



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0061726
(43) 공개일자 2016년06월01일

(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2006.01)

(21) 출원번호 10-2014-0164480

(22) 출원일자 2014년11월24일

심사청구일자 없음

(71) 출원인

삼성전자주식회사

경기도 수원시 영통구 삼성로 129 (매탄동)

(72) 발명자

민정희

경기도 고양시 덕양구 화중로 219, 116동 1602호
(화정동, 달빛마을1단지아파트)

류형우

경상남도 김해시 내외로 55, 309동 701호 (내동,
동아아파트)

라광현

경기도 의왕시 새롭길 21, 201동 2006호 (포일동,
인덕원대우아파트푸른마을)

(74) 대리인

특허법인가산

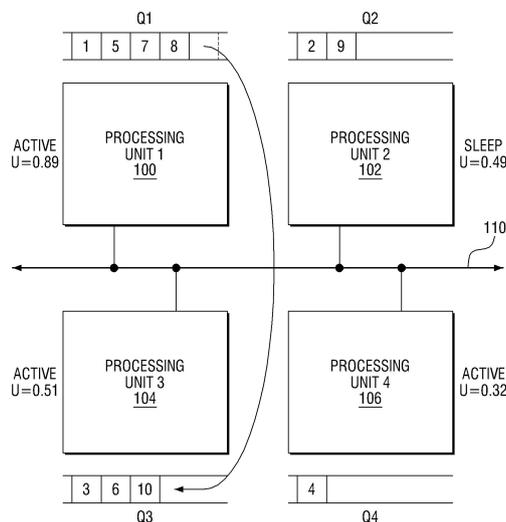
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **인터럽트 핸들링 방법**

(57) 요약

인터럽트 핸들링 방법이 제공된다. 인터럽트 핸들링 방법은 제1 인터럽트를 수신하고, 복수의 프로세싱 유닛 중 제1 프로세싱 유닛의 제1 작업 큐에 상기 제1 인터럽트를 할당하고, 제2 인터럽트를 수신하고, 상기 제1 작업 큐에 상기 제2 인터럽트를 할당하고, 상기 제1 프로세싱 유닛 상에서 상기 제1 작업 큐에 할당된 상기 제1 인터럽트를 처리하고, 상기 제1 인터럽트를 처리하는 동안, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하고, 상기 제1 작업 큐에 할당된 상기 제2 인터럽트를 상기 선정된 제2 프로세싱 유닛의 제2 작업 큐로 이전하는 것을 포함한다.

대표도 - 도3



명세서

청구범위

청구항 1

제1 인터럽트를 수신하고, 복수의 프로세싱 유닛 중 제1 프로세싱 유닛의 제1 작업 큐에 상기 제1 인터럽트를 할당하고,

제2 인터럽트를 수신하고, 상기 제1 작업 큐에 상기 제2 인터럽트를 할당하고,

상기 제1 프로세싱 유닛 상에서 상기 제1 작업 큐에 할당된 상기 제1 인터럽트를 처리하고,

상기 제1 인터럽트를 처리하는 동안, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하고,

상기 제1 작업 큐에 할당된 상기 제2 인터럽트를 상기 선정된 제2 프로세싱 유닛의 제2 작업 큐로 이전하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 2

제1항에 있어서,

상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 3

제2항에 있어서,

상기 각각의 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

액티브(active) 상태인 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 4

제2항에 있어서,

상기 각각의 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 제1 프로세싱 유닛보다 이용률(utilization)이 낮은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 5

제1항에 있어서,

상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛의 작업 큐의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 6

제5항에 있어서,

상기 각각의 프로세싱 유닛의 작업 큐의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선

정하는 것은,

상기 제1 프로세싱 유닛보다 작업 큐에 할당된 인터럽트의 개수가 적은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 7

제1항에 있어서,

상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛에 대한 인터럽트의 발생 빈도에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 8

제7항에 있어서,

상기 각각의 프로세싱 유닛에 대한 인터럽트 발생 빈도에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 제1 프로세싱 유닛보다 인터럽트의 발생 빈도가 낮은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 9

제1항에 있어서,

상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛의 캐시 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 10

제9항에 있어서,

상기 각각의 프로세싱 유닛의 캐시 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 제1 프로세싱 유닛보다 캐시 미스의 발생 빈도 수가 높지 않은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 11

제1항에 있어서,

상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은,

상기 제1 프로세싱 유닛이 펜딩(pending) 상태인 동안, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 12

제1항에 있어서,

상기 선정된 제2 프로세싱 유닛 상에서 상기 제2 작업 큐로 이전된 상기 제2 인터럽트를 처리하는 것을 더 포함하는 인터럽트 핸들링 방법.

청구항 13

제1항에 있어서,

상기 복수의 프로세싱 유닛 중 상기 제2 작업 큐로 이전된 상기 제2 인터럽트를 처리할 제3 프로세싱 유닛을 선

정하고,

상기 제2 작업 큐로 이전된 상기 제2 인터럽트를 상기 선정된 제3 프로세싱 유닛의 제3 작업 큐로 이전하는 것을 더 포함하는 인터럽트 핸들링 방법.

청구항 14

제13항에 있어서,

상기 선정된 제3 프로세싱 유닛 상에서 상기 제3 작업 큐로 이전된 상기 제2 인터럽트를 처리하는 것을 더 포함하는 인터럽트 핸들링 방법.

청구항 15

제13항에 있어서,

상기 제3 프로세서는 상기 제1 프로세서를 포함하고, 상기 제3 작업 큐는 상기 제1 작업 큐를 포함하는 인터럽트 핸들링 방법.

청구항 16

제1항에 있어서,

상기 제1 프로세싱 유닛 및 상기 제2 프로세싱 유닛은 각각 제1 CPU(Central Processing Unit) 및 제2 CPU를 포함하는 인터럽트 핸들링 방법.

청구항 17

제1항에 있어서,

상기 제1 프로세싱 유닛 및 상기 제2 프로세싱 유닛은 각각 제1 코어 및 제2 코어를 포함하는 인터럽트 핸들링 방법.

청구항 18

복수의 프로세싱 유닛에 대해 복수의 인터럽트를 할당하고,

상기 복수의 인터럽트의 개수가 상기 복수의 프로세싱 유닛의 개수보다 많은 경우, 상기 복수의 프로세싱 유닛 중, 제1 인터럽트 및 제2 인터럽트를 포함하는 둘 이상의 인터럽트가 할당된 제1 프로세싱 유닛에 대해, 상기 제1 인터럽트는 상기 제1 프로세싱 유닛을 이용하여 처리하고, 상기 제2 인터럽트는 상기 복수의 프로세싱 유닛 중에서 선정된 제2 프로세싱 유닛을 이용하여 처리하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 19

복수의 프로세싱 유닛 중 제1 프로세싱 유닛의 제1 작업 큐에 삽입할 제1 인터럽트를 수신하고,

상기 제1 작업 큐의 상태를 모니터링하고,

상기 제1 작업 큐에 기 삽입된 인터럽트의 개수가 제1 임계값을 초과하는 경우, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하고,

상기 제1 인터럽트를 상기 제2 프로세싱 유닛의 제2 작업 큐에 삽입하는 것을 포함하는 인터럽트 핸들링 방법.

청구항 20

복수의 프로세싱 유닛 중 제1 프로세싱 유닛에 지정된 제1 인터럽트를 수신하여 상기 제1 프로세싱 유닛의 제1 작업 큐에 삽입하고,

상기 제1 프로세싱 유닛에 지정된 제2 인터럽트를 수신하고,

상기 제1 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하고,

상기 복수의 프로세싱 유닛 중 상기 제2 프로세싱 유닛의 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하고,

상기 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간이 상기 제1 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간보다 짧은 경우, 상기 제2 인터럽트를 상기 제2 작업 큐에 삽입하는 것을 포함하는 인터럽트 핸들링 방법.

발명의 설명

기술 분야

[0001] 본 발명은 인터럽트 핸들링 방법에 관한 것이다.

배경 기술

[0002] 컴퓨팅 시스템에 예컨대 데이터 입출력 작업을 위한 다수의 인터럽트가 발생하면, 컴퓨팅 시스템을 운용하는 운영체제는 발생된 다수의 인터럽트를 컴퓨팅 시스템을 이루는 다양한 자원을 이용하여 처리한다. 특히 다중 프로세서 또는 다중 코어를 포함하는 컴퓨팅 시스템에 있어서는 인터럽트를 신속하고 효율적으로 처리하기 위해 자원을 적절하게 선정할 필요가 있다. 이에 따라 컴퓨팅 시스템의 처리 능력 또는 상태를 고려하여 다수의 인터럽트를 최적의 자원에게 할당하는 방안이 요구된다.

[0003] 일본공개특허 제2008-225641호는 인터럽트 제어 방법 및 그 프로그램을 개시하고 있다.

발명의 내용

해결하려는 과제

[0004] 본 발명이 해결하고자 하는 기술적 과제는 컴퓨팅 시스템의 처리 능력 또는 상태를 기초로 다수의 인터럽트를 효율적으로 처리하기 위한 자원을 선정하는 인터럽트 핸들링 방법을 제공하는 것이다.

[0005] 본 발명의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0006] 상기 기술적 과제를 달성하기 위한 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법은, 제1 인터럽트를 수신하고, 복수의 프로세싱 유닛 중 제1 프로세싱 유닛의 제1 작업 큐에 제1 인터럽트를 할당하고, 제2 인터럽트를 수신하고, 제1 작업 큐에 제2 인터럽트를 할당하고, 제1 프로세싱 유닛 상에서 제1 작업 큐에 할당된 제1 인터럽트를 처리하고, 제1 인터럽트를 처리하는 동안, 복수의 프로세싱 유닛 중 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하고, 제1 작업 큐에 할당된 제2 인터럽트를 선정된 제2 프로세싱 유닛의 제2 작업 큐로 이전하는 것을 포함한다.

[0007] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.

[0008] 본 발명의 몇몇의 실시예에서, 상기 각각의 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 액티브(active) 상태인 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.

[0009] 본 발명의 몇몇의 실시예에서, 상기 각각의 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 이용률(utilization)이 낮은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.

[0010] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛의 작업 큐의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.

[0011] 본 발명의 몇몇의 실시예에서, 상기 각각의 프로세싱 유닛의 작업 큐의 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 작업 큐에 할당된 인터럽트의 개수가 적은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.

- [0012] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛에 대한 인터럽트의 발생 빈도에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0013] 본 발명의 몇몇의 실시예에서, 상기 각각의 프로세싱 유닛에 대한 인터럽트 발생 빈도에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 인터럽트의 발생 빈도가 낮은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.
- [0014] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 복수의 프로세싱 유닛 중 각각의 프로세싱 유닛의 캐시 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0015] 본 발명의 몇몇의 실시예에서, 상기 각각의 프로세싱 유닛의 캐시 상태에 기초하여 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 캐시 미스의 발생 빈도 수가 높지 않은 프로세싱 유닛을 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.
- [0016] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛이 펜딩(pending) 상태인 동안, 상기 복수의 프로세싱 유닛 중 상기 제2 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0017] 본 발명의 몇몇의 실시예에서, 상기 인터럽트 핸들링 방법은, 상기 선정된 제2 프로세싱 유닛 상에서 상기 제2 작업 큐로 이전된 상기 제2 인터럽트를 처리하는 것을 더 포함할 수 있다.
- [0018] 본 발명의 몇몇의 실시예에서, 상기 인터럽트 핸들링 방법은, 상기 복수의 프로세싱 유닛 중 상기 제2 작업 큐로 이전된 상기 제2 인터럽트를 처리할 제3 프로세싱 유닛을 선정하고, 상기 제2 작업 큐로 이전된 상기 제2 인터럽트를 상기 선정된 제3 프로세싱 유닛의 제3 작업 큐로 이전하는 것을 더 포함할 수 있다.
- [0019] 본 발명의 몇몇의 실시예에서, 상기 인터럽트 핸들링 방법은, 상기 선정된 제3 프로세싱 유닛 상에서 상기 제3 작업 큐로 이전된 상기 제2 인터럽트를 처리하는 것을 더 포함할 수 있다.
- [0020] 본 발명의 몇몇의 실시예에서, 상기 제3 프로세서는 상기 제1 프로세서를 포함하고, 상기 제3 작업 큐는 상기 제1 작업 큐를 포함할 수 있다.
- [0021] 본 발명의 몇몇의 실시예에서, 상기 제1 프로세싱 유닛 및 상기 제2 프로세싱 유닛은 각각 제1 CPU(Central Processing Unit) 및 제2 CPU를 포함할 수 있다.
- [0022] 본 발명의 몇몇의 실시예에서, 상기 제1 프로세싱 유닛 및 상기 제2 프로세싱 유닛은 각각 제1 코어 및 제2 코어를 포함할 수 있다.
- [0023] 본 발명의 몇몇의 실시예에서, 상기 제1 코어 및 상기 제2 코어는 동일한 프로세서 내에 포함될 수 있다.
- [0024] 상기 기술적 과제를 달성하기 위한 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법은, 복수의 프로세싱 유닛에 대해 복수의 인터럽트를 할당하고, 복수의 인터럽트의 개수가 복수의 프로세싱 유닛의 개수보다 많은 경우, 복수의 프로세싱 유닛 중, 제1 인터럽트 및 제2 인터럽트를 포함하는 둘 이상의 인터럽트가 할당된 제1 프로세싱 유닛에 대해, 제1 인터럽트는 제1 프로세싱 유닛을 이용하여 처리하고, 제2 인터럽트는 복수의 프로세싱 유닛 중에서 선정된 제2 프로세싱 유닛을 이용하여 처리하는 것을 포함한다.
- [0025] 본 발명의 몇몇의 실시예에서, 상기 제1 인터럽트는 상기 제1 프로세싱 유닛을 이용하여 처리하고, 상기 제2 인터럽트는 상기 복수의 프로세싱 유닛 중에서 선정된 제2 프로세싱 유닛을 이용하여 처리하는 것은, 상기 제1 인터럽트를 상기 제1 프로세싱 유닛을 이용하여 처리하는 동안, 상기 복수의 프로세싱 유닛 중에서 상기 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0026] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중에서 상기 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 이용률(utilization)이 낮은 프로세싱 유닛을 상기 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.
- [0027] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중에서 상기 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 작업 큐에 할당된 인터럽트의 개수가 적은 프로세싱 유닛을 상기 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.

- [0028] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중에서 상기 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 인터럽트의 발생 빈도가 낮은 프로세싱 유닛을 상기 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.
- [0029] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중에서 상기 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 프로세싱 유닛보다 캐시 미스의 발생 빈도 수가 높지 않은 프로세싱 유닛을 상기 제2 프로세싱 유닛으로 선정하는 것을 포함할 수 있다.
- [0030] 본 발명의 몇몇의 실시예에서, 상기 제1 인터럽트는 상기 제1 프로세싱 유닛을 이용하여 처리하고, 상기 제2 인터럽트는 상기 복수의 프로세싱 유닛 중에서 선정된 제2 프로세싱 유닛을 이용하여 처리하는 것은, 상기 제1 인터럽트를 상기 제1 프로세싱 유닛을 이용하여 처리하는 동안, 상기 제2 인터럽트를 상기 제2 프로세싱 유닛의 작업 큐로 이전하는 것을 포함할 수 있다.
- [0031] 상기 기술적 과제를 달성하기 위한 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법은, 복수의 프로세싱 유닛 중 제1 프로세싱 유닛의 제1 작업 큐에 삽입할 제1 인터럽트를 수신하고, 제1 작업 큐의 상태를 모니터링하고, 제1 작업 큐에 기 삽입된 인터럽트의 개수가 제1 임계값을 초과하는 경우, 복수의 프로세싱 유닛 중 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하고, 제1 인터럽트를 제2 프로세싱 유닛의 제2 작업 큐에 삽입하는 것을 포함한다.
- [0032] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제1 작업 큐에 기 삽입된 인터럽트를 상기 제1 프로세싱 유닛을 이용하여 처리하는 동안, 상기 복수의 프로세싱 유닛 중에서 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0033] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제2 작업 큐의 상태를 모니터링하고, 상기 제2 작업 큐에 기 삽입된 인터럽트의 개수가 제2 임계값 이하인 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0034] 본 발명의 몇몇의 실시예에서, 상기 제1 임계값 및 상기 제2 임계값은 서로 동일할 수 있다.
- [0035] 본 발명의 몇몇의 실시예에서, 상기 인터럽트 핸들링 방법은, 상기 제1 프로세싱 유닛의 상태를 모니터링하고, 상기 제1 프로세싱 유닛이 비액티브(inactive) 상태인 경우, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 더 포함할 수 있다.
- [0036] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제2 프로세싱 유닛의 상태를 모니터링하고, 상기 제2 프로세싱 유닛이 액티브(active) 상태인 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0037] 본 발명의 몇몇의 실시예에서, 상기 인터럽트 핸들링 방법은, 상기 제1 프로세싱 유닛의 이용률(utilization)을 모니터링하고, 상기 제1 프로세싱 유닛의 이용률이 제3 임계값을 초과하는 경우, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 더 포함할 수 있다.
- [0038] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제2 프로세싱 유닛의 이용률을 모니터링하고, 상기 제2 프로세싱 유닛의 이용률이 제4 임계값 이하인 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0039] 본 발명의 몇몇의 실시예에서, 상기 인터럽트 핸들링 방법은, 상기 제1 프로세싱 유닛에 지정되어 수신된 인터럽트의 발생 빈도를 모니터링하고, 상기 제1 프로세싱 유닛에 지정되어 수신된 인터럽트의 발생 빈도가 제5 임계값을 초과하는 경우, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것을 더 포함할 수 있다.
- [0040] 본 발명의 몇몇의 실시예에서, 상기 복수의 프로세싱 유닛 중 상기 제1 인터럽트를 처리할 제2 프로세싱 유닛을 선정하는 것은, 상기 제2 프로세싱 유닛에 지정되어 수신된 인터럽트의 발생 빈도를 모니터링하고, 상기 제2 프로세싱 유닛에 지정되어 수신된 인터럽트의 발생 빈도가 제6 임계값 이하인 제2 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0041] 상기 기술적 과제를 달성하기 위한 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법은, 복수의 프로세싱 유닛 중 제1 프로세싱 유닛에 지정된 제1 인터럽트를 수신하여 제1 프로세싱 유닛의 제1 작업 큐에 삽입하고,

제1 프로세싱 유닛에 지정된 제2 인터럽트를 수신하고, 제1 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하고, 복수의 프로세싱 유닛 중 제2 프로세싱 유닛의 제2 작업 큐에서의 제2 인터럽트의 처리 대기 시간을 연산하고, 제2 작업 큐에서의 제2 인터럽트의 처리 대기 시간이 제1 작업 큐에서의 제2 인터럽트의 처리 대기 시간보다 짧은 경우, 제2 인터럽트를 제2 작업 큐에 삽입하는 것을 포함한다.

[0042] 본 발명의 몇몇의 실시예에서, 상기 제1 작업 큐 또는 상기 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것은, 상기 제1 프로세싱 유닛을 이용하여 상기 제1 인터럽트를 처리하는 동안 상기 제1 작업 큐 또는 상기 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것을 포함할 수 있다.

[0043] 본 발명의 몇몇의 실시예에서, 상기 제1 작업 큐 또는 상기 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것은, 상기 제1 작업 큐에 기 삽입된 인터럽트의 개수 또는 상기 제2 작업 큐에 기 삽입된 인터럽트의 개수에 기초하여 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것을 포함할 수 있다.

[0044] 본 발명의 몇몇의 실시예에서, 상기 제1 작업 큐 또는 상기 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것은, 상기 제1 프로세싱 유닛의 상태 또는 상기 제2 프로세싱 유닛의 상태에 기초하여 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것을 포함할 수 있다.

[0045] 본 발명의 몇몇의 실시예에서, 상기 제1 작업 큐 또는 상기 제2 작업 큐에서의 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것은, 상기 제1 프로세싱 유닛에 대한 인터럽트의 발생 빈도 또는 상기 제2 프로세싱 유닛에 대한 인터럽트의 발생 빈도에 기초하여 상기 제2 인터럽트의 처리 대기 시간을 연산하는 것을 포함할 수 있다.

[0046] 기타 실시예들의 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.

도면의 간단한 설명

[0047] 도 1은 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 컴퓨팅 시스템을 설명하기 위한 개략도이다.

도 2는 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법이 수행되는 컴퓨팅 시스템을 설명하기 위한 개략도이다.

도 3은 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

도 4는 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

도 5는 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

도 6은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

도 7은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

도 8은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

도 9는 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 다중 프로세서를 포함하는 컴퓨팅 시스템을 설명하기 위한 개략도이다.

도 10은 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 다중 코어를 포함하는 컴퓨팅 시스템을 설명하기 위한 개략도이다.

도 11은 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.

도 12는 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.

도 13은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.

도 14는 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.

도 15 내지 도 17은 본 발명의 실시예들에 따른 반도체 장치를 적용할 수 있는 예시적인 반도체 시스템들이다.

발명을 실시하기 위한 구체적인 내용

[0048] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로

다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 도면에서 층 및 영역들의 상대적인 크기는 설명의 명료성을 위해 과장된 것일 수 있다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.

- [0049] 하나의 소자(elements)가 다른 소자와 "접속된(connected to)" 또는 "커플링된(coupled to)" 이라고 지칭되는 것은, 다른 소자와 직접 연결 또는 커플링된 경우 또는 중간에 다른 소자를 개재한 경우를 모두 포함한다. 반면, 하나의 소자가 다른 소자와 "직접 접속된(directly connected to)" 또는 "직접 커플링된(directly coupled to)"으로 지칭되는 것은 중간에 다른 소자를 개재하지 않은 것을 나타낸다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다. "및/또는"은 언급된 아이템들의 각각 및 하나 이상의 모든 조합을 포함한다.
- [0050] 소자(elements) 또는 층이 다른 소자 또는 층의 "위(on)" 또는 "상(on)"으로 지칭되는 것은 다른 소자 또는 층의 바로 위뿐만 아니라 중간에 다른 층 또는 다른 소자를 개재한 경우를 모두 포함한다. 반면, 소자가 "직접 위(directly on)" 또는 "바로 위"로 지칭되는 것은 중간에 다른 소자 또는 층을 개재하지 않은 것을 나타낸다.
- [0051] 비록 제1, 제2 등이 다양한 소자, 구성요소 및/또는 섹션들을 서술하기 위해서 사용되나, 이들 소자, 구성요소 및/또는 섹션들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 소자, 구성요소 또는 섹션들을 다른 소자, 구성요소 또는 섹션들과 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제1 소자, 제1 구성요소 또는 제1 섹션은 본 발명의 기술적 사상 내에서 제2 소자, 제2 구성요소 또는 제2 섹션일 수도 있음은 물론이다.
- [0052] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소, 단계, 동작 및/또는 소자는 하나 이상의 다른 구성요소, 단계, 동작 및/또는 소자의 존재 또는 추가를 배제하지 않는다.
- [0053] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.
- [0054] 도 1은 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 컴퓨팅 시스템을 설명하기 위한 개략도이다.
- [0055] 도 1을 참조하면, 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 컴퓨팅 시스템(1)은 하드웨어(10), 운영체제(20) 및 어플리케이션(30)을 포함할 수 있다.
- [0056] 운영체제(20)는 하드웨어(10)를 제어하고 및 어플리케이션(30)의 실행을 지원하는 등 컴퓨팅 시스템(1)을 전반적으로 운용한다. 예를 들어, 운영체제(20)는 어플리케이션(30)으로부터 작업 요청을 수신하고, 요청받은 작업을 처리하기 위한 일련의 작업들을 설정하고, 하드웨어(10)에 이들 작업을 할당할 수 있다. 또한, 운영체제(20)는 하드웨어(10)를 이용하여 처리한 일련의 작업들의 결과를 어플리케이션(30)에 전달할 수 있다.
- [0057] 본 발명의 몇몇의 실시예에서, 운영체제(20)는 애플(Apple, Inc)사의 OSX, 마이크로소프트(Microsoft)사의 윈도 우즈(Windows), 유닉스(UNIX), 리눅스(Linux) 등의 운영체제일 수도 있고, 애플(Apple, Inc)사의 iOS, 구글(Google)사의 안드로이드(Android) 등 모바일 장치에 특화된 운영체제일 수도 있으며, 이러한 예들에 한정되지 않는다.
- [0058] 한편, 본 발명의 몇몇의 실시예에서, 하드웨어(10)는 CPU(Central Processing Unit), GPU(Graphic Processing Unit), AP(Application Processor), CP(Cellular Processor), DSP(Digital Signal Processor) 등을 비롯한 프로세싱 유닛, ROM(Read Only Memory), RAM(Random Access Memory) 등을 비롯한 메모리 장치, HDD(Hard Disk Drive), SSD(Solid State Drive) 등을 비롯한 스토리지 장치 및 기타 주변 장치들을 포함할 수 있으며, 이러한 예들에 한정되지 않는다.
- [0059] 특히, 본 발명의 몇몇의 실시예에서, 프로세싱 유닛은 다중 프로세싱 유닛(12)일 수 있다. 예를 들어, 다중 프로세싱 유닛(12)은 복수의 프로세서, 예컨대 복수의 CPU를 포함하는 다중 프로세서일 수 있다. 이와 다르게, 다중 프로세싱 유닛(12)은 복수의 코어를 포함하는 다중 코어 프로세서일 수 있다.

- [0060] 도 1을 계속 참조하면, 어플리케이션(30)은 사용자로부터 데이터 입출력 요청(USER REQUEST)을 수신하여 운영체제(20)에 대해 인터럽트(INTERRUPT)를 발생시킬 수 있다. 운영체제(20)는 인터럽트 핸들러(24)를 이용하여 어플리케이션(30)으로부터 발생된 인터럽트(INTERRUPT)를 처리할 수 있다. 구체적으로 운영체제(20)는 인터럽트 핸들러(24)를 이용하여 인터럽트(INTERRUPT)를 처리하기 위해 필요한 명령 및 데이터(COMMAND, DATA)를 하드웨어(10)에 전달하고, 하드웨어(10)를 이용하여 인터럽트(INTERRUPT)를 처리할 수 있다.
- [0061] 본 발명의 몇몇의 실시예에서, 인터럽트(INTERRUPT)는 하드웨어(10)의 다중 프로세싱 유닛(12)을 이용하여 처리될 수 있다. 이 때, 운영체제(20) 내의 프로세스 관리부(22)는 본 발명의 다양한 실시예에 따른 인터럽트 핸들링 방법을 수행할 수 있다. 구체적으로 프로세스 관리부(22)는 처리할 인터럽트(INTERRUPT)를 다중 프로세싱 유닛(12)에 적절히 할당할 수 있다. 본 발명의 몇몇의 실시예에서, 프로세스 관리부(22)는 운영체제(20)의 일부로서 소프트웨어로 구현될 수 있으나, 구체적인 구현 형태가 이에 한정되는 것은 아니다. 프로세스 관리부(22)의 동작에 관한 구체적인 설명은 도 3 내지 도 8을 참조하여 후술하도록 한다.
- [0062] 도 2는 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법이 수행되는 컴퓨팅 시스템을 설명하기 위한 개략도이다.
- [0063] 도 2를 참조하면, 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법이 수행되는 컴퓨팅 시스템은 복수의 프로세싱 유닛(100, 102, 104, 106) 및 복수의 프로세싱 유닛(100, 102, 104, 106)에 각각 구비된 작업 큐(Q1, Q2, Q3, Q4)를 포함한다. 복수의 프로세싱 유닛(100, 102, 104, 106)은 버스(110)를 통해 서로 데이터를 주고 받을 수 있다.
- [0064] 본 발명의 몇몇의 실시예에서, 제1 프로세싱 유닛(100)은 제1 CPU를 포함하고, 제2 프로세싱 유닛(102)은 제2 CPU를 포함하고, 제3 프로세싱 유닛(104)은 제3 CPU를 포함하고, 제4 프로세싱 유닛(106)은 제4 CPU를 포함할 수 있다. 즉, 복수의 프로세싱 유닛(100, 102, 104, 106)은 하나의 다중 프로세서를 형성할 수 있다.
- [0065] 이와 다르게, 본 발명의 몇몇의 실시예에서, 제1 프로세싱 유닛(100)은 제1 코어를 포함하고, 제2 프로세싱 유닛(102)은 제2 코어를 포함하고, 제3 프로세싱 유닛(104)은 제3 코어를 포함하고, 제4 프로세싱 유닛(106)은 제4 코어를 포함할 수 있다. 즉, 복수의 프로세싱 유닛(100, 102, 104, 106)은 다중 코어를 포함하는 하나의 프로세서를 형성할 수 있다.
- [0066] 도 2를 계속 참조하면, 제1 프로세싱 유닛(100)은 제1 프로세싱 유닛(100)에서 처리할 작업들을 관리하기 위한 작업 큐(Q1)를 구비할 수 있다. 제1 프로세싱 유닛(100)에서 처리할 작업들이 할당이 되었는데 제1 프로세싱 유닛(100)이 다른 작업을 처리하고 있어 대기해야 하는 경우, 대기해야 할 작업은 작업 큐(Q1)에 삽입될 수 있다. 제1 프로세싱 유닛(100)이 다른 작업 처리를 완료한 경우, 작업 큐(Q1)에 삽입되었던 작업은 작업 큐(Q1)로부터 인출될 수 있다. 이후 인출된 작업은 제1 프로세싱 유닛(100)에 의해 처리될 수 있다. 이러한 내용은 제2 프로세싱 유닛 내지 제4 프로세싱 유닛(102, 104, 106)에 각각 구비된 작업 큐(Q2, Q3, Q4)에도 동일하게 적용되므로, 중복되는 설명은 생략하도록 한다.
- [0067] 본 발명의 몇몇의 실시예에서, 작업 큐(Q1, Q2, Q3, Q4)는 운영체제(20)에 의해 관리될 수 있다. 즉, 작업 큐(Q1, Q2, Q3, Q4)는 운영체제(20)에 의해 생성되고, 유지되고, 삭제될 수 있다. 본 발명의 몇몇의 실시예에서, 작업 큐(Q1, Q2, Q3, Q4)는 우선순위 큐(priority queue)로서 구현될 수 있으나, 구체적인 구현 방식은 이러한 예에 한정되지 않는다.
- [0068] 도 2를 계속 참조하면, 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에는 5 개의 작업이 삽입되어 있다. 예를 들어, 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에는 5 개의 인터럽트가 할당되어 있다. 또한, 제2 프로세싱 유닛(102)의 제2 작업 큐(Q2)에는 2 개의 인터럽트가, 제3 프로세싱 유닛(104)의 제3 작업 큐(Q3)에는 2 개의 인터럽트가, 그리고 제4 프로세싱 유닛(106)의 제4 작업 큐(Q4)에는 1 개의 인터럽트가 할당되어 있다.
- [0069] 엄밀히 말해 프로세싱 유닛(100, 102, 104, 106)의 각각의 작업 큐(Q1, Q2, Q3, Q4)에 할당되는 것은 인터럽트를 처리하기 위한 일련의 세부 작업들일 수 있다. 그러나 설명의 편의를 위해, 프로세싱 유닛(100, 102, 104, 106)의 각각의 작업 큐(Q1, Q2, Q3, Q4)에 인터럽트가 삽입 또는 할당된다는 표현은 프로세싱 유닛(100, 102, 104, 106)의 각각의 작업 큐(Q1, Q2, Q3, Q4)에 인터럽트를 처리하기 위한 일련의 세부 작업이 삽입 또는 할당된다는 의미를 포함하는 것으로 한다.
- [0070] 도 3은 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.
- [0071] 도 3을 참조하면, 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법에서, 운영체제(20)는 제1 인터럽트를 수신

하고, 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 제1 인터럽트를 할당한다. 여기서 제1 인터럽트는 예컨대 도 3에 도시된 제1 작업 큐(Q1)에 기 삽입된 1번 인터럽트, 5번 인터럽트, 7번 인터럽트 및 8번 인터럽트 중 어느 하나 일 수 있다. 도 1과 관련하여 앞서 설명한 바와 같이 도 3에 도시된 인터럽트들은 예컨대 데이터 입출력 작업을 처리하기 위한 인터럽트일 수 있다.

[0072] 다음으로 운영체제(20)는 제2 인터럽트를 수신하고, 제1 작업 큐(Q1)에 제2 인터럽트를 할당할 수 있다. 여기서 제2 인터럽트는 도 3에 도시된 10번 인터럽트로서, 제1 작업 큐(Q1)에 기 삽입된 1번 인터럽트, 5번 인터럽트, 7번 인터럽트 및 8번 인터럽트 이후에 운영체제(20)가 수신한 인터럽트를 말한다.

[0073] 운영체제는(20), 구체적으로 프로세스 관리부(22)는 제1 프로세싱 유닛(100) 상에서 제1 작업 큐(Q1)에 할당된 제1 인터럽트가 처리되는 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다. 본 실시예에서는 제2 인터럽트를 처리할 프로세싱 유닛으로 제3 프로세싱 유닛(104)이 선정되었다. 이어서 프로세스 관리부(22)는 제1 작업 큐(Q1)에 할당된 제2 인터럽트를 선정된 제3 프로세싱 유닛(104)의 제3 작업 큐(Q3)로 이전할 수 있고, 제3 작업 큐(Q3)로 이전된 제2 인터럽트는 제3 프로세싱 유닛(104) 상에서 처리될 수 있다. 이에 따라 제1 프로세싱 유닛(100) 상에서 제1 작업 큐(Q1)에 기 할당된 인터럽트의 수가 많아 제2 인터럽트의 처리 대기 시간이 상당한 경우, 제2 인터럽트를 다른 프로세싱 유닛으로 이전하여 제2 인터럽트를 신속하게 처리할 수 있다.

[0074] 도 3을 계속 참조하면, 본 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 각각의 프로세싱 유닛(100, 102, 104, 106)의 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것을 포함할 수 있다.

[0075] 일례로, 각각의 프로세싱 유닛(100, 102, 104, 106)의 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 액티브(active) 상태인 프로세싱 유닛을 제2 인터럽트를 처리할 프로세싱 유닛으로 선정하는 것을 포함할 수 있다. 도 3에서, 제1 프로세싱 유닛(100), 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106)은 액티브 상태인 반면, 제2 프로세싱 유닛(102)은 슬립(sleep) 상태이므로, 프로세스 관리부(22)는 액티브 상태인 제1 프로세싱 유닛(100), 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106) 중 어느 하나를 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 도 3에서는 제2 인터럽트를 처리할 프로세싱 유닛으로서 제3 프로세싱 유닛(104)을 선정하여, 제2 인터럽트를 제3 작업 큐(Q3)로 이전하였다.

[0076] 다른 예로, 각각의 프로세싱 유닛(100, 102, 104, 106)의 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제1 프로세싱 유닛(100)보다 이용률(utilization)이 낮은 프로세싱 유닛을 제2 인터럽트를 처리할 프로세싱 유닛으로 선정하는 것을 포함할 수 있다. 도 3에서, 제2 프로세싱 유닛(102), 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106)은 모두 이용률(U)이 각각 0.49, 0.51, 0.32로서 제1 프로세싱 유닛(100)의 이용률(U)인 0.89보다 낮으므로, 프로세스 관리부(22)는 제2 프로세싱 유닛(102), 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106) 중 어느 하나를 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 이러한 경우, 도 3에 도시된 바와 달리, 제2 인터럽트를 처리할 프로세싱 유닛으로서 이용률(U)이 가장 낮은 제4 프로세싱 유닛(106)을 선정하여, 제2 인터럽트를 제4 작업 큐(Q3)로 이전할 수 있다.

[0077] 한편, 본 발명의 몇몇의 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제1 프로세싱 유닛(100)이 펜딩(pending) 상태인 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것을 포함할 수 있다. 다시 말해서 제1 프로세싱 유닛(100)이 가용인(available) 상태에서는 제2 인터럽트를 다른 프로세싱 유닛에 이전하지 않고, 제1 프로세싱 유닛(100)에서 처리할 수 있다.

[0078] 이와 같이, 컴퓨팅 시스템의 처리 능력 또는 하드웨어(10)의 상태를 고려하여 인터럽트를 비롯한 작업들을 최적의 자원(예컨대, 프로세싱 유닛)에 분배함으로써, 다량의 작업들을 효율적이고 신속하게 처리할 수 있다. 이하에서는 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 다양한 방법에 중점을 두어, 본 발명의 인터럽트 핸들링 방법의 여러 실시예들을 설명하기로 한다.

[0079] 도 4는 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.

[0080] 도 4를 참조하면, 도 3에서 설명된 바와 같이, 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법에서 제1 인터럽트 및 제2 인터럽트는 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 할당될 수 있다. 이후 프로세스 관리부(22)는 제1 작업 큐(Q1)에 할당된 제1 인터럽트가 처리되는 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다.

- [0081] 도 4를 계속 참조하면, 본 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 각각의 프로세싱 유닛(100, 102, 104, 106)의 작업 큐(Q1, Q2, Q3, Q4)의 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0082] 일례로, 각각의 프로세싱 유닛(100, 102, 104, 106)의 작업 큐(Q1, Q2, Q3, Q4)의 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제1 프로세싱 유닛(100)보다 작업 큐에 할당된 인터럽트의 개수가 적은 프로세싱 유닛을 제2 인터럽트를 처리할 프로세싱 유닛으로 선정하는 것을 포함할 수 있다. 도 4에서, 제2 프로세싱 유닛(102), 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106)은 모두 각각의 작업 큐(Q2, Q3, Q4)에 할당된 인터럽트의 개수가 각각 2, 2, 1로서 제1 프로세싱 유닛(100)의 작업 큐(Q1)에 할당된 인터럽트의 개수인 4보다 적으므로, 프로세스 관리부(22)는 제2 프로세싱 유닛(102), 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106) 중 어느 하나를 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 도 4에서는 제2 인터럽트를 처리할 프로세싱 유닛으로서 제4 프로세싱 유닛(106)을 선정하여, 제2 인터럽트를 제4 작업 큐(Q4)로 이전하였다.
- [0083] 도 5는 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.
- [0084] 도 5를 참조하면, 도 3에서 설명된 바와 같이, 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법에서 제1 인터럽트 및 제2 인터럽트는 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 할당될 수 있다. 이후 프로세스 관리부(22)는 제1 작업 큐(Q1)에 할당된 제1 인터럽트가 처리되는 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다.
- [0085] 도 5를 계속 참조하면, 본 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 각각의 프로세싱 유닛(100, 102, 104, 106)에 대한 인터럽트의 발생 빈도에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0086] 일례로, 각각의 프로세싱 유닛(100, 102, 104, 106)에 대한 인터럽트의 발생 빈도에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제1 프로세싱 유닛(100)보다 인터럽트의 발생 빈도가 낮은 프로세싱 유닛을 제2 인터럽트를 처리할 프로세싱 유닛으로 선정하는 것을 포함할 수 있다. 도 5에서, 제2 프로세싱 유닛(102)의 인터럽트 발생 빈도(F2) 및 제3 프로세싱 유닛(104)의 인터럽트 발생 빈도(F3)는 제1 프로세싱 유닛(100)의 인터럽트 발생 빈도(F1)보다 낮으므로, 프로세스 관리부(22)는 제2 프로세싱 유닛(102) 및 제3 프로세싱 유닛(104) 중 어느 하나를 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 도 5에서는 제2 인터럽트를 처리할 프로세싱 유닛으로서 제2 프로세싱 유닛(102)을 선정하여, 제2 인터럽트를 제2 작업 큐(Q2)로 이전하였다.
- [0087] 도 6은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.
- [0088] 도 6을 참조하면, 도 3에서 설명된 바와 같이, 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법에서 제1 인터럽트 및 제2 인터럽트는 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 할당될 수 있다. 이후 프로세스 관리부(22)는 제1 작업 큐(Q1)에 할당된 제1 인터럽트가 처리되는 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다.
- [0089] 도 6을 계속 참조하면, 본 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 각각의 프로세싱 유닛(100, 102, 104, 106)의 캐시 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0090] 일례로, 각각의 프로세싱 유닛(100, 102, 104, 106)의 작업 큐(Q1, Q2, Q3, Q4)의 상태에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제2 인터럽트를 처리할 때, 제1 프로세싱 유닛(100)보다 캐시 미스의 발생 빈도 수가 높지 않은 프로세싱 유닛을 제2 인터럽트를 처리할 프로세싱 유닛으로 선정하는 것을 포함할 수 있다. 도 6에서, 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106)은 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106)에서 제2 인터럽트를 처리할 때의 캐시 미스의 발생 빈도수(C)가 각각 0.27, 0.17로서 제1 프로세싱 유닛(100)에서 제2 인터럽트를 처리할 때의 캐시 미스의 발생 빈도수(C)인 0.3보다 낮으므로, 프로세스 관리부(22)는 제3 프로세싱 유닛(104) 및 제4 프로세싱 유닛(106) 중 어느 하나를 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 도 6에서는 제2 인터럽트를 처리할 프로세싱 유닛으로서 제3 프로세싱 유닛

(104)을 선정하여, 제2 인터럽트를 제3 작업 큐(Q3)로 이전하였다.

- [0091] 도 7은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.
- [0092] 도 7을 참조하면, 도 6에 도시된 실시예와 다른 점은 제4 프로세싱 유닛(106)이 슬립 상태에서 액티브 상태로 변경된 것이다. 도 6에 도시된 실시예에서는 제4 프로세싱 유닛(106)의 캐시 미스 발생 빈도수(C)가 0.17로 가장 낮았음에도 불구하고, 제4 프로세싱 유닛(106)이 슬립 상태에 있었기 때문에, 제4 프로세싱 유닛(106)이 제2 인터럽트를 처리할 프로세싱 유닛으로서 선정되지 않았다. 그러나 도 7에서는 제4 프로세싱 유닛(106)이 슬립 상태에서 액티브 상태로 변경되었기 때문에 제2 인터럽트를 처리하기에 더 적합하게 되었다.
- [0093] 이러한 경우, 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법에서, 프로세스 관리부(22)는 제3 프로세싱 유닛(104)의 제3 작업 큐(Q3)로 이전된 제2 인터럽트를 새롭게 선정된 제4 프로세싱 유닛(106)의 제4 작업 큐(Q4)로 이전할 수 있다. 이에 따라 제4 프로세싱 유닛(106)은 제4 작업 큐(Q4)로 이전된 제2 인터럽트를 처리할 수 있다.
- [0094] 한편, 본 발명의 몇몇의 실시예에서, 제3 프로세싱 유닛(104)의 제3 작업 큐(Q3)로 이전된 제2 인터럽트는 다시 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)로 다시 이전될 수도 있다. 예컨대, 제2 인터럽트가 제3 프로세싱 유닛(104)의 제3 작업 큐(Q3)로 이전된 동안 제1 프로세싱 유닛(100)의 상태가 변경되어 제2 인터럽트의 처리에 더 적합하도록 되었다면, 제3 작업 큐(Q3)로 이전된 제2 인터럽트를 다시 제1 작업 큐(Q1)로 이전할 수도 있다.
- [0095] 도 8은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 개략도이다.
- [0096] 도 8을 참조하면, 도 3에서 설명된 바와 같이, 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법에서 제1 인터럽트 및 제2 인터럽트는 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 할당될 수 있다. 이후 프로세스 관리부(22)는 제1 작업 큐(Q1)에 할당된 제1 인터럽트가 처리되는 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다.
- [0097] 도 8을 계속 참조하면, 본 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 각각의 프로세싱 유닛(100, 102, 104, 106)의 작업 큐(Q1, Q2, Q3, Q4)에서 제2 인터럽트의 처리 대기 시간에 기초하여 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것을 포함할 수 있다.
- [0098] 일례로, 프로세스 관리부(22)는 각각의 작업 큐(Q1, Q2, Q3, Q4)에서의 제2 인터럽트의 처리 대기 시간(WT)을 연산한 후, 제1 프로세싱 유닛(100)보다 제2 인터럽트의 처리 대기 시간이 적은 프로세싱 유닛을 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 도 8에서, 제2 프로세싱 유닛(102) 및 제4 프로세싱 유닛(106)은 모두 각각의 작업 큐(Q2, Q4)에서의 제2 인터럽트의 처리 대기 시간이 9.3으로서 제1 프로세싱 유닛(100)의 작업 큐(Q1)에서의 제2 인터럽트의 처리 대기 시간인 10보다 적으므로, 프로세스 관리부(22)는 제2 프로세싱 유닛(102) 및 제4 프로세싱 유닛(106) 중 어느 하나를 제2 인터럽트를 처리할 프로세싱 유닛으로 선정할 수 있다. 도 8에서는 제2 인터럽트를 처리할 프로세싱 유닛으로서 제2 프로세싱 유닛(102)을 선정하여, 제2 인터럽트를 제2 작업 큐(Q2)로 이전하였다.
- [0099] 도 9는 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 다중 프로세서를 포함하는 컴퓨팅 시스템을 설명하기 위한 개략도이다.
- [0100] 도 9를 참조하면, 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 다중 프로세서를 포함하는 컴퓨팅 시스템(2)은 제1 CPU(200), 제2 CPU(202), 제3 CPU(204) 및 제4 CPU(206)를 포함할 수 있고, 제1 CPU(200), 제2 CPU(202), 제3 CPU(204) 및 제4 CPU(206)는 버스(210)를 통해 서로 데이터를 주고 받을 수 있다. 이에 따라 제1 CPU(200), 제2 CPU(202), 제3 CPU(204) 및 제4 CPU(206)는 각각 고유의 작업 큐를 구비할 수 있다.
- [0101] 도 10은 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 다중 코어를 포함하는 컴퓨팅 시스템을 설명하기 위한 개략도이다.
- [0102] 도 10을 참조하면, 본 발명의 몇몇의 실시예에 따른 인터럽트 핸들링 방법이 수행되는 다중 코어를 포함하는 컴퓨팅 시스템(3)은 제1 코어(300), 제2 코어(302), 제3 코어(304) 및 제4 코어(306)를 포함할 수 있고, 제1 코어(300), 제2 코어(302), 제3 코어(304) 및 제4 코어(306)는 버스(310)를 통해 서로 데이터를 주고 받을 수 있다. 이에 따라 제1 코어(300), 제2 코어(302), 제3 코어(304) 및 제4 코어(306)는 각각 고유의 작업 큐를 구비

할 수 있다.

- [0103] 도 11은 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.
- [0104] 도 11을 참조하면, 본 발명의 일 실시예에 따른 인터럽트 핸들링 방법은 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 제1 인터럽트를 할당(S1101)하고, 제1 작업 큐(Q1)에 제2 인터럽트를 할당(S1103)하는 것을 포함할 수 있다. 상기 방법은, 제1 프로세싱 유닛(100) 상에서 제1 작업 큐(Q1)에 할당된 제1 인터럽트를 처리(S1105)하고, 제1 인터럽트를 처리하는 동안, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제2 인터럽트를 처리할 프로세싱 유닛을 선정(S1107)하고, 제1 작업 큐(Q1)에 할당된 제2 인터럽트를 선정된 프로세싱 유닛의 작업 큐로 이전(S1109)하는 것을 포함할 수 있다.
- [0105] 도 12는 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.
- [0106] 도 12를 참조하면, 본 발명의 다른 실시예에 따른 인터럽트 핸들링 방법은, 복수의 프로세싱 유닛(100, 102, 104, 106)에 대해 복수의 인터럽트를 할당(S1201)하고, 복수의 인터럽트의 개수가 복수의 프로세싱 유닛(100, 102, 104, 106)의 개수보다 많은지 검사(S1203)하는 것을 포함할 수 있다. 복수의 인터럽트의 개수가 복수의 프로세싱 유닛(100, 102, 104, 106)의 개수보다 많은 경우, 제1 인터럽트 및 제2 인터럽트를 포함하는 둘 이상의 인터럽트가 할당된 제1 프로세싱 유닛(100)에 대해, 제1 인터럽트는 제1 프로세싱 유닛(100)을 이용하여 처리(S1205)하고, 복수의 프로세싱 유닛(100, 102, 104, 106) 중에서 제2 인터럽트를 처리할 프로세싱 유닛을 선정(S1207)하고, 제2 인터럽트를 선정된 프로세싱 유닛을 이용하여 처리(S1209)하는 것을 포함할 수 있다.
- [0107] 본 발명의 몇몇의 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중에서 제2 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제1 인터럽트를 제1 프로세싱 유닛(100)을 이용하여 처리하는 동안 수행될 수 있다. 한편, 도 3 내지 도 8과 관련하여 앞서 설명한 바와 같이, 본 발명의 몇몇의 실시예에서, 복수의 프로세싱 유닛(100, 102, 104, 106) 중에서 제2 인터럽트를 처리할 프로세싱 유닛은, 프로세싱 유닛의 이용률, 작업 큐에 할당된 인터럽트의 개수, 인터럽트의 발생 빈도, 캐시 미스의 발생 빈도 수 등을 고려하여 선정될 수 있다.
- [0108] 본 발명의 몇몇의 실시예에서, 제2 인터럽트는, 제1 인터럽트를 제1 프로세싱 유닛(100)을 이용하여 처리하는 동안, 선정된 프로세싱 유닛의 작업 큐로 이전될 수도 있다.
- [0109] 도 13은 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.
- [0110] 도 13을 참조하면, 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 삽입할 제1 인터럽트를 수신(S1301)하고, 제1 작업 큐(Q1)의 상태를 모니터링(S1303)하고, 제1 작업 큐(Q1)에 기 삽입된 인터럽트의 개수가 제1 임계값을 초과하는 경우, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제1 인터럽트를 처리할 프로세싱 유닛을 선정(S1305)하고, 제1 인터럽트를 선정된 프로세싱 유닛의 작업 큐에 삽입(S1307)하는 것을 포함할 수 있다. 이 경우, 제1 인터럽트를 처리할 프로세싱 유닛은, 작업 큐에 기 삽입된 인터럽트의 개수가 제2 임계값 이하인 프로세싱 유닛 중에서 선정될 수 있다. 한편, 본 발명의 몇몇의 실시예에서 제1 임계값 및 제2 임계값은 서로 동일한 값일 수 있다.
- [0111] 또한, 본 발명의 몇몇의 실시예에서, 제1 인터럽트를 처리할 프로세싱 유닛을 선정하는 것은, 제1 작업 큐(Q1)에 기 삽입된 인터럽트를 제1 프로세싱 유닛(100)을 이용하여 처리하는 동안 수행될 수 있다.
- [0112] 한편, 본 발명의 몇몇의 실시예에서, 상기 방법은, 제1 프로세싱 유닛의 상태(100)를 모니터링하고, 제1 프로세싱 유닛(100)이 비액티브(inactive) 상태인 경우, 제1 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다. 이 경우, 제1 인터럽트를 처리할 프로세싱 유닛은, 액티브 상태인 프로세싱 유닛 중에서 선정될 수 있다.
- [0113] 한편, 본 발명의 몇몇의 실시예에서, 상기 방법은, 제1 프로세싱 유닛(100)의 이용률을 모니터링하고, 제1 프로세싱 유닛(100)의 이용률이 제3 임계값을 초과하는 경우, 제1 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다. 이 경우, 제1 인터럽트를 처리할 프로세싱 유닛은, 이용률이 제4 임계값 이하인 프로세싱 유닛 중에서 선정될 수 있다.
- [0114] 한편, 본 발명의 몇몇의 실시예에서, 상기 방법은, 제1 프로세싱 유닛(100)에 지정되어 수신된 인터럽트의 발생 빈도를 모니터링하고, 제1 프로세싱 유닛(100)에 지정되어 수신된 인터럽트의 발생 빈도가 제5 임계값을 초과하는 경우, 제1 인터럽트를 처리할 프로세싱 유닛을 선정할 수 있다. 이 경우, 제1 인터럽트를 처리할 프로세싱

유닛은, 지정되어 수신된 인터럽트의 발생 빈도가 제6 임계값 이하인 프로세싱 유닛 중에서 선정될 수 있다.

- [0115] 도 14는 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법을 설명하기 위한 순서도이다.
- [0116] 도 14를 참조하면, 본 발명의 또 다른 실시예에 따른 인터럽트 핸들링 방법은, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 제1 프로세싱 유닛(100)에 지정된 제1 인터럽트를 수신하여 제1 프로세싱 유닛(100)의 제1 작업 큐(Q1)에 삽입(S1401)하고, 제1 프로세싱 유닛(100)에 지정된 제2 인터럽트를 수신(S1403)하고, 제1 작업 큐(Q1)에서의 제2 인터럽트의 처리 대기 시간을 연산(S1405)하고, 복수의 프로세싱 유닛(100, 102, 104, 106) 중 다른 프로세싱 유닛의 각각의 작업 큐에서의 제2 인터럽트의 처리 대기 시간을 연산(S1407)하고, 다른 프로세싱 유닛의 작업 큐에서의 제2 인터럽트의 처리 대기 시간이 제1 작업 큐(Q1)에서의 제2 인터럽트의 처리 대기 시간 보다 짧은 경우, 제2 인터럽트를 해당 작업 큐에 삽입(S1409)하는 것을 포함할 수 있다.
- [0117] 본 발명의 몇몇의 실시예에서, 제2 인터럽트의 처리 대기 시간을 연산하는 것은, 제1 프로세싱 유닛(100)을 이용하여 상기 제1 인터럽트를 처리하는 동안 수행될 수 있다. 또한, 본 발명의 몇몇의 실시예에서, 제2 인터럽트의 처리 대기 시간을 연산하는 것은, 복수의 프로세싱 유닛(100, 102, 104, 106) 각각의 작업 큐(Q1, Q2, Q3, Q4)에 기 삽입된 인터럽트의 개수, 각각의 프로세싱 유닛(100, 102, 104, 106)의 상태 또는 각각의 프로세싱 유닛(100, 102, 104, 106)에 대한 인터럽트의 발생 빈도에 기초하여 제2 인터럽트의 처리 대기 시간을 연산하는 것을 포함할 수 있다.
- [0118] 상술한 본 발명의 다양한 실시예에 따르면, 컴퓨팅 시스템의 처리 능력 또는 하드웨어(10)의 상태를 고려하여 인터럽트를 비롯한 작업들을 최적의 자원(예컨대, 프로세싱 유닛)에 분배함으로써, 다량의 작업들을 효율적이고 신속하게 처리할 수 있다.
- [0119] 도 15 내지 도 17은 본 발명의 실시예들에 따른 반도체 장치를 적용할 수 있는 예시적인 반도체 시스템들이다.
- [0120] 도 15는 태블릿 PC(1200)을 도시한 도면이고, 도 16은 노트북(1300)을 도시한 도면이며, 도 17은 스마트폰(1400)을 도시한 것이다. 본 발명의 실시예들에 따른 반도체 장치 및 반도체 장치의 동작 방법은 이러한 태블릿 PC(1200), 노트북(1300), 스마트폰(1400) 등에 사용될 수 있다.
- [0121] 또한, 본 발명의 몇몇 실시예들에 따른 반도체 장치 및 반도체 장치의 동작 방법은 예시하지 않은 다른 집적 회로 장치에도 적용될 수 있음은 통상의 기술자에게 자명하다. 즉, 이상에서는 본 실시예에 따른 반도체 장치 및 반도체 장치의 동작 방법의 예로, 태블릿 PC(1200), 노트북(1300), 및 스마트폰(1400)만을 들었으나, 이에 제한되는 것은 아니다. 본 발명의 몇몇 실시예에서, 반도체 시스템은, 컴퓨터, UMPC (Ultra Mobile PC), 워크스테이션, 넷북(net-book), PDA (Personal Digital Assistants), 포터블(portable) 컴퓨터, 무선 전화기(wireless phone), 모바일 폰(mobile phone), e-북(e-book), PMP(portable multimedia player), 휴대용 게임기, 네비게이션(navigation) 장치, 블랙박스(black box), 디지털 카메라(digital camera), 3차원 수상기(3-dimensional television), 디지털 음성 녹음기(digital audio recorder), 디지털 음성 재생기(digital audio player), 디지털 영상 녹화기(digital picture recorder), 디지털 영상 재생기(digital picture player), 디지털 동영상 녹화기(digital video recorder), 디지털 동영상 재생기(digital video player) 등으로 구현될 수도 있다.
- [0122] 이상 첨부된 도면을 참조하여 본 발명의 실시예들을 설명하였으나, 본 발명은 상기 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 제조될 수 있으며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

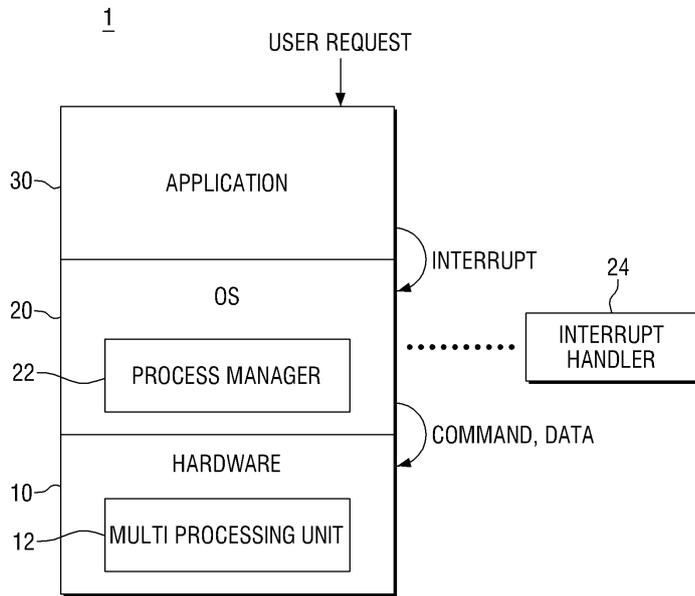
부호의 설명

- [0123] 1: 컴퓨팅 시스템 2: 다중 프로세서 시스템
- 3: 다중 코어 시스템 10: 하드웨어
- 12: 다중 프로세싱 유닛 20: 운영체제
- 22: 프로세서 관리부 24: 인터럽트 핸들러
- 30: 어플리케이션 100, 102, 104, 106: 프로세싱 유닛
- 110, 210, 310: 버스 Q1, Q2, Q3, Q4: 작업 큐

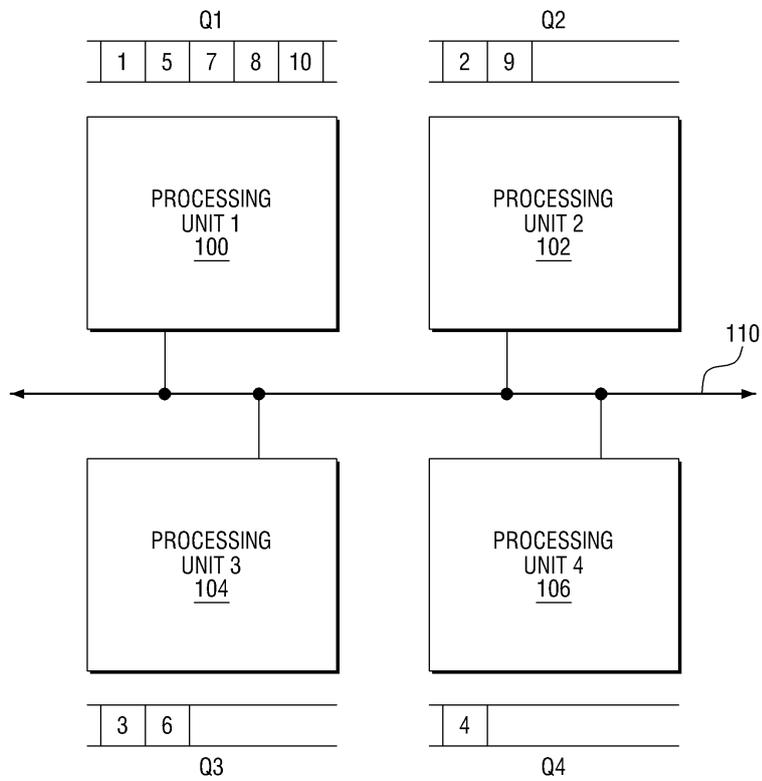
200, 202, 204, 206: CPU 300, 302, 304, 306: 코어

도면

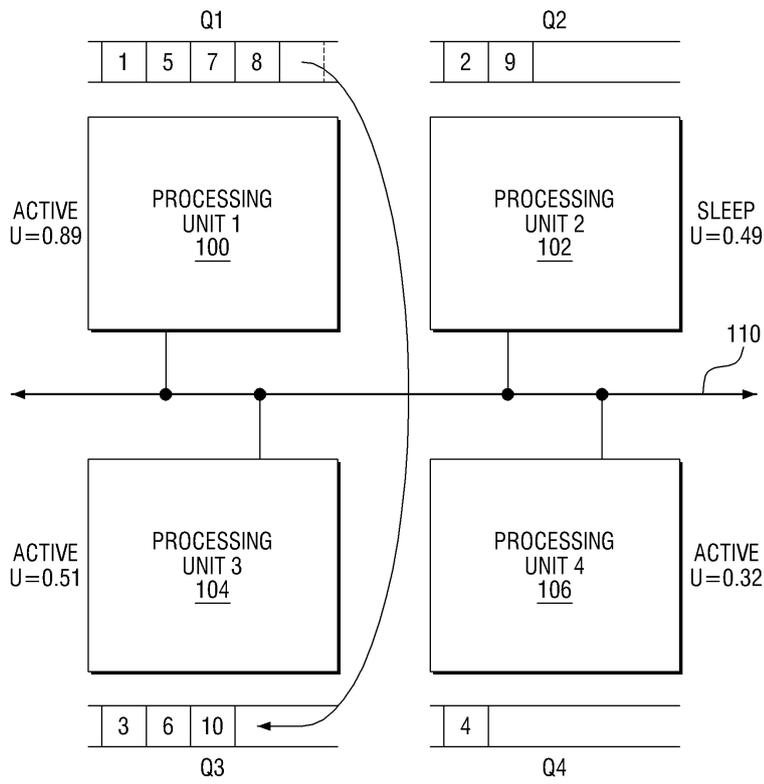
도면1



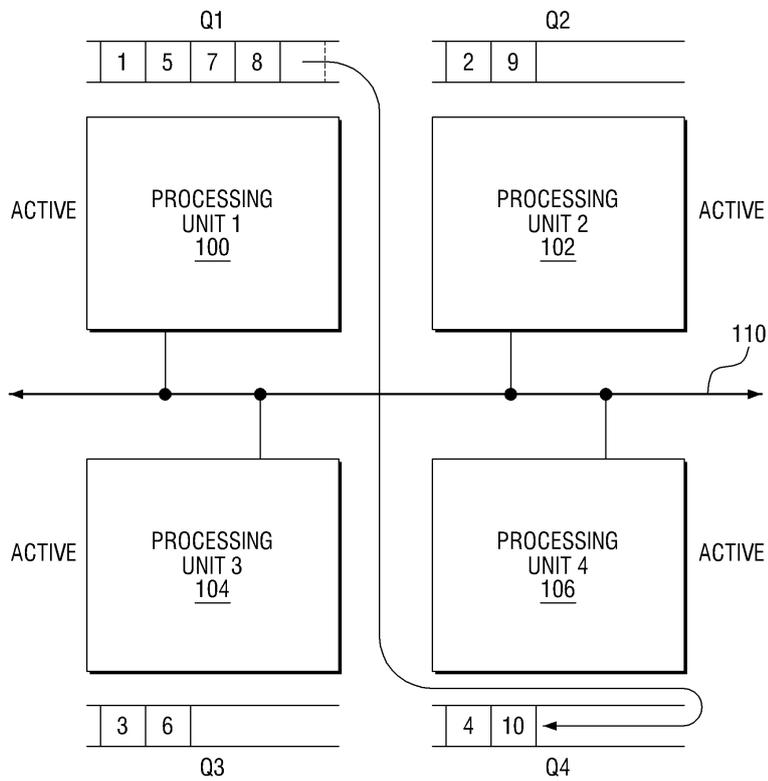
도면2



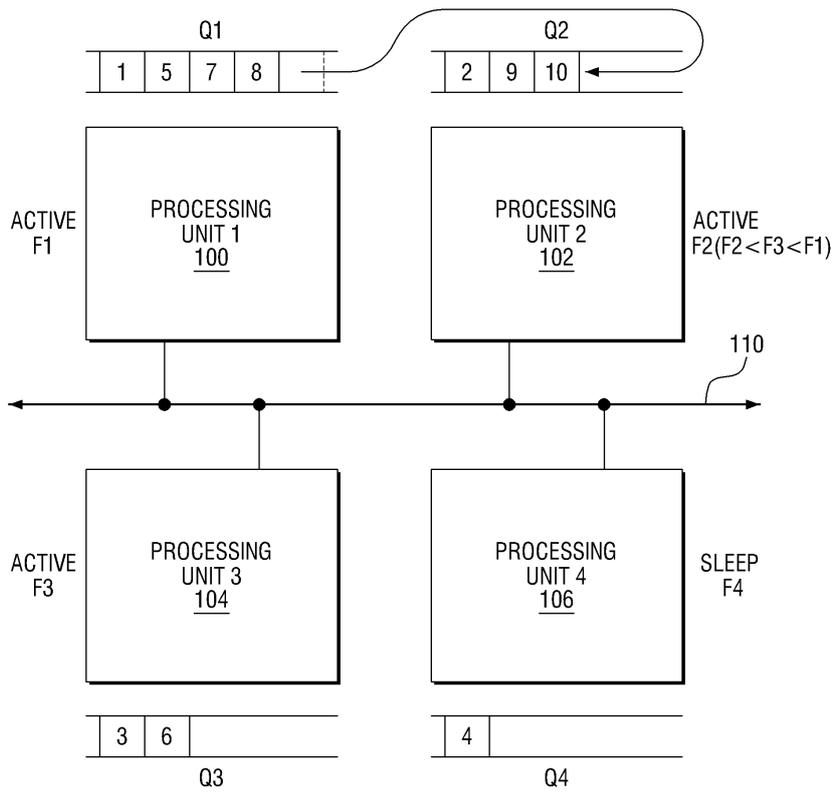
도면3



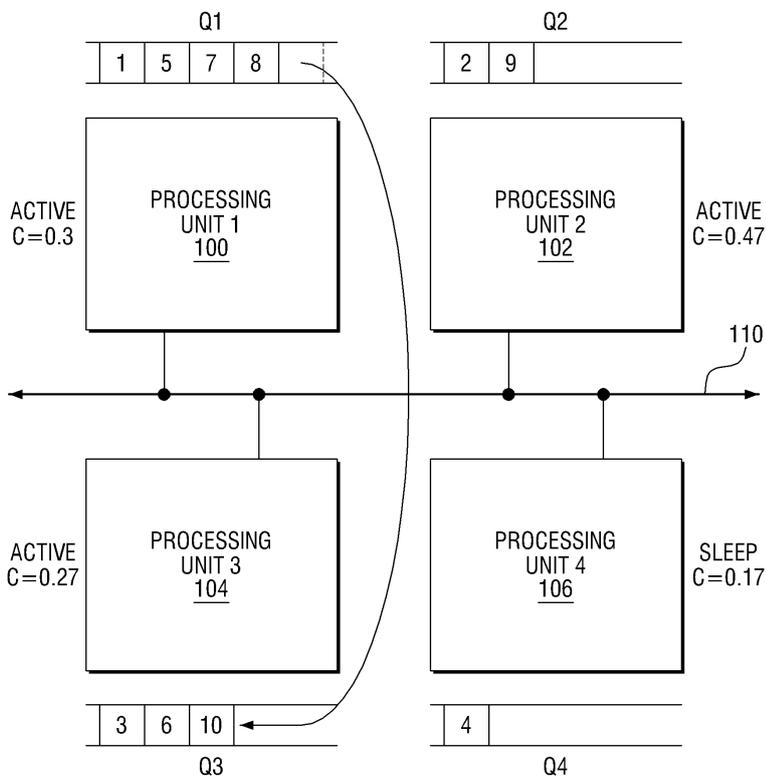
도면4



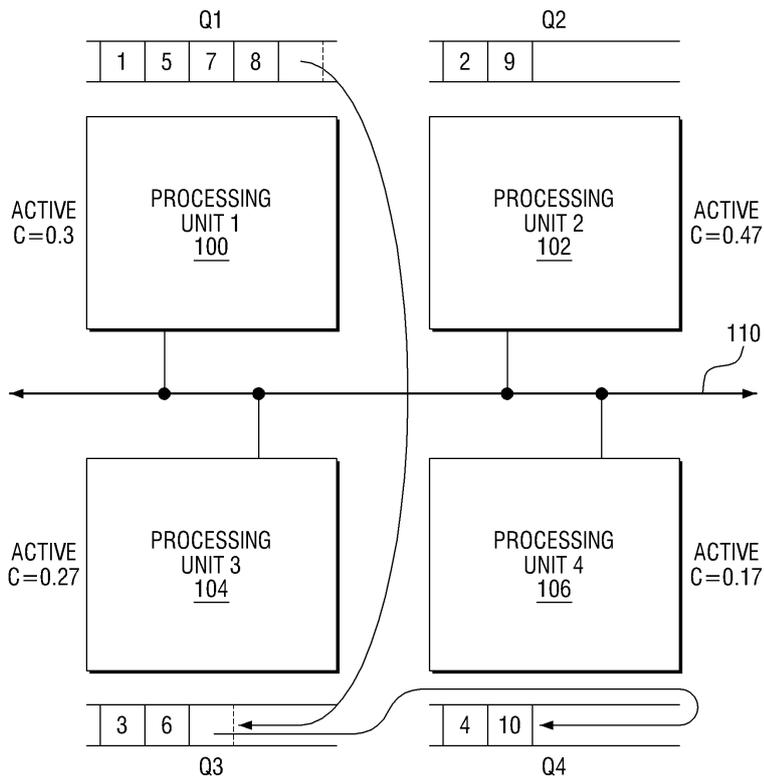
도면5



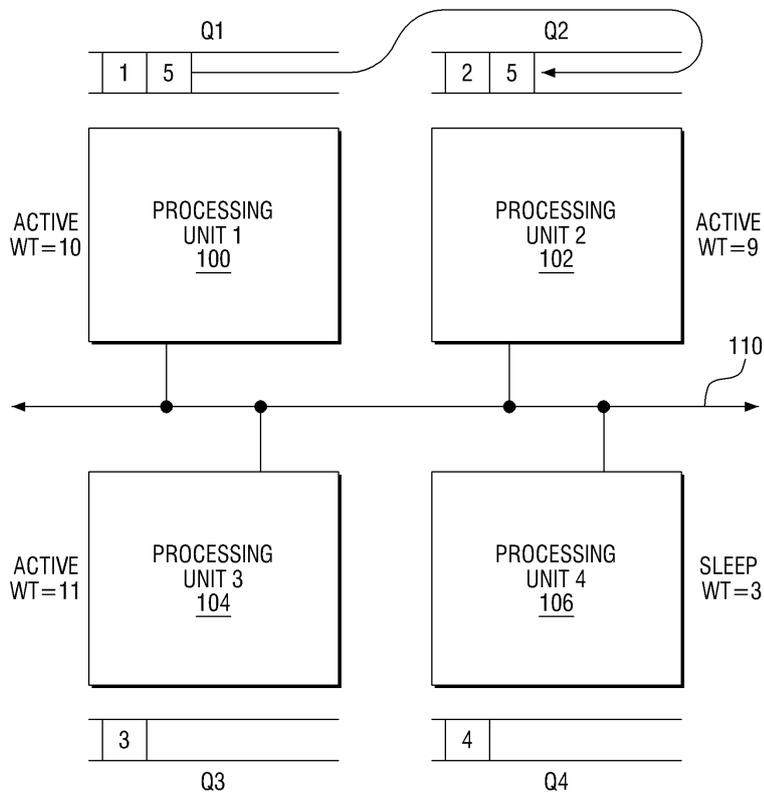
도면6



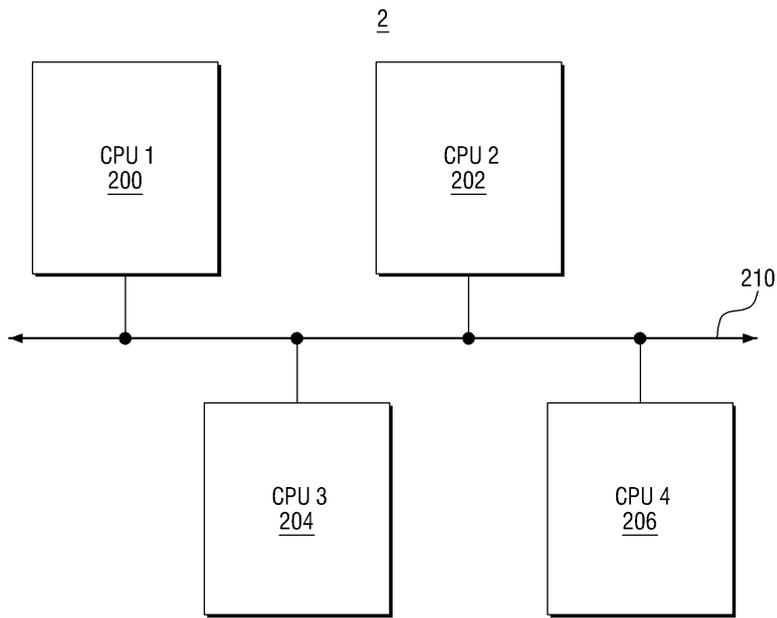
도면7



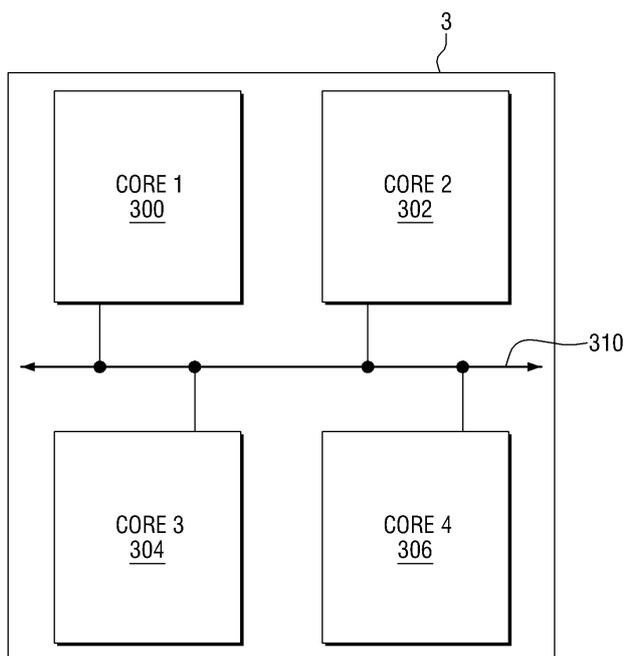
도면8



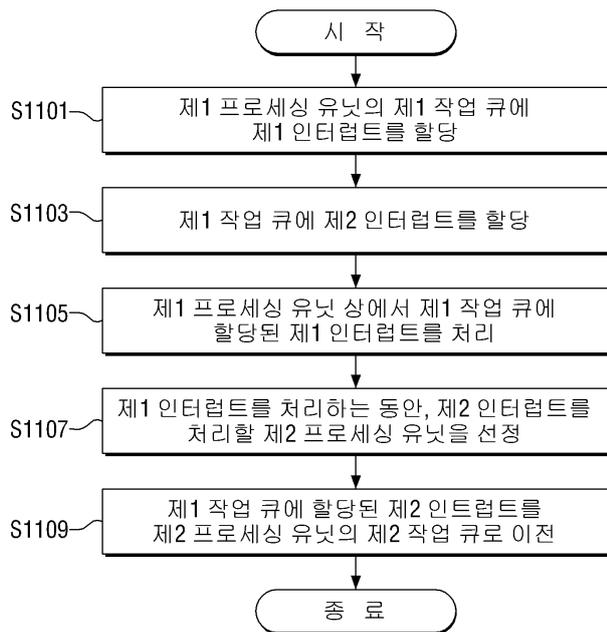
도면9



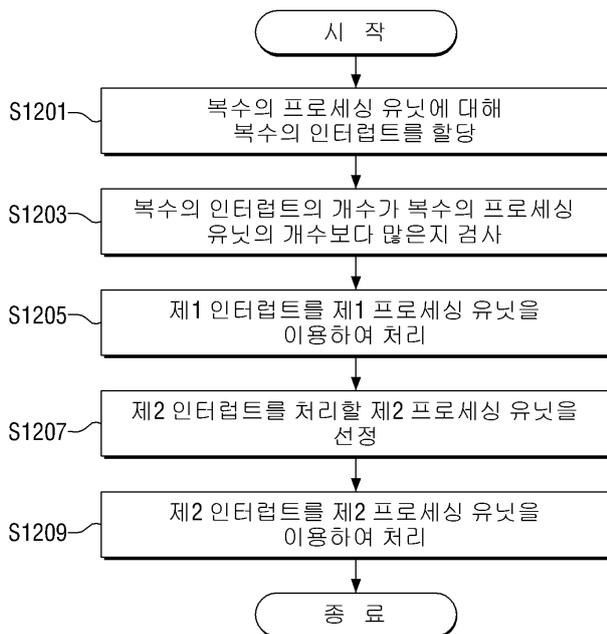
도면10



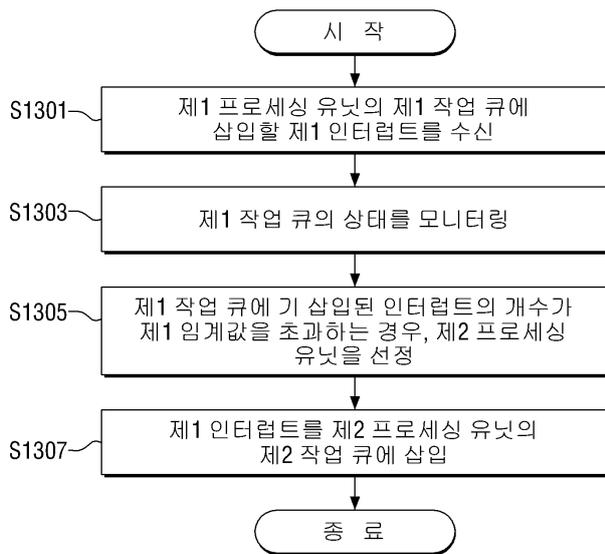
도면11



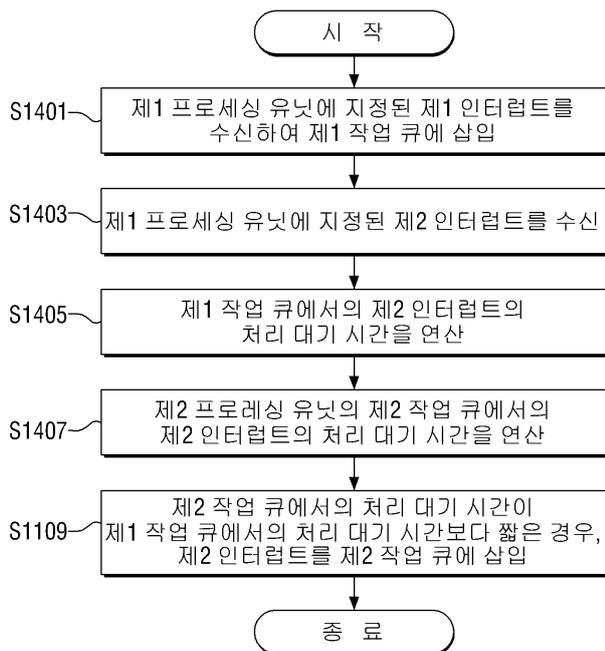
도면12



도면13

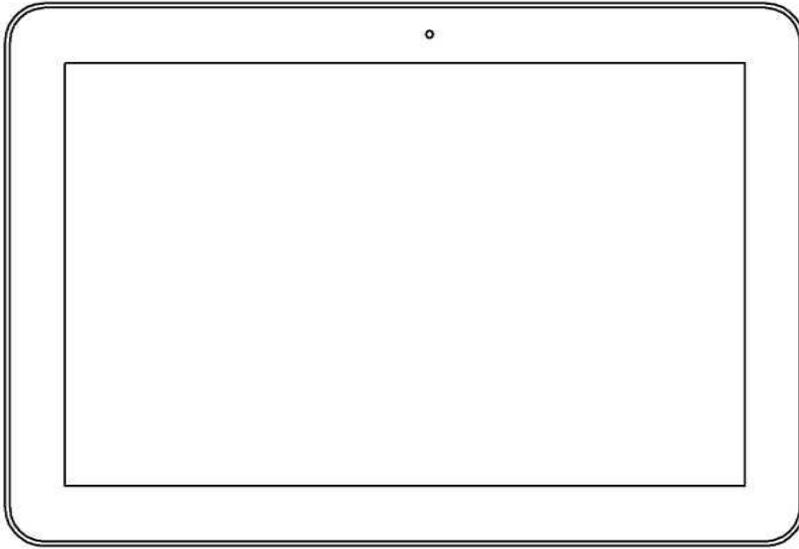


도면14



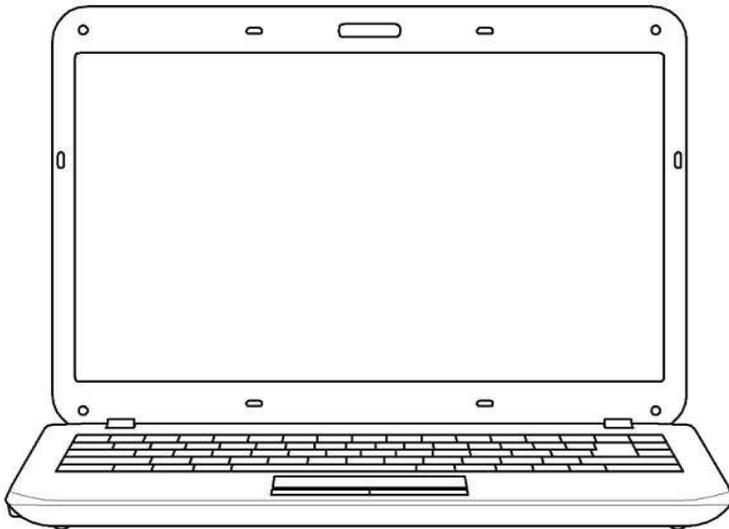
도면15

1200



도면16

1300



도면17

1400

