



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2016년04월08일  
 (11) 등록번호 10-1610828  
 (24) 등록일자 2016년04월04일

(51) 국제특허분류(Int. Cl.)  
 G06F 9/46 (2006.01) G06F 13/24 (2006.01)  
 (21) 출원번호 10-2009-0090288  
 (22) 출원일자 2009년09월23일  
 심사청구일자 2014년09월23일  
 (65) 공개번호 10-2011-0032675  
 (43) 공개일자 2011년03월30일  
 (56) 선행기술조사문헌  
 US5321825 A\*  
 US20070204268 A1\*  
 KR1020070073916 A  
 KR1019990007812 A  
 \*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
 삼성전자주식회사  
 경기도 수원시 영통구 삼성로 129 (매탄동)  
 (72) 발명자  
 이주평  
 경기도 수원시 영통구 권광로276번길 20, 133동  
 303호 (매탄동, 현대힐스테이트)  
 (74) 대리인  
 특허법인 신지, 유경열

전체 청구항 수 : 총 18 항

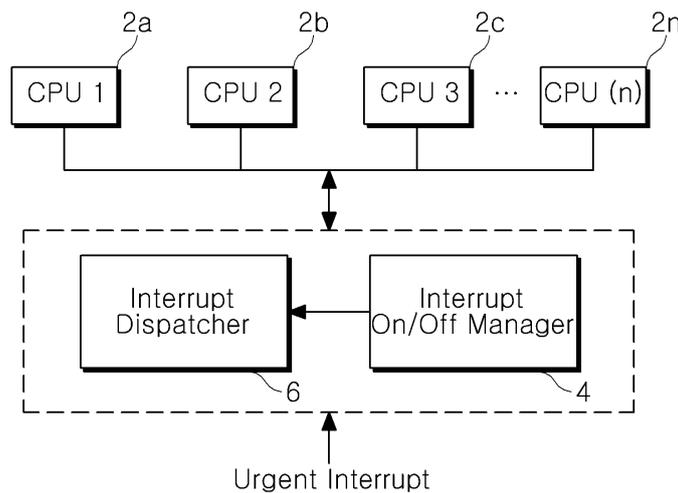
심사관 : 유진태

(54) 발명의 명칭 **멀티코어 프로세서의 인터럽트 온/오프 관리 장치와 방법**

**(57) 요약**

복수의 CPU 코어를 포함하는 멀티코어 프로세서의 인터럽트 온/오프 관리 장치와 방법을 개시한다. 인터럽트 온/오프 관리 장치는 타깃 CPU 세트에 포함되는 둘 이상의 CPU 코어 가운데 적어도 하나의 CPU 코어는 긴급 인터럽트의 실행이 가능한 상태가 되도록 멀티코어 프로세서들을 관리한다. 이를 위하여, 인터럽트 온/오프 관리 장치는 비임계 구간에서 있는 CPU 코어가 임계 구간으로 진입하는 것을 제어함으로써 적어도 하나의 CPU 코어가 비임계 구간에 위치하도록 CPU 코어들을 제어할 수 있다. 임계 구간은 인터럽트 불능 구간만을 포함하거나 또는 커널 비선점 구간도 포함할 수 있으며, 비임계 구간은 인터럽트 가능 구간만을 포함하거나 또는 인터럽트 가능 구간이면서 커널 선점가능 구간인 경우만을 포함할 수 있다.

**대표도** - 도1a



## 명세서

### 청구범위

#### 청구항 1

$n$  ( $n \geq 2$ )개의 CPU(Central Processing Unit) 코어를 포함하는 멀티코어 프로세서의 인터럽트 온/오프 관리 장치로서,

상기  $n$ 개의 CPU 코어 중에서  $m$  ( $2 \leq m \leq n$ )개의 CPU 코어로 구성된 타깃 CPU 세트에 포함되는 CPU 코어가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단하는 판단부; 및

상기 판단부의 판단 결과에 기초하여, 상기  $m$ 개의 CPU 코어 가운데 적어도 하나의 CPU 코어는 상기 비임계 구간에 위치하도록 상기  $m$ 개의 CPU 코어를 제어하는 제어부를 포함하고,

상기 임계 구간은 인터럽트 불능 구간과 커널 비선점 구간 중에서 어느 하나의 구간인 경우를 포함하고, 상기 비임계 구간은 인터럽트 가능 구간이면서 동시에 커널 선점가능 구간인 경우를 포함하는 인터럽트 온/오프 관리 장치.

#### 청구항 2

제1항에 있어서,

상기 제어부는 상기  $m$ 개의 CPU 코어 중에서 비임계 구간에 위치하는 제1 CPU 코어가 임계 구간으로 진입하는 것을 제어하는 인터럽트 온/오프 관리 장치.

#### 청구항 3

제2항에 있어서,

상기 판단부가 상기  $m$ 개의 CPU 코어 중에서 상기 제1 CPU 코어를 제외한 ( $m-1$ )개의 CPU 코어가 모두 임계 구간에 위치하는 것으로 판단할 경우에,

상기 제어부는 상기 제1 CPU 코어가 임계 구간으로 진입하는 것을 허용하지 않는 인터럽트 온/오프 관리 장치.

#### 청구항 4

제2항에 있어서,

상기 제어부는 상기 제1 CPU 코어가 실행 시간의 소정의 시간 이상이 소요되는 임계 구간으로 진입하는 것만을 제어하는 인터럽트 온/오프 관리 장치.

#### 청구항 5

제1항에 있어서,

상기 임계 구간은 인터럽트 불능 구간인 경우를 포함하고, 상기 비임계 구간은 인터럽트 가능 구간인 경우를 포함하는 인터럽트 온/오프 관리 장치.

#### 청구항 6

삭제

#### 청구항 7

제1항에 있어서,

상기 판단부는 상기 타깃 CPU 세트에 포함되는 각각의 CPU 코어가 위치하는 구간을 가리키는 인터럽트 온/오프 정보를 이용하여 판단을 수행하는 인터럽트 온/오프 관리 장치.

**청구항 8**

제1항에 있어서,

상기 n개의 CPU 코어들 중에서 상기 타깃 CPU 세트에 포함되는 CPU 코어의 종류 및/또는 개수를 지정하는 타깃 지정부를 더 포함하는 인터럽트 온/오프 관리 장치.

**청구항 9**

제1항에 있어서,

긴급 인터럽트가 발생하면, 상기 판단부의 판단 결과를 이용하여 비임계 구간에 위치하는 상기 적어도 하나의 CPU 코어 중에서 어느 하나의 CPU 코어로 상기 긴급 인터럽트가 전달되도록 하거나 또는 상기 타깃 CPU 세트에 포함되는 모든 CPU 코어로 상기 긴급 인터럽트가 전달되도록 인터럽트 브로드캐스트를 하는 인터럽트 전달부를 더 포함하는 인터럽트 온/오프 관리 장치.

**청구항 10**

제9항에 있어서,

상기 긴급 인터럽트는 로컬 타이머 인터럽트를 포함하고,

상기 인터럽트 전달부는 상기 타깃 CPU 세트에 포함되는 CPU 코어들 사이에서 상기 로컬 타이머 인터럽트를 실행하는데 필요한 정보를 서로 교환해 주는 인터럽트 온/오프 관리 장치.

**청구항 11**

제9항에 있어서,

상기 인터럽트 전달부는 상기 긴급 인터럽트를 직접 전달하거나 또는 외부에서 상기 긴급 인터럽트를 전달하는 인터럽트 디스패처를 제어하는 인터럽트 온/오프 관리 장치.

**청구항 12**

제1항에 있어서,

상기 m과 n은 각각 3이상인 정수이고,

상기 제어부는 상기 타깃 CPU 세트에 포함되는 m개의 CPU 코어 가운데 적어도 두 개의 CPU 코어가 비임계 구간에 위치하도록 상기 m개의 CPU 코어를 제어하는 인터럽트 온/오프 관리 장치.

**청구항 13**

n( $\geq 2$ )개의 CPU(Central Processing Unit) 코어를 포함하는 멀티코어 프로세서의 인터럽트 온/오프 관리 방법으로서,

상기 n개의 CPU 코어 중에서 m( $2 \leq m \leq n$ )개의 CPU 코어로 구성된 타깃 CPU 세트에 포함되는 CPU 코어가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단하는 제1 판단 단계; 및

상기 제1 판단 단계에서의 판단 결과에 기초하여, 상기 m개의 CPU 코어 가운데 적어도 하나의 CPU 코어는 상기 비임계 구간에 위치하도록 상기 m개의 CPU 코어를 제어하는 제어 단계를 포함하고,

상기 임계 구간은 인터럽트 불능 구간과 커널 비선점 구간 중에서 어느 하나의 구간인 경우를 포함하고, 상기 비임계 구간은 인터럽트 가능 구간이면서 동시에 커널 선점가능 구간인 경우를 포함하며,

상기 제어 단계에서는 상기 m개의 CPU 코어 중에서 비임계 구간에 위치하는 제1 CPU 코어가 임계 구간으로 진입하는 것을 제어하는 인터럽트 온/오프 관리 방법.

**청구항 14**

제13항에 있어서,

상기 임계 구간은 인터럽트 불능 구간인 경우를 포함하고, 상기 비임계 구간은 인터럽트 가능 구간인 경우를 포

함하며,

상기 제어 단계에서는 상기 m개의 CPU 코어 중에서 인터럽트 가능 구간에 위치하는 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 제어하는 인터럽트 온/오프 관리 방법.

**청구항 15**

제14항에 있어서,

상기 제1 판단 단계 이전에 상기 타깃 CPU 세트에 포함되는 CPU 코어의 상태 정보를 고정하는 로킹(locking) 단계를 더 포함하고,

상기 제1 판단 단계에서 (m-1)개의 CPU 코어 전부가 인터럽트 불능 구간에 위치하는 것으로 판단되는 경우에는, 상기 제어 단계에서는 상기 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용하지 않으면서 상기 판단 단계를 반복하여 수행하는 인터럽트 온/오프 관리 방법.

**청구항 16**

제14항에 있어서,

상기 제1 판단 단계 이전에 상기 타깃 CPU 세트에 포함되는 CPU 코어의 상태 정보를 고정하는 록킹 단계를 더 포함하고,

상기 제1 판단 단계에서 (m-1)개의 CPU 코어 중에서 적어도 하나가 인터럽트 가능 구간에 위치하는 것으로 판단되는 경우에는, 상기 상태 정보를 언록킹(unlocking)하여 상기 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용하는 언록킹 단계를 더 포함하는 인터럽트 온/오프 관리 방법.

**청구항 17**

삭제

**청구항 18**

제13항에 있어서, 상기 제1 판단 단계 이전에

상기 타깃 CPU 세트에 포함되는 CPU 코어의 상태 정보를 고정하는 로킹(locking) 단계; 및

상기 제1 CPU가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단하는 제2 판단 단계를 더 포함하고,

상기 제2 판단 단계에서 상기 제1 CPU가 임계 구간에 위치하는 것으로 판단되면, 상기 상태 정보를 언록킹(unlocking)하여 상기 제1 CPU 코어가 인터럽트 가능 구간 또는 커널 비선점 구간으로 진입하는 것을 허용하는 언록킹 단계를 더 포함하는 인터럽트 온/오프 관리 방법.

**청구항 19**

제13항에 있어서, 상기 제1 판단 단계 이전에

상기 타깃 CPU 세트에 포함되는 CPU 코어의 상태 정보를 고정하는 로킹(locking) 단계; 및

상기 제1 CPU가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단하는 제2 판단 단계를 더 포함하고,

상기 제2 판단 단계에서 상기 제1 CPU가 임계 구간에 위치하는 것으로 판단되면 상기 제1 판단 단계를 수행하고,

상기 제1 판단 단계에서 (m-1)개의 CPU 코어 전부가 임계 구간에 위치하는 것으로 판단되는 경우에는, 상기 제어 단계에서는 상기 제1 CPU 코어가 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하는 것을 허용하지 않으면서 상기 제1 판단 단계를 반복하여 수행하는 인터럽트 온/오프 관리 방법.

**청구항 20**

제13항에 있어서, 상기 제1 판단 단계 이전에

상기 타깃 CPU 세트에 포함되는 CPU 코어의 상태 정보를 고정하는 로킹(locking) 단계; 및

상기 제1 CPU가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단하는 제2 판단 단계를 더 포함하고,

상기 제2 판단 단계에서 상기 제1 CPU가 임계 구간에 위치하는 것으로 판단되면 상기 제1 판단 단계를 수행하고,

상기 제1 판단 단계에서 (m-1)개의 CPU 코어 중에서 적어도 하나가 비임계 구간에 위치하는 것으로 판단되는 경우에는, 상기 상태 정보를 언로킹(unlocking)하여 상기 제1 CPU 코어가 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하는 것을 허용하는 언로킹 단계를 더 포함하는 인터럽트 온/오프 관리 방법.

**발명의 설명**

**발명의 상세한 설명**

**기술 분야**

[0001] 인터럽트 처리(interrupt handling)에 관한 것으로, 보다 구체적으로 멀티코어 프로세서(multi-core processor)에서 인터럽트를 처리하는 것과 관련된다.

**배경 기술**

[0002] 운영체제(Operation System, OS)에서는 동기화(synchronization)를 목적으로 배타적으로 실행되어야 하는 임계 구간(critical section)이 여러 곳에 존재한다. '배타적'이라는 것은 다른 상황에 상관없이 CPU(Central Processing Unit)에서 해당 프로세스가 반드시 실행되어야 하는 것을 의미한다. 따라서 아무리 중요한 또는 긴급한 인터럽트 신호가 발생하더라도, 만일 임계 구간에서 해당 인터럽트 신호가 발생한 경우에는 처리될 수가 없거나 처리 지연이 불가피하다.

[0003] 현재 컴퓨터의 운영체제에서는 이들 임계 구간이 인터럽트 불능구간(interrupt-disabled section)으로 구현되거나, 커널 비선점 구간(kernel non-preemptible section)으로 구현되거나, 또는 이들의 조합으로 구현되고 있다. 인터럽트 불능구간은 인터럽트 핸들러(interrupt handler)가 실행되지 않아서 인터럽트 처리가 이루어지지 않는 구간을 가리키며, 커널 비선점 구간은 비록 인터럽트 핸들러가 실행된다고 하더라도 프로세스 스위치(process switch) 또는 컨텍스트 스위치(context switch)가 발생하지 못하도록 하여 커널 선점이 되지 않도록 하는 구간을 가리킨다. 이러한 인터럽트 불능구간과 커널 비선점 구간 외에도 인터럽트 신호를 처리하는 시간도 넓게는 임계 구간으로 볼 수 있다. 왜냐하면, 일반적으로 CPU가 특정 인터럽트의 처리를 시작하면 해당 인터럽트의 처리가 종료될 때까지는 다른 인터럽트를 처리할 수가 없기 때문이다.

[0004] 이와 같이, 인터럽트 불능 구간 및/또는 커널 비선점 구간 등과 같은 임계 구간이 운영체제에 존재하는 것은, 인터럽트 처리로 인하여 운영체제에서 동기화 문제가 발생하는 것을 방지하기 위한 것이다. 따라서 컴퓨터의 운영체제에서는 임계 구간이 반드시 존재해야 한다. 하지만, 이러한 임계 구간의 존재는 신속히 처리되어야 하는 실시간 프로세스 등을 깨워주는 긴급 인터럽트(urgent interrupt)의 실행을 지연시키는 원인이 되고 있으며, 멀티코어 시스템에서도 이러한 긴급 인터럽트의 처리 지연은 불가피하다. 긴급 인터럽트의 실행 지연 문제를 해결하기 위하여 여러 가지 방법이 제안되었는데, 예를 들어 락 브레이킹(lock breaking) 기법, 듀얼 커널(dual kernel) 기법, 프로세서 차폐(shielded CPU) 기법, 임계 구간 진입 지연(delayed locking) 기법 등이 있다.

**발명의 내용**

**해결 하고자하는 과제**

[0005] 긴급 인터럽트(urgent interrupt)의 처리를 보장하거나 또는 처리 지연을 최소화함으로써, 멀티코어 시스템 상에서 구동되는 범용 운영체제의 실시간성을 향상시킬 수 있으며 또한 고성능 실시간 운영체제에 사용될 수 있는 멀티코어 프로세서의 인터럽트 온/오프 관리 장치와 방법, 및 이를 포함하는 멀티코어 컴퓨터를 제공한다.

[0006] 신속히 처리되어야 하는 하나 또는 복수의 긴급 인터럽트가 인터럽트 불능 구간이나 커널 비선점 구간의 영향을 받지 않고 최소 지연시간 내에 처리될 수 있도록 할 수 있는 멀티코어 프로세서의 인터럽트 온/오프 관리 장치와 방법, 및 이를 포함하는 멀티코어 컴퓨터를 제공한다.

**과제 해결수단**

[0007] 일 실시예에 따른 인터럽트 온/오프 관리 장치는  $n(\geq 2)$ 개의 CPU(Central Processing Unit) 코어를 포함하는 멀티코어 프로세서의 인터럽트 온/오프 관리 장치로서, 판단부 및 제어부를 포함한다. 판단부는  $n$ 개의 CPU 코어 중에서  $m(2 \leq m \leq n)$ 개의 CPU 코어로 구성된 타깃 CPU 세트에 포함되는 CPU 코어가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단한다. 그리고 제어부는 판단부의 판단 결과에 기초하여,  $m$ 개의 CPU 코어 가운데 적어도 하나의 CPU 코어는 비임계 구간에 위치하도록  $m$ 개의 CPU 코어를 제어한다. 예를 들어, 제어부는  $m$ 개의 CPU 코어 중에서 비임계 구간에 위치하는 CPU 코어가 임계 구간으로 진입하는 것을 제어할 수 있다.

[0008] 일 실시예에 따른 인터럽트 온/오프 관리 방법은  $n(\geq 2)$ 개의 CPU(Central Processing Unit) 코어를 포함하는 멀티코어 프로세서의 인터럽트 온/오프 관리 방법이다. 먼저,  $n$ 개의 CPU 코어 중에서  $m(2 \leq m \leq n)$ 개의 CPU 코어로 구성된 타깃 CPU 세트에 포함되는 CPU 코어가 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 판단한다. 그리고 판단 결과에 기초하여,  $m$ 개의 CPU 코어 가운데 적어도 하나의 CPU 코어는 비임계 구간에 위치하도록  $m$ 개의 CPU 코어를 제어한다.

[0009] 이러한 실시예들에서, 임계 구간은 인터럽트 불능 구간인 경우만을 포함하거나 또는 커널 비선점 구간인 경우도 포함할 수 있다. 그리고 비임계 구간은 인터럽트 가능 구간인 경우만을 포함하거나 또는 인터럽트 가능 구간이면서 동시에 커널 선점가능 구간인 경우만을 포함할 수 있다.

**효과**

[0010] 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 적어도 하나의 CPU 코어는 긴급 인터럽트의 처리가 가능하도록 함으로써, 실시간 프로세스 등과 같은 긴급 인터럽트의 처리를 보장하거나 또는 처리 지연을 최소화할 수 있다.

**발명의 실시를 위한 구체적인 내용**

[0011] 이하, 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세하게 설명한다. 본 발명의 양상을 설명함에 있어 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 또한, 후술되는 용어들은 본 발명의 실시예에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0012] 도 1a는 일 실시예에 따른 멀티코어 컴퓨터(multi-core computing system)의 구성의 일례를 보여 주는 블록도이다. 그리고 도 1b는 도 1a의 인터럽트 온/오프 관리 장치(4)의 세부 구성의 일례를 보여 주는 블록도이다. 도 1a에는 전체 멀티코어 컴퓨터 중에서 본 실시예를 설명하기 위하여 필요한 구성요소들만을 도시하였으며, 다른 구성요소들에 대한 도시는 생략하였다. 도 1a를 참조하면, 멀티코어 컴퓨터는  $n(\geq 2)$  개의 CPU 코어(2a, 2b, 2c, ..., 2n)와 이들 CPU 코어(2a, 2b, 2c, ..., 2n)의 인터럽트 온/오프 관리 장치(Interrupt On/Off Manager, 4)를 포함한다. 그리고 도 1b를 참조하면, 인터럽트 온/오프 관리 장치(4)는 판단부(4a), 제어부(4b), 타깃 지정부(4c), 및 인터럽트 전달부(4d)를 포함하는데, 타깃 지정부(4c)와 인터럽트 전달부(4d)는 임의적인 구성 요소이다.

[0013] 멀티코어 컴퓨터는 2개 이상의 CPU 코어(2a, 2b, 2c, ..., 2n)를 포함한다. 모든 CPU 코어는 단일 운영체제 상에서 구동될 수 있다. 즉, 멀티코어 컴퓨터는 반드시 복수 개의 운영체제 커널이 함께 구동되는 시스템일 필요는 없으며, 단일 운영체제 커널이 구동되는 일반적인 환경의 시스템이어도 무방하다. 그리고 실시간 프로세스나 중요한 인터럽트(이하, 멀티코어 시스템에서 현재 실행 중인 다른 프로세스에 비하여 우선하여 실행되어야 하는 이러한 실시간 프로세스를 깨워주는 인터럽트를 '긴급 인터럽트(urgent interrupt)'라고 한다)는 반드시 특정

CPU 코어에서 실행되어야 할 필요는 없으며, 모든 CPU 코어(보다 정확하게는, 후술하는 바와 같은 '타겟 CPU 세트'에 포함되는 모든 CPU 코어)에서 실행될 수 있다.

[0014] n개의 CPU 코어(2a, 2b, 2c, ..., 2n) 중에서,  $m(2 \leq m \leq n)$ 개의 CPU 코어는 타겟 CPU 세트(Target CPU Set)에 포함된다. '타겟 CPU 세트'란 인터럽트 온/오프 관리 장치(4), 보다 구체적으로는 인터럽트 온/오프 관리 장치(4)의 제어부(4b)에 의하여 비임계 구간(non-critical section)에 위치하는 CPU 코어가 임계 구간(critical section)으로 진입하는 것이 제어되는 CPU 코어의 집합을 가리킨다. 타겟 CPU 세트는 n개의 CPU 코어(2a, 2b, 2c, ..., 2n) 중의 전부 또는 일부로 구성될 수 있는데, 적어도 두 개의 CPU 코어를 포함한다. 도 2에는  $4(n=4)$ 개의 CPU 코어를 포함하는 멀티코어 컴퓨터에서 타겟 CPU 세트를 구성하는 방법이 예시되어 있는데, 도 2의 (a)는 4개의 CPU 코어 모두가 타겟 CPU 세트에 포함되는 경우( $m=4$ )이고, 도 2의 (b)는 4개의 CPU 코어 중에서 2개의 CPU 코어가 타겟 CPU 세트에 포함되는 경우( $m=2$ )이다.

[0015] 이러한 타겟 CPU 세트에 포함되는 CPU 코어의 개수 및/또는 종류는 멀티코어 컴퓨터에서 미리 결정되어 있거나 또는 인터럽트 온/오프 관리 장치(4)에 의하여 결정될 수 있다. 후자의 경우에, 인터럽트 온/오프 관리 장치(4), 보다 구체적으로 제어부(4b) 또는 타겟 지정부(4c)는 임의적으로 또는 외부로부터의 입력에 기초하여 타겟 CPU 세트에 포함되는 CPU 코어의 개수 및/또는 종류를 결정할 수 있다. 그리고 타겟 CPU 세트에 포함되는 CPU 코어의 개수 및/또는 종류는 가변적일 수 있다. 또는, 인터럽트 온/오프 관리 장치(4)의 제어부(4b) 또는 타겟 지정부(4c)는, 각 CPU 코어로부터의 요청(예컨대, 타겟 CPU 세트로의 결합 요청(Request for joining to Target CPU Set) 또는 타겟 CPU 세트로부터의 이탈 요청(Request for leaving from Target CPU Set) 등)에 응답하여, 타겟 CPU 세트에 포함되는 CPU 코어의 개수 및/또는 종류를 조정할 수도 있다.

[0016] 타겟 CPU 세트에 포함되는 m개의 CPU 코어들 각각은 임계 구간(인터럽트 불능 구간 및/또는 커널 비선점 구간)으로의 진입과 종료로 반복한다. 이 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 m개의 CPU 코어들 중에서 적어도 하나의 CPU 코어는 '비임계 구간'에 위치하며, 나머지 CPU 코어는 '임계 구간'에 위치하도록 m개의 CPU 코어들을 제어한다. 임계 구간과 비임계 구간의 구별은, 예를 들어, 해당 CPU 코어가 인터럽트 불능 구간(interrupt-disabled section, DI)에 위치하는지 또는 인터럽트 가능 구간(interrupt-enabled section, EN)에 위치하는지를 기준으로 할 수 있다. 이 경우에, 해당 CPU 코어가 인터럽트 불능 구간(DI)에 위치하면 임계 구간에 위치하는 것이 되며, 또한 인터럽트 가능 구간(EN)에 위치하면 비임계 구간에 위치하는 것이 된다.

[0017] 또는, 인터럽트 불능 구간(DI)에 위치하는지 또는 인터럽트 가능 구간(EN)에 위치하는지 외에, 해당 CPU 코어가 커널 비선점 구간(kernel non-preemptible section, NP)에 위치하는지 또는 커널 선점가능 구간(kernel preemptible section, P)에 위치하는지도 추가로 함께 임계 구간과 비임계 구간의 구별에 고려될 수 있다. 이 경우에, 해당 CPU 코어가 인터럽트 불능 구간(EN)에 위치하거나 또는 커널 비선점 구간(NP)에 위치하면 임계 구간에 위치하는 것이 되며, 또한 인터럽트 가능 구간(EN)이면서 동시에 커널 선점가능 구간(P)에 위치하면 비임계 구간에 위치하는 것이 된다.

[0018] 도 3은 4개의 CPU 코어(CPU1, CPU2, CPU3, CPU4)로 구성된 타겟 CPU 세트에서 임계 구간에 위치하는 CPU 코어와 비임계 구간에 위치하는 CPU 코어를 도식적으로 표현한 블록도이다. 도 3에 도시된 도면은 임계 구간에 위치하는 CPU 코어와 비임계 구간에 위치하는 CPU 코어의 구별을 위하여, 해당 CPU 코어가 인터럽트 가능 구간(EN)/인터럽트 불능 구간(DI)에 위치하는지에 추가하여 커널 선점가능 구간(P)/커널 비선점 구간(NP)에 위치하는지도 함께 고려한 것이다. 도 3을 참조하면, 제1 CPU 코어 내지 제3 CPU 코어(CPU1 ~ CPU3)는 인터럽트 불능 구간(DI)이나 또는 커널 비선점 구간(NP)에 위치하여 임계 구간에 위치하고, 제4 CPU 코어(CPU4)는 인터럽트 가능 구간(EN)이면서 동시에 커널 선점가능 구간(P)에 위치하기 때문에 비임계 구간에 위치한다는 것을 알 수 있다.

[0019] 도 4는 도 3의 제1 및 제2 CPU 코어(CPU1, CPU2)와 같이 인터럽트 불능 구간(DI)에 위치한 CPU 코어로 긴급 인터럽트가 전달된 경우를 설명하기 위한 도면이다. 도 4를 참조하면, 긴급 인터럽트 신호는 해당 CPU 코어가 인터럽트 불능 구간(DI)에 위치한 경우에 도착하였으므로, 이 인터럽트 신호는 해당 CPU에 즉시 전달될 수 없으며 인터럽트의 처리가 가능해질 때까지 소정 시간의 처리 지연이 발생한다. 그리고 해당 CPU가 인터럽트 가능 구간(EN)으로 진입한 경우, 즉 긴급 인터럽트 신호의 발생 시점으로부터 소정의 시간 동안 지연된 후에 인터럽트 핸들러(interrupt handler)가 실행된다.

[0020] 도 5는 도 3의 제1 및 제3 CPU 코어(CPU1, CPU3)와 같이 커널 비선점 구간(NP)에 위치한 CPU 코어로 긴급 인터럽트가 전달된 경우를 설명하기 위한 도면이다. 커널 비선점 구간(NP)에서는 긴급 인터럽트에 의해 깨어난 실시간 프로세스로의 긴급 인터럽트의 처리를 위한 컨텍스트 스위치(context switch) 또는 프로세스 스위치

(process switch)가 허용되지 않는다. 도 5에 도시된 바와 같이, 긴급 인터럽트가 인터럽트 가능 구간(EN)에 도착했다고 가정할 경우에 이 긴급 인터럽트에 대한 인터럽트 핸들러는 즉시 실행이 가능하다. 그러나, 인터럽트 핸들러에 의해 깨어난 우선 순위가 높은 실시간 프로세스(RT task)가 있을 경우에, 이 프로세스로의 프로세스 스위치(또는 컨텍스트 스위치)는 커널 비선점 구간(NP)이 종료되고 다시 커널 선점이 가능해질 때까지 지연될 수 밖에 없다. 따라서 긴급 인터럽트를 실행하기 위한 우선 순위가 높은 실시간 프로세스(RT Task)의 선점은, 도 5에 도시된 바와 같이, 커널 선점가능 구간(P)이 될 때까지 지연될 수 밖에 없다.

[0021] 도 6은 도 3의 제4 CPU 코어(CPU4)와 같이 인터럽트 가능 구간(EN)이면서 커널 선점가능 구간(P)에 위치한 CPU 코어로 긴급 인터럽트가 전달된 경우를 설명하기 위한 도면이다. 도 6을 참조하면, 이와 같은 경우에는 인터럽트 핸들러도 즉시 실행될 수 있고 또한 프로세스 스위치도 즉시 이루어질 수 있다. 따라서 인터럽트 가능 구간(EN)이면서 커널 선점가능 구간(P)에 위치한 CPU 코어에서는 긴급 인터럽트가 즉시 처리될 수 있으므로, 긴급 인터럽트의 처리가 가능할 뿐만 아니라 처리 지연 시간의 지연을 최소화할 수 있다.

[0022] 계속해서 도 1a 및 도 1b를 참조하면, 인터럽트 온/오프 관리 장치(4)는 타깃 CPU 세트에 포함되는 각각의 CPU 코어가 현재 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 지시하는 인터럽트 온/오프 정보(interrupt on/off information)를 이용하여, 타깃 CPU 세트에 포함되는 CPU 코어들 가운데 하나 또는 그 이상의 CPU 코어는 비임계 구간에 위치하도록 타깃 CPU 세트의 CPU 코어들을 관리한다. 이것은 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 적어도 하나의 CPU 코어는, 도 6에 도시된 것과 같은 긴급 인터럽트의 실행이 가능한 구간(즉, 비임계 구간)에 위치하도록 하고 이들 CPU 코어로 긴급 인터럽트를 전달함으로써, 긴급 인터럽트 및 긴급 인터럽트에 의해 깨어나는 실시간 프로세스가 인터럽트 불능 구간 및/또는 커널 비선점 구간의 영향을 받지 않고 신속하게(또는 최소 지연시간 내에) 처리할 수 있도록 하기 위해서이다.

[0023] 따라서 멀티코어 컴퓨터에서 2개의 긴급 인터럽트의 실행을 지원하고자 할 경우에는, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 적어도 두 개의 CPU 코어가 비임계 구간에 위치하도록 타깃 CPU 세트의 CPU 코어들을 제어할 수도 있다. 단, 이 경우에는 멀티코어 컴퓨터는 세 개 또는 그 이상의 CPU 코어를 포함해야 하며, 타깃 CPU 세트에도 세 개 또는 그 이상의 CPU 코어가 포함되어야 한다. 두 개의 긴급 인터럽트를 실행할 수 있도록 제어하는 경우는 하나의 긴급 인터럽트를 실행할 수 있도록 제어하는 경우를 단순히 양적으로 확장한 것이므로, 이하 이에 대한 상세한 설명은 생략한다.

[0024] 후술하는 인터럽트 온/오프 관리 장치(4)는 소프트웨어로 운영체제 소스코드에 포함될 수도 있고 또는 하드웨어로 구현되어 소프트웨어와 분리될 수도 있다. 인터럽트 온/오프 관리 장치(4)가 하드웨어로 구현될 경우에 운영체제의 소스 코드의 추가 수정이 필요 없다. 이 경우에, 각 CPU 코어가 운영체제를 통해서 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하려고 할 때, 이 요청은 하드웨어로 구현된 인터럽트 온/오프 관리 장치(4)로 전달되고, 이 하드웨어는 인터럽트 온/오프 정보에 따라서 이러한 구간 진입을 즉시 허락하여 해당 CPU 코어가 인터럽트의 처리가 불가능하도록 하거나 또는 일정한 지연 시간 이후에 허락을 할 수도 있다.

[0025] 인터럽트 온/오프 관리 장치(4)는 타깃 CPU 세트에 포함되는 CPU 코어들을 제어함에 있어서 인터럽트 온/오프 정보를 이용할 수 있다. 여기서, 인터럽트 온/오프 정보는 타깃 CPU 세트에 포함되는 각각의 CPU 코어가 현재 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 지시하는 정보로서, 그 명칭은 임의적인 것이다. 인터럽트 온/오프 정보는, 예를 들어, 인터럽트 불능 비트(interrupt disable bit)를 이용하여 나타낼 수 있다. 이 경우에 해당 CPU 코어가 임계 구간(인터럽트 불능 구간 또는 커널 비선점 구간)에 위치하는 경우에는 '1'로 설정되고, 해당 CPU 코어가 비임계 구간(인터럽트 가능 구간이면서 커널 선점가능 구간)에 위치하는 경우에는 '0'으로 설정된다. 또는, 인터럽트 온/오프 정보는 인터럽트 가능 비트(interrupt enable bit)를 이용하여 나타낼 수도 있는데, 이 경우는 전자의 경우에서 단지 '0'과 '1'의 값만 바뀌는 경우이므로 이에 대한 상세한 설명은 생략한다.

[0026] 타깃 CPU 세트에 포함되는 CPU 코어들 각각에 대한 인터럽트 온/오프 정보는 비트맵(bitmap) 형태로 표현되어 소정의 저장 장치(예컨대, 멀티코어 컴퓨터의 공유 메모리)에 저장되어 있을 수 있다. 예를 들어, 타깃 CPU 세트에 4개의 CPU 코어가 포함되는 경우에, 인터럽트 온/오프 정보는 4비트(각 자리수의 비트는 타깃 CPU 세트에 포함되는 각각의 CPU 코어를 나타낸다. 예컨대, 최상위 비트부터 순차적으로 제4, 제3, 제2, 및 제1 CPU 코어를 지시할 수 있다)의 비트맵을 이용하여 각 CPU 코어가 현재 임계 구간에 위치하는지 또는 비임계 구간에 위치하는지를 표시할 수 있다.

[0027] 이러한 인터럽트 온/오프 정보는 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 적어도 하나의 CPU 코어의 현재 상태가 변경되거나 및/또는 상태의 변경이 없더라도 주기적으로 업데이트가 이루어지도록 관리될 수 있다. 인

터럽트 온/오프 정보는 현재 상태의 변경이 있는 CPU 코어가 직접 변경을 하거나 및/또는 인터럽트 온/오프 관리 장치(4)(예컨대, 제어부(4b))가 대신 변경을 할 수도 있다. 인터럽트 온/오프 정보의 업데이트를 위하여 필요한 경우에, 인터럽트 온/오프 관리 장치(4)는 각 CPU 코어와 정보 교환을 통하여 인터럽트 온/오프 정보와 해당 CPU 코어의 현재 상태를 동기화시킬 수 있다.

[0028] 인터럽트 온/오프 관리 장치(4)는 타깃 CPU 세트 정보도 함께 관리할 수 있다. 예를 들어, 인터럽트 온/오프 관리 장치(4)의 타깃 지정부(4c)가 타깃 CPU 세트 정보를 관리할 수도 있다. 전술한 바와 같이, 멀티코어 컴퓨터에 포함되는 전체 CPU 중에서 타깃 CPU 코어에 포함되는 CPU의 종류 및/개수는 인터럽트 온/오프 관리 장치(4)의 타깃 지정부(4c)가 결정할 수 있다. 또는, 각 CPU 코어는 인터럽트 온/오프 관리 장치(4)(예컨대, 타깃 지정부(4c))와의 정보 교환(타깃 CPU 세트에 포함되기를 요청하는 신호나 타깃 CPU 세트로부터 탈퇴를 요청하는 신호 등)을 통하여 타깃 CPU 코어에 가입하거나 탈퇴할 수 있다. 이러한 각 CPU 코어의 타깃 CPU 세트에로의 가입 및 탈퇴에 관한 사항은 타깃 CPU 비트(target CPU bit) 등과 같은 타깃 CPU 세트 정보로서, 인터럽트 온/오프 관리 장치(4) 또는 공유 메모리에 의하여 저장되어 관리될 수도 있다.

[0029] 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 비임계 구간에 위치하는 어떤 CPU 코어가 비임계 구간에서 임계 구간으로 진입하는 것을 제어함으로써, 타깃 CPU 세트의 CPU 코어들 중에서 적어도 하나의 CPU 코어가 비임계 구간에 위치하도록 제어할 수 있다. 이 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는, 후술하는 바와 같이, 모든 임계 구간으로의 진입을 제어할 수 있고 또는 소정의 기준에 따라서 결정된 일부 임계 구간으로의 진입만을 제어할 수도 있다.

[0030] 전자의 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 비임계 구간에 위치한 CPU 코어가 임계 구간으로 진입하는 것 자체를 제어한다. 반면, 후자의 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 특정한 임계 구간으로 진입하는 것만을 제어할 수 있다. 예를 들어, 제어부(4b)는 인터럽트 불능 구간이나 커널 비선점 구간의 실행 시간이 짧은 경우(즉, 실행 시간이 소정의 기준값 이하인 경우)에는 진입을 허용하지만, 인터럽트 불능 구간이나 커널 비선점 구간의 실행 시간이 소정의 기준값 이상일 경우에만 임계 구간으로의 진입을 제어할 수도 있다. 후자의 경우와 같이, 인터럽트 온/오프 관리 장치(4)에서 제한적으로 임계 구간으로의 진입 제어 기법을 적용하면, 전자의 경우에 비하여 보다 많은 경우에 CPU 코어가 인터럽트 불능 구간이나 커널 비선점 구간으로 진입하는 시도가 불허되어 허가되기 까지 기다려야 하는 경우가 줄어들기 때문에, 타깃 CPU 세트에 포함되는 CPU 코어들의 사용 효율(utilization)을 향상시킬 수가 있다. 이하, 전자의 경우를 중심으로 설명하며, 후자의 경우에도 동일하게 적용될 수 있다는 것은 당업자에게 자명하다.

[0031] 도 7은 인터럽트 온/오프 관리 장치(4)의 제어부(4b)에서 비임계 구간에 위치하는 CPU 코어가 임계 구간으로 진입하는 것을 제어하는 절차의 일례를 보여 주는 흐름도이다. 도 7에 도시된 흐름도는 임계 구간으로서 해당 CPU 코어가 인터럽트 불능 구간에 위치하는 것만이 포함되고, 비임계 구간으로서 CPU 코어가 인터럽트 가능 구간에 위치하는 것만이 포함되는 경우이다.

[0032] 도 7을 참조하면, 인터럽트 온/오프 관리 장치(4)는 타깃 CPU 세트에 포함되는 어느 하나의 CPU 코어가 인터럽트 불능 구간으로 진입하려고 할 때, 인터럽트 온/오프 정보(비트맵)에 대한 로킹(locking)을 획득한다(101). 이것은 서로 다른 두 CPU 코어가 동시에 인터럽트 온/오프 정보(비트맵)을 갱신함에 의해 인터럽트 온/오프 정보(비트맵)의 동기화에 문제가 발생함으로 막기 위한 것이다. 제1 CPU 코어의 입장에서 보면, 인터럽트 불능 구간으로 진입하려고 하는 경우에, 인터럽트 온/오프 관리 장치(4)와 신호를 교환하여 우선 로킹을 획득하거나 또는 자체적으로 로킹을 수행한 다음, 후술하는 단계 102의 절차를 수행하는 것으로 표현할 수 있다. 그리고 인터럽트 온/오프 관리 장치(4)는 제1 CPU 코어와 통신함으로써, 제1 CPU 코어가 인터럽트 불능 구간으로 진입하려고 한다는 것을 알 수 있다.

[0033] 그리고 인터럽트 온/오프 관리 장치(4)의 판단부(4a)는 타깃 CPU 세트에 포함되는 m개의 CPU 코어 중에서 제1 CPU 코어를 제외한 나머지 (m-1)개의 CPU 코어(other\_cpu) 중에서 인터럽트 불능 구간에 위치하는 CPU 코어가 존재하는지를 판단한다(102). 판단부(4a)에서의 이러한 판단 과정은, 저장되어 있는 인터럽트 온/오프 정보를 이용하여 이루어질 수 있다. 예를 들어, 인터럽트 온/오프 정보로서 인터럽트 불능 비트를 이용하는 경우에, 판단부(4a)는 제1 CPU 코어를 제외한 (m-1)개의 CPU 코어(other\_cpu) 중에서 인터럽트 불능 비트의 비트맵에서 '1'로 설정된 것이 있는지를 판단할 수 있다. 판단 결과, (m-1)개의 CPU 코어 전체가 인터럽트 불능 구간에 위치하는 경우에는, 즉 인터럽트 불능 비트의 비트맵이 모두 '1'인 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용하지 않는다. 그리고 인터럽트 온/오프 관리 장치(4)의 판단부(4a)는 상기 판단 단계(102)를 반복하는데, 이러한 판단 단계(102)를 반복한다. 판단 단

계(102)는 예를 들어, 주기적으로 반복하거나 또는 미리 결정된 시간 및/또는 미리 결정된 회수만큼 반복할 수 있다.

[0034] 이와 같이, 판단부(4a)에서 판단 과정(102)을 수행하는 것은, 만일 (m-1)개의 CPU 코어 전체가 인터럽트 불능 구간에 위치하는 경우에도 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용하면, 타깃 CPU 세트에 포함되는 모든 CPU 코어가 인터럽트 불능 구간에 위치하게 되므로, 긴급 인터럽트를 처리할 수가 없기 때문이다. 따라서 타깃 CPU 세트에 포함되는 모든 CPU 코어가 인터럽트 불능 구간에 위치하는 것을 방지할 수 있도록, 본 실시예에서는 판단부(4a)는 판단 단계(102)를 반복하여 수행함으로써, (m-1)개의 CPU 코어 중에서 적어도 하나가 인터럽트 불능 구간에서 인터럽트 가능 구간으로 나올 때까지 기다린다. 그리고 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는, 적어도 하나의 CPU 코어가 인터럽트 가능 구간에 위치하는 것을 확인한 이후에, 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용한다.

[0035] 보다 구체적으로, 단계 102에서의 판단 결과, (m-1)개의 CPU 코어 중에서 적어도 하나가 인터럽트 가능 구간에 위치하는 경우, 즉 인터럽트 불능 비트의 비트맵에서 적어도 하나가 '0'인 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 제1 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용한다. 왜냐하면, (m-1)개의 CPU 코어 중에서 적어도 하나가 인터럽트 가능 구간에 위치하는 경우에는, 제1 CPU 코어는 인터럽트 불능 구간에 들어가더라도 타깃 CPU 세트에 포함되는 CPU 코어에서 적어도 하나의 긴급 인터럽트를 처리하는 데는 아무 문제가 없기 때문이다.

[0036] 이를 위하여, 인터럽트 온/오프 관리 장치(4) 또는 제1 CPU 코어는 제1 CPU 코어에 대한 인터럽트 온/오프 정보를 인터럽트 불능 구간을 지시하는 것으로 설정한다(103). 즉, 제1 CPU 코어에 해당되는 비트맵의 인터럽트 불능 비트(my\_cpu)를 '1'로 설정되도록 한다. 그리고 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 인터럽트 온/오프 정보에 대한 로킹을 해제하여, 제1 CPU 코어가 인터럽트 불능 구간으로 진입하도록 허용한다(104). 즉, 제1 CPU 코어는 로킹을 해제한 후에 인터럽트 불능 구간으로 진입한다.

[0037] 본 발명의 일 실시예에 의하면, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 것을 제어할 필요는 없다. 하지만, 이것이 인터럽트 온/오프 관리 장치(4)가 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 것을 제어하는 것을 배제하는 것이 아니라는 것은 당업자에게 자명하다. 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 것이 인터럽트 온/오프 관리 장치(4)에 의하여 제어되지 않는다고 하더라도, 인터럽트 온/오프 관리 장치(4)는 어떤 CPU 코어가 임계 구간에서 비임계 구간으로 진입하는 지에 관한 정보는 알 필요가 있다.

[0038] 도 8에 도시된 흐름도는 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 절차의 일례를 보여 주는 흐름도이다. 도 8에 도시된 흐름도도 임계 구간으로서 해당 CPU 코어가 인터럽트 불능 구간에 위치하는 것만이 포함되고, 비임계 구간으로서 CPU 코어가 인터럽트 가능 구간에 위치하는 것만이 포함되는 경우이다. 도 8을 참조하면, 인터럽트 불능 구간에서 인터럽트 가능 구간으로 나오려고 하는 제1 CPU 코어는 아무런 제약 없이 인터럽트 가능 구간으로 진입한다(111). 그리고 제1 CPU 코어 또는 인터럽트 온/오프 관리 장치(4)는 제1 CPU 코어에 대한 인터럽트 온/오프 정보(비트맵의 해당 비트, bitmap(my\_cpu))를 '0'으로 해제시킨다(112). 후자의 경우에, 인터럽트 온/오프 관리 장치(4)는 제1 CPU 코어와 신호를 교환함으로써 제1 CPU 코어에 대한 인터럽트 온/오프 정보를 해제할 수 있다.

[0039] 도 9는 인터럽트 온/오프 관리 장치(4)의 제어부(4b)가 비임계 구간에 위치하는 CPU 코어가 임계 구간으로 진입하는 것을 제어하는 절차의 다른 예를 보여 주는 흐름도이다. 도 9에 도시된 흐름도는 임계 구간으로서 해당 CPU 코어가 인터럽트 불능 구간에 위치하거나 또는 커널 비선점 구간에 위치하는 경우이고, 비임계 구간으로서 CPU 코어가 인터럽트 가능 구간이면서 동시에 커널 선점가능 구간에 위치하는 경우이다.

[0040] 도 9를 참조하면, 인터럽트 온/오프 관리 장치(4) 또는 해당 CPU 코어는 타깃 CPU 세트에 포함되는 어느 하나의 CPU 코어가 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하려고 할 때, 인터럽트 온/오프 정보(비트맵)에 대한 로킹(locking)을 획득한다(121). 이것은 서로 다른 두 CPU 코어가 동시에 인터럽트 온/오프 정보(비트맵)을 갱신함에 의해 인터럽트 온/오프 정보(비트맵)의 동기화에 문제가 발생함으로써 막기 위한 것이다. 제2 CPU 코어의 입장에서 보면, 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하려는 하는 경우에, 인터럽트 온/오프 관리 장치(4)와 신호를 교환하여 우선 로킹을 획득한 다음, 후술하는 단계 122의 절차를 수행하는 것으로 표현할 수 있다. 그리고 인터럽트 온/오프 관리 장치(4)는 제2 CPU 코어와 통신함으로써, 제2 CPU 코어가 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하려고 한다는 것을 알 수 있다.

- [0041] 그리고 인터럽트 온/오프 관리 장치(4)의 판단부(4a)는 인터럽트 온/오프 정보에서 제2 CPU 코어의 현재 상태를 체크하여, 제2 CPU 코어가 임계 구간(커널 비선점 구간 또는 인터럽트 불능 구간)에 위치하는지를 판단한다(122). 본 단계는 인터럽트 온/오프 관리 장치(4)가 아니고 제2 CPU에 의하여 수행될 수 있다. 이 경우에, 제2 CPU 코어는 인터럽트 온/오프 정보에서 자신의 상태가 임계 구간에 위치하는지를 판단한다.
- [0042] 판단 결과, 제2 CPU 코어의 인터럽트 온/오프 정보가 설정되어 있는 경우에는, 제2 CPU 코어가 커널 비선점 구간 또는 인터럽트 불능 구간으로 진입할 수 있도록, 단계 121에서 설정된 제2 CPU 코어에 대한 로킹을 해제한다. 본 단계에서 제2 CPU 코어의 인터럽트 온/오프 정보가 설정되어 있는 것으로 판단되는 것은, 제2 CPU 코어가 이미 커널 비선점 구간에 진입해 있으면서 인터럽트 불능 구간으로 진입하려고 하거나 또는 제2 CPU 코어가 이미 인터럽트 불능 구간에 진입해 있으면서 커널 비선점 구간으로 진입하려고 하는 것을 의미한다. 따라서 이 경우는 제2 CPU 코어가 이미 임계 구간, 즉 인터럽트 불능 구간이나 커널 비선점 구간에 위치하는 경우이므로, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 제2 CPU 코어가 커널 비선점 구간이나 인터럽트 불능 구간으로 진입하는 것을 허용해도 무방하다.
- [0043] 반면, 제2 CPU 코어의 인터럽트 온/오프 정보가 설정되어 있지 않은 것으로 판단되는 경우에는, 인터럽트 온/오프 관리 장치(4)의 판단부(4a)는 저장되어 있는 인터럽트 온/오프 정보를 이용하여 타깃 CPU 세트에 포함되는 m개의 CPU 코어 중에서 제2 CPU 코어를 제외한 나머지 (m-1)개의 CPU 코어(other\_cpu) 중에서 인터럽트 불능 구간에 위치하는 CPU 코어가 존재하는지를 판단한다(123). 예를 들어, 인터럽트 온/오프 정보로서 인터럽트 불능 비트를 이용하는 경우에, 제2 CPU 코어를 제외한 (m-1)개의 CPU 코어(other\_cpu) 중에서 인터럽트 불능 비트의 비트맵에서 '1'로 설정된 것이 있는지를 판단할 수 있다. 판단 결과, (m-1)개의 CPU 코어 전체가 인터럽트 불능 구간 또는 커널 비선점 구간에 위치하는 경우에는, 즉 인터럽트 불능 비트의 비트맵이 모두 '1'인 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 제2 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용하지 않는다. 그리고 인터럽트 온/오프 관리 장치(4)의 판단부(4a)는 상기 판단 단계(123)를 반복한다.
- [0044] 이것은 만일 (m-1)개의 CPU 코어 전체가 인터럽트 불능 구간 또는 커널 비선점 구간에 위치하는 경우에도 제2 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용하면, 타깃 CPU 세트에 포함되는 모든 CPU 코어가 인터럽트 불능 구간이나 커널 비선점 구간에 위치하게 되므로, 타깃 CPU 세트에 포함된 모든 CPU 코어는 긴급 인터럽트를 처리할 수가 없기 때문이다. 따라서 타깃 CPU 세트에 포함되는 모든 CPU 코어가 인터럽트 불능 구간 또는 커널 비선점 구간에 위치하는 것을 방지할 수 있도록, 본 실시예에서는 판단 단계(123)를 반복하여 (m-1)개의 CPU 코어 중에서 적어도 하나가 임계 구간에서 인터럽트 가능 구간이면서 동시에 커널 선점가능 구간으로 나올 때까지 기다린 후에, 인터럽트 온/오프 관리 장치(4)는 제2 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용한다.
- [0045] 보다 구체적으로, 단계 123에서의 판단 결과, (m-1)개의 CPU 코어 중에서 적어도 하나가 인터럽트 가능 구간이면서 동시에 커널 선점가능 구간에 위치하는 경우, 즉 인터럽트 불능 비트의 비트맵에서 적어도 하나가 '0'인 경우에, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 제2 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 허용한다. 왜냐하면, (m-1)개의 CPU 코어 중에서 적어도 하나가 인터럽트 가능 구간이면서 커널 선점가능 구간에 위치하는 경우에는, 제2 CPU 코어가 인터럽트 불능 구간이나 커널 비선점 구간에 들어가더라도 타깃 CPU 세트에 포함되는 CPU 코어에서 적어도 하나의 긴급 인터럽트를 처리하는 데는 아무 문제가 없기 때문이다.
- [0046] 이를 위하여, 인터럽트 온/오프 관리 장치(4)는 제2 CPU 코어에 대한 인터럽트 온/오프 정보를 인터럽트 불능 구간을 지시하는 것으로 설정한다(124). 즉, 제2 CPU 코어에 해당되는 비트맵의 인터럽트 불능 비트(my\_cpu)를 '1'로 설정한다. 본 단계는 제2 CPU 코어에 의하여 수행될 수도 있다. 그리고 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 제2 CPU 코어에 대한 로킹을 해제하여, 제2 CPU 코어가 커널 비선점 구간 또는 인터럽트 불능 구간으로 진입하도록 허용한다(125). 즉, 제2 CPU 코어는 로킹을 해제한 후에 커널 비선점 구간 또는 인터럽트 불능 구간으로 진입한다.
- [0047] 본 발명의 일 실시예에 의하면, 인터럽트 온/오프 관리 장치(4)의 제어부(4b)는 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 것을 제어할 필요는 없다. 하지만, 이것이 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 것을 인터럽트 온/오프 관리 장치(4)가 제어하는 것을 배제하는 것이 아니라는 것은 당업자에게 자명하다. 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 것이 인터럽트 온/오프 관리 장치(4)에 의하여 제어되지 않는다고 하더라도, 인터럽트 온/오프 관리 장치(4)는 어떤 CPU 코어가 임계 구간에서 비임계 구간으로 진입하는 지에 관한 정보는 알 필요가 있다.
- [0048] 도 10에 도시된 흐름도는 임계 구간에 위치하는 CPU 코어가 비임계 구간으로 진입하는 절차를 보여 주는 흐름도

로서, 도 10의 (a)는 커널 비선점 구간에서 커널 선점 구간으로 진입하는 경우이고 도 10의 (b)는 인터럽트 불능 구간에서 인터럽트 가능 구간으로 진입하는 경우이다.

[0049] 도 10의 (a)와 (b)를 참조하면, 커널 비선점 구간에서 커널 선점가능 구간으로 나오려고 하거나 또는 인터럽트 불능 구간에서 인터럽트 가능 구간으로 나오려고 하는 제2 CPU 코어는, 아무런 제약 없이 커널 선점가능 구간 또는 인터럽트 가능 구간으로 진입한다(131). 그리고 제2 CPU 코어 또는 인터럽트 온/오프 관리 장치(4)의 판단부(4a)는 제2 CPU 코어가 인터럽트 가능 구간에 위치하는지(도 10의 (a) 참조) 또는 커널 선점가능 구간에 위치하는지(도 10의 (b) 참조)를 판단한다(132). 판단 결과, 제2 CPU 코어 또는 인터럽트 온/오프 관리 장치(4)는 제2 CPU 코어가 인터럽트 가능 구간에 위치하거나 또는 커널 선점가능 구간에 위치하는 경우에는, 제2 CPU 코어에 대한 인터럽트 온/오프 정보(비트맵의 해당 비트, `bitmap(my_cpu)`)를 '0'으로 해제시킨다(133). 후자의 경우에, 인터럽트 온/오프 관리 장치(4)는 제2 CPU 코어와 신호를 교환함으로써 제2 CPU 코어에 대한 인터럽트 온/오프 정보를 해제할 수 있다.

[0050] 도 11은 도 7에 도시된 흐름도에 따라서 멀티코어 프로세서의 인터럽트 온/오프 관리 방법의 일 실시예를 보여주는 도면이다. 도 11에 도시된 도면은 4개의 CPU 코어를 포함하는 멀티코어 컴퓨터에서 타깃 CPU 세트로는 2개의 CPU 코어가 포함되는 경우인데, 제3 CPU 코어(CPU3) 및 제4 CPU 코어(CPU4)가 타깃 CPU 세트에 포함되어 있다. 인터럽트 불능 비트로 표현되는 인터럽트 온/오프 정보(비트맵)는 초기에는 '0000'으로 세팅되어 있는데, 최상위 비트부터 순차적으로 제4 CPU 코어, 제3 CPU 코어, 제2 CPU 코어(CPU2), 및 제1 CPU 코어(CPU1)를 가리킨다. 도 11을 참조하여 후술하는 실시예는 각 CPU 코어를 주체로 하여 서술하지만, 이것이 CPU 코어의 동작의 전부 또는 일부가 전술한 인터럽트 온/오프 관리 장치(4)에 의하여 제어될 수 있다는 것을 배제하지는 않는다.

[0051] 도 11을 참조하면, 인터럽트 불능 구간(DI)으로 진입하려고 하는 제3 CPU 코어는 먼저 다른 CPU 코어(즉, 제4 CPU 코어)의 비트맵을 확인한다. 도 11에서는 제4 CPU 코어에 대한 비트맵은 '0'이므로, 제3 CPU 코어는 자신의 비트맵을 '1'로 설정하여 전체 비트맵을 (0100)으로 설정한 다음, 바로 인터럽트 불능 구간으로 진입한다. 이후에, 인터럽트 불능 구간을 나올 때에는 다시 제3 CPU에 대한 비트맵을 '0'으로 변경하여 전체 비트맵을 (0000)으로 설정한 다음, 인터럽트 가능 구간으로 진입한다.

[0052] 이후에, 제4 CPU 코어는 인터럽트 불능 구간으로 진입하기 전에 다른 CPU 코어(즉, 제3 CPU 코어)의 비트맵을 확인한다. 확인 결과, 제3 CPU 코어에 대한 비트맵의 값이 '0'이므로, 제4 CPU 코어는 자신의 비트맵을 '1'로 설정하여 전체 비트맵을 '1000'으로 설정한 다음, 인터럽트 불능 가능으로 진입한다. 계속해서, 제4 CPU 코어가 인터럽트 불능 구간에 위치한 상태에서 인터럽트 불능 구간으로 진입하려고 하는 제3 CPU 코어는, 다른 CPU 코어(즉, 제4 CPU 코어)의 비트맵을 확인한다.

[0053] 확인 결과, 제4 CPU 코어에 대한 비트맵을 '1'로 설정되어 있으므로, 제3 CPU 코어는 제4 CPU 코어에 대한 비트맵이 '0'이 될 때까지, 즉 제4 CPU 코어가 인터럽트 가능 구간으로 진입할 때까지 기다린다. 이후에, 인터럽트 가능 구간으로 진입하려고 하는 제4 CPU 코어는, 자신에 대한 비트맵을 '0'으로 변경하여 전체 비트맵을 '0000'으로 설정한 다음 인터럽트 불능 구간을 탈출한다. 그리고 제3 CPU 코어는 자신에 대한 비트맵을 '1'로 변경하여 전체 비트맵을 '0100'으로 설정한 다음, 인터럽트 불능 구간으로 진입한다. 이와 같이, 본 실시예에서는 타깃 CPU 세트에 포함되는 모든 CPU 코어들이 동시에 인터럽트 불능 구간에 위치하는 것을 방지한다.

[0054] 계속해서 도 1a 및 도 1b를 참조하면, 멀티코어 컴퓨터는 인터럽트 디스패처(6)를 더 포함할 수 있다. 인터럽트 디스패처(6)는 멀티코어 컴퓨터에서 인터럽트가 발생한 경우에,  $n(\geq 2)$  개의 CPU 코어(2a, 2b, 2c, ..., 2n) 중에서 특정 CPU 코어로 인터럽트를 전달하는 장치이다. 본 실시예에 의하면, 인터럽트 디스패처(6)는 n개의 CPU 코어(2a, 2b, 2c, ..., 2n)가 아닌 타깃 CPU 세트에 포함되는 m개의 CPU 코어에서 선택된 어느 하나의 CPU로 긴급 인터럽트를 전달하거나 또는 m개의 CPU 코어 전체를 위하여 인터럽트 브로드캐스트를 수행할 수 있다.

[0055] 인터럽트 온/오프 관리 장치(4)의 인터럽트 전달부(4d)는 긴급 인터럽트가 발생한 경우에 이를 비임계 구간에 위치하는 CPU 코어에게만 전달되도록 인터럽트 디스패처(6)를 제어할 수도 있다. 이 경우에, 인터럽트 디스패처(6)는, 도 1에 도시된 것과 같이, 인터럽트 온/오프 관리 장치(4)와 구분되는 별도의 블록(하드웨어나 소프트웨어)으로 구현될 수 있다. 또는, 인터럽트 디스패처(6)는 인터럽트 온/오프 관리 장치(4)의 일부로서 인터럽트 온/오프 관리 장치(4)의 내부에 존재하는 소정의 블록으로 구현될 수도 있다. 이 경우에, 인터럽트 전달부(4d)가 인터럽트 디스패처로서의 기능을 수행할 수 있다.

[0056] 중요 인터럽트를 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 비임계 구간에 위치하는 특정 CPU 코어로 전달해

주기 위하여, 인터럽트 디스패치 테이블(interrupt dispatch table)을 이용할 수 있다. 인터럽트 디스패치 테이블은 각 인터럽트는 어떤 CPU 코어에서 처리해야 한다는 정보를 담고 있는 것으로서, 예컨대 전술한 인터럽트 온/오프 정보와 동일한 것일 수 있다. 보다 구체적으로, 각 CPU 코어(또는 인터럽트 온/오프 관리 장치(4))는 임계 구간과 비임계 구간 사이에서 상태가 변경될 때마다, 인터럽트 디스패치 테이블을 수정할 수 있다. 그리고 인터럽트 온/오프 관리 장치(4)는 이 인터럽트 디스패치 테이블을 이용하여, 비임계 구간에 위치한 CPU 코어로 긴급 인터럽트가 전달되도록 제어할 수 있다.

[0057] 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 하나의 CPU 코어만 비임계 구간에 위치하는 경우에는, 인터럽트 온/오프 관리 장치(4)는 비임계 구간에 위치하는 CPU 코어로만 긴급 인터럽트를 전달할 수 있다. 만일, 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 비임계 구간에 위치하는 CPU 코어가 둘 이상인 경우에는, 인터럽트 온/오프 관리 장치(4)는 이 중에서 소정의 알고리즘을 이용하여 선택된 CPU 코어에게만 긴급 인터럽트를 전달할 수도 있다. 상기 알고리즘에 대해서는 특별한 제한이 없는데, 예컨대 가장 낮은 프로세스를 실행 중인 CPU 코어, 부하가 가장 낮은 CPU 코어, 또는 이전에 해당 프로세스를 실행했던 CPU 코어 등을 기준으로 하여 선택된 CPU 코어로 긴급 인터럽트를 전달할 수도 있다.

[0058] 인터럽트 온/오프 관리 장치(4)는 또한 인터럽트 브로드캐스팅을 이용하여 2개 이상의 CPU 코어에게 동시에 긴급 인터럽트를 전달되도록 인터럽트 디스패처(6)를 제어할 수도 있다. 이 경우에는 인터럽트 디스패처(6)가 인터럽트 브로드캐스팅을 지원해야 한다. 그리고 브로드캐스팅되는 인터럽트를 수신한 CPU 코어들 중에서 비임계 구간(인터럽트 가능 구간 또는 인터럽트 가능 구간이면서 커널 선점가능 구간)에 위치하는 CPU 코어만이 해당 인터럽트를 처리하고 다른 CPU 코어는 이를 무시한다. 따라서 인터럽트 온/오프 관리 장치(4)는 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 반드시 하나의 CPU 코어만이 비임계 구간에 위치하게 하도록 제어하거나 또는 둘 이상의 CPU 코어가 비임계 구간에 위치하는 경우에는 소정의 우선 순위에 기초하여 우선 순위가 가장 높은 CPU 코어만이 긴급 인터럽트를 처리하도록 제어할 필요가 있다.

[0059] 도 12는 긴급 인터럽트가 로컬 타이머 인터럽트인 경우에, 본 발명의 실시예를 적용하는 일례를 보여 주는 도면이다. 멀티코어 컴퓨터에서 각 CPU 코어는 자신만의 로컬 타이머 인터럽트를 가지고 있으며, 이 인터럽트에 의하여 각 CPU 코어에서는 특정 시간에 특정 프로세스를 깨우는 등의 작업을 수행한다. 이 로컬 타이머 인터럽트에 본 실시예에 따른 방법을 적용하는데 있어서 문제는, 각 로컬 타이머 인터럽트는 특정 CPU 코어에만 전달되게 고정되어 있어서, 로컬 타이머 인터럽트를 인터럽트 가능 구간/커널 선점가능 구간에 있는 다른 CPU 코어에 전달하는 것이 불가능하다는 것이다. 로컬 타이머 인터럽트는 실시간 프로세스의 스케줄에 있어서 빈번히 사용되므로, 이 문제를 해결할 필요가 있다.

[0060] 이러한 문제를 해결하기 위하여, 긴급 인터럽트가 로컬 타이머 인터럽트인 경우에, 인터럽트 온/오프 관리 장치(4)는 전술한 인터럽트 브로드캐스팅을 이용하여 긴급 인터럽트를 타깃 CPU 세트에 포함되는 모든 CPU 코어들에게 전달한다. 그리고 인터럽트 온/오프 관리 장치(4)는 타깃 CPU 세트에 포함되는 어느 하나의 CPU 코어에 대하여 중요한 로컬 타이머 인터럽트를 세팅할 때에는, 동일한 시간에 동일한 인터럽트가 발생하도록 타깃 CPU 세트에 포함되는 다른 CPU 코어도 세팅한다.

[0061] 도 12에 도시된 바와 같이 타깃 CPU 세트에 제3 CPU 코어(CPU3)와 제4 CPU 코어(CPU4)가 포함되는 경우라고 가정하자. 이 경우에, 제3 CPU 코어에 중요한 로컬 타이머 인터럽트를 세팅할 때에는 동일한 시간에 인터럽트가 발생하도록 제4 CPU 코어도 세팅한다('A' 부분 참조). 마찬가지로, 제4 CPU 코어에 중요한 로컬 타이머 인터럽트를 세팅할 때에는 동일한 시간에 동일한 인터럽트가 발생하도록 제3 CPU 코어도 세팅한다('B' 부분 참조). 이렇게 함으로써, 특정 CPU 코어에 묶여 있는 로컬 타이머 인터럽트에 대해서도, 전술한 인터럽트 브로드캐스팅과 같은 효과를 낼 수 있다. 따라서 전술한 경우에는, 제3 CPU 코어와 제4 CPU 코어 중에서 비임계 구간에 위치하는 CPU 코어만이 로컬 타이머 인터럽트를 처리하도록 한다.

[0062] 이를 위하여, 타깃 CPU 세트에 포함되는 CPU 코어들 사이에는 소정의 정보 교환이 필요하다. 예를 들어, 각 CPU 코어의 로컬 인터럽트 발생 시점에 관한 정보나 인터럽트가 실행해야 하는 함수 등의 정보가 CPU 코어들 사이에서 교환될 수 있다. 이러한 정보의 교환은 CPU 코어들 사이에서 독립적으로 이루어지거나 또는 인터럽트 온/오프 관리 장치(4)를 통해서 이루어질 수 있다.

[0063] 도 13은 타깃 CPU 세트에 두 개의 CPU 코어(제3 CPU 코어(CPU3)와 제4 CPU 코어(CPU4))에 포함되며, 인터럽트 브로드캐스팅을 사용하여 긴급 인터럽트를 전달할 경우에 발생 가능한 네 가지 경우를 예시한 도면이다. 도 13은 임계 구간으로서 인터럽트 불능 구간만을 포함하며, 비임계 구간으로 인터럽트 가능 구간만을 포함하는 경우가 도시되어 있으나, 커널 비선점 구간과 커널 선점가능 구간이 각각 함께 포함될 경우에도 동일하게 적용될 수

있다는 것은 자명하다.

[0064] 먼저, 도 13의 (a)와 같이, 두 CPU 코어가 모두 인터럽트 가능 구간(EN)에 있을 때 긴급 인터럽트가 발생한 경우에, 두 CPU 코어는 모두 즉시 긴급 인터럽트의 처리가 가능하다. 따라서 긴급 인터럽트의 처리 지연이 최소화될 수 있다. 그리고 도 13의 (b)와 같이, 제3 CPU 코어는 인터럽트 가능 구간에 위치하지만 제4 CPU 코어는 인터럽트 불능 구간에 위치할 때 긴급 인터럽트가 발생한 경우에, 비록 제4 CPU 코어는 긴급 인터럽트를 처리하는데 지연이 발생하지만 제3 CPU 코어는 즉시 긴급 인터럽트의 처리가 가능하므로, 역시 긴급 인터럽트의 처리 지연을 최소화할 수 있다. 도 13의 (c)와 같이, 제3 CPU 코어는 인터럽터 불능 구간에 위치하지만 제4 CPU 코어는 인터럽트 가능 구간에 위치할 때 긴급 인터럽트가 발생한 경우에, 비록 제3 CPU 코어는 긴급 인터럽트를 처리하는데 지연이 발생하지만 제4 CPU 코어는 즉시 긴급 인터럽트의 처리가 가능하므로, 역시 긴급 인터럽트의 처리 지연을 최소화할 수 있다.

[0065] 이와 같이, 본 발명의 실시예에서와 같이, 타깃 CPU 세트에 포함되는 CPU 코어들 중에서 적어도 하나의 CPU 코어는 인터럽트 가능 구간에 위치하도록 제어할 경우에는, 인터럽트 디스패처(6)에서 인터럽트 브로드캐스팅을 이용하여 인터럽트를 모든 CPU 코어로 전달하더라도, 긴급 인터럽트의 처리 지연을 최소화할 수 있다. 반면, 도 13의 (d)에 도시된 바와 같이, 제3 및 제4 CPU 코어가 모두 인터럽트 불능 구간에 위치할 때 긴급 인터럽트가 발생하면, 제3 및 제4 CPU 코어 모두 이 긴급 인터럽트를 처리할 수 없으므로, 긴급 인터럽트의 처리 지연이 불가피하다.

[0066] 이상에서 상세히 설명한 바와 같이, 타깃 CPU 코어에 포함되는 CPU 세트들 중에서 적어도 어느 하나의 CPU 코어 (또는 적어도 두 개의 CPU 코어)가 비임계 구간(인터럽트 가능 구간이거나 또는 인터럽트 가능 구간이면서 커널 선점가능 구간)에 위치하도록 제어를 하면, 실시간 프로세스와 같은 긴급 인터럽트의 처리 지연을 방지하거나 최소화할 수 있다. 위에서는 도 1을 참조하여 이러한 인터럽트 온/오프 제어 장치 및 이를 포함하는 멀티코어 컴퓨터에 관해서 설명하였지만, 도 1에 도시된 블록도는 단지 예시적이라는 것은 당업자에게 자명하다.

[0067] 이상의 설명은 본 발명의 일 실시예에 불과할 뿐, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명의 본질적 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현할 수 있을 것이다. 따라서, 본 발명의 범위는 전술한 실시예에 한정되지 않고 특허 청구범위에 기재된 내용과 동등한 범위 내에 있는 다양한 실시 형태가 포함되도록 해석되어야 할 것이다.

**도면의 간단한 설명**

- [0068] 도 1a는 일 실시예에 따른 멀티코어 컴퓨터의 구성의 일례를 보여 주는 블록도이다.
- [0069] 도 1b는 도 1a의 인터럽트 온/오프 관리 장치의 구성의 일례를 보여 주는 블록도이다.
- [0070] 도 2는 4개의 CPU 코어를 포함하는 멀티코어 컴퓨터에서 타깃 CPU 세트를 구성하는 방법의 일례를 보여 주는 도면이다.
- [0071] 도 3은 4개의 CPU 코어로 구성된 타깃 CPU 세트에서 임계 구간에 위치하는 CPU 코어와 비임계 구간에 위치하는 CPU 코어를 도식적으로 표현한 블록도이다.
- [0072] 도 4는 인터럽트 불능 구간(DI)에 위치한 CPU 코어로 긴급 인터럽트가 전달된 경우를 설명하기 위한 도면이다.
- [0073] 도 5는 커널 비선점 구간(NP)에 위치한 CPU 코어로 긴급 인터럽트가 전달된 경우를 설명하기 위한 도면이다.
- [0074] 도 6은 인터럽트 가능 구간(EN)이면서 커널 선점가능 구간(P)에 위치한 CPU 코어로 긴급 인터럽트가 전달된 경우를 설명하기 위한 도면이다.
- [0075] 도 7은 인터럽트 가능 구간에 위치하는 CPU 코어가 인터럽트 불능 구간으로 진입하는 것을 제어하는 절차의 일례를 보여 주는 흐름도이다.
- [0076] 도 8은 인터럽트 불능 구간에 위치하는 CPU 코어가 인터럽트 가능 구간으로 진입하는 절차를 보여 주는 흐름도이다.
- [0077] 도 9는 CPU 코어가 인터럽트 불능 구간 또는 커널 비선점 구간으로 진입하는 것을 제어하는 절차의 일례를 보여 주는 흐름도이다.

[0078] 도 10은 CPU 코어가 인터럽트 가능 구간 또는 커널 비선점 구간으로 진입하는 절차를 보여 주는 흐름도이다.

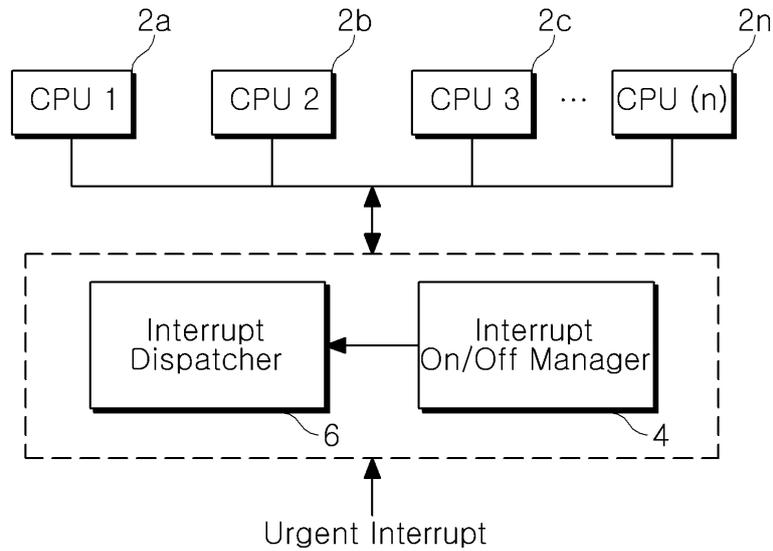
[0079] 도 11은 도 7에 도시된 흐름도에 따라서 멀티코어 프로세서의 인터럽트 온/오프 관리 방법의 일 실시예를 보여 주는 도면이다.

[0080] 도 12는 긴급 인터럽트가 로컬 타이머 인터럽트인 경우에 인터럽트 브로드캐스팅을 이용하는 실시예를 적용하는 일례를 보여 주는 도면이다.

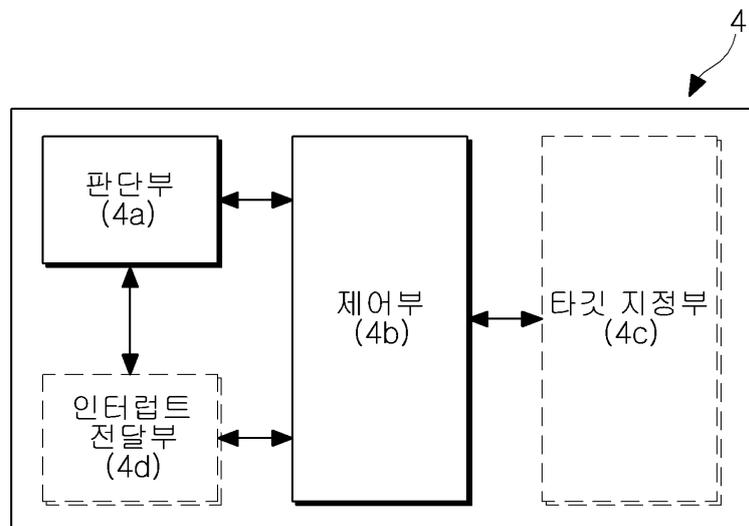
[0081] 도 13은 인터럽트 브로드캐스팅을 사용하여 긴급 인터럽트를 전달할 경우에 발생 가능한 경우의 일례를 보여 주는 도면이다.

도면

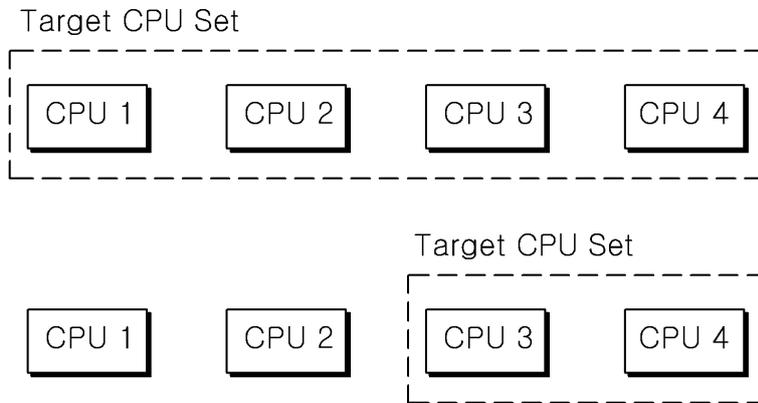
도면1a



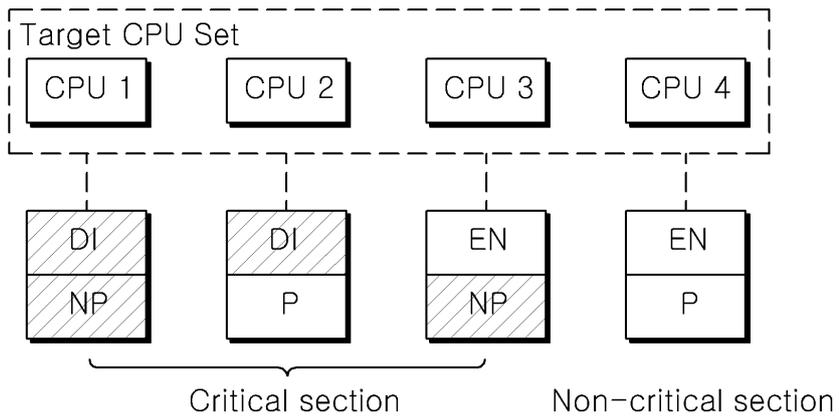
도면1b



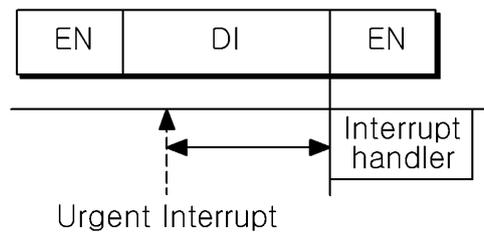
도면2



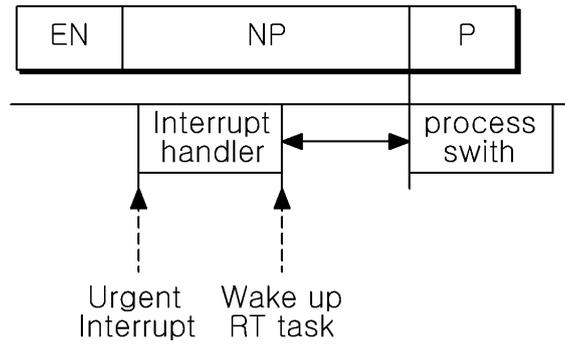
도면3



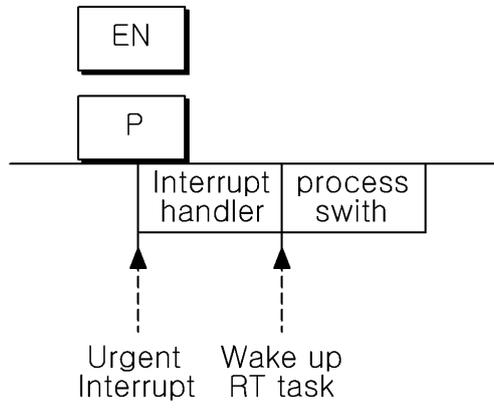
도면4



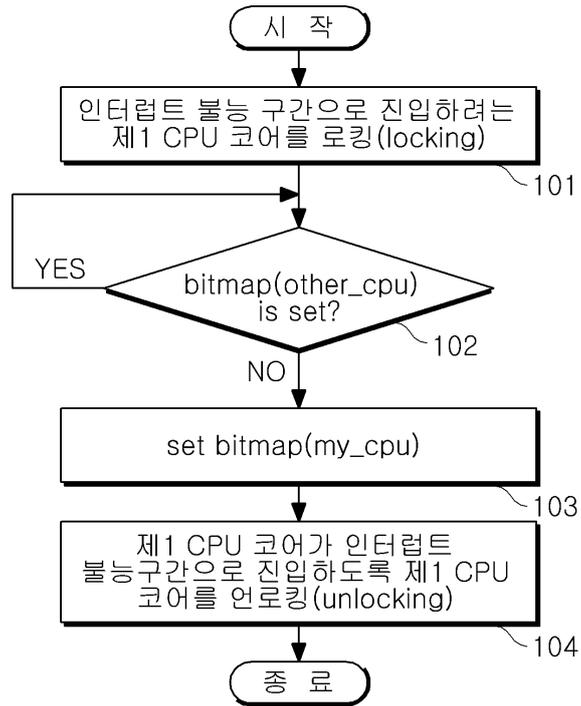
도면5



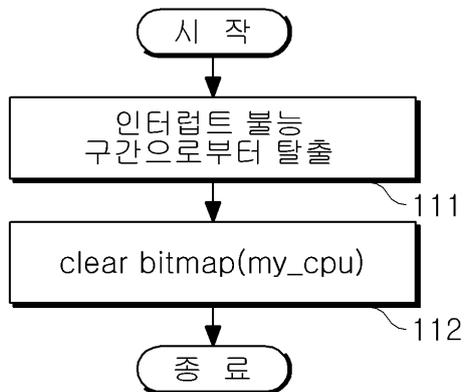
도면6



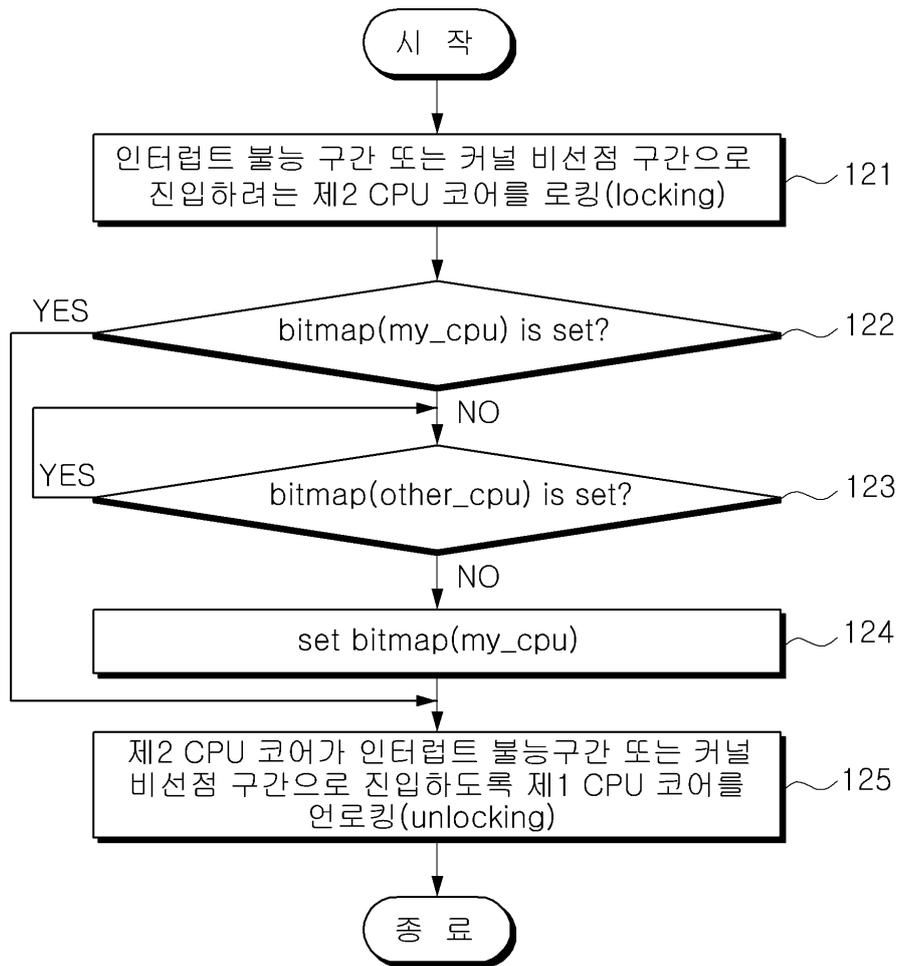
도면7



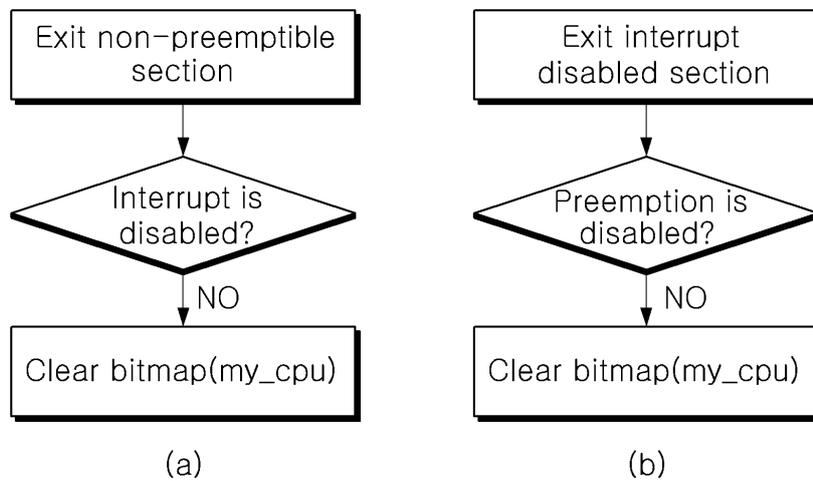
도면8



도면9



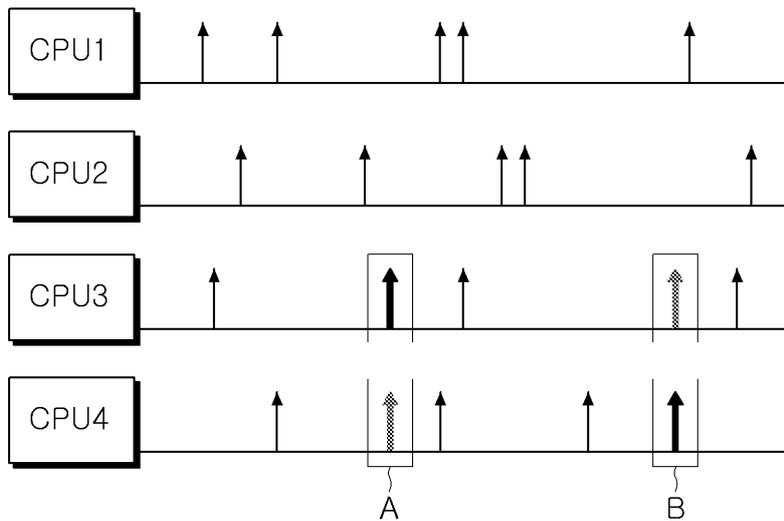
도면10



도면11

bitmap	CPU3	CPU4
0000		
0100	About to enter	
	DI	
0000	exit	
1000		About to enter
	About to enter	DI
0000		
1000	DI	
0000	exit	

도면12



도면13

