



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0075370  
(43) 공개일자 2014년06월19일

(51) 국제특허분류(Int. Cl.)  
G06F 9/46 (2006.01) G06F 12/08 (2006.01)  
G06F 9/44 (2006.01)  
(21) 출원번호 10-2012-0143647  
(22) 출원일자 2012년12월11일  
심사청구일자 없음

(71) 출원인  
한국전자통신연구원  
대전광역시 유성구 가정로 218 (가정동)  
(72) 발명자  
이재진  
대전 유성구 배울2로 6, 107동 1504호 (관평동,  
한화꿈에그린)  
(74) 대리인  
특허법인이상

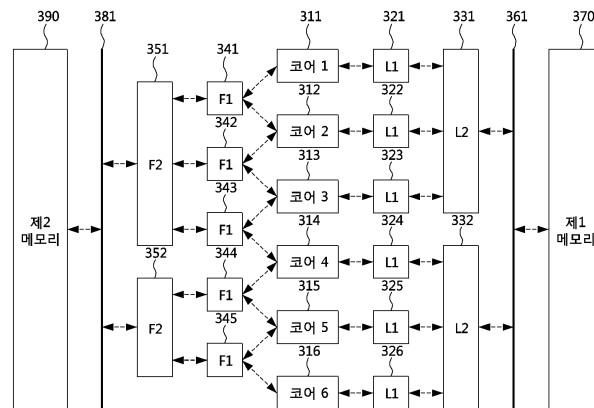
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 계층적 캐시 구조를 가지는 멀티코어 프로세서

**(57) 요약**

계층적 캐시 구조를 가지는 멀티코어 프로세서가 개시된다. 멀티코어 프로세서는 복수의 코어와, 복수의 코어 각각과 독립적으로 연결되는 복수의 제1 캐시와, 복수의 제1 캐시 중 적어도 하나의 제1 캐시와 각각 연결되는 적어도 하나의 제2 캐시와, 복수의 코어 중 적어도 하나의 코어와 각각 연결되는 복수의 제3 캐시 및 복수의 제3 캐시 중 적어도 하나의 제3 캐시와 각각 연결되는 제4 캐시를 포함한다. 따라서, 각 코어간 통신 오버헤드를 감소시킬 수 있고, 데이터 수준 병렬화 처리를 지원함으로써 애플리케이션의 실행 속도를 향상시킬 수 있다.

**대표도**



이 발명을 지원한 국가연구개발사업

과제고유번호 10042395

부처명 지식경제부

연구사업명 산업원천기술개발사업(ETRI지원사업)

연구과제명 다중코어 기반 고성능 SoC의 SW 에뮬레이션 및 Rapid Prototyping 기술 개발

기여율 1/1

주관기관 한국전자통신연구원

연구기간 2012.05.01 ~ 2015.02.28

---

**특허청구의 범위**

**청구항 1**

복수의 코어;  
 상기 복수의 코어 각각과 독립적으로 연결되는 복수의 제1 캐시;  
 상기 복수의 제1 캐시 중 적어도 하나의 제1 캐시와 각각 연결되는 적어도 하나의 제2 캐시;  
 상기 복수의 코어 중 적어도 하나의 코어와 각각 연결되는 복수의 제3 캐시; 및  
 상기 복수의 제3 캐시 중 적어도 하나의 제3 캐시와 각각 연결되는 제4 캐시를 포함하는 멀티코어 프로세서.

**청구항 2**

청구항 1에 있어서, 상기 제1 및 제2 캐시는 상기 복수의 코어에 의해 실행되는 애플리케이션의 처리를 위한 명령어 및 데이터가 저장되고, 상기 제3 및 제4 캐시는 상기 복수의 코어가 애플리케이션을 실행할 때 공유하는 데이터가 저장되는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 3**

청구항 1에 있어서,  
 상기 복수의 제3 캐시 각각은 처리 데이터를 서로 공유하는 적어도 두 개의 코어와 연결되는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 4**

청구항 1에 있어서,  
 상기 복수의 제3 캐시 각각은 서로 인접한 두 코어간에 연결되는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 5**

청구항 1에 있어서,  
 상기 복수의 코어는 상기 제3 캐시 및 상기 제4 캐시 중 상기 제3 캐시를 우선적으로 사용하여 코어간 통신을 수행하는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 6**

청구항 1에 있어서,  
 상기 적어도 하나의 제2 캐시 및 상기 적어도 하나의 제4 캐시는 각각 서로 다른 버스를 통해 서로 다른 메모리와 연결되는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 7**

청구항 1에 있어서,  
 상기 적어도 하나의 제4 캐시는 각각 서로 다른 개수의 제3 캐시와 연결되는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 8**

청구항 1에 있어서,  
 상기 적어도 하나의 제2 캐시 각각은 상기 복수의 코어 중 클러스터링된 코어들별로 각각 연결된 적어도 하나의 제1 캐시에 각각 연결되는 것을 특징으로 하는 멀티코어 프로세서.

**청구항 9**

청구항 1에 있어서,

상기 적어도 하나의 제4 캐시 각각은 상기 복수의 코어 중 클러스터링된 코어들별로 연결된 적어도 하나의 제3 캐시에 각각 연결되는 것을 특징으로 하는 멀티코어 프로세서.

## 명세서

### 기술분야

[0001] 본 발명은 멀티코어 프로세서 기술에 관한 것으로, 더욱 상세하게는 계층적 캐시 구조를 가지는 멀티코어 프로세서에 관한 것이다.

### 배경기술

[0002] 고성능 및 다기능화에 대한 사용자의 요구에 부응하여 스마트폰, 패드형 단말 등과 같은 이동 단말들에 구비된 프로세서는 싱글 코어에서 두 개 이상의 코어를 포함하는 구조로 진화하고 있으며, 프로세서 기술의 발전과 소형화 추세를 고려할 때 쿼드 코어 이상을 포함하는 형태로 발전할 것으로 전망된다. 또한, 차세대 이동 단말들은 수십 내지 수백 개의 프로세서가 집적된 멀티 코어 프로세서를 이용하여 생체 인식이나 증강 현실 등의 서비스들을 가능하게 할 것으로 기대된다.

[0003] 한편, 프로세서의 성능 향상을 위한 방법으로는 클럭 주파수를 높이는 방법이 주로 사용된다. 그러나, 프로세서의 클럭 주파수를 높이면, 이에 상응하여 전력 소모 및 발열이 증가하기 때문에 클럭 주파수를 높이는 방법으로 프로세서의 성능을 향상시키는 것은 한계가 있다.

[0004] 상기한 문제를 해결하기 위해 하나의 프로세서가 복수의 코어를 가지는 멀티 코어 프로세서가 제시되었다. 멀티 코어 프로세서는 각각의 코어가 싱글 코어 프로세서의 동작 클럭 주파수보다 낮은 클럭 주파수에서 동작할 수 있고, 싱글 코어에서 소모되는 전력을 복수의 코어로 분산시킬 수 있기 때문에 싱글 코어 프로세서 보다 처리 효율이 좋은 특징이 있다.

[0005] 멀티 코어 프로세서는 중앙 처리 장치(CPU: Central Processing Unit)가 두 개 이상 들어간 것과 유사하기 때문에 멀티 코어 프로세서를 지원하는 응용 프로그램을 실행할 때, 싱글 코어 프로세서에 비해서 빠르게 작업을 처리할 수 있다. 또한, 멀티 코어 프로세서가 멀티미디어 처리를 기본으로 하는 차세대 이동 단말에 적용될 경우, 동영상의 압축이나 복원, 높은 시스템 사양을 요구하는 게임, 증강 현실 등의 애플리케이션 처리에서 싱글 코어 프로세서에 비해 높은 성능을 나타낼 수 있다.

[0006] 멀티 코어 프로세서에서 고려해야 할 가장 중요한 요소는 데이터 수준 및 기능적 병렬화의 지원과 코어들 사이의 통신 오버헤드를 줄일 수 있는 효율적인 캐시 구조이다.

[0007] 멀티 코어 프로세서 환경에서 처리 효율을 향상시키기 위한 방법으로, 고성능, 고용량의 데이터 캐시를 사용하여 코어간 가능한 많은 데이터를 공유하면서 작업 효율(performance)을 높이고, 통신 오버헤드를 줄이기 위한 방법이 제안되었다. 그러나, 이와 같은 방법은 동영상의 복호화 처리와 같이 복수의 코어가 동일한 정보를 공유하는 경우에는 효율적이지만, 각 코어가 서로 다른 정보를 사용하는 경우에는 효율이 좋지 않은 단점이 있다.

[0008] 또한, 멀티 프로세서 환경에서 병렬처리를 효율적으로 수행하기 위한 방법으로, 정보를 생성하는 정보생성 프로세스와 생성된 정보를 소비하는 정보소비 프로세스가 서로 공유하는 정보를 저장하는 공유 큐(또는 공유 메모리)의 상태에 기초하여 정보소비 프로세스에 할당된 프로세서의 수나 정보 할당단위를 조정하고 정보소비 프로세스가 사용하는 작업 큐의 접근을 적절히 제한하는 방법이 제안되었다. 그러나, 이 방법은 공유 메모리의 감시 및 각 프로세서의 공유 메모리 접근 제어를 위한 추가적인 기능 모듈이 요구되며, 공유 메모리의 접근 제한으로 인해 성능이 저하될 수 있다는 단점이 있다.

## 발명의 내용

### 해결하려는 과제

[0009] 상술한 문제를 해결하기 위한 본 발명의 목적은 복수의 코어간 통신 오버헤드를 줄일 수 있고, 애플리케이션의 처리 성능을 향상시킬 수 있는 계층적 캐시 구조를 가지는 멀티코어 프로세서를 제공하는 것이다.

**과제의 해결 수단**

- [0010] 상술한 본 발명의 목적을 달성하기 위한 본 발명의 일 측면에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서는, 복수의 코어와, 상기 복수의 코어 각각과 독립적으로 연결되는 복수의 제1 캐시와, 상기 복수의 제1 캐시 중 적어도 하나의 제1 캐시와 각각 연결되는 적어도 하나의 제2 캐시와, 상기 복수의 코어 중 적어도 하나의 코어와 각각 연결되는 복수의 제3 캐시 및 상기 복수의 제3 캐시 중 적어도 하나의 제3 캐시와 각각 연결되는 제4 캐시를 포함한다.
- [0011] 여기서, 상기 제1 및 제2 캐시는 상기 복수의 코어에 의해 실행되는 애플리케이션의 처리를 위한 명령어 및 데이터가 저장되고, 상기 제3 및 제4 캐시는 상기 복수의 코어가 애플리케이션을 실행할 때 공유하는 데이터가 저장되도록 구성될 수 있다.
- [0012] 여기서, 상기 복수의 제3 캐시 각각은 처리 데이터를 서로 공유하는 적어도 두 개의 코어와 연결될 수 있다. 또한, 상기 복수의 제3 캐시 각각은 서로 인접한 두 코어간에 연결될 수 있다.
- [0013] 여기서, 상기 복수의 코어는 상기 제3 캐시 및 상기 제4 캐시 중 상기 제3 캐시를 우선적으로 사용하여 코어간 통신을 수행할 수 있다.
- [0014] 여기서, 상기 적어도 하나의 제2 캐시 및 상기 적어도 하나의 제4 캐시는 각각 서로 다른 버스를 통해 서로 다른 메모리와 연결될 수 있다.
- [0015] 여기서, 상기 적어도 하나의 제4 캐시는 각각 서로 다른 개수의 제3 캐시와 연결될 수 있다.
- [0016] 여기서, 상기 적어도 하나의 제2 캐시 각각은 상기 복수의 코어 중 클러스터링된 코어들별로 각각 연결된 적어도 하나의 제1 캐시에 각각 연결될 수 있다.
- [0017] 여기서, 상기 적어도 하나의 제4 캐시 각각은 상기 복수의 코어 중 클러스터링된 코어들별로 연결된 적어도 하나의 제3 캐시에 각각 연결될 수 있다.

**발명의 효과**

- [0018] 상술한 바와 같은 계층적 캐시 구조를 가지는 멀티코어 프로세서에 따르면, 각 코어가 애플리케이션의 실행하기 위한 코드 및 데이터를 저장하는 L1 및 L2 캐시를 계층적으로 구성하고, 각 코어가 애플리케이션을 실행할 때 데이터 공유를 위해 사용하는 F1 및 F2 캐시를 계층적으로 구성한 후, 각 코어가 낮은 수준(level)의 캐시를 우선적으로 사용하여 통신을 수행하고, 필요한 경우 캐시 수준을 단계적으로 높이면서 계층적으로 캐시를 이용한 통신을 수행한다.
- [0019] 따라서, 각 코어간 통신 오버헤드를 감소시킬 수 있고, 데이터 수준 병렬화 처리를 지원함으로써 애플리케이션의 실행 속도를 향상시킬 수 있다.
- [0020] 또한, 코어의 개수가 증가하는 경우에도 본 발명의 일 실시예에 따른 계층적 캐시 구조를 적용함으로써 다양한 멀티코어 환경 또는 다양한 애플리케이션 환경에서 처리 성능을 향상시킬 수 있다.

**도면의 간단한 설명**

- [0021] 도 1은 멀티코어 프로세서 환경에서 데이터 분할을 통한 처리 병렬화 방법을 설명하기 위한 개념도이다.
- 도 2는 멀티코어 프로세서 환경에서 수행되는 동영상 복호화 과정을 나타내는 흐름도이다.
- 도 3은 본 발명의 일 실시예에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서의 구조를 나타내는 블록도이다.
- 도 4는 멀티코어 프로세서 환경에서 실행되는 애플리케이션의 데이터 의존성을 설명하기 위한 개념도이다.
- 도 5는 본 발명의 일 실시예에 따른 계층적 구조를 가지는 멀티코어 프로세서의 데이터 수준 병렬화 처리 방법을 설명하기 위한 개념도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0022] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고

상세하게 설명하고자 한다.

- [0023] 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0024] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0025] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가진 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0026] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 본 발명을 설명함에 있어 전체적인 이해를 용이하게 하기 위하여 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.
- [0027] 본 발명의 일 실시예에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서는 각 코어가 공유하는 캐시를 계층적으로 분할하여 사용하도록 함으로써 애플리케이션의 데이터 수준 병렬화를 실행하고, 각 코어간 통신 오버헤드를 최소화할 수 있도록 한다.
- [0028] 도 1은 멀티코어 프로세서 환경에서 데이터 분할을 통한 처리 병렬화 방법을 설명하기 위한 개념도이다.
- [0029] 도 1을 참조하면, 데이터 분할을 통한 처리 병렬화 방법은 처리하고자 하는 전체 데이터(110)를 복수의 데이터(111 내지 116)로 분할한 후, 분할된 각 데이터(111 내지 116)를 서로 다른 코어(130, 140, 150)에서 수행하도록 하는 방법으로, 분할된 각 데이터간(111 내지 116)의 의존성이 낮은 경우에 효율적으로 적용될 수 있는 병렬화 방법이다.
- [0030] 즉, 도 1에 도시한 바와 같이 멀티코어 프로세서가 3개의 코어(130, 140, 150)로 구성된 경우, 처리하고자 하는 전체 데이터(110)를 데이터 1(111) 내지 데이터 6(116)으로 분할하고, 분할된 데이터 1(111) 및 데이터 4(114)는 코어 1(130)이 처리하고, 분할된 데이터 2(112)는 코어 2(140)가 처리하며, 분할된 데이터 3(113), 데이터 5(115) 및 데이터 6(116)은 코어 3(150)이 처리하도록 함으로써, 각 코어(130, 140, 150)는 서로 다른 데이터를 가지고 동일한 기능을 수행하여 전체적인 처리 속도를 향상시킬 수 있다.
- [0031] 도 2는 멀티코어 프로세서 환경에서 수행되는 동영상 복호화 과정을 나타내는 흐름도로서, 멀티코어 프로세서 환경에서 복수의 코어가 데이터를 분할하여 처리하는 예로 동영상의 복호화 과정을 도시한 것이다.
- [0032] 동영상 복호화 과정에서는 복수의 코어가 처리할 데이터는 프레임(frame), 슬라이스(slice), 매크로 블록(MB: Macro-Block), 블록 단위로 구분될 수 있다.
- [0033] 도 2를 참조하면, 복호화 과정은 입력 스트림 전처리 단계(S201), 가변길이 복호화 단계(S203), 역양자화 및 역이산코사인 변환 단계(S205), 인트라 예측 또는 움직임 보상 단계(S207), 디블록킹 단계(S209) 및 데이터 저장 단계(S211)를 포함할 수 있고, 동영상 복호화 과정의 각 단계에서는 복수의 코어가 각각 동일한 데이터에 대해 동일한 기능을 수행할 수 있다.
- [0034] 입력 스트림 전처리 단계(S201)에서는, 부호기에서 생성된 데이터를 NAL(Network Abstraction Layer) 단위로 입력 버퍼에 저장하고, NAL 단위의 헤더에 포함된 NAL 유형 정보(nal\_unit\_type)를 독출하여 NAL 유형에 따라 나머지 NAL 데이터의 복호화 방법을 결정한다.
- [0035] 가변길이 복호화 단계(S203)에서는, 입력 버퍼에 입력된 데이터에 대한 엔트로피 복호화를 수행하고, 스캔 순서에 따라 엔트로피 복호화된 데이터를 재정렬 한다. 여기서 재정렬된 데이터는 부호기에서 양자화된 데이터이다.

- [0036] 역양자화 및 역이산코사인 변환 단계(S205)에서는 재정렬된 데이터에 대해 역양자화를 수행한 후, 역이산코사인 변환을 수행한다.
- [0037] 인트라 예측 또는 움직임 보상 단계(S207)에서는 역이산코사인 변환이 수행된 데이터(예를 들면, 매크로 블록 또는 블록 데이터)에 대해 인트라 예측 또는 움직임 보상을 수행하여 예측 데이터를 생성한다. 여기서, 생성된 예측 데이터는 역이산코사인 변환된 데이터와 더해진 후, 디블록킹 단계(S209)에서 블록 왜곡 필터링을 수행한 후 복호화된 영상(또는 복원 영상)이 된다. 복원된 영상(또는 복원된 데이터)은 이후의 영상 복호화를 위한 참조 영상으로 사용되기 위해 저장된다(S211).
- [0038] 도 2에 도시한 바와 같은 동영상 복호화 과정에서 각 코어는 서로 다른 데이터(예를 들면, 매크로 블록 또는 블록)에 대해 동일한 기능을 수행함으로써 처리 속도를 향상시킬 수 있다. 그러나, 복수의 코어가 하나의 공유 캐시를 사용하는 경우, 복수의 코어가 공유 캐시를 접근하는 과정에서 발생하는 병목 현상으로 인하여 처리 성능이 저하될 수 있고, 코어의 개수가 증가할수록 코어들간의 통신 오버헤드가 증가하여 전체적인 처리 성능이 저하될 수 있다.
- [0039] 따라서, 멀티코어 프로세서 환경에서 처리 속도를 향상시키기 위해서는 데이터 수준 병렬화의 지원과 이에 따른 각 코어의 효율적인 통신 구조가 요구된다. 본 발명의 일 실시예에 따른 멀티코어 프로세서는 애플리케이션의 실행을 위한 캐시를 별도로 구성하고, 이 캐시들을 계층적으로 구성함으로써 인접 코어간의 통신 오버헤드를 감소시키고, 애플리케이션의 실행시 데이터 수준 병렬화를 지원함으로써 전체적인 처리 성능을 향상시킨다.
- [0040] 도 3은 본 발명의 일 실시예에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서의 구조를 나타내는 블록도로서, 멀티미디어 애플리케이션 중 하나인 동영상 복호화를 수행하는 멀티코어 프로세서의 계층적 캐시 구조를 예를 들어 도시하였다. 도 4는 멀티코어 프로세서 환경에서 실행되는 애플리케이션의 데이터 의존성을 설명하기 위한 개념도이다.
- [0041] 도 3 및 도 4를 참조하면, 본 발명의 일 실시예에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서는 복수의 코어(311 내지 316)와, 복수의 L1 캐시(321 내지 326), L2 캐시(331, 332), F1 캐시(341 내지 345), F2 캐시(351, 352)를 포함할 수 있고, L1, L2, F1 및 F2 캐시는 계층적인 구조로 구성된다.
- [0042] 구체적으로, L1 캐시(321 내지 326) 및 L2 캐시(331, 332)는 애플리케이션의 실행을 위한 코드 및 데이터를 저장하기 위한 캐시 메모리로, 각 L1 캐시(321 내지 326)는 각 코어(311 내지 316)에 독립적으로 할당될 수 있고, L2 캐시(331, 332)는 미리 정해진 개수의 L1 캐시와 연결되도록 구성될 수 있다. 또는, 각 L2 캐시는 클러스터링(clustering)된 코어들별로 연결될 수 있도록 클러스터링된 코어들 각각에 연결된 L1 캐시들과 연결될 수 있다.
- [0043] 예를 들어, 코어 1(311), 코어 2(312) 및 코어 3(313)이 클러스터링되고, 코어 4(314), 코어 5(315) 및 코어 6(316)이 클러스터링된 경우, L2 캐시(331)는 클러스터링된 코어 1, 2 및 3(311 내지 313)과 각각 연결된 L1 캐시들(321 내지 323)과 연결되고, L2 캐시(332)는 클러스터링된 코어 4(314), 5(315) 및 6(316)과 각각 연결된 L1 캐시들(324 내지 326)과 연결될 수 있다.
- [0044] 각각의 L1 캐시(321 내지 326)는 연결된 각 코어(311 내지 316)에 의해 자주 반복되는 연산을 처리하기 위한 저장공간으로, 각 코어(311 내지 316)가 처리할 명령 또는 처리 직전의 데이터를 저장하기 위한 공간으로 사용될 수 있다. 또한, L2 캐시(331, 332)는 각 코어(311 내지 316)가 연결된 해당 L1 캐시(321 내지 326)를 이용하여 데이터를 처리하는 동안 해당 코어가 이후에 처리할 데이터를 미리 저장하는 공간으로 사용될 수 있다.
- [0045] 각 L1 캐시(321 내지 326)의 크기는 서로 동일하게 구성될 수도 있고 각 코어별로 서로 상이한 크기를 가지도록 구성될 수도 있다. 또한, 각 L2 캐시(331, 332)에 연결된 복수의 L1 캐시의 수는 서로 동일하게 구성될 수도 있고, 서로 다르게 구성될 수도 있다. 각 L2 캐시(331, 332)는 예를 들어 2개 내지 10개의 L1 캐시와 연결될 수 있다.
- [0046] 도 3에 도시한 바와 같이 L2 캐시(331)는 3개의 L1 캐시(321, 322, 323)와 연결되고, L2 캐시(332)는 3개의 L1 캐시(324, 325, 326)와 연결되도록 구성될 수도 있고, L2 캐시(331)와 L2 캐시(332)가 서로 다른 개수의 L1 캐시와 연결되도록 구성될 수도 있다. 또한, 각 L2 캐시(331, 332)는 각 L1 캐시(321 내지 316)의 크기보다 더 큰 크기로 구성될 수 있고, 각각의 L2 캐시(331, 332)의 크기는 서로 동일할 수도 있고, 서로 다른 크기를 가지도록 구성될 수도 있다.



- [0047] 또한, 각 L2 캐시(331, 332)는 제1 버스(361)를 통해 제1 메모리(370)와 연결될 수 있다. 여기서, 제1 메모리(370)는 애플리케이션의 실행을 위한 명령어 및 데이터를 저장하기 위한 메모리로 사용된다.
- [0048] 한편, 멀티코어 프로세서 환경에서 각 코어가 병렬로 처리를 수행하는 경우 데이터의 의존성을 고려해야 한다.
- [0049] 예를 들어, 멀티코어 프로세서가 동영상의 복호화를 수행하는 경우, 도 4에 도시한 바와 같이 현재의 매크로 블록에 대한 화면내 예측을 수행하기 위해서는 좌측(left, 421), 상단(up, 422), 좌측상단(upper left, 423) 및 우측 상단(upper right, 424)에 위치하는 매크로 블록들을 참조해야 하기 때문에 상기 매크로 블록(421 내지 424)들의 처리가 선행되어야 한다.
- [0050] 또한, 멀티코어 프로세서 환경에서 데이터 수준 병렬화를 통해 동영상의 복호화를 수행하는 경우, 동일한 행(row)에 위치하는 매크로 블록들은 동일한 코어가 처리하기 때문에 기본적으로 데이터의 공유가 행해진다. 그러나, 인접한 행은 서로 다른 코어가 처리하기 때문에 인접한 두 코어들 사이에 데이터를 효율적으로 공유하기 위한 방법이 요구된다.
- [0051] 예를 들어, 도 4에서 N-1 행에 위치한 매크로 블록들은 코어 1이 처리하고, N 행에 위치한 매크로 블록들은 코어 2가 처리를 수행하는 경우, 코어 2가 현재 매크로 블록(410)의 복호화 처리를 수행하기 위해서는 코어 1에 의해 복호화가 수행되는 N-1 행의 매크로 블록들의 복호화 결과를 참조해야 하기 때문에 코어 1과 코어 2 사이에는 데이터의 공유가 필요하게 된다.
- [0052] 본 발명의 일 실시예에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서는 상기한 바와 같은 요구를 만족시키기 위하여 각 코어간 공유할 수 있고, 계층적 구조로 구성되는 F1 캐시(341 내지 345) 및 F2 캐시(351, 352)를 포함한다.
- [0053] 구체적으로, F1 캐시(341 내지 345)는 데이터 수준 병렬성을 지원하는 멀티코어 프로세서 환경에서 복수의 코어가 서로 처리 데이터를 공유하기 위한 캐시로서, 서로 인접한 두 개의 코어가 하나의 F1 캐시와 연결되도록 구성될 수도 있고, 서로 인접한 위치에 있지는 않아서 처리 데이터를 공유하는 복수의 코어가 하나의 F1 캐시에 연결되도록 구성될 수도 있다. 여기서, 각각의 F1 캐시(341 내지 345)는 서로 동일한 크기로 구성될 수도 있고, 각 코어(311 내지 316)가 연결되는 코어에 따라 서로 다른 크기로 구성될 수도 있다.
- [0054] 각 F2 캐시(351, 352)는 복수의 F1 캐시(예를 들면, 2 내지 10개의 F1 캐시)와 연결되도록 구성됨으로써 서로 인접한 코어가 아닌 경우에도 클러스터링된 코어들 사이의 효율적인 데이터 공유를 지원하기 위해 사용된다. 예를 들어, 코어 1(311), 코어 2(312) 및 코어 3(313)이 클러스터링되고, 코어 4(314), 코어 5(315) 및 코어 6(316)이 클러스터링된 경우, F2 캐시(351)는 클러스터링된 코어들(311, 312, 313)과 연결된 F1 캐시들(341, 342, 343)과 연결되도록 구성될 수 있고, F2 캐시(352)는 클러스터링된 코어들(314, 315, 316)과 연결된 F1 캐시들(344, 345)와 연결되도록 구성될 수 있다.
- [0055] 각각의 F2 캐시(351, 352)는 서로 동일한 크기로 구성될 수도 있고, 서로 다른 크기로 구성될 수도 있다. 또한, 각 F2 캐시(351, 352)와 연결된 F1 캐시의 수는 서로 동일하게 구성될 수도 있고, 서로 다르게 구성될 수도 있다.
- [0056] 멀티코어 프로세서가 동영상 부호화 또는 동영상 복호화를 수행하는 경우, F1 캐시(341 내지 345) 및 F2 캐시(351, 352)는 인접 코어간 부호화 또는 복호화할 데이터(예를 들면, 매크로블록 데이터)를 공유하기 위하여 사용될 수 있다.
- [0057] 또한, 각 F2 캐시(351, 352)는 제2 버스(381)를 통해 제2 메모리(390)와 연결될 수 있다. 여기서, 제2 메모리(390)는 애플리케이션의 실행시 사용되는 소스 데이터를 저장하기 위한 메모리로 사용될 수 있다. 예를 들어, 멀티 코어 프로세서가 동영상의 부호화 또는 복호화를 실행하는 경우, 제2 메모리는 동영상의 부호화 또는 복호화 과정에서 필요한 프레임 데이터를 저장하기 위한 용도로 사용될 수 있다.
- [0058] 도 3에 도시한 바와 같이 본 발명의 일 실시예에 따른 계층적 캐시 구조를 가지는 멀티코어 프로세서는 캐시를 사용 용도 또는 데이터 공유 여부에 따라 따라 L1 캐시(321 내지 326), L2 캐시(331, 332)와, F1 캐시(341 내지 345), F2 캐시(351, 352)로 분리하여 구성하고, L1, L2 캐시 및 F1, F2 캐시를 계층적으로 구성한 후, 각 코어가 낮은 수준(level)의 캐시를 우선적으로 사용하여 통신을 수행하고 캐시 수준을 한 단계씩 올리면서 계층적으로 통신을 수행함으로써 통신 오버헤드를 줄이면서 멀티미디어 애플리케이션의 처리 성능을 향상시킬 수 있다.
- [0059] 도 3에서는 6개의 코어(311 내지 316)와, 6개의 L1 캐시(321 내지 326), 2개의 L2 캐시(331, 332), 5개의 F1



캐시(341 내지 3345) 및 2개의 F2 캐시(351, 352)를 포함하는 멀티코어 프로세서의 계층적 캐시 구조를 예를 들어 도시하였으나, 본 발명의 기술적 사상이 도 3에 도시한 멀티코어 프로세서의 구조에만 한정되는 것은 아니며, 본 발명의 기술적 사상은 다양한 개수의 코어를 포함하는 멀티코어 프로세서 환경에서 사용 용도 또는 코어간 데이터 공유 여부에 따라 캐시를 구분하고, 구분된 캐시를 계층적으로 구성하는 다양한 형태의 멀티코어 프로세서의 구성을 모두 포함할 수 있다.

- [0060] 도 5는 본 발명의 일 실시예에 따른 계층적 구조를 가지는 멀티코어 프로세서의 데이터 수준 병렬화 처리 방법을 설명하기 위한 개념도이다.
- [0061] 도 5에서는 6개의 코어를 구비한 멀티코어 프로세서가 720×480의 해상도를 가지는 영상 프레임을 복호화하는 과정을 예를 들어 도시하였다.
- [0062] 이하, 도 3 및 도 5를 참조하여 멀티코어 프로세서의 데이터 수준 병렬화 처리 방법을 설명한다.
- [0063] 먼저, 720×480의 해상도를 가지는 영상 프레임이 순차적으로 제공되면, 720×480의 해상도를 가지는 영상 프레임은 16×16 픽셀 크기를 가지는 매크로 블록(MB) 45×30개로 구분되고, 각 코어(311 내지 316)는 자신에게 배치된 특정 행에 위치하는 매크로 블록들에 대한 복호화를 수행한다.
- [0064] 예를 들어, 멀티코어 프로세서가 6개의 코어(311 내지 316)로 구성된 경우, 코어 1(311)은 45개의 행으로 구성된 매크로 블록들 중 1, 7, 13, 19 및 25행에 위치하는 매크로 블록들의 가변길이 복호화를 수행하여 양자화된 데이터 및 복호화를 위한 파라미터를 획득한다.
- [0065] 또한, 코어 2(312)는 45개의 행으로 구성된 매크로 블록들 중 2, 8, 14, 20 및 26행에 위치하는 매크로 블록들의 가변길이 복호화를 수행한다.
- [0066] 즉, 코어 1(311) 및 코어 2(312)는 서로 인접한 행(예를 들면, 1행 및 2행 또는 7행 및 8행 등)에 위치하는 행들에 대한 가변길이 복호화를 수행한다. 여기서, 740×480의 해상도를 가지는 영상 프레임은 제2 메모리(390)에 저장될 수 있고, 45×30개의 매크로 블록 중 서로 인접한 적어도 두 행 이상의 매크로 블록들은 F2(351) 캐시에 저장될 수 있다. 또한, F2 캐시(351)에 저장된 복수의 매크로 블록들 중 각 코어(311, 312)가 복호화를 처리하는 현재 매크로 블록의 데이터 및/또는 복호화가 수행된 적어도 하나 이상의 매크로 블록들에 대한 복호화 데이터는 F1 캐시(341, 342) 또는 F2 캐시(351)에 저장되어 인접 매크로 블록을 수행하는 다른 코어에 의해 참조되도록 구성될 수 있다.
- [0067] 또한, 코어 3(313)은 45개의 행으로 구성된 매크로 블록들 중 코어 2(312)가 복호화를 수행하는 매크로 블록들이 위치하는 행의 바로 다음 행인 3, 9, 15, 21 및 27행에 위치하는 매크로 블록들의 가변길이 복호화를 수행하여 양자화된 데이터 및 복호화를 위한 파라미터를 획득한다. 여기서, 코어 3(313)은 F1 캐시(342)에 저장된 복호화된 데이터를 참조하여 복호화를 수행할 수 있고, 복호화가 수행된 매크로 블록의 데이터를 코어 4(314)가 담당하는 매크로 블록의 복호화시 참조할 수 있도록 F1 캐시(343)에 저장할 수 있다.
- [0068] 코어 4(314)는 45개의 행으로 구성된 매크로 블록들 중 코어 3(313)가 복호화를 수행하는 매크로 블록들이 위치하는 행의 바로 다음 행인 4, 10, 16, 22 및 28행에 위치하는 매크로 블록들의 가변길이 복호화를 수행하여 양자화된 데이터 및 복호화를 위한 파라미터를 획득한다. 여기서, 코어 4(314)는 F1 캐시(343)에 저장된 복호화된 데이터를 참조하여 복호화를 수행할 수 있고, 복호화가 수행된 매크로 블록의 데이터를 코어 5(315)가 담당하는 매크로 블록의 복호화시 참조할 수 있도록 F1 캐시(344)에 저장할 수 있다.
- [0069] 코어 5(315)는 45개의 행으로 구성된 매크로 블록들 중 코어 4(314)가 복호화를 수행하는 매크로 블록들이 위치하는 행의 바로 다음 행인 5, 11, 17, 23 및 29행에 위치하는 매크로 블록들의 가변길이 복호화를 수행하여 양자화된 데이터 및 복호화를 위한 파라미터를 획득한다. 여기서, 코어 5(315)는 F1 캐시(344)에 저장된 복호화된 데이터를 참조하여 복호화를 수행할 수 있고, 복호화가 수행된 매크로 블록의 데이터를 코어 6(316)가 담당하는 매크로 블록의 복호화시 참조할 수 있도록 F1 캐시(345)에 저장할 수 있다.
- [0070] 코어 6(316)은 45개의 행으로 구성된 매크로 블록들 중 코어 5(315)가 복호화를 수행하는 매크로 블록들이 위치하는 행의 바로 다음 행인 6, 12, 18, 24 및 30행에 위치하는 매크로 블록들의 가변길이 복호화를 수행하여 양자화된 데이터 및 복호화를 위한 파라미터를 획득한다. 여기서, 코어 6는 F1 캐시(345)에 저장된 복호화된 데이터를 참조하여 복호화를 수행할 수 있다.

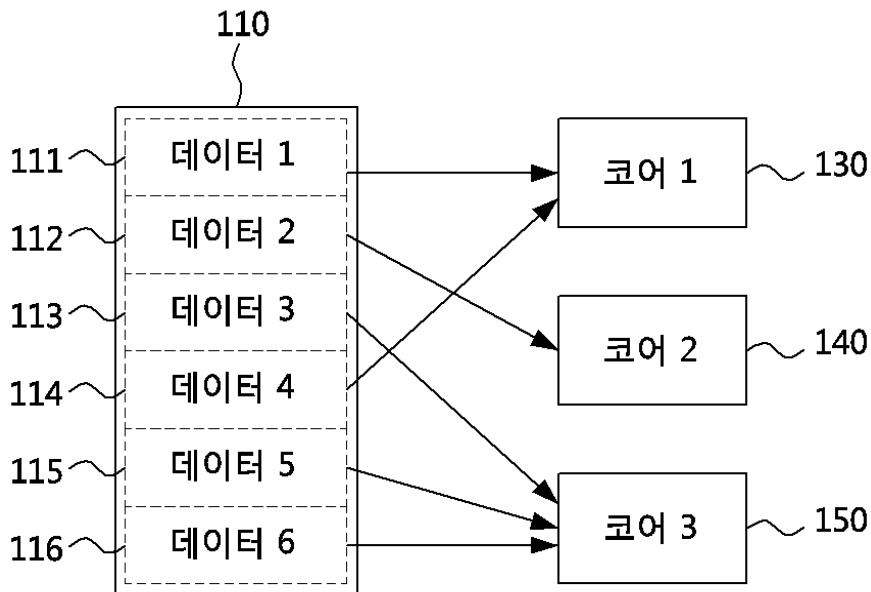
[0071] 이상 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

**부호의 설명**

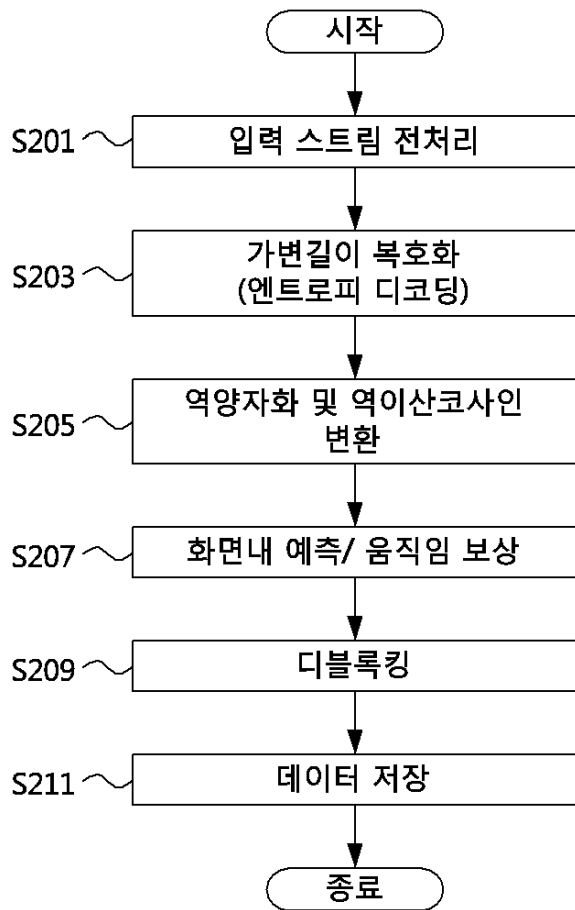
- [0072]
- |                   |                  |
|-------------------|------------------|
| 311 : 코어 1        | 312 : 코어 2       |
| 313 : 코어 3        | 314 : 코어 4       |
| 315 : 코어 5        | 316 : 코어 6       |
| 321 ~ 326 : L1 캐시 | 331, 332 : L2 캐시 |
| 341 ~ 345 : F1 캐시 | 351, 352 : F2 캐시 |
| 361 : 제1 버스       | 370 : 제1 메모리     |
| 381 : 제2 버스       | 390 : 제2 메모리     |

**도면**

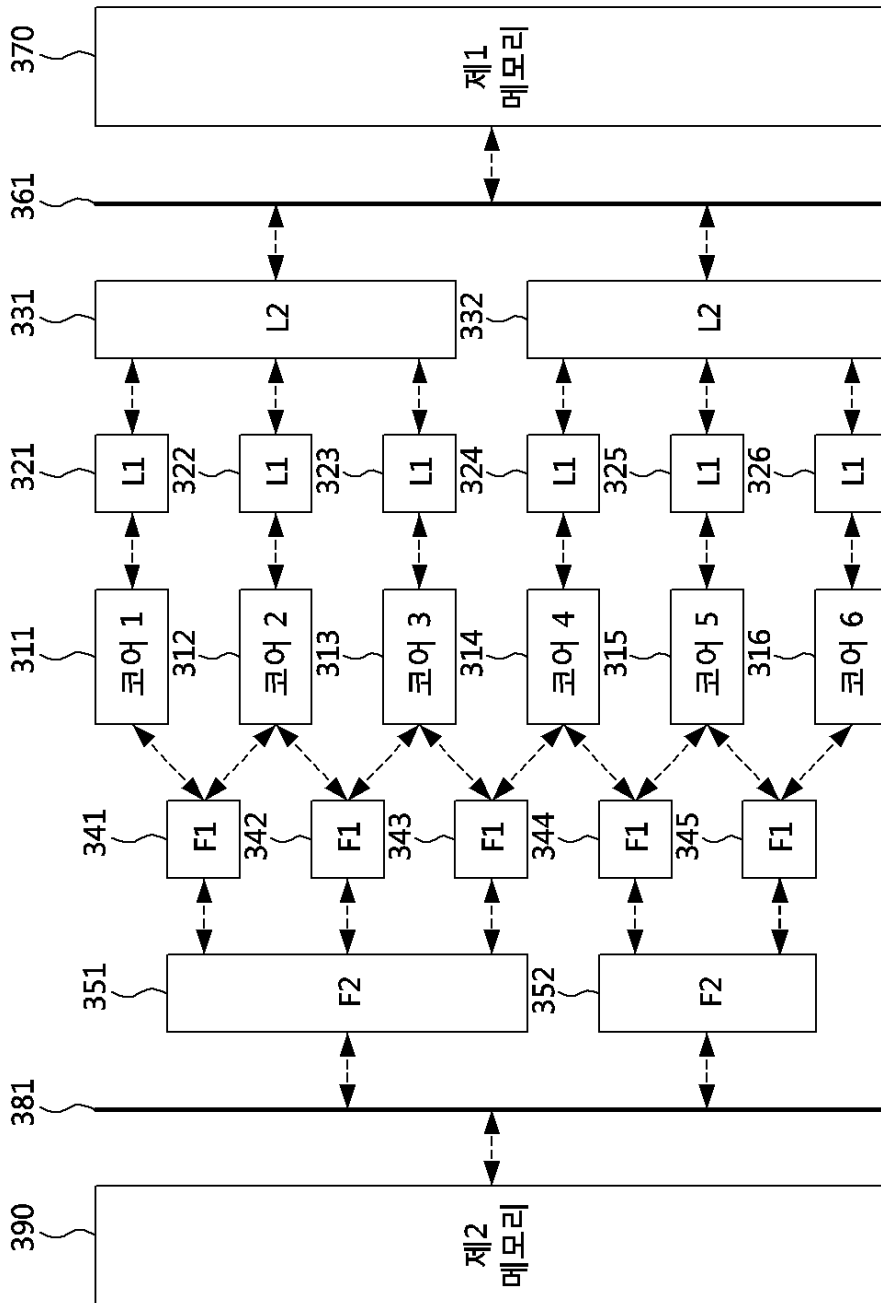
**도면1**



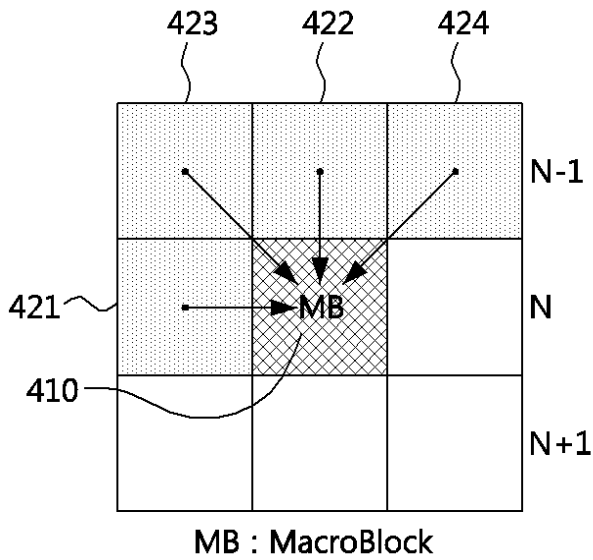
도면2



도면3



도면4



도면5

