

는 그 초과의 스트림들 중 하나에서 식별된다. 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 메모리 요청들의 상이한 스트림으로부터의 이전의 메모리 요청이 계류중이라고 결정할 시에 지연된다. 메모리 액세스 카운터는 완료하도록 보장되는 각각의 메모리 요청에 대해 감분된다. 메모리 액세스 카운터의 초기화된 상태와는 상이한 메모리 액세스 카운터 내의 카운트 값은, 계류중인 메모리 요청들이 존재한다는 것을 표시한다. 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 추가적인 계류중인 메모리 요청들이 존재하지 않는다고 결정할 시에 프로세싱된다.

(72) 발명자

사토리우스, 토마스 앤드류

미국 92121 캘리포니아주 샌 디에고 모어하우스
드라이브 5775

스피어, 토마스 필립

미국 92121 캘리포니아주 샌 디에고 모어하우스
드라이브 5775

특허청구의 범위

청구항 1

순서화된 메모리 액세스들을 요구하는 메모리 요청들을 순서화하기 위한 방법으로서,

어드레스 속성에 따라 단일 소스로부터의 메모리 요청들의 스트림을 메모리 요청들의 2개 또는 그 초과의 스트림들로 분할하는 단계;

상기 메모리 요청들의 2개 또는 그 초과의 스트림들 중 하나에서 순서화된 메모리 액세스들을 요구하고 제 1 어드레스 속성을 갖는 메모리 요청을 식별하는 단계; 및

상기 메모리 요청들의 2개 또는 그 초과의 스트림들 중 상이한 스트림으로부터의 이전의 메모리 요청이 계류 중이라고 결정할 시에 상기 순서화된 메모리 액세스들을 요구하고 제 2 어드레스 속성을 갖는 메모리 요청을 지연(stall)시키는 단계를 포함하는, 메모리 요청들을 순서화하기 위한 방법.

청구항 2

제 1 항에 있어서,

상기 어드레스 속성은 어드레스 경계값인, 메모리 요청들을 순서화하기 위한 방법.

청구항 3

제 2 항에 있어서,

상기 어드레스 경계값은, 메모리 영역이 홀수(odd) 어드레스 메모리 영역인지 또는 짝수 어드레스 메모리 영역인지를 결정하는 어드레스 비트인, 메모리 요청들을 순서화하기 위한 방법.

청구항 4

제 2 항에 있어서,

상기 어드레스 경계값은, N개의 메모리 영역들 중 하나를 식별하는 메모리 요청 어드레스 내의 복수의 비트들로부터 결정되는, 메모리 요청들을 순서화하기 위한 방법.

청구항 5

제 1 항에 있어서,

상기 메모리 요청들의 스트림은, 상기 메모리 요청들의 2개 또는 그 초과의 스트림들 중 메모리 요청들의 각각의 스트림 상에서 데이터를 전달하기 위한 이용가능도에 따라 분할되는, 메모리 요청들을 순서화하기 위한 방법.

청구항 6

제 1 항에 있어서,

상기 메모리 요청들의 스트림 내의 메모리 요청은, 어드레스 경계값에 따라 N개의 메모리 영역들 중 하나로 안내되는, 메모리 요청들을 순서화하기 위한 방법.

청구항 7

제 1 항에 있어서,

상기 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 강하게(strongly) 순서화된 메모리 요청 또는 디바이스 메모리 요청(SO/DV)인, 메모리 요청들을 순서화하기 위한 방법.

청구항 8

제 1 항에 있어서,

상기 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 원자(atomic) 메모리 액세스들을 제공하는 메모리 액세스 배타적인 명령(memory access exclusive instruction)들의 시퀀스인, 메모리 요청들을 순서화하기 위한 방법.

청구항 9

제 1 항에 있어서,

지연된 메모리 요청은, 계류중인 이전의 메모리 요청이 순서대로 완료하도록 보장된다는 표시를 수신한 이후 프로세싱되는, 메모리 요청들을 순서화하기 위한 방법.

청구항 10

제 1 항에 있어서,

각각의 메모리 요청에 대해 메모리 액세스 카운터를 증분시키는 단계; 및

원하는 순서대로 완료하도록 보장되기 위해 표시되는 각각의 메모리 요청에 대해 상기 메모리 액세스 카운터를 감분시키는 단계 – 상기 메모리 액세스 카운터의 초기화된 상태와는 상이한 상기 메모리 액세스 카운터 내의 카운트 값은, 하나 또는 그 초과의 계류중인 요청들이 상기 원하는 순서대로 완료하도록 보장되지 않는다는 것을 표시함 – 를 더 포함하는, 메모리 요청들을 순서화하기 위한 방법.

청구항 11

순서화된 메모리 액세스들을 요구하는 메모리 요청들을 순서화하기 위한 장치로서,

단일 소스로부터 메모리 요청들의 제 1 스트림을 모니터링하도록 구성되고, 각각의 메모리 요청의 어드레스 속성에 기초하여, 상기 메모리 요청들의 제 1 스트림을 메모리 요청들의 제 2 스트림 및 메모리 요청들의 제 3 스트림으로 분할하도록 구성된 스트림 분할 회로;

제 2 인터리빙된 메모리 어드레스에서 계류중인 메모리 요청을 갖는 상기 메모리 요청들의 제 3 스트림과는 상이한 메모리 요청들의 스트림인 상기 메모리 요청들의 제 2 스트림으로부터 순서화된 메모리 액세스들을 요구하는 메모리 요청을 제 1 인터리빙된 메모리 어드레스에서 검출하도록 구성된 트래킹(tracking) 회로; 및

상기 계류중인 메모리 요청이 순서대로 완료하도록 보장될 때까지, 상기 계류중인 메모리 요청에 응답하여, 상기 순서화된 메모리 액세스들을 요구하는 메모리 요청들의 제 2 스트림을 지연시키도록 구성된 지연 회로를 포함하는, 메모리 요청들을 순서화하기 위한 장치.

청구항 12

제 11 항에 있어서,

상기 분할은, 메모리 영역들의 수 및 메모리 영역 사이즈에 기초하여 상기 어드레스 속성에 따라 결정되는, 메모리 요청들을 순서화하기 위한 장치.

청구항 13

제 11 항에 있어서,

상기 메모리 요청들의 제 1 스트림은, 메모리 요청들의 2개 또는 그 초과의 스트림들 중 메모리 요청들의 각각의 스트림 상에서 데이터를 전달하기 위한 이용가능도에 따라 분할되는, 메모리 요청들을 순서화하기 위한 장치.

청구항 14

제 11 항에 있어서,

상기 트래킹 회로는, 상기 메모리 요청들의 제 2 스트림으로부터, 또는 이전의 메모리 요청과 동일한 메모리 요청들의 스트림인 상기 메모리 요청들의 제 3 스트림으로부터 메모리 요청을 수신할 시에 증분하고, 수신된 메모리 요청이 계류중이라는 것을 표시하도록 구성된 카운터를 포함하며,

상기 카운터는, 상기 계류중인 메모리 요청이 순서대로 완료하도록 보장된다는 확인응답을 수신할 시에 감분

하도록 구성되는, 메모리 요청들을 순서화하기 위한 장치.

청구항 15

제 11 항에 있어서,

제 1 메모리 영역 및 제 2 메모리 영역 – 상기 메모리 요청들의 제 2 스트림은 상기 제 1 메모리 영역과 연관되고, 상기 메모리 요청들의 제 3 스트림은 상기 제 2 메모리 영역과 연관됨 – ; 및

상기 메모리 요청들의 제 1 스트림을 생성하도록 구성된 프로세서 – 각각의 메모리 요청은, 홀수 어드레스 메모리 영역으로서 상기 제 1 메모리 영역을 식별하고, 짝수 어드레스 메모리 영역으로서 상기 제 2 메모리 영역을 식별하는 연관된 어드레스 속성을 포함함 – 를 더 포함하는, 메모리 요청들을 순서화하기 위한 장치.

청구항 16

제 11 항에 있어서,

상기 어드레스 속성은, 메모리 요청들의 2개 또는 그 초과의 스트림들을 식별하는 하나 또는 그 초과의 어드레스 비트들인, 메모리 요청들을 순서화하기 위한 장치.

청구항 17

제 11 항에 있어서,

N개의 메모리 영역들 – 상기 메모리 요청들의 제 1 스트림은 모니터링되고, 메모리 요청들의 N개의 스트림들로 분할되며, 메모리 요청들의 각각의 스트림은 상기 N개의 메모리 영역들의 상이한 메모리 영역과 연관됨 – ; 및

상기 메모리 요청들의 제 1 스트림을 생성하도록 구성된 프로세서 – 각각의 메모리 요청은 상기 N개의 메모리 영역들 중 하나를 식별하는 연관된 어드레스 속성을 포함함 – 를 더 포함하는, 메모리 요청들을 순서화하기 위한 장치.

청구항 18

제 11 항에 있어서,

상기 트래킹 회로는, 강하게 순서화된 또는 디바이스(S0/DV) 메모리 요청들의 제 2 스트림으로부터, 또는 이전의 디바이스 메모리 요청과 동일한 디바이스 메모리 요청들의 스트림인 디바이스 메모리 요청들의 제 3 스트림으로부터 상기 S0/DV 메모리 요청을 수신할 시에 충분하고, 수신된 S0/DV 메모리 요청이 계류중임을 표시하도록 구성된 제 2 카운터 – 상기 제 2 카운터는, 계류중인 디바이스 메모리 요청이 순서대로 완료하도록 보장된다는 확인응답을 수신할 시에 감분하도록 구성됨 – 를 포함하는, 메모리 요청들을 순서화하기 위한 장치.

청구항 19

제 11 항에 있어서,

상기 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 원자 메모리 액세스를 제공하는 메모리 액세스 배타적인 명령인, 메모리 요청들을 순서화하기 위한 장치.

청구항 20

제 11 항에 있어서,

상기 계류중인 메모리 요청은 메모리 액세스 배타적인 명령에 응답하며,

상기 메모리 요청들의 지연된 제 2 스트림의 프로세싱은, 상기 계류중인 메모리 요청이 순서대로 완료하도록 보장된다는 표시를 수신한 이후, 인에이블되는, 메모리 요청들을 순서화하기 위한 장치.

청구항 21

메모리 액세스들을 순서화하기 위한 장치로서,

2개 또는 그 초과의 대응하는 버스 포트들과 연관된 속성에 따라 상기 대응하는 버스 포트들에 안내된 메모리 요청들의 2개 또는 그 초과의 스트림들로, 메모리 요청들의 스트림을 인터리빙하도록 구성된 스위치 및 제어 회로;

메모리 요청들의 제 1 스트림에서 순서화된 메모리 액세스를 요구하는 메모리 요청을 제 1 인터리빙된 메모리 어드레스에서 검출하고, 메모리 요청들의 제 2 스트림에서 적어도 하나의 계류중인 메모리 요청을 제 2 인터리빙된 메모리 어드레스에서 검출하도록 구성된 트래킹 회로; 및

더 이상 계류중인 메모리 요청들이 존재하지 않는다는 것을 상기 트래킹 회로가 표시할 때까지 상기 순서화된 메모리 액세스 요청을 자연시키도록 구성된 자연 회로를 포함하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 22

제 21 항에 있어서,

상기 속성은, 메모리 영역들의 수 및 메모리 영역 사이즈에 기초하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 23

제 21 항에 있어서,

상기 메모리 요청들의 스트림은, 단일 소스에서 비롯되고, 상기 메모리 요청들의 2개 또는 그 초과의 스트림들 중 메모리 요청들의 각각의 스트림 상에서 데이터를 전달하기 위한 이용가능도에 기초하여, 상기 메모리 요청들의 2개 또는 그 초과의 스트림들로 인터리빙되는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 24

제 21 항에 있어서,

상기 속성은 메모리 인터리브 어드레스에 기초하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 25

제 21 항에 있어서,

상기 속성은 메모리 요청의 목적지 어드레스의 해시(hash)에 기초하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 26

제 21 항에 있어서,

상기 스위치 및 제어 회로는, 상기 속성을 결정하기 위해 상기 메모리 요청의 목적지 어드레스 필드 내의 비트 필드를 모니터링하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 27

제 21 항에 있어서,

상기 순서화된 메모리 액세스 요청은 어써트(assert)된 강하게 순서화된 속성을 갖는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 28

제 21 항에 있어서,

상기 트래킹 회로는, 각각의 수신된 메모리 요청에 대해 충분하고, 완료하도록 보장되는 각각의 수신된 메모리 요청에 대해 감분하도록 구성된 메모리 액세스 카운터 – 상기 메모리 액세스 카운터의 초기화된 상태와는 상이한 상기 메모리 액세스 카운터 내의 카운트 값은, 완료하도록 보장되지 않는 하나 또는 그 초과의 계류중인 메모리 요청들이 존재한다는 것을 표시함 – 를 포함하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 29

제 21 항에 있어서,

메모리 요청들의 제 1 스트림에서 어써트된 디바이스 속성을 갖는 메모리 요청을 검출하고, 메모리 요청들의 제 2 스트림에서 하나 또는 그 초과의 계류중인 메모리 요청들을 검출하도록 구성된 트래킹 회로; 및

더 이상 계류중인 메모리 요청들이 존재하지 않는다는 것을 상기 트래킹 회로가 표시할 때까지 디바이스 메모리 요청을 지연시키도록 구성된 지연 회로를 더 포함하는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 30

제 21 항에 있어서,

상기 속성은, 메모리 영역이 훌수 어드레스 메모리 영역인지 또는 짹수 어드레스 메모리 영역인지를 결정하는 어드레스 비트인, 메모리 액세스들을 순서화하기 위한 장치.

청구항 31

제 21 항에 있어서,

상기 지연된 순서화된 메모리 액세스 요청은, 상기 계류중인 메모리 요청들이 순서대로 완료하도록 보장된다 는 확인응답들을 수신한 이후 프로세싱되는, 메모리 액세스들을 순서화하기 위한 장치.

청구항 32

컴퓨터 판독가능 프로그램 데이터 및 코드로 인코딩된 컴퓨터 판독가능 비-일시적인 매체로서,

상기 프로그램 데이터 및 코드는 실행될 경우,

어드레스 속성에 따라 단일 소스로부터의 메모리 요청들의 스트림을 메모리 요청들의 2개 또는 그 초과의 스트림들로 분할하고;

상기 메모리 요청들의 2개 또는 그 초과의 스트림들 중 하나에서 순서화된 메모리 액세스들을 요구하고 제 1 속성을 갖는 메모리 요청을 식별하고; 및

상기 메모리 요청들의 2개 또는 그 초과의 스트림들 중 상이한 스트림으로부터의 이전의 메모리 요청이 계류중이라고 결정할 시에 상기 순서화된 메모리 액세스들을 요구하고 제 2 어드레스 속성을 갖는 메모리 요청을 지연시키도록

동작 가능한, 컴퓨터 판독가능 비-일시적인 매체.

청구항 33

순서화된 메모리 액세스들을 요구하는 메모리 요청들을 순서화하기 위한 장치로서,

단일 소스로부터 메모리 요청들의 제 1 스트림을 모니터링하고, 각각의 메모리 요청의 어드레스 속성에 기초 하여 상기 메모리 요청들의 제 1 스트림을 메모리 요청들의 제 2 스트림 및 메모리 요청들의 제 3 스트림으로 분할하기 위한 수단;

제 2 인터리빙된 메모리 어드레스에서 계류중인 메모리 요청을 갖는 상기 메모리 요청들의 제 3 스트림과는 상이한 메모리 요청들의 스트림인 상기 메모리 요청들의 제 2 스트림으로부터 순서화된 메모리 액세스들을 요구하는 메모리 요청을 제 1 인터리빙된 메모리 어드레스에서 검출하기 위한 수단; 및

상기 계류중인 메모리 요청이 순서대로 완료하도록 보장될 때까지, 상기 계류중인 메모리 요청에 응답하여, 상기 순서화된 메모리 액세스들을 요구하는 메모리 요청들의 제 2 스트림을 지연시키기 위한 수단을 포함하는, 메모리 요청들을 순서화하기 위한 장치.

청구항 34

메모리 액세스들을 순서화하기 위한 장치로서,

2개 또는 그 초과의 대응하는 버스 포트들과 연관된 속성에 따라 상기 대응하는 버스 포트들에 안내된 메모리

요청들의 2개 또는 그 초과의 스트림들로, 메모리 요청들의 스트림을 인터리빙하기 위한 수단;

메모리 요청들의 제 1 스트림에서 순서화된 메모리 액세스를 요구하는 메모리 요청을 제 1 인터리빙된 메모리 어드레스에서 검출하고, 메모리 요청들의 제 2 스트림에서 적어도 하나의 계류중인 메모리 요청을 제 2 인터리빙된 메모리 어드레스에서 검출하기 위한 수단; 및

더 이상 계류중인 메모리 요청들이 존재하지 않는다는 표시가 존재할 때까지 상기 순서화된 메모리 액세스 요청을 지연시키기 위한 수단을 포함하는, 메모리 액세스들을 순서화하기 위한 장치.

명세서

기술분야

[0001]

본 발명은 일반적으로 프로세싱 시스템들에 관한 것으로, 더 상세하게는, 약하게 순서화된 저장 시스템에서 원자 동작들을 제어하기 위해 강하게 순서화된 동작들, 디바이스 메모리 동작들, 및 배타적인 액세스 요청들의 순서를 유지하기 위한 기술들에 관한 것이다.

배경기술

[0002]

셀 전화기들, 랩탑 컴퓨터들, 태블릿 개인용 컴퓨터(PC)들, 개인 휴대 정보 단말(PDA)들 등과 같은 많은 휴대용 제품들은, 통신 및 멀티미디어 애플리케이션들을 지원하는 프로그램들을 실행하는 하나 또는 그 초과의 프로세서들을 갖는 프로세싱 시스템을 포함한다. 그런 제품들에 대한 프로세싱 시스템은 또한, 명령들 및 데이터를 저장하기 위한 복합(complex) 메모리 시스템들, 제어기들, 및 하나 또는 그 초과의 버스들을 통해 프로세서들 및 메모리와 인터페이싱하도록 구성된 주변 디바이스들을 포함할 수도 있다.

[0003]

그러한 프로세싱 시스템들에서, 프로세서들은 종종, 메모리 동작들이 비순차적인 순서(out-of-order)로 수행되게 함으로써 성능 이점들을 획득한다. 예를 들어, 메모리 동작들의 시퀀스는, 메모리 내의 동일한 페이지에 대한 그들 동작들이 새로운 페이지가 개방되기 전에 실행되게 하도록 재순서화될 수도 있다. 메모리 동작들을 재순서화하도록 허용되는 프로세싱 시스템들은 일반적으로, "약하게 순서화된" 프로세싱 시스템들로 지칭된다.

[0004]

특정한 예시들에서, 메모리 동작들의 재순서화는 예측가능하지 않게 프로그램 작동에 영향을 줄 수도 있다. 예를 들어, 버스 마스터는, 이더넷 제어기와 같은 주변 디바이스의 FIFO(first in first out)로 데이터를 기입하며, 데이터가 송신될 수 있다는 것을 표시하는 동일한 주변 기기 내의 제어 레지스터로의 기입과 함께 최종 데이터 기입이 후속한다. 제어 레지스터로의 기입 및 데이터 레지스터로의 기입이 재순서화되면, 불완전한 패킷이 송신될 것이다.

[0005]

다른 예에서, 애플리케이션은, 제 1 메모리 어드레스에서 시작하여 데이터의 블록을 기입하고, 데이터의 블록이 저장된다는 것을 표시하는 데이터 준비 플래그(data ready flag)를 제 2 메모리 어드레스에 기입하도록 제 1 프로세서에게 요구할 수도 있다. 제 2 프로세서는, 데이터 준비 플래그를 모니터링하며, 데이터가 저장되었다는 것을 플래그가 표시할 경우, 데이터의 블록을 판독하는 것이다. 약하게 순서화된 프로세싱 시스템에서, 이러한 순서의 동작들이 발생할 것이라는 보장이 존재하지 않으며, 이는 수용가능하지 않을 수도 있다. 또한, 약하게 순서화된 프로세싱 시스템에서, 프로세서들에 부가하여, 주변 디바이스들은, 메모리로 데이터를 전송하고 메모리로부터 데이터를 수신하기 위한 버스 마스터들로서 동작할 수도 있으며, 이는 프로그램 저장 동작들의 순서를 유지하는 것을 추가적으로 복잡하게 할 수도 있다.

[0006]

약하게 순서화된 프로세싱 시스템에서 순서화된 메모리 동작들을 실행하기 위한 다양한 기술들이 이용되었다. 하나의 기술은, 순서화된 메모리 동작이 요구되는 경우 이슈되는 장벽 커맨드로 지정된 버스 커맨드를 사용하는 것이다. 장벽 커맨드는, 장벽 커맨드가 실행된 이후 프로세서에 의해 이슈된 임의의 메모리 액세스 요청들이 이전에, 장벽 커맨드 이전에 프로세서에 의해 이슈된 모든 메모리 액세스 요청들이 완료된다는 것을 보장하기 위해 사용될 수도 있다. 또한, 이전의 예에서, 장벽 커맨드는 데이터 준비 플래그를 기입하기 전에 제 1 프로세서에 의해 이슈될 수 있다. 이러한 장벽 커맨드는, 제 2 프로세서가 새로이 저장된 데이터를 판독한다는 것을 보장하는 데이터 준비 플래그가 기입되기 전에, 데이터의 블록이 메모리에 기입된다는 것을 보장할 것이다.

[0007]

몇몇 프로세서들, 주변 디바이스들, 및 버스 구현들은 장벽 커맨드들을 인식하지 못할 수도 있다. 장벽 커맨드들을 이용하는 약하게 순서화된 프로세싱 시스템들은, 그러한 프로세서들, 주변 디바이스들, 및 버스들을

사용하는 것이 제약될 것이다. 결과로서, 약하게 순서화된 프로세싱 시스템들은 효율적으로 수행되지 않을 수도 있다. 부가적으로, 장벽들을 지원하는 시스템에서의 장벽 커맨드의 실행은 일반적으로 비효율적이며, 매우 시간 소모적이다. 상기 예들에서, 데이터 및 플래그에 대한 기입들만이 순서화되도록 요구되지만, 장벽 커맨드는 데이터 및 플래그에 부가하여 모든 메모리 동작들을 순서화한다. 대안적인 접근법들은, 칩셋과 메모리 제어기 사이의 순서화된 트랜잭션들을 핸들링하기 위한 컴퓨터 네트워크를 설명하는 Willke의 미국 특허 제 6,754,751호를 포함한다(Willke 요약서). Willke는 또한, 확장자 브리지에 의해 수신된 모든 요청 트래픽이 그에 따라, 어느 업스트림 커맨드 요청 큐에 수신된 요청을 배치할지(예를 들어, A-측 또는 B-측)를 결정하기 위해 소스 식별자(identification)(ID)로 태그될 수도 있다는 것을 설명한다(Willke, col 5, line 18-22). 다른 접근법은, 제 2 메모리 제어기가 동일한 소스로부터의 임의의 계류중인 메모리 요청들을 포함하는지를 결정하기 위해, 메모리 요청이 발신되었던 곳을 표시하는 제 1 메모리 제어기 내의 메모리 요청에 대한 소스 태그를 제 2 메모리 제어기 내의 계류중인 메모리 요청들에 대한 소스 태그들과 비교하는 것을 설명하는 Jeddelloh 등의 미국 특허 제 6,275,913호에 의해 설명된다(Jeddelloh 요약서). 계류중인 메모리 요청이 제 2 메모리 제어기에서 발견되면, 제 1 메모리 제어기로부터의 메모리 요청은, 제 2 메모리 제어기 내의 동일한 소스로부터의 계류중인 메모리 요청들이 완료될 때까지, 이슈되는 것이 방지된다(Jeddelloh 요약서).

발명의 내용

[0008]

그의 수 개의 양상들 중에서, 본 발명은, 약하게 순서화된 저장 시스템에서 순서화된 액세스들을 제어하기 위한 더 효율적인 방법들 및 장치들을 제공하는 것이 프로세싱 시스템에서 성능을 개선시키고 전력 요건들을 감소시킬 수도 있음을 인식한다. 그러한 목적들을 위해, 본 발명의 실시예는, 순서화된 메모리 액세스들을 요구하는 메모리 요청들을 순서화하기 위한 방법을 어드레싱(address)한다. 메모리 요청들의 스트림은 메모리 요청들의 2개 또는 그 초과의 스트림들로 분할된다. 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 메모리 요청들의 2개 또는 그 초과의 스트림들 중 하나에서 식별된다. 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 메모리 요청들의 상이한 스트림으로부터의 이전의 메모리 요청이 계류중이라고 결정할 시에 지연(stall)된다.

[0009]

다른 실시예는, 순서화된 메모리 액세스들을 요구하는 메모리 요청들을 순서화하기 위한 장치를 어드레싱한다. 스트림 분할 회로는, 메모리 요청들의 제 1 스트림을 모니터링하도록 구성되고, 메모리 요청들의 제 2 스트림 및 메모리 요청들의 제 3 스트림으로 메모리 요청들의 제 1 스트림을 분할하도록 구성된다. 트래킹 회로는, 계류중인 메모리 요청을 갖는 메모리 요청들의 제 3 스트림과는 상이한 메모리 요청들의 스트림인 메모리 요청들의 제 2 스트림으로부터 순서화된 메모리 액세스들을 요구하는 메모리 요청을 검출하도록 구성된다. 지연 회로는, 계류중인 메모리 요청이 순서대로 완료하도록 보장될 때까지 계류중인 메모리 요청에 응답하여, 순서화된 메모리 액세스들을 요구하는 메모리 요청들의 제 2 스트림을 지연시키도록 구성된다.

[0010]

다른 실시예는 메모리 액세스들을 순서화하기 위한 장치를 어드레싱한다. 스위치 및 제어 회로는, 2개 또는 그 초과의 대응하는 버스 포트들과 연관된 속성에 따라 그 2개 또는 그 초과의 대응하는 버스 포트들로 안내된 메모리 요청들의 2개 또는 그 초과의 스트림들로 메모리 요청들의 스트림을 분할하도록 구성된다. 트래킹 (tracking) 회로는, 메모리 요청들의 제 1 스트림에서 순서화된 메모리 액세스를 요구하는 메모리 요청을 검출하고, 메모리 요청들의 제 2 스트림에서 하나 또는 그 초과의 계류중인 메모리 요청들을 검출하도록 구성된다. 지연 회로는, 계류중인 메모리 요청들이 더 이상 존재하지 않는다는 것을 트래킹 회로가 표시할 때까지, 순서화된 메모리 액세스 요청을 지연시키도록 구성된다.

[0011]

다른 실시예는, 시스템을 동작시키기 위한 컴퓨터 판독가능 프로그램 데이터 및 코드로 인코딩된 컴퓨터 판독 가능 비-일시적인 매체를 어드레싱한다. 메모리 요청들의 스트림은 메모리 요청들의 2개 또는 그 초과의 스트림들로 분할된다. 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 메모리 요청들의 2개 또는 그 초과의 스트림들 중 하나에서 식별된다. 순서화된 메모리 액세스들을 요구하는 메모리 요청은, 메모리 요청들의 상이한 스트림으로부터의 이전의 메모리 요청이 계류중이라고 결정할 시에, 지연된다.

[0012]

다른 실시예는, 순서화된 메모리 액세스들을 요구하는 메모리 요청들을 순서화하기 위한 장치를 어드레싱한다. 메모리 요청들의 제 1 스트림을 모니터링하고, 메모리 요청들의 제 2 스트림 및 메모리 요청들의 제 3 스트림으로 메모리 요청들의 제 1 스트림을 분할하기 위한 수단이 포함된다. 계류중인 메모리 요청을 갖는 메모리 요청들의 제 3 스트림과는 상이한 메모리 요청들의 스트림인 메모리 요청들의 제 2 스트림으로부터, 순서화된 메모리 액세스들을 요구하는 메모리 요청을 검출하기 위한 수단이 포함된다. 계류중인 메모리 요청이 순서대로 완료하도록 보장될 때까지 계류중인 메모리 요청에 응답하여, 순서화된 메모리 액세

스들을 요구하는 메모리 요청들의 제 2 스트림을 지연시키기 위한 수단이 포함된다.

[0013] 다른 실시예는 메모리 액세스들을 순서화하기 위한 장치를 어드레싱한다. 2개 또는 그 초과의 대응하는 버스 포트들과 연관된 속성에 따라 그 2개 또는 그 초과의 대응하는 버스 포트들로 안내된 메모리 요청들의 2개 또는 그 초과의 스트림들로 메모리 요청들의 스트림을 분할하기 위한 수단이 포함된다. 메모리 요청들의 제 1 스트림에서 순서화된 메모리 액세스를 요구하는 메모리 요청을 검출하고, 메모리 요청들의 제 2 스트림에서 하나 또는 그 초과의 계류중인 메모리 요청들을 검출하기 위한 수단이 포함된다. 계류중인 메모리 요청들이 더 이상 존재하지 않는다는 것을 트래킹 회로가 표시할 때까지, 순서화된 메모리 액세스 요청을 지연시키기 위한 수단이 포함된다.

[0014] 본 발명의 다른 실시예들이 다음의 상세한 설명으로부터 당업자들에게 용이하게 명백해질 것이며, 여기서, 본 발명의 다양한 실시예들은 예시에 의해 도시되고 설명됨을 이해한다. 본 발명이 다른 및 상이한 실시예들을 가능하게 하고, 그의 수 개의 세부사항들이 다양한 다른 관점들에서의 변경을 가능하게 하며, 그들 모두가 본 발명의 사상 및 범위를 벗어나지 않음을 인식할 것이다. 따라서, 도면들 및 상세한 설명은, 속성상 제한이 아니라 예시적인 것으로서 간주될 것이다.

[0015] 본 발명의 다양한 양상들은 첨부한 도면들에서 제한이 아닌 예로서 도시된다.

도면의 간단한 설명

[0016] 도 1은, 본 발명의 실시예가 유리하게 이용될 수도 있는 무선 통신 시스템을 도시한다.

도 2a는, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 동작하도록 구성된 예시적인 프로세싱 시스템을 도시한다.

도 2b는, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 강하게 순서화된 그리고 로드/저장 배타적인 동작들을 조정하도록 구성된 제 2 예시적인 프로세싱 시스템을 도시한다.

도 3은, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 동작하도록 구성된 예시적인 확장된 프로세싱 시스템을 도시한다.

도 4는, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 순서화된 액세스들을 제어하기 위한 예시적인 프로세스를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0017] 첨부된 도면들과 관련하여 아래에 기재된 상세한 설명은 본 발명의 다양한 예시적인 실시예들의 설명으로서 의도되며, 본 발명이 실시될 수도 있는 실시예들만을 표현하도록 의도되지 않는다. 상세한 설명은 본 발명의 완전한 이해를 제공하려는 목적을 위한 특정한 세부사항들을 포함한다. 그러나, 본 발명이 이를 특정한 세부사항들 없이도 실시될 수도 있다는 것은 당업자들에게는 명백할 것이다. 몇몇 예시들에서, 잘 알려진 구조들 및 컴포넌트들은 본 발명의 개념들을 불명료하게 하는 것을 회피하기 위해 블록도 형태로 도시된다.

[0018] 도 1은, 본 발명의 실시예가 유리하게 이용될 수도 있는 예시적인 무선 통신 시스템(100)을 도시한다. 예시의 목적들을 위해, 도 1은 3개의 원격 유닛들(120, 130, 및 150) 및 2개의 기지국들(140)을 도시한다. 일반적인 무선 통신 시스템들이 훨씬 더 많은 원격 유닛들 및 기지국들을 가질 수도 있음을 인식할 것이다. 각각 컴포넌트들(125A, 125C, 125B, 및 125D)에 의해 표현된 바와 같은 하드웨어 컴포넌트들, 소프트웨어 컴포넌트들, 또는 그 양자를 포함하는 원격 유닛들(120, 130, 및 150) 및 기지국들(140)은, 추가적으로 후술되는 바와 같은 본 발명을 구현하도록 적응되었다. 도 1은, 기지국들(140)로부터 원격 유닛들(120, 130, 및 150)로의 순방향 링크 신호들(180) 및 원격 유닛들(120, 130, 및 150)로부터 기지국들(140)로의 역방향 링크 신호들(190)을 도시한다.

[0019] 도 1에서, 원격 유닛(120)은 모바일 전화기로서 도시되고, 원격 유닛(130)은 휴대용 컴퓨터로서 도시되며, 원격 유닛(150)은 무선 로컬 루프 시스템 내의 고정 위치 원격 유닛으로서 도시된다. 예로서, 원격 유닛들은 대안적으로, 셀 전화기들, 페이저들, 위키 토키들, 핸드헬드 개인용 통신 시스템(PCS) 유닛들, 개인 휴대 정보 단말들과 같은 휴대용 데이터 유닛들, 또는 겸침(meter reading) 장비와 같은 고정 위치 데이터 유닛들일 수도 있다. 도 1이 본 발명의 교시들에 따른 원격 유닛들을 도시하지만, 본 발명은 이를 예시적인 도시된 유닛들로 제한되지 않는다. 본 발명의 실시예들은, 약하게 순서화된 저장 서브시스템으로 동작하는 임의의 프로세싱 시스템에서 적절히 이용될 수도 있다.

[0020]

메모리 동작들이 프로그램 순서로 실행된다는 것을 보장하기 위한 하나의 기술은, 그들 에이전트들이 사후-장벽(post-barrier) 명령들로부터 초래되는 메모리 액세스들을 관측하기 전에, 사전-장벽(pre-barrier) 명령들로부터 초래되는 모든 메모리 액세스들이 순서대로 완료하도록 확인응답된다는 것을 보장하는 메모리 장벽 명령을 사용하는 것이다. 예를 들어, 자신의 어드레스들이 특정된 메모리 영역 또는 메모리 블록 사이즈 내에 있는 그들 요청들에 대하여, 약하게 순서화된 메모리 시스템에서 강하게 순서화된(SO) 및 디바이스(DV) 메모리 동작들의 엄격한 프로그램 순서화를 보장하는 버스 아키텍처가 이용될 수도 있다. 더 상세히 후술되고 SO/DV 메모리 요청들로서 참조되는 바와 같이, 강하게 순서화된(SO) 요청 및 디바이스(DV) 메모리 요청은 유사한 방식으로 처리된다. 그러나, 그러한 시스템에서, 상이한 메모리 영역들에 대한 SO/DV 메모리 요청들은, 서로에 관해 순서화되도록 보장되지 않는다. 결과로서, 엄격한 프로그램 순서화가 정확한 동작을 보장하도록 요구되는 상이한 메모리 영역들에 대한 SO/DV 요청들 사이에 장벽 명령을 삽입함으로써 SO/DV 요청들의 순서화 제약을 강제하도록 소프트웨어가 요구된다. 그러나, 장벽 명령을 어디에 배치할지를 결정하는 것은 용이하지 않으며, 요구되는 장벽 명령들 없이 그리고 그들이 정확한 동작을 보장하도록 필요하지 않을 수도 있는 장벽 명령들을 이용하여 코드가 개발되는 것이 초래된다.

[0021]

프로세서는, 도 2a에 관해 후술되는 바와 같은 기술들을 사용하여 개선된 성능을 달성할 수도 있으며, SO/DV 메모리 요청들의 정확한 동작을 보장할 수도 있다. 도 2a는, 도 1의 컴포넌트들(125A, 125C, 125B, 및 125D)에서 적절히 이용될 수도 있는 약하게 순서화된 저장 시스템에서 동작하도록 구성된 예시적인 프로세싱 시스템(200)을 도시한다. 프로세싱 시스템(200)은 메인 메모리(202), 및 버스 브리지(204)를 통해 메인 메모리(202) 및 외부 디바이스들(242 및 244)에 커플링된 시스템 코어(203)를 포함한다. 시스템 코어(203)는, 프로세서(206), 레벨 1 명령 캐시(L1 I-캐시)(208), 레벨 1 데이터 캐시(L1 D-캐시)(210), 및 버스 및 메모리 관리 유닛(BMMU)(212)을 포함한다. BMMU(212)는, 마스터 포트 0(MP0)(216), 마스터 포트 1(MP1)(217), 레벨 2 통합 캐시(L2 캐시)(220), 및 경로 및 카운트 회로(222)를 포함한다. 경로 및 카운트 회로(222)는 스위치(224), 프로세서 요청들에 대한 카운터 회로(countP)(226), 제어 회로(228), 버퍼(bfr)(230), 및 약하게 순서화된 저장 시스템에서 순서화된 액세스들을 제어하기 위한 구성가능한 엘리먼트들로서의 외부 디바이스 요청들에 대한 카운터 회로(countE)(232)를 포함한다. 메인 메모리(202)는, 더블 데이터 레이트 0(DDR0) 메모리 디바이스(234) 및 DDR1 메모리 디바이스(236)로 구성된다. 외부 디바이스들은, DDRx 메모리와 같은 외부 디바이스 X(242), 및 버스 마스터로서 작동하고 내부 프로세서 및 로컬 메모리를 갖는 외부 디바이스 Y(244)를 포함한다. 버스 브리지(204)는, 경로(250)를 통해 MP0(216)로부터 DDR0 메모리 디바이스(234)로의 직접 경로, 및 경로(252)를 통해 MP1(217)으로부터 DDR1 메모리 디바이스(236)로의 직접 경로를 제공한다. 버스 브리지(204)는 또한, 더 상세히 후술될 바와 같이 예시적인 경로들(257-259)을 제공한다.

[0022]

프로세서(206) 및 외부 디바이스 Y(244)는 각각, 범용 프로세서 또는 멀티프로세서, 디지털 신호 프로세서(DSP), 애플리케이션 특정 프로세서(ASP) 등과 같은 버스 마스터링 디바이스로서 구현될 수도 있다. 외부 디바이스 Y(244)와 같은 외부 주변 디바이스는 메모리 디바이스, 메모리 제어기, 다른 버스 상호접속 디바이스에 상호접속시키기 위한 브리지 디바이스, 하드 디스크 제어기와 같은 주변 디바이스, 유니버설 시리얼 버스(USB) 제어기, 인터액티브(interactive) 디스플레이 디바이스, 송신기 및 수신기에 제어기를 커플링시키는 라디오 디바이스 등일 수도 있다. 외부 디바이스 Y(244)는 또한, 메인 메모리(202)에 대한 데이터를 판독 또는 기입하기 위한 직접 메모리 액세스(DMA) 기술들을 이용할 수도 있다.

[0023]

프로세서(206) 및 외부 디바이스 Y(244)는, 예를 들어, 비-일시적인 컴퓨터 판독가능 저장 매체 상에 저장된 프로그램의 제어 하에서 명령들을 실행하도록 구성될 수도 있다. 예를 들어, 컴퓨터 판독가능 저장 매체는, 명령 캐시들(208 및 220), 외부 디바이스 Y(244)와 연관된 로컬 메모리 중 하나를 통해 이용가능할 수도 있거나, 버스 브리지(204)를 통해 외부 디바이스 Y(244)에 액세스가능할 수도 있는 것과 같이, 프로세서(206) 또는 외부 디바이스 Y(244)와 직접 로컬적으로 연관될 수도 있다. 프로세서(206)는 프로그램의 실행 시에 메인 메모리(202)로부터의 데이터에 액세스할 수도 있다. 외부 디바이스 Y(244)는 또한, 예를 들어, 각각의 외부 디바이스와 로컬적으로 직접 연관되거나, DDR0 메모리 디바이스(234)와 같은 다른 프로세서 메모리 디바이스로부터 버스 브리지(204)를 통해 액세스가능한 메모리 디바이스에 상주하는 데이터에 액세스할 수도 있다.

[0024]

버스 브리지(204)는, 시스템 코어(203), 외부 디바이스 Y(244), 메인 메모리(202), 및 외부 디바이스 X(242)와 같은 다른 주변 디바이스들 사이의 버스 트래픽을 관리하고, 접속 경로들을 제공한다. 버스 브리지(204)는, 예를 들어, 각각 MP0(216)와 MP1(217) 그리고 DDR0 메모리 디바이스(234)와 DDR1 메모리 디바이스(236) 사이의 전용 채널들을 이용하여 구성될 수도 있다. 2개의 메모리 디바이스들, 즉 DDR0(234) 및 DDR1(236)이 도 2a에 도시되지만, 단일 메모리 디바이스는, 2개의 마스터 포트들 사이의 인터리빙된 액세스들을 이용하기

위해 그 2개의 마스터 포트들, 즉 MP0(216) 및 MP1(217)과 함께 사용될 수 있다. 또한, 인터리빙된 액세스가 사용되지 않더라도, 하나 또는 그 초과의 마스터 포트(MP)들이 사용될 수도 있음을 유의한다. 예를 들어, 메인 메모리(202)가 단일 포트된 메모리 디바이스로서 구현되면, 요청들은 MP0(216) 및 MP1(217)으로부터 단일 포트된 메모리 디바이스로 여전히 이슈될 수도 있다. 외부 디바이스 X(242)에 액세스할 경우 그리고 단일 포트된 메인 메모리 디바이스에 액세스할 경우 MP0(216) 및 MP1(217)로부터의 메모리 액세스 요청들이 재수령하므로, 메모리 요청들은, 본 발명의 기술들의 사용 없이 비순차적인 프로그램 순서로 획득될 수도 있다.

[0025] 브리지(204)가 메모리 요청을 확인응답할 경우, 확인응답은, 타겟팅된 외부 디바이스에 대한 요청이 요구된 순서대로 완료할 것이라는 것을 요청 디바이스에게 보장한다. 그러므로, SO/DV 메모리 요청이 MP0(216)를 통해 전달될 경우, MP0에 대한 요청이 그러한 확인응답을 수신할 때까지, MP1에 대한 임의의 새로운 SO/DV 요청들은 지연된다. 판독 요청들 및 기입 요청들에 대한 확인응답들이 상이하지만, 시스템이 적절한 응답을 대기함을 유의한다. 예를 들어, 브리지(204)로부터의 확인응답은 요청이 완료했다는 것을 프로세서(206)에게 표시하지 않을 수도 있지만, 외부 슬레이브 디바이스가, 그것이 MP1으로부터의 메모리 액세스를 관측하기 전에, MP0로부터의 메모리 액세스를 관측할 것이라는 것을 표시하기에는 충분하다. 몇몇 시스템들에서, 확인응답은 실제로는 완료 표시일 수도 있다. 본 발명의 구현에서, 확인응답은, 프로그램 순서화가 보장되는 시스템의 포인트에 메모리 요청이 도달했다는 보장을 표현한다. 예를 들어, BMMU(212)는 메모리 요청들의 스트림을 분할하며, 그 후, 요청 스트림들, 즉 경로들(257 및 258)은, 외부 디바이스 X(242), 또는 예를 들어, 단일 포트된 메인 메모리 디바이스 내의 DDR-X 메모리와 같은 외부 디바이스로 재수령된다. 경로들(257 및 258) 중 다른 하나에 대한 요청들을 지연시킴으로써 요청들이 경로들(257 및 258) 중 하나 상에서 제시될 경우, 스위치(224)에 진입했던 SO/DV 메모리 요청들의 순서는, 외부 디바이스 X(242) 내의 DDR-X 메모리에 도달하는 것과 동일하며, 예시적인 단일 포트된 메모리 디바이스에 도달하는 것과 또한 동일할 것이다. 요청이 BMMU(212)를 떠난 이후라도, 요청이 타겟팅된 메모리에서 실제로 수신되기 전에 몇몇 시간이 경과할 수도 있지만, BMMU(212)는 순서를 보존하기 위해 자신의 역할을 행한다.

[0026] 프로세싱 시스템(200)에서, 버스 및 메모리 관리 유닛(BMMU)(212)은, SO/DV 메모리 액세스 동작들의 정확한 순서화를 보장한다. 특히, 예시적인 디바이스들은 기입 데이터, 어드레스, 및 판독 데이터 버스들 및 연관된 제어 신호들을 이용할 수도 있다. 많은 시스템들에서, 결합된 기입 및 판독 데이터 버스가 사용될 수도 있다. 일반적으로, 별개의 어드레스, 판독 데이터, 및 기입 데이터 버스들은, 특정한 시스템의 요건들에 의존하여 특정한 버스 아키텍처, 또는 표준에 따른다. 메모리 액세스 요청은, 저장 명령의 실행에 응답하여 생성된 기입 또는 저장 요청, 로드 명령의 실행에 응답하여 생성된 판독 또는 로드 요청, 또는 버스 관련 요청들을 포함할 수도 있으며, 이들은 메모리 장벽 커맨드들을 포함할 수도 있다. 또한, 정의에 의해, SO/DV 요청은 비-캐시가능하다. 외부 마스터들에 의해 사용된 모든 버스 인터페이스들은, 메모리 장벽 커맨드가 외부 마스터에 의해 이슈되었더라도, 메모리 장벽 커맨드들을 지원 또는 인식하지는 않는다. 또한, 모든 디바이스들이 메모리 장벽 커맨드들을 지원하지는 않는다.

[0027] 제 1 예에서, 프로세서(206)는, bfr(230)에 버퍼링되고 스위치(224)에 의해 MP0(216) 또는 MP1(217)로 분할된 메인 메모리(202)에 대한 메모리 요청들의 스트림을 생성한다. 예를 들어, 특정된 메모리 영역 내의 위치에 대한 메모리 요청에 기초하여 메모리 요청들의 스트림을 다수의 스트림들로 분할함으로써, 프로세서에 의해 실행되는 코드에 메모리 장벽 커맨드들을 삽입하는 것과 같은 소프트웨어 기술들의 사용 없이, 스트림들 사이의 메모리 요청들이 자동적으로 순서화될 수도 있다. 예를 들어, 메모리 요청들의 스트림은, 용량 및 홀수 및 짹수 메모리 영역 어드레스들에 기초하여 분할될 수도 있으며, 홀수 및 짹수 메모리 영역들 사이의 메모리 요청들은 자동적으로 순서화될 수도 있다. 시스템 정의에 의해, DDR0 메모리 디바이스(234)는, 데이터의 메모리 영역이 1KB 용량을 갖는 데이터의 짹수 메모리 영역들을 저장하고, 메모리 요청 어드레스의 비트 10은, 메모리 요청이 짹수 메모리 영역 또는 홀수 메모리 영역 내의 데이터에 대한 것인지를 결정하기 위해 이용될 수도 있다. DDR1 메모리 디바이스(236)는, 또한 각각 1KB 용량의 데이터의 홀수 어드레싱된 메모리 영역들을 저장한다. 1KB 메모리 영역들에 대한 이러한 예시적인 시나리오에서, 제어 회로(228)는, 스위치(224)를 제어하기 위해 메모리 요청들의 스트림, 및 메모리 요청 어드레스들의 비트 10을 모니터링한다. 제 1 상태의 메모리 요청 어드레스의 비트 10에 관해, 짹수 메모리 영역에 대한 메모리 요청이 MP0(216)에 전달된다. 제 2 상태의 메모리 요청 어드레스의 비트 10에 관해, 홀수 메모리 영역에 대한 메모리 요청이 MP1(217)에 전달된다.

[0028] 다른 실시예에서, DDR0 메모리 디바이스(234)는, 데이터의 메모리 영역이 요청의 어드레스를 해시(hash)함으로써 정의된 데이터의 짹수 메모리 영역들을 저장하며, 해시의 결과는, 메모리 요청이 짹수 메모리 영역 또는 홀수 메모리 영역 내의 데이터에 대한 것인지를 결정하기 위해 이용될 수도 있다. DDR1 메모리 디바이스

(236)는 홀수의 해시된 메모리 영역들을 저장한다. 제어 회로(228)는, 스위치(224)를 제어하기 위해 메모리 요청들의 스트림을 모니터링하고, 메모리 요청 어드레스들의 해시를 계산한다. 제 1 상태의 메모리 요청 어드레스의 해시의 결과에 관해, 짹수 메모리 영역에 대한 메모리 요청이 MP0(216)에 전달된다. 제 2 상태의 메모리 요청 어드레스의 해시의 결과에 관해, 홀수 메모리 영역에 대한 메모리 요청은 MP1(217)에 전달된다.

[0029]

메모리 요청들의 스트림을 모니터링함으로써, 계류중인 요청들이 순서대로 완료하도록 보장된다는 표시가 수신될 때까지, 홀수 및 짹수 메모리 영역들 사이에서 변하는 요청들은 식별 및 지연될 수도 있다. 트래킹 접근법은, 각각의 메모리 요청의 수신 시에 충분하고, 요청이 순서대로 완료할 것이라는 것을 표시하는 요청에 대한 응답이 수신될 경우 감분하는 카운터(countP)(226)와 같은 카운터를 사용한다. 이전의 메모리 디바이스에 관해 계류중인 메모리 요청들이 존재하는 경우 DDR0 메모리 디바이스(234)와 DDR1 메모리 디바이스(236) 사이에서 변하는 새로운 SO/DV 메모리 요청들은, 계류중인 메모리 요청들이 완료하도록 보장될 때까지 지연된다. 지연된 메모리 요청은, 프로세싱되도록 인에이블될 때까지 계류중으로 유지된다. 홀수 및 짹수 메모리 영역들 사이에서 변하고, 지연되는 SO/DV 메모리 요청들은 카운터가 충분되게 하지 않는다. 계류중인 요청들이 완료되고 카운터가 제로와 같은 초기 카운트 값으로 적절히 감분된 이후, 지연된 요청이 프로세싱되고, 그 후, 지연된 요청이 프로세싱되는 동안 카운터가 충분된다. 메모리 요청들의 홀수 어드레스 스트림 내의 또는 메모리 요청들의 짹수 어드레스 스트림 내의 요청들이 상술된 확인응답들에 의해 순서대로 완료하도록 보장되므로, 스트림들 사이에서 변하는 요청들만이 반대의 스트림에 대한 요청들을 지연시키려는 목적을 위해 모니터링된다. 요청이 스트림들 사이의 변화를 표현한다고 결정하는 것은, 이전의 요청 또는 요청들의 시퀀스의 메모리 요청 어드레스 경계가 현재의 메모리 요청의 어드레스 경계와는 상이하다는 것을 표시하는 상태를 저장함으로써 달성될 수도 있다. 예를 들어, 홀수 어드레스 1K 메모리 영역들에 대한 하나 또는 그 초과의 요청들의 스트림은, 현재의 요청의 비트 10 내의 제로값과 비교할 경우, 카운터에서 계류중인 카운트가 존재한다면, 현재의 요청이 지연될 필요가 있을 수도 있다는 것을 표시하는 단일 어쩌트(assert)된 계류중인 스트림 비트에 의해 표현될 수도 있다. 일단 계류중인 요청들이 확인응답되고 카운터가 자신의 초기 상태로 리턴하면, 짹수 어드레스 1K 메모리 영역들에 대한 요청들이 프로세스 중이거나 계류중이라는 것을 표시하기 위해 지연된 요청이 프로세싱되고, 계류중인 스트림 비트가 디-어쩌트(de-assert)된다. 카운터(countP)(226)와 같은 카운터를 충분시키기 위해 SO/DV 요청이 요청들의 동일한 스트림 내에 있더라도, SO/DV 요청들이 모니터링된다.

[0030]

외부 디바이스 Y(244)와 같은 외부 마스터 디바이스는, SO/DV 요청들을 제어하며, DDR0 메모리 디바이스(234)에 대한 짹수 어드레스 요청들과 DDR1 메모리 디바이스(236)에 대한 홀수 어드레스 요청들 사이에서 변하는 요청들을 모니터링하기 위해 유사한 방식으로 동작하도록 구성된다. 외부 디바이스 Y(244)에 의해 이슈된 메모리 요청들의 스트림이 모니터링되며, 스위치(224)는, 제어 회로(228)가 DDR0 메모리 디바이스(234)에 짹수 어드레스 요청들을 전달하고, 짹수 어드레스 요청들이 계류중이라는 것을 표시하기 위한 상태를 저장하며, 카운터(countE)(232)를 충분시킴으로써 제어된다. 이전에 액세스된 메모리(예를 들어, DDR0 메모리 디바이스(234)) 내지 상이한 메모리(예를 들어, DDR1 메모리 디바이스(236)) 사이에서 변하는 외부 디바이스 Y(244)로부터의 요청들은, 제로와 같은 자신의 초기값으로 감분되는 countE 값에 의해 표시되는 바와 같이, 이전의 메모리에 대한 계류중인 요청들이 완료하도록 보장될 때까지 지연된다. 그 후, 저장된 상태는 DDR1 메모리 디바이스(236)에 대한 지연된 요청이 프로세싱되고 있다는 것을 표시하도록 어쩌트된다. 예시적인 경로(259)는, L2 캐시(220)에 캐싱되거나 외부 디바이스 또는 메인 메모리(202)를 타겟팅하는 BMMU(212)를 통해 안내되는 요청들에 대한 것이다.

[0031]

도 2b는, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 강하게 순서화된 및 로드/저장 배타적인 동작들을 조정하도록 구성된 예시적인 제 2 프로세싱 시스템(260)을 도시한다. 제 2 프로세싱 시스템(260)은, countP(226)과 같은 프로세서 요청들에 대한 카운터 회로가 하나의 카운터로부터 2 개의 카운터들로 확장되었다는 것을 제외하고, 도 2a의 프로세싱 시스템(200)과 유사하다. 순서화된 메모리 액세스들을 요구하는 메모리 요청들의 일 타입은, 원자 메모리 액세스들을 제공하는 메모리 액세스 배타적인 명령들의 시퀀스이다. 순서화된 메모리 액세스들을 요구하는 메모리 요청들의 제 2 타입은, 상술된 강하게 순서화된 또는 디바이스(SO/DV) 요청이다. 로드 배타적인 명령 또는 저장 배타적인 명령 카운터(countPx)(262) 및 SO/DV 카운터(countPs)(264)는, 요청 스트림이 타겟 디바이스에서 수렴하는 다수의 경로들로 분할되는 경우라도, 타겟 디바이스로의 요청 스트림에서 메모리 요청들의 본래 순서를 유지하기 위해 이용된다. countPx(262) 및 countPs(264) 카운터들은, 더 상세히 후술되는 바와 같이 제어 회로(266)에 의해 제어된다. 아래의 표 1은, 본 발명의 다양한 실시예들을 예시하기 위해 제 2 프로세싱 시스템(260)에 적용되는 강하게 순서화된 또는 로드/저장 배타적인 속성들을 갖는 메모리 요청들의 스트림을 도시한다.

표 1

	메모리 요청들	메모리 어드레스	버스 마스터 포트
1	LDREX	A	MPO
2	LDREX	B	MP1
3	SO/DV	C	MP1
4	STREX	B	MP1

[0032]

메모리 요청들은, 로드 배타적인(LDREX) 명령, 저장 배타적인(STREX) 명령, 및 강하게 순서화된 또는 디바이스(SO/DV) 메모리 요청을 포함한다. LDREX 및 STREX 명령들은, 록(lock)됨 또는 언록됨과 같은 메모리 값 또는 메모리 영역의 상태를 표현하는 플래그 또는 변수의 원자 액세스를 보장하는 메커니즘을 제공함으로써 신호장치(semaphore)들을 구현하기 위해 적절히 사용될 수도 있다. 표 1의 4개의 메모리 요청들은 간략히 설명되며, 그 후, 제 2 프로세싱 시스템(260)의 동작이 이러한 메모리 요청 스트림을 이용하여 설명된다.

[0034]

제 1 LDREX 명령은 버스 마스터 포트 MPO를 통해 안내되고, 메모리 어드레스 "A"로부터 값을 판독하고, 그것을 특정된 레지스터에 로드하며, 배타적인 모니터에 메모리 어드레스 "A"를 태그한다. 제 2 LDREX 명령은 MP1을 통해 안내되고, 메모리 어드레스 "B"로부터 제 2 값을 판독하고, 그것을 특정된 레지스터에 로드하며, 다른 배타적인 모니터에 메모리 어드레스 "B"를 태그한다. SO/DV 요청은, 메모리 어드레스 "C"의 데이터로의 액세스를 위하여 MP1을 통해 안내되고, 제 1 및 제 2 LDREX 명령들과는 별개로 동작한다. STREX 명령은 MP1을 통해 안내되고, 메모리 어드레스 "B"로 태그된 배타적인 모니터가 배타적인 상태에 여전히 있는 경우에만 메모리 어드레스 "B"에 데이터 값을 저장하며, 저장 동작이 성공적이었다면 표시를 리턴한다.

[0035]

카운터들(countPx(262) 및 countPs(264))은 초기 카운트 값으로 초기화된다. 제 1 LDREX 명령이 버스 마스터 포트 MPO를 통해 안내되는 경우, 카운터(countPx)(262)는 증분되며, MPO로서 선택된 포트를 표시하는 비트로 태그된다. 제 2 LDREX 명령이 버퍼(230)로부터 수신되는 경우, 그것은, 제 1 LDREX 명령이 자신의 초기 카운트 값이 아니라 countPx(262)로 여전히 계류중이므로, 버스 마스터 포트 MP1을 통해 안내되는 대신에 지연된다. 제어 회로(266)는, 버퍼(230)에서 미리보고(look ahead), SO/DV 요청을 식별하며, countPs(264)가 자신의 초기값에 있다는 것을 또한 검증한다. SO/DV 요청이 로드 및 저장 배타적인 명령들과는 별개로 동작할 수도 있으므로, 제어 회로(266)는 SO/DV 요청으로 하여금 버스 마스터 포트 MP1을 통해 안내되게 하고, 카운터(countPs)(264)를 증분시키며, MP1으로서 선택된 포트를 표시하는 비트로 태그된다. 제 1 LDREX 명령이 여전히 계류중이므로, 제 4 요청, 즉 STREX 명령이 또한 지연된다. 일단 제 1 LDREX 명령이 완료하도록 보장되는 확인응답이 수신되면, 카운터(countPx)(262)는 자신의 초기값으로 감분된다. 제 2 LDREX 명령은 릴리즈되고 버스 마스터 포트 MP1을 통해 안내되며, 선택된 포트가 MP1이라는 것을 표시하기 위해 countPx(262)가 증분 및 태그되게 한다. 동일한 포트를 통과하는 모든 요청들이 이슈된 순서대로 완료하도록 보장되므로, STREX 명령이 또한 릴리즈되고 버스 마스터 포트 MP1을 통해 안내된다.

[0036]

표 1에 리스트된 메모리 요청들의 동일한 스트림이 도 2a의 프로세싱 시스템(200)에서 수신될 수 있음을 유의한다. 단일 카운터(countP)(226)에 관해, 제 1 LDREX 명령에 대한 확인응답을 수신하여 계류중인 경우, SO/DV 요청이 지연될 것이다. 또한, 프로세싱 시스템(200)에서 도시된 바와 같이 다수의 버스 마스터 포트들로의 액세스를 갖는 프로세서가 대안적인 실시예에서, 전달들을 위한 다수의 버스 포트들의 이용가능도에 기초하여 메모리 요청 스트림을 분할할 수도 있음을 유의한다. 예를 들어, 도 2a에서, 2개의 버스 포트들(216 및 217) 중 하나가 전달될 계류중인 요청들로 백업된다고 프로세서(206)가 결정하면, 다른 버스 포트가 백업된 전달들에 대해 이용가능하다고 가정하면, 이를 백업된 전달들을 위해 다른 버스 포트를 사용하여 프로세서(206)가 선택적으로 시작할 수도 있다.

[0037]

도 3은, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 동작하도록 구성된 예시적인 확장된 프로세싱 시스템(300)을 도시한다. 확장된 프로세싱 시스템(300)은 도 2a에 도시된 프로세싱 시스템(200)의 확장이다. 확장된 프로세싱 시스템(300)의 메인 메모리(302)는, 도 2a의 프로세싱 시스템(200)의 메인 메모리(202)와 비교하여 N개의 메모리 엘리먼트들, 즉 메모리 디바이스들 DDR0(334₀), DDR1(334₁), ..., DDRN(334_N)으로 확장된다. 제 2 프로세서, 즉 프로세서2(306₂)는, 자신의 연관된 L1 I 캐시(308₂) 및 L1 D 캐시(310₂)를 포함하는 시스템 코어(303)에 부가된다. 제 2 프로세서와 연관된 것은 버퍼

(bfr)(330₂) 및 제 2 카운터(countP2)(326₂)이다.

[0038] 메인 메모리(302) 내의 하나 또는 그 초과의 메모리 엘리먼트들의 부가로 인해, 브리지(304)는 경로들(350₀, 350₁, ..., 350_N)에 의해 표시된 부가적인 메모리 채널들로 확장된다. 시스템 코어(303)는 또한, N개의 마스터 포트들 MPO(316₀), MP1(316₁), ..., MPN(316_N)로 확장된다. 이제, 스위치(324)는 N개의 마스터 포트들 (316₀-316_N)을 지원한다. 이들 부가들에 관해, 홀수/짝수 메모리 영역 어드레싱 방식은 메모리 영역들에 기초한 기술로 변경된다. 예를 들어, 1KB 메모리 영역들에 관해, 메모리 요청 어드레스 필드의 어드레스 비트들 11 및 10은 이제, 적절한 메모리 엘리먼트에 메모리 요청들을 안내하기 위해 사용된다. N이 4와 동일한 일 실시예에서, "00"과 동일한 어드레스 비트들 11 및 10을 갖는 메모리 요청들은 스위치(324)를 통해 DDR0(334₀)로 안내된다. "01"과 동일한 어드레스 비트들 11 및 10을 갖는 메모리 요청들은 스위치(324)를 통해 DDR1(334₁)로 안내된다. "10"과 동일한 어드레스 비트들 11 및 10을 갖는 메모리 요청들은 스위치(324)를 통해 DDR2(334₂)로 안내된다. "11"과 동일한 어드레스 비트들 11 및 10을 갖는 메모리 요청들은 스위치(324)를 통해 DDR3(334₃)으로 안내된다. 도 2a의 메인 메모리(202)가 3개의 메모리 디바이스들, 즉 DDR0, DDR1, 및 DDR2를 갖는 시스템과 같이 홀수개의 부가적인 메모리 디바이스들로 확장될 수도 있음을 유의한다. 본 발명의 기술들은, 적절한 어드레스 매핑 접근법을 이용하는 그러한 시스템에 적용가능하다.

[0039] 확장된 프로세싱 시스템(300)에 대한 트래킹 접근법은 프로세싱 시스템(200)에 대한 트래킹 접근법과 유사하다. 카운터(countP)(326₁)과 유사한 카운터, 즉 countP₂(326₂)가 사용된다. 카운터(countP2)(326₂)는 프로세서2(306₂)로부터의 각각의 메모리 요청의 수신 시에 충분하고, 요청이 순서대로 완료할 것이라는 것을 표시하는 요청에 대한 응답이 수신된 경우 감분한다. 각각의 부가적인 프로세서는 또한, 카운터(countP2)(326₂) 및 버퍼(bfr)(330₂)와 유사한 카운터 및 버퍼를 부가할 것이다. DDR0 메모리 디바이스(334₀)와 같은 제 1 DDR 메모리 디바이스 내지 계류중인 메모리 요청들이 이전의 메모리 디바이스와 연관된 나머지 메모리 디바이스들, 즉 DDR1 메모리 디바이스(334₁), 또는 DDR2 메모리(334₂), 또는 ..., DDRN 메모리(334_N) 중 하나와 같은 상이한 디바이스 사이에서 변하는 새로운 메모리 요청들은, 계류중인 메모리 요청들이 완료하도록 보장될 때까지 지연된다. 지연된 메모리 요청은, 프로세싱되도록 인에이블될 때까지 계류중으로 유지된다. 액세스되는 현재의 메모리 영역과는 상이한 메모리 영역을 타겟팅하는 메모리 요청들은 지연되고, 카운터를 충분시키지 않는다. 계류중인 요청들이 완료되고 카운터가 제로와 같은 초기 카운트 값으로 적절히 감분된 이후, 지연된 요청이 프로세싱을 위해 인에이블되며, 그 후, 카운터는 이전에 지연된 요청이 이제 프로세싱되기 위해 충분된다. 각각의 메모리 영역 내의 메모리 요청들이 순서대로 완료하도록 보장되므로, 상이한 메모리 영역들 사이에서 변하는 메모리 요청들만이 상이한 스트림에 대한 요청들을 지연시키려는 목적을 위해 모니터링된다. 카운터(countP1)(326₁)와 같은 카운터를 충분시키기 위해 SO/DV 요청이 요청들의 동일한 스트림 내에 있더라도 SO/DV 요청들이 모니터링된다.

[0040] BMMU(312)는 메모리 요청들의 스트림을 분할하며, 그 후, 경로들(357₀, 357₁, ..., 357_N)로부터 선택된 요청 스트림들은 외부 디바이스 X(342) 내의 DDR-X 메모리와 같은 외부 슬레이브 디바이스로 재수령된다. 경로들(357₀, 357₁, ..., 357_N) 중 다른 하나에 대한 요청들을 지연시킴으로써 요청들이 경로들(357₀, 357₁, ..., 357_N) 중 하나 상에서 제시되는 경우, 스위치(324)에 진입했던 SO/DV 메모리 요청들의 순서는, 외부 디바이스 X(342) 내의 DDR-X 메모리에 도달하는 것과 동일하다. 프로세서(306₁)로부터의 요청 스트림들이 프로세서(306₂)로부터의 요청 스트림들과는 별개로 모니터링되고, 그 역도 가능함을 유의한다.

[0041] 몇몇 구현들에서, 도 2a의 제어 블록(228)과 같은 제어 블록은 도 3의 확장된 프로세싱 시스템(300) 내의 2개의 제어 블록들로서 복제될 것이며, 각각의 복제된 제어 블록은 2개의 프로세서들(306₁ 및 306₂) 중 하나와 연관된다. 도 2b의 제 2 프로세싱 시스템(260)에 기초한 대안적인 실시예에서, countP1(326₁)은 countPx1으로 확장될 것이고, countPs1 및 countP2(326₂)는 countPx2 및 countPs2로 확장될 것이다. 제어 회로(328)는 도 2b에 관해 상술된 바와 같이, LDREX 및 STREX 명령들 및 SO/DV 요청들의 순서화된 제어를 지원하도록 적절히 확장된다. 각각의 프로세서의 요청들은 도 2b에 관해 상술된 바와 같이 제어된다. 이들 접근법들에서, N개의 CPU들에 대해, 상술된 바와 같은 기술들을 구현하도록 요구되는 하드웨어의 N개의 카피들이 존재한다.

[0042]

도 4는, 본 발명의 실시예가 유리하게 이용될 수도 있는 약하게 순서화된 저장 시스템에서 순서화된 액세스들을 제어하기 위한 예시적인 프로세스(400)를 도시한다. 블록(404)에서, 순서화된 액세스가 요구된다는 것을 표시하는 속성이 어쩌트된 속성을 갖는 도 2a 또는 2b의 프로세서(206) 또는 도 3의 프로세서(306₁)로부터의 메모리 액세스 요청은, 각각, 버스 및 메모리 관리 유닛(BMMU)(212) 또는 BMMU(312)와 같은 BMMU에서 수신된다. 결정 블록(406)에서, 수신된 순서화된 메모리 액세스 요청과는 상이한 메모리 엘리먼트 및 연관된 마스터 포트에 대한 하나 또는 그 초과의 계류중인 요청들이 존재하는지에 대한 결정이 제어 회로(228, 266, 또는 328)에서 행해진다. 상이한 메모리 엘리먼트에 대한 계류중인 순서화된 메모리 액세스 요청이 존재하면, 프로세스(400)는 블록(408)으로 진행한다. 블록(408)에서, 새로운 액세스 순서화된 메모리 액세스 요청은, 계류중인 순서화된 메모리 액세스 요청 또는 요청들이 완료하도록 보장될 때까지 지연된다. 그 후, 프로세스(400)는 블록(406)으로 리턴한다. 상이한 메모리 엘리먼트 및 연관된 마스터 포트에 대한 계류중인 요청들이 존재하지 않으면, 프로세스(400)는 블록(410)으로 진행한다. 블록(410)에서, 새로운 순서화된 메모리 액세스 요청이 프로세싱되고, 액세스 카운터가 증분된다. 블록(412)에서, 대응하는 응답이 새로운 순서화된 메모리 액세스에 대해 수신되었는지에 대한 결정이 행해진다. 현재 계류중인 새로운 순서화된 메모리 액세스 요청에 대한 어느 응답도 수신되지 않으면, 프로세스는 블록(412)에서 유지되며, 계류중인 순서화된 메모리 액세스 요청의 레졸루션(resolution)을 대기한다. 계류중인 순서화된 메모리 액세스 요청에 대한 응답이 수신되었다면, 프로세스(400)는 블록(414)으로 진행한다. 블록(414)에서, 프로세스(400)는 액세스 카운터를 감분시킨다. 블록(416)에서, 프로세스(400)는 그 순서화된 메모리 액세스 요청에 대해 종료한다.

[0043]

본 명세서에 기재된 실시예들과 관련하여 설명된 다양한 예시적인 로지컬 블록들, 모듈들, 회로들, 엘리먼트들, 또는 컴포넌트들은, 주문형 집적 회로(ASIC), 필드 프로그래밍가능 게이트 어레이(FPGA) 또는 다른 프로그래밍가능 로직 컴포넌트들, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트들, 또는 본 명세서에 설명된 기능들을 수행하도록 설계된 이들의 임의의 결합을 사용하여 구현될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안적으로, 프로세서는 임의의 종래의 프로세서, 특수 목적 제어기, 또는 마이크로-코딩된 제어기일 수도 있다. 시스템 코어는 또한, 컴퓨팅 컴포넌트들의 결합, 예를 들어, DSP와 마이크로프로세서의 결합, 복수의 마이크로프로세서들, DSP 코어와 결합한 하나 또는 그 초과의 마이크로프로세서들, 또는 원하는 애플리케이션에 적합한 임의의 다른 그러한 구성으로서 구현될 수도 있다.

[0044]

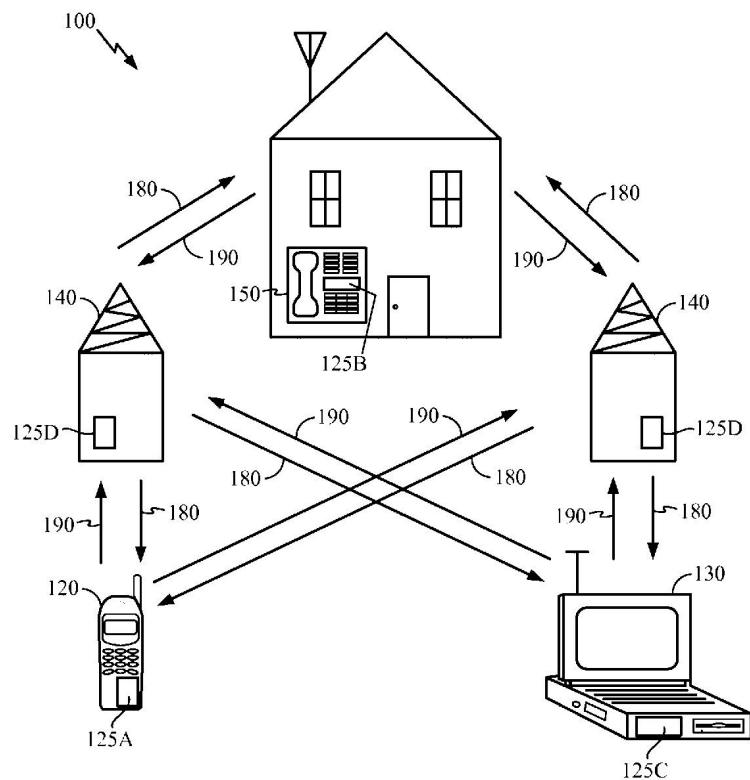
본 명세서에 기재된 실시예들과 관련하여 설명된 방법들은 하드웨어, 및 프로세서에 의해 실행되는 소프트웨어로 구현될 수도 있다. 프로세서(206)는, 예를 들어, 프로그램의 실행 시에 메인 메모리(202)로부터의 명령들 및 데이터에 액세스할 수도 있다. 하나 또는 그 초과의 프로세서 프로그램들을 포함하는 소프트웨어는 컴퓨터 판독가능 저장 매체 상에 비-일시적인 신호들로서 저장된다. 컴퓨터 판독가능 저장 매체는, 프로세서(206), 프로세서(306₁), 프로세서(306₂), 또는 외부 디바이스 Y(244)와 같은 주변 디바이스들 중 하나 내의 프로세서와 같은 프로세서와 직접 연관될 수도 있거나, 예를 들어, 버스 및 메모리 관리 유닛(BMMU)(212) 또는 다른 인터페이싱 수단을 통해 액세스가능할 수도 있다. 저장 매체는, 프로세서가 저장 매체로부터 정보를 판독할 수 있고 몇몇 경우들에서는 정보를 저장 매체에 기입할 수 있도록 프로세서에 커플링될 수도 있다. 프로세서에 커플링한 저장 매체는 회로 구현에 직접 커플링 통합될 수도 있거나, 하나 또는 그 초과의 인터페이스들을 이용할 수도 있어서, 직접적인 액세스들 또는 다운로딩 기술들을 사용하는 데이터 스트리밍을 지원한다. 컴퓨터 판독가능 저장 매체는 랜덤 액세스 메모리(RAM), 동적 랜덤 액세스 메모리(DRAM), 동기식 동적 랜덤 액세스 메모리(SDRAM), 플래시 메모리, 판독 전용 메모리(ROM), 프로그래밍가능 판독 전용 메모리(PROM), 소거가능한 프로그래밍가능 판독 전용 메모리(EPROM), 전기적으로 소거가능한 프로그래밍가능 판독 전용 메모리(EEPROM), 컴팩트 디스크(CD), 디지털 비디오 디스크(DVD), 다른 타입들의 착탈형 디스크들, 또는 임의의 다른 적절한 저장 매체를 포함할 수도 있다.

[0045]

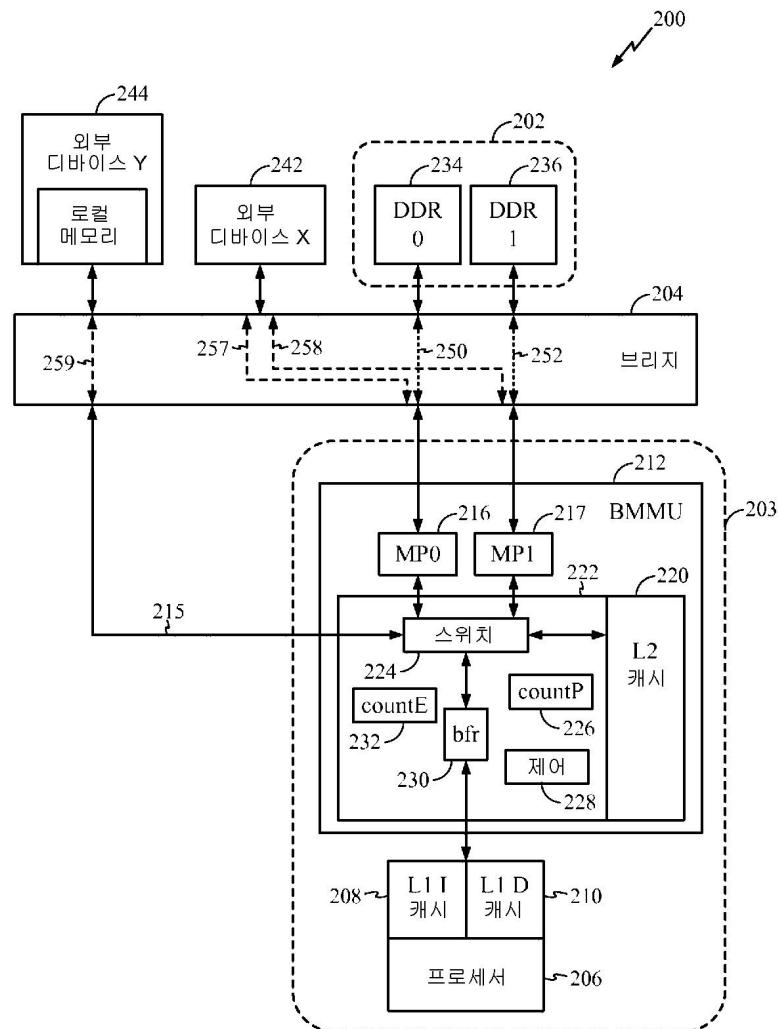
본 발명이 프로세서 시스템들에서의 사용을 위해 예시적인 실시예들의 맥락에서 기재되지만, 광범위하게 다양한 구현들이 상기 설명 및 아래에 후속하는 청구항들에 부합하여 당업자들에 의해 이용될 수도 있음을 인식할 것이다. 예를 들어, 고정 기능 구현들은 또한 본 발명의 다양한 실시예들을 이용할 수도 있다.

도면

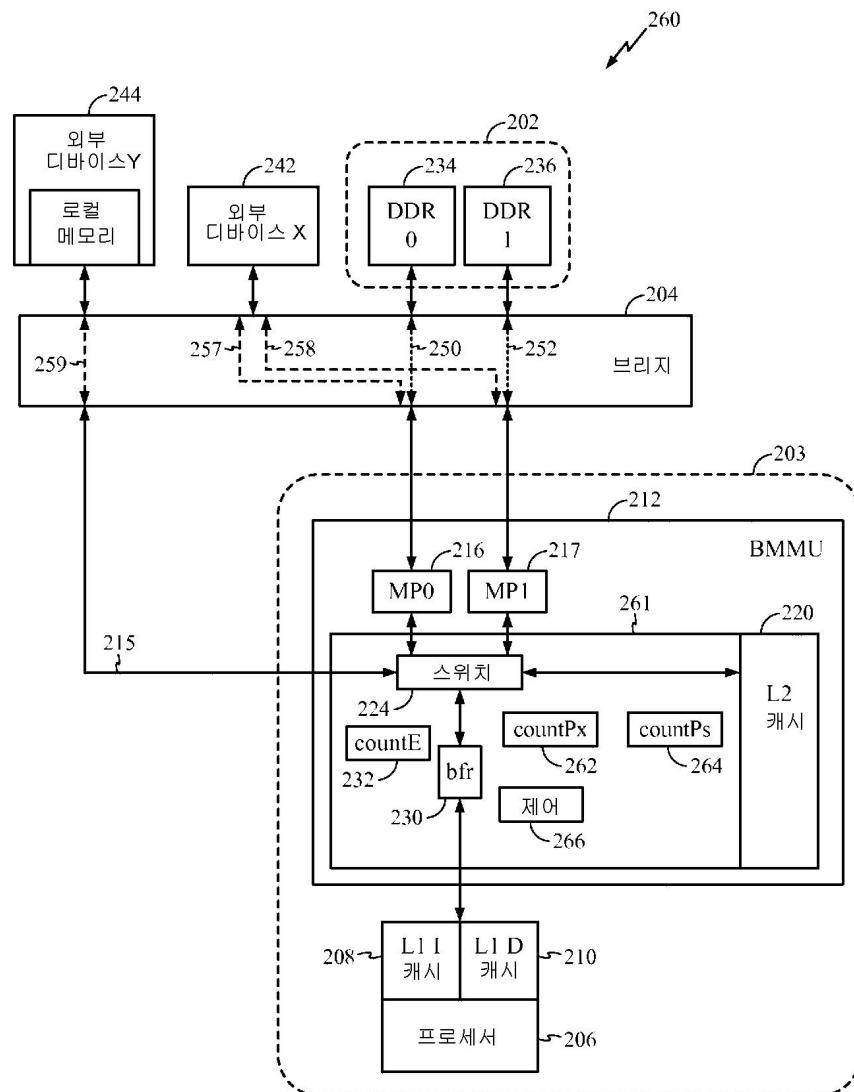
도면1



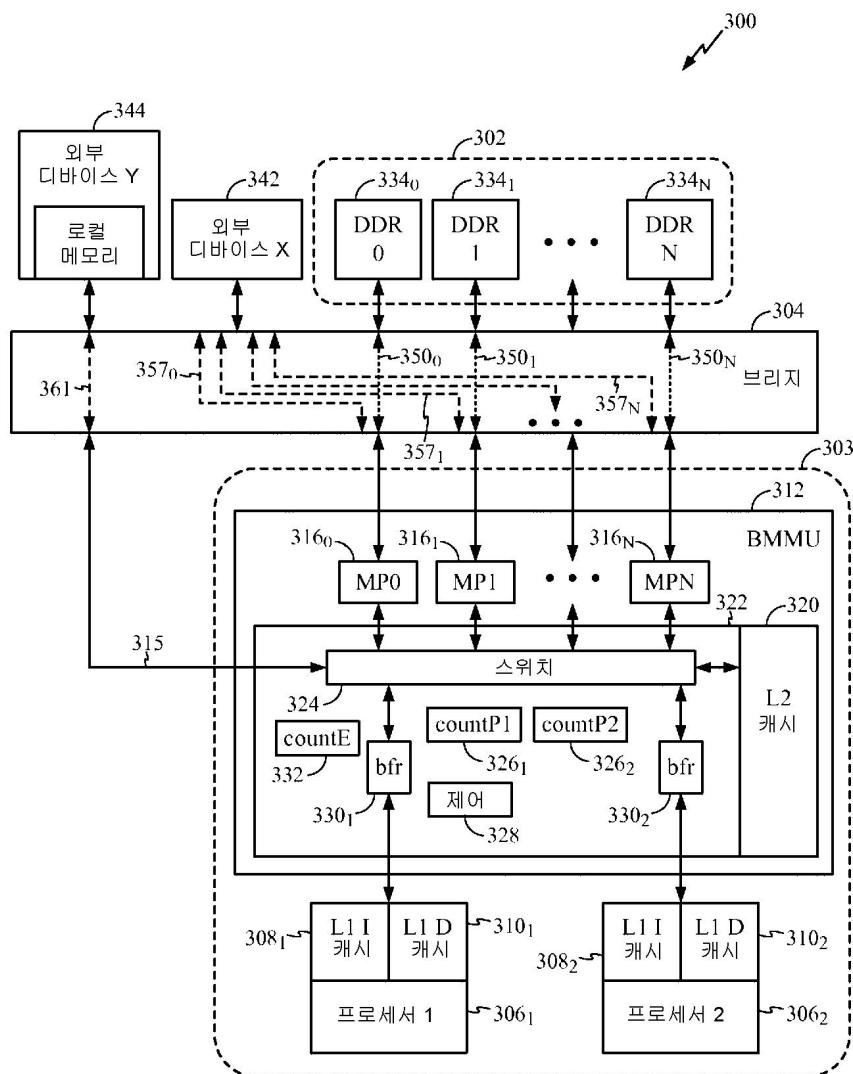
도면2a



도면2b



도면3



도면4

