



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2013-0007792  
(43) 공개일자 2013년01월21일

(51) 국제특허분류(Int. Cl.)  
G06F 13/24 (2006.01) G06F 13/16 (2006.01)  
(21) 출원번호 10-2011-0068339  
(22) 출원일자 2011년07월11일  
심사청구일자 없음

(71) 출원인  
삼성전자주식회사  
경기도 수원시 영통구 삼성로 129 (매탄동)  
(72) 발명자  
한동구  
경기도 수원시 권선구 세권로 334, 주공아파트 333동 1005호 (권선동)  
이경하  
경기도 용인시 기흥구 용구대로2518번길 15, 201동 2201호 (보정동, 신촌마을포스홈타운)  
이강민  
경기도 수원시 영통구 덕영대로1555번길 20, 벽적골9단지아파트 945동 209호 (영통동)  
(74) 대리인  
이정순, 권혁록

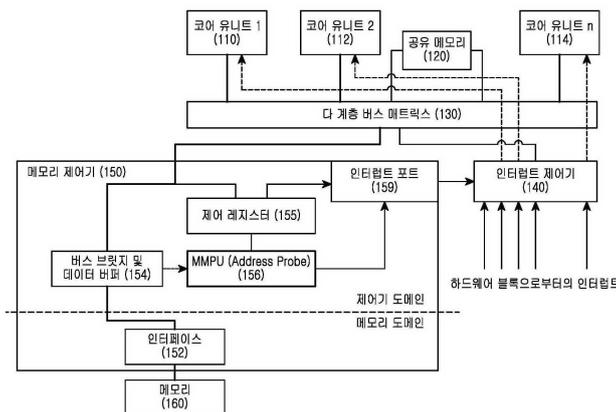
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **다중 프로세서를 지원하는 인터럽트 발생을 위한 장치 및 방법**

**(57) 요약**

본 발명은 인터럽트 지연 제거에 관한 것으로, 다중 프로세서를 지원하는 메모리 제어 장치에서의 인터럽트 발생 방법에 있어서 특정 메모리 영역에 대한 액세스가 발생하는 지 검사하는 과정과 메모리 액세스가 발생하는 경우 상기 액세스가 권한이 있는지 검사하는 과정과 상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하는 과정과 결정된 코어로 인터럽트 발생을 요청하는 과정을 포함하는 것으로 발명은 DMA 또는 코어가 메모리 액세스 시 MMPU에서 자동적으로 해당 인터럽트를 발생시키므로 인터럽트 지연을 제거할 수 있는 이점이 있다.

**대표도 - 도1**



## 특허청구의 범위

### 청구항 1

다중 프로세서를 지원하는 메모리 제어 장치에서의 인터럽트 발생 방법에 있어서,  
특정 메모리 영역에 대한 액세스가 발생하는 지 검사하는 과정과,  
메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하는 과정과,  
상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하는 과정과,  
결정된 코어로 인터럽트 발생을 요청하는 과정을 포함하는 것을 특징으로 하는 방법.

### 청구항 2

제 1항에 있어서,  
메모리 액세스를 요청한 블록으로 인터럽트 발생을 요청하는 과정을 더 포함하는 것을 특징으로 하는 방법.

### 청구항 3

제 1항에 있어서,  
상기 액세스가 권한이 있는지 검사하는 과정은,  
액세스가 발생한 메모리 주소가 기 설정되어 있는지 검사하는 과정을 포함하는 것을 특징으로 하는 방법.

### 청구항 4

제 1항에 있어서,  
메모리에 대한 액세스는 코어 또는 DMA(Direct Memory Access) 제어기에 의해 수행되는 것을 특징으로 하는 방법.

### 청구항 5

인터럽트를 발생하고 다중 프로세서를 사용하는 장치에 있어서,  
메모리에 대한 액세스 발생시 액세스되는 메모리 주소를 출력하는 버스 브릿지 및 데이터 버퍼와,  
상기 메모리 브릿지를 통해 메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트 발생을 인터럽트 포트를 통해 요청하는 메모리 보호부를 포함하는 것을 특징으로 하는 장치.

### 청구항 6

제 5항에 있어서,  
상기 메모리 보호부는,

메모리 액세스를 요청한 블록으로 인터럽트 발생을 요청하는 것을 특징으로 하는 장치.

#### 청구항 7

제 5항에 있어서,  
상기 메모리 보호부는,  
상기 액세스가 권한이 있는지 검사할 시,  
액세스가 발생한 메모리 주소가 기 설정되어 있는지 검사하는 것을 특징으로 하는 장치.

#### 청구항 8

제 5항에 있어서,  
메모리에 대한 액세스는 코어 또는 DMA(Direct Memory Access) 제어기에 의해 수행되는 것을 특징으로 하는 장치.

#### 청구항 9

인터럽트를 발생하고 다중 프로세서를 지원하는 장치에 있어서,  
다수의 코어 유니트와,  
메모리와,  
상기 다수의 코어 유니트와 메모리 사이에서, 상기 다수의 코어 유니트로부터의 메모리 액세스를 처리하고, 상기 메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트 발생을 요청하는 메모리 제어기와  
인터럽트 발생 요청을 수신하여 해당 코어로 인터럽트를 발생하는 인터럽트 제어기를 포함하는 것을 특징으로 하는 장치.

#### 청구항 10

제 9항에 있어서,  
상기 메모리 제어기는,  
메모리에 대한 액세스 발생시 액세스되는 메모리 주소를 출력하는 버스 브릿지 및 데이터 버퍼와,  
상기 메모리 브릿지를 통해 메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트 발생을 인터럽트 포트를 통해 요청하는 메모리 보호부를 포함하는 것을 특징으로 하는 장치.

#### 청구항 11

제 10항에 있어서,  
상기 메모리 보호부는,

메모리 액세스를 요청한 블록으로 인터럽트 발생을 요청하는 것을 특징으로 하는 장치.

**청구항 12**

제 10항에 있어서,  
상기 메모리 보호부는,  
상기 액세스가 권한이 있는지 검사할 시,  
액세스가 발생한 메모리 주소가 기 설정되어 있는지 검사하는 것을 특징으로 하는 장치.

**청구항 13**

제 10항에 있어서,  
메모리에 대한 액세스는 상기 코어 유닛 또는 DMA(Direct Memory Access) 제어기에 의해 수행되는 것을 특징으로 하는 장치.

**청구항 14**

인터럽트를 발생하고 다중 프로세서를 사용하는 제어장치에 있어서,  
메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있으면서 특정한 주소 범위 안에 포함되는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트를 요청하는 기능이 포함되는 메모리 보호부를 포함하는 것을 특징으로 하는 장치.

**청구항 15**

제 14항에 있어서,  
상기 메모리 보호부는,  
메모리 액세스를 감지하는 적어도 하나의 블록으로 인터럽트 발생을 요청하는 것을 특징으로 하는 장치.

**청구항 16**

제 15항에 있어서,  
메모리에 액세스의 감지는 코어 또는 DMA(Direct Memory Access) 제어기에 의해 메모리 액세스 시 수행되는 것을 특징으로 하는 장치.

**청구항 17**

인터럽트를 발생하고 다중 프로세서를 사용하는 제어장치에 있어서,  
다중 인터럽트 발생기를 포함하는 메모리 제어기를 포함하는 것을 특징으로 하는 장치.

**청구항 18**

제 17항에 있어서

상기 메모리 제어기는 다수의 프로세서 사이의 인터럽트 전달을 목적으로 SoC내에 별도의 신호연결 및 내부 메모리를 생략할 수 있는 것을 특징으로 하는 장치.

**청구항 19**

제 17항에 있어서

상기 메모리 제어기는

데이터 흐름에 있어서 송수신 단의 요청인터럽트에 반응하는 응답 인터럽트 발생 시간을 앞당겨서 처리 지연 시간을 줄이는 것을 특징으로 하는 장치.

**청구항 20**

제 17항에 있어서

상기 메모리 제어기는

DMA를 대상으로 MPU(Memory Protection Unit)을 동작시키는 것을 특징으로 하는 장치.

**명세서**

**기술분야**

[0001] 본 발명은 메모리 제어기에 관한 것으로, 특히 다중 프로세서를 구비한 시스템에서 인터럽트로 인한 시스템 상의 지연을 제거하기 위한 방법 및 장치에 관한 것이다.

**배경기술**

[0002] 프로세서의 개념을 확대하면 DMA (Direct Memory Access) 제어기도 독립된 프로세서로 간주 될 수 있다. 즉, 상기 DMA 제어기도 마스터 권한으로 메모리를 액세스할 권한을 가진다.

[0003] 다중 프로세서 사이의 데이터 교환 시 데이터가 메모리 내에 존재한다면, 상기 데이터를 수신할 프로세서에게 메모리 상의 위치정보와 상기 데이터에 대한 인터럽트 신호를 제공할 수 있다. 이러한 방식은 대표적으로 프로세서 간 메일 박스 방식을 들 수 있다. 여기서 메일 박스 방식은 메시지와 하드웨어적인 인터럽트 신호(예외 신호)로 구성된다.

[0004] 상기 프로세서 또는 상기 DMA 제어기가 동작할 때에는 항상 인터럽트 신호를 발생한다. 즉, 메시지 교환을 제어하는 수단으로 인터럽트가 사용된다.

[0005] 하드웨어에서 발생하는 인터럽트에 반응하는 ISR (Interrupt Service S/W Routine)은 컨텍스트 스위칭(Context Switching)을 위해 최소한의 전처리가 필요하고, 이로 인한 지연은 운영체제에 따라 10us~50us 정도 소요된다.

[0006] 하지만 이러한 인터럽트를 ISR 내에서만 모두 처리하는 방식으로는 장치 핸들러 소프트웨어 개발은 불가능하다. 그 이유는, ISR 로 인한 영향은 사용자 작업에 따라 다르며, ISR 동작시 운영 체제가 제공하는 API (Application Programming Interface) 즉, 시스템 호출을 수반해야 하고, 이는 필연적으로 지연(delay)을 발생시켜 시스템 성능에 부정적인 영향을 미치기 때문이다.

[0007] 따라서, 인터럽트 지연을 제거하기 위한 기술이 필요하다.

**발명의 내용**

**해결하려는 과제**

- [0008] 본 발명의 목적은 다중 프로세서를 지원하는 인터럽트 발생을 위한 장치 및 방법을 제공함에 있다.
- [0009] 본 발명의 다른 목적은 프로세서 간 통신에서 인터럽트 지연을 없애기 위한 방법 및 장치를 제공함에 있다.
- [0010] 본 발명의 또 다른 목적은 프로세서 간 통신에서 소프트웨어적인 유연성을 가지는 인터럽트 발생을 위한 방법 및 장치를 제공함에 있다.

**과제의 해결 수단**

- [0011] 상기 목적들을 달성하기 위한 제 1 견지에 따르면, 다중 프로세서를 지원하는 메모리 제어 장치에서의 인터럽트 발생 방법에 있어서 특정 메모리 영역에 대한 액세스가 발생하는 지 검사하는 과정과 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있고 임의의 주소범위 내에 포함되어 있는 경우 검사하는 과정과 상기 액세스가 권한이 있고 임의의 주소범위 안에 포함되는 경우, 인터럽트를 발생할 코어를 결정하는 과정과 결정된 코어로 인터럽트 발생을 요청하는 과정을 포함하는 것을 특징으로 한다.
- [0012] 상기 목적들을 달성하기 위한 제 2 견지에 따르면, 인터럽트를 발생하는 다중 프로세서를 지원하는 장치에 있어서 메모리에 대한 액세스 발생시 액세스되는 메모리 주소를 출력하는 버스 브릿지 및 데이터 버퍼와 상기 메모리 브릿지를 통해 메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트 발생을 인터럽트 포트를 통해 요청하는 메모리 보호부를 포함하는 것을 특징으로 한다.
- [0013] 상기 목적들을 달성하기 위한 제 3 견지에 따르면, 인터럽트를 발생하는 다중 프로세서를 지원하는 장치에 있어서, 다수의 코어 유니트와 메모리와 상기 다수의 코어 유니트와 메모리 사이에서, 상기 다수의 코어 유니트로부터의 메모리 액세스를 처리하고, 상기 메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트 발생을 요청하는 메모리 제어기와 인터럽트 발생 요청을 수신하여 해당 코어로 인터럽트를 발생하는 인터럽트 제어기를 포함하는 것을 특징으로 한다.
- [0014] 본 발명의 목적들을 달성하기 위한 제 4 견지에 따르면, 인터럽트를 발생하고 다중 프로세서를 사용하는 제어장치에 있어서, 메모리에 대한 액세스가 발생하는 지 검사하고, 메모리 액세스가 발생하는 경우, 상기 액세스가 권한이 있는지 검사하고, 상기 액세스가 권한이 있으면서 특정한 주소 범위 안에 포함되는 경우, 인터럽트를 발생할 코어를 결정하고 결정된 코어로 인터럽트를 요청하는 기능이 포함되는 메모리 보호부를 포함하는 것을 특징으로 한다.
- [0015] 본 발명의 목적들을 달성하기 위한 제 5 견지에 따르면, 인터럽트를 발생하고 다중 프로세서를 사용하는 제어장치에 있어서, 다중 인터럽트 발생기를 포함하는 메모리 제어기를 포함하는 것을 특징으로 한다.

**발명의 효과**

- [0016] 기존의 구조는 DMA 또는 코어가 데이터를 읽고/쓰기 위하여 메모리에 액세스 해도 별도의 인터럽트 제어 레지스터 블록 설정을 통해 상대 블록에 인터럽트를 전달해야 하므로, 데이터 처리(쓰기/읽기)시, 전송 단위로 데이터 흐름이 중단되고, 이러한 과정에서 인터럽트 처리를 위한 지연이 발생하지만, 본 발명은 DMA 또는 코어가 메모리 액세스시 MMPU (Modified Memory Protection Unit)에서 데이터 접근의 선별적인 경우에 따라 자동적으로 인터럽트 전달을 발생시키므로 인터럽트를 생략하거나 지연을 줄일 수 있는 이점이 있다.

**도면의 간단한 설명**

- [0017] 도 1은 본 발명의 실시 예에 따른 다중 프로세서를 구비한 시스템의 블록 구성을 도시한 도면이다.
- 도 2는 본 발명의 실시 예에 따른 MMPU 에서 프로그램 가능한 레지스터를 도시한 도면이다.
- 도 3은 본 발명의 실시 예에 따른 MMPU 를 사용시 동작 과정을 도시한 도면이다.
- 도 4는 본 발명의 실시 예에 따른 MMPU 동작 과정을 도시한 흐름도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0018] 이하 본 발명의 바람직한 실시 예를 첨부된 도의 참조와 함께 상세히 설명한다. 그리고, 본 발명을 설명함에 있어서, 관련된 공지기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단된 경우 그 상세한 설명은 생략할 것이다. 그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.
- [0019] 이하, 본 발명은 다중 프로세서를 지원하는 인터럽트 발생을 위한 장치 및 방법에 대해 설명하기로 한다.
- [0020] 본 발명은 기존의 메모리 제어기(Memory Controller) 내부에 개선된 MPU (MMPU, Modified Memory Protection Unit)를 추가하여 상기 MMPU가 인터럽트 발생을 처리하는 구성을 가진다. 본 발명에서 프로세서는 코어와 동일한 의미를 가진다. 본 발명에서는 MMPU는 메모리 보호 부라고 칭할 수 있다.
- [0021] 도 1은 본 발명의 실시 예에 따른 다중 프로세서를 구비한 시스템의 블록 구성을 도시한 도면이다.
- [0022] 상기 도 1을 참조하면, 상기 다중 프로세서를 구비한 시스템은 CPU(Central Processing Unit) 혹은 DMA 제어기와 같은 다수의 코어 유니트(110, 112, 114), 인터럽트 제어기(140), 메모리 제어기(150), 메모리(160)를 포함하여 구성된다.
- [0023] 상기 메모리 제어기(150), 인터럽트 제어기(140) 및 다수의 코어 유니트(110, 112, 114)는 다 계층 버스 매트릭스(130)를 통해 서로 통신한다 그리고 상기 메모리 제어기(150)는 전술한 MMPU의 기능을 내장한다.
- [0024] 일반적인 메모리 보호기(MMU 혹은 MPU)는 코어 유니트에 밀착하여 존재하고, 그 기능은 액세스 금지(Access Protected)된 영역으로 권한이 없는 액세스가 발생될 경우, 코어에 예외 발생을 알리는 기능을 수행하고, 이는 운영체제에서는 치명적 오류 발생을 알리는 기능을 수행하였다 그리고 기존의 메모리 제어기는 코어 유니트간 IPC(Inter Processor Communication)용으로 인터럽트 신호를 만들어내지는 않는다.
- [0025] 하지만, 상기 메모리 제어기(150)는 인터럽트 발생 및 전달 기능을 자체 수행하기 위해 내부에 MMPU(156)를 추가적으로 구비한다.
- [0026] 상기 MMPU (156)는 버스 브릿지 및 데이터 버퍼(154)로부터, 액세스되는 메모리의 주소를 획득하여, 인터럽트를 위하여 임의로 설정된 주소 영역이 액세스 되면, 액세스 되는 주소 영역에 동기화되어 있는 인터럽트 수신자 코어를 검색하여 결정하고, 인터럽트 포트(159)를 통해 인터럽트 제어기(140)로 해당 인터럽트 발생을 요청한다.
- [0027] 여기서, 상기 MMPU (156)는 DMA 제어기 기능을 수행하는 하드웨어 모듈에 의해서도 인터럽트를 발생시킬 수 있다. 부가적으로, 만약, DMA 제어기가 액세스 하는 메모리 영역이 접근 권한이 없는 경우, 상기 MMPU(156)는 권한이 없다고 판단할 수 있다.
- [0028] 상기 버스 브릿지 및 데이터 버퍼(154)는 상기 다수의 코어 유니트(110, 112, 114)의 메모리 액세스(읽기/쓰기) 요청을 수신하여 상기 요청대로 상기 메모리(160)를 인터페이스(152)를 통해 액세스한다 그리고 제어 레지스터(155)는 기존의 메모리 제어기(Memory Controller) 제어기능을 수행하면서 동시에 인터럽트를 위한 MMPU(156)의 주소 영역 선별 및 인터럽트 방식 설정을 포함할 수 있다.
- [0029] 상기 제어 레지스터(155)에 소프트웨어에 의하여 인터럽트를 위한 주소 영역 선별 정보와 인터럽트 방식 정보(수신자/접근조건/발생사유/신호방법 등)이 포함될 시, 상기 MMPU(156)는 상기 제어 레지스터(155)로부터 이러한 정보를 수신하여 설정할 수 있다.
- [0030] 상기 인터럽트 제어기(140)는 상기 인터럽트 포트(159)를 통한 상기 MMPU(156)의 인터럽트 발생요청을 수신하여 해당 코어로 인터럽트를 발생한다. 상기 인터럽트 포트(159)는 상기 인터럽트 제어기(140)와 상기 MMPU(156)사이에서 인터페이스 역할을 수행한다.
- [0031] 도 2는 본 발명의 실시 예에 따른 MMPU에서 프로그램 가능한 레지스터를 도시한 도면이다.
- [0032] 상기 도 2를 참조하면, 상기 MMPU의 프로그램 가능한 레지스터는 도면에서와 같이, 코어(210, 220, 230)마다 할

당된 메모리 주소 영역(시작 주소 및 끝 주소)와 인터럽트 발생을 위한 제어 레지스터(해당 코어 ID(이벤트 목적 코어 ID)/읽기쓰기 상황 별 선택/인터럽트 상태정보/신호레벨 등을 비트단위로 제어)를 가질 수 있다.

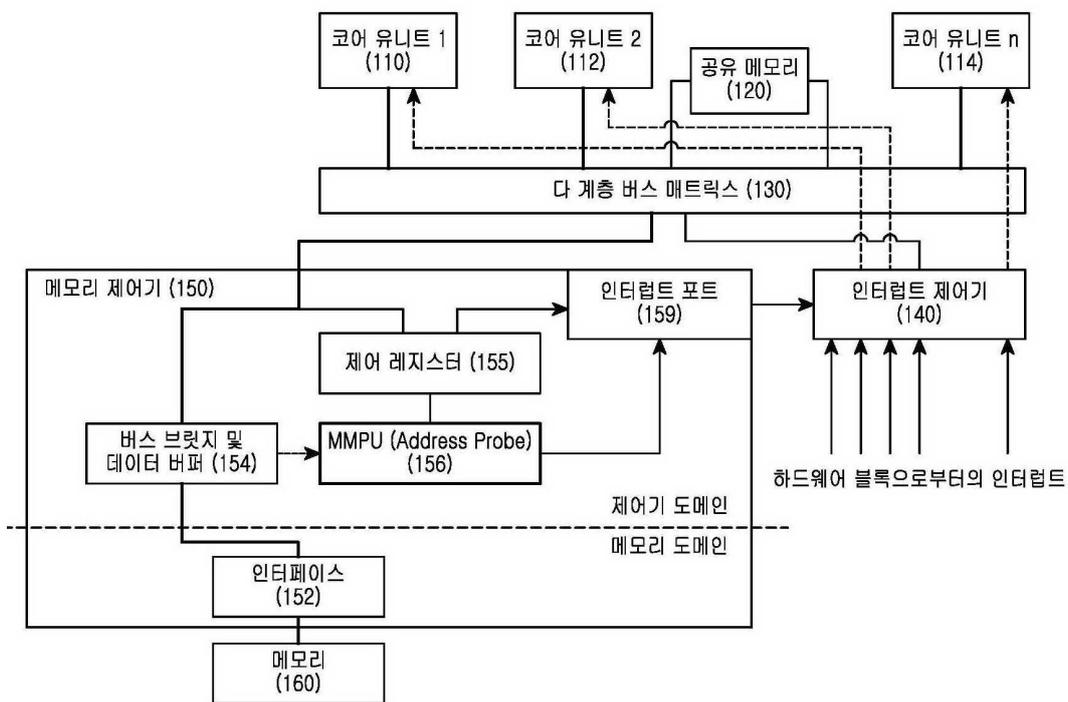
- [0033] 다시 설명하면, 각 코어에 대하여 각각 4쌍(예시)의 레지스터 그룹(Register Group)이 도시되어 있다. 상기 레지스터 그룹을 기반으로 특정 메모리 주소 영역(시작 주소 레지스터 값 ~ 끝 주소 레지스터 값) 내에 액세스(읽기/쓰기)가 발생하는 경우, MPU는 제어레지스터 정보를 토대로 해당 코어로 인터럽트를 발생시킨다.
- [0034] 상기 MPU는 프로그램 가능한 구조이다. 즉, 프로그래밍을 통해 정보(시작 주소, 끝 주소 및 인터럽트 방법)를 갱신하거나 새로 작성하거나 삭제할 수 있다.
- [0035] 예를 들어, 특정 인터럽트 발생시, 다수의 코어에 인터럽트를 발생하고자 하는 경우, MPU에, 특정 인터럽트 발생을 위한 주소 영역에 다수의 코어가 설정되면 가능하다.
- [0036] 상기 MPU는 도 1에서 버스 브릿지 및 데이터 버퍼(154)로부터 액세스 되는 메모리의 주소를 제공받는 경우, 구비하고 있는 주소 리스트 중에서 일치하는 주소가 있는 지를 검사하고, 일치하는 주소가 존재하는 경우, 일치하는 주소에 해당하는 코어에 대한 인터럽트 발생을 도 1의 인터럽트 제어기(140)에 요청한다.
- [0037] 도 3은 본 발명의 실시 예에 따른 MPU를 사용시 동작 과정을 도시한 도면이다.
- [0038] 상기 도 3을 참조하면, 본 발명의 기존과 달리 메모리 액세스 시 2개의 인터럽트 발생만이 필요하다.
- [0039] 모뎀 부 코어에서의 외부 MAC 프레임 데이터가 DMA 제어기에 의해 메모리에 옮겨지면서 최초 액세스 되고 사용자 데이터로 변환되면서 DMA 제어기가 한번 더 관여하고 CPU 코어에게 인터럽트를 알린다(310 단계), 수신 측 사용자 코어에 의해 상기 데이터가 읽혀서 처리되고(320 단계), 처리완료 인터럽트가 발생한다(330 단계).
- [0040] 이것은 기존의 절반으로 인터럽트 수를 줄일 수 있는 경우에 해당하며, 여기서 DMA -> DMA 간 인터럽트 알림 및 데이터를 읽는 동안 처리되고 있음을 역으로 자동 인지시키는 동작이 MPU에 의해 생성되어 인터럽트 전처리 과정을 시간상 중첩되게 하여 전체 지연을 줄이게 된다.
- [0041] 본 발명에서, 미리 설정된 주소 영역에 액세스가 발생할 시 목적 코어(Destination)에게 자동으로 인터럽트가 전달된다.
- [0042] 또한, 데이터 처리 과정에서, 마지막 데이터를 독출하는 시점에, 본 발명의 MPU는 데이터 독출 처리가 끝나고 있음을 자동으로 인지하여 최초 코어(혹은 DMA)에게 해당 인터럽트를 역으로 전달하게 되어 ACK 전송 지연을 줄이는 것도 가능하다. 즉, 액세스를 발생시킨 해당 모듈(최초 코어)에게도 자동으로 인터럽트가 전달될 수 있다. 상기 최초 코어는 DMA 제어기가 될 수 있다.
- [0043] 도 4는 본 발명의 실시 예에 따른 MPU 동작 과정을 도시한 흐름도이다.
- [0044] 상기 도 4를 참조하면, 상기 MPU는 메모리에 대한 액세스가 발생하는 경우(410 단계), 상기 액세스가 발생한 메모리의 번지(주소)를 분석한다(420 단계).
- [0045] 분석 결과, 상기 액세스가 유효하여 액세스 권한이 있는 경우(430 단계), 액세스 메모리 번지에 해당하는 코어를 결정하고(440 단계), MPU는 해당 코어로 인터럽트 발생을 인터럽트 제어기로 요청한다(450 단계).
- [0046] 만약, 분석 결과, 상기 액세스가 유효하지 않아 액세스 권한이 없는 경우(435 단계), 기존과 같이 해당 코어로 "권한없는 액세스"를 통보한다(435 단계),
- [0047] 여기서, 권한이 있는 액세스를 판단하는 기준은, 도 2와 같이 인터럽트 발생을 위한 레지스터 그룹에 액세스 요청 주소가 포함되어 있는지 없는지가 기준이 될 수 있다. 상기 액세스 요청 주소는 특정 또는 임의의 주소범위로 포함될 수 있다.
- [0048] 참고적으로, 본 발명은 메일 박스(IPC)의 기능이 공유 메모리가 필요하지 않고 메모리 자원에 구애받지 않고 구현될 수 있다. 메일박스 송수신자에 DMA 제어기도 포함될 수 있는 방법을 제공하며 그리고 본 발명은 MPU 레지스터 설정시, 해당 레지스터를 설정하는 방법에 따라 소프트웨어 디버깅을 포함한 다양한 목적으로 CPU 코어에 방해가 주지 않고 DMA를 연동하여 데이터 덤프(Data Dump)등을 효율적으로 수행할 수 있다.

[0049] 본 발명의 인터럽트를 발생하고 다중 프로세서를 사용하는 제어장치는 다중 인터럽트 발생기를 포함하는 메모리 제어기를 포함한다. 상기 상기 메모리 제어기는 다수의 프로세서 사이의 인터럽트 전달을 목적으로 SoC내에 별도의 신호연결 및 내부 메모리를 생략할 수 있다 그리고 상기 메모리 제어기는 데이터 흐름에 있어서 송수신 단의 요청인터럽트에 반응하는 응답 인터럽트 발생 시간을 앞당겨서 처리 지연시간을 줄일 수 있다 그리고 상기 메모리 제어기는 DMA를 대상으로 MPU(Memory Protection Unit)을 동작시킬 수 있다.

[0050] 한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 아니 되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

도면

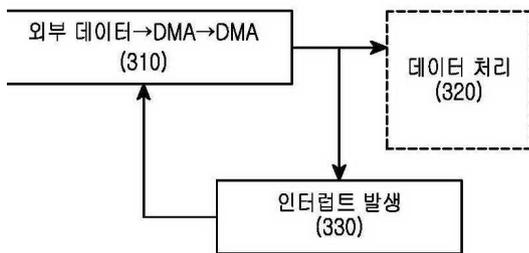
도면1



도면2



도면3



도면4

