



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년02월05일
(11) 등록번호 10-0801630
(24) 등록일자 2008년01월30일

(51) Int. Cl.

G06F 9/06 (2006.01)

(21) 출원번호 10-2007-0058669

(22) 출원일자 2007년06월15일

심사청구일자 2007년06월15일

(56) 선행기술조사문헌

KR 10-2000-0072875 A

(뒷면에 계속)

(73) 특허권자

디비코 주식회사

경기도 성남시 분당구 구미동 192-2 온세통신사옥 4층

(72) 발명자

기창서

경기 용인시 수지구 죽전동 죽전택지개발 11B 광명샤인빌 403동403호

(74) 대리인

김도형

전체 청구항 수 : 총 8 항

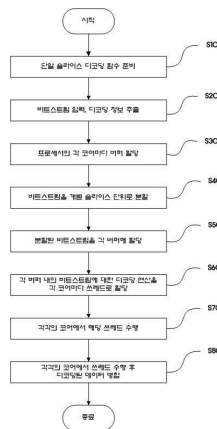
심사관 : 노지명

(54) 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치 및 방법

(57) 요약

본 발명은 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치 및 방법에 관한 것으로, 특히 эм팩 데이터의 비트 스트림을 슬라이스 단위로 분할하고 각 슬라이스에 대한 디코딩 작업을 프로세서 내의 각 코어마다 쓰레드로 분산시켜 개별적으로 디코딩한 후 이를 병합하는 분산 디코딩 처리 기술에 관한 것이다. 본 발명에 따르면, 복잡한 연산을 거치지 않고도 эм팩 데이터의 디코딩 작업을 여러 개의 디코딩 쓰레드로 분할할 수 있으며, эм팩 디코딩 처럼 연산량이 많은 작업을 처리할 때 멀티코어 프로세서 내의 일부 CPU에 연산량이 과중되는 현상을 방지할 수 있으며, 멀티코어 프로세서 내의 여러 CPU 자원을 효율적으로 활용할 수 있는 효과가 있다.

대표도 - 도2



(56) 선행기술조사문헌

KR 10-1995-0002475 A

KR 10-1998-0007751 A

KR 10-1999-0010657 A

KR 10-1999-0075517 A

KR 10-2001-0023653 A

KR 10-2004-0014604 A

KR 10-2007-0049226 A

roc. of 15th IPDPS 2000 Workshops (Bhandarker S.M. 외 1인 공저, "Parallel Parsing of MPEG Video in a Multi-threaded Multiprocessor Environment" , LNCS Vol. 1800, pp. 194-201, 2000 출간)

KR 10-2007-0037427 A

특허청구의 범위

청구항 1

쓰레드 단위로 처리하는 복수 개의 코어를 구비한 멀티코어 프로세서;

엠펙 데이터를 입력받아 디코딩 정보 추출 후 개별 슬라이스 단위로 분할하고, 하나의 슬라이스를 개별적으로 디코딩하는 단일 슬라이스 디코딩 함수를 생성하여 제공함으로써 상기 분할된 개별 슬라이스를 대입하여 디코딩 연산을 수행하는 쓰레드가 상기 멀티코어 프로세서 내의 각 코어마다 분산 배치되도록 하는 엠펙 데이터 분할 모듈;

상기 엠펙 데이터 분할 모듈로부터 개별 슬라이스를 입력받아 저장하고, 상기 저장된 슬라이스를 상기 멀티코어 프로세서 내의 각 코어마다 제공하는 버퍼 영역을 구비하는 메모리; 및

상기 멀티코어 프로세서의 각 코어를 통해 디코딩된 데이터를 병합하는 디코딩 병합 모듈;

를 포함하여 구성되는 분산 디코딩 처리 장치.

청구항 2

제 1 항에 있어서,

상기 엠펙 데이터 분할 모듈은,

상기 엠펙 데이터를 비트스트림으로 입력받아 디코딩 정보를 추출하고, 상기 메모리 내의 영역을 상기 멀티코어 프로세서의 각 코어에 대응되는 복수 개의 버퍼로 분할 할당하는 헤더 파싱부;

상기 비트스트림 내의 슬라이스 시작 코드를 감지하여 개별 슬라이스 단위로 분할하는 슬라이스 분할부;

상기 분할된 슬라이스 단위의 비트스트림을 상기 버퍼마다 할당하여 입력하는 분산 배치부; 및

하나의 슬라이스를 개별적으로 디코딩하는 단일 슬라이스 디코딩 함수를 생성하여 상기 멀티코어 프로세서의 각 코어에 제공하는 단일 슬라이스 디코딩 함수 생성부;

를 포함하여 구성되는 것을 특징으로 하는 분산 디코딩 처리 장치.

청구항 3

제 2 항에 있어서,

상기 분산 배치부는 상기 슬라이스 단위의 비트스트림을 각 코어의 해당 버퍼마다 동일한 분량으로 할당하여 순차적으로 입력하는 것을 특징으로 하는 분산 디코딩 처리 장치.

청구항 4

제 2 항에 있어서,

상기 엠펙 데이터 분할 모듈은,

각 코어의 연산 점유량을 체크하는 코어 연산량 측정부를 더 포함하여 구성되고,

상기 분산 배치부는 상기 슬라이스 단위의 비트스트림을 상기 각 코어의 연산 점유량에 반비례하도록 할당하여 각 코어의 해당 버퍼에 입력하는 것을 특징으로 하는 분산 디코딩 처리 장치.

청구항 5

삭제

청구항 6

복수 개의 코어를 구비한 프로세서를 통해 엠펙 데이터를 디코딩하는 방법에 있어서,

(a) 엠펙 데이터 내의 각 슬라이스를 개별적으로 디코딩하는 단일 슬라이스 디코딩 함수를 준비하는 단계;

(b) 엠펙 데이터를 비트스트림으로 입력받고 상기 엠펙 데이터 내의 디코딩 정보를 추출하는 단계;

- (c) 프로세서의 코어 개수만큼 메모리 내의 영역을 분할하여 상기 코어 개수만큼의 버퍼를 준비하고 각 코어마다 할당하는 단계;
 - (d) 상기 비트스트림을 개별 슬라이스 단위로 분할하는 단계;
 - (e) 상기 분할된 슬라이스 단위의 비트스트림을 상기 버퍼마다 할당하여 입력하는 단계;
 - (f) 상기 버퍼마다 입력된 슬라이스 단위의 비트스트림에 대한 디코딩 연산을 해당 코어에 스레드로 할당하는 단계;
 - (g) 상기 각 코어가 상기 단일 슬라이스 디코딩 함수를 사용하여 디코딩 스레드를 수행하는 단계; 및
 - (h) 모든 코어의 디코딩 스레드가 완료되면 상기 디코딩된 데이터를 병합하는 단계;
- 를 포함하여 구성되는 분산 디코딩 처리 방법.

청구항 7

제 6 항에 있어서,
 상기 (d) 단계는 상기 비트스트림 내의 슬라이스 시작 코드를 감지하여 각 슬라이스를 구분하고 분할하는 단계를 포함하여 구성되는 것을 특징으로 하는 분산 디코딩 처리 방법.

청구항 8

제 7 항에 있어서,
 상기 (e) 단계는,
 각 코어의 연산 점유량을 체크하는 단계; 및
 상기 슬라이스 단위의 비트스트림을 상기 각 코어의 연산 점유량에 반비례하도록 할당하여 각 코어의 해당 버퍼에 입력하는 단계;
 를 더 포함하여 구성되는 분산 디코딩 처리 방법.

청구항 9

제 7 항에 있어서,
 상기 (e) 단계는 상기 슬라이스 단위의 비트스트림을 각 코어의 해당 버퍼마다 동일한 분량으로 할당하여 순차적으로 입력하는 단계를 더 포함하여 구성되는 분산 디코딩 처리 방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <5> 본 발명은 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치 및 방법에 관한 것으로, 특히 압축 데이터의 비트스트림을 슬라이스 단위로 분할하고 각 슬라이스에 대한 디코딩 작업을 프로세서 내의 각 코어마다 스레드로 분산시켜 개별적으로 디코딩한 후 이를 병합하는 분산 디코딩 처리 기술에 관한 것이다.
- <6> 멀티코어 프로세서는 2개 이상의 CPU가 붙어 있는 집적회로를 의미하며, 단일 CPU의 동작 주파수를 높이는데 있어서 많은 한계점이 도출되어 최근에는 이처럼 2개 이상의 CPU를 사용함으로써 단일 프로세서의 주파수 한계를 극복할 수 있는 멀티코어 프로세서가 널리 쓰이고 있다.
- <7> 멀티코어 프로세서의 동작 단위는 스레드이며, 멀티코어 프로세서는 자동으로 연산량이 적은 CPU에 매 스레드를 배치하여 실행한다.
- <8> 이처럼 멀티코어 프로세서를 사용하여 압축 데이터를 디코딩하면, 기존의 압축 디코딩 알고리즘의 경우 단일

CPU에서 동작하는 것을 가정하여 디코딩이 이루어지므로 멀티코어 프로세서 내의 각 CPU 중에서 연산량이 가장 적은 CPU에 엡팩 디코딩 쓰레드를 배치하게 된다.

<9> 즉, 2개의 CPU를 사용한 듀얼 프로세서나 4개의 CPU를 사용한 쿼드 프로세서에서 기존의 엡팩 디코딩 알고리즘을 사용하여 엡팩 데이터를 디코딩하게 되면 1개의 CPU에 집중적으로 연산량이 가중되며 나머지 1개 또는 3개의 CPU는 상대적으로 낮은 연산량이 부과되어 각 CPU의 연산량이 불균일하게 유지되는 문제점이 있다.

<10> 이와 같이 엡팩 디코딩처럼 많은 연산량이 필요한 작업을 처리하는 경우에 멀티코어 프로세서 내의 각 CPU 자원을 고르게 활용하지 못하는 문제점이 있다.

발명이 이루고자 하는 기술적 과제

<11> 본 발명의 목적은 엡팩 데이터를 디코딩할 때 멀티코어 프로세서 내의 여러 코어마다 디코딩 작업을 적절히 분산 배치시킴으로써 프로세서 내 각 코어의 연산 자원을 효율적으로 활용할 수 있으며, 복잡한 연산을 거치지 않고도 엡팩 데이터의 디코딩 작업을 여러 개의 디코딩 쓰레드로 분할할 수 있는 분산 디코딩 처리 장치 및 방법을 제공하는 것이다.

발명의 구성 및 작용

<12> 본 발명의 바람직한 실시예에 의한 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치는, 쓰레드 단위로 동작하는 복수 개의 코어를 구비한 멀티코어 프로세서; 엡팩 데이터를 입력받아 디코딩 정보 추출 후 개별 슬라이스 단위로 분할하고, 개별 슬라이스에 대한 디코딩 연산을 멀티코어 프로세서 내의 각 코어마다 쓰레드로 분산 배치하는 엡팩 데이터 분할 모듈; 엡팩 데이터 분할 모듈로부터 개별 슬라이스를 입력받아 저장하고, 저장된 슬라이스를 멀티코어 프로세서 내의 각 코어마다 제공하는 버퍼 영역을 구비하는 메모리; 및 멀티코어 프로세서의 각 코어를 통해 디코딩된 데이터를 병합하는 디코딩 병합 모듈을 포함하여 구성된다.

<13> 본 발명의 바람직한 실시예에 의한 멀티코어 프로세서를 이용한 분산 디코딩 처리 방법은, 복수 개의 코어를 구비한 프로세서를 통해 엡팩 데이터를 디코딩하는 방법에 있어서, (a) 엡팩 데이터 내의 각 슬라이스를 개별적으로 디코딩하는 단일 슬라이스 디코딩 함수를 준비하는 단계; (b) 엡팩 데이터를 비트스트림으로 입력받고 엡팩 데이터 내의 디코딩 정보를 추출하는 단계; (c) 프로세서의 코어 개수만큼 메모리 내의 영역을 분할하여 코어 개수만큼의 버퍼를 준비하고 각 코어마다 할당하는 단계; (d) 비트스트림을 개별 슬라이스 단위로 분할하는 단계; (e) 분할된 슬라이스 단위의 비트스트림을 버퍼마다 할당하여 입력하는 단계; (f) 버퍼마다 입력된 슬라이스 단위의 비트스트림에 대한 디코딩 연산을 해당 코어에 쓰레드로 할당하는 단계; (g) 각 코어가 단일 슬라이스 디코딩 함수를 사용하여 디코딩 쓰레드를 수행하는 단계; 및 (h) 모든 코어의 디코딩 쓰레드가 완료되면 디코딩된 데이터를 병합하는 단계를 포함하여 구성된다.

<14> 이하, 본 발명의 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

<15> 도 1은 본 발명의 바람직한 실시예에 의한 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치의 전체 구성을 나타낸 블록도이다.

<16> 도시된 바와 같이, 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치는 멀티코어 프로세서(20), 메모리(10), 엡팩 데이터 분할 모듈(30), 디코딩 병합 모듈(40)로 크게 구성된다.

<17> 엡팩 알고리즘으로 인코딩된 비트스트림 내의 한 프레임 데이터(30)를 메모리(10)에 저장하고, 이를 멀티코어 프로세서 내의 각 코어(20)마다 쓰레드로 할당하여 디코딩한 후에 병합하게 된다. 이때, 엡팩 데이터의 매 슬라이스는 서로 독립적으로 인코딩되어 있음에 착안하여 매 슬라이스별로 디코딩한 후에 이를 병합하는 것이 본 발명의 아이디어의 핵심이다.

<18> 멀티코어 프로세서(20)는 쓰레드 단위로 동작하는 복수 개의 코어(CPU)를 구비한 프로세서를 의미하며, 각 코어는 서로 독립적으로 동작할 수 있다.

<19> 메모리(10)는 엡팩 데이터 분할 모듈(30)로부터 개별 슬라이스를 입력받으면 이를 저장하고, 저장된 슬라이스를 멀티코어 프로세서(20)의 각 코어마다 제공하는 버퍼 영역을 구비한다.

<20> 엡팩 데이터 분할 모듈(30)은 엡팩 데이터를 입력받으면 먼저 디코딩 정보를 추출 후 개별 슬라이스 단위로 분할하고, 분할된 개별 슬라이스 단위의 비트스트림에 대한 디코딩 연산을 멀티코어 프로세서 내의 각 코어마다 쓰레드로 분산 배치시킨다. 이때, 분할된 슬라이스 단위의 비트스트림은 우선 메모리(10)의 버퍼에 저장된 후에

각각 대응되는 코어로 제공된다.

- <21>

엠펙 데이터 분할 모듈(30)은 헤더 과성부(31), 슬라이스 분할부(32), 분산 배치부(34)으로 크게 구성되며, 코어 연산량 측정부(33)와 단일 슬라이스 디코딩 함수 생성부가 추가될 수 있다.
- <22>

헤더 과성부(31)는 엠펙 데이터를 비트스트림으로 입력받아 디코딩 정보를 추출하는 등 이에 대한 기본적인 헤더 과성을 수행하고, 메모리(10) 내의 영역을 분할 할당하여 각 슬라이스용 버퍼를 준비한다. 즉, 메모리(10) 내의 영역을 멀티코어 프로세서(20)의 각 코어에 대응되도록 복수 개의 버퍼로 분할 할당한다.
- <23>

슬라이스 분할부(32)는 비트스트림 내의 슬라이스 시작 코드를 감지하여 개별 슬라이스 단위로 분할한다.
- <24>

분산 배치부(34)는 슬라이스 단위로 분할된 비트스트림을 각 버퍼마다 적절히 분산 배치한다. 이때, 각 버퍼마다 균일하게 분산시킬 수도 있으며, 각 코어의 연산 점유량에 반비례하도록 각 버퍼에 분산시킬 수도 있다. 후자의 경우에 각 코어의 연산 점유량을 측정하기 위해 멀티코어 프로세서 내에 있는 각 코어의 연산 점유량을 측정하는 코어 연산량 측정부(33)를 추가로 더 포함하게 된다.
- <25>

또한, 전자와 같이 분산 배치부(34)에서 구현상 용이를 위해 코어의 연산 점유량을 고려하지 않을 수도 있다. 이때는, 코어의 연산 점유량에 상관없이 매 슬라이스가 각 버퍼마다 고루 분포된다.
- <26>

슬라이스 단위로 분할된 비트스트림은 분산 배치부(34)에 의해 메모리(10) 내의 각 버퍼에 분산 배치되며, 각 버퍼는 그에 대응되는 코어마다 슬라이스 디코딩 쓰레드를 할당한다. 그리고, 멀티코어 프로세서(20) 내의 각 코어에서는 대응되는 버퍼로부터 데이터를 입력받아 각자의 슬라이스 디코딩 쓰레드를 수행한다.
- <27>

이때, 각 코어에서 매 슬라이스에 대해 디코딩을 수행하기 위한 디코딩 함수는 기존의 엠펙 데이터를 디코딩하기 위한 디코딩 함수와는 다소 차이가 있다. 즉, 기존의 디코딩 함수는 슬라이스 단위로 분할되지 않은 엠펙 데이터를 디코딩 대상으로 삼기 때문에, 슬라이스 단위의 디코딩 함수를 새로 정의할 필요가 있다.
- <28>

따라서, 엠펙 데이터 분할 모듈(30)은 엠펙 데이터의 디코딩 정보를 이용하여 엠펙 데이터 내의 각 슬라이스를 개별적으로 디코딩하는 단일 슬라이스 디코딩 함수를 생성하여 멀티코어 프로세서의 각 코어에 제공하는 단일 슬라이스 디코딩 함수 생성부를 더 포함하도록 구성될 수 있다.
- <29>

디코딩 병합부(80)는 멀티코어 프로세서 내의 각 코어의 디코딩 쓰레드를 통해 디코딩된 결과물을 병합하며, 이를 통해 전체 비트스트림을 디코딩한 효과를 가져온다.
- <30>

도 2는 본 발명의 바람직한 실시예에 의한 멀티코어 프로세서를 이용한 분산 디코딩 처리 방법의 전체 동작과정을 나타낸 순서도이다.
- <31>

도시된 바와 같이, 먼저 엠펙 데이터 내의 각 슬라이스를 개별적으로 디코딩하는 단일 슬라이스 디코딩 함수를 준비한다(S10).
- <32>

그 다음으로, 엠펙 데이터를 비트스트림으로 입력받고 엠펙 데이터 내의 디코딩 정보를 추출한다(S20). 엠펙 데이터 내의 픽처헤더를 판독함으로써 디코딩을 위한 화면크기나 기본 정보를 얻을 수 있다.
- <33>

그리고, 프로세서의 코어 개수만큼 메모리 내의 영역을 분할하여 코어 개수만큼의 버퍼를 준비하고, 각 버퍼마다 대응되는 코어를 할당함으로써 앞으로 버퍼를 통해 들어오는 데이터가 각 코어마다 입력되도록 한다(S30).
- <34>

그 다음으로, 엠펙 데이터의 비트스트림을 개별 슬라이스 단위로 분할한다(S40). 이때, 비트스트림 내의 슬라이스 시작 코드를 감지함으로써 각 슬라이스를 구분하여 분할할 수 있다.
- <35>

이와 같이 분할된 슬라이스 단위의 비트스트림을 앞서 각 코어에 대응된 버퍼마다 할당하여 입력한다(S50). 이때, 매 슬라이스 단위의 비트스트림을 어떻게 분산배치하느냐에 따라서 각 코어에 가해지는 연산 부담에 차이가 생기며, 각 코어마다 아웃풋이 출력되는 시간차가 크게 발생할 수 있다.
- <36>

여기에서, 엠펙 데이터 내의 매크로 블럭은 인터(inter) 코딩되어 있거나 인트라(intra) 코딩되어 있는지의 여부에 상관없이 디코딩 연산량이 비슷하며, 따라서 개별 슬라이스들도 각각 연산량이 비슷하다. 또한, 매 슬라이스는 서로 독립적으로 인코딩되어 있다.
- <37>

따라서, 일반적으로 각 코어의 연산 점유량이 비슷하다고 가정하면, 슬라이스 단위의 비트스트림을 각 코어의 해당 버퍼마다 동일한 분량으로 할당하여 순차적으로 입력하는 것이 바람직하다.
- <38>

또한, 각 코어의 연산 점유량을 체크한 후에 각 코어의 연산 점유량에 반비례하도록 매 슬라이스를 분산 배치할

수도 있으며, 이 경우에도 역시 마찬가지로 각 코어의 부담을 균일하게 유지시킬 수 있고 각 코어의 디코딩 쓰레드 완료 시점을 비슷하게 유지시킬 수 있다.

- <39> 그 다음으로, 각 버퍼마다 입력된 슬라이스 단위의 비트스트림에 대한 디코딩 연산을 해당 코어에 쓰레드로 할당한다(S60).
- <40> 그러면, 각 코어가 앞서 준비된 단일 슬라이스 디코딩 함수를 사용하여 해당 버퍼로부터 제공받은 슬라이스에 대하여 디코딩 쓰레드를 수행한다(S70).
- <41> 마지막으로, 모든 코어의 디코딩 쓰레드가 완료될 때까지 기다린 후에 각 코어마다 디코딩된 데이터를 병합한다(S80).
- <42> 도 3은 도 2에서 슬라이스 단위의 비트스트림을 각 코어에 할당된 버퍼마다 분산 배치하는 단계(S50)의 동작과정을 보다 자세하게 나타낸 순서도이다.
- <43> 도시된 바와 같이, 멀티코어 프로세서 내 각 코어의 연산 점유량을 체크한다(S51).
- <44> 그 다음으로, 슬라이스 단위의 비트스트림을 각 코어의 연산 점유량에 반비례하도록 할당하여 각 코어의 해당 버퍼에 입력한다(S52).
- <45> 특히 각 코어의 연산 점유량이 서로 크게 다른 경우에는 이처럼 매 슬라이스를 각 코어의 연산 점유량에 반비례하도록 분산 배치하는 것이 프로세서 자원 활용에 유리하다. 즉, 상대적으로 사용량이 많은 코어에는 소량의 쓰레드가 배치되고 사용량이 적은 코어에는 많은 양의 쓰레드가 배치된다.
- <46> 또한, 매 슬라이스를 각 코어의 해당 버퍼마다 동일한 분량으로 분산 배치할 수도 있다. 즉, 일반적으로 각 코어의 연산 점유량이 균일하게 유지되고 있다고 가정하면, 매 슬라이스마다 연산량에 거의 차이가 없으므로 매 슬라이스를 동일한 분량으로 분산시키더라도 각 코어에 할당되는 쓰레드가 균일하게 유지된다. 이때는 각 코어의 연산 점유량을 일일이 체크할 필요가 없으므로 보다 간단한 구성으로 구현이 가능해진다.
- <47> 도 4는 엠팩 데이터의 비트스트림 내의 슬라이스 구조를 나타낸 도면이다.
- <48> 엠팩 데이터는 크게 시퀀스, GOP(group of picture), 픽처, 슬라이스, 매크로블럭, 블럭의 여섯 개의 계층으로 이루어져 있다. 매 단위의 구분은 슬라이스까지 외부에서 위치를 알 수 있도록 시작코드(start code)로 구분되어 있다.
- <49> 도시된 바와 같이, 슬라이스 데이터의 시작코드는 00 00 01 xx의 값으로 나타나므로, 여기서 01, 02, 03 등 N개의 슬라이스를 구분할 수 있으므로 각 슬라이스를 모을 수 있다.
- <50> 또한, 매 슬라이스의 시작과 끝은 인트라 매크로블럭으로 코딩이 되어 있어서 매 슬라이스는 연속성이 없으며 다른 슬라이스 데이터의 영향을 받지 않아 매 슬라이스는 서로 독립적이다. 따라서, 매 슬라이스는 서로 독립적으로 인코딩되어 있기 때문에 슬라이스 단위로 비트스트림을 나누어서 개별적으로 디코딩하더라도 데이터에 손상이 가지 않는다.
- <51> 이상에서 실시예를 들어 본 발명을 더욱 상세하게 설명하였으나, 본 발명은 반드시 이러한 실시예로 국한되는 것은 아니고, 본 발명의 기술사상을 벗어나지 않는 범위 내에서 다양하게 변형실시될 수 있다. 따라서, 본 발명에 개시된 실시예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

발명의 효과

- <52> 본 발명에 따르면, 복잡한 연산을 거치지 않고도 엠팩 데이터의 디코딩 작업을 여러 개의 디코딩 쓰레드로 분할할 수 있으며, 엠팩 디코딩처럼 연산량이 많은 작업을 처리할 때 멀티코어 프로세서 내의 일부 CPU에 연산량이 과중되는 현상을 방지할 수 있으며, 멀티코어 프로세서 내의 여러 CPU 자원을 효율적으로 활용할 수 있는 효과가 있다.

도면의 간단한 설명

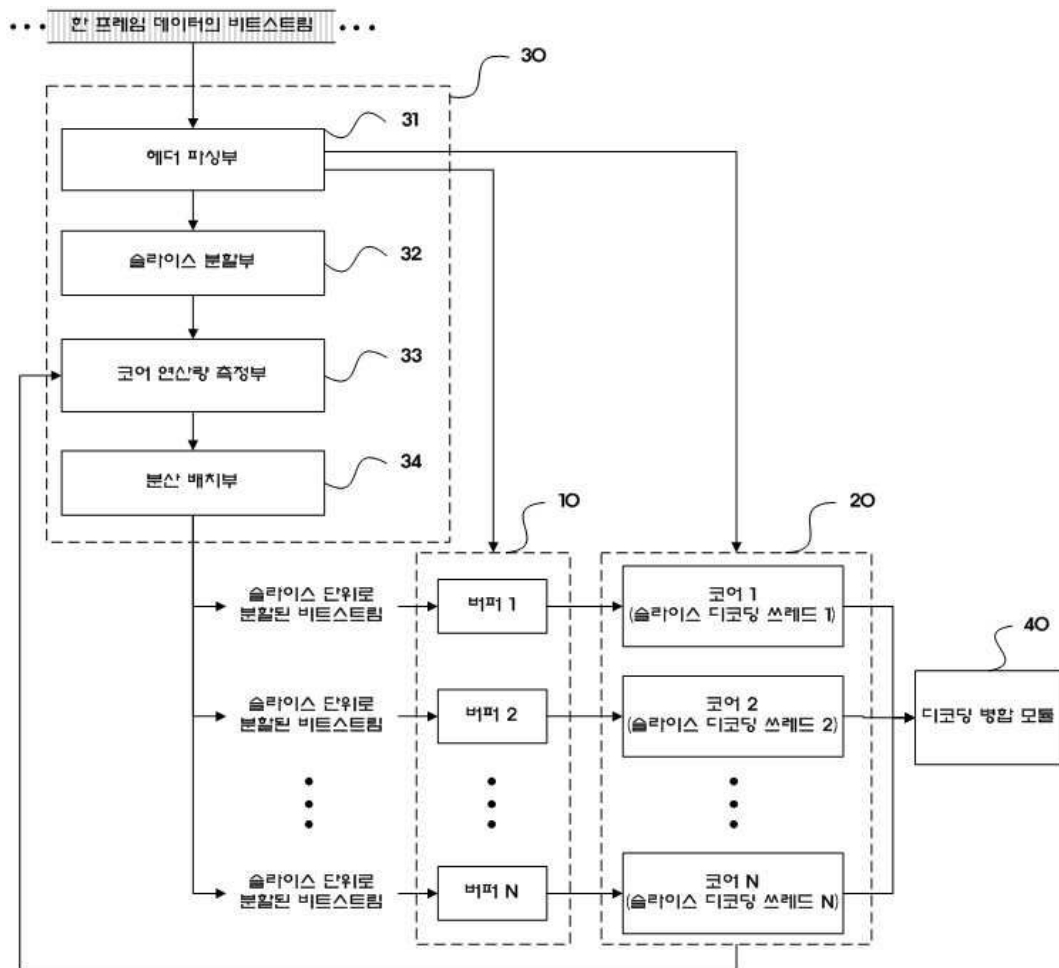
- <1> 도 1은 본 발명의 바람직한 실시예에 의한 멀티코어 프로세서를 이용한 분산 디코딩 처리 장치의 전체 구성을

나타낸 블록도,

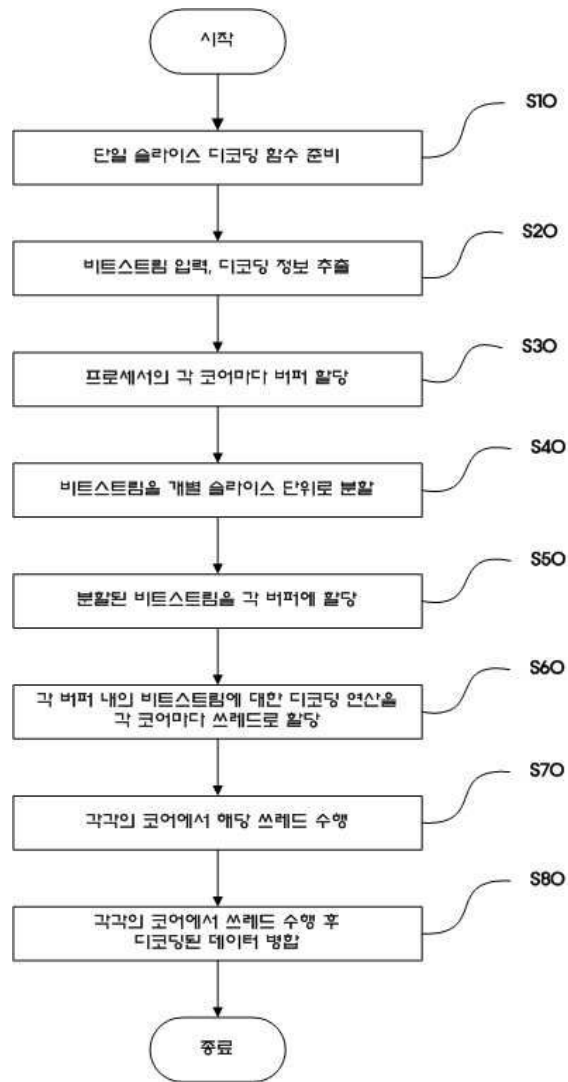
- <2> 도 2는 본 발명의 바람직한 실시예에 의한 멀티코어 프로세서를 이용한 분산 디코딩 처리 방법의 전체 동작과정을 나타낸 순서도,
- <3> 도 3은 도 2에서 슬라이스 단위의 비트스트림을 각 코어에 할당된 버퍼마다 분산 배치하는 단계(S50)의 동작과정을 보다 자세히 나타낸 순서도,
- <4> 도 4는 앰팩 데이터의 비트스트림 내의 슬라이스 구조를 나타낸 도면이다.

도면

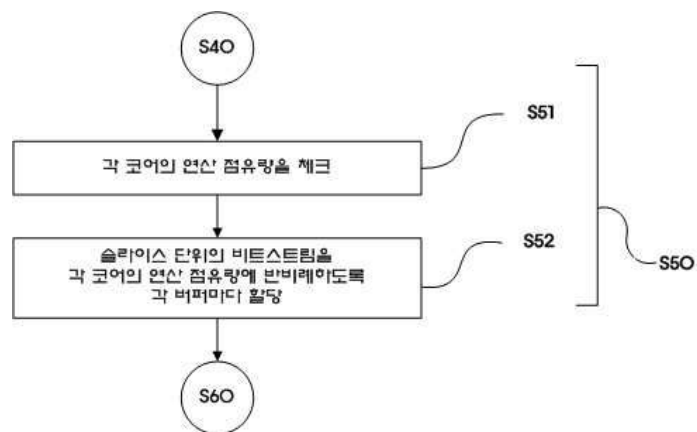
도면1



도면2



도면3



도면4

