



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년05월08일
(11) 등록번호 10-1262313
(24) 등록일자 2013년05월02일

(51) 국제특허분류(Int. Cl.)
G06F 13/16 (2006.01) G06F 13/14 (2006.01)
(21) 출원번호 10-2008-7008617
(22) 출원일자(국제) 2006년08월23일
심사청구일자 2011년08월23일
(85) 번역문제출일자 2008년04월10일
(65) 공개번호 10-2008-0043406
(43) 공개일자 2008년05월16일
(86) 국제출원번호 PCT/US2006/032742
(87) 국제공개번호 WO 2007/032866
국제공개일자 2007년03월22일
(30) 우선권주장
11/222,615 2005년09월09일 미국(US)
(56) 선행기술조사문헌
US20030070018 A1
US20030185032 A1
전체 청구항 수 : 총 8 항

(73) 특허권자
어드밴스드 마이크로 디바이시스, 인코포레이티드
미국 캘리포니아 94088-3453 서니베일 원 에이엠 디 플레이스 메일 스톱68
(72) 발명자
티셴러 브렛 에이.
미국 콜로라도 80501 롱몬트 선라이트 드라이브 1921
(74) 대리인
박장원

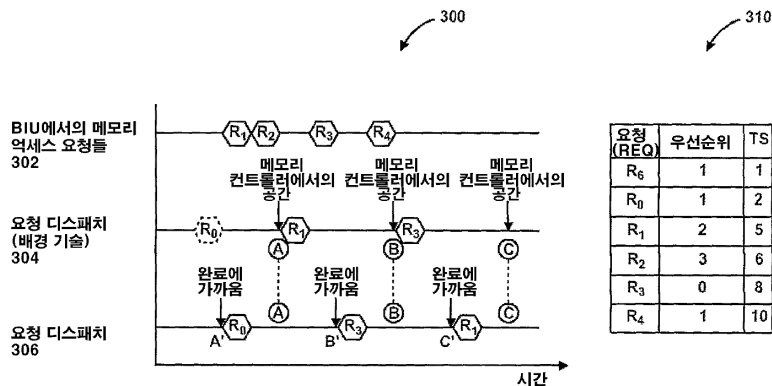
심사관 : 고재용

(54) 발명의 명칭 지연된 메모리 액세스 요청 조정

(57) 요약

지연된(delayed) 메모리 액세스 요청 조정(arbitration) 방법은 제 1 메모리 액세스 요청을 메모리 컨트롤러(102)에 디스패치하고, 상기 제 1 메모리 액세스 요청에 의해 나타나는 메모리 액세스 동작의 예측된 완료에 대한 응답으로 제 2 메모리 액세스 요청을 메모리 컨트롤러(102)에 디스패칭하는 과정을 포함한다. 또 다른 방법은 제 1 메모리 액세스 요청을 제 1 시각(time)에 버스 인터페이스 유닛에서 수신하는 단계와, 제 1 시각 다음인 제 2 시각에 제 2 메모리 액세스 요청을 메모리 컨트롤러에 디스패칭하는 단계와, 제 2 시각 다음인 제 3 시각에 버스 인터페이스 유닛에서 제 3 메모리 액세스 요청을 수신하는 단계와, 제 3 시각 다음인 제 4 시각에 제 3 메모리 액세스 요청을 메모리 컨트롤러에 디스패칭하고 제 4 시각 다음인 제 5 시각에 제 1 메모리 액세스 요청을 메모리 컨트롤러에 디스패칭하는 단계를 포함한다.

대표도



특허청구의 범위

청구항 1

제 1 메모리 액세스 요청(memory access request)을 메모리 컨트롤러(memory controller)(102)에 디스패치(dispatch)하는 단계와;

제 2 메모리 액세스 요청을 상기 메모리 컨트롤러(102)에 디스패치하는 단계와;

상기 제 2 메모리 액세스 요청에 근거하여 상기 메모리 컨트롤러(102)에 의한 제 1 메모리 액세스 동작을 개시하는 단계와; 그리고

상기 메모리 컨트롤러(102)가 상기 제 1 메모리 액세스 동작을 개시한 것에 응답하여 상기 메모리 컨트롤러(102)가 다음 메모리 액세스 요청을 수신할 수 있는 가장 빠른 시간 이후 상기 제 1 메모리 액세스 동작의 실제 완료 이전의 소정의 지속구간(predetermined duration)에서 상기 메모리 컨트롤러(102)가 상기 제 1 메모리 액세스 요청에 근거하여 제 2 메모리 액세스 동작을 개시하기 전에 제 3 메모리 액세스 요청을 상기 메모리 컨트롤러(102)에 디스패치하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 2

제1항에 있어서,

상기 메모리 컨트롤러(102)가 다음 메모리 액세스 요청을 수신할 수 있는 상기 가장 빠른 시간은 메모리(104)와 상기 메모리 컨트롤러(102) 간의 시그널링(signaling)에 의해 나타내지는 것을 특징으로 하는 방법.

청구항 3

제2항에 있어서,

상기 제 2 메모리 액세스 요청은, 읽기 액세스 요청(read access request)과 쓰기 액세스 요청(write access request) 중 하나를 포함하고,

상기 읽기 액세스 요청의 경우, 상기 시그널링에는 요청된 데이터가 상기 메모리(104)로부터 상기 메모리 컨트롤러(102)로의 송신(transmission)에 대해 이용가능함을 나타내는 신호가 포함되고; 그리고

상기 쓰기 액세스 요청의 경우, 상기 시그널링에는 쓰기 데이터가 상기 메모리(104)에 제공되고 있음을 나타내는 신호가 포함되는 것을 특징으로 하는 방법.

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 제 3 메모리 액세스 요청을 상기 메모리 컨트롤러(102)에 디스패치하는 단계는, 상기 메모리 컨트롤러(102)에 제공될 이용가능한 복수 개의 메모리 액세스 요청들로부터 상기 제 3 메모리 액세스 요청을 선택하는 것을 포함하는 것을 특징으로 하는 방법.

청구항 6

제2항에 있어서,

상기 제 3 메모리 액세스 요청을 디스패치하는 단계는, 상기 메모리 컨트롤러(102)가 다음 메모리 액세스 요청을 수신할 수 있는 상기 가장 빠른 시간에 응답하여 상기 메모리 컨트롤러(102)로부터 버스 인터페이스 유닛(bus interface unit)(106)으로 신호(134)를 제공하는 것을 포함하고, 여기서 상기 버스 인터페이스 유닛(106)은 상기 신호(134)를 수신하는 것에 응답하여 복수 개의 메모리 액세스 요청들로부터 상기 제 3 메모리 액세스 요청을 선택하는 것을 특징으로 하는 방법.

청구항 7

시스템(100)으로서,

버스 인터페이스 유닛(106)과; 그리고

상기 버스 인터페이스 유닛(106)에 연결됨과 아울러 메모리(104)에 연결된 메모리 컨트롤러(102)를 포함하여 구성되고,

상기 버스 인터페이스 유닛(106)은,

하나 이상의 메모리 액세스 요청들을 저장하기 위한 버퍼(132)와, 그리고

제 1 입력 및 제 2 입력을 갖는 조정기(130)를 포함하고,

여기서 상기 조정기(130)의 제 1 입력은 상기 버퍼(132)에 연결되어 메모리 액세스 요청들을 수신하고, 상기 조정기(130)의 제 2 입력은 신호(134)를 수신하며, 상기 조정기(130)는 상기 신호(134)에 응답하여 디스패치를 위한 다음 메모리 액세스 요청을 선택하고,

상기 메모리 컨트롤러(102)는 상기 버스 인터페이스 유닛(106)으로부터 디스패치된 복수 개의 메모리 액세스 요청들을 저장하는 버퍼(120)를 포함하고, 상기 버스 인터페이스 유닛(106)으로부터 디스패치된 복수 개의 메모리 액세스 요청들은 우선순위(priority)가 가장 높은 메모리 액세스 요청을 포함하고, 상기 우선순위가 가장 높은 메모리 액세스 요청은 우선순위가 낮은 메모리 액세스 요청 이후에 상기 메모리 컨트롤러(102)의 버퍼(120)에 수신되며,

상기 메모리 컨트롤러(102)는 상기 우선순위가 낮은 메모리 액세스 요청에 근거하여 메모리 액세스 동작을 개시하기 전에 상기 우선순위가 가장 높은 메모리 액세스 요청에 근거하여 현재 메모리 액세스 동작을 개시하며,

상기 메모리 컨트롤러(102)는, 상기 현재 메모리 액세스 동작의 개시에 응답하여 상기 메모리 컨트롤러(102)의 버퍼(120)에서 버퍼 위치가 이용가능하게 된 이후 상기 현재 메모리 액세스 동작의 실제 완료 이전의 소정의 지속구간에 상기 신호(134)를 상기 버스 인터페이스 유닛(106)에 제공하는 것을 특징으로 하는 시스템.

청구항 8

제7항에 있어서,

상기 현재 메모리 액세스 요청은, 읽기 액세스 요청과 쓰기 액세스 요청 중 하나를 포함하고,

상기 읽기 액세스 요청의 경우, 상기 신호(134)는 요청된 데이터가 상기 메모리(104)로부터 상기 메모리 컨트롤러(102)로의 송신에 대해 이용가능함을 나타내는 표시에 응답하여 제공되고; 그리고

상기 쓰기 액세스 요청의 경우, 상기 신호(134)는 쓰기 데이터가 상기 메모리(104)에 제공되고 있음을 나타내는 표시에 응답하여 제공되는 것을 특징으로 하는 시스템.

청구항 9

삭제

청구항 10

제7항에 있어서,

상기 조정기(130)는 상기 하나 이상의 메모리 액세스 요청들과 관련된 우선순위에 근거하여 상기 다음 메모리 액세스 요청을 선택하는 것을 특징으로 하는 시스템.

명세서

기술분야

[0001] 본 발명의 개시사항은 일반적으로 메모리 액세스 요청들을 프로세싱하는 것에 관한 것이다.

배경기술

[0002] 메모리 컨트롤러는 프로세싱 시스템들에서 자주 사용되는데, 메모리에 데이터를 저장하거나 메모리로부터의 데이터에 액세스하려는 소자의 메모리 리소스들에의 액세스를 제어하도록 사용된다. 종래의 시스템들에서, 메모리 액세스 요청들은 메모리 컨트롤러가 메모리 액세스 요청들을 받아들일 수 있게 되는대로 메모리 컨트롤러에 공급된다. 그 다음 메모리 컨트롤러는 전형적으로 메모리 액세스 요청들을 버퍼링하고, 할당된 특정 우선순위들(assigned priorities)에 기초하여 버퍼링된 메모리 액세스 요청들을 프로세싱한다. 그러나, 메모리 컨트롤러가 다음(next) 메모리 액세스 요청을 프로세싱하거나 버퍼링할 능력이 있게 되자마자 메모리 액세스 요청들을 메모리 컨트롤러에 디스패치(dispatch)하는 종래의 기술로 인하여, 메모리 컨트롤러가 우선 다음 메모리 액세스 요청을 수신할 능력을 갖추게 된 이후에 특정한 높은 우선순위의 메모리 액세스 요청들이 선택이 가능하게 된 때에는, 그 높은 우선순위의 메모리 액세스 요청들은 가장 효율적인 방식으로 프로세싱되지 못할 수 있다. 예를 들어 설명해 보면, 메모리 컨트롤러가 다음 요청을 받아들일 가능성으로 인해 높은 우선순위의 메모리 액세스 요청이 낮은 우선순위의 메모리 액세스 요청의 디스패치 직후에 수신된 경우에는, 높은 우선순위 메모리 액세스 요청의 프로세싱은 메모리 컨트롤러가 다시 다음 메모리 액세스 요청을 받아들일 능력이 될 때까지 지연될 수 있다.

[0003] 따라서, 메모리 액세스 요청들을 프로세싱하는 향상된 기술이 유익할 것이다.

발명의 상세한 설명

삭제

실시예

[0018] 하기의 설명은 메모리 액세스 요청들의 조정(arbitration)에 관한 여러 개의 구체적인 실시예들과 세부사항들을 제공함에 의해 본 개시사항에 대한 철저한 이해를 전달하기 위한 것이다. 그러나, 본 개시사항은 이러한 구체적 실시예들과 세부사항들로 한정되는 것이 아니고, 단지 예시적인 것일 뿐임이 이해되어야 한다. 또한 알려진 시스템들과 방법들의 관점에서, 당업자는 특정 디자인 및 다른 필요들에 따라 임의 개수의 대안적 실시예들로 본 개시사항의 용도를 의도된 목적 및 이점들에 맞게 인식할 수 있음이 이해되어야 한다.

[0019] 본 개시사항의 일 실시형태에 따르면, 지연된 메모리 액세스 요청 조정 방법은 제 1 메모리 액세스 요청을 메모리 컨트롤러에 디스패치하고, 제 1 메모리 액세스 요청에 의해 나타난 메모리 액세스 동작의 예측된 완료에 대한 응답으로 제 2 메모리 액세스 요청을 메모리 컨트롤러에 디스패치하는 단계를 포함한다.

[0020] 본 개시사항의 또 다른 실시형태에 따르면, 지연된 메모리 액세스 요청 조정 방법은 제 1 메모리 액세스 요청을 제 1 시각(time)에 버스 인터페이스 유닛(bus interface unit)에서 수신하는 단계와, 제 1 시각 다음인 제 2 시각에 제 2 메모리 액세스 요청을 메모리 컨트롤러에 디스패치하는 단계와, 제 2 시각 다음인 제 3 시각에 버스 인터페이스 유닛에서 제 3 메모리 액세스 요청을 수신하는 단계와, 제 3 시각 다음인 제 4 시각에 제 3 메모리 액세스 요청을 메모리 컨트롤러에 디스패치하고 제 4 시각 다음인 제 5 시각에 제 1 메모리 액세스 요청을 메모리 컨트롤러에 디스패치하는 단계를 포함한다.

[0021] 본 개시사항의 추가적인 실시형태에 따르면, 시스템은 메모리에 연결된 것으로서 메모리 액세스 요청들을 프로세싱하기 위한 메모리 컨트롤러와, 이 메모리 컨트롤러에 연결된 버스 인터페이스 유닛을 포함한다. 일 실시예에서, 버스 인터페이스 유닛은 하나 또는 그 이상의 메모리 액세스 요청들을 저장하기 위한 버퍼와, 메모리 액세스 요청들을 수신하기 위해 버퍼에 연결된 제 1 입력 및 메모리 컨트롤러로부터의 신호를 수신하기 위한 제 2 입력을 갖는 조정기(arbiter)로 구성되고, 상기 조정기는 상기 신호에 대한 응답으로 메모리 컨트롤러에 디스패치하기 위해 제 2 메모리 액세스 요청을 선택한다. 메모리 컨트롤러는 메모리 컨트롤러에 먼저 디스패치된 제 1 메모리 액세스 요청에 의해 나타나는 메모리 액세스 동작의 예측된 완료에 대한 응답으로 상기 신호를 공급한다.

[0022] 도 1을 참조해 보면, 본 개시사항의 적어도 하나의 실시예에 따라, 지연된 메모리 액세스 요청 디스패치를 사용하는 예시적인 프로세싱 시스템(100)이 도시되어 있다. 도시된 예에서, 시스템(100)은 메모리 컨트롤러(102), 메모리(104)(예컨대 DRAM 또는 SRAM), 버스 인터페이스 유닛(BIU)(106)과 BIU(106)에 연결된 하나 또는 그 이상의 구성요소들, 예를 들어 중앙 처리 유닛(CPU)(108), 그래픽 엔진(110)과 하나 또는 그 이상의 장치들(114, 116)에 연결된 하나 또는 그 이상의 직접 메모리 액세스(DMA) 마스터(112)와 같은 구성요소들을 포함한다. 일 실시예에서, 상기 메모리 컨트롤러(102)는 하나 또는 그 이상의 큐 엔트리들(예컨대 큐 엔트리들(122, 124))을 갖는 메모리 액세스 요청 큐(120)와, 선택기(126) 그리고 스케줄 관리자(schedule manager)(128)를 포함한다.

일 실시예에서, BIU(106)는 BIU(106)에 연결된, CPU(108), 그래픽 엔진(110), DMA 마스터(112)와 같은 장치들로부터의 메모리 액세스 요청들을 버퍼링하기 위해 조정기(130)와 메모리 액세스 요청 큐(132)를 포함한다. 일 실시예에서, 조정기(130)는 스케줄 관리기(128) 또는 메모리 컨트롤러(102)의 또 다른 구성요소에 의해 공급되는 디스패치_req 신호(134)에 응답한다.

[0023] 동작에 있어서, BIU(106)는 BIU(106)에 연결된 하나 또는 그 이상의 장치들로부터 메모리 액세스 요청들을 수신하고 큐(132)에서 메모리 액세스 요청들을 버퍼링한다. 메모리 액세스 요청들에 할당된 우선순위들 또는 메모리 액세스 요청들이 BIU(106)에서 수신되는 시퀀스와 같은 다양한 특성들에 기초하여, 조정기(130)는 프로세싱을 위해 큐(132)로부터의 메모리 액세스 요청을 선택하고 메모리 컨트롤러(102)에 요청을 디스패치한다. 큐(132)에서 버퍼링된 메모리 액세스 요청들은 스케줄 관리기(128)에 의해 결정된 스케줄 또는 우선순위와 같은 메모리 액세스 요청들의 특정한 특성에 기초하여 순서대로 또는 번갈아 순서에 관계없이(선택기(126)를 이용하여) 프로세싱된다. 선택된 메모리 액세스 요청을 수신함에 있어서, 스케줄 관리기(128)는 선택된 메모리 액세스 요청을 선택된 메모리 액세스 요청에 의해 나타나는 메모리 액세스 동작을 수행하도록 프로세싱한다. 예를 들어, 이 메모리 액세스 동작은 데이터가 메모리(104)에 쓰여지는(written) 쓰기 동작 또는 데이터가 메모리(104)로부터 읽혀지는(read) 읽기 동작을 포함할 수 있다.

[0024] 스케줄 관리기(128)가 선택된 메모리 액세스 요청의 프로세싱을 개시할 때, 프로세싱되고 있는 선택된 메모리 액세스 요청에 의해 비워진(vacated) 큐 엔트리들(122, 124) 중 하나에서 또는 직접적으로 선택기(126)를 통해, 메모리 컨트롤러(102)는 BIU(106)로부터 또 다른 메모리 액세스 요청을 수신할 수 있게 된다. 그러나, 일 실시예에서, 스케줄 관리기(128)는 메모리 컨트롤러(102)가 BIU(106)로부터 다음 메모리 액세스 요청을 수신할 수 있게 되자마자 조정기(130)가 디스패치를 위한 다음 메모리 액세스 요청을 선택하도록 지시하기보다는, 현재 메모리(104)를 사용하여 프로세싱되고 있는 메모리 액세스 동작의 완료가 가까워졌을 때까지(다시 말해, 메모리 액세스 동작의 예측된 완료시까지) 다음 메모리 액세스 요청을 위해 조정기(130)에 대한 시그널링(signaling)을 지연시킨다. 현재 메모리 액세스 동작의 완료가 잘 알려졌는지를 판단함에 있어서, 스케줄 관리기(128)는 디스패치_req 신호(134)를 어서트(assert)하고, 그에 의해 다음 메모리 액세스 요청을 메모리 컨트롤러(102)에 디스패치하도록 조정기(130)에 시그널링을 행한다. 상기 신호(134)에 대한 응답으로, 조정기(130)는 우선순위 또는 시간 스탬프들과 같은 하나 또는 그 이상의 특성들에 기초하여 큐(132)로부터 메모리 액세스 요청을 선택하고, 임시의 버퍼링과 후속되는 프로세싱을 위해 메모리 컨트롤러(102)에 선택된 메모리 액세스 요청을 디스패치한다. 그래서, 도 1의 프로세싱 시스템(100)은 종래의 시스템들에 의해 시행되던 것처럼 메모리 컨트롤러(102)에서 메모리 액세스 요청의 시작에 뒤이어 즉시 메모리 액세스 요청을 디스패치하도록 조정기(130)에 지시하기보다는, 지연된 조정 기술을 구현하고 그것에 의해, 메모리 컨트롤러(102)가 메모리 액세스 요청을 받아들일 수 있는 제 1 모멘트 후에, BIU(106)에서 수신되는 어느 높은 우선순위의 메모리 액세스 요청들이 조정기(130)에 의해 선택가능하고 메모리 컨트롤러(102)에 디스패치됨이 가능하게 하기 위하여 하나의 다음 메모리 액세스 요청이 선택될 필요가 있기 이전의 거의 마지막 모멘트에 이르기까지 메모리 컨트롤러에 디스패치돼야 할 다음 메모리 액세스 요청이 선택되지 않도록 하며, 그로써 시스템(100)의 효율을 향상시킨다.

[0025] 스케줄 관리기(128)는 조정기(130)가 다양한 기술들 중 어느 것을 이용하여 다음 메모리 액세스 요청을 디스패치하도록 조정기(130)에 시그널링을 행하기 위해 메모리 액세스 동작의 예측된 완료 여부를 판단할 수 있다. 일 실시예에서, 메모리 액세스 동작의 예측된 완료는 메모리(104)와 메모리 컨트롤러(102) 사이의 시그널링을 이용하여 나타내어질 수 있다. 예를 들어, 읽기(read) 액세스 동작 중 메모리(104)는 요청된 데이터가 메모리 컨트롤러(102)와 메모리(104)에 연결된 버스 라인들에서 이용가능하다는 것을 나타내는 신호를 메모리 컨트롤러(102)에 공급할 수 있는데, 이는 상기 데이터가 다음 클럭 주기 도중 래칭(latching)되게 하기 위함이다. 따라서, 이러한 경우에, 메모리(104)로부터의 상기 신호를 받는 대로 메모리 컨트롤러(102)는 디스패치_req 신호(134)를 어서트할 수 있고 그럼으로써 조정기(130)가 다음 메모리 액세스 요청의 디스패치를 개시하도록 조정기에 시그널링을 행한다. 또 다른 예에서, 메모리 액세스 동작은 쓰기 액세스 동작을 포함할 수 있고 상기 시그널링은 메모리 컨트롤러로부터 메모리에 공급되는 신호를 포함할 수 있는데, 이 신호는 쓰기 데이터가 메모리 컨트롤러(102)와 메모리(104)를 연결하는 버스에 위치함을 나타내고, 다음 주기 동안 메모리(104)가 데이터를 래치(latch)하는 것에 대한 응답이다. 이러한 경우에 스케줄 관리기(128)는, 메모리(104)에 공급될 수 있는 또는 메모리(104)로부터 메모리 컨트롤러(102)에 공급될 수 있는 쓰기 데이터의 어서트에 대한 응답으로, 상기 신호(134)를 어서트할 수 있다. 또 다른 실시예에서, 메모리 컨트롤러(102)에 의해 프로세싱되는 메모리 액세스 동작들의 일부 또는 전부는 미리 결정된 지속구간(duration)을 가질 수 있다. 예를 들어 설명하자면, 읽기 액세스 동작들의 특정 타입들은 여러 개의 클럭 주기들 집합을 필요로 함이 메모리 컨트롤러(102)에 의해 알려질 수 있다. 따라서, 그러한 읽기 액세스 동작들의 예측된 완료는 메모리 액세스 동작의 실제 완료 이전에 미리 결정된

지속구간에 의해 나타날 수 있다. 예를 들어, 메모리 액세스 동작이 8 개의 클럭 주기들을 사용한다고 알려진 경우에, 메모리 액세스 동작의 예측된 완료는 예를 들어 7 번째 클럭 주기에서 발생할 수 있고, 이 때에 스케줄 관리기(128)는 메모리 컨트롤러(102)에 다음 메모리 액세스 요청 디스패치를 위해 조정기(130)에 시그널링을 행하도록 디스패치_req 신호(134)를 어썬트할 수 있다.

[0026] 도 2를 참조해 보면, 본 개시사항의 적어도 하나의 일 실시예에 따라, 지연된 메모리 액세스 요청을 메모리 컨트롤러에 디스패치하는 예시적인 방법(200)이 도시되어 있다. 상기 방법(200)은 블럭(202)에서 개시되는바, 여기서 메모리 컨트롤러(102)는 BIU(106)으로부터 수신된 선택된 메모리 액세스 요청에 의해 나타내어지는, 메모리(104)에 관한 메모리 액세스 동작을 개시한다. 블럭(204)에서, 스케줄 관리기(128)는 메모리 액세스 동작을 모니터링하고, 스케줄 관리기(128)가 현재 메모리 액세스 동작의 예측된 완료를 확인한 때에 스케줄 관리기(128)는 디스패치_req 신호(134)를 어썬트하거나 그렇지 않으면 조정기(130)가 블럭(206)에서 BIU(106)로부터의 다음 메모리 액세스 요청의 디스패치를 개시하도록 조정기(130)에 신호를 전달한다. 블럭(208)에서 조정기(130)는 어썬트된 디스패치_req 신호(134)를 수신한다. 그에 호응하여, 블럭(210)에서 조정기(130)는 메모리 컨트롤러(102)에의 디스패치를 위하여 큐(132)로부터 메모리 액세스 요청을 선택한다. 상기 언급된 바와 같이, 조정기(130)는 BIU(106)에서 수신되는 메모리 접근 요청들의 우선순위와 그와 같은 다양한 특성들에 기초하여 다음 메모리 액세스 요청을 선택할 수 있다. 블럭(212)에서, 조정기(130)는 선택된 메모리 액세스 요청을 메모리 컨트롤러(102)에 디스패치한다. 블럭(214)에서, 디스패치된 메모리 액세스 요청은 메모리 컨트롤러(102)에서 수신되고 메모리 액세스 요청 큐(120)에서 버퍼링될 수 있거나 선택기(126)를 사용하는 즉시의 프로세싱을 위해 스케줄 관리기(128)에 직접 공급될 수도 있는데, 이것은 스케줄 관리기(128) 또는 메모리 컨트롤러(102)의 다른 구성요소에 의해 제어될 수 있다.

[0027] 도 3과 도 4를 참조해 보면, 본 개시사항의 적어도 하나의 일 실시예에 따라, 도 1의 시스템에서의 도 2의 방법에 대한 예시적인 구현이 도시되어 있다. 도 3은 본 개시사항의 적어도 하나의 일 실시예에 따라, BIU(106)에서의 메모리 액세스 요청들의 수신을 나타내는 시퀀스(302)와, 메모리 컨트롤러에 메모리 액세스 요청들을 디스패치하는 종래의 기술을 나타내는 시퀀스(304), 그리고 메모리 컨트롤러(102)에 메모리 요청들을 디스패치하는 기술을 나타내는 시퀀스(306)를 갖는 타이밍 다이어그램(300)을 도시하고 있다. 도 3은 또한 메모리 컨트롤러(102)에의 디스패치를 위해 메모리 액세스 요청들을 선택하는 데에 있어서 조정기(130)에 의해 사용되고 조정기(130)에 의해 유지될 수 있는 예시적인 특성 차트(310)를 도시하고 있다. 도 4는 종래의 디스패치 시퀀스(304)와 지연된 디스패치 시퀀스(306) 모두에 대한 타이밍 다이어그램(300)의 다양한 점들에서 메모리 컨트롤러(102)의 큐(120)의 예시적인 상태들을 도시하고 있다. 설명을 용이하게 하기 위해, 메모리 액세스 요청들은 메모리 컨트롤러(102)에 의한 추가적 프로세싱을 위해 큐(120)의 2개의 큐 엔트리들(122, 124)로부터 얻어진다는 점이 가정된다.

[0028] 시퀀스(302)는 BIU(106)에서의 메모리 액세스 요청들(R_1 - R_4)의 수신을 나타낸다. 도시된 대로, 메모리 액세스 요청들(R_1 , R_2)은 포인트(A)에 앞서 수신되고 메모리 액세스 요청들(R_3 , R_4)은 포인트(A)와 포인트(B) 사이에서 수신된다. 도시되지는 아니하였으나, 요청들(R_0 , R_6)은 BIU(106)에서 메모리 액세스 요청(R_1)의 수신에 앞서서 수신된다. 차트(310)에 도시하는 대로, 메모리 액세스 요청들(R_6 , R_0)은 1의 우선순위를 가지고, 메모리 액세스 요청(R_1)이 2의 우선순위를 가지며, 메모리 액세스 요청(R_2)이 3의 우선순위를, 메모리 액세스 요청(R_3)이 0의 우선순위를, 메모리 액세스 요청(R_4)이 1의 우선순위를 가지고, 여기서 0의 우선순위는 가장 높은 우선순위를 나타내고 3의 우선순위는 가장 낮은 우선순위를 나타낸다. 타이밍 다이어그램(300)에서, 포인트들(A, B, C)은 메모리 컨트롤러(102)가 메모리 컨트롤러(102)에서 메모리 액세스 요청에 의해 나타나는 메모리 액세스 동작의 프로세싱을 개시하는 때의 포인트들을 나타내고, 그러므로 메모리 컨트롤러(102)가 BIU(106)로부터 또 다른 메모리 액세스 요청을 수신할 능력을 처음으로 갖게 되는 때를 나타낸다. 포인트들(A', B', C')은 개별적으로 스케줄 관리기(128)가 프로세싱되고 있는 메모리 액세스 동작의 예측된 완료를 식별하는 때의 포인트들을 나타낸다.

[0029] 상기 언급된 바와 같이, 종래의 메모리 디스패치 기술은 메모리 컨트롤러가 또 다른 메모리 액세스 요청을 수신할 능력을 갖자마자 메모리 액세스 요청이 메모리 컨트롤러에 디스패치되는 것으로 구현된다. 이 종래의 기술을 나타내는 시퀀스(304)에 도시된 바와 같이, 포인트(A)일 때 BIU에서 버퍼링된, 가장 높은 우선순위를 갖는 메모리 액세스 요청(이 예에서는 메모리 액세스 요청 R_1)은, 포인트(A) 직후에, 프로세싱을 위해 메모리 컨트롤러에 디스패치된다. 마찬가지로, 종래의 기술 사용시 포인트(B)에서, 메모리 컨트롤러에서 공간이 다시 이용가능하게 되는 경우에는, 종래의 디스패치 시퀀스(304)는 가장 높은 우선순위의 메모리 액세스 요청이 디스패치를 위해

선택되는 포인트 B에서 버퍼링됨을 예시하고 있다(이 경우 메모리 액세스 요청 R_3). 그러나, 상기에 언급된 대로 시퀀스(304)에 의해 나타내어진 이 종래의 기술은 종종 메모리 액세스 요청들의 덜 효율적인 스케줄링을 제공한다. 예를 들어, 메모리 액세스 요청 R_3 가 CPU(108)(도 1)로부터의 보다 높은 우선순위의 메모리 액세스 요청이라고 가정하면, 메모리 액세스 요청들(R_1 , R_2)은 예컨대 DMA 마스터(112)에 연결된 장치들로부터의 낮은 우선순위의 메모리 액세스 요청들이다. 종래의 디스패치 시퀀스(304)에서, 메모리 액세스 요청(R_1)은 메모리 액세스 요청(R_3)의 디스패치와 프로세싱에 앞서 메모리 컨트롤러(102)에서 디스패치되고 프로세싱되는데(메모리 액세스 요청(R_3)가 메모리 액세스 요청(R_1)보다 높은 우선순위를 가지고 있는 경우에도 그러하다), 이는 메모리 컨트롤러(102)가 버스 인터페이스 유닛(106)으로부터 메모리 액세스 요청을 수신할 능력을 처음으로 갖게 되는 때의 포인트(A) 이후의 시각에 메모리 액세스 요청(R_3)이 수신되고 그로 인해 메모리 액세스 요청(R_3)이 상기 수신으로 인해 디스패치를 위한 선택이 가능하지 않게 됨에 기인한다.

[0030] 반면에, 도 2의 방법(200)의 예시적인 구현방안을 나타내는 디스패치 시퀀스(306)는, 조정 지연과 메모리 컨트롤러(102)에서 (메모리 컨트롤러(102)에 의해 수행되고 있는 메모리 액세스 동작의 예측된 완료에 의해 나타날 수 있는) 거의 가장 마지막 가능 모멘트에 이르기까지의 메모리 액세스 요청의 선택에 의해, 메모리 액세스 요청들의 더욱 효율적인 조정을 달성할 수 있음을 예시한다. 예를 들어 설명해보자면, 스케줄 관리기(128)는 현재 프로세싱 중인 메모리 액세스 동작이 포인트(A')에서 완료에 가까워졌는지를 판단한다. 이에 호응하여, 스케줄 관리기(128)는 조정기(130)에 의해 수신되는 디스패치_req 신호(134)를 어썬트한다. 이 신호(134)의 어썬트에 대한 응답으로, 조정기(130)는 큐(130)에서 그때에 이용가능한 메모리 액세스 요청들(다시 말해, 메모리 액세스 요청들 R_6 , R_0)을 선택한다. 이 경우, 메모리 액세스 요청(R_0)이 메모리 액세스 요청(R_6)보다 높은 우선순위를 가지기 때문에, 메모리 액세스 요청(R_0)은 메모리 컨트롤러(102)에의 디스패치를 위하여 조정기(130)에 의해 선택된다. 시퀀스(306)는 메모리 컨트롤러(102)가 현재의 메모리 액세스 동작을 완료하는 때인 포인트(A)까지 계속되나, 종래의 기술들과는 달리, 스케줄 관리기(128)는 메모리 컨트롤러(102)가 또 다른 메모리 액세스 요청을 수신할 능력을 갖게 되는 경우에도 다음 메모리 액세스 요청을 위해 시그널링을 행하지 않는다. 대신, 스케줄 관리기(128)는 다음 메모리 액세스 동작의 예측된 완료시까지 기다리고, 예측된 완료시인 포인트(B')에서 신호(134)를 어썬트한다. 포인트(B')에서의 신호(134) 어썬트에 대한 응답으로, 조정기(130)는 메모리 컨트롤러(102)에의 디스패치를 위해 다음 메모리 액세스 요청을 선택한다. 시퀀스(302)에서 예시된 바와 같이, 메모리 컨트롤러(102)가 처음으로 다음 메모리 액세스 요청을 수신할 능력을 가지게 되는 때인 포인트(A)부터 현재 메모리 액세스 동작이 메모리 컨트롤러(102)에서 완료에 가까워진 때인 포인트(B') 사이의 지연은, BIU(106)가 높은 우선순위의 메모리 액세스 요청(R_3)을 수신할 수 있게 한다. 포인트(B')에서의 신호(134) 어썬트에 대한 응답에 따라, 조정기(130)는 그것의 큐(132)에서 이용가능한 메모리 액세스 요청(R_3)을 큐(132)에서 가장 높은 우선순위의 요청으로서 가지고, 조정기(130)는 포인트(B') 이후에 메모리 컨트롤러(102)에의 디스패치를 위해 메모리 액세스 요청(R_3)을 선택한다. 비슷한 방식으로, 메모리 컨트롤러(102)에 의해 수행 중인 메모리 액세스 동작이 포인트(C')에서 완료에 가까워졌을 때, 조정기(130)가 메모리 컨트롤러(102)에의 디스패치를 위해 메모리 액세스 요청(R_1)을 선택함에 대한 응답으로, 스케줄 관리기(128)는 신호(134)를 어썬트한다.

[0031] 도 4에 예시된 바와 같이, 메모리 액세스 요청의 지연된 선택 및 디스패치는 일반적으로 메모리 컨트롤러(102)에서 메모리 액세스 요청들의 보다 효율적인 스케줄링을 가능하게 한다. 예를 들어 설명해보면, 종래의 디스패치 시퀀스(304)의 포인트(B)에서 메모리 컨트롤러는 프로세싱을 위해 그것의 큐로부터 메모리 액세스 요청을 선택한다. 메모리 액세스 큐 상태(402)에 의해 예시된 바와 같이, 종래의 디스패치 기술을 사용하는 메모리 컨트롤러는 포인트(B)에서 큐 내부에 메모리 액세스 요청들(R_1 과 R_0)을 가질 수 있는데, 메모리 액세스 요청(R_0)이 일반적으로 그것의 높은 우선순위로 인해 프로세싱을 위하여 요청들(R_1 과 R_0)로부터 선택될 수 있다. 또한, 종래의 기술을 사용하는 메모리 컨트롤러의 메모리 액세스 요청 큐는 메모리 액세스 요청들(R_1 , R_3)을 가질 수 있는데, 메모리 액세스 요청(R_3)이 일반적으로 그것의 높은 우선순위로 인해 프로세싱을 위하여 요청들(R_1 과 R_3)로부터 선택될 수 있다. 반대로, 메모리 액세스 큐 상태(404)에 예시된 대로, 본 명세서에 개시된 지연된 메모리 액세스 요청 디스패치 기술을 갖는 메모리 컨트롤러(102)는, 포인트(B)에서, 그것의 큐(120) 내부에 메모리 액세스 요청(R_3)을 가질 수 있는데, 이 메모리 액세스 요청(R_3)은 프로세싱을 위해 선택될 수 있으며, 포인트(C)에서 큐(120)는 메모리 액세스 요청들(R_1 과 R_0)을 포함할 수 있고, 메모리 액세스 요청(R_0)이 프로세싱을 위해 요청들

(R_1 과 R_0)로부터 선택될 수 있다. 따라서, 도 3과 도 4가 예시하는 바와 같이, 본 명세서에 개시된 지연된 메모리 액세스 요청 디스패치 기술은 가장 높은 우선순위의 메모리 액세스 요청(R_3)이 종래의 디스패치 기술을 사용하는 프로세싱 시스템에 있을 때보다 더 빨리 프로세싱될 수 있게 한다. 그래서, 본 명세서에 개시된 상기 지연된 메모리 액세스 요청 디스패치 기술은 종래의 기술들(이에 의해 메모리 액세스 요청들은 메모리 컨트롤러가 그것들을 수신할 능력을 갖자마자 즉시 디스패치됨)과 비교하여 메모리 액세스 요청들의 시퀀싱이 보다 효율적으로 이루어지도록 한다.

[0032] 본 명세서에 개시된 실시예들을 면밀히 검토할 경우 본 개시사항의 다른 실시예들, 용도들, 이점들은 당업자에게 자명해질 것이다. 본 명세서와 도면들은 단지 예시적인 것으로 고려되어야 하고, 본 개시사항의 범위는 단지 특허청구범위와 그것의 균등물에 의해서만 제한될 수 있다.

도면의 간단한 설명

[0013] 본 개시사항의 목적과 이점들은 하기의 상세한 설명과 첨부된 도면들의 조합으로부터 당업자에게는 명확해질 것이고, 도면에서 비슷한 참조 문자들은 비슷한 구성요소들을 지시하는데에 이용되며, 도면에서:

[0014] 도 1은 본 개시사항의 적어도 하나의 실시예에 따라, 지연된 메모리 액세스 요청 디스패치 기술을 사용하는 예시적인 프로세싱 시스템을 도시하는 블록 다이어그램이다.

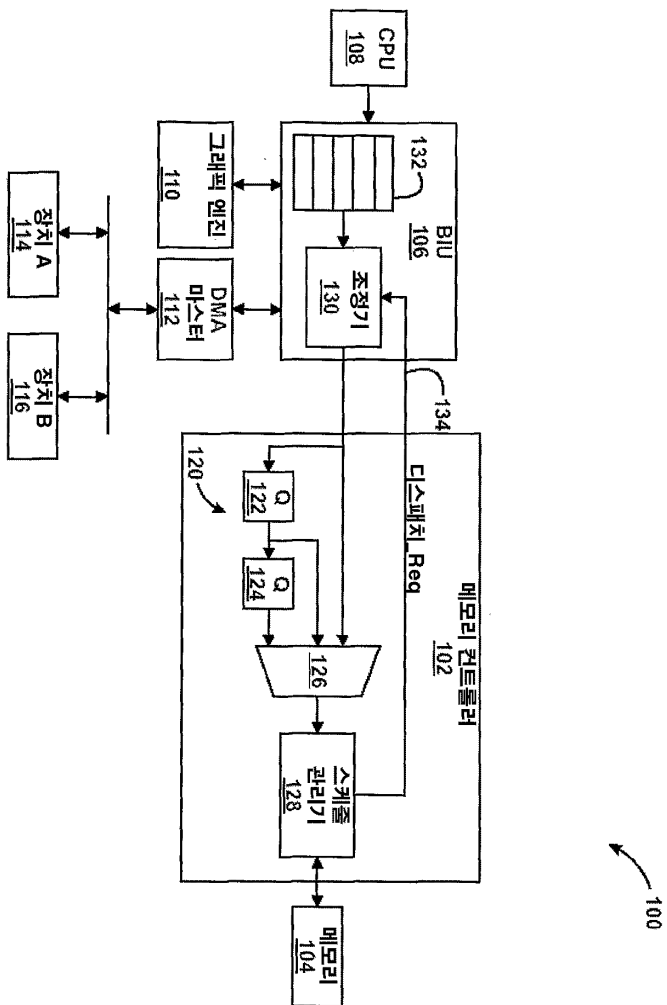
[0015] 도 2는 본 개시사항의 적어도 하나의 실시예에 따라, 지연된 메모리 액세스 요청 디스패치에 대한 예시적인 방법을 도시하는 순서도이다.

[0016] 도 3은 본 개시사항의 적어도 하나의 실시예에 따라, 도 2의 방법의 예시적인 구현방안을 도시하는 타이밍 다이어그램과 우선순위 차트이다.

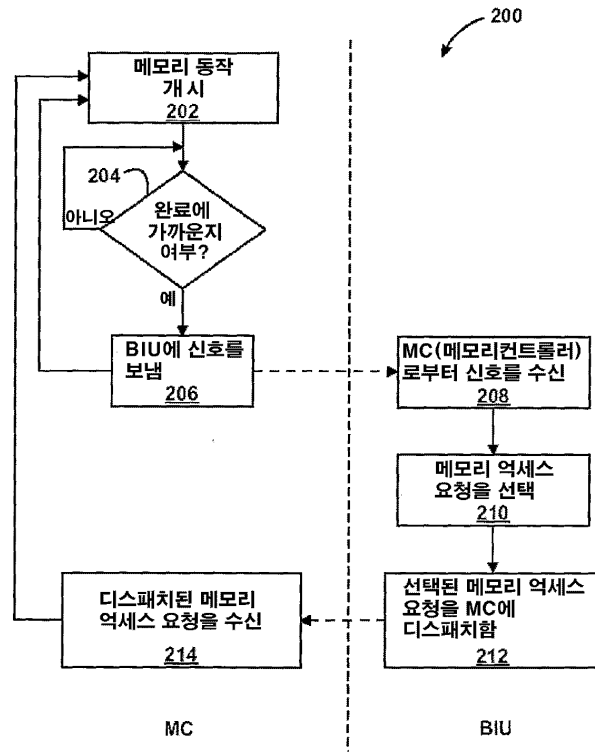
[0017] 도 4는 본 개시사항의 적어도 하나의 실시예에 따라, 도 3의 타이밍 다이어그램에 대응되는 메모리 컨트롤러 큐(queue)의 예시적인 상태들을 도시하는 다이어그램이다.

도면

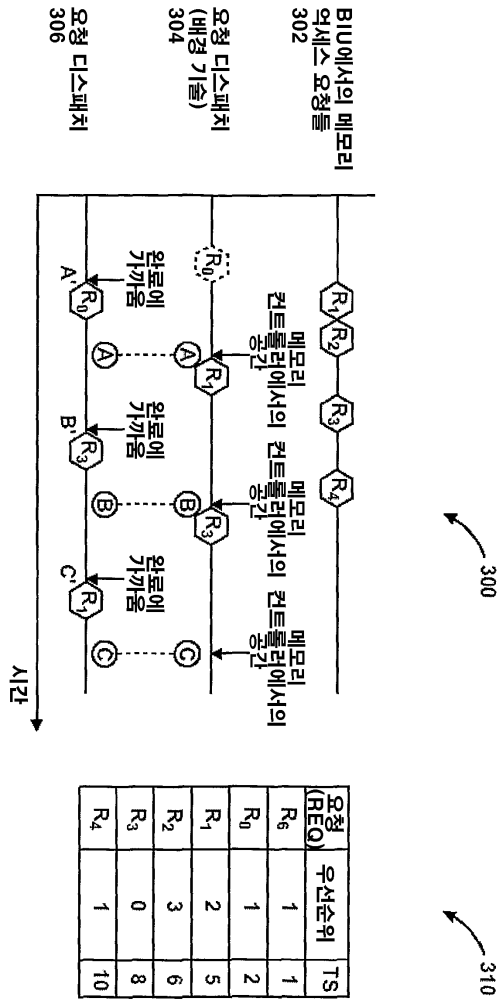
도면1



도면2



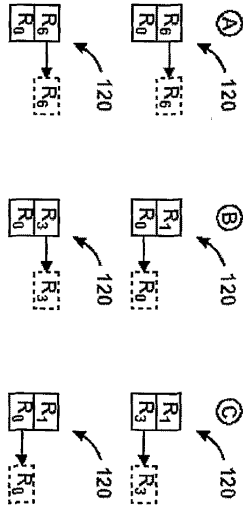
도면3



도면4

메모리 액세스
동작 디스패치
(배경 기술)
402

메모리 액세스
동작 디스패치
404



400