는 검출기(407)와 로더(408)(Loader)를 포함할 수 있다. 검출기(407)는 어플리케이션 또는 가상 머신의 런타임 특징들(409)을 검출하고, 검출된 런타임 특징들(409)에 대응되는 어플리케이션 또는 가상 머신들과 관련된 가속기 로직(410)을 확인한다. 다른 실시 예로서, 가속기 컨트롤러(406)는 런타임 시에 가상 머신(VM) 또는 어플리케이션을 프로파일(Profile)하고, 런타임 서명서명(Signature)을 수집하고, 가속기 로직(410)의 보관 장소를 찾기 위해 런타임 서명서명을 이용한다. 런타임 서명은 기능 활용도, CPU 활용도, 레이턴시(Latency), 및/또는 다른 마이크로 구조(Micro-architecture) 통계 등을 포함할 수 있다. 다만, 본 발명은 이에 제한되는 것이 아니다. 일 실시 예로서, 가상 머신(VM) 또는 어플리케이션의 런타임 특징들을 검출하는데 사용되는 어플리케이션의 런타임 특징들(409)은 가속기 컨트롤러(406) 내에 저장되거나 가속기 컨트롤러(406)의 일부일 수 있다. 예를 들어, 다른 실시 예로서, 런타임 특징들(409)은 스토리지 장치(403) 내에 저장되고, 가속기 컨트롤러(406)가 초기화될 때 가속기 로직(410)에 로드(Load)될 수 있다.
- [0039] 가상 머신(VM) 또는 어플리케이션의 검출된 런타임 특징들(409)에 기반하여, 가속기 컨트롤러(406)는 로더(408)를 선택하고, 가속기 로직(410)을 회수하고, 가상 머신(VM)을 위한 가속기 로직(410)을 메모리 채널 저장 명령들을 이용하여 DRAM-FPGA(411) 내에 로드한다. 가속기 로직(410)은 대용량 스토리지 장치(403) 내에 저장될 수 있고, 가상 머신에 의해 이용될 수 있다. 일 실시 예로서, 회수된 가속기 로직(410)은 가상 머신(VM)에 의해 요청되거나, 또는 선택적으로 가상 머신(VM)에 의해 요청될 수 있다. 다른 실시 예로서, 가속기 로직(410)은 OS(405) 또는 유저에 의해 요청되거나 권장(Recommend)될 수 있다. 다른 실시 예로서, 가속기 컨트롤러(406)는 가상 머신(VMn)에 대한 가속기 로직(410)을 확인할 수 있고, 가상 머신(VMn)은 메모리 채널 저장 명령들을 이용하여 가속기 로직(410)을 회수하여 DRAM-FPGA(411) 내에 저장할 수 있다. 가속기 로직(410)이 시스템 구조(400)의 메모리 채널(402) 내에 있는 DRAM-FPGA(411) 내에 저장되기 때문에, 이에 따라 가속기 로직(410)은 CPU(401)에 의해 직접 액세스될 수 있는 캐시 일관성(Cache-Coherent) 주소 공간 내에 저장된다. 이러한 점에서, 가속기 로직(410)은 OS(405)에 의해 액세스 가능한 캐시 일관성(Cach-Coherent) 주소 공간 내에 저장된다.
- [0040] 도 4를 참조하면, 가상 머신(VMn), 런타임 특징들(409), 가속기 로직(410), 가속기 로직(410)이 저장된 특정 DRAM-FPGA(411), 그리고 가상 머신(VMn)에 대한 DRAM-FPGA 드라이버(412)는 좌측 위부터 우측 아래까지 연장된

교차 해칭(Crosshatching)에 의해 표시되어 있다. 다른 가상 머신들, 그들 각각의 런타임 특징들, 가속기 로직과 DRAM-FPGA 배치들 또한 유사하게 대응되는 교차 해칭에 의해 표시되어 있다. 일 실시 예로서, 가속기 컨트롤러(406)는 가상 머신(VM)과 대응되는 DRAM-FPGA 드라이버(412)의 주소를 가상 머신(VM)에 전달한다. 가상 머신(VM)이 DRAM-FPGA 드라이버(412)에 액세스함에 따라, DRAM-FPGA 드라이버(412)는 DRAM-FPGA(411)에 저장된 가상 머신(VM)에 대한 가속기 로직(410)에 액세스한다.

[0041] 도 5는 본 발명에 따른 가상 머신 또는 어플리케이션의 런타임 특징을 검출하고, 재구성 가능한 DRAM 서브 어레이 내의 가상 머신 또는 어플리케이션에 대응하는 가속기 로직을 저장하는 방법을 보여주는 순서도이다. 도 4를 참조하여, 여기서 가상 머신(VM<sub>n</sub>)이 도 5의 S501 단계에서 실체화된 경우를 가정한다.

[0042] S501 단계에서, 가속기 컨트롤러(406)는 가상 머신(VM<sub>n</sub>)이 가상 머신(VM<sub>n</sub>)과 관련된 런타임 특징들(409)에 기반하여 실체화되었는지 검출한다. S502 단계에서, 가속기 컨트롤러(406)는 가상 머신(VM<sub>n</sub>)에 대한 가속기 로직(410)을 선택하고 회수한다.

[0043] S503 단계에서, 가속기 컨트롤러(406)의 로더(408)는 메모리 채널 저장 명령들을 이용하여 DRAM-FPGA(411) 내에 선택된 가속기 로직(410)을 저장한다. 다른 실시 예에서, S503 단계에서 가속기 컨트롤러(406)는 가상 머신(VM<sub>n</sub>)에 대한 가속기 로직(410)을 확인한다. 그리고, 가상 머신(VM<sub>n</sub>)은 도 4의 502a의 메모리 채널 저장 명령들을 이용하여 DRAM-FPGA(411) 내의 가속기 로직(410)을 회수하고 저장한다.

[0044] S504 단계에서, 가상 머신(VM<sub>n</sub>)은 DRAM-FPGA 드라이버(412)를 통해 DRAM-FPGA(411) 내의 가속기 로직(410)에 액세스한다. 다른 실시 예로서, 시스템 구조(400)는 QPI(QuickPath Interconnect) 환경과 같은 포인트 투 포인트(Point-to-Point) 프로세서 인터커넥트 환경에 적용될 수 있다. 다만, 본 발명은 이에 한정되는 것이 아니다.

[0045] 도 6은 본 발명에 따른 메모리 채널 내에서 하나 또는 하나 이상의 DRAM-FPGA를 사용하는 QPI 기반의 시스템 구조(600)를 보여주는 블록도이다. 도 6을 참조하면, 시스템 구조(600)는 메모리 채널들(602)을 통해 RAM(Random Access Memory)와 같은 시스템 메모리에, 그리고 하나 또는 하나 이상의 대용량 스토리지 장치들(603)에 연결된 두 개의 CPU들(601)을 포함할 수 있다. 시스템 메모리의 RAM은 SRAM 및/또는 DRAM을 포함할 수 있다. 일 실시 예로서, 시스템 메모리는 하나 또는 하나 이상의 DIMM(604)(Dual In-line Memory Module) 내에 구현된 DRAM을 포함할 수 있다. 하나 또는 하나 이상의 DRAM-FPGA(605)는 CPU들(601) 각각의 메모리 채널들(602) 내에서 사용될 수 있다. CPU들(601)은 서로 QPI 기반의 인터커넥트(606)를 통해 연결된다.

[0046] 도 7은 본 발명에 따른 메모리 채널 내에서 하나 또는 하나 이상의 DRAM-FPGA를 포함하는 시스템 구조를 사용할 수 있는 정보 처리 시스템(700)을 보여주는 기능 블록도이다. 정보 처리 시스템(700)은 본 발명에 따라 구성된 하나 또는 하나 이상의 장치들을 포함할 수 있다.

[0047] 다양한 실시 예들에서, 정보 처리 시스템(700)은 컴퓨팅 장치로서 구현될 수 있다. 예를 들어, 컴퓨팅 장치는 랩탑(Laptop), 데스크탑(Desktop), 워크스테이션(Workstation), 서버(Server), 블레이드 서버(Blade Server), 개인용 디지털 보조기(Personal Digital Assistant), 스마트폰, 태블릿(Tablet), 그리고 다른 다양한 컴퓨터들, 또는 가상 머신이나 그것들의 가상 컴퓨팅 장치들을 포함할 수 있다. 다양한 실시 예들에서, 정보 처리 시스템(700)은 유저(미도시)에 의해 사용될 수 있다.

[0048] 정보 처리 시스템(700)은 CPU(Central Processing Unit), 로직, 또는 프로세서(710)를 더 포함할 수 있다. 일부 실시 예들에서, 프로세서(710)는 하나 또는 하나 이상의 FUB(Functional Unit Block) 또는 CLB(715)(Combinational Logic Block)를 포함할 수 있다. 이러한 실시 예에서, CLB는 다양한 부울(Boolean) 로직 연산들(NAND, NOR, NOT, XOR 등), 안정화된 로직 장치들(플립-플롭들, 래치들 등), 다른 로직 장치들, 또는 이들의 조합을 포함할 수 있다. 조합 논리 연산(Combinational Logic Operation)들은 원하는 결과를 얻기 위해 입력 신호들을 처리하기 위한 단순하거나 복잡한 형태로 구성될 수 있다. 여기서 동기화된 조합 논리 연산들의 실시 예들이 기술되었지만, 본 발명은 이에 제한되지 않으며 비 동기화된 조합 논리 연산들 또는 이들의 조합을 포함할 수 있다는 것은 쉽게 이해될 수 있다.

[0049] 일 실시 예에서, 조합 논리 연산들은 복수의 CMOS(Complementary Metal Oxide Semiconductors) 트랜지스터를 포함할 수 있다. 다양한 실시 예들에서, 이러한 CMOS 트랜지스터들은 로직 연산들을 수행하는 게이트(Gate)들로서 배치될 수 있다. 다만, 다른 기술들이 사용될 수 있고, 다른 기술들이 본 발명의 범위 내에 있다는 것은 쉽게 이해될 수 있다. 다른 실시 예들에서, 프로세서(710)를 포함하는 구성 요소들은 본 발명에 따른 가속기 컨트롤러를 구현하는 구성 요소들을 포함할 수 있다.

[0050] 본 발명에 따른 정보 처리 시스템(700)은 RAM과 같은 휘발성 메모리(720)를 더 포함할 수 있다. 휘발성 메모리

(720)는 메모리 채널을 통해 프로세서(710)에 의해 액세스 가능하다. 본 발명에 따른 정보 처리 시스템(700)은 하드 드라이브, 광학 메모리, NAND 또는 플래시 메모리 등과 같은 불휘발성 메모리(730)를 더 포함할 수 있다. 일부 실시 예들에서, 휘발성 메모리(720), 불휘발성 메모리(730), 또는 이들의 조합이나 부분적 조합들 모두는 '기억 매체'라 칭해질 수 있다. 다양한 실시 예들에서, 휘발성 메모리(720) 및/또는 불휘발성 메모리(730)는 반영구적(Semi-Permanent) 또는 실질적으로 영구적인 형태로서 데이터를 저장하도록 구성될 수 있다.

[0051] 다양한 실시 예들에서, 상술한 하나 또는 하나 이상의 RLUT들은 휘발성 메모리(720) 또는 심지어 불휘발성 메모리(730) 내에 포함될 수 있다. 상술한 바와 같이, RLUT는 DRAM 또는 다른 메모리의 일부로서 포함될 수 있다. 상술한 바와 같이, 일부 실시 예들에서, 휘발성 메모리(720) 또는 불휘발성 메모리(730)의 일 부분은 데이터를 저장하는 데에 사용될 수 있고, 다른 부분은 RLUT로서 사용될 수 있다. 일부 실시 예들에서, RLUT는 또한 프로세서 및/또는 로직(710)의 일부로서 여겨질 수 있다. 상술한 바와 같이, RLUT는 하나 또는 하나 이상의 로직 기능들을 수행할 수 있고, 이에 따라 명령들을 수행할 수 있다.

[0052] 다양한 실시 예들에서, 정보 처리 시스템(700)은 하나 또는 하나 이상의 네트워크 인터페이스들(740)을 포함할 수 있다. 네트워크 인터페이스들(740)은 정보 처리 시스템(700)이 통신 네트워크의 일부가 되도록 구성되며, 또한 통신 네트워크를 통해 통신하도록 구성된다. 예를 들어, Wi-Fi 프로토콜은 IEEE(Institute of Electrical and Electronics Engineers) 802.11g, IEEE 802.11n 등을 포함할 수 있다. 예를 들어, 셀룰러(Cellular) 프로토콜은 IEEE 802.16m(Wireless-MAN(Metropolitan Area Network) Advanced으로 알려짐), LTE(Long Term Evolution) Advanced, EDGE(Enhanced Data rates for GSM(Global System for Mobile Communications) Evolution), HSPA+(Evolved High-Speed Packet Access) 등을 포함할 수 있다. 예를 들어, 유선 프로토콜은 IEEE 802.3(Ethernet으로 알려짐), 파이버(Fibre) 채널, 전력 라인 통신(HomePlug, IEEE 1901 등) 등을 포함할 수 있다. 다만, 본 발명은 상술한 예들에 제한되는 것이 아니다.

[0053] 본 발명에 따른 정보 처리 시스템(700)은 유저 인터페이스 유닛(750)을 포함할 수 있다. 유저 인터페이스 유닛(750)은 디스플레이 어댑터(Display Adapter), 햅틱(Haptic) 인터페이스, 휴먼(Human) 인터페이스 장치 등을 포함할 수 있다. 다양한 실시 예들에서, 유저 인터페이스 유닛(750)은 유저로부터 입력을 수신하고 또는 유저에 출력을 제공하도록 구성될 수 있다. 마찬가지로, 다른 형태의 장치들은 유저와의 상호 작용을 제공하기 위해 사용될 수 있다. 예를 들어, 유저에 제공되는 피드백(Feedback)은 시각적 피드백, 청각적 피드백, 또는 촉각적 피드백 등의 어떠한 형태의 감각상의(Sensory) 피드백일 수 있다. 예를 들어, 유저로부터의 입력은 음향, 음성, 또는 촉각 입력 등을 포함하는 어떠한 형태로도 수신될 수 있다.

[0054] 다양한 실시 예들에서, 정보 처리 시스템(700)은 하나 또는 하나 이상의 다른 장치들 또는 하드웨어 구성 요소들(760)을 포함할 수 있다. 하드웨어 구성 요소들(760)은 디스플레이 또는 모니터, 키보드, 마우스, 카메라, 지문 인식기, 비디오 프로세서 등을 포함할 수 있다. 다만, 본 발명은 이에 제한되는 것이 아니다.

[0055] 본 발명에 따른 정보 처리 시스템(700)은 하나 또는 하나 이상의 시스템 버스들(705)을 더 포함할 수 있다. 이러한 실시 예에서, 시스템 버스들(705)은 통신의 관점에서(Communicatively) 프로세서(710), 휘발성 메모리(720), 불휘발성 메모리(730), 네트워크 인터페이스(740), 유저 인터페이스 유닛(750), 그리고 하나 또는 하나 이상의 하드웨어 구성 요소들(760)과 연결되도록 구성될 수 있다. 프로세서(710)에 의해 처리되는 데이터 또는 불휘발성 메모리(730)의 외부로부터의 데이터 입력은 불휘발성 메모리(730) 또는 휘발성 메모리(720) 중 하나의 내부에 저장될 수 있다.

[0056] 다양한 실시 예들에서, 정보 처리 시스템(700)은 하나 또는 하나 이상의 소프트웨어 구성 요소들(770)을 포함하거나 실행할 수 있다. 일부 실시 예들에서, 소프트웨어 구성 요소들(770)은 OS 및/또는 어플리케이션을 포함할 수 있다. 일부 실시 예들에서, OS는 어플리케이션에 하나 또는 하나 이상의 서비스들을 제공하도록 구성되고, 정보 처리 시스템(700)의 다양한 하드웨어 구성 요소들(프로세서(710), 네트워크 인터페이스(740) 등)과 어플리케이션 사이의 중개자(Intermediary)로서 동작하거나 이를 제어하도록 구성될 수 있다. 이러한 실시 예에서, 정보 처리 시스템(700)은 하나 또는 하나 이상의 네이티브(Native) 어플리케이션들을 포함할 수 있다. 네이티브 어플리케이션들은 불휘발성 메모리(730) 등과 같이 지역적으로(Locally) 설치되고 프로세서(710)에 의해 실행되도록 구성되어 OS와 직접 상호 작용할 수 있다. 이러한 실시 예에서, 네이티브 어플리케이션들은 미리 컴파일(Compile)된 기계 실행(Machine-executable) 코드를 포함할 수 있다. 일부 실시 예들에서, 네이티브 어플리케이션들은 스크립트 인터프리터(Script Interpreter)(C shell(csh), AppleScript, AutoHotkey 등) 또는 가상 실행 머신(VM)(Java Virtual Machine, Microsoft Common Language Runtime 등)을 포함할 수 있고, 이러한 스크립트 인터프리터 또는 가상 실행 머신은 소스 코드 또는 오브젝트(Object) 코드를 프로세서(710)에 의해 실행되는 실행

행 가능한 코드로 번역하도록 구성된다. 일부 실시 예들에서, 하나 또는 하나 이상의 소프트웨어 구성 요소들 (770)은 본 발명의 실시 예에 따른 가속기 컨트롤러를 구현하는 실행 가능한 명령어들을 포함할 수 있다.

[0057] 위에서 설명한 내용은 본 발명을 실시하기 위한 구체적인 예들이다. 본 발명에는 위에서 설명한 실시 예들뿐만 아니라, 단순히 설계 변경하거나 용이하게 변경할 수 있는 실시 예들도 포함될 것이다. 또한, 본 발명에는 위에서 설명한 실시 예들을 이용하여 앞으로 용이하게 변형하여 실시할 수 있는 기술들도 포함될 것이다.

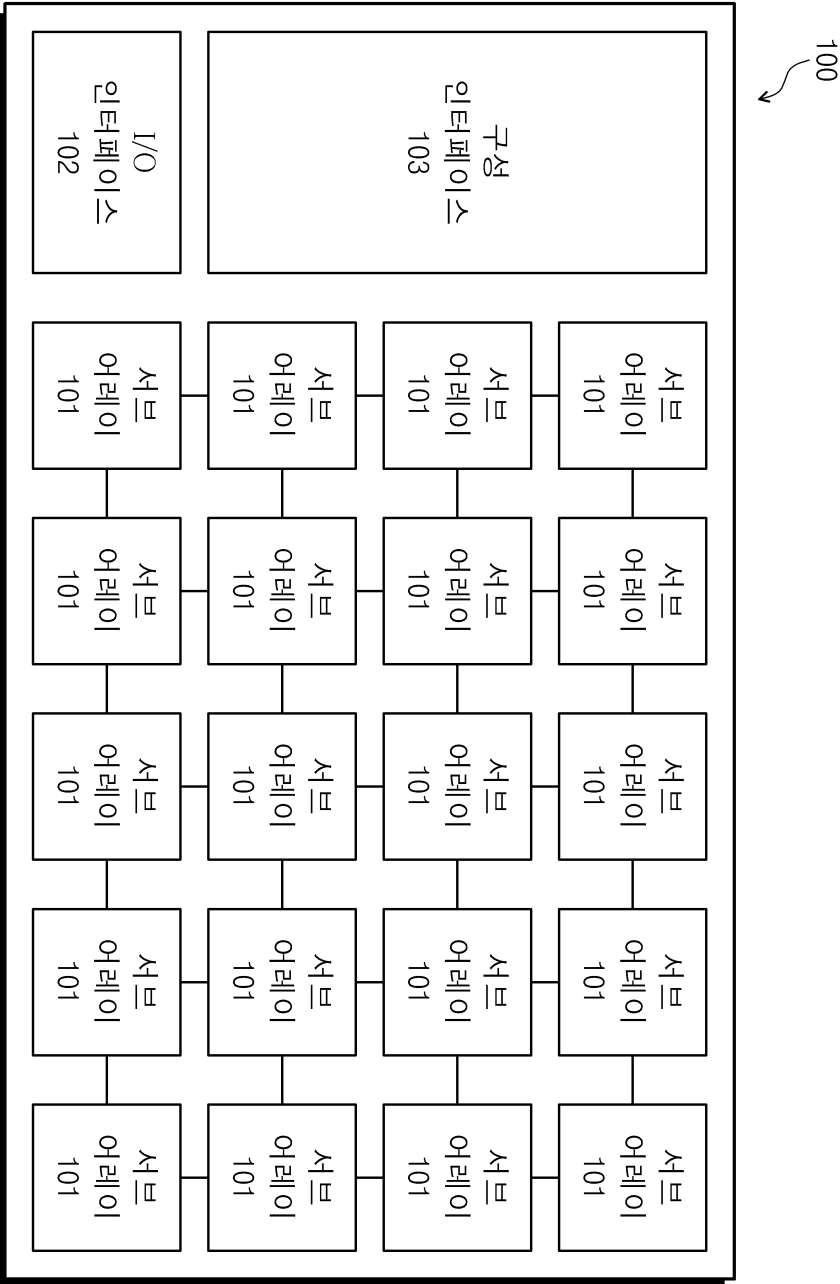
## 부호의 설명

[0058] 100, 200 : 재구성 가능한 메모리 로직 장치

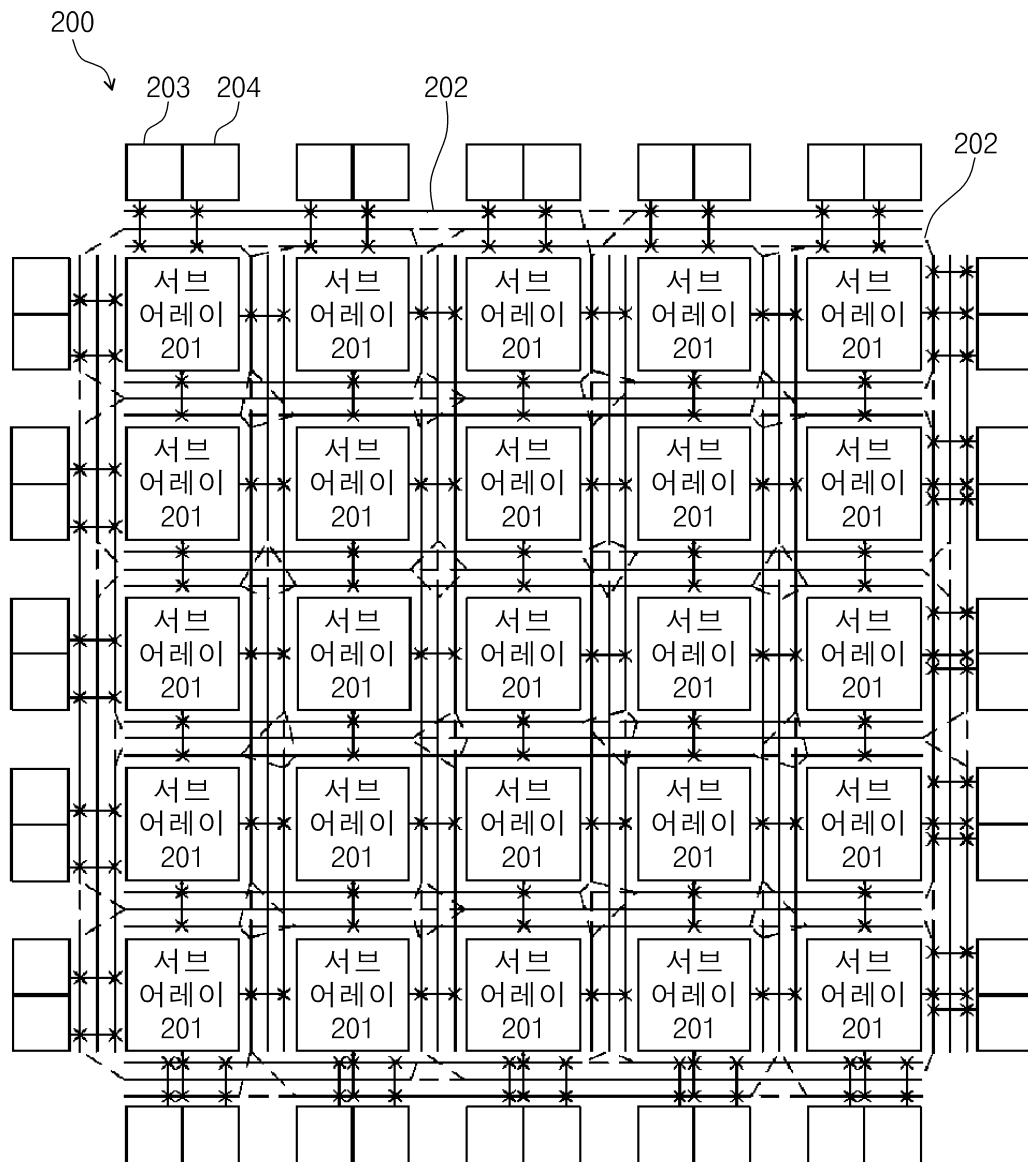
101, 201 : 서버 어레이들	102 : I/O 인터페이스
103 : 구성 인터페이스	202 : 통신 버스들
203 : 입력 신호 패드들	204 : 출력 신호 패드들
300, 310, 320, 330 : DIMM	301 : DRAM-FPGA 모듈들
302 : 정규 DRAM 모듈들	303 : 하이브리드 DRAM-FPGA 모듈들
400 : 시스템 구조	401 : CPU
402 : 메모리 채널	403 : 스토리지 장치
404 : DIMM들	405 : OS
406 : 가속기 컨트롤러	407 : 검출기
408 : 로더	409 : 런타임 특징들
410 : 가속기 로직	411 : DRAM-FPGA
412 : DRAM-FPGA 드라이버	600 : 시스템 구조
601 : CPU들	602 : 메모리 채널들
603 : 스토리지 장치들	604 : DIMM
605 : DRAM-FPGA	606 : 인터커넥트
700 : 정보 처리 시스템	705 : 시스템 버스들
710 : 프로세서	715 : CLB
720 : 휘발성 메모리	730 : 불휘발성 메모리
740 : 네트워크 인터페이스	750 : 유저 인터페이스 유닛
760 : 다른 하드웨어 구성 요소들	770 : 소프트웨어

도면

도면1



도면2

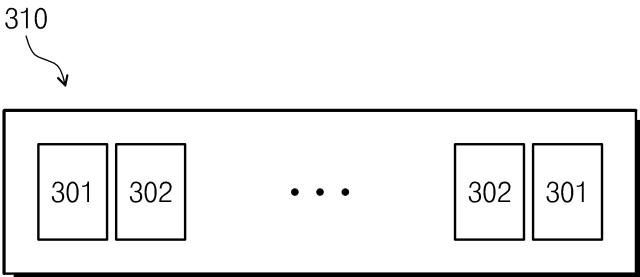


도면3a

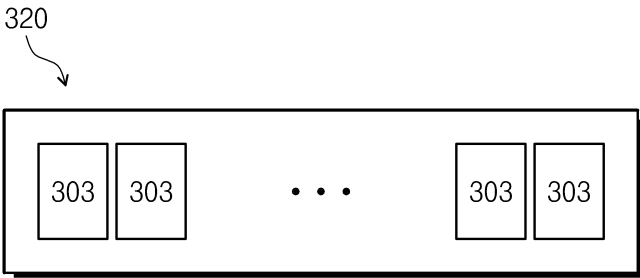




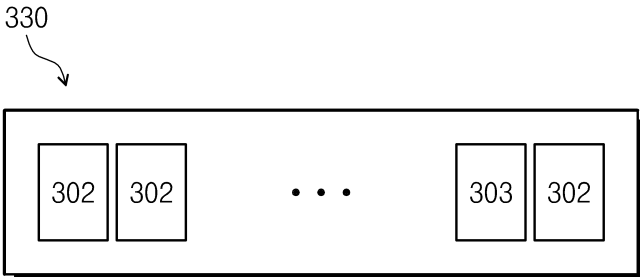
도면3b



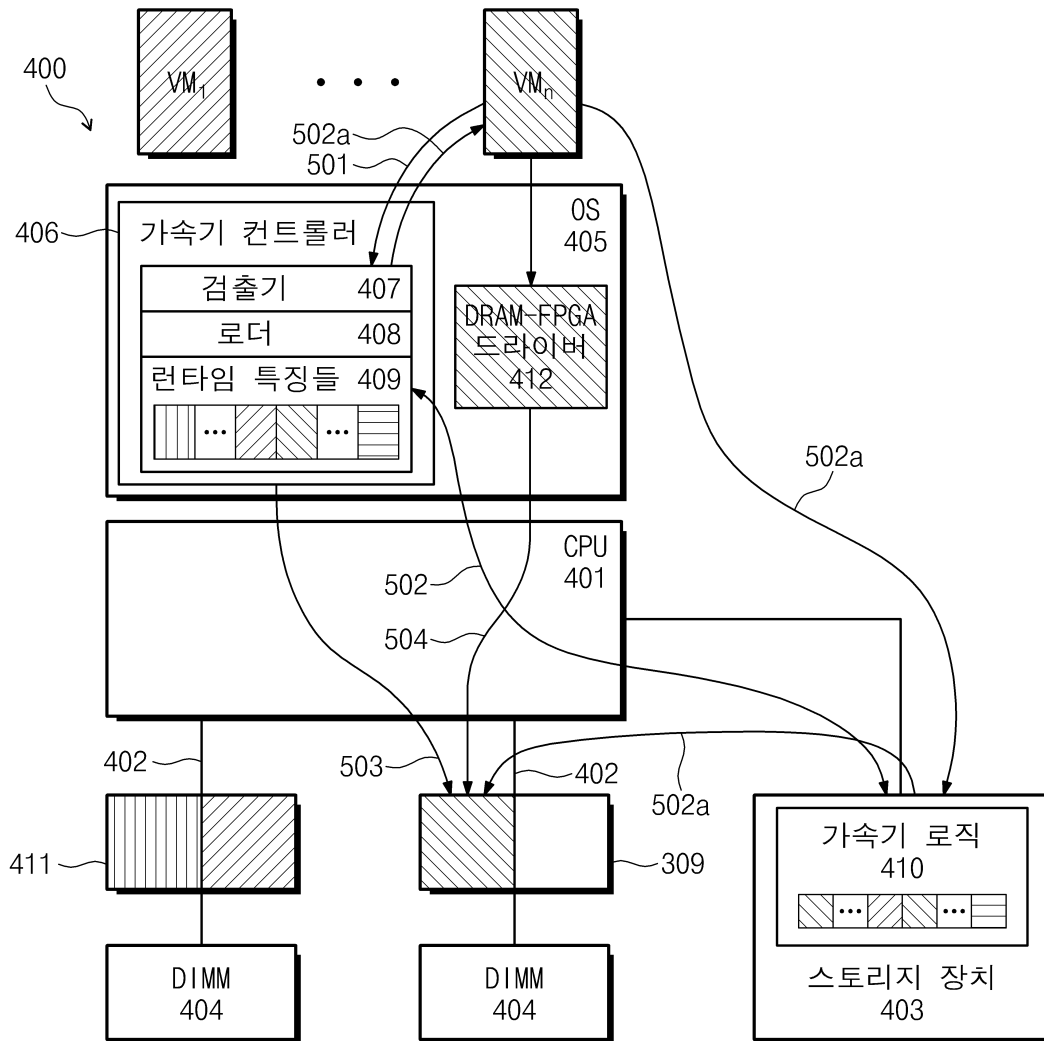
도면3c



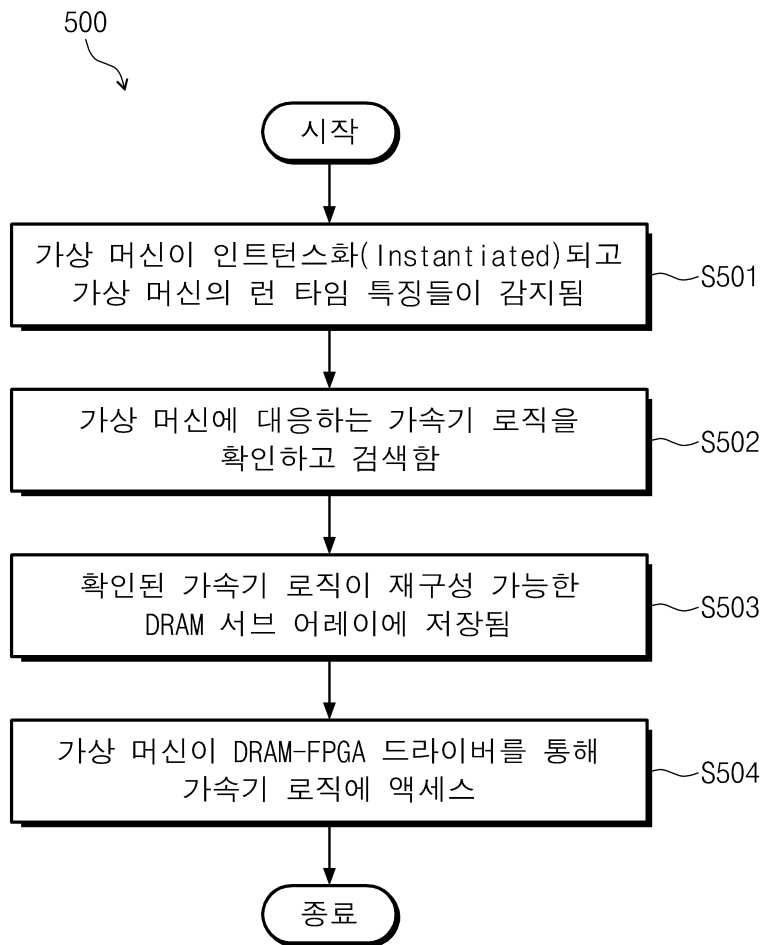
도면3d



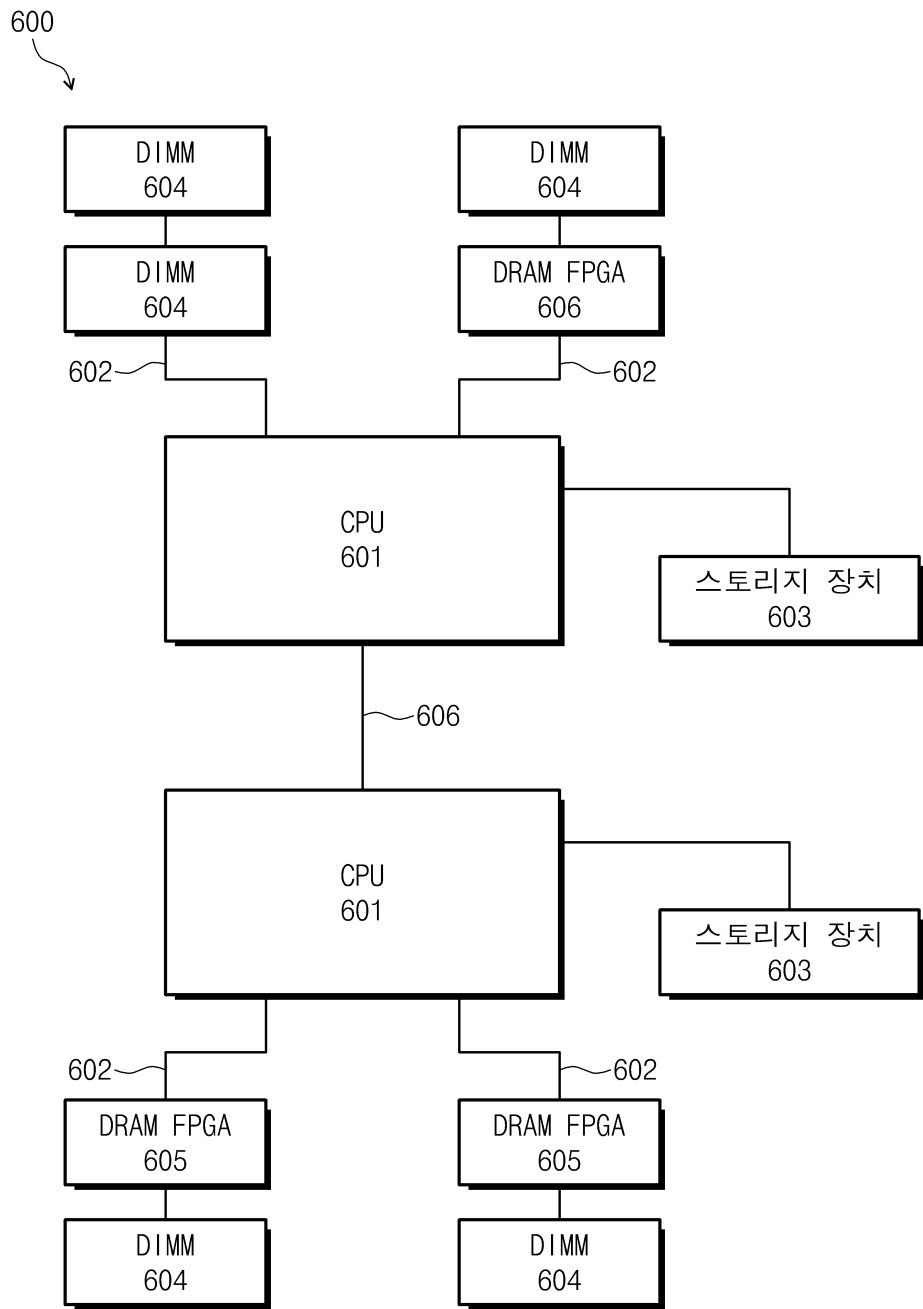
도면4



도면5



도면6



도면7

