(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년03월26일
(11) 등록번호 10-1841930
(24) 등록일자 2018년03월20일

(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2018.01)
(21) 출원번호 10-2012-0008822
(22) 출원일자 2012년01월30일
심사청구일자 2016년10월31일
(65) 공개번호 10-2013-0087735
(43) 공개일자 2013년08월07일
(56) 선행기술조사문헌
US20030105798 A1*
(뒷면에 계속)

(73) 특허권자
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
이재곤
경기 용인시 수지구 수지로 166, 109동 1801호 (풍덕천동, 정자뜰마을태영데시앙1차아파트)
김동근
서울 서초구 효령로74길 57, 101동 902호 (서초동, 동원베네스트아파트)
(뒷면에 계속)
(74) 대리인
박영우

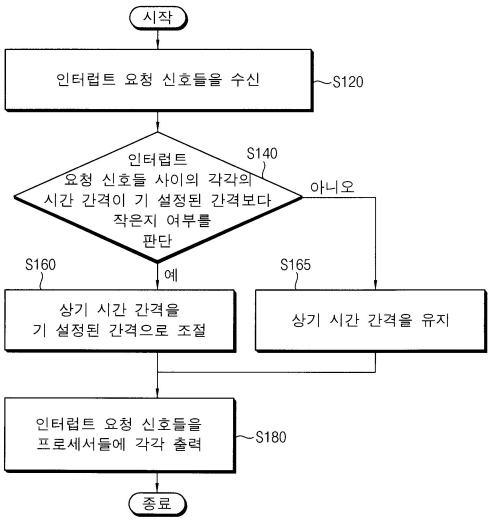
전체 청구항 수 : 총 14 항 심사관 : 유진태

(54) 발명의 명칭 **인터럽트 스프레드 방법, 인터럽트 스프레드 장치 및 이를 구비하는 시스템 온-칩**

(57) 요 약

인터럽트 스프레드 방법은 복수의 인터럽트 요청 신호들을 수신하고, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단하며, 상기 시간 간격이 기 설정된 간격보다 작은 경우 상기 시간 간격을 기 설정된 간격으로 조절하고, 인터럽트 요청 신호들을 복수의 프로세서들에 각각 출력할 수 있다. 이에, 인터럽트 스프레드 방법은 짧은 시간 내에 연속적으로 생성되는 인터럽트들에 의하여 프로세서들이 급작스러운 웨이크-업을 하는 것을 방지할 수 있다.

대 표 도 - 도1



(72) 발명자

김시영

서울 강남구 광평로10길 15, 211동 202호 (일원동, 상록수아파트)

허정훈

경기 수원시 영통구 효원로 363, 131동 1703호 (매탄동, 매탄위브하늘채아파트)

(56) 선행기술조사문헌

US5765003 A*

US5708814 A*

KR1020090043211 A

JP2005100237 A

JP11149459 A

US05765003 A*

US05708814 A*

*는 심사관에 의하여 인용된 문헌

명 세 서

청구범위

청구항 1

인터럽트 스프레드 장치가 복수의 인터럽트 요청 신호들을 수신하는 단계;

상기 인터럽트 스프레드 장치가 상기 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단하는 단계;

상기 인터럽트 스프레드 장치가 상기 시간 간격이 상기 기 설정된 간격보다 작은 경우, 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 단계; 및

상기 인터럽트 스프레드 장치가 상기 인터럽트 요청 신호들을 복수의 프로세서들에 각각 출력하는 단계를 포함하고,

상기 인터럽트 요청 신호들은 복수의 인터럽트 소스(source)들로부터 출력되는 복수의 인터럽트들에 기초하여 생성되고, 상기 인터럽트 요청 신호들은 상기 프로세서들에 각각 할당되며,

상기 기 설정된 간격은 상기 프로세서들이 인액티브(inactive) 상태에서 액티브(active) 상태로 변경될 때, 상기 프로세서들 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정되는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

제 1 항에 있어서, 상기 기 설정된 간격으로 조절하는 단계는

인접하는 제 k(단, k는 1이상인 정수) 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 0보다 큰 경우, 상기 제 k 시간 간격이 상기 기 설정된 간격으로 될 때까지 상기 제 k+1 인터럽트 요청 신호의 출력을 지연시키는 단계를 포함하는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 5

제 4 항에 있어서, 상기 기 설정된 간격으로 조절하는 단계는

기 설정된 우선순위에 따라 상기 인터럽트 요청 신호들의 출력 순서를 변경하는 단계를 더 포함하는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 6

제 5 항에 있어서, 상기 기 설정된 간격으로 조절하는 단계는

상기 제 k 시간 간격이 0인 경우, 상기 기 설정된 우선순위에 따라 상기 제 k 인터럽트 요청 신호와 상기 제 k+1 인터럽트 요청 신호 중에서 하나의 출력을 상기 기 설정된 간격만큼 지연시키는 단계를 더 포함하는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 7

인터럽트 스프레드 장치가 복수의 인터럽트 요청 신호들을 수신하는 단계;

상기 인터럽트 스프레드 장치가 상기 인터럽트 요청 신호들이 각각 출력될 복수의 프로세서들을 액티브(active)

상태인 프로세서들과 인액티브(inactive) 상태인 프로세서들로 분류하는 단계;

상기 인터럽트 스프레드 장치가 상기 인터럽트 요청 신호들 중에서 상기 액티브 상태인 프로세서들에 출력될 비대상(non-target) 인터럽트 요청 신호들을 상기 액티브 상태인 프로세서들에 각각 출력하는 단계;

상기 인터럽트 스프레드 장치가 상기 인터럽트 요청 신호들 중에서 상기 인액티브 상태인 프로세서들에 출력될 대상(target) 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단하는 단계;

상기 인터럽트 스프레드 장치가 상기 시간 간격이 상기 기 설정된 간격보다 작은 경우, 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 단계; 및

상기 인터럽트 스프레드 장치가 상기 대상 인터럽트 요청 신호들을 상기 인액티브 상태인 프로세서들에 각각 출력하는 단계를 포함하고,

상기 기 설정된 간격으로 조절하는 단계는

인접하는 제 k(단, k는 1이상인 정수) 대상 인터럽트 요청 신호와 제 k+1 대상 인터럽트 요청 신호 사이의 제 k 시간 간격이 0보다 큰 경우, 상기 제 k 시간 간격이 상기 기 설정된 간격으로 될 때까지 상기 제 k+1 대상 인터럽트 요청 신호의 출력을 지연시키는 단계를 포함하는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 8

삭제

청구항 9

제 7 항에 있어서, 상기 기 설정된 간격으로 조절하는 단계는

기 설정된 우선순위에 따라 상기 대상 인터럽트 요청 신호들의 출력 순서를 변경하는 단계를 더 포함하는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 10

제 9 항에 있어서, 상기 기 설정된 간격으로 조절하는 단계는

상기 제 k 시간 간격이 0인 경우, 상기 기 설정된 우선순위에 따라 상기 제 k 대상 인터럽트 요청 신호와 상기 제 k+1 대상 인터럽트 요청 신호 중에서 하나의 출력을 상기 기 설정된 간격만큼 지연시키는 단계를 더 포함하는 것을 특징으로 하는 인터럽트 스프레드 방법.

청구항 11

제 1 내지 제 m(단, m은 2이상의 정수) 인터럽트 요청 신호들을 수신하고, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 적어도 기 설정된 간격 이상을 두어 제 1 내지 제 m 프로세서들에 각각 출력하는 제 1 내지 제 m 인터럽트 홀더들; 및

상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격이 상기 기 설정된 간격보다 작은 경우, 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 인터럽트 아비터를 포함하고,

상기 제 1 내지 제 m 인터럽트 요청 신호들은 복수의 인터럽트 소스들로부터 출력되는 복수의 인터럽트들에 기초하여 생성되고, 상기 제 1 내지 제 m 인터럽트 요청 신호들은 상기 제 1 내지 제 m 인터럽트 홀더들에 각각 수신되며,

상기 제 1 내지 제 m 인터럽트 홀더들은 상기 제 1 내지 제 m 프로세서들에 각각 연결되고, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 적어도 상기 기 설정된 간격 이상을 두어 상기 제 1 내지 제 m 프로세서들에 각각 출력하고,

상기 기 설정된 간격은 상기 제 1 내지 제 m 프로세서들이 인액티브 상태에서 액티브 상태로 변경될 때, 상기 제 1 내지 제 m 프로세서들 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정되는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

제 11 항에 있어서, 상기 제 1 내지 제 m 인터럽트 홀더들은 상기 제 1 내지 제 m 인터럽트 요청 신호들을 각각 수신하면, 상기 인터럽트 아비터에 출력 요청 신호를 송신하고, 상기 인터럽트 아비터로부터 출력 허가 신호를 수신하면, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 상기 제 1 내지 제 m 프로세서들에 각각 출력하는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 16

제 15 항에 있어서, 상기 제 1 내지 제 m 인터럽트 홀더들 각각은 상기 제 1 내지 제 m 인터럽트 요청 신호들이 수신되지 않았음을 나타내는 아이들 스테이트(idle state), 상기 인터럽트 아비터로부터 상기 출력 허가 신호를 기다리고 있음을 나타내는 웨이트 스테이트(wait state) 및 상기 제 1 내지 제 m 인터럽트 요청 신호들이 상기 제 1 내지 제 m 프로세서들에 출력되고 있음을 나타내는 어서트 스테이트(assert state)로 구성된 스테이트 머신(state machine)으로 구현되는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 17

제 15 항에 있어서, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 인터럽트 홀더들로부터 각각 상기 출력 요청 신호를 수신하면, 상기 제 1 내지 제 m 인터럽트 홀더들에 상기 출력 허가 신호를 순차적으로 출력하여, 상기 제 1 내지 제 m 인터럽트 요청 신호들이 적어도 상기 기 설정된 간격 이상으로 상기 제 1 내지 제 m 프로세서들에 각각 출력되도록 제어하는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 18

제 17 항에 있어서, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 어느 하나도 상기 제 1 내지 제 m 프로세서들에 출력되고 있지 않음을 나타내는 아이들 스테이트, 및 상기 제 1 내지 제 n 인터럽트 요청 신호들 중에서 적어도 하나 이상이 상기 제 1 내지 제 m 프로세서들에 출력되고 있음을 나타내는 웨이트 스테이트로 구성된 스테이트 머신으로 구현되는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 19

제 11 항에 있어서, 상기 인터럽트 아비터는 기 설정된 우선순위에 따라 상기 제 1 내지 제 m 인터럽트 요청 신호들의 출력 순서를 변경하는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 20

제 11 항에 있어서, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 프로세서들을 액티브 상태인 프로세서들로 구성된 제 1 그룹과 인액티브 상태인 프로세서들로 구성된 제 2 그룹으로 분류하고, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 제 1 그룹에 할당된 인터럽트 요청 신호들에 대해서는 상기 기 설정된 간격으로 조절하지 않고, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 제 2 그룹에 할당된 인터럽트 요청 신호들에 대해서만 상기 기 설정된 간격으로 조절하는 것을 특징으로 하는 인터럽트 스프레드 장치.

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 멀티 프로세서 시스템(또는, 멀티 코어 시스템)에 관한 것으로서, 더욱 상세하게는 복수의 인터럽트 소스들에서 생성되는 복수의 인터럽트들에 대한 인터럽트 스프레드 방법, 인터럽트 스프레드 장치 및 이를 구비하는 시스템 온-칩에 관한 것이다.

배 경 기 술

[0002] 일반적으로, 전자 기기는 인터럽트 처리를 위한 인터럽트 컨트롤러를 구비하고, 인터럽트 소스들이 복수의 인터럽트들을 생성하면, 인터럽트들에 우선순위를 부여하여 프로세서에 출력할 수 있다. 최근에는 전자 기기의 소형화 및 경량화 추세에 따라 시스템 온-칩(System On-Chip; SOC)이 널리 사용되고, 시스템 온-칩에는 모듈 형태의 복수의 아이피(Intellectual Property; IP)들 및 복수의 프로세서들(또는, 복수의 코어(core)들을 가진 프로세서)이 실장될 수 있다.

[0003] 한편, 시스템 온-칩에 실장되는 복수의 프로세서들은 전력 소모를 줄이기 위하여 저전력 모드로 빈번하게 진입을 하게 되고, 인터럽트 소스들로부터 인터럽트들이 생성되어 인터럽트 요청 신호가 입력되면 저전력 모드에서 각각 탈출할 수 있다. 이 때, 시스템 온-칩에 실장되는 복수의 프로세서들이 저전력 모드에서 동시에 탈출하게 되는 경우, 급작스러운 웨이크-업(wake-up)으로 인하여 상기 프로세서들 내부에 인러쉬 전류(in-rush current)가 생성될 수 있다.

발명의 내용

해결하려는 과제

[0004] 본 발명의 일 목적은 복수의 인터럽트 소스들에서 짧은 시간 내에 연속적으로 생성되는 복수의 인터럽트들에 의하여 인액티브 상태(예를 들어, 파워 다운 상태, 파워 오프 상태 등)에 있는 복수의 프로세서들(또는, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태(예를 들어, 파워 온 상태 등)로 연속적으로 변경되는 것(즉, 급작스러운 웨이크-업)을 방지할 수 있는 인터럽트 스프레드 방법을 제공하는 것이다.

[0005] 본 발명의 다른 목적은 복수의 인터럽트 소스들에서 짧은 시간 내에 연속적으로 생성되는 복수의 인터럽트들에 의하여 인액티브 상태(예를 들어, 파워 다운 상태, 파워 오프 상태 등)에 있는 복수의 프로세서들(또는, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태(예를 들어, 파워 온 상태 등)로 연속적으로 변경되는 것(즉, 급작스러운 웨이크-업)을 방지할 수 있는 인터럽트 스프레드 장치를 제공하는 것이다.

[0006] 본 발명의 또 다른 목적은 상기 인터럽트 스프레드 장치를 구비하는 시스템 온-칩을 제공하는 것이다.

[0007] 다만, 본 발명의 해결하고자 하는 과제는 상기 언급된 과제에 한정되는 것이 아니며, 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위에서 다양하게 확장될 수 있을 것이다.

과제의 해결 수단

[0008] 본 발명의 일 목적을 달성하기 위하여, 본 발명의 실시예들에 따른 인터럽트 스프레드 방법은 복수의 인터럽트 요청 신호들을 수신하는 단계, 상기 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단하는 단계, 상기 시간 간격이 상기 기 설정된 간격보다 작은 경우 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 단계, 및 상기 인터럽트 요청 신호들을 복수의 프로세서들에 각각 출력하는 단계를 포함할 수 있다.

[0009] 일 실시예에 의하면, 상기 인터럽트 요청 신호들은 복수의 인터럽트 소스(source)들로부터 출력되는 복수의 인터럽트들에 기초하여 생성되고, 상기 인터럽트 요청 신호들 각각은 상기 프로세서들에 각각 할당될 수 있다.

[0010] 일 실시예에 의하면, 상기 기 설정된 간격은 상기 프로세서들이 인액티브(inactive) 상태에서 액티브(active) 상태로 변경될 때, 상기 프로세서들 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정될 수 있다.

[0011] 일 실시예에 의하면, 상기 기 설정된 간격으로 조절하는 단계는 제 k(단, k는 1이상인 정수) 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 0보다 큰 경우, 상기 제 k 시간 간격이 상기 기 설정된 간격으로 될 때까지 상기 제 k+1 인터럽트 요청 신호의 출력을 지연시키는 단계를 포함할 수 있다.

[0012] 일 실시예에 의하면, 상기 기 설정된 간격으로 조절하는 단계는 기 설정된 우선순위에 따라 상기 인터럽트 요청 신호들의 출력 순서를 변경하는 단계를 더 포함할 수 있다.

[0013] 일 실시예에 의하면, 상기 기 설정된 간격으로 조절하는 단계는 상기 제 k 시간 간격이 0인 경우, 상기 기 설정된 우선순위에 따라 상기 제 k 인터럽트 요청 신호와 상기 제 k+1 인터럽트 요청 신호 중에서 하나의 출력을 상기 기 설정된 간격만큼 지연시키는 단계를 더 포함할 수 있다.

[0014] 본 발명의 일 목적을 달성하기 위하여, 본 발명의 실시예들에 따른 인터럽트 스프레드 방법은 복수의 인터럽트 요청 신호들을 수신하는 단계, 상기 인터럽트 요청 신호들이 각각 출력될 복수의 프로세서들을 액티브(active) 상태인 프로세서들과 인액티브(inactive) 상태인 프로세서들로 분류하는 단계, 상기 인터럽트 요청 신호들 중에서 상기 액티브 상태인 프로세서들에 출력될 비대상(non-target) 인터럽트 요청 신호들을 상기 액티브 상태인 프로세서들에 각각 출력하는 단계, 상기 인터럽트 요청 신호들 중에서 상기 인액티브 상태인 프로세서들에 출력될 대상(target) 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단하는 단계, 상기 시간 간격이 상기 기 설정된 간격보다 작은 경우 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 단계, 및 상기 대상 인터럽트 요청 신호들을 상기 인액티브 상태인 프로세서들에 각각 출력하는 단계를 포함할 수 있다.

[0015] 일 실시예에 의하면, 상기 기 설정된 간격으로 조절하는 단계는 인접하는 제 k(단, k는 1이상인 정수) 대상 인터럽트 요청 신호와 제 k+1 대상 인터럽트 요청 신호 사이의 제 k 시간 간격이 0보다 큰 경우, 상기 제 k 시간 간격이 상기 기 설정된 간격으로 될 때까지 상기 제 k+1 대상 인터럽트 요청 신호의 출력을 지연시키는 단계를 포함할 수 있다.

[0016] 일 실시예에 의하면, 상기 기 설정된 간격으로 조절하는 단계는 기 설정된 우선순위에 따라 상기 대상 인터럽트 요청 신호들의 출력 순서를 변경하는 단계를 더 포함할 수 있다.

[0017] 일 실시예에 의하면, 상기 기 설정된 간격으로 조절하는 단계는 상기 제 k 시간 간격이 0인 경우, 상기 기 설정된 우선순위에 따라 상기 제 k 대상 인터럽트 요청 신호와 상기 제 k+1 대상 인터럽트 요청 신호 중에서 하나의 출력을 상기 기 설정된 간격만큼 지연시키는 단계를 더 포함할 수 있다.

[0018] 본 발명의 다른 목적을 달성하기 위하여, 본 발명의 실시예들에 따른 인터럽트 스프레드 장치는 제 1 내지 제 m(단, m은 2이상의 정수) 인터럽트 요청 신호들을 수신하고, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 적어도 기 설정된 간격 이상을 두어 제 1 내지 제 m 프로세서들에 각각 출력하는 제 1 내지 제 m 인터럽트 홀더들, 및 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격이 상기 기 설정된 간격보다 작은 경우 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 인터럽트 아비터를 포함할 수 있다.

[0019] 일 실시예에 의하면, 상기 제 1 내지 제 m 인터럽트 요청 신호들은 복수의 인터럽트 소스들로부터 출력되는 복수의 인터럽트들에 기초하여 생성되고, 상기 제 1 내지 제 m 인터럽트 요청 신호들은 상기 제 1 내지 제 m 인터럽트 홀더들에 각각 수신될 수 있다.

[0020] 일 실시예에 의하면, 상기 제 1 내지 제 m 인터럽트 홀더들은 상기 제 1 내지 제 m 프로세서들에 각각 연결되고, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 적어도 상기 기 설정된 간격 이상을 두어 상기 제 1 내지 제 m 프로세서들에 각각 출력할 수 있다.

[0021] 일 실시예에 의하면, 상기 기 설정된 간격은 상기 제 1 내지 제 m 프로세서들이 인액티브 상태에서 액티브 상태로 변경될 때, 상기 제 1 내지 제 m 프로세서들 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정될 수 있다.

[0022] 일 실시예에 의하면, 상기 제 1 내지 제 m 인터럽트 홀더들은 상기 제 1 내지 제 m 인터럽트 요청 신호들을 각각 수신하면, 상기 인터럽트 아비터에 출력 요청 신호를 송신하고, 상기 인터럽트 아비터로부터 출력 허가 신호를 수신하면, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 상기 제 1 내지 제 m 프로세서들에 각각 출력할 수 있다.

[0023] 일 실시예에 의하면, 상기 제 1 내지 제 m 인터럽트 홀더들 각각은 상기 제 1 내지 제 m 인터럽트 요청 신호들이 수신되지 않았음을 나타내는 아이들 스테이트(idle state), 상기 인터럽트 아비터로부터 상기 출력 허가 신호를 기다리고 있음을 나타내는 웨이트 스테이트(wait state) 및 상기 제 1 내지 제 m 인터럽트 요청 신호들이 상기 제 1 내지 제 m 프로세서들에 출력하고 있음을 나타내는 어서트 스테이트(assert state)로 구성된 스테이트 머신(state machine)으로 구현될 수 있다.

[0024] 일 실시예에 의하면, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 인터럽트 홀더들로부터 각각 상기 출력 요청 신호를 수신하면, 상기 제 1 내지 제 m 인터럽트 홀더들에 상기 출력 허가 신호를 순차적으로 출력하여, 상기 제 1 내지 제 m 인터럽트 요청 신호들이 적어도 상기 기 설정된 간격 이상으로 상기 제 1 내지 제 m 프로세서들에 각각 출력되도록 제어할 수 있다.

[0025] 일 실시예에 의하면, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 어느 하나도 상기 제 1 내지 제 m 프로세서들에 출력되고 있지 않음을 나타내는 아이들 스테이트, 및 상기 제 1 내지 제 n 인터럽트 요청 신호들 중에서 적어도 하나 이상이 상기 제 1 내지 제 m 프로세서들에 출력되고 있음을 나타내는 웨이트 스테이트로 구성된 스테이트 머신으로 구현될 수 있다.

[0026] 일 실시예에 의하면, 상기 인터럽트 아비터는 기 설정된 우선순위에 따라 상기 제 1 내지 제 m 인터럽트 요청 신호들의 출력 순서를 변경할 수 있다.

[0027] 일 실시예에 의하면, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 프로세서들을 액티브 상태인 프로세서들로 구성된 제 1 그룹과 인액티브 상태인 프로세서들로 구성된 제 2 그룹으로 분류하고, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 제 1 그룹에 할당된 인터럽트 요청 신호들에 대해서는 상기 기 설정된 간격으로 조절하지 않고, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 제 2 그룹에 할당된 인터럽트 요청 신호들에 대해서만 상기 기 설정된 간격으로 조절할 수 있다.

[0028] 본 발명의 또 다른 목적을 달성하기 위하여, 본 발명의 실시예들에 따른 시스템 온-칩은 복수의 인터럽트들을 생성하는 복수의 인터럽트 소스들, 상기 인터럽트들에 기초하여 제 1 내지 제 m(단, m은 2이상의 정수) 인터럽트 요청 신호들을 생성하는 인터럽트 컨트롤러, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격을 적어도 기 설정된 간격 이상으로 조절하는 인터럽트 스프레드 장치, 및 상기 제 1 내지 제 m 인터럽트 요청 신호들에 응답하여 상기 인터럽트 소스들을 위한 인터럽트 처리를 각각 수행하는 제 1 내지 제 m 프로세서들을 포함할 수 있다.

[0029] 일 실시예에 의하면, 상기 인터럽트 스프레드 장치는 상기 제 1 내지 제 m 인터럽트 요청 신호들을 수신하고, 상기 제 1 내지 제 m 인터럽트 요청 신호들을 적어도 상기 기 설정된 간격 이상을 두어 상기 제 1 내지 제 m 프로세서들에 각각 출력하는 제 1 내지 제 m 인터럽트 홀더들, 및 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 인접하는 인터럽트 요청 신호들 사이의 상기 시간 간격이 상기 기 설정된 간격보다 작은 경우 상기 시간 간격을 상기 기 설정된 간격으로 조절하는 인터럽트 아비터를 포함할 수 있다.

[0030] 일 실시예에 의하면, 상기 제 1 내지 제 m 인터럽트 요청 신호들은 상기 제 1 내지 제 m 인터럽트 홀더들에 각각 수신되고, 상기 제 1 내지 제 m 인터럽트 홀더들은 상기 제 1 내지 제 m 프로세서들에 각각 연결될 수 있다.

[0031] 일 실시예에 의하면, 상기 인터럽트 아비터는 기 설정된 우선순위에 따라 상기 제 1 내지 제 m 인터럽트 요청 신호들의 출력 순서를 변경할 수 있다.

[0032] 일 실시예에 의하면, 상기 인터럽트 아비터는 상기 제 1 내지 제 m 프로세서들을 액티브 상태인 프로세서들로

구성된 제 1 그룹과 인액티브 상태인 프로세서들로 구성된 제 2 그룹으로 분류하고, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 제 1 그룹에 할당된 인터럽트 요청 신호들에 대해서는 상기 기 설정된 간격으로 조절하지 않고, 상기 제 1 내지 제 m 인터럽트 요청 신호들 중에서 상기 제 2 그룹에 할당된 인터럽트 요청 신호들에 대해서만 상기 기 설정된 간격으로 조절할 수 있다.

[0033] 일 실시예에 의하면, 상기 시스템 온-칩은 휴대폰, 스마트폰, 스마트패드 등과 같은 모바일 기기에 구비될 수 있다.

[0034] 본 발명의 또 다른 목적을 달성하기 위하여, 본 발명의 실시예들에 따른 인터럽트 처리 방법은 복수의 프로세서들 중에서 일부가 액티브(active) 상태이고, 다른 일부는 인액티브(inactive) 상태인 상황에서, 복수의 인터럽트 소스들이 복수의 인터럽트들을 생성하면, 상기 프로세서들로 하여금 상기 인터럽트들을 처리할 수 있다. 구체적으로, 상기 인터럽트 처리 방법은 인터럽트 컨트롤러로 하여금 상기 인터럽트 소스들로부터 상기 인터럽트들을 수신하고, 상기 인터럽트들을 기 설정된 방식에 따라 상기 프로세서들 각각에 할당하여 복수의 인터럽트 요청 신호들을 출력하게 하고, 인터럽트 스프레더로 하여금 상기 인터럽트 컨트롤러로부터 상기 인터럽트 요청 신호들을 수신하고, 상기 인터럽트 요청 신호들을 상기 액티브 상태인 프로세서들에 출력될 비대상 인터럽트 요청 신호들과 상기 인액티브 상태인 프로세서들에 출력될 대상 인터럽트 요청 신호들로 분류하게 하며, 상기 인터럽트 스프레더로 하여금 상기 비대상 인터럽트 요청 신호들을 각각 상기 액티브 상태인 프로세서들에 바로 출력하게 하고, 상기 인터럽트 스프레더가 상기 대상 인터럽트 요청 신호들 사이의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단하고, 상기 시간 간격이 상기 기 설정된 간격보다 작은 경우, 상기 시간 간격을 상기 기 설정된 간격으로 조절하게 하며, 상기 인터럽트 스프레더가 상기 대상 인터럽트 요청 신호들을 각각 상기 인액티브 상태인 프로세서들에 순차적으로 출력하게 할 수 있다.

발명의 효과

[0035] 본 발명의 실시예들에 따른 인터럽트 스프레드 방법은 복수의 인터럽트 소스들이 짧은 시간 내에 복수의 인터럽트들을 연속적으로 출력할 때, 상기 인터럽트들에 기초하여 생성되는 복수의 인터럽트 요청 신호들 사이의 각각의 시간 간격을 적어도 기 설정된 간격 이상으로 조절함으로써, 인액티브 상태(예를 들어, 파워 다운 상태, 파워 오프 상태 등)에 있는 복수의 프로세서들(또는, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태(예를 들어, 파워 온 상태 등)로 연속적으로 변경되는 것(즉, 급작스러운 웨이크-업)을 방지할 수 있다. 이에, 상기 프로세서들 내부에 인러쉬 전류가 생성되는 것을 방지할 수 있다.

[0036] 본 발명의 실시예들에 따른 인터럽트 스프레드 장치는 복수의 인터럽트 소스들이 짧은 시간 내에 복수의 인터럽트들을 연속적으로 출력할 때, 상기 인터럽트들에 기초하여 생성되는 복수의 인터럽트 요청 신호들 사이의 각각의 시간 간격을 적어도 기 설정된 간격 이상으로 조절함으로써, 인액티브 상태(예를 들어, 파워 다운 상태, 파워 오프 상태 등)에 있는 복수의 프로세서들(예를 들어, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태(예를 들어, 파워 온 상태 등)로 연속적으로 변경되는 것(즉, 급작스러운 웨이크-업)을 방지할 수 있다. 이에, 상기 프로세서들 내부에 인러쉬 전류가 생성되는 것을 방지할 수 있다.

[0037] 본 발명의 실시예들에 따른 시스템 온-칩은 상기 인터럽트 스프레드 장치를 구비함으로써, 인액티브 상태(예를 들어, 파워 다운 상태, 파워 오프 상태 등)에 있는 복수의 프로세서들(예를 들어, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태(예를 들어, 파워 온 상태 등)로 연속적으로 변경됨(즉, 급작스러운 웨이크-업)에 따른 오동작을 방지하여 높은 동작 안정성을 확보할 수 있다.

[0038] 다만, 본 발명의 효과는 상기 언급한 효과에 한정되는 것이 아니며, 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위에서 다양하게 확장될 수 있을 것이다.

도면의 간단한 설명

[0039] 도 1은 본 발명의 일 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

도 2a 및 도 2b는 도 1의 인터럽트 스프레드 방법이 수행되는 일 예를 나타내는 도면들이다.

도 3a 및 도 3b는 도 1의 인터럽트 스프레드 방법이 수행되는 다른 예를 나타내는 도면들이다.

도 4a 및 도 4b는 도 1의 인터럽트 스프레드 방법이 수행되는 또 다른 예를 나타내는 도면들이다.

도 5는 본 발명의 다른 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

도 6a 및 도 6b는 도 5의 인터럽트 스프레드 방법이 수행되는 일 예를 나타내는 도면들이다.

도 7은 본 발명의 또 다른 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

도 8a 및 도 8b는 도 7의 인터럽트 스프레드 방법이 수행되는 일 예를 나타내는 도면들이다.

도 9는 본 발명의 또 다른 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

도 10은 도 9의 인터럽트 스프레드 방법을 설명하기 위한 도면이다.

도 11은 본 발명의 일 실시예에 따른 인터럽트 스프레드 장치를 나타내는 블록도이다.

도 12는 도 11의 인터럽트 스프레드 장치에 구비된 인터럽트 홀더의 스테이트 머신을 나타내는 도면이다.

도 13은 도 11의 인터럽트 스프레드 장치에 구비된 인터럽트 아비터의 스테이트 머신을 나타내는 도면이다.

도 14는 도 11의 인터럽트 스프레드 장치가 동작하는 일 예를 나타내는 타이밍도이다.

도 15는 본 발명의 일 실시예에 따른 시스템 온-칩을 나타내는 블록도이다.

도 16은 도 15의 시스템 온-칩에서 인터럽트 스프레드 장치가 동작하는 일 예를 나타내는 도면이다.

도 17은 도 15의 시스템 온-칩에서 인터럽트 스프레드 장치가 동작하는 다른 예를 나타내는 도면이다.

도 18은 도 15의 시스템 온-칩에서 인터럽트 스프레드 장치가 동작하는 또 다른 예를 나타내는 도면이다.

도 19는 임의의 시스템 온-칩이 인터럽트 스프레드 장치를 구비하고 있는지 여부를 검증하는 방법의 일 예를 나타내는 순서도이다.

도 20은 임의의 시스템 온-칩이 인터럽트 스프레드 장치를 구비하고 있는지 여부를 검증하는 방법의 일 예를 설명하기 위한 도면이다.

도 21은 임의의 시스템 온-칩이 인터럽트 스프레드 장치를 구비하고 있는지 여부를 검증하는 방법의 다른 예를 나타내는 순서도이다.

도 22는 본 발명의 일 실시예에 따른 멀티 코어 시스템을 나타내는 블록도이다.

도 23은 도 22의 멀티 코어 시스템이 스마트폰으로 구현되는 일 예를 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0040] 본문에 개시되어 있는 본 발명의 실시예들에 대해서, 특정한 구조적 내지 기능적 설명들은 단지 본 발명의 실시예를 설명하기 위한 목적으로 예시된 것으로, 본 발명의 실시예들은 다양한 형태로 실시될 수 있으며 본문에 설명된 실시예들에 한정되는 것으로 해석되어서는 아니 된다.

[0041] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는바, 특정 실시예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 개시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0042] 제 1, 제 2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로 사용될 수 있다. 예를 들어, 본 발명의 권리 범위로부터 이탈되지 않은 채 제 1 구성요소는 제 2 구성요소로 명명될 수 있고, 유사하게 제 2 구성요소도 제 1 구성요소로 명명될 수 있다.

[0043] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0044] 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 설시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이

존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0045] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미이다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미인 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0046] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[0047] 도 1은 본 발명의 일 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

[0048] 도 1을 참조하면, 도 1의 인터럽트 스프레드 방법은 복수의 인터럽트 요청 신호들을 수신(Step S120)하고, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단(Step S140)할 수 있다. 이 때, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작으면, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들 사이의 각각의 시간 간격을 기 설정된 간격으로 조절(Step S160)할 수 있다. 반면에, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 크면, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들 사이의 각각의 시간 간격을 그대로 유지(Step S165)시킬 수 있다. 이후, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들을 복수의 프로세서들에 각각 출력(Step S180)할 수 있다. 한편, 복수의 프로세서들 각각은 독립적인 프로세서들에 상응할 수 있고, 멀티 코어 프로세서(예를 들어, 듀얼 코어 프로세서, 쿼드 코어 프로세서 등)의 복수의 코어(core)들에 상응할 수도 있다. 한편, 설명의 편의를 위하여, 본 명세서에서는 액티브(active) 상태와 인액티브(inactive) 상태로 구분하였지만, 인액티브 상태는 파워 다운 상태(power-down state), 파워 오프 상태(power-off state)와 같은 저전력 모드의 다양한 상태들을 모두 포함하는 개념으로 해석되어야 한다. 이하, 도 1의 인터럽트 스프레드 방법에 대하여 구체적으로 설명하기로 한다.

[0049] 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들을 순차적으로 수신(Step S120)할 수 있다. 일 실시예에서, 인터럽트 요청 신호들은 인터럽트 소스(interrupt source)들로부터 출력되는 인터럽트들에 기초하여 생성될 수 있고, 인터럽트 요청 신호들은 프로세서들에 각각 할당될 수 있다. 인터럽트 소스들은 각각 멀티 코어 시스템(또는, 멀티 프로세서 시스템) 내에서 소정의 동작을 수행하는 아이피(Intellectual Property; IP)들로서, 시스템 온-칩(System On-Chip; SOC)을 구성하는 여러 구성 요소들 예를 들어, 비디오 모듈, 사운드 모듈, 디스플레이 모듈, 메모리 모듈, 통신 모듈, 카메라 모듈 등과 같은 소정의 모듈들에 상응할 수 있다. 일반적으로, 인터럽트 소스들은 하나의 시스템 온-칩 내에 대략 수십 내지 수백 개 정도가 실장될 수 있다. 이러한 인터럽트 소스들이 인터럽트들을 발생시키면, 상기 인터럽트들에 기초하여 생성된 인터럽트 요청 신호들이 프로세서들에 각각 입력될 수 있고, 상기 프로세서들은 인터럽트 요청 신호들에 응답하여 상기 인터럽트 소스들을 위한 인터럽트 처리를 각각 수행할 수 있다. 최근, 전자 기기의 소형화 및 경량화 추세에 따라, 복수의 아이피들과 복수의 프로세서들이 하나의 칩에 실장되는 시스템 온-칩이 널리 사용되고 있고, 시스템 온-칩 내부의 프로세서들은 전력 소모를 줄이기 위하여 저전력 모드로 빈번하게 진입을 하기 때문에, 상기 인터럽트 소스들로부터 생성된 인터럽트들에 기초하여 복수의 프로세서들이 저전력 모드에서 동시에 탈출(즉, 급작스러운 웨이크-업)하게 되면, 상기 프로세서들 내부에 인러쉬 전류(in-rush current)가 생성되어 인러쉬 전류에 의한 오동작이 야기될 수 있다. 특히, 시스템 온-칩의 사이즈가 점점 작아짐에 따라 다이나믹 전류(dynamic current)의 변화에 따른 인러쉬 전류의 발생은 시스템 온-칩의 동작 안정성에 큰 문제가 되고 있다.

[0050] 이에, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들이 수신되면, 이러한 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단(Step S140)할 수 있다. 일 실시예에서, 기 설정된 간격은 복수의 프로세서들이 인액티브 상태에서 액티브 상태로 변경될 때, 상기 프로세서들 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정될 수 있다. 이 때, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작으면, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들 사이의 각각의 시간 간격을 기 설정된 간격으로 조절(Step S160)할 수 있다. 반면에, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 크면, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들 사이의 각각의 시간 간격을 그대로 유지(Step S165)시킬 수 있다. 다시 말하면, 도 1의 인터럽트 스프레드 방법은 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은 경우에만, 그 시간 간격을 기 설정된 간격으로 조절하여 프

로세서들에 각각 출력하고, 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 큰 경우에는, 그 시간 간격을 조절하지 않은 채 프로세서들에 각각 출력할 수 있다. 그 결과, 인터럽트 요청 신호들은 기 설정된 간격 이상을 두어 프로세서들에 각각 입력될 수 있다.

[0051] 구체적으로, 도 1의 인터럽트 스프레드 방법은, 인터럽트 요청 신호들 사이의 각각의 시간 간격을 기 설정된 간격으로 조절(Step S160)함에 있어서, 순차적으로 입력되고 인접하는 제 k(단, k는 1이상인 정수) 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 0보다 큰 경우, 상기 제 k 시간 간격이 기 설정된 간격으로 될 때까지 제 k+1 인터럽트 요청 신호의 출력을 지연시킬 수 있다. 즉, 순차적으로 입력되고 인접하는 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 기 설정된 간격보다 작음에도 불구하고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 그대로 두 개의 프로세서들에 각각 출력되면, 두 개의 프로세서들이 짧은 시간 내에 급작스러운 웨이크-업을 하게 되므로, 상기 프로세서들 내부에 인러쉬 전류가 생성될 수 있다. 그러므로, 도 1의 인터럽트 스프레드 방법은 제 k+1 인터럽트 요청 신호의 출력을 지연시킴으로써 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 기 설정된 간격이 되도록 조절하는 것이다. 한편, 도 1의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 0인 경우(즉, 동시에 입력되는 경우)에는, 기 설정된 우선순위에 따라 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 중에서 하나의 출력을 기 설정된 간격만큼 지연시킬 수 있다. 그 결과, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 동시에 입력되더라도, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 기 설정된 간격이 되므로, 상기 프로세서들 내부에 인러쉬 전류가 생성되지 않게 된다.

[0052] 나아가, 도 1의 인터럽트 스프레드 방법은, 인터럽트 요청 신호들 사이의 각각의 시간 간격을 기 설정된 간격으로 조절(Step S160)함에 있어서, 기 설정된 우선순위에 따라 인터럽트 요청 신호들의 출력 순서를 변경할 수 있다. 예를 들어, 제 k 인터럽트 요청 신호, 제 k+1 인터럽트 요청 신호 및 제 k+2 인터럽트 요청 신호가 순차적으로 입력되고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격 및 제 k+1 인터럽트 요청 신호와 제 k+2 인터럽트 요청 신호 사이의 제 k+1 시간 간격이 모두 기 설정된 간격보다 작은 경우에, 도 1의 인터럽트 스프레드 방법은 상기 제 k 시간 간격이 기 설정된 간격이 되도록 제 k+1 인터럽트 요청 신호의 출력을 지연시킨 후, 상기 제 k+1 시간 간격이 기 설정된 간격이 되도록 제 k+2 인터럽트 요청 신호의 출력을 지연시키게 된다. 그러나, 도 1의 인터럽트 스프레드 방법은 요구되는 조건에 따라 제 k 인터럽트 요청 신호, 제 k+1 인터럽트 요청 신호 및 제 k+2 인터럽트 요청 신호의 출력을 기 설정된 우선순위에 기초하여 비순차적으로 지연시킬 수 있다. 예를 들어, 제 k 인터럽트 요청 신호, 제 k+1 인터럽트 요청 신호 및 제 k+2 인터럽트 요청 신호가 순차적으로 입력되고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격 및 제 k+1 인터럽트 요청 신호와 제 k+2 인터럽트 요청 신호 사이의 제 k+1 시간 간격이 모두 기 설정된 간격보다 작은 경우에, 제 k 인터럽트 요청 신호의 우선순위가 가장 높고 제 k+1 인터럽트 요청 신호의 우선순위가 가장 낮다면, 도 1의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호, 제 k+2 인터럽트 요청 신호 및 제 k+1 인터럽트 요청 신호 순으로 출력시킬 수 있다.

[0053] 일반적으로, 프로세서는 인터럽트 요청 신호가 수신되면, 인액티브 상태에서 액티브 상태로 변경된 후, 인터럽트 처리(즉, 인터럽트 서비스)를 수행하게 된다. 이 때, 프로세서 내부를 구성하는 다수의 트랜지스터들이 갑작스럽게 동작을 시작하면서 인러쉬 전류가 생성되게 된다. 특히, 저전력 프로세서의 경우에는, 인액티브 상태에서 소모 전력을 줄이기 위해 다수의 클럭 게이팅 회로(clock gating circuit)를 구비하기 때문에, 인액티브 상태에서 대부분의 클럭들이 토글(toggle)하지 않다가, 저전력 프로세서가 인액티브 상태에서 액티브 상태로 변경되면, 대부분의 클럭들이 토글을 동시에 시작하게 되어 인러쉬 전류가 크게 생성될 수 있다. 이 때, 복수의 프로세서들이 짧은 시간 내에 급작스러운 웨이크-업을 하게 되면, 복수의 프로세서들에서 생성된 인러쉬 전류가 더해져서, 주변 회로의 전력 공급이 불안정하게 됨으로써 오동작이 발생할 수 있다. 이에, 도 1의 인터럽트 스프레드 방법은 복수의 인터럽트 소스들이 짧은 시간 내에 복수의 인터럽트들을 연속적으로 출력할 때, 상기 인터럽트들에 기초하여 생성되는 복수의 인터럽트 요청 신호들 사이의 시간 간격을 적어도 기 설정된 간격 이상으로 조절함으로써, 인액티브 상태에 있는 복수의 프로세서들(예를 들어, 멀티 코어 프로세서 등)이 짧은 시간 내에 액티브 상태로 연속적으로 변경되는 것(즉, 급작스러운 웨이크-업)을 방지할 수 있다. 그 결과, 시스템 온-칩에서 복수의 프로세서들(또는, 멀티 코어 프로세서의 복수의 코어들) 내부에 급작스러운 웨이크-업에 의한 인러쉬 전류가 생성되는 것이 방지되어, 시스템 온-칩 및 이를 포함하는 전자 기기는 높은 동작 안정성을 확보할 수 있다. 한편, 도 1에서는 복수의 프로세서들이 액티브 상태인지 인액티브 상태인지 여부와 상관없이, 모든 프로세서들에 입력되는 인터럽트 요청 신호들에 대하여 도 1의 인터럽트 스프레드 방법이 적용되는 것으로 설명되었지만, 인액티브 상태인 프로세서들에 입력되는 인터럽트 요청 신호들에 대해서만 도 1의 인터럽트 스프레드

방법이 적용될 수도 있다.

[0054] 도 2a 및 도 2b는 도 1의 인터럽트 스프레드 방법이 수행되는 일 예를 나타내는 도면들이다.

[0055] 도 2a 및 도 2b를 참조하면, 도 2a는 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 순차적으로 입력되는 것을 보여주고 있고, 도 2b는 도 1의 인터럽트 스프레드 방법에 의하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 제 1 내지 제 4 프로세서들에 순차적으로 출력되는 것을 보여주고 있다.

[0056] 구체적으로, 도 1의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 순차적으로 수신할 수 있다. 도 2a에 도시된 바와 같이, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)은 기 설정된 간격(PS)보다 작고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)도 기 설정된 간격(PS)보다 작으며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)도 기 설정된 간격(PS)보다 작다. 이에, 도 2b에 도시된 바와 같이, 도 1의 인터럽트 스프레드 방법은 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)이 기 설정된 간격(PS)이 되도록 제 2 인터럽트 요청 신호(nIRQ_2)의 출력을 지연시키고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)이 기 설정된 간격(PS)이 되도록 제 3 인터럽트 요청 신호(nIRQ_3)의 출력을 지연시키며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)이 기 설정된 간격(PS)이 되도록 제 4 인터럽트 요청 신호(nIRQ_4)의 출력을 지연시킬 수 있다. 이와 같이, 도 1의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 기 설정된 간격(PS)을 두어 제 1 내지 제 4 프로세서들에 순차적으로 출력할 수 있다.

[0057] 도 3a 및 도 3b는 도 1의 인터럽트 스프레드 방법이 수행되는 다른 예를 나타내는 도면들이다.

[0058] 도 3a 및 도 3b를 참조하면, 도 3a는 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 순차적으로 입력되는 것을 보여주고 있고, 도 3b는 도 1의 인터럽트 스프레드 방법에 의하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 제 1 내지 제 4 프로세서들에 순차적으로 출력되는 것을 보여주고 있다.

[0059] 구체적으로, 도 1의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 순차적으로 수신할 수 있다. 도 3a에 도시된 바와 같이, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)은 기 설정된 간격(PS)보다 크고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)은 기 설정된 간격(PS)보다 작으며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)은 기 설정된 간격(PS)보다 작다. 이에, 도 3b에 도시된 바와 같이, 도 1의 인터럽트 스프레드 방법은 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)은 그대로 유지시키고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)이 기 설정된 간격(PS)이 되도록 제 3 인터럽트 요청 신호(nIRQ_3)의 출력을 지연시키며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)이 기 설정된 간격(PS)이 되도록 제 4 인터럽트 요청 신호(nIRQ_4)의 출력을 지연시킬 수 있다. 이와 같이, 도 1의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 기 설정된 간격(PS) 이상을 두어 제 1 내지 제 4 프로세서들에 순차적으로 출력할 수 있다.

[0060] 도 4a 및 도 4b는 도 1의 인터럽트 스프레드 방법이 수행되는 또 다른 예를 나타내는 도면들이다.

[0061] 도 4a 및 도 4b를 참조하면, 도 4a는 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 순차적으로 입력되는 것을 보여주고 있고, 도 4b는 도 1의 인터럽트 스프레드 방법에 의하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 제 1 내지 제 4 프로세서들에 순차적으로 출력되는 것을 보여주고 있다.

[0062] 구체적으로, 도 1의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 순차적으로 수신할 수 있다. 도 4a에 도시된 바와 같이, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)은 기 설정된 간격(PS)보다 작고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)은 기 설정된 간격(PS)보다 작으며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)도 기 설정된 간격(PS)보다 작다. 그러나, 시스템 온-칩의 성능에 따라, 2개의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업을 하더라도, 2개의 프로세서들 내부에 인러쉬 전류가 생성되지 않을 수 있고, 3개의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업을 하더라도, 3개의 프로세서들 내부에 인러쉬 전류가 생성되지 않을 수 있다. 이하,

도 4a 및 도 4b에서는 2개의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업하는 것을 허용하나, 3개의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업하는 것은 허용하지 않는 것으로 가정하고 설명하기로 한다. 예를 들어, 도 4b에 도시된 바와 같이, 도 1의 인터럽트 스프레드 방법은 기 설정된 개수(즉, 2개)의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업을 하는 것을 허용하고 있으므로, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)이 기 설정된 간격(PS)보다 작더라도, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)을 그대로 유지시킬 수 있다. 그러나, 도 1의 인터럽트 스프레드 방법은 3개의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업을 하는 것을 허용하지 않으므로, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)이 기 설정된 간격(PS)보다 작기 때문에, 상기 제 2 시간 간격(S2)이 기 설정된 간격(PS)이 되도록 제 3 인터럽트 요청 신호(nIRQ_3)의 출력을 지연시킬 수 있다. 이후, 도 1의 인터럽트 스프레드 방법은 2개의 프로세서들이 짧은 시간 내에 연속적으로 웨이크-업을 하는 것을 허용하고 있으므로, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)이 기 설정된 간격(PS)보다 작더라도, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)을 그대로 유지시킬 수 있다.

[0063] 도 5는 본 발명의 다른 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

[0064] 도 5를 참조하면, 도 5의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호를 수신(Step S210)하고, 제 k+1 인터럽트 요청 신호를 수신(Step S220)한 후, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0보다 큰지 여부를 판단(Step S230)할 수 있다. 이 때, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0보다 큰 경우, 상기 시간 간격이 기 설정된 간격으로 될 때까지 제 k 인터럽트 요청 신호의 출력을 지연(Step S240)시키고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0인 경우, 기 설정된 우선순위에 따라 제 k 인터럽트 요청 신호 또는 제 k+1 인터럽트 요청 신호의 출력을 기 설정된 간격만큼 지연(Step S250)시킬 수 있다. 다만, 도 5에서는 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 기 설정된 간격보다 작은 것으로 가정되어 있다.

[0065] 도 5의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호를 순차적으로 수신하고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단함에 있어서, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0보다 큰지 여부까지 판단한다. 즉, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 순차적으로 입력된다고 보면, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격은 0보다 큰 것이 일반적이나, 실시예에 따라 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 실질적으로 동시에 입력될 수도 있으므로, 도 5의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0보다 큰지 여부까지 판단하는 것이다. 이 때, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0보다 큰 경우, 도 5의 인터럽트 스프레드 방법은 상기 시간 간격이 기 설정된 간격으로 될 때까지 제 k 인터럽트 요청 신호의 출력을 지연시킬 수 있고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 0인 경우, 도 5의 인터럽트 스프레드 방법은 기 설정된 우선순위에 따라 제 k 인터럽트 요청 신호 또는 제 k+1 인터럽트 요청 신호의 출력을 기 설정된 간격만큼 지연시킬 수 있다. 그 결과, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 동시에 입력되는 경우에도 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 기 설정된 간격을 두어 각각 제 k 프로세서와 제 k+1 프로세서에 출력되기 때문에, 도 5의 인터럽트 스프레드 방법은 시스템 온-칩에서 짧은 시간 내에 복수의 프로세서들이 인액티브 상태에서 액티브 상태로 변경(즉, 급작스러운 웨이크-업)되는 것을 방지할 수 있다.

[0066] 도 6a 및 도 6b는 도 5의 인터럽트 스프레드 방법이 수행되는 일 예를 나타내는 도면들이다.

[0067] 도 6a 및 도 6b를 참조하면, 도 6a는 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 순차적으로 입력되는 것을 보여주고 있고, 도 6b는 도 5의 인터럽트 스프레드 방법에 의하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 제 1 내지 제 4 프로세서들에 순차적으로 출력되는 것을 보여주고 있다.

[0068] 구체적으로, 도 5의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 순차적으로 수신할 수 있는데, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4) 중에서 인접하는 인터럽트 요청 신호들이 실질적으로 동시에 입력될 수 있다. 예를 들어, 도 6a에 도시된 바와 같이, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)은 기 설정된 간격(PS)보다 작고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)도

기 설정된 간격(PS)보다 작으며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)은 0에 상응한다. 따라서, 제 4 인터럽트 요청 신호(nIRQ_4)의 우선순위가 제 3 인터럽트 요청 신호(nIRQ_3)의 우선순위보다 높다고 가정하면, 도 6b에 도시된 바와 같이, 도 5의 인터럽트 스프레드 방법은 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)이 기 설정된 간격(PS)이 되도록 제 2 인터럽트 요청 신호(nIRQ_2)의 출력을 지연시키고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 2 시간 간격(S2)이 기 설정된 간격(PS)이 되도록 제 4 인터럽트 요청 신호(nIRQ_4)의 출력을 지연시키며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)은 기 설정된 간격(PS)이 되도록 제 3 인터럽트 요청 신호(nIRQ_3)의 출력을 지연시킬 수 있다. 이 때, 제 4 인터럽트 요청 신호(nIRQ_4)의 우선순위가 제 3 인터럽트 요청 신호(nIRQ_3)의 우선순위보다 높기 때문에, 제 3 인터럽트 요청 신호(nIRQ_3)보다 제 4 인터럽트 요청 신호(nIRQ_4)가 먼저 출력된 것이다. 이와 같이, 도 5의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4) 중에서 인접하는 인터럽트 요청 신호들이 실질적으로 동시에 입력되더라도, 기 설정된 우선순위에 따라 기 설정된 간격(PS)을 두어, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 제 1 내지 제 4 프로세서들에 순차적으로 출력할 수 있다.

[0069] 도 7은 본 발명의 또 다른 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이다.

[0070] 도 7을 참조하면, 도 7의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호를 수신(Step S310)하고, 제 k+1 인터럽트 요청 신호를 수신(Step S320)한 후, 제 k 인터럽트 요청 신호가 제 k+1 인터럽트 요청 신호보다 우선순위가 높은지 여부를 판단(Step S330)할 수 있다. 이 때, 도 7의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호의 우선순위가 제 k+1 인터럽트 요청 신호의 우선순위보다 높으면, 제 k+1 인터럽트 요청 신호를 지연(Step S340)시킬 수 있고, 제 k 인터럽트 요청 신호의 우선순위가 제 k+1 인터럽트 요청 신호의 우선순위보다 낮으면, 제 k 인터럽트 요청 신호를 지연(Step S345)시킬 수 있다. 다만, 도 7에서는 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 기 설정된 간격보다 작은 것으로 가정되어 있다.

[0071] 도 7의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호를 순차적으로 수신하고, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단함에 있어서, 제 k 인터럽트 요청 신호가 제 k+1 인터럽트 요청 신호보다 우선순위가 높은지 여부까지 판단할 수 있다. 이에, 도 7의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호의 입력 순서와는 별개로, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호의 출력 순서를 결정할 수 있다. 구체적으로, 도 7의 인터럽트 스프레드 방법은 제 k 인터럽트 요청 신호의 우선순위가 제 k+1 인터럽트 요청 신호의 우선순위보다 높으면, 제 k+1 인터럽트 요청 신호를 지연시킬 수 있고, 제 k 인터럽트 요청 신호의 우선순위가 제 k+1 인터럽트 요청 신호의 우선순위보다 낮으면, 제 k 인터럽트 요청 신호를 지연시킬 수 있다. 그 결과, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 순차적으로 입력되는 경우에도, 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호가 기 설정된 우선순위에 따라 비순차적으로 출력되고, 그 때에도 제 k+1 인터럽트 요청 신호와 제 k 인터럽트 요청 신호가 기 설정된 간격을 두어 각각 제 k 프로세서와 제 k+1 프로세서에 출력되기 때문에, 도 7의 인터럽트 스프레드 방법은 시스템 온-칩에서 짧은 시간 내에 복수의 프로세서들이 인액티브 상태에서 액티브 상태로 변경(즉, 급작스러운 웨이크-업)되는 것을 방지할 수 있다.

[0072] 도 8a 및 도 8b는 도 7의 인터럽트 스프레드 방법이 수행되는 일 예를 나타내는 도면들이다.

[0073] 도 8a 및 도 8b를 참조하면, 도 8a는 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 순차적으로 입력되는 것을 보여주고 있고, 도 8b는 도 7의 인터럽트 스프레드 방법에 의하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 기 설정된 우선순위에 따라 제 1 내지 제 4 프로세서들에 각각 출력되는 것을 보여주고 있다.

[0074] 구체적으로, 도 7의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 순차적으로 수신할 수 있다. 도 8a에 도시된 바와 같이, 제 1 인터럽트 요청 신호(nIRQ_1)와 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 제 1 시간 간격(S1)은 기 설정된 간격(PS)보다 작고, 제 2 인터럽트 요청 신호(nIRQ_2)와 제 3 인터럽트 요청 신호(nIRQ_3) 사이의 제 2 시간 간격(S2)도 기 설정된 간격(PS)보다 작으며, 제 3 인터럽트 요청 신호(nIRQ_3)와 제 4 인터럽트 요청 신호(nIRQ_4) 사이의 제 3 시간 간격(S3)도 기 설정된 간격(PS)보다 작다. 이 때, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)이 제 1 내지 제 4 프로세서들에 각각 출력됨에 있어서, 요구되는 조건에 따라, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)의 입력 순서와는 다르게, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)의 출력 순서가

변경될 필요도 있다. 따라서, 도 7의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)의 출력 순서를 기 설정된 우선순위에 따라 변경시킴으로써, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)에 응답하여 인터럽트 처리를 수행하는 제 1 내지 제 4 프로세서들의 웨이크-업 순서를 변경시킬 수 있다. 예를 들어, 도 8b에 도시된 바와 같이, 도 7의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_1, ..., nIRQ_4)을 제 1 인터럽트 요청 신호(nIRQ_1), 제 3 인터럽트 요청 신호(nIRQ_3), 제 4 인터럽트 요청 신호(nIRQ_4) 및 제 2 인터럽트 요청 신호(nIRQ_2) 순으로 출력할 수 있다. 마찬가지로, 제 1 인터럽트 요청 신호(nIRQ_1), 제 3 인터럽트 요청 신호(nIRQ_3), 제 4 인터럽트 요청 신호(nIRQ_4) 및 제 2 인터럽트 요청 신호(nIRQ_2) 사이의 각각의 시간 간격은 기 설정된 간격(PS)으로 조절된다. 이와 같이, 도 7의 인터럽트 스프레드 방법은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)을 기 설정된 우선순위에 따라 출력 순서를 결정하여 제 1 내지 제 4 프로세서들에 각각 출력할 수 있다.

[0075] 도 9는 본 발명의 또 다른 실시예에 따른 인터럽트 스프레드 방법을 나타내는 순서도이고, 도 10은 도 9의 인터럽트 스프레드 방법을 설명하기 위한 도면이다.

[0076] 도 9 및 도 10을 참조하면, 도 9의 인터럽트 스프레드 방법은 인터럽트 요청 신호들을 순차적으로 수신(Step S410)하면, 상기 인터럽트 요청 신호들을 대상(target) 인터럽트 요청 신호들과 비대상(non-target) 인터럽트 요청 신호들로 구분(Step S420)할 수 있다. 구체적으로, 대상 인터럽트 요청 신호들은 인액티브 상태인 프로세서들(TP_1, ..., TP_4)에 출력될 인터럽트 요청 신호들을 의미하고, 비대상 인터럽트 요청 신호들은 액티브 상태인 프로세서들(NTP_1, NTP_2)에 출력될 인터럽트 요청 신호들을 의미한다. 이후, 도 9의 인터럽트 스프레드 방법은 비대상 인터럽트 요청 신호들을 액티브 상태인 프로세서들(NTP_1, NTP_2)에 각각 출력(Step S430)할 수 있다. 한편, 도 9의 인터럽트 스프레드 방법은 대상 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은지 여부를 판단(Step S440)하여, 대상 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 작은 경우, 상기 시간 간격을 기 설정된 간격으로 조절(Step S450)할 수 있고, 대상 인터럽트 요청 신호들 사이의 각각의 시간 간격이 기 설정된 간격보다 큰 경우, 상기 시간 간격을 그대로 유지(Step S460)시킬 수 있다. 이후, 도 9의 인터럽트 스프레드 방법은 대상 인터럽트 요청 신호들을 인액티브 상태인 프로세서들(TP_1, ..., TP_4)에 각각 출력(Step S470)할 수 있다. 일반적으로, 인러쉬 전류는 짧은 시간 내에 복수의 프로세서들이 인액티브 상태에서 액티브 상태로 연속적으로 변경(즉, 급작스러운 웨이크-업)될 때 생성되므로, 도 9의 인터럽트 스프레드 방법은 웨이크-업이 필요 없는 액티브 상태인 프로세서들(NTP_1, NTP_2)에 출력되는 비대상 인터럽트 요청 신호들에 대해서는 그 시간 간격을 기 설정된 간격으로 조절하지 않는다. 그 결과, 도 9의 인터럽트 스프레드 방법은 인액티브 상태인 프로세서들(TP_1, ..., TP_4)에 출력되는 대상 인터럽트 요청 신호들에 대해서만 그 시간 간격을 기 설정된 간격으로 조절하기 때문에, 대상이 되는 인터럽트 요청 신호들이 상대적으로 감소(즉, 상대적으로 로드(load)가 감소)하여 고속으로 동작할 수 있다.

[0077] 도 11은 본 발명의 일 실시예에 따른 인터럽트 스프레드 장치를 나타내는 블록도이다.

[0078] 도 11을 참조하면, 인터럽트 스프레드 장치(100)는 제 1 내지 제 m(단, m은 2이상의 정수) 인터럽트 홀더들(120_1, ..., 120_m) 및 인터럽트 아비터(140)를 포함할 수 있다.

[0079] 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)은 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)을 각각 수신하고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)을 기 설정된 간격 이상을 두어 제 1 내지 제 m 프로세서들(미도시)에 각각 출력할 수 있다. 도 11에 도시된 바와 같이, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)은 복수의 인터럽트 소스들(미도시)로부터 출력되는 복수의 인터럽트들에 기초하여 생성되고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)은 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)에 각각 수신될 수 있다. 한편, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)은 제 1 내지 제 m 프로세서들에 각각 연결되고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)을 기 설정된 간격 이상을 두어 제 1 내지 제 m 프로세서들에 각각 출력할 수 있다. 구체적으로, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)을 수신하면, 인터럽트 아비터(140)에 출력 요청 신호(RS1, ..., RSm)를 송신하고, 인터럽트 아비터(140)로부터 출력 허가 신호(AS1, ..., ASm)를 수신하면, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)을 제 1 내지 제 m 프로세서들에 각각 출력할 수 있다. 이를 위하여, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 아이들 스테이트(idle state), 웨이트 스테이트(wait state) 및 어서트 스테이트(assert state)로 구성된 스테이트 머신(state machine)으로 구현될 수 있는데, 이에 대해서는 도 12를 참조하여 후술하기로 한다.

[0080] 인터럽트 아비터(140)는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격이 기 설정된 간격보다 작은 경우, 상기 시간 간격을 기 설정된 간격으로 조절할 수 있다. 이 때, 상기 기 설정된 간격은 제 1 내지 제 m 프로세서들이 인액티브 상태에서 액티브 상태로 변경될 때, 제 1 내지 제 m 프로세서들 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정될 수 있다. 구체적으로, 인터럽트 아비터(140)는 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)로부터 각각 출력 요청 신호(RS1, ..., RSm)를 수신하면, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)에 출력 허가 신호(AS1, ..., ASm)를 순차적으로 출력함으로써, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)이 기 설정된 간격 이상으로 제 1 내지 제 m 프로세서들에 각각 출력되게 할 수 있다. 이를 위하여, 인터럽트 아비터(140)는 아이들 스테이트 및 웨이트 스테이트로 구성된 스테이트 머신으로 구현될 수 있는데, 이에 대해서는 도 13을 참조하여 후술하기로 한다. 일 실시예에서, 인터럽트 아비터(140)는 기 설정된 우선순위에 따라 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)의 출력 순서를 변경할 수 있다. 그 결과, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)에 입력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)의 입력 순서와 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)로부터 출력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)의 출력 순서는 서로 상이할 수 있다.

[0081] 한편, 인터럽트 아비터(140)는 제 1 내지 제 m 프로세서들을 액티브 상태인 프로세서들로 구성된 제 1 그룹과 인액티브 상태인 프로세서들로 구성된 제 2 그룹으로 분류하고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 제 1 그룹에 할당된 인터럽트 요청 신호들에 대해서는 기 설정된 간격으로 조절하지 않고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 제 2 그룹에 할당된 인터럽트 요청 신호들에 대해서만 기 설정된 간격으로 조절할 수 있다. 다만, 이에 대해서는 상술한 바 있으므로, 그에 대한 중복되는 설명은 생략하기로 한다. 이와 같이, 인터럽트 스프레드 장치(100)는 복수의 인터럽트 소스들이 짧은 시간 내에 복수의 인터럽트들을 연속적으로 출력할 때, 상기 인터럽트들에 기초하여 생성되는 복수의 인터럽트 요청 신호들 사이의 시간 간격을 기 설정된 간격 이상으로 조절함으로써, 인액티브 상태에 있는 복수의 프로세서들(또는, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태로 연속적으로 변경(즉, 급작스러운 웨이크-업)되는 것을 방지할 수 있다. 그 결과, 시스템 온-칩에서 복수의 프로세서들 내부에 급작스러운 웨이크-업에 의한 인러쉬 전류가 생성되는 것이 방지되어, 시스템 온-칩 및 이를 포함하는 전자 기기는 높은 동작 안정성을 확보할 수 있다.

[0082] 도 12는 도 11의 인터럽트 스프레드 장치에 구비된 인터럽트 홀더의 스테이트 머신을 나타내는 도면이다.

[0083] 도 12를 참조하면, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각의 스테이트 머신(200)은 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)이 수신되지 않았음을 나타내는 아이들 스테이트(220), 인터럽트 아비터(140)로부터 출력 허가 신호(AS1, ..., ASm)를 기다리고 있음을 나타내는 웨이트 스테이트(240) 및 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)이 제 1 내지 제 m 프로세서들에 출력되고 있음을 나타내는 어서트 스테이트(260)를 포함할 수 있다. 이 때, 아이들 스테이트(220)는 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각의 디폴트 스테이트(default state)에 해당할 수 있다. 따라서, 아이들 스테이트(220)는 인터럽트가 발생하지 않은 초기 상태를 의미하고, 웨이트 스테이트(240)는 다른 인터럽트로 인한 인러쉬 전류가 사라지기를 기다리는 상태를 의미하며, 어서트 스테이트(260)는 프로세서로 향하는 인터럽트 요청 신호를 갱신하는 상태를 의미하는 것으로 해석할 수 있다.

[0084] 구체적으로, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)이 수신되지 않은 경우에 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 아이들 스테이트(220)에 머물러 있다. 이후, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)이 수신되면, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 웨이트 스테이트(240) 또는 어서트 스테이트(260)에 진입할 수 있다. 예를 들어, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)이 수신되었을 때, 인터럽트 아비터(140)가 출력 허가 신호(AS1, ..., ASm)를 바로 송신하는 경우, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 어서트 스테이트(260)에 바로 진입(RA)할 수 있다. 그러나, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)이 수신되었을 때, 인터럽트 아비터(140)가 이전 인터럽트 요청 신호와의 시간 간격을 기 설정된 간격을 조절하기 위하여 출력 허가 신호(AS1, ..., ASm)를 송신하지 않는 경우, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 웨이트 스테이트(240)에 진입(RNA)할 수 있다. 이후, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 웨이트 스테이트(240)에 머물러 있는 중에 인터럽트 아비터(140)로부터 출력 허가 신호(AS1, ..., ASm)를 수신하면, 어서트 스테이트(260)로 진입(AR)할 수 있다. 나아가, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 어서트 스테이트(260)에 머물러 있는 중에, 제 1 내지 제 m 인터럽트 요청

신호들(nIRQ_01, ..., nIRQ_0m)의 출력이 완료되면, 아이들 스테이트(220)로 진입(AL)할 수 있다. 이와 같이, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 각각은 간단한 구조의 스테이트 머신(200)으로 구현됨으로써, 소형으로 제조되면서도 저전력으로 동작할 수 있다.

[0085] 도 13은 도 11의 인터럽트 스프레드 장치에 구비된 인터럽트 아비터의 스테이트 머신을 나타내는 도면이다.

[0086] 도 13을 참조하면, 인터럽트 아비터(140)의 스테이트 머신(300)은 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_01, ..., nIRQ_0m) 중에서 어느 하나도 제 1 내지 제 m 프로세서들에 출력되고 있지 않음을 나타내는 아이들 스테이트(320) 및 제 1 내지 제 n 인터럽트 요청 신호들(nIRQ_01, ..., nIRQ_0m) 중에서 적어도 하나 이상이 제 1 내지 제 m 프로세서들에 출력되고 있음을 나타내는 웨이트 스테이트(340)를 포함할 수 있다. 이 때, 아이들 스테이트(320)는 인터럽트 아비터(140)의 디폴트 스테이트에 해당할 수 있다. 따라서, 아이들 스테이트(320)는 어떠한 인터럽트도 존재하지 않은 상태를 의미하고, 웨이트 스테이트(340)는 어떠한 인터럽트가 발생하여 이를 프로세서에 전달하였고, 그로 인한 인러쉬 전류가 사라지기를 기다리는 상태를 의미하는 것으로 해석될 수 있다.

[0087] 구체적으로, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m)이 출력 요청 신호(RS1, ..., RSm)를 송신하지 않으면, 인터럽트 아비터(140)는 아이들 스테이트(320)에 머물러 있다. 이후, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 중에서 어느 하나가 출력 요청 신호(RS1, ..., RSm)를 송신하면, 인터럽트 아비터(140)는 웨이트 스테이트(340)에 진입(REQ)하고, 해당 인터럽트 홀더에 출력 허가 신호(AS1, ..., ASm)를 출력할 수 있다. 이후, 인터럽트 아비터(140)는 기 설정된 간격만큼 웨이트 스테이트(340)에 머물러 있은 후, 아이들 스테이트(320)로 진입(EXP)할 수 있다. 이 때, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 중에서 어느 하나가 출력 요청 신호(RS1, ..., RSm)를 송신하여, 인터럽트 아비터(140)가 웨이트 스테이트(340)에 머물러 있는 경우에, 제 1 내지 제 m 인터럽트 홀더들(120_1, ..., 120_m) 중에서 다른 하나가 출력 요청 신호(RS1, ..., RSm)를 송신하더라도, 인터럽트 아비터(140)는 해당 인터럽트 홀더에 출력 허가 신호(AS1, ..., ASm)를 송신하지 않는다. 따라서, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_01, ..., nIRQ_0m)이 기 설정된 간격 이상을 두어 제 1 내지 제 m 프로세서들에 각각 출력될 수 있는 것이다. 이와 같이, 인터럽트 아비터(140)도 간단한 구조의 스테이트 머신(300)으로 구현됨으로써, 소형으로 제조되면서도 저전력으로 동작할 수 있다.

[0088] 도 14는 도 11의 인터럽트 스프레드 장치가 동작하는 일 예를 나타내는 타이밍도이다.

[0089] 도 14를 참조하면, 인터럽트 스프레드 장치(100)에서 제 i 인터럽트 요청 신호(nIRQ_Ii)가 제 i 인터럽트 홀더(120_i)에 입력되면, 제 i 인터럽트 홀더(120_i)는 인터럽트 아비터(140)에 출력 요청 신호(RSi, req)를 출력할 수 있다. 이 때, 인터럽트 아비터(140)는 이전 인터럽트 요청 신호가 없어 아이들 스테이트(IDLE)에 머무르고 있었으므로, 제 i 인터럽트 홀더(120_i)에 바로 출력 허가 신호(ASi, ack)를 출력할 수 있다. 그 결과, 제 i 인터럽트 홀더(120_i)는 아이들 스테이트(IDLE)에서 어서트 스테이트(ASSERT)로 진입하게 되고, 인터럽트 아비터(140)는 아이들 스테이트(IDLE)에서 웨이트 스테이트(WAIT)로 진입하게 되어, 제 i 인터럽트 요청 신호(nIRQ_0i)가 제 i 프로세서로 출력될 수 있다.

[0090] 한편, 제 i 인터럽트 홀더(120_i)가 어서트 스테이트(ASSERT)로 머물러 있는 동안에, 제 j 인터럽트 요청 신호(nIRQ_Ij)가 제 j 인터럽트 홀더(120_j)에 입력되면, 제 j 인터럽트 홀더(120_j)는 인터럽트 아비터(140)에 출력 요청 신호(RSj, req)를 출력할 수 있다. 이 때, 인터럽트 아비터(140)는 웨이트 스테이트(WAIT)에 머무르고 있으므로, 제 j 인터럽트 홀더(120_j)에 출력 허가 신호(ASj, ack)를 출력하지 않는다. 인터럽트 아비터(140)가 웨이트 스테이트(WAIT)에 머무르는 시간은 미리 정해져 있으며(예를 들어, 물리적인 특성 값에 의해 결정), 카운터 등과 같은 회로 등을 이용하여 상기 시간을 조절할 수 있다. 이에, 제 j 인터럽트 홀더(120_j)는 아이들 스테이트(IDLE)에서 어서트 스테이트(ASSERT)가 아닌 웨이트 스테이트(WAIT)로 진입할 수 있다. 이후, 인터럽트 아비터(140)는 제 i 인터럽트 요청 신호(nIRQ_0i)와 제 j 인터럽트 요청 신호(nIRQ_0j)가 기 설정된 간격을 가지게 되는 시점에 웨이트 스테이트(WAIT)에서 아이들 스테이트(IDLE)로 진입하고, 제 j 인터럽트 홀더(120_j)에 출력 허가 신호(ASj, ack)를 출력할 수 있다. 따라서, 제 j 인터럽트 홀더(120_j)는 상기 출력 허가 신호(ASj, ack)에 기초하여 웨이트 스테이트(WAIT)에서 어서트 스테이트(ASSERT)로 진입할 수 있고, 제 j 인터럽트 요청 신호(nIRQ_0j)는 제 j 프로세서에 출력될 수 있다. 그 결과, 인터럽트 스프레드 장치(100)에 의하여 제 i 인터럽트 요청 신호(nIRQ_0i)와 제 j 인터럽트 요청 신호(nIRQ_0j)는 기 설정된 간격을 두어 각각 제 i 프로세서와 제 j 프로세서에 각각 출력될 수 있다. 이와 같이, 인터럽트 스프레드 장치(100)는 짧은 시간 내에 복수의 인터럽트 소스들에서 복수의 인터럽트들을 생성되더라도, 상기 인터럽트들에 기초하여 생성되는 복수의 인터럽트 요청 신호들 사이의 각각의 시간 간격을 기 설정된 간격 이상으로 조절함으로써, 인액티브 상태에 있는 복수의 프

로세서들(또는, 멀티 코어 프로세서의 복수의 코어들)이 짧은 시간 내에 액티브 상태로 연속적으로 변경(즉, 급작스러운 웨이크-업)되는 것을 방지할 수 있다.

[0091] 도 15는 본 발명의 일 실시예에 따른 시스템 온-칩을 나타내는 블록도이다.

[0092] 도 15를 참조하면, 시스템 온-칩(500)은 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n), 인터럽트 컨트롤러(540), 인터럽트 스프레드 장치(560) 및 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)을 포함할 수 있다. 일반적으로, 시스템 온-칩(500)에서 프로세서들(580_1, ..., 580_m)의 개수보다 인터럽트 소스들(520_1, ..., 520_n)의 개수가 많으므로, n이 m보다 큰 것이 일반적이나 그에 한정되지는 않는다. 한편, 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)이라고 명명하였지만, 제 1 내지 제 m 프로세서들(580_1, ..., 580_m) 각각은 독립적인 프로세서들에 상응할 수 있고, 멀티 코어 프로세서(예를 들어, 듀얼 코어 프로세서, 쿼드 코어 프로세서 등)의 복수의 코어들에 상응할 수도 있다.

[0093] 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)은 각각 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 생성할 수 있다. 이 때, 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)은 각각 멀티 프로세서 시스템(또는, 멀티 코어 시스템) 내에서 소정의 동작을 수행하는 아이피(IP)들로서, 시스템 온-칩을 구성하는 여러 구성 요소들 예를 들어, 비디오 모듈, 사운드 모듈, 디스플레이 모듈, 메모리 모듈, 통신 모듈, 카메라 모듈 등과 같은 소정의 모듈들에 상응할 수 있다. 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)이 각각 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 발생시키면, 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)이 생성되고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)은 기 설정된 간격 이상으로 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 각각 입력될 수 있다. 그 결과, 제 1 내지 제 m 프로세서들(580_1, ..., 580_m) 각각은 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)에 응답하여 상기 제 1 내지 제 n 소스들을 위한 인터럽트 처리를 수행할 수 있다.

[0094] 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)에서 생성된 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)을 생성할 수 있다. 일 실시예에서, 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)로부터 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 입력받은 후, 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)이 각각 수행되는 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)을 결정할 수 있다. 즉, 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 할당시킬 수 있다. 일반적으로, 시스템 온-칩(500)에서 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)의 개수는 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)의 개수보다 작기 때문에, 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 시간적으로 및/또는 공간적으로 적절하게 할당할 수 있다.

[0095] 인터럽트 스프레드 장치(560)는 인터럽트 컨트롤러(540)로부터 입력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격을 기 설정된 간격 이상으로 조절할 수 있다. 이 때, 상기 기 설정된 간격은 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)이 인액티브 상태에서 액티브 상태로 변경될 때, 제 1 내지 제 m 프로세서들(580_1, ..., 580_m) 내부에 인러쉬 전류가 생성되지 않는 범위 내에서 결정될 수 있다. 일 실시예에서, 인터럽트 스프레드 장치(560)는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)을 수신하고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)을 기 설정된 간격 이상을 두어 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 각각 출력하는 제 1 내지 제 m 인터럽트 홀더들, 및 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격이 기 설정된 간격보다 작은 경우 상기 시간 간격을 기 설정된 간격으로 조절하는 인터럽트 아비터를 포함할 수 있다. 즉, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)은 제 1 내지 제 m 인터럽트 홀더들에 각각 수신되고, 제 1 내지 제 m 인터럽트 홀더들은 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 각각 연결될 수 있다. 일 실시예에서, 인터럽트 홀더들 및 인터럽트 아비터는 하나의 클럭 도메인(clock domain) 상에서 동작하며, 인터럽트 홀더들과 인터럽트 아비터 사이에 요청/응답 핸드쉐이킹 동작(request/acknowledge handshaking operation)이 수행될 수 있다.

[0096] 구체적으로, 인터럽트 스프레드 장치(560)는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 사이의 각각의 시간 간격을 기 설정된 간격으로 조절함에 있어서, 인접하는 제 k(단, k는 1이상인 정수) 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 0보다 큰 경우, 상기 제 k 시간 간격이 기 설

정된 간격으로 될 때까지 제 k+1 인터럽트 요청 신호의 출력을 지연시킬 수 있다. 또한, 인터럽트 스프레드 장치(560)는 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 사이의 제 k 시간 간격이 0인 경우(즉, 동시에 입력되는 경우), 기 설정된 우선순위에 따라 제 k 인터럽트 요청 신호와 제 k+1 인터럽트 요청 신호 중에서 하나의 출력을 기 설정된 간격만큼 지연시킬 수 있다. 나아가, 인터럽트 스프레드 장치(560)는 기 설정된 우선순위에 따라 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)의 입력 순서와 다르게, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)의 출력 순서를 변경할 수도 있다. 예를 들어, 인터럽트 스프레드 장치(560)는 제 k 인터럽트 요청 신호, 제 k+1 인터럽트 요청 신호 및 제 k+2 인터럽트 요청 신호를 순차적으로 입력받은 경우에도, 제 k 인터럽트 요청 신호, 제 k+1 인터럽트 요청 신호 및 제 k+2 인터럽트 요청 신호를 기 설정된 우선순위에 기초하여 비순차적으로 출력시킬 수 있다.

[0097] 실시예에 따라, 인터럽트 스프레드 장치(560)는 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)을 액티브 상태인 프로세서들로 구성된 제 1 그룹과 인액티브 상태인 프로세서들로 구성된 제 2 그룹으로 분류하고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 액티브 상태인 프로세서들로 구성된 제 1 그룹에 할당된 인터럽트 요청 신호들에 대해서는 기 설정된 간격으로 조절하지 않고, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 인액티브 상태인 프로세서들로 구성된 제 2 그룹에 할당된 인터럽트 요청 신호들에 대해서만 기 설정된 간격으로 조절할 수 있다. 이와 같이, 시스템 온-칩(500)은 인터럽트 스프레드 장치(560)를 구비함으로써, 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)로부터 출력되는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 생성되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 사이의 각각의 시간 간격을 기 설정된 간격 이상으로 조절함으로써, 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)이 짧은 시간 내에 액티브 상태로 연속적으로 변경되는 것(즉, 급작스러운 웨이크-업)을 방지할 수 있다. 그 결과, 시스템 온-칩(500)은 복수의 프로세서들(또는, 멀티 코어 프로세서의 복수의 코어들) 내부에 급작스러운 웨이크-업에 의한 인러쉬 전류가 생성되는 것이 방지할 수 있어 높은 동작 안정성을 확보할 수 있다.

[0098] 도 16은 도 15의 시스템 온-칩에서 인터럽트 스프레드 장치가 동작하는 일 예를 나타내는 도면이다.

[0099] 도 16을 참조하면, 도 16은 시스템 온-칩(500) 내에서 인터럽트 스프레드 장치(560) 내의 제 1 내지 제 m 인터럽트 홀더들(562_1, ..., 562_m)이 인터럽트 아비터(미도시)에 의하여 제어됨으로써, 제 1 내지 제 m 인터럽트 홀더들(562_1, ..., 562_m)에 입력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ... nIRQ_Im)이 기 설정된 간격을 두어 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 출력되는 일 예를 보여주고 있다. 다만, 설명의 편의를 위하여, 도 16에서는 m이 4라고 가정한다.

[0100] 구체적으로, 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)이 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 출력하면, 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)을 생성할 수 있다. 도 16에 도시된 바와 같이, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)은 제 1 인터럽트 요청 신호(nIRQ_I1), 제 3 인터럽트 요청 신호(nIRQ_I3), 제 2 인터럽트 요청 신호(nIRQ_I2), 제 4 인터럽트 요청 신호(nIRQ_I4) 순으로 입력될 수 있다. 이 때, 제 1 인터럽트 홀더(562_1)는 제 1 인터럽트 요청 신호(nIRQ_I1)보다 이전의 인터럽트 요청 신호가 없기 때문에 제 1 인터럽트 요청 신호(nIRQ_O1)를 바로 제 1 프로세서(580_1)에 출력할 수 있다. 이후, 제 3 인터럽트 홀더(562_3)는 제 1 인터럽트 요청 신호(nIRQ_I1)와 제 3 인터럽트 요청 신호(nIRQ_I3) 사이의 시간 간격이 기 설정된 간격(PS)보다 작으므로, 제 3 인터럽트 요청 신호(nIRQ_I3)를 소정의 시간(DL1)만큼 지연시킨 후, 제 3 인터럽트 요청 신호(nIRQ_O3)를 제 3 프로세서(580_3)에 출력할 수 있다. 이후, 제 2 인터럽트 홀더(562_2)는 제 3 인터럽트 요청 신호(nIRQ_I3)와 제 2 인터럽트 요청 신호(nIRQ_I2) 사이의 시간 간격이 기 설정된 간격(PS)보다 작으므로, 제 2 인터럽트 요청 신호(nIRQ_I2)를 소정의 시간(DL2)만큼 지연시킨 후, 제 2 인터럽트 요청 신호(nIRQ_O2)를 제 2 프로세서(580_2)에 출력할 수 있다. 이후, 제 4 인터럽트 홀더(562_4)는 제 2 인터럽트 요청 신호(nIRQ_I2)와 제 4 인터럽트 요청 신호(nIRQ_I4) 사이의 시간 간격이 기 설정된 간격(PS)보다 작으므로, 제 4 인터럽트 요청 신호(nIRQ_I4)를 소정의 시간(DL3)만큼 지연시킨 후, 제 4 인터럽트 요청 신호(nIRQ_O4)를 제 4 프로세서(580_4)에 출력할 수 있다. 그 결과, 제 1 내지 제 4 프로세서들(580_1, ..., 580_4)은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_O1, ... nIRQ_O4)을 기 설정된 간격(PS)을 두어 수신할 수 있으므로, 제 1 내지 제 4 프로세서들(580_1, ..., 580_4) 내부에 급작스러운 웨이크-업에 의한 인러쉬 전류가 생성되는 것을 방지할 수 있다.

[0101] 도 17은 도 15의 시스템 온-칩에서 인터럽트 스프레드 장치가 동작하는 다른 예를 나타내는 도면이다.

[0102] 도 17을 참조하면, 도 17은 시스템 온-칩(500) 내에서 인터럽트 스프레드 장치(560) 내의 제 1 내지 제 m 인터럽트 홀더들(562_1, ..., 562_m)이 인터럽트 아비터(미도시)에 의하여 제어됨으로써, 제 1 내지 제 m 인터럽트 홀더들(562_1, ..., 562_m)에 입력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ... nIRQ_Im)이 기 설정된 간격을 두어 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 출력되는 다른 예를 보여주고 있다. 다만, 설명의 편의를 위하여, 도 17에서는 m이 4라고 가정한다.

[0103] 구체적으로, 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)이 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 출력하면, 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)을 생성할 수 있다. 도 17에 도시된 바와 같이, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)은 제 1 인터럽트 요청 신호(nIRQ_I1), 제 2 인터럽트 요청 신호(nIRQ_I2), 제 4 인터럽트 요청 신호(nIRQ_I4), 제 3 인터럽트 요청 신호(nIRQ_I3) 순으로 입력될 수 있다. 이 때, 제 1 인터럽트 홀더(562_1)는 제 1 인터럽트 요청 신호(nIRQ_I1)보다 이전의 인터럽트 요청 신호가 없기 때문에 제 1 인터럽트 요청 신호(nIRQ_01)를 바로 제 1 프로세서(580_1)에 출력할 수 있다. 이후, 제 2 인터럽트 홀더(562_2)는 제 1 인터럽트 요청 신호(nIRQ_I1)와 제 2 인터럽트 요청 신호(nIRQ_I2) 사이의 시간 간격이 기 설정된 간격(PS)보다 크므로, 제 2 인터럽트 요청 신호(nIRQ_02)를 바로 제 2 프로세서(580_2)에 출력할 수 있다. 이후, 제 4 인터럽트 홀더(562_4)는 제 2 인터럽트 요청 신호(nIRQ_I2)와 제 4 인터럽트 요청 신호(nIRQ_I4) 사이의 시간 간격이 기 설정된 간격(PS)보다 작으므로, 제 4 인터럽트 요청 신호(nIRQ_I4)를 소정의 시간(DL1)만큼 지연시킨 후, 제 4 인터럽트 요청 신호(nIRQ_04)를 제 4 프로세서(580_4)에 출력할 수 있다. 이후, 제 3 인터럽트 홀더(562_3)는 제 4 인터럽트 요청 신호(nIRQ_I4)와 제 3 인터럽트 요청 신호(nIRQ_I4) 사이의 시간 간격이 기 설정된 간격(PS)보다 작으므로, 제 3 인터럽트 요청 신호(nIRQ_I3)를 소정의 시간(DL2)만큼 지연시킨 후, 제 3 인터럽트 요청 신호(nIRQ_03)를 제 3 프로세서(580_3)에 출력할 수 있다. 그 결과, 제 1 내지 제 4 프로세서들(580_1, ..., 580_4)은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_01, ... nIRQ_0m)을 기 설정된 간격 이상을 두어 수신할 수 있으므로, 제 1 내지 제 4 프로세서들(580_1, ..., 580_4) 내부에 급작스러운 웨이크-업에 의한 인러쉬 전류가 생성되는 것을 방지할 수 있다.

[0104] 도 18은 도 15의 시스템 온-칩에서 인터럽트 스프레드 장치가 동작하는 또 다른 예를 나타내는 도면이다.

[0105] 도 18을 참조하면, 도 18은 시스템 온-칩(500) 내에서 인터럽트 스프레드 장치(560) 내의 제 1 내지 제 m 인터럽트 홀더들(562_1, ..., 562_m)이 인터럽트 아비터(미도시)에 의하여 제어됨으로써, 제 1 내지 제 m 인터럽트 홀더들(562_1, ..., 562_m)에 입력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ... nIRQ_Im)이 기 설정된 간격을 두어 제 1 내지 제 m 프로세서들(580_1, ..., 580_m)에 출력되는 또 다른 예를 보여주고 있다. 다만, 설명의 편의를 위하여, 도 18에서는 m이 4라고 가정한다.

[0106] 구체적으로, 제 1 내지 제 n 인터럽트 소스들(520_1, ..., 520_n)이 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 출력하면, 인터럽트 컨트롤러(540)는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)을 생성할 수 있다. 도 18에 도시된 바와 같이, 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_I4)은 제 1 인터럽트 요청 신호(nIRQ_I1)와 제 2 인터럽트 요청 신호(nIRQ_I2)가 동시에 입력된 이후, 제 4 인터럽트 요청 신호(nIRQ_I4)와 제 3 인터럽트 요청 신호(nIRQ_I3)가 순차적으로 입력될 수 있다. 이 때, 제 1 인터럽트 요청 신호(nIRQ_I1)와 제 2 인터럽트 요청 신호(nIRQ_I2)가 동시에 입력되기 때문에, 제 1 인터럽트 홀더(562_1)는 제 2 인터럽트 요청 신호(nIRQ_02)보다 우선순위가 높은 제 1 인터럽트 요청 신호(nIRQ_01)를 바로 제 1 프로세서(580_1)에 출력할 수 있다. 반면에, 제 2 인터럽트 홀더(562_2)는 제 1 인터럽트 요청 신호(nIRQ_I1)보다 우선순위가 낮은 제 2 인터럽트 요청 신호(nIRQ_I2)를 기 설정된 간격만큼 지연(DL1)시킨 후, 제 2 인터럽트 요청 신호(nIRQ_02)를 제 2 프로세서(580_2)에 출력할 수 있다. 이후, 제 4 인터럽트 요청 신호(nIRQ_I4)와 제 3 인터럽트 요청 신호(nIRQ_I3)가 순차적으로 입력되지만, 제 4 인터럽트 요청 신호(nIRQ_I4)와 제 3 인터럽트 요청 신호(nIRQ_I3)는 기 설정된 우선순위에 기초하여 비순차적으로 출력될 수 있다. 예를 들어, 우선순위가 높은 제 3 인터럽트 요청 신호(nIRQ_03)가 우선순위가 낮은 제 4 인터럽트 요청 신호(nIRQ_04)보다 먼저 출력될 수 있다. 즉, 제 3 인터럽트 홀더(562_3)가 제 2 인터럽트 요청 신호(nIRQ_I2)와 제 3 인터럽트 요청 신호(nIRQ_I3) 사이의 시간 간격이 기 설정된 간격(PS)보다 작으므로, 제 3 인터럽트 요청 신호(nIRQ_I3)를 소정의 시간(DL2)만큼 지연시킨 후, 제 3 인터럽트 요청 신호(nIRQ_03)를 제 3 프로세서(580_3)에 출력할 수 있다. 이후, 제 4 인터럽트 홀더(562_4)는 제 4 인터럽트 요청 신호(nIRQ_I4)를 소정의 시간(DL3)만큼 지연시킨 후, 제 4 인터럽트 요청 신호(nIRQ_04)를 제 4 프로세서(580_4)에 출력할 수 있다. 그 결과, 제 1 내지 제 4 프로세서들(580_1, ..., 580_4)은 제 1 내지 제 4 인터럽트 요청 신호들(nIRQ_01, ... nIRQ_0m)을 기 설정된 간격을 두어 수신할 수 있으므로, 제 1 내지

제 4 프로세서들(580_1, ..., 580_4) 내부에 급작스러운 웨이크-업에 의한 인러쉬 전류가 생성되는 것을 방지할 수 있다.

[0107] 도 19는 임의의 시스템 온-칩이 인터럽트 스프레드 장치를 구비하고 있는지 여부를 검증하는 방법의 일 예를 나타내는 순서도이고, 도 20은 임의의 시스템 온-칩이 인터럽트 스프레드 장치를 구비하고 있는지 여부를 검증하는 방법의 일 예를 설명하기 위한 도면이다.

[0108] 도 19 및 도 20을 참조하면, 도 19의 검증하는 방법은 복수의 인터럽트들에 상응하는 복수의 인터럽트 요청 신호들(IAT)을 제 1 시간 간격(ATP)으로 복수의 프로세서들에 순차적으로 인가(Step S510)하고, 상기 프로세서들이 웨이크-업(TOT)되는 제 2 시간 간격(ETP)을 검출(Step S520)하며, 상기 제 1 시간 간격(ATP)이 0이 되었는지 여부를 판단(Step S530)할 수 있다. 이 때, 상기 제 1 시간 간격(ATP)이 0이 되지 않은 경우, 도 19의 검증하는 방법은 상기 제 1 시간 간격(ATP)을 감소(Step S540)시킨 후, 상기 단계들(Step S510, Step S520, Step S530)을 반복할 수 있다. 반면에, 상기 제 1 시간 간격(ATP)이 0이 된 경우, 도 19의 검증하는 방법은 상기 제 2 시간 간격(ETP)이 일정한 값으로 수렴했는지 여부를 판단(Step S550)할 수 있다. 그 결과, 상기 제 2 시간 간격(ETP)이 일정한 값으로 수렴한 경우, 도 19의 검증하는 방법은 임의의 시스템 온-칩에 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용된 것으로 판단(Step S560)할 수 있고, 상기 제 2 시간 간격(ETP)이 일정한 값으로 수렴하지 않고 계속적으로 작아진 경우, 도 19의 검증하는 방법은 임의의 시스템 온-칩에 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용되지 않은 것으로 판단(Step S570)할 수 있다. 즉, 상기 제 2 시간 간격(ETP)이 일정한 값으로 수렴한다는 것은, 복수의 인터럽트 요청 신호들(IAT) 사이의 제 1 시간 간격(ATP)이 일정한 값 이하로 작아지더라도, 복수의 인터럽트 요청 신호들(IAT)이 제 2 시간 간격(ETP)(즉, 기 설정된 간격)을 두어 복수의 프로세서들에 입력되고 있는 것을 의미하므로, 임의의 시스템 온-칩에 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용된 것으로 판단될 수 있다. 다만, 도 19의 검증하는 방법은 하나의 예시로서, 요구되는 조건에 따라 다양한 방법으로 실시될 수 있다.

[0109] 도 21은 임의의 시스템 온-칩이 인터럽트 스프레드 장치를 구비하고 있는지 여부를 검증하는 방법의 다른 예를 나타내는 순서도이다.

[0110] 도 21을 참조하면, 도 21의 검증하는 방법은 복수의 인터럽트들에 상응하는 복수의 인터럽트 요청 신호들을 제 1 시간 간격으로 복수의 프로세서들에 순차적으로 인가(Step S610)하고, 공급 전압의 전압 강하(voltage drop)를 측정(Step S620)하며, 상기 제 1 시간 간격이 0이 되었는지 여부를 판단(Step S630)할 수 있다. 이 때, 상기 제 1 시간 간격이 0이 되지 않은 경우, 도 21의 검증하는 방법은 상기 제 1 시간 간격을 감소(Step S640)시킨 후, 상기 단계들(Step S610, Step S620, Step S630)을 반복할 수 있다. 반면에, 상기 제 1 시간 간격이 0이 된 경우, 도 21의 검증하는 방법은 상기 공급 전압의 전압 강하가 일정한 값으로 수렴했는지 여부를 판단(Step S650)할 수 있다. 그 결과, 상기 공급 전압의 전압 강하가 일정한 값으로 수렴한 경우, 도 21의 검증하는 방법은 임의의 시스템 온-칩에 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용된 것으로 판단(Step S660)할 수 있고, 상기 공급 전압의 전압 강하가 일정한 값으로 수렴하지 않고 계속적으로 커진 경우, 도 21의 검증하는 방법은 임의의 시스템 온-칩에 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용되지 않은 것으로 판단(Step S670)할 수 있다. 즉, 상기 공급 전압의 전압 강하가 일정한 값으로 수렴한다는 것은, 복수의 인터럽트 요청 신호들 사이의 제 1 시간 간격이 일정한 값 이하로 작아지더라도, 복수의 인터럽트 요청 신호들이 기 설정된 간격을 두어 복수의 프로세서들에 입력되고 있는 것을 의미하므로, 임의의 시스템 온-칩에 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용된 것으로 판단될 수 있다. 도 21의 검증하는 방법은 도 19의 검증하는 방법과 함께 수행됨으로써, 본 발명의 실시예들에 따른 인터럽트 스프레드 장치 및 방법이 적용 여부에 대하여 보다 정확한 판단을 가능하게 할 수 있다.

[0111] 도 22는 본 발명의 일 실시예에 따른 멀티 코어 시스템을 나타내는 블록도이고, 도 23은 도 21의 멀티 코어 시스템이 스마트폰으로 구현되는 일 예를 나타내는 도면이다.

[0112] 도 22 및 도 23을 참조하면, 멀티 코어 시스템(600)은 제 1 내지 제 n 인터럽트 소스들(610_1, ..., 610_n), 인터럽트 컨트롤러(620), 인터럽트 스프레드 장치(630), 멀티 코어 프로세서(640), 버스 인터페이스(645), 램(Random Access Memory; RAM) 디바이스(650), 롬(Read Only Memory; ROM) 디바이스(660), 스토리지 디바이스(670), 시스템 버스(680) 등을 포함할 수 있다. 이 때, 멀티 코어 프로세서(640)는 제 1 내지 제 m 코어들(P1, ..., Pm)을 포함할 수 있는데, m이 2인 경우 듀얼 코어 프로세서, m이 4인 경우 쿼드 코어 프로세서 등으로 명명될 수 있다. 이 때, 멀티 코어 시스템(600)은 스마트폰(700)에 상응할 수 있으나, 그에 한정되는 것은 아니다. 예를 들어, 멀티 코어 시스템(600)은 스마트패드, 휴대폰 등과 같은 모바일 기기 및 스마트 텔레비전

등과 같은 전자 기기로 구현될 수 있다. 나아가, 멀티 코어 시스템(600)은 복수의 코어들을 가진 멀티 코어 프로세서를 구비하는 시스템뿐 만 아니라, 단일의 코어를 가진 복수의 프로세서들을 구비하는 멀티 프로세서 시스템으로도 해석되어야 한다.

[0113] 제 1 내지 제 n 인터럽트 소스들(610_1, ..., 610_n)은 각각 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)을 생성할 수 있다. 이 때, 제 1 내지 제 n 인터럽트 소스들(610_1, ..., 610_n)은 각각 멀티 코어 시스템(600) 내에서 소정의 동작을 수행하는 아이피들(610_1, ..., 610_n)로서, 시스템 온-칩을 구성하는 여러 구성요소들 예를 들어, 비디오 모듈, 사운드 모듈, 디스플레이 모듈, 메모리 모듈, 통신 모듈, 카메라 모듈 등과 같은 소정의 모듈들에 상응할 수 있다. 인터럽트 컨트롤러(620)는 제 1 내지 제 n 인터럽트 소스들(610_1, ..., 610_n)로부터 출력되는 제 1 내지 제 n 인터럽트들(INT_R1, ..., INT_Rn)에 기초하여 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im)을 생성할 수 있다. 인터럽트 스프레드 장치(630)는 인터럽트 컨트롤러(620)로부터 입력되는 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_I1, ..., nIRQ_Im) 중에서 인접하는 인터럽트 요청 신호들 사이의 시간 간격을 적어도 기 설정된 간격 이상으로 조절할 수 있다. 도 21에 도시된 바와 같이, 제 1 내지 제 m 인터럽트 요청 신호들(nIRQ_O1, ..., nIRQ_Om)은 멀티 코어 프로세서(640) 내부의 제 1 내지 제 m 코어들(P1, ..., Pm)에 각각 출력될 수 있다. 이를 위하여, 인터럽트 스프레드 장치(630)는 제 1 내지 제 m 인터럽트 홀더들 및 상기 제 1 내지 제 m 인터럽트 홀더들을 제어하기 위한 인터럽트 아비터를 포함할 수 있다. 다만, 이에 대해서는 상술한 바 있으므로, 그에 대한 중복되는 설명은 생략하기로 한다.

[0114] 한편, 멀티 코어 프로세서(640)는 버스 인터페이스(645)를 통해 시스템 버스(680)에 연결되어 다른 구성요소들과 데이터를 송수신할 수 있다. 즉, 멀티 코어 프로세서(640)는 어드레스 버스(address bus), 제어 버스(control bus) 및 데이터 버스(data bus) 등과 같은 시스템 버스(680)를 통하여 복수의 아이피들(610_1, ..., 610_n), 램 디바이스(650), 롬 디바이스(660), 스토리지 디바이스(670) 등과 통신을 수행할 수 있다. 이 때, 스토리지 장치(670)는 하드 디스크 드라이브(Hard Disk Drive; HDD), 솔리드 스테이트 드라이브(Solid State Drive; SSD), 레이드(Redundant Array of Independent Disk; RAID) 등일 수 있다. 실시예에 따라, 멀티 코어 프로세서(640)는 주변 구성요소 상호연결(Peripheral Component Interconnect; PCI) 버스와 같은 확장 버스에도 연결될 수 있다. 한편, 도 21에는 도시되지 않았지만, 멀티 코어 시스템(600)은 비휘발성 메모리 장치 및/또는 휘발성 메모리 장치를 더 포함할 수 있다. 이 때, 비휘발성 메모리 장치는 EPROM(Erasable Programmable Read-Only Memory), EEPROM(Electrically Erasable Programmable Read-Only Memory), 플래시 메모리(Flash Memory), PRAM(Phase Change Random Access Memory), RRAM(Resistance Random Access Memory), NFGM(Nano Floating Gate Memory), PoRAM(Polymer Random Access Memory), MRAM(Magnetic Random Access Memory), FRAM(Ferroelectric Random Access Memory) 등으로 구현될 수 있고, 휘발성 메모리 장치는 DRAM(Dynamic Random Access Memory), SRAM(Static Random Access Memory), 모바일 DRAM 등으로 구현될 수 있다.

[0115] 이와 같이, 멀티 코어 시스템(600)은 크게 데이터를 송수신하기 위한 시스템 버스 경로(system bus path)와 인터럽트를 처리하기 위한 인터럽트 경로(interrupt path)를 가질 수 있고, 멀티 코어 프로세서(640)는 상기 시스템 버스 경로 및 인터럽트 경로를 통해 복수의 아이피들(610_1, ..., 610_n)로 하여금 소정의 동작을 수행하게 하고, 상기 아이피들(610_1, ..., 610_n)이 소정의 동작을 수행함에 따른 인터럽트 처리를 수행할 수 있다. 한편, 멀티 코어 시스템(600)은 다양한 형태들의 패키지를 이용하여 실장될 수 있다. 예를 들어, PoP(Package on Package), BGAs(Ball grid arrays), CSPs(Chip scale packages), PLCC(Plastic Leaded Chip Carrier), PDIP(Plastic Dual In-Line Package), Die in Waffle Pack, Die in Wafer Form, COB(Chip On Board), CERDIP(Ceramic Dual In-Line Package), MQFP(Plastic Metric Quad Flat Pack), TQFP(Thin Quad Flat-Pack), SOIC(Small Outline Integrated Circuit), SSOP(Shrink Small Outline Package), TSOP(Thin Small Outline Package), TQFP(Thin Quad Flat-Pack), SIP(System In Package), MCP(Multi Chip Package), WFP(Wafer-level Fabricated Package), WSP(Wafer-Level Processed Stack Package) 등과 같은 패키지들이 이용될 수 있다.

산업상 이용가능성

[0116] 본 발명은 복수의 프로세서들(또는, 멀티 코어 프로세서)을 구비하는 멀티 프로세서 시스템에 적용될 수 있다. 예를 들어, 본 발명은 복수의 프로세서들(또는, 멀티 코어 프로세서)을 구비하는 컴퓨터, 노트북, 휴대폰, 스마트폰, 스마트패드, 보안 시스템 등에 적용될 수 있다.

[0117] 이상에서는 본 발명의 실시예들을 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

[0118] 100: 인터럽트 스프레드 장치 120: 인터럽트 홀더

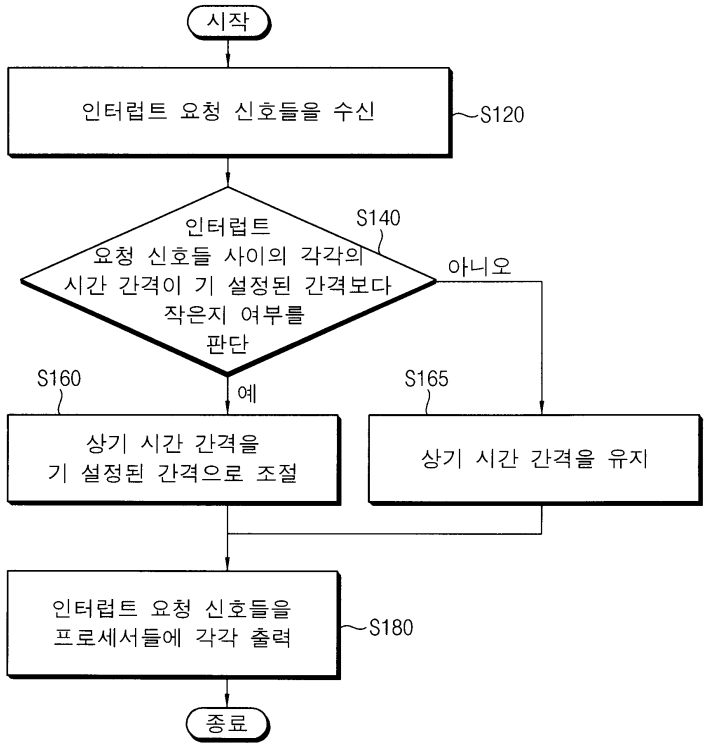
140: 인터럽트 아비터 500: 시스템 온-칩

520: 인터럽트 소스 540: 인터럽트 컨트롤러

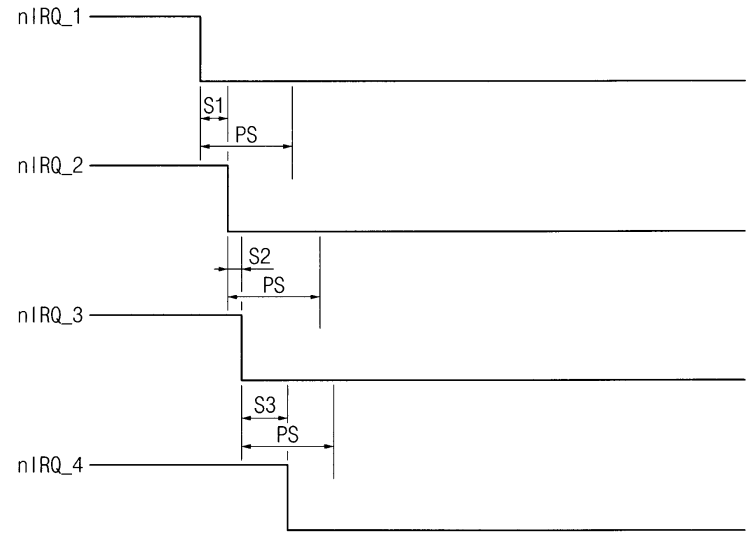
560: 인터럽트 스프레드 장치 580: 프로세서

도면

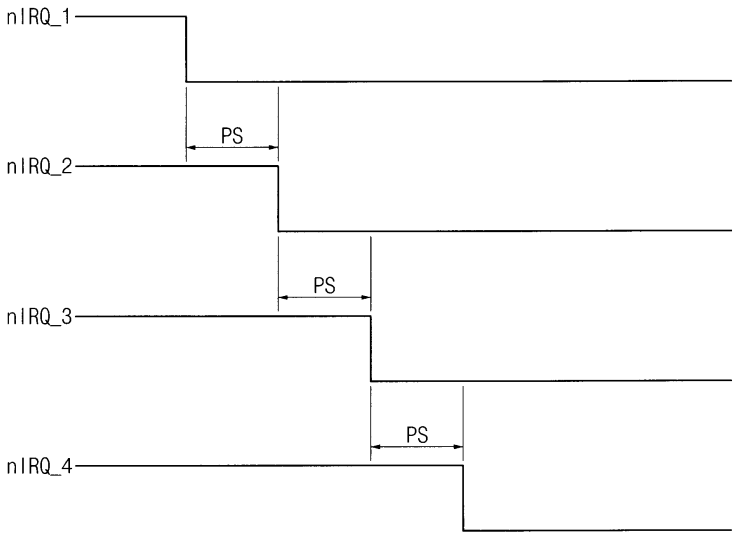
도면1



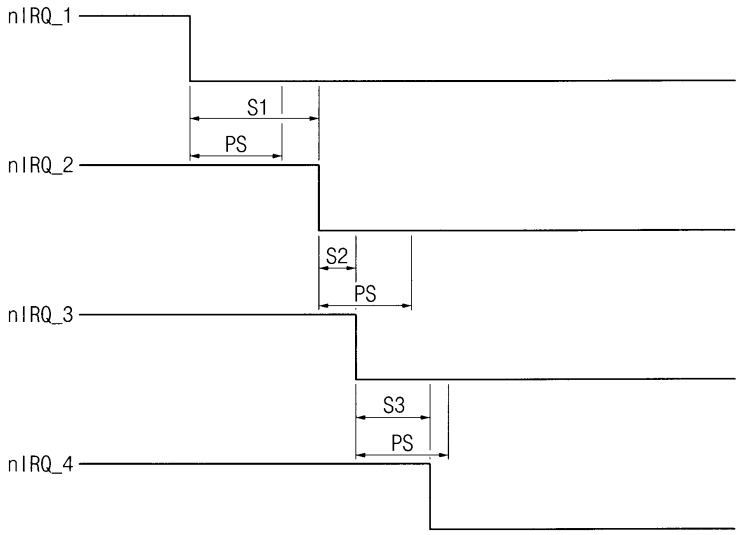
도면2a


nIRQ_1
S1
PS
nIRQ_2
S2
PS
nIRQ_3
S3
PS
nIRQ_4

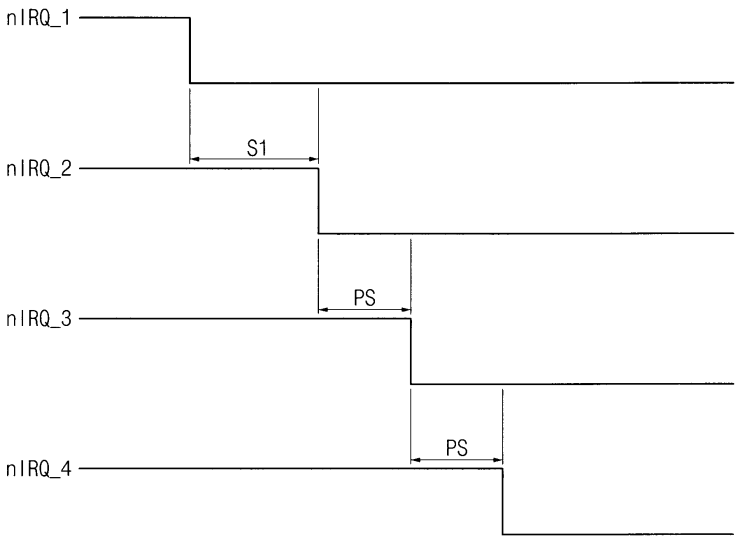
도면2b


nIRQ_1
PS
nIRQ_2
PS
nIRQ_3
PS
nIRQ_4

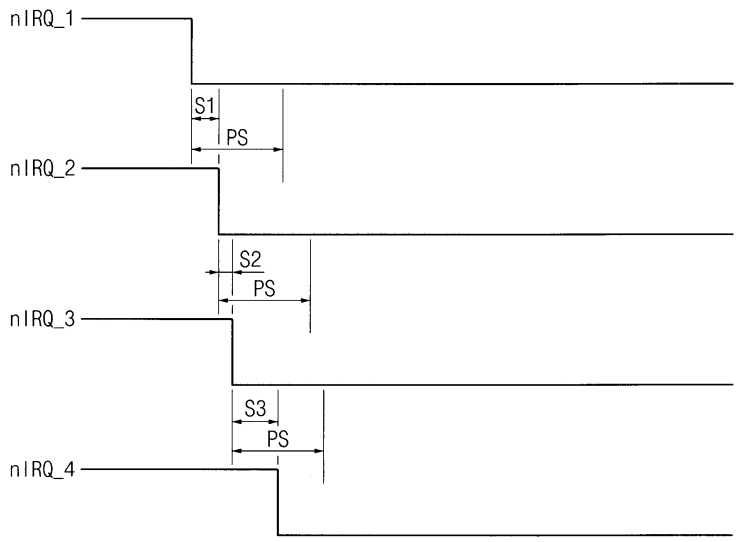
도면3a


nIRQ_1
S1
PS
nIRQ_2
S2
PS
nIRQ_3
S3
PS
nIRQ_4

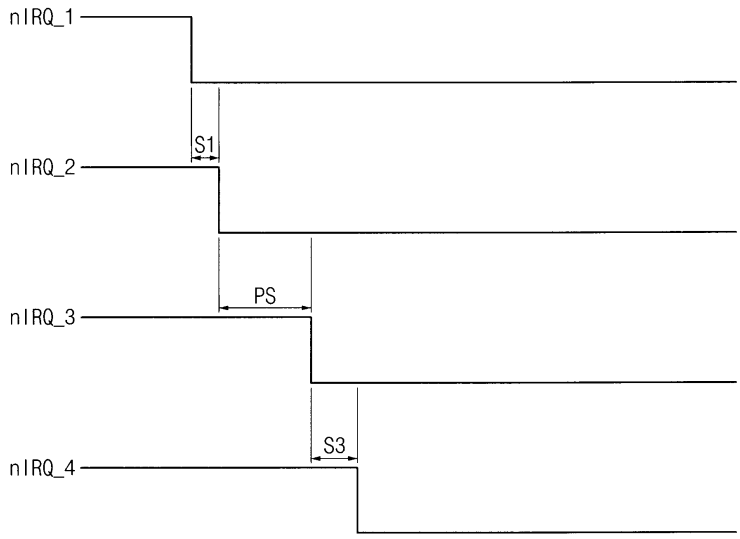
도면3b


nIRQ_1
S1
nIRQ_2
PS
nIRQ_3
PS
nIRQ_4

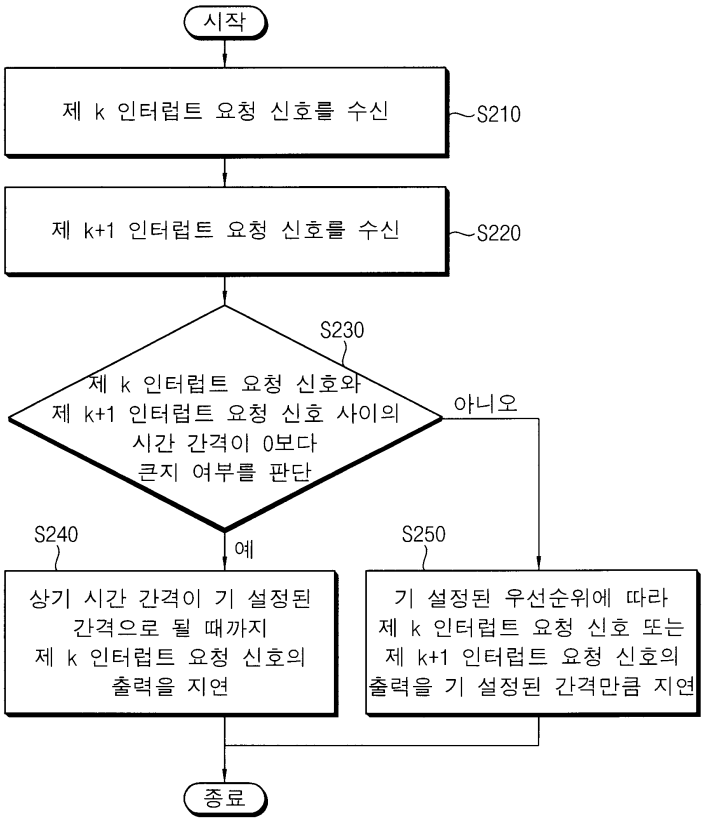
도면4a


nIRQ_1
S1
PS
nIRQ_2
S2
PS
nIRQ_3
S3
PS
nIRQ_4

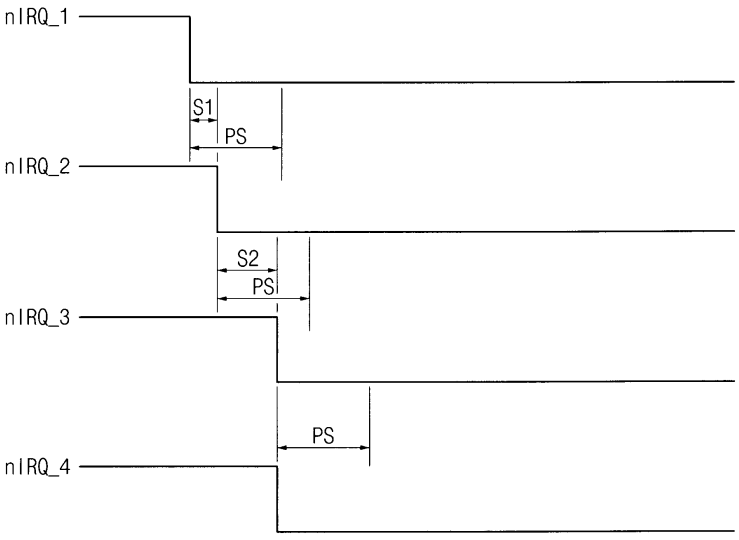
도면4b


nIRQ_1
S1
nIRQ_2
PS
nIRQ_3
S3
nIRQ_4

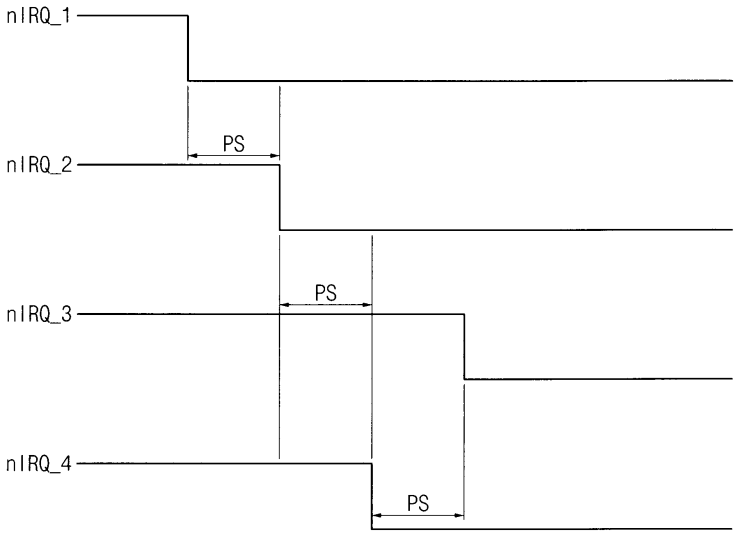
도면5


시작
제 k 인터럽트 요청 신호를 수신
S210
제 k+1 인터럽트 요청 신호를 수신
S220
S230
제 k 인터럽트 요청 신호와
제 k+1 인터럽트 요청 신호 사이의
시간 간격이 0보다
큰지 여부를 판단
아니오
예
S240
상기 시간 간격이 기 설정된
간격으로 될 때까지
제 k 인터럽트 요청 신호의
출력을 지연
S250
기 설정된 우선순위에 따라
제 k 인터럽트 요청 신호 또는
제 k+1 인터럽트 요청 신호의
출력을 기 설정된 간격만큼 지연
종료

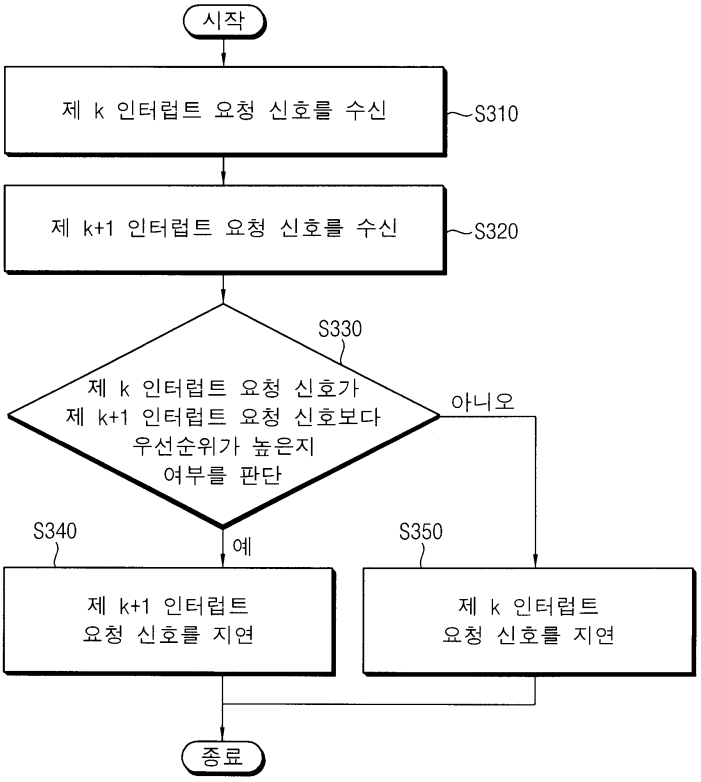
도면6a


nIRQ_1
S1
PS
nIRQ_2
S2
PS
nIRQ_3
PS
nIRQ_4

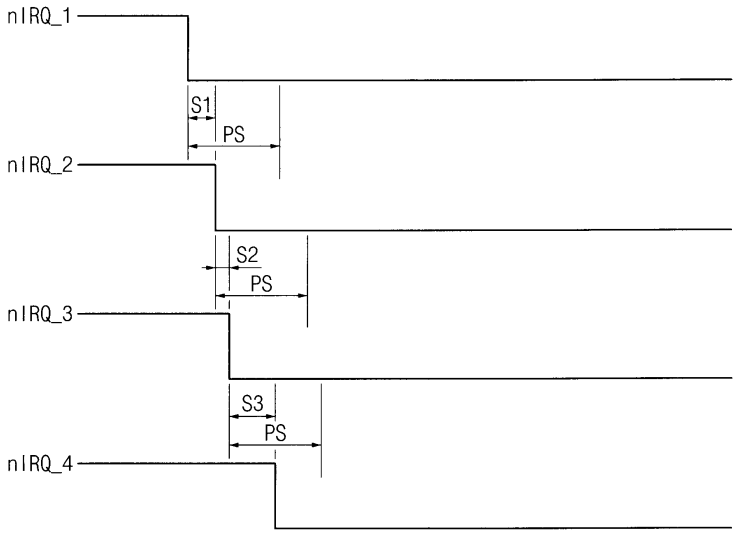
도면6b


nIRQ_1
nIRQ_2
nIRQ_3
nIRQ_4
PS
PS
PS

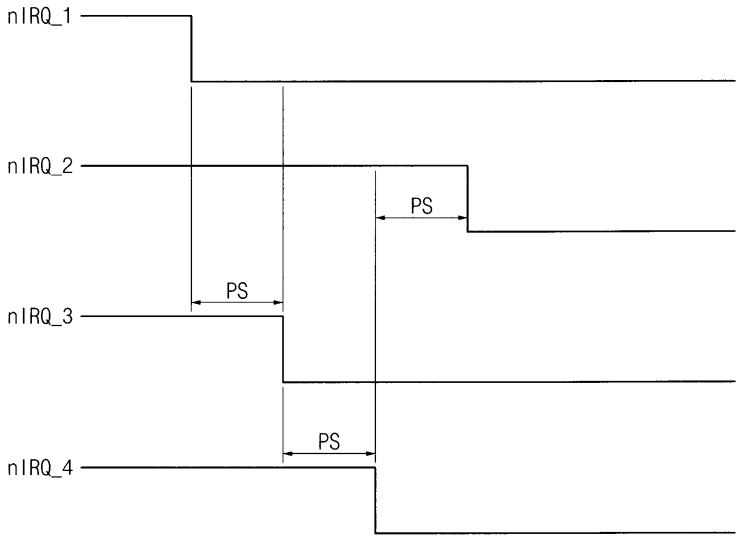
도면7


시작
제 k 인터럽트 요청 신호를 수신
S310
제 k+1 인터럽트 요청 신호를 수신
S320
S330
제 k 인터럽트 요청 신호가
제 k+1 인터럽트 요청 신호보다
우선순위가 높은지
여부를 판단
아니오
예
S340
제 k+1 인터럽트
요청 신호를 지연
S350
제 k 인터럽트
요청 신호를 지연
종료

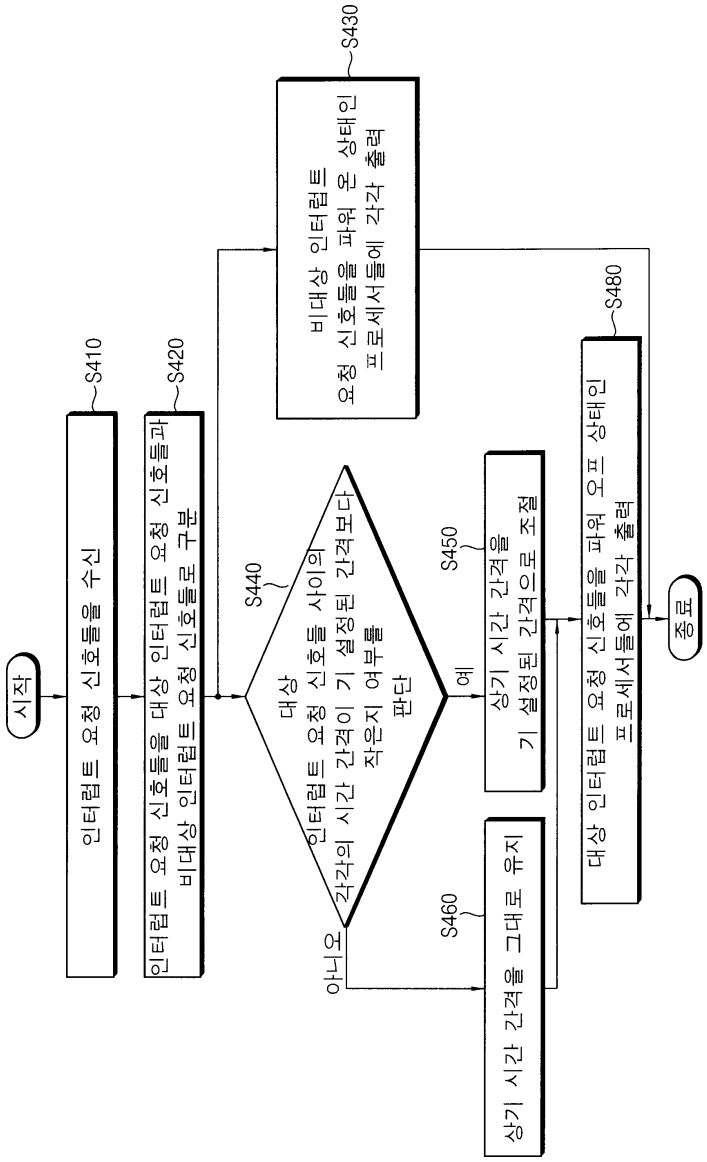
도면8a


nIRQ_1
S1
PS
nIRQ_2
S2
PS
nIRQ_3
S3
PS
nIRQ_4

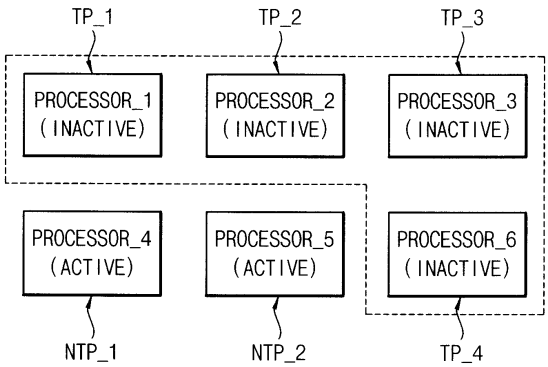
도면8b


nIRQ_1
nIRQ_2
PS
PS
nIRQ_3
PS
nIRQ_4

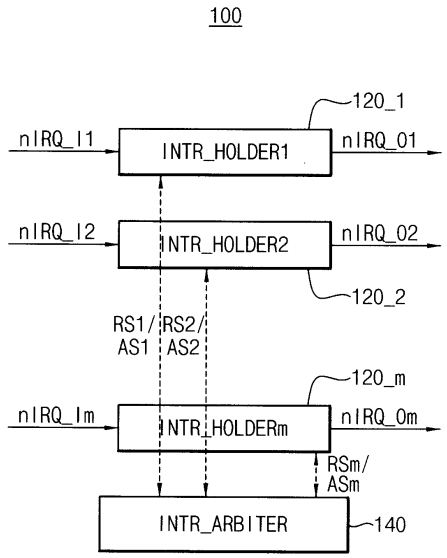
도면9


시작
인터럽트 요청 신호들을 수신
S410
인터럽트 요청 신호들을 대상 인터럽트 요청 신호들과 비대상 인터럽트 요청 신호들로 구분
S420
대상 인터럽트 요청 신호들 사이의 간격이 기 설정된 간격보다 작은지 여부를 판단
S440
비대상 인터럽트 요청 신호들을 파워 온 상태인 프로세서들에 각각 출력
S430
예
S450
상기 시간 간격을 기 설정된 간격으로 조절
아니오
S460
상기 시간 간격을 그대로 유지
대상 인터럽트 요청 신호들을 파워 오프 상태인 프로세서들에 각각 출력
S480
종료

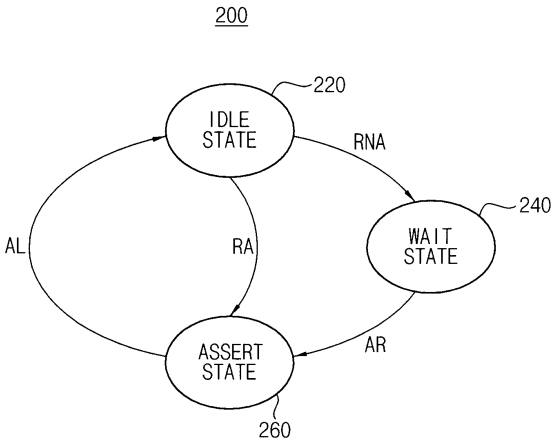
도면10


TP_1
TP_2
TP_3
PROCESSOR_1 (INACTIVE)
PROCESSOR_2 (INACTIVE)
PROCESSOR_3 (INACTIVE)
PROCESSOR_4 (ACTIVE)
PROCESSOR_5 (ACTIVE)
PROCESSOR_6 (INACTIVE)
NTP_1
NTP_2
TP_4

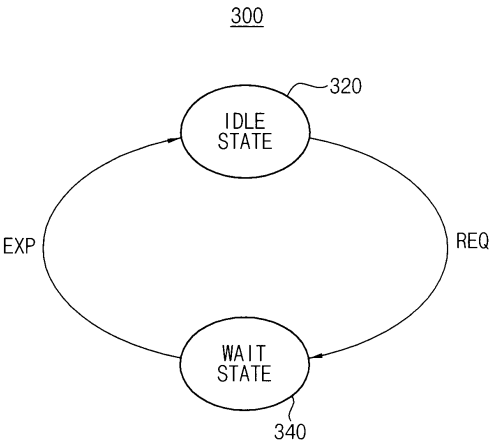
도면11



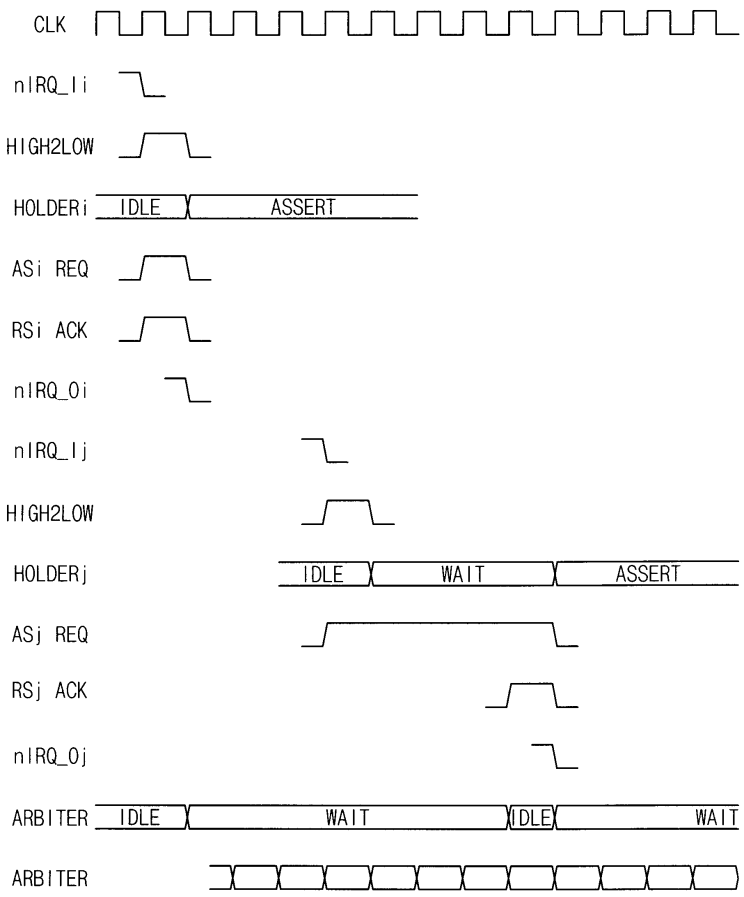
도면12



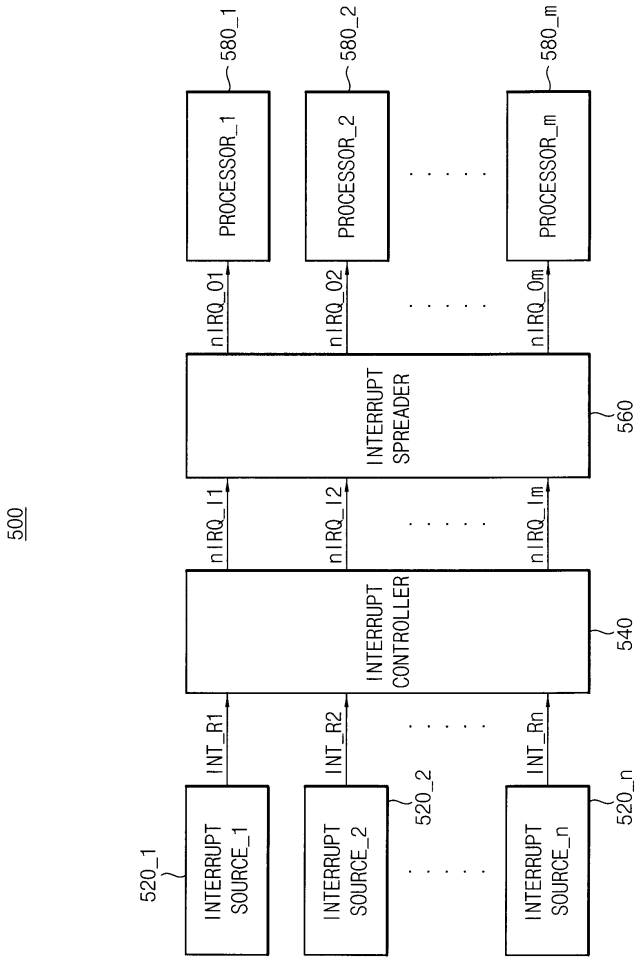
도면13



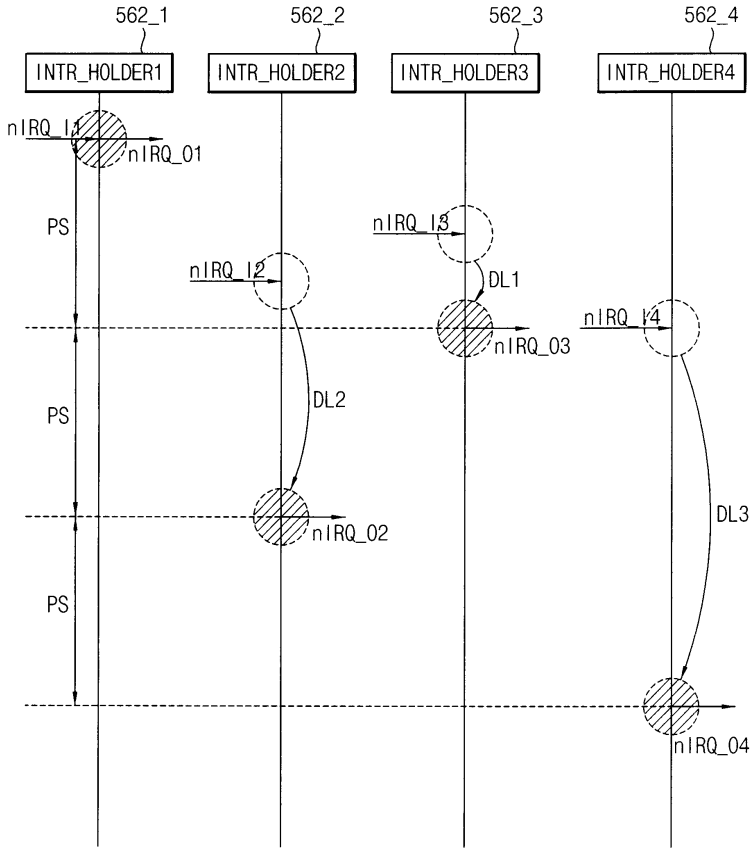
도면14


CLK
nIRQ_Ii
HIGH2LOW
HOLDERi
IDLE
ASSERT
ASi REQ
RSi ACK
nIRQ_Oi
nIRQ_Ij
HIGH2LOW
HOLDERj
IDLE
WAIT
ASSERT
ASj REQ
RSj ACK
nIRQ_Oj
ARBITER
IDLE
WAIT
IDLE
WAIT
ARBITER

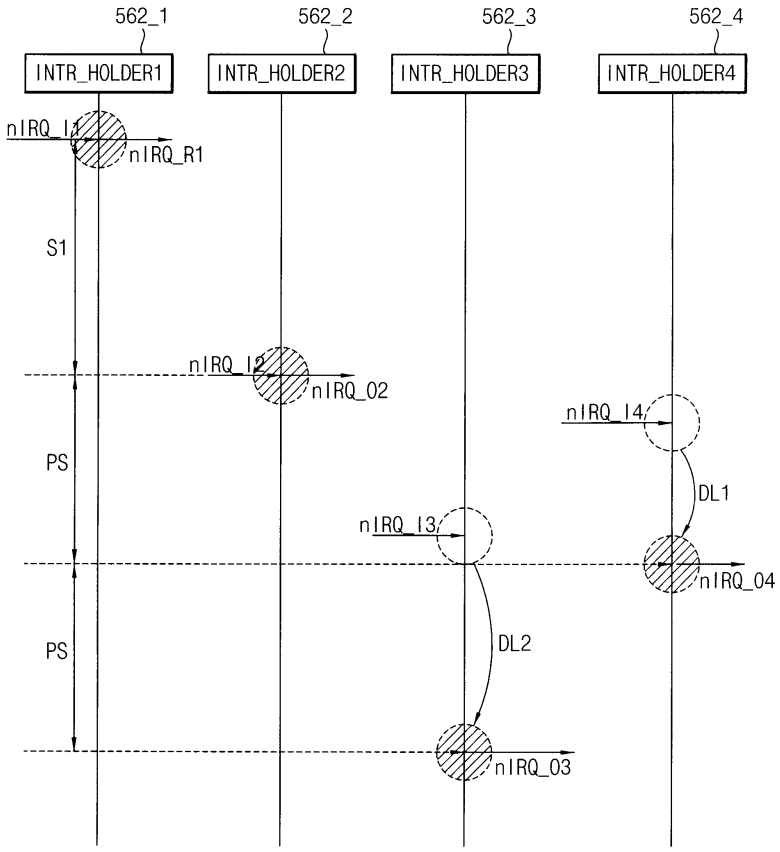
도면15


500
520_1
INTERRUPT SOURCE_1
INTERRUPT SOURCE_2
520_2
INTERRUPT SOURCE_n
520_n
INT_R1
INT_R2
INT_Rn
INTERRUPT CONTROLLER
540
nIRQ_I1
nIRQ_I2
nIRQ_Im
INTERRUPT SPREADER
560
nIRQ_O1
nIRQ_O2
nIRQ_Om
PROCESSOR_1
PROCESSOR_2
PROCESSOR_m
580_1
580_2
580_m

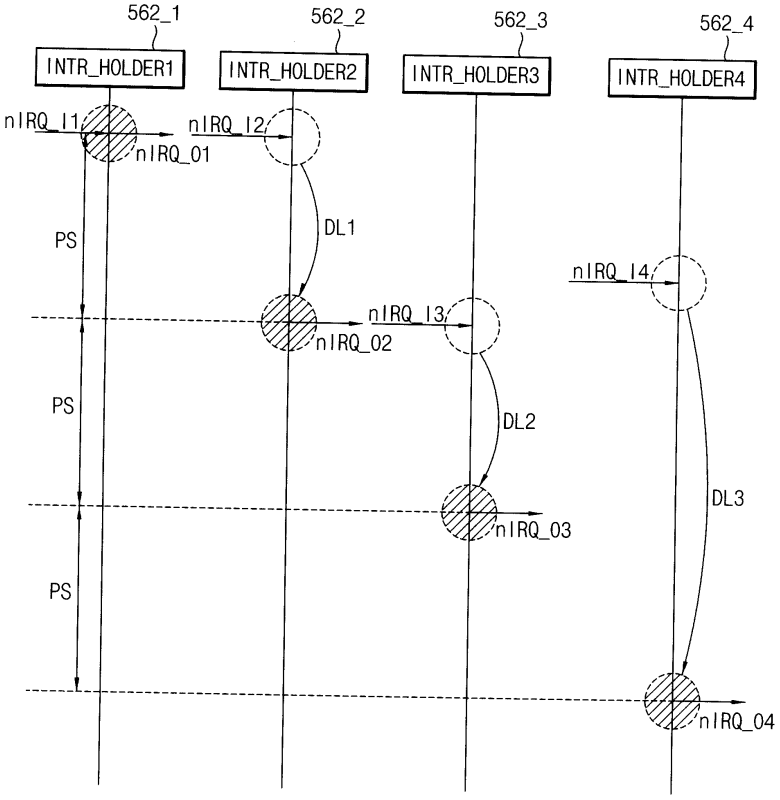
도면16


562_1
562_2
562_3
562_4
INTR_HOLDER1
INTR_HOLDER2
INTR_HOLDER3
INTR_HOLDER4
nIRQ_I1
nIRQ_01
PS
nIRQ_I3
DL1
nIRQ_I2
nIRQ_I4
nIRQ_03
DL2
PS
DL3
nIRQ_02
PS
nIRQ_04

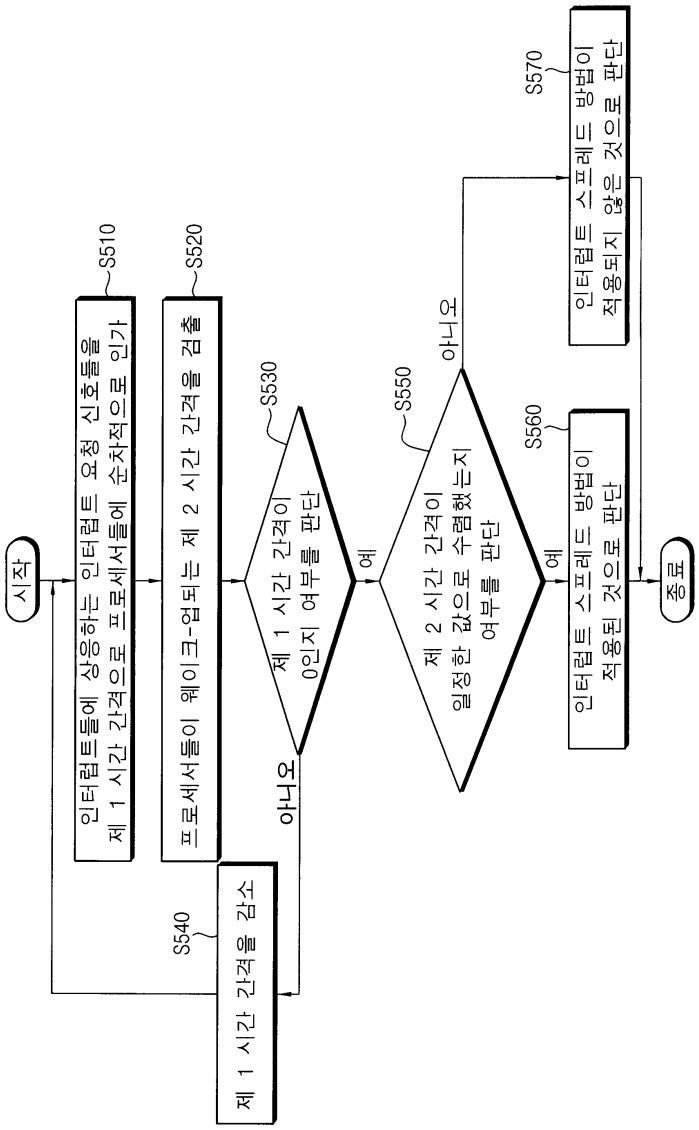
도면17


562_1
562_2
562_3
562_4
INTR_HOLDER1
INTR_HOLDER2
INTR_HOLDER3
INTR_HOLDER4
nIRQ_I1
nIRQ_R1
S1
nIRQ_I2
nIRQ_02
nIRQ_I4
DL1
PS
nIRQ_I3
nIRQ_04
DL2
PS
nIRQ_03

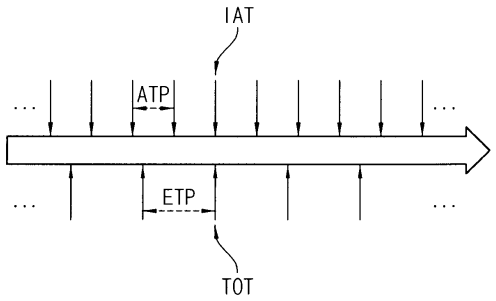
도면18


562_1
562_2
562_3
562_4
INTR_HOLDER1
INTR_HOLDER2
INTR_HOLDER3
INTR_HOLDER4
nIRQ_I1
nIRQ_I2
nIRQ_01
DL1
PS
nIRQ_I4
nIRQ_I3
nIRQ_02
DL2
PS
DL3
nIRQ_03
PS
nIRQ_04

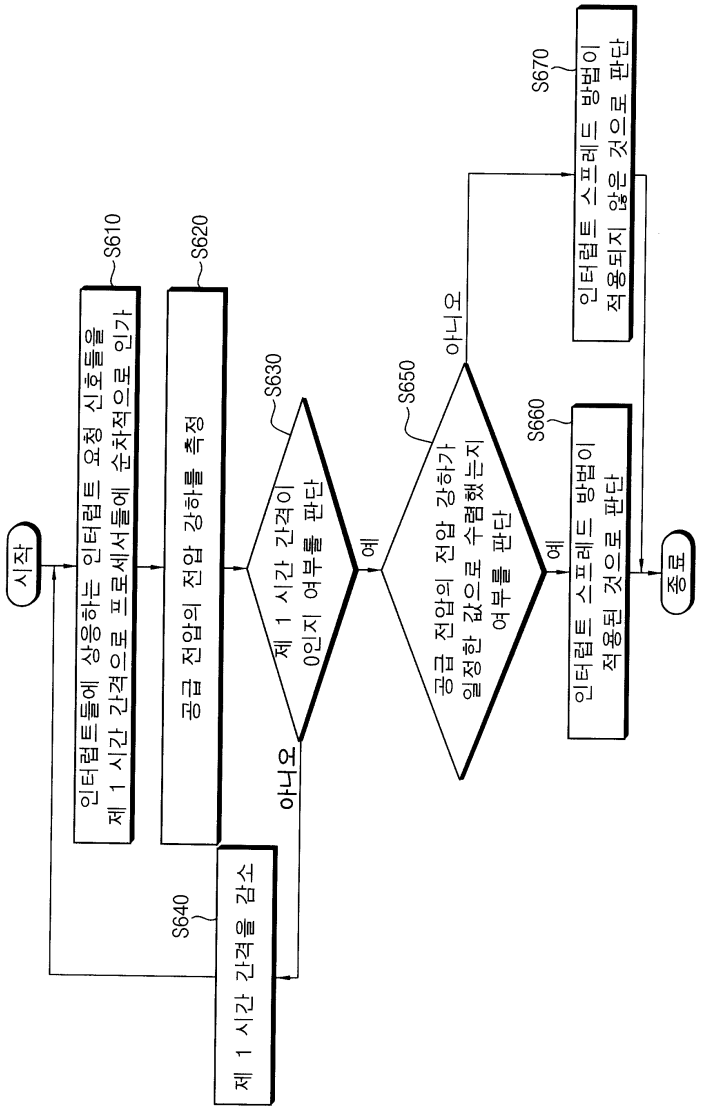
도면19


시작
인터럽트들에 상응하는 인터럽트 요청 신호들을 제 1 시간 간격으로 프로세서들에 순차적으로 인가
S510
프로세서들이 웨이크-업되는 제 2 시간 간격을 검출
S520
제 1 시간 간격이 0인지 여부를 판단
S530
아니오
예
제 1 시간 간격을 감소
S540
제 2 시간 간격이 일정한 값으로 수렴했는지 여부를 판단
S550
아니오
예
인터럽트 스프레드 방법이 적용된 것으로 판단
S560
인터럽트 스프레드 방법이 적용되지 않은 것으로 판단
S570
종료

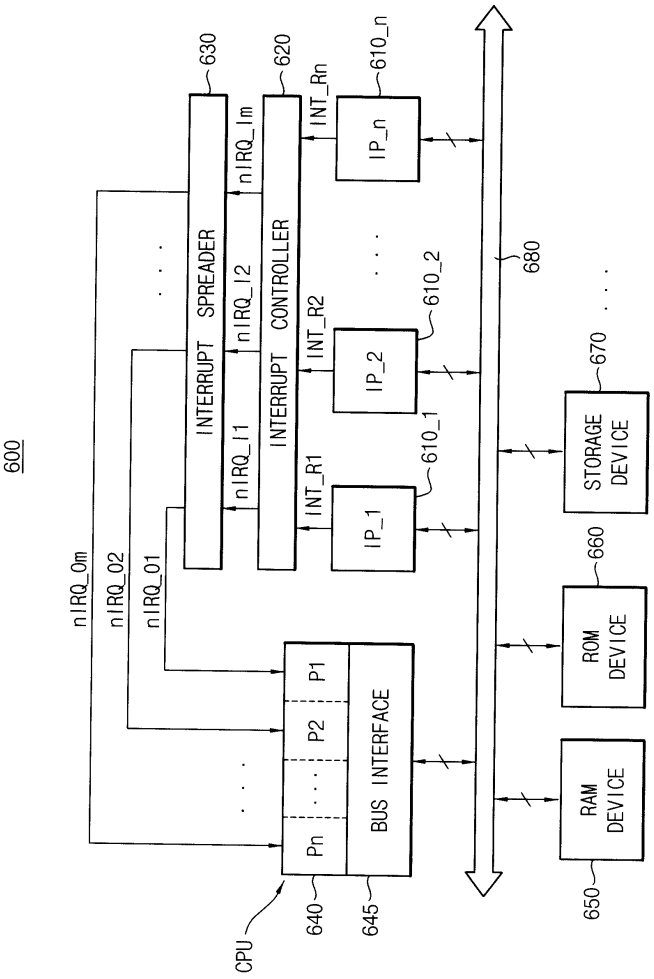
도면20


IAT
ATP
ETP
TOT

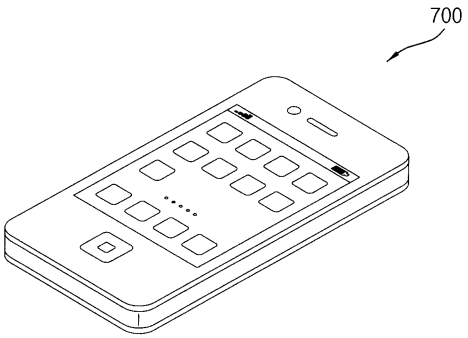
도면21


시작
인터럽트들에 상응하는 인터럽트 요청 신호들을 제 1 시간 간격으로 프로세서들에 순차적으로 인가
S610
공급 전압의 전압 강하를 측정
S620
제 1 시간 간격이 0인지 여부를 판단
S630
아니오
예
제 1 시간 간격을 감소
S640
공급 전압의 전압 강하가 일정한 값으로 수렴했는지 여부를 판단
S650
아니오
예
인터럽트 스프레드 방법이 적용된 것으로 판단
S660
인터럽트 스프레드 방법이 적용되지 않은 것으로 판단
S670
종료

도면22


600
630
620
610_n
INTERRUPT SPREADER
INTERRUPT CONTROLLER
nIRQ_Im
nIRQ_I2
nIRQ_I1
INT_Rn
INT_R2
INT_R1
IP_n
IP_2
IP_1
610_2
610_1
680
670
STORAGE DEVICE
660
ROM DEVICE
RAM DEVICE
650
nIRQ_0m
nIRQ_02
nIRQ_01
P1
P2
Pn
BUS INTERFACE
CPU
640
645

도면23


700