(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년11월15일
(11) 등록번호 10-2602844
(24) 등록일자 2023년11월13일

(51) 국제특허분류(Int. Cl.)
G06T 1/60 (2006.01) *G06F 13/16* (2006.01)
G06F 9/38 (2006.01) *G06T 1/20* (2018.01)
(52) CPC특허분류
G06T 1/60 (2013.01)
G06F 13/1668 (2013.01)
(21) 출원번호 10-2023-0047873
(22) 출원일자 2023년04월12일
심사청구일자 2023년04월12일
(56) 선행기술조사문헌
KR1020230022439 A
(뒷면에 계속)
전체 청구항 수 : 총 7 항

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
구건재
경기도 남양주시 불암로 337, 5103동 2004호
정종현
서울특별시 성북구 안암로9가길 9-16, 301호
(74) 대리인
김준석, 박민욱

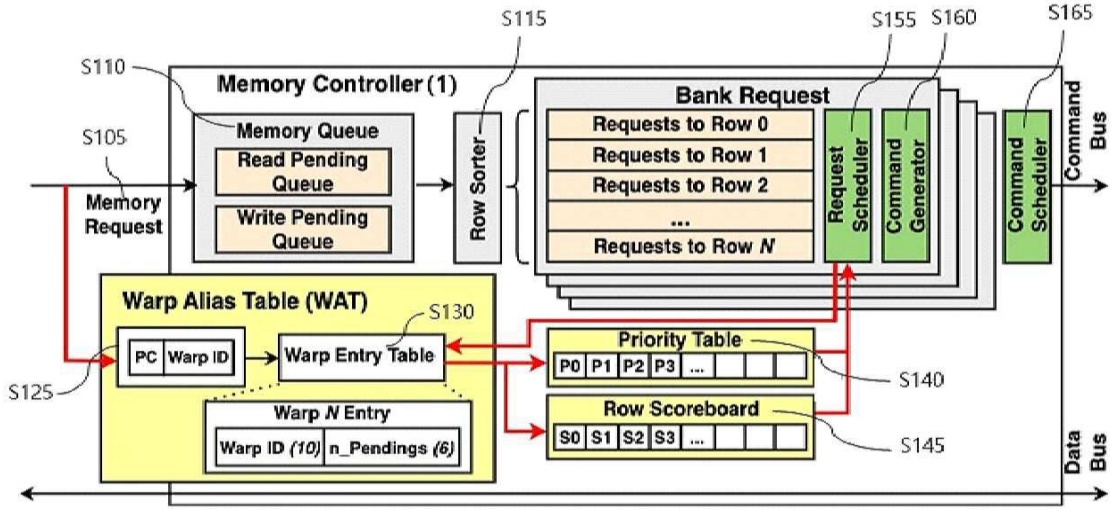
심사관 : 이후락

(54) 발명의 명칭 **그래픽스 프로세서의 워프 구조를 고려한 메모리 컨트롤러 및 그것을 이용한 메모리 컨트롤 방법**

(57) 요 약

본 발명은 그래픽스 프로세서의 워프 구조를 고려한 메모리 컨트롤러 및 그것을 이용한 메모리 컨트롤 방법에 관한 것으로, 메모리 요청이 메모리 대기열에 전송된 동시에, WAT(Warp Alias Table)에서 메모리 요청이 포함된 PC의 해쉬값을 기반으로 글로벌 워프 ID를 생성하는 단계; WET(Warp Entry Table)에서 글로벌 워프 ID(Global Warp ID) 및 글로벌 워프 ID에 남은 메모리 요청 수를 포함하는 워프의 실행 상태 정보를 기록하는 단계; 워프의 실행 상태 정보를 기반으로 로우 버퍼(Row Buffer)에 접근하는 메모리 요청들 간의 우선순위를 기록하는 우선순위 테이블(Priority Table, PT) 및 DRAM 뱅크 내의 개별 로우(Row)의 중요도를 기록하는 로우 스코어보드(Row Scoreboard, RS)를 관리하는 단계; 및 우선순위 테이블 및 로우 스코어보드를 기반으로 메모리 요청을 예약하는 단계; 를 포함한다.

대 표 도 - 도3



(52) CPC특허분류
G06F 9/38 (2023.08)
G06T 1/20 (2013.01)
G06T 2200/28 (2013.01)

(56) 선행기술조사문헌
KR1020210090442 A
KR1020200098744 A
KR1020200114702 A
KR1020160111122 A
KR101940523 B1
KR102408350 B1

이 발명을 지원한 국가연구개발사업
과제고유번호 1711186523
과제번호 2021R1C1C1012172
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)
연구과제명 대규모 그래프 구조의 신경망 처리를 위한 효율적인 프로세서 및 메모리 구조에 대한 연구
기 여 율 80/100
과제수행기관명 고려대학교
연구기간 2023.03.01 ~ 2024.02.29
이 발명을 지원한 국가연구개발사업
과제고유번호 1711193663
과제번호 2020-0-01819-004
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 ICT명품인재양성(고려대학교)
기 여 율 10/100
과제수행기관명 고려대학교산학협력단
연구기간 2023.01.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711193387
과제번호 2022-0-01198-002
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 융합보안대학원(고려대학교)
기 여 율 10/100
과제수행기관명 고려대학교산학협력단
연구기간 2023.01.01 ~ 2023.12.31

명 세 서

청구범위

청구항 1

삭제

청구항 2

삭제

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

메모리 요청이 포함된 PC의 해쉬값을 기반으로 글로벌 워프 ID(Global Warp ID)를 생성하는 WAT(Warp Alias Table);

상기 글로벌 워프 ID 및 상기 글로벌 워프 ID에 남은 메모리 요청 수를 포함하는 워프의 실행 상태 정보를 기록하고, 상기 워프의 실행 상태 정보를 기반으로 로우 버퍼(Row Buffer)에 접근하는 메모리 요청들 간의 우선순위를 기록하는 우선순위 테이블(Priority Table, PT) 및 DRAM 뱅크 내의 개별 로우(Row)의 중요도를 기록하는 로우 스코어보드(Row Scoreboard, RS)를 관리하는 WET(Warp Entry Table);

상기 우선순위 테이블 및 상기 로우 스코어보드를 기반으로 상기 메모리 요청을 예약하는 뱅크 요청부;를 포함하고,

상기 WET(Warp Entry Table)는,

상기 WET에서 상기 글로벌 워프 ID에 남은 메모리 요청 수가 1일 때,

상기 남은 하나의 메모리 요청이 로우 버퍼 힛(Row Buffer Hit)인 경우에 상기 우선순위 테이블에 메모리 요청의 우선순위를 기록하고, 상기 남은 하나의 메모리 요청이 로우 버퍼 힛이 아닌 경우에 상기 로우 스코어보드에 상기 남은 하나의 메모리 요청에 대하여 로우 버퍼를 교체할 시 우선순위를 얻을 수 있도록 로우의 중요도를 기록하는 메모리 컨트롤러.

청구항 8

제7항에 있어서,

상기 메모리 요청이 접근하는 로우 ID 및 뱅크 ID를 디코딩하는 Row sorter; 및

상기 예약한 메모리 요청을 기반으로 가장 우선순위가 높은 메모리 요청이 대기중인 뱅크에 명령어를 발행하는 명령어 스케줄러;를 더 포함하는 메모리 컨트롤러.

청구항 9

제8항에 있어서,

상기 명령어 스케줄러는,

DRAM에서 상기 메모리 요청이 처리가 완료되면, 처리가 완료된 메모리 요청은 제거되고, 상기 WET에 기록된 글로벌 워프 ID의 실행 상태 정보가 업데이트되는 메모리 컨트롤러.

청구항 10

제7항에 있어서,

상기 뱅크 요청부는,

상기 메모리 요청을 예약하는 요청 스케줄러; 및

상기 메모리 요청 전송 및 처리에 필요한 DRAM 명령어를 생성하는 명령어 생성부;를 더 포함하는 메모리 컨트롤러.

청구항 11

제7항에 있어서,

상기 WET(Warp Entry Table)는,

상기 글로벌 워프 ID를 이용하여 처음으로 메모리 요청을 기록하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 초기화되고,

상기 글로벌 워프 ID가 이미 존재하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 증가되는 메모리 컨트롤러.

청구항 12

제7항에 있어서,

메모리 컨트롤러에 미리 연결된 L2 캐시로부터 수신한 상기 메모리 요청을 메모리 대기열에 저장하는 메모리 대기부;를 더 포함하는 메모리 컨트롤러

청구항 13

제12항에 있어서,

상기 뱅크 요청부는,

상기 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛인 경우에 우선순위 테이블을 기반으로 가장 우선순위가 높은 메모리 요청을 예약하고,

상기 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛인 아닌 경우에 로우 스코어보드를 기반으로 가장 중요도가 높은 로우에 접근하는 메모리 요청을 예약하는 메모리 컨트롤러.

발명의 설명

기 술 분 야

[0001] 본 발명은 그래픽스 프로세서의 워프 구조를 고려한 메모리 컨트롤러 및 그것을 이용한 메모리 컨트롤 방법에 관한 것이다.

배 경 기 술

[0002] GPU는 GPU에서 실행되는 커널(Kernel)에서 수천 개의 쓰레드(Thread)를 발생시켜 병렬적으로 처리하는 실행 방식을 가지고 있으며 이 때 생성된 많은 수의 쓰레드를 몇십개씩 모아서(보통 32개) 워프(Warp (NVIDIA에서 쓰는 용어)), 혹은 웨이브프론트(Wavefront (AMD에서 쓰는 용어))라고 하는 집합 단위로 묶어서 한꺼번에 실행한다.

[0003] 워프 내에 있는 각각의 쓰레드가 메모리 요청을 발생시키면, 메모리 요청이 완료될 때까지 해당 워프는 실행을

기다려야 하기 때문에 가장 느리게 응답받는 메모리 요청이 해당 워프의 실행 시간을 결정하게 된다.

[0004] 메모리 요청이 내부 캐시 메모리에서 서비스를 받지 못하면, 해당 메모리 요청은 GPU 내에 있는 메모리 컨트롤러를 통해 외부 메모리인 DRAM으로 전송되어야 하는데, 이때, DRAM에서 데이터를 가지고 오는 메모리 요청은 긴 소요시간을 요구하기 때문에 외부 메모리에 접근하는 메모리 요청 때문에 워프는 오랜 시간을 기다려야 하고, 이로 인하여 전체적인 GPU의 성능이 저하되게 되는 문제점이 있다.

[0005] GPU에서 외부 DRAM을 접근하기 위해서는 메모리 컨트롤러를 거쳐야 하고, 메모리 컨트롤러는 프로세서에서 외부 메모리로 메모리 요청을 전송하기 이에 대한 데이터 및 응답을 받아 처리하는 역할을 하며, 정해진 메모리 스케줄링 정책에 따라 메모리 요청에 대한 우선순위를 정하게 된다.

[0006] 종래의 메모리 컨트롤러는 FR-FCFS (First-Ready, First Come First Serve) 방식을 이용하고 있으며, FR-FCFS 스케줄링 방식은 DRAM 뱅크(Bank) 내의 Row buffer에 접근하는 메모리 요청에 대해서 우선순위를 높여서 처리를 하기 때문에 연속된 메모리 요청 전송 및 처리에 필요한 DRAM 명령어(Command) 및 지연시간을 절감할 수 있으나, 메모리 컨트롤러에 도착한 메모리 요청의 대기시간을 고려하지 않기 때문에 종종 먼저 도착한 요청이 우선순위가 밀리면서 나중에 처리하는 메모리 요청의 불균형 (Memory latency divergence)를 유발할 수 있는 문제점이 있다.

[0007] 따라서, GPU의 워프에서 발생한 다수의 메모리 요청에 대해서 우선순위가 밀리게 된 메모리 요청이 발생할 경우, 해당 워프는 낮은 우선순위의 메모리 요청을 오랜 시간 동안 기다리게 되어 워프의 실행시간이 길어지고, GPU의 성능이 저하될 수 있는 문제점이 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 대한민국등록특허공보 제10-2408350호 ("에너지 효율을 개선할 수 있는 그래픽 처리 장치를 위한 메모리 컨트롤러 및 이의 메모리 제어 방법", 연세대학교 산학협력단, 공개일자: 2022.01.06)

발명의 내용

해결하려는 과제

[0009] 본 발명은 우선순위 테이블과 로우 스코어보드를 이용하여 메모리 요청을 예약하는 그래픽스 프로세서의 워프 구조를 고려한 메모리 컨트롤러 및 그것을 이용한 메모리 컨트롤 방법을 제공하는 것을 해결하고자 하는 과제로 한다.

과제의 해결 수단

[0010] 상술한 과제를 해결하기 위하여 그래픽스 프로세서의 워프 구조를 고려한 메모리 컨트롤러 및 그것을 이용한 메모리 컨트롤 방법이 제공된다.

[0011] 메모리 컨트롤 방법은 메모리 요청이 메모리 대기열에 전송된 동시에, WAT(Warp Alias Table)에서 상기 메모리 요청이 포함된 PC의 해쉬값을 기반으로 글로벌 워프 ID(Global Warp ID)를 생성하는 단계; WET(Warp Entry Table)에서 상기 글로벌 워프 ID 및 상기 글로벌 워프 ID에 남은 메모리 요청 수를 포함하는 워프의 실행 상태 정보를 기록하는 단계; 상기 워프의 실행 상태 정보를 기반으로 로우 버퍼(Row Buffer)에 접근하는 메모리 요청들 간의 우선순위를 기록하는 우선순위 테이블(Priority Table, PT) 및 DRAM 뱅크 내의 개별 로우(Row)의 중요도를 기록하는 로우 스코어보드(Row Scoreboard, RS)를 관리하는 단계; 및 상기 우선순위 테이블 및 상기 로우 스코어보드를 기반으로 상기 메모리 요청을 예약하는 단계; 를 포함하고, 상기 우선순위 테이블 및 로우 스코어보드를 관리하는 단계는, 상기 WET에서 상기 글로벌 워프 ID에 남은 메모리 요청 수가 1일 때, 상기 남은 하나의 메모리 요청이 로우 버퍼 힛(Row Buffer Hit)인 경우에 상기 우선순위 테이블에 우선순위가 기록되고, 상기 남은 하나의 메모리 요청이 로우 버퍼 힛이 아닌 경우에 상기 로우 스코어보드에 상기 남은 하나의 메모리 요청에 대하여 로우 버퍼를 교체할 시 우선순위를 얻을 수 있도록 중요도가 기록된다.

[0012] 일 실시예에서, 상기 메모리 요청이 접근하는 로우 ID 및 뱅크 ID를 디코딩하는 단계; 상기 메모리 요청 전송 및 처리에 필요한 DRAM 명령어를 생성하는 단계; 및 상기 예약한 메모리 요청을 기반으로 가장 우선순위가 높은

메모리 요청이 대기중인 뱅크에 명령어를 발행하는 단계; 를 더 포함할 수 있다.

[0013] 일 실시예에서, 상기 워프의 실행 상태 정보를 기록하는 단계는, 상기 글로벌 워프 ID를 이용하여 처음으로 메모리 요청을 상기 WET에 기록하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 초기화되고, 상기 WET에 글로벌 워프 ID가 이미 존재하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 증가될 수 있다.

[0014] 일 실시예에서, 상기 메모리 대기열은, 상기 메모리 컨트롤러에 미리 연결된 L2 캐시로부터 수신한 상기 메모리 요청을 저장할 수 있다.

[0015] 일 실시예에서, 상기 메모리 요청을 예약하는 단계는, 상기 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛인 경우에 우선순위 테이블을 기반으로 가장 우선순위가 높은 메모리 요청을 예약하고, 상기 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛인 아닌 경우에 로우 스코어보드를 기반으로 가장 중요도가 높은 로우에 접근하는 메모리 요청을 예약할 수 있다.

[0016] 일 실시예에서, DRAM에서 상기 메모리 요청이 처리가 완료되면, 처리가 완료된 메모리 요청은 제거되고, 상기 WET에 기록된 글로벌 워프 ID(Global Warp ID)의 실행 상태 정보가 업데이트되는 단계;를 더 포함할 수 있다.

[0017] 메모리 컨트롤러는 메모리 요청이 포함된 PC의 해쉬값을 기반으로 글로벌 워프 ID를 생성하는 WAT(Warp Alias Table); 상기 글로벌 워프 ID 및 상기 글로벌 워프 ID에 남은 메모리 요청 수를 포함하는 워프의 실행 상태 정보를 기록하고, 상기 워프의 실행 상태 정보를 기반으로 로우 버퍼(Row Buffer)에 접근하는 메모리 요청들 간의 우선순위를 기록하는 우선순위 테이블(Priority Table, PT) 및 DRAM 뱅크 내의 개별 로우(Row)의 중요도를 기록하는 로우 스코어보드(Row Scoreboard, RS)를 관리하는 WET(Warp Entry Table); 상기 우선순위 테이블 및 상기 로우 스코어보드를 기반으로 상기 메모리 요청을 예약하는 뱅크 요청부;를 포함하고, 상기 WET(Warp Entry Table)는, 상기 WET에서 상기 글로벌 워프 ID에 남은 메모리 요청 수가 1일 때, 상기 남은 하나의 메모리 요청이 로우 버퍼 힛(Row Buffer Hit)인 경우에 상기 우선순위 테이블에 우선순위가 기록되고, 상기 남은 하나의 메모리 요청이 로우 버퍼 힛이 아닌 경우에 상기 로우 스코어보드에 상기 남은 하나의 메모리 요청에 대하여 로우 버퍼를 교체할 시 우선순위를 얻을 수 있도록 중요도가 기록된다.

[0018] 일 실시예에서, 상기 메모리 요청이 접근하는 로우 ID 및 뱅크 ID를 디코딩하는 Row sorter; 및 상기 예약한 메모리 요청을 기반으로 가장 우선순위가 높은 메모리 요청이 대기중인 뱅크에 명령어를 발행하는 명령어 스케줄러;를 더 포함할 수 있다.

[0019] 일 실시예에서, 상기 명령어 스케줄러는, DRAM에서 상기 메모리 요청이 처리가 완료되면, 처리가 완료된 메모리 요청은 제거되고, 상기 WET에 기록된 글로벌 워프 ID의 실행 상태 정보가 업데이트될 수 있다.

[0020] 일 실시예에서, 상기 뱅크 요청부는, 상기 메모리 요청을 예약하는 요청 스케줄러; 및 상기 메모리 요청 전송 및 처리에 필요한 DRAM 명령어를 생성하는 명령어 생성부;를 더 포함할 수 있다.

[0021] 일 실시예에서, 상기 WET(Warp Entry Table)는, 상기 글로벌 워프 ID를 이용하여 처음으로 메모리 요청을 기록하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 초기화되고, 상기 글로벌 워프 ID가 이미 존재하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 증가될 수 있다.

[0022] 일 실시예에서, 메모리 컨트롤러에 미리 연결된 L2 캐시로부터 수신한 상기 메모리 요청을 메모리 대기열에 저장하는 메모리 대기부;를 더 포함할 수 있다.

[0023] 일 실시예에서, 상기 뱅크 요청부는, 상기 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛인 경우에 우선순위 테이블을 기반으로 가장 우선순위가 높은 메모리 요청을 예약하고, 상기 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛인 아닌 경우에 로우 스코어보드를 기반으로 가장 중요도가 높은 로우에 접근하는 메모리 요청을 예약할 수 있다.

발명의 효과

[0024] 상술한 그래픽스 프로세서의 워프 구조를 고려한 메모리 컨트롤러 및 그것을 이용한 메모리 컨트롤 방법에 의하면, 워프의 실행 상태를 인식하여 메모리 컨트롤러가 동작함으로써, 워프에서 메모리 지연 시간 차이를 완화할 수 있으며, 그로 인해 워프의 수행 시간을 단축시킬 수 있다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 일 실시예에 따른 메모리 컨트롤러의 구성을 설명하기 위한 블록도이다.

도 2는 본 발명의 일 실시예에 따른 메모리 컨트롤 방법을 설명하기 위한 순서도이다.

도 3은 본 발명의 일 실시예에 따른 메모리 컨트롤 방법을 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 메모리 컨트롤러의 스케줄링 알고리즘을 설명하기 위한 도면이다.

도 5는 메모리 요청을 스케줄링한 결과를 설명하기 위한 도면이다.

도 6은 정규화된 메모리 컨트롤러의 사이클당 명령(IPC)을 나타내는 도면이다.

도 7은 정규화된 메모리 컨트롤에서 워프의 평균 실행 시간을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0026] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다.

[0027] 본 명세서에서 사용되는 용어에 대해 간략히 설명하고, 본 발명에 대해 구체적으로 설명하기로 한다.

[0028] 본 발명에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 판례, 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 발명에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 발명의 전반에 걸친 내용을 토대로 정의되어야 한다.

[0029] 명세서 전체에서 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있음을 의미한다. 또한, 명세서에서 사용되는 "부", "모듈", "유닛" 등의 용어는 적어도 하나의 기능 또는 동작을 처리하는 단위를 의미하며, 소프트웨어, FPGA 또는 ASIC과 같은 하드웨어 구성요소, 또는 소프트웨어와 하드웨어의 결합으로 구현될 수 있다. 그렇지만 "부", "모듈", "유닛" 등의 용어가 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. "부", "모듈", "유닛" 등은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 ""부", "모듈", "유닛" 등의 용어는 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다.

[0030] 아래에서는 첨부한 도면을 참고하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략한다.

[0031] "제1" "제2" 등과 같이 서수를 포함하는 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 구성 요소들은 용어들에 의해 한정되지는 않는다. 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는" 이라는 용어는 복수의 관련된 항목들의 조합 또는 복수의 관련된 항목들 중의 어느 하나의 항목을 포함한다.

[0033] 이하, 도면을 참조하여 메모리 컨트롤러(1)를 설명하도록 한다.

[0034] 도 1은 본 발명의 일 실시예에 따른 메모리 컨트롤러의 구성을 설명하기 위한 블록도이다.

[0035] 도 1에서 도시한 바와 같이, 본 발명의 일 실시예에 따른 메모리 컨트롤러(Memory Controller)(1)는 메모리 대기부(100), WAT(Warp Alias Table)(200), WET(Warp Entry Table)(300), Row sorter(400), 뱅크 요청부(Bank Request)(500), 명령어 스케줄러(Command Scheduler)(600)를 포함한다.

[0036] 메모리 대기부(100)은 메모리 컨트롤러(1)에 미리 연결된 L2 캐시로부터 수신한 메모리 요청을 메모리 대기열에

저장한다.

[0037] 메모리 대기부(100)는 메모리 대기열에 존재하는 메모리 요청에 대한 뱅크 ID(Bank ID) 및 로우 ID(Row ID)를 식별하고, 뱅크 ID 별로 정렬할 수 있다.

[0038] 메모리 요청이 메모리 대기부(100)에 전송된 동시에, WAT(200)는 메모리 요청이 포함된 PC의 해쉬값을 기반으로 글로벌 워프 ID(Global Warp ID)를 생성한다.

[0039] WAT(200)는 PC의 해쉬 값, SM(Streaming multiprocessor)의 워프 ID, SM ID를 결합하여 글로벌 워프 ID를 생성할 수 있다.

[0040] WET(300)은 글로벌 워프 ID 및 글로벌 워프 ID에 남은 메모리 요청 수(n_pendings)를 포함하는 워프의 실행 상태 정보를 기록한다.

[0041] 이때, 메모리 컨트롤러(1)는 오프칩 메모리 액세스를 생성하는 워프를 인식하고, WET(300)은 메모리 컨트롤러(1)가 관리하는 워프의 실행 상태 정보를 저장할 수 있다.

[0042] 남은 메모리 요청 수는 WAT(200)에서 생성된 글로벌 워프 ID와 인덱싱될 수 있고, WET(300)은 글로벌 워프 ID를 이용하여 워프의 실행 상태 정보에 액세스할 수 있다.

[0043] 새로운 글로벌 워프 ID를 이용하여 처음으로 메모리 요청을 WET(300)에 기록하는 경우, 상기 글로벌 워프 ID에 남은 메모리 요청 수는 1로 초기화될 수 있다.

[0044] WET(300)에 글로벌 워프 ID가 이미 존재하는 경우, 글로벌 워프 ID에 남은 메모리 요청 수는 증가될 수 있다.

[0045] 그리고, WET(300)는 우선순위 테이블(Priority Table, PT)(310), 로우 스코어보드(Row Scoreboard, RS)(320)를 포함할 수 있다.

[0046] 우선순위 테이블(310)은 로우 버퍼(Row Buffer)에 접근하는 메모리 요청들 간의 우선순위를 기록한 테이블이고, 로우 스코어보드(320)는 DRAM 뱅크 내의 개별 로우의 중요도를 기록하는 테이블일 수 있다.

[0047] WET(300)는 워프의 실행 상태 정보를 기반으로 로우 버퍼에 접근하는 메모리 요청들 간의 우선순위를 기록하는 우선순위 테이블(310) 및 DRAM 뱅크 내의 개별 로우의 중요도를 기록하는 로우 스코어보드(320)를 관리할 수 있다.

[0048] 더욱 자세하게는, 글로벌 워프 ID에 남은 메모리 요청 수가 1일 때, 해당 글로벌 워프 ID는 실행 완료를 위해 단 하나의 메모리 요청 처리만 남은 상태를 의미한다.

[0049] 따라서, 글로벌 워프 ID에 남은 하나의 메모리 요청이 로우 버퍼 힛(Row Buffer Hit)인 경우, 우선순위 테이블(310)은 남은 하나의 메모리 요청에 대하여 우선순위 "1"로 기록될 수 있고, 남은 하나의 메모리 요청이 로우 버퍼 힛이 아닌 경우, 로우 스코어보드(320)는 남은 하나의 메모리 요청에 대하여 우선순위가 기록될 수 있다.

[0050] 예를 들면, 글로벌 워프 ID에 남은 하나의 메모리 요청이 [Bank 0]에 속한 로우 버퍼에 대한 메모리 요청이라고 가정한다.

[0051] 남은 하나의 메모리 요청이 [Bank 0]에 속한 로우 버퍼에 접근하는 경우, 남은 하나의 메모리 요청은 우선순위 테이블(310)에 우선순위가 기록될 수 있고, 남은 하나의 메모리 요청이 [Bank 0]에 속한 로우 버퍼에 접근하지 않는 경우, 남은 하나의 메모리 요청은 로우 스코어보드(320)에 우선순위가 기록될 수 있다.

[0052] 로우 스코어보드(320)는 남은 하나의 메모리 요청에 대하여 로우 버퍼를 교체할 시 우선순위를 얻을 수 있도록 중요도를 기록할 수 있다.

[0053] 한편, Row Sorter(400)는 메모리 요청에 대응하는 로우 ID 및 뱅크 ID를 디코딩하고, 로우 ID 및 뱅크 ID로 디코딩된 메모리 요청을 뱅크 요청부(500)로 전달할 수 있다.

[0054] 뱅크 요청부(500)는 복수의 뱅크 각각에 대하여 로우 별로 메모리 요청을 예약할 수 있다.

[0055] 뱅크 요청부(500)는 요청 스케줄러(Request Scheduler)(510) 및 명령어 생성부(Command Generator)(520)를 포함한다.

[0056] 요청 스케줄러(510)는 우선순위 테이블(310) 및 로우 스코어보드(320)를 기반으로 메모리 요청을 예약한다.

[0057] 요청 스케줄러(510)는 뱅크별로 메모리 요청을 예약할 수 있다.

[0058] 이때, 요청 스케줄러(510)는 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛(Row Buffer hit)인 경우에 우선순위 테이블(310)을 기반으로 가장 우선순위가 높은 로우에 접근하는 메모리 요청을 예약할 수 있다.

[0059] 또한, 요청 스케줄러(510)는 메모리 대기열에 남은 메모리 요청이 로우 버퍼 힛이 아닌 경우에 로우 스코어보드(320)를 기반으로 가장 중요도가 높은 로우에 접근하는 메모리 요청을 예약할 수 있다.

[0060] 즉, 요청 스케줄러(510)는 로우 버퍼 미스(Row Buffer Miss)인 경우에 로우 스코어보드(320)를 기반으로 메모리 요청을 예약할 수 있다.

[0061] 명령어 생성부(520)는 메모리 요청 전송 및 처리에 필요한 DRAM 명령어를 생성한다.

[0062] 명령어 스케줄러(600)는 각 뱅크 별로 요청 스케줄러(510)가 예약한 메모리 요청들을 비교하여 가장 우선순위가 높은 메모리 요청이 대기중인 뱅크에 명령어를 발행한다.

[0063] 명령어 스케줄러(600)는 뱅크에 예약된 메모리 요청을 처리하기 위한 DRAM 명령어를 발행할 수 있다.

[0064] 그리고, DRAM에서 메모리 요청이 처리가 완료되면, 메모리 대기부(100)에서 처리가 완료된 메모리 요청은 제거되고, WET(300)에 기록된 해당 글로벌 워프 ID의 실행 상태 정보는 업데이트될 수 있다.

[0066] 이하, 도면을 참고하여 본 발명의 일 실시예에 따른 메모리 컨트롤 방법을 설명하도록 한다.

[0067] 도 2는 본 발명의 일 실시예에 따른 메모리 컨트롤 방법을 설명하기 위한 순서도이고, 도 3은 본 발명의 일 실시예에 따른 메모리 컨트롤 방법을 설명하기 위한 도면이다.

[0068] 먼저, 메모리 컨트롤러(1)에 미리 연결된 L2 캐시는 메모리 요청(Memory Request)을 메모리 컨트롤러(1)에 전송한다(S105).

[0069] 메모리 대기부(100)는 L2 캐시로부터 전송된 메모리 요청을 메모리 대기열(Memory Queue)에 저장한다(S110).

[0070] S110 단계는 도 1의 메모리 대기부(100)에서 이미 설명하였는 바, 자세한 설명은 생략한다.

[0071] Row sorter(400)는 메모리 요청이 접근하는 로우 ID 및 뱅크 ID를 디코딩한다(S115).

[0072] Row sorter(400)는 로우 ID 및 뱅크 ID로 디코딩된 메모리 요청을 뱅크별로 요청 스케줄러(510)에 전달한다.

[0073] 요청 스케줄러(510)는 디코딩된 메모리 요청이 로우 버퍼 힛 요청인지 아닌지 판단한다(S120).

[0074] 이때, 로우 버퍼 힛 요청은 로우 버퍼에 접근하는 메모리 요청이다.

[0075] 요청 스케줄러(510)는 디코딩된 메모리 요청이 로우 버퍼 힛 요청인지에 대응하여 메모리 요청을 예약한다.

[0076] 더욱 자세하게는, 디코딩된 메모리 요청이 로우 버퍼 힛 요청이면(S120의 YES), 요청 스케줄러(510)는 메모리 요청을 예약하고(S155), 디코딩된 메모리 요청이 로우 버퍼 힛 요청이 아니면(S120의 NO), 요청 스케줄러(510)는 메모리 요청에 대응되는 로우를 선택하여(S150) 메모리 요청을 예약할 수 있다(S155).

[0077] 이때, 요청 스케줄러(510)는 가장 오래된 메모리 요청을 예약할 수 있다.

[0078] 한편, 메모리 요청이 메모리 대기부(100)에 저장되는 동시에, WAT(200)는 메모리 요청이 포함된 PC의 해쉬값을 기반으로 글로벌 워프 ID를 생성한다(S125).

[0079] WET(300)는 글로벌 워프 ID 및 글로벌 워프 ID에 남은 메모리 요청 수를 포함하는 워프의 실행 상태 정보를 기록한다(S130).

[0080] 도 3에 기재된 S130에서 도시한 바와 같이, WET(300)은 "Warp ID"와 "n_Pendings"를 기록할 수 있다.

[0081] WET(300)은 글로벌 워프 ID에 남은 메모리 요청 수가 1개일 때, 해당 메모리 요청이 로우 버퍼 힛인지 아닌지 판별한다(S135).

[0082] WET(300)은 해당 메모리 요청이 로우 버퍼 힛인 경우(S135의 YES)에 우선순위 테이블(310)에 전달할 수 있다.

[0083] S130 및 S135 단계는 도 1의 WET(300)에서 이미 설명하였는 바, 자세한 설명은 생략한다.

[0084] 우선순위 테이블(310)은 해당 메모리 요청의 우선순위를 "1"로 기록할 수 있다(S140).

[0085] 예를 들어, 도 3과 같이, 우선순위 테이블(310)은 우선순위가 높은 순서대로 메모리 요청을 [P0. P1, P2, P3,

...]과 같이 정렬할 수 있다.

[0086] S140 단계는 도 1의 우선순위 테이블(310)에서 이미 설명하였는 바, 자세한 설명은 생략한다.

[0087] 요청 스케줄러(510)는 우선순위 테이블(310)을 기반으로 우선순위가 가장 높은 메모리 요청을 예약할 수 있다(S155).

[0088] 또한, WET(300)은 글로벌 워프 ID에 남은 메모리 요청 수가 1개일 때, 해당 메모리 요청이 로우 버퍼 힛이 아닌 경우(S135의 NO)에 로우 스코어보드(320)에 전달할 수 있다.

[0089] 로우 스코어보드(320)는 로우 버퍼를 교체할 시 우선순위를 얻을 수 있도록 해당 메모리 요청에 대한 중요도를 기록할 수 있다(S145).

[0090] 예를 들어, 도 3과 같이, 로우 스코어보드(320)는 중요도가 높은 순서대로 뱅크의 로우 ID를 [S0, S1, S2, S3, ...]과 같이 정렬할 수 있다.

[0091] S145 단계는 도 1의 로우 스코어보드(320)에서 이미 설명하였는 바, 자세한 설명은 생략한다.

[0092] 요청 스케줄러(510)는 중요도가 가장 높은 로우를 선택하여(S150) 메모리 요청을 예약할 수 있다(S155).

[0093] S155 단계는 도 1의 요청 스케줄러(510)에서 이미 설명하였는 바, 자세한 설명은 생략한다.

[0094] 이와 같이 예약된 메모리 요청에 대하여, 명령어 생성부(520)는 메모리 요청 전송 및 처리에 필요한 DRAM 명령어를 생성할 수 있다(S160).

[0095] 명령어 스케줄러(600)는 각 뱅크별로 요청 스케줄러(510)가 예약한 메모리 요청들을 기반으로 가장 우선순위가 높은 메모리 요청이 대기중인 뱅크에 명령어를 발행할 수 있다(S165).

[0096] 그리고, DRAM에서 메모리 요청이 처리되면, 메모리 대기부(100)에서 해당 메모리 요청이 제거되고, WET(300)에 기록된 해당 글로벌 워프 ID에 남은 메모리 요청 수가 업데이트될 수 있다.

[0097] 본 발명의 일 실시예에 따른 메모리 컨트롤 방법에서, S110 단계 내지 S120 단계와 S125 단계 내지 S140 단계는 병렬적으로 동시에 실행될 수 있다.

[0099] 이하, 도면을 참고하여 본 발명의 일 실시예에 따른 메모리 컨트롤러의 스케줄링 알고리즘을 설명하도록 한다.

[0100] 도 4는 본 발명의 일 실시예에 따른 메모리 컨트롤러의 스케줄링 알고리즘을 설명하기 위한 도면이다.

[0101] 도 4에서 도시한 바와 같이, 요청 우선순위 과정(procedure REQUEST PROPRITY)은 워프 ID로 식별된 보류 중인 메모리 요청의 수(warp_entry[wid].n_pending--)를 모니터링한다.

[0102] 보류 중인 메모리 대기열에 존재하는 메모리 요청이 워프의 마지막 보류 중인 요청(즉, 워프의 보류 중인 메모리 요청 수가 1개)인 경우(if warp_entry[wid].n_pending == 1 then), 메모리 컨트롤러(1)는 워프의 마지막 보류 중인 요청의 우선순위를 증가시킨다(last request ← priority H).

[0103] 메모리 대기열에 보류 중인 메모리 요청이 여러 개 있는 워프의 메모리 요청이 Row Buffer에 접근하면(warp_entry[wid].n_pending ≥ 2 then), 메모리 컨트롤러(1)는 메모리 요청의 우선순위를 두번째로 높일 수 있다(pending requests of warp A ← priority M).

[0104] 따라서, 메모리 컨트롤러(1)는 메모리 요청이 예약되었을 때 우선순위가 더 높은 워프의 긴급 요청을 예약할 수 있다.

[0105] 또한, 로우 우선순위 과정(procedure ROW PRIORITY)은 메모리 대기열에 보류 중인 메모리 요청을 Row ID별로 그룹화하여 메모리 요청의 중요성을 기반으로 보류 중인 로우에 대한 점수를 기록한다.

[0106] 로우에 워프의 가장 느린 메모리 요청이 포함된 경우, 메모리 컨트롤러(1)는 보류 중인 로우의 점수를 높일 수 있다.

[0107] 메모리 컨트롤러(1)는 로우가 누락되어 로우 교체가 필요한 경우에 가장 높은 점수를 나타내는 보류중인 로우를 새로운 로우로 선택한다.

[0108] 로우 버퍼가 교체될 시, 메모리 컨트롤러(1)는 메모리 요청의 우선순위 절차에 따라 선택한 보류 중인 로우의 메모리 요청을 예약한다.

[0109] 그러면, 선택한 로우의 점수는 0이 될 수 있다.

[0110] 또한, 뱅크 우선순위 과정(procedure BANK PRIORITY)은 우선순위가 더 높은 메모리 요청을 예약할 뱅크에 더 높은 우선순위를 설정할 수 있다.

[0111] 요청 스케줄러(510)는 대상 뱅크의 로우 버퍼에 도달하는 가장 우선순위가 높은 요청을 예약할 수 있다.

[0112] 명령어 스케줄러(600)는 DRAM 명령을 수신할 수 있는 사용 가능한 뱅크(if bank is idle then) 중에서 우선 순위가 높은 뱅크 풀(Bank Pool)에서 라운드 로빈 방식으로 선택한다. 다음에 우선순위가 낮은 뱅크 풀을 스캔하여 사용 가능한 뱅크를 선택할 수 있다.

[0113] 이때, 뱅크의 우선 순위는 뱅크에 예약된 메모리 요청의 우선 순위이다(bank_priority ← scheduled_request.priority).

[0114] 이하, 도면을 참고하여 메모리 요청을 스케줄링한 결과를 설명하도록 한다.

[0115] 도 5는 메모리 요청을 스케줄링한 결과를 설명하기 위한 도면이다.

[0116] 도 5의 (a)는 종래 기술의 메모리 컨트롤러를 이용하여 메모리 요청을 스케줄링한 결과이고, 도 5의 (b)는 본 발명의 일 실시예에 따른 메모리 컨트롤러를 이용하여 메모리 요청을 스케줄링한 결과이다.

[0117] 도 5의 (a)에서 도시한 바와 같이, [Warp 0]와 [Warp 4]는 실행 완료까지 단 하나의 메모리 요청만 남겨두고 있다.

[0118] [Warp 4]에 남은 메모리 요청은 Row 2에 접근하므로, 로우 교체(Row conflict)가 2번 진행된 후에 처리될 수 있다.

[0119] 그리고, [Warp 0]에 남은 메모리 요청은 Row 2에서 58번째[Req 58]로 처리될 수 있다.

[0120] 따라서, 종래의 메모리 컨트롤러는 Latency divergence 현상인 메모리 대기시간 차이가 발생할 수 있다.

[0121] 도 5의 (b)에서 도시한 바와 같이, 본 발명의 일 실시예에 따른 메모리 컨트롤러는 [Bank 0]에서 [Warp 4]의 마지막 메모리 요청 처리를 위해 Row 2에 접근하는 메모리 요청을 먼저 처리하도록 스케줄링할 수 있고, [Bank 1]에서 [Warp 0]에 남은 메모리 요청인 Row2에서 [Req 58]의 순서를 앞당겨서 먼저 처리하도록 스케줄링될 수 있다.

[0123] 이하, 도면을 참고하여 본 발명의 일 실시예에 따른 메모리 컨트롤러의 성능을 설명하도록 한다.

[0124] 도 6은 정규화된 메모리 컨트롤러의 사이클당 명령(IPC)을 나타내는 도면이고, 도 7은 정규화된 메모리 컨트롤에서 워프의 평균 실행 시간을 나타내는 도면이다.

[0125] 도 6에서 도시한 바와 같이, 메모리 컨트롤러(1)는 MOA 및 HOA 애플리케이션에 대해 기준선 대비 평균(GMEAN) 8.86%의 성능을 향상시킬 수 있고, KMN에 대해 최대 45.8% 증가됨을 알 수 있다.

[0126] 도 7에서 도시한 바와 같이, 메모리 컨트롤러(1)는 워프의 평균 실행 시간을 평균 19.3%, KMN에서 최대 39.7% 단축시킬 수 있고, GMV에서 워프의 실행 시간을 6.45% 단축시킬 수 있음을 알 수 있다.

[0128] 이상에서 설명한 바와 같이, 본 발명에 따르면, 워프의 실행 상태를 인식하여 메모리 컨트롤러가 동작함으로써, 워프에서 메모리 지연 시간 차이를 완화할 수 있으며, 그로 인해 워프의 수행 시간을 단축시킬 수 있다.

[0130] 본원 발명의 실시예들과 관련된 기술 분야에서 통상의 지식을 가진 자는 상기 기재의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로, 개시된 방법들은 한정적인 관점이 아닌 설명적 관점에서 고려되어야 한다. 본 발명의 범위는 발명의 상세한 설명이 아닌 특허청구 범위에 나타나며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0132] 1: 메모리 컨트롤러

100: 메모리 대기부

200: WAT

300: WET

310: 우선순위 테이블

320: 로우 스코어보드

400: Row sorter

500: 뱅크 요청부

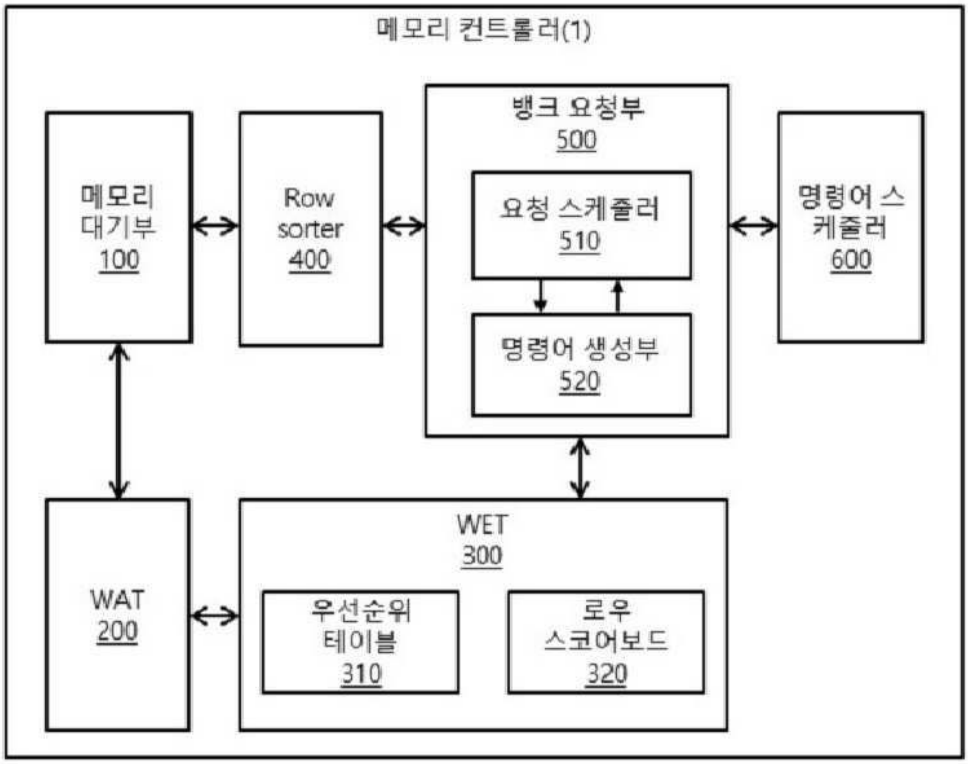
510: 요청 스케줄러

520: 명령어 생성부

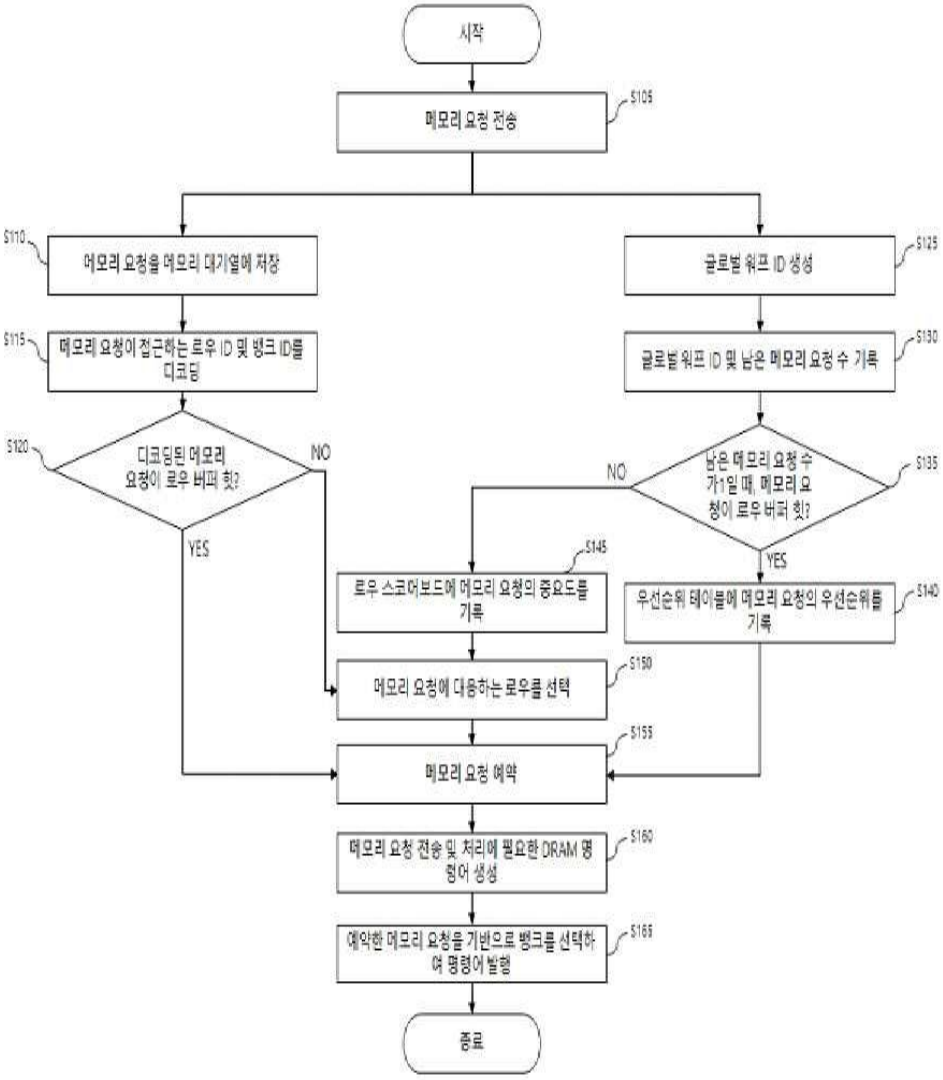
600: 명령어 스케줄러

도면

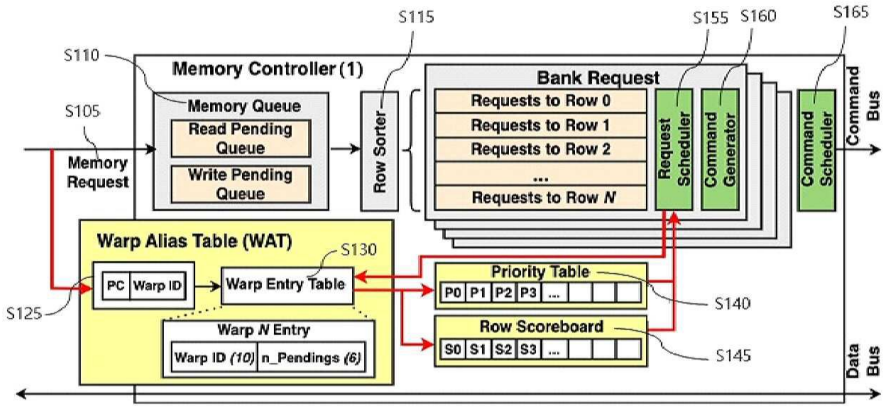
도면1



도면2


시작
메모리 요청 전송
S105
S110
메모리 요청을 메모리 대기열에 저장
S115
메모리 요청이 접근하는 로우 ID 및 뱅크 ID를 디코딩
S120
디코딩된 메모리 요청이 로우 버퍼 힛?
NO
YES
S125
글로벌 워프 ID 생성
S130
글로벌 워프 ID 및 남은 메모리 요청 수 기록
S135
남은 메모리 요청 수가 1일 때, 메모리 요청이 로우 버퍼 힛?
NO
YES
S140
우선순위 테이블에 메모리 요청의 우선순위를 기록
S145
로우 스코어보드에 메모리 요청의 중요도를 기록
S150
메모리 요청에 대응하는 로우를 선택
S155
메모리 요청 예약
S160
메모리 요청 전송 및 처리에 필요한 DRAM 명령어 생성
S165
예약한 메모리 요청을 기반으로 뱅크를 선택하여 명령어 발행
종료

도면3

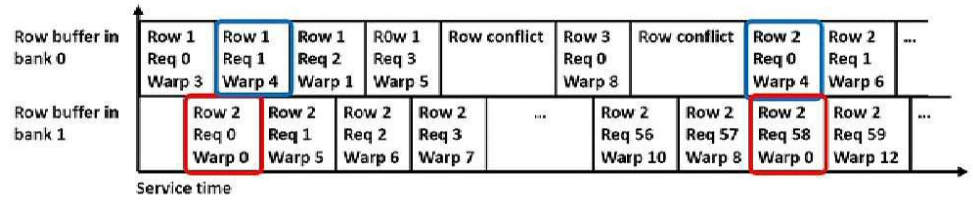

S115
S155
S160
S165
S110
Memory Controller (1)
Memory Queue
Read Pending Queue
Write Pending Queue
S105
Memory Request
Row Sorter
Bank Request
Requests to Row 0
Requests to Row 1
Requests to Row 2
...
Requests to Row N
Request Scheduler
Command Generator
Command Scheduler
Command Bus
Warp Alias Table (WAT)
S130
PC
Warp ID
Warp Entry Table
Warp N Entry
Warp ID (10)
n_Pendings (6)
S125
Priority Table
P0 P1 P2 P3 ...
S140
Row Scoreboard
S0 S1 S2 S3 ...
S145
Data Bus

도면4

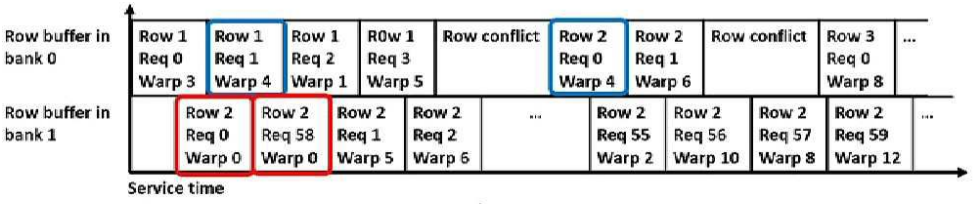
Algorithm 1 Warped-MC scheduling

```
1: warp ID of warp A ← wid
2: bank ID of request ← bid
3: row ID of request ← rid
4: procedure REQUEST PRIORITY
5:     if schedule request then
6:         warp_entry[wid].n_pending--
7:         if warp_entry[wid].n_pending == 1 then
8:             last request ← priority H
9:         if warp_entry[wid].n_pending ≥ 2 then
10:            pending requests of warp A ← priority M
11: procedure ROW PRIORITY
12:    if schedule request then
13:        if warp_entry[wid].n_pending == 1 then
14:            row_scoreboard[bid][rid]++
15: procedure BANK PRIORITY
16:    if bank is idle then
17:        bank_priority ← scheduled_request.priority
```

도면5

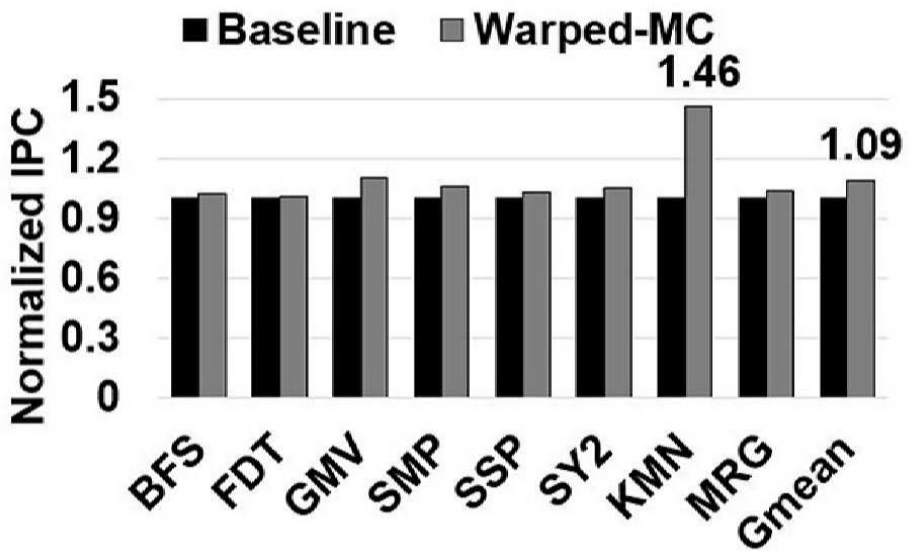


(a)

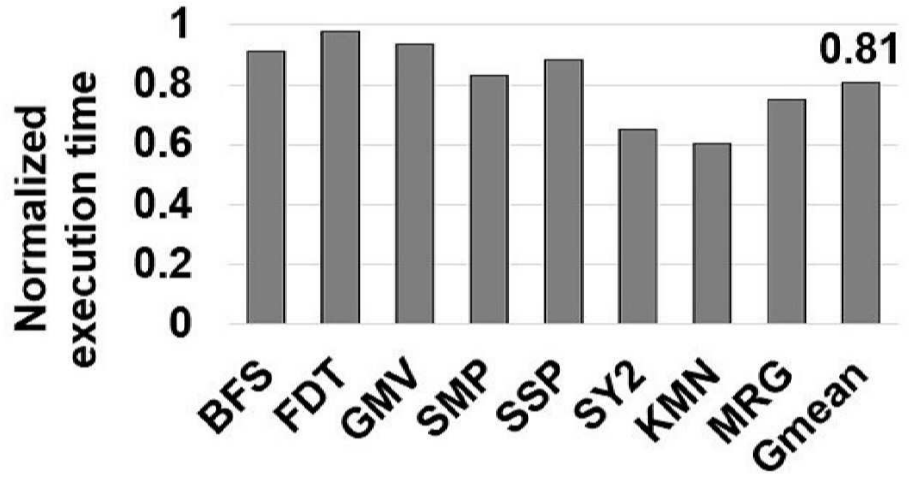


(b)

도면6


Baseline
Warped-MC
Normalized IPC
1.5
1.2
0.9
0.6
0.3
0
1.46
1.09
BFS
FDT
GMV
SMP
SSP
SY2
KMN
MRG
Gmean

도면7


Normalized execution time
1
0.8
0.6
0.4
0.2
0
0.81
BFS
FDT
GMV
SMP
SSP
SY2
KMN
MRG
Gmean