



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0151455
(43) 공개일자 2024년10월18일

(51) 국제특허분류(Int. Cl.)
G06F 12/0862 (2016.01) G06F 3/06 (2006.01)
(52) CPC특허분류
G06F 12/0862 (2013.01)
G06F 3/0604 (2013.01)
(21) 출원번호 10-2023-0047479
(22) 출원일자 2023년04월11일
심사청구일자 2023년04월11일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
윤명국
서울특별시 종로구 통일로 246-20, 102동 603호 (무악동, 무악현대아파트)
박은성
경기도 김포시 풍무로68번길 39, 101동 1703호 (풍무동, 한화유로메트로아파트)
(74) 대리인
특허법인 무한

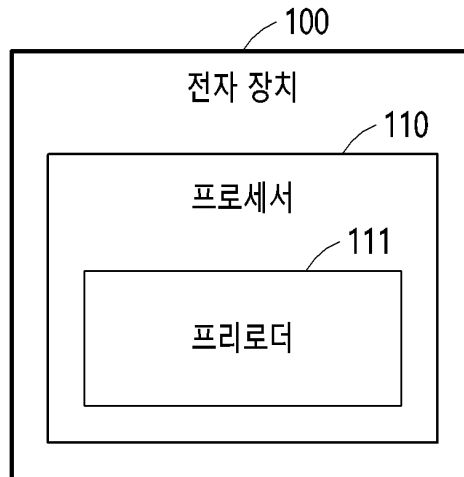
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 메모리 주소를 예측하여 데이터를 프리페칭하는 전자 장치 및 그 동작 방법

(57) 요약

메모리 주소를 예측하여 데이터를 프리페칭하는 전자 장치 및 그 동작 방법이 도시된다. 본 개시의 일실시예에 따른 전자 장치의 동작 방법은 이전 요구 메모리 요청(prior demand memory request)에 포함된 이전 요구 메모리 주소들(prior demand memory address)을 샘플링하는 단계, 상기 샘플링된 상기 이전 요구 메모리 주소들 간의 보폭(stride)에 기반하여 프리페치 메모리 요청(prefetch memory request)을 생성하는 단계, 상기 프리페치 메모리 요청에 따른 프리페치 메모리 응답(prefetch response)을 프리로더에 전달하는 단계, 및 상기 프리페치 메모리 응답에 포함되는 프리페치 데이터 및 상기 프리페치 데이터에 대응하는 프리페치 메모리 주소(prefetch memory address)를 프리로더 캐시에 저장하는 단계를 포함할 수 있다.

대표도 - 도1



(52) CPC특허분류

G06F 2212/1016 (2013.01)

G06F 2212/602 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711179344
과제번호	00155966
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	인공지능융합혁신인재양성(R&D)
연구과제명	인공지능융합혁신인재양성(이화여자대학교)
기 여 율	30/100
과제수행기관명	이화여자대학교산학협력단
연구기간	2022.07.01 ~ 2025.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711193622
과제번호	2021-0-02068-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성
연구과제명	인공지능 혁신 허브 연구 개발
기 여 율	30/100
과제수행기관명	이화여자대학교산학협력단
연구기간	2023.01.01 ~ 2023.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711189100
과제번호	2021R1G1A1092196
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)
연구과제명	대규모 병렬 프로세서를 위한 미세 전력 관리 기법
기 여 율	40/100
과제수행기관명	이화여자대학교
연구기간	2023.03.01 ~ 2024.02.29

명세서

청구범위

청구항 1

전자 장치의 동작 방법에 있어서,

이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링하는 단계;

상기 샘플링된 상기 이전 요구 메모리 주소들 간의 보폭에 기반하여 프리패치 메모리 요청을 생성하는 단계;

상기 프리패치 메모리 요청에 따른 프리패치 메모리 응답을 프리로더에 전달하는 단계; 및

상기 프리패치 메모리 응답에 포함되는 프리패치 데이터 및 상기 프리패치 데이터에 대응하는 프리패치 메모리 주소를 프리로더 캐시에 저장하는 단계

를 포함하는 전자 장치의 동작 방법.

청구항 2

제1항에 있어서,

상기 프리패치 메모리 요청을 생성하는 단계는,

상기 보폭에 기반하여 상기 프리패치 메모리 주소를 예측하고, 상기 프리패치 메모리 주소를 포함하는 상기 프리패치 메모리 요청을 생성하는, 동작 방법.

청구항 3

제2항에 있어서,

상기 보폭은,

인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭 중 하나 또는 둘 이상의 조합을 포함하고,

상기 프리패치 메모리 요청을 생성하는 단계는,

상기 인트라 워프 패턴 보폭의 우선 순위를 상기 인터 워프 패턴 보폭 보다 높게 결정하여 상기 프리패치 메모리 요청을 결정하는, 동작 방법.

청구항 4

제2항에 있어서,

상기 프리패치 메모리 요청을 생성하는 단계는,

상기 보폭이 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭을 모두 포함하고 상기 인트라 워프 패턴 보폭 및 상기 인터 워프 패턴 보폭 각각에 기반한 상기 프리패치 메모리 주소가 상이한 경우, 상기 인트라 워프 패턴 보폭에 기반한 상기 프리패치 메모리 요청을 생성하는, 동작 방법.

청구항 5

제1항에 있어서,

LDST 유닛으로부터 다음 연산을 위한 요구 메모리 요청을 상기 프리로더로 전달하는 단계; 및

상기 요구 메모리 요청에 포함된 요구 메모리 주소 및 상기 프리패치 메모리 주소에 기반하여 상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는 단계를 더 포함하는, 동작 방법.

청구항 6

제5항에 있어서,
상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는 단계는,
상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 경우, 상기 프리패치 메모리 주소에 대응하는 상기 프리패치 데이터를 상기 프리로더 캐시로부터 상기 LDST 유닛으로 전송하는, 동작 방법.

청구항 7

제5항에 있어서,
상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는 단계는,
상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 상이한 경우, 상기 요구 메모리 요청을 하위 캐시로 전달하고, 상기 요구 메모리 요청을 수신한 상기 하위 캐시로부터 상기 데이터를 상기 LDST 유닛으로 전송하는, 동작 방법.

청구항 8

제1항에 있어서,
상기 프리로더 캐시는,
복수의 엔트리들을 포함하고,
상기 복수의 엔트리들 각각은 엔트리의 상태를 나타내는 상태 비트를 포함하는, 동작 방법.

청구항 9

제8항에 있어서,
상기 프리패치 메모리 요청이 생성되었는지 여부, 요구 메모리 요청이 생성되기 전에 상기 프리패치 메모리 응답이 프리로더에 도달했는지 여부, 및 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 여부에 따라 상기 상태 비트들을 업데이트하는 단계를 더 포함하는, 동작 방법.

청구항 10

제1항에 있어서,
상기 프리로더 캐시에 저장하는 단계는,
복수의 서브 프리패치 메모리 요청들을 생성하고, 상기 복수의 서브 프리패치 메모리 요청들에 기반하여 상기 프리패치 메모리 주소를 저장하는, 동작 방법.

청구항 11

전자 장치의 동작 방법에 있어서,

샘플링된 이전 요구 메모리 주소들의 보폭에 기반하여 프리패치 메모리 요청을 생성하는 단계;

상기 프리패치 메모리 요청에 대한 응답인 프리패치 메모리 응답에 포함되는 프리패치 데이터 및 상기 프리패치 데이터에 대응하는 프리패치 메모리 주소를 프리로더 캐시에 저장하는 단계;

LDST 유닛으로부터 생성된 다음 연산을 위한 요구 메모리 요청에 포함된 요구 메모리 주소 및 상기 프리로더 캐시에 저장된 상기 프리패치 메모리 주소가 일치하는지 여부를 판단하는 단계; 및

상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 경우, 상기 프리패치 데이터를 상기 LDST 유닛으로 전송하고, 상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하지 않는 경우, 상기 요구 메모리 요청을 상기 프리로더 캐시의 하위 캐시로 전달하는 단계

를 포함하는, 전자 장치의 동작 방법.

청구항 12

이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링하고, 상기 샘플링된 상기 이전 요구 메모리 주소들 간의 보폭에 기반하여 프리패치 메모리 요청을 생성하고, 상기 프리패치 메모리 요청에 따른 프리패치 메모리 응답을 프리로더에 전달하고, 상기 프리패치 메모리 응답에 포함되는 프리패치 데이터 및 상기 프리패치 데이터에 대응하는 프리패치 메모리 주소를 프리로더 캐시에 저장하는 프로세서를 포함하고,

상기 프로세서는,

상기 프리 패치 메모리 요청을 생성하는 상기 프리로더를 포함하고,

상기 프리로더는,

상기 프리패치 메모리 주소 및 상기 프리 패치 데이터를 저장하는 상기 프리로더 캐시

를 포함하는, 전자 장치.

청구항 13

제12항에 있어서,

상기 프로세서는,

상기 보폭에 기반하여 상기 프리패치 메모리 주소를 예측하고, 상기 프리패치 메모리 주소를 포함하는 상기 프리패치 메모리 요청을 생성하는, 전자 장치.

청구항 14

제13항에 있어서,

상기 보폭은,

인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭 중 하나 또는 둘 이상의 조합을 포함하고,

상기 프로세서는,

상기 인트라 워프 패턴 보폭의 우선 순위를 상기 인터 워프 패턴 보폭 보다 높게 결정하여 상기 프리패치 메모리 요청을 결정하는, 전자 장치.

청구항 15

제13항에 있어서,

상기 프로세서는,

상기 보폭이 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭을 모두 포함하고 상기 인트라 워프 패턴 보폭 및 상기 인터 워프 패턴 보폭 각각에 기반한 상기 프리패치 메모리 주소가 상이한 경우, 상기 인트라 워프 패턴 보폭에 기반한 상기 프리패치 메모리 요청을 생성하는, 전자 장치.

청구항 16

제12항에 있어서,

상기 프로세서는,

LDST 유닛으로부터 다음 연산을 위한 요구 메모리 요청을 상기 프리로더로 전달하고, 상기 요구 메모리 요청에 포함된 요구 메모리 주소 및 상기 프리패치 메모리 주소에 기반하여 상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는, 전자 장치.

청구항 17

제16항에 있어서,

상기 프로세서는,

상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 경우, 상기 프리패치 메모리 주소에 대응하는 상기 프리패치 데이터를 상기 프리로더 캐시로부터 상기 LDST 유닛으로 전송하는, 전자 장치.

청구항 18

제16항에 있어서,

상기 프로세서는,

상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 상이한 경우, 상기 요구 메모리 요청을 하위 캐시로 전달하고, 상기 요구 메모리 요청을 수신한 상기 하위 캐시로부터 상기 데이터를 상기 LDST 유닛으로 전송하는, 전자 장치.

청구항 19

제12항에 있어서,

상기 프리로더 캐시는,

복수의 엔트리들을 포함하고,

상기 복수의 엔트리들 각각은 엔트리의 상태를 나타내는 상태 비트를 포함하는, 전자 장치.

청구항 20

제12항에 있어서,

상기 프로세서는,

복수의 서브 프리패치 메모리 요청들을 생성하고, 상기 복수의 서브 프리패치 메모리 요청들에 기반하여 상기 프리패치 메모리 주소를 생성하는, 전자 장치.

발명의 설명

기술 분야

[0001] 메모리 주소를 예측하여 데이터를 프리페칭하는 전자 장치 및 그 동작 방법이 도시된다.

배경 기술

[0002] 프로세서는 머신 러닝 및 블록 체인과 같은 애플리케이션을 실행할 수 있다. 프로세서에서 애플리케이션을 실행할 때 하나의 명령어에 의해 연산이 종료된 후 다음 명령어에 의한 연산이 실행되어 파이프라인 스톱(pipeline stall)이 발생할 수 있다. 파이프라인 스톱이 감소할수록 프로세서는 더 높은 성능을 발휘할 수 있다. 따라서, 이러한 파이프라인 스톱을 줄이기 연구가 활발히 수행되고 있다.

발명의 내용

해결하려는 과제

[0003] 본 개시는 연산에 이용된 메모리 주소를 이용하여 다음 연산에 필요한 메모리를 프리로더 캐시에 미리 로딩함으로써 파이프라인 스톱을 줄일 수 있다.

[0004] 본 개시는 프리로더 캐시에 다음 연산에 필요한 메모리를 프리페치함으로써 데이터가 L1 캐시에서 적출되더라도 다시 데이터를 L1 캐시로 로딩할 필요가 없어 파이프라인 스톱을 줄일 수 있다.

과제의 해결 수단

[0005] 본 개시의 일실시예에 따른 전자 장치의 동작 방법은 이전 요구 메모리 요청(prior demand memory request)에 포함된 이전 요구 메모리 주소들(prior demand memory address)을 샘플링하는 단계, 상기 샘플링된 상기 이전 요구 메모리 주소들 간의 보폭(stride)에 기반하여 프리페치 메모리 요청(prefetch memory request)을 생성하는 단계, 상기 프리페치 메모리 요청에 따른 프리페치 메모리 응답(prefetch response)을 프리로더에 전달하는 단계, 및 상기 프리페치 메모리 응답에 포함되는 프리페치 데이터 및 상기 프리페치 데이터에 대응하는 프리페치 메모리 주소(prefetch memory address)를 프리로더 캐시(preloader cache)에 저장하는 단계를 포함할 수 있다.

[0006] 상기 프리페치 메모리 요청을 생성하는 단계는, 상기 보폭에 기반하여 상기 프리페치 메모리 주소를 예측하고, 상기 프리페치 메모리 주소를 포함하는 상기 프리페치 메모리 요청을 생성할 수 있다.

[0007] 상기 보폭은, 인트라 워프 패턴 보폭 (intra-warp pattern stride) 및 인터 워프 패턴 보폭 (inter-warp pattern stride) 중 하나 또는 둘 이상의 조합을 포함하고, 상기 프리페치 메모리 요청을 생성하는 단계는, 상기 인트라 워프 패턴 보폭의 우선 순위를 상기 인터 워프 패턴 보폭 보다 높게 결정하여 상기 프리페치 메모리 요청을 결정할 수 있다.

[0008] 상기 프리페치 메모리 요청을 생성하는 단계는, 상기 보폭이 인트라 워프 패턴 및 인터 워프 패턴 보폭을 모두 포함하고 상기 인트라 워프 패턴 보폭 및 상기 인터 워프 패턴 보폭 각각에 기반한 상기 프리페치 메모리 주소가 상이한 경우, 상기 인트라 워프 패턴 보폭에 기반한 상기 프리페치 메모리 요청을 생성할 수 있다.

[0009] LDST 유닛(load/store unit)으로부터 다음 연산을 위한 요구 메모리 요청 (demand memory request)을 상기 프리로더로 전달하는 단계, 및 상기 요구 메모리 요청에 포함된 요구 메모리 주소(demand memory address) 및 상기 프리페치 메모리 주소에 기반하여 상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는 단계를 더 포함할 수 있다.

[0010] 상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는 단계는, 상기 요구 메모리 주소와 상기 프리페치 메모리 주소가 일치하는 경우, 상기 프리페치 메모리 주소에 대응하는 상기 프리페치 데이터를 상기 프리로더 캐시로부터 상기 LDST 유닛으로 전송할 수 있다.

[0011] 상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송하는 단계는, 상기 요구 메모리 주소와 상기 프리페치 메모리 주소가 상이한 경우, 상기 요구 메모리 요청을 하위 캐시로 전달하고, 상기 요구 메모리 요청을 수신한 상기 하위 캐시로부터 상기 데이터를 상기 LDST 유닛으로 전송할 수 있다.

- [0012] 상기 프리로더 캐시는, 복수의 엔트리들을 포함하고, 상기 복수의 엔트리들 각각은 엔트리의 상태를 나타내는 상태 비트를 포함할 수 있다.
- [0013] 상기 프리패치 메모리 요청이 생성되었는지 여부, 요구 메모리 요청이 생성되기 전에 상기 프리패치 메모리 응답이 프리로더에 도달했는지 여부, 및 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 여부에 따라 상기 상태 비트들을 업데이트하는 단계를 더 포함할 수 있다.
- [0014] 상기 프리로더 캐시에 저장하는 단계는, 복수의 서브 프리패치 메모리 요청들(sub-prefetch memory requests)을 생성하고, 상기 복수의 서브 프리패치 메모리 요청들에 기반하여 상기 프리패치 메모리 주소를 저장할 수 있다.
- [0015] 본 개시의 일실시예에 따른, 전자 장치의 동작 방법은 샘플링된 이전 요구 메모리 주소들의 보폭에 기반하여 프리패치 메모리 요청을 생성하는 단계, 상기 프리패치 메모리 요청에 대한 응답인 프리패치 메모리 응답에 포함되는 프리패치 데이터 및 상기 프리패치 데이터에 대응하는 프리패치 메모리 주소(prefetch memory address)를 프리로더 캐시에 저장하는 단계, LDST 유닛으로부터 생성된 다음 연산을 위한 요구 메모리 요청에 포함된 요구 메모리 주소 및 상기 프리로더 캐시에 저장된 상기 프리패치 메모리 주소가 일치하는지 여부를 판단하는 단계, 및 상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 경우, 상기 프리패치 데이터를 상기 LDST 유닛으로 전송하고, 상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하지 않는 경우, 상기 요구 메모리 요청을 상기 프리로더 캐시의 하위 캐시로 전달하는 단계를 포함할 수 있다.
- [0016] 본 발명의 일실시예에 따른 전자 장치는, 이전 요구 메모리 요청(prior demand memory request)에 포함된 이전 요구 메모리 주소들(prior demand memory address)을 샘플링하고, 상기 샘플링된 상기 이전 요구 메모리 주소들 간의 보폭(stride)에 기반하여 프리패치 메모리 요청(prefetch memory request)을 생성하고, 상기 프리패치 메모리 요청에 따른 프리패치 메모리 응답(prefetch response)을 프리로더에 전달하고, 상기 프리패치 메모리 응답에 포함되는 프리패치 데이터 및 상기 프리패치 데이터에 대응하는 프리패치 메모리 주소(prefetch memory address)를 프리로더 캐시에 저장하는 프로세서를 포함하고, 상기 프로세서는, 상기 프리 패치 메모리 요청을 생성하는 상기 프리로더를 포함하고, 상기 프리로더는, 상기 프리패치 메모리 주소 및 상기 프리 패치 데이터를 저장하는 상기 프리로더 캐시를 포함할 수 있다.
- [0017] 상기 프로세서는, 상기 보폭에 기반하여 상기 프리패치 메모리 주소를 예측하고, 상기 프리패치 메모리 주소를 포함하는 상기 프리패치 메모리 요청을 생성할 수 있다.
- [0018] 상기 보폭은, 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭 중 하나 또는 둘 이상의 조합을 포함하고, 상기 프로세서는, 상기 인트라 워프 패턴 보폭의 우선 순위를 상기 인터 워프 패턴 보폭 보다 높게 결정하여 상기 프리패치 메모리 요청을 결정할 수 있다.
- [0019] 상기 프로세서는, 상기 보폭이 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭을 모두 포함하고 상기 인트라 워프 패턴 보폭 및 상기 인터 워프 패턴 보폭 각각에 기반한 상기 프리패치 메모리 주소가 상이한 경우, 상기 인트라 워프 패턴 보폭에 기반한 상기 프리패치 메모리 요청을 생성할 수 있다.
- [0020] 상기 프로세서는, LDST 유닛(load/store unit)으로부터 다음 연산을 위한 요구 메모리 요청 (demand memory request)을 상기 프리로더로 전달하고, 상기 요구 메모리 요청에 포함된 요구 메모리 주소(demand memory address) 및 상기 프리패치 메모리 주소에 기반하여 상기 다음 연산에 필요한 데이터를 상기 LDST 유닛에 전송할 수 있다.
- [0021] 상기 프로세서는, 상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 일치하는 경우, 상기 프리패치 메모리 주소에 대응하는 상기 프리패치 데이터를 상기 프리로더 캐시로부터 상기 LDST 유닛으로 전송할 수 있다.
- [0022] 상기 프로세서는, 상기 요구 메모리 주소와 상기 프리패치 메모리 주소가 상이한 경우, 상기 요구 메모리 요청을 하위 캐시로 전달하고, 상기 요구 메모리 요청을 수신한 상기 하위 캐시로부터 상기 데이터를 상기 LDST 유닛으로 전송할 수 있다.
- [0023] 상기 프리로더 캐시는, 상기 프리로더 캐시에 포함되는 엔트리들의 상태를 나타내는 상태 비트를 포함할 수 있다.
- [0024] 상기 프로세서는, 복수의 서브 프리패치 메모리 요청들(sub-prefetch memory requests)을 생성하고, 상기 복수의 서브 프리패치 메모리 요청들을 병합하여 상기 프리패치 메모리 주소를 생성할 수 있다.

발명의 효과

[0025] 본 개시의 일실시예에 따르면, 다음 연산에 필요한 메모리를 프리로더 캐시에 미리 로딩함으로써 파이프라인 스톱을 줄일 수 있다.

[0026] 본 개시의 일실시예에 따르면, 프리로더 캐시에 다음 연산에 필요한 메모리를 프리패치함으로써 데이터가 L1 캐시에서 적출되더라도 다시 데이터를 L1 캐시로 로딩할 필요가 없다.

도면의 간단한 설명

[0027] 도 1은 본 개시의 일실시예에 따른 전자 장치를 도시한 도면이다.

도 2는 본 개시의 일실시예에 따른 GPU를 설명하기 위한 도면이다.

도 3은 보폭을 설명하기 위한 도면이다.

도 4a 및 도 4b는 복수의 프리패칭 시나리오를 설명하기 위한 도면이다.

도 5는 본 개시의 일실시예에 따른 프리로더를 포함하는 전자 장치의 동작을 설명하기 위한 플로우 차트이다.

도 6은 본 개시의 일실시예에 따른 GPU의 동작을 설명하기 위한 플로우 차트이다.

도 7은 본 개시의 일실시예에 따른 프리로더의 동작을 설명하기 위한 도면이다.

도 8은 본 개시의 일실시예에 따른 엔트리의 상태 업데이트를 설명하기 위한 도면이다.

도 9는 본 개시의 일실시예에 따른 전자 장치의 동작을 설명하기 위한 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0028] 이하, 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0029] 아래 설명하는 실시예들에는 다양한 변경이 가해질 수 있다. 아래 설명하는 실시예들은 실시 형태에 대해 한정하려는 것이 아니며, 이들에 대한 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0030] 제1 또는 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 이해되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.

[0031] 실시예에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 실시예를 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수 개의 표현을 포함한다. 본 명세서에서, "A 또는 B", "A 및 B 중 적어도 하나", "A 또는 B 중 적어도 하나", "A, B 또는 C", "A, B 및 C 중 적어도 하나", 및 "A, B, 또는 C 중 적어도 하나"와 같은 문구들 각각은 그 문구들 중 해당하는 문구에 함께 나열된 항목들 중 어느 하나, 또는 그들의 모든 가능한 조합을 포함할 수 있다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 명세서 상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0032] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0033] 또한, 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다. 실시예를 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 실시예의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다.

[0034] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0036] 도 1은 본 개시의 일실시예에 따른 전자 장치를 도시한 도면이다.

- [0037] 도 1을 참조하면 전자 장치(100)가 도시된다. 전자 장치(100)는 프로세서(110)를 포함할 수 있다. 도 1에 도시된 전자 장치(100)에는 본 개시의 실시예들과 관련된 구성요소들만이 도시되어 있다. 따라서, 전자 장치(100)는 도 5에 도시된 구성 요소 외에 다른 범용적인 구성요소들을 더 포함할 수 있음은 통상의 기술자에게 자명하다.
- [0038] 프로세서(110)는 전자 장치(100)를 제어하기 위한 전반적인 기능을 수행하는 역할을 수행할 수 있다. 프로세서(110)는 메모리(미도시)에 저장된 프로그램들 및/또는 명령어들을 실행함으로써, 전자 장치(100)를 전반적으로 제어할 수 있다. 프로세서(110)는 전자 장치(100) 내에 구비된 CPU(central processing unit), GPU(graphics processing unit), TPU(tensor processing unit) 및 NPU(neural processing unit) 등으로 구현될 수 있으나, 이에 제한되지 않는다.
- [0039] 프로세서(110)는 프리로더(preloader)(111)를 포함할 수 있다. 프리로더(111)는 캐시(미도시)를 포함할 수 있다. 프리로더(111)는 프리패치의 일종이지만, 캐시(미도시)를 포함하지 않는 프리패치와 구별하기 위해 프리로더(111)라고 하겠다. 프리로더(111)에 포함되는 캐시(미도시)를 프리로더 캐시라고 할 수 있다. 프리로더(111)는 이전 연산에서 이미 생성된 메모리 주소를 사용하여 보폭(stride)을 계산할 수 있다. 프리로더(111)는 이 보폭을 이용하여 다음 메모리 주소를 예측할 수 있다. 프리로더(111)는 예측한 다음 메모리 주소에 저장된 데이터를 프리로더 캐시로 미리 로딩하는 프리패치 메모리 요청을 생성할 수 있다.
- [0040] 이하에서는, 설명의 편의를 위해 프로세서(110)의 예시 중 하나인 GPU를 이용하여 본 개시를 설명하도록 하겠다. 다만, 이하의 설명이 GPU뿐만 아니라 프리로더를 포함하는 경우, CPU, TPU 및 NPU에도 적용될 수 있음은 통상의 기술자에게 자명하다.
- [0042] 도 2는 본 개시의 일실시예에 따른 GPU를 설명하기 위한 도면이다.
- [0043] 도 2를 참조하면, 본 개시의 실시예에 따른 GPU(200)가 도시된다. GPU(200)는 스레드 블록 스케줄러(thread block scheduler), 복수의 스트리밍 멀티 프로세서들(streaming multiprocessor)(210-1, 210-2, 210-N), 인터 커넥션(inter connection), 공유 L2 캐시(shared L2 cache)(220) 및 글로벌 메모리(global memory)(230)를 포함할 수 있다.
- [0044] 복수의 스트리밍 멀티프로세서들(210-1, 210-2, 210-N)(이하, SM) 각각은 동시에 수천개의 스레드들(threads)을 실행할 수 있다. 각 SM은 스트리밍 프로세서(streaming processor, 이하 SP)(213)라고 불리는 수 백개의 코어들을 포함할 수 있다. SP(213) 산술 연산(arithmetic operation)을 수행할 수 있다. SM들에 스케줄링된 스레드들은 복수의 워프들(warps)로 나누어질 수 있다. 복수의 워프들은 워프 스케줄러(211)에 의해 다양한 스케줄링 정책(scheduling policy)에 의해 SP(213)들에 스케줄링될 수 있다.
- [0045] GPU(200)는 CPU와 유사한 메모리 계층(memory hierarchy)을 가질 수 있다. SM은 큰 크기의 레지스터 파일(register file)(212)을 포함할 수 있다. SM은 레지스터 파일(212)의 하위 계층으로 L1 캐시(216) 및 스크래치패드 메모리(scratchpad memory)(217)를 포함할 수 있다. 마지막으로, GPU(200)는 스케줄링된 모든 스레드들에 의해 공유되는 공유 L2 캐시(220) 및 글로벌 메모리(230)를 포함할 수 있다.
- [0046] 데이터가 명령어에 의해 사용되기 위해서는 레지스터 파일(212)에 로딩되어야 할 수 있다. 따라서, 데이터를 레지스터 파일(212)의 직전 하위 캐시에 미리 로딩하는 프리패칭(prefetching)을 통해 메모리 지연을 줄일 수 있다.
- [0047] SM은 프리로더(preloader)(215)를 포함할 수 있다. 프리로더(215)는 프리패치를 수행할 수 있다. 프리로더(215)는 프리패치를 수행하는 프리패처(prefetcher)의 일종일 수 있다. 다만, 프리로더(215)는 내부에 프리패치 메모리 주소 및 프리패치 데이터를 저장하는 프리로더 캐시를 더 포함할 수 있다.
- [0048] 프리로더(215)는 하위 캐시인 L1 캐시(216), 공유 L2 캐시(220) 또는 글로벌 메모리(230)로부터 데이터를 프리패치(prefetch)할 수 있다. 이하에서, L1 캐시(216), 공유 L2 캐시(220) 또는 글로벌 메모리(230)로부터 프리패치되는 데이터를 프리패치 데이터(prefetch data)라고 하겠다. 또한, 프리패치 데이터가 저장된 메모리의 주소를 프리패치 메모리 주소라고 하겠다. 프리로더(215)는 프리패치 데이터를 프리로더 캐시에 저장할 수 있다. 프리로더 캐시는 L1 캐시보다 상위라는 의미에서 L0 캐시(L0 cache)라고도 할 수 있다. 다시 말해, L0 캐시는 프리패치 데이터의 저장에 사용될 수 있다. 프리로더(215)는 LDST 유닛(load/store unit)으로부터 요구 메모리 요청(demand memory request)을 수신하면, 요구 메모리 요청에 포함된 요구 메모리 주소와 프리로더 캐시에 포함된 프리패치 메모리 주소를 비교할 수 있다. 비교 결과에 따라서, 프리로더(215)는 프리패치 데이터를 LDST 유

닛에 전달하거나 요구 메모리 요청을 하위 캐시로 전달할 수 있다.

- [0049] 이하에서는, 프리로더(215) 및 프리패치가 프리패치 메모리 주소(prefetch memory address)를 예측하는 방법에 대해서 설명하도록 하겠다.
- [0051] 도 3은 보폭을 설명하기 위한 도면이다. 도 3에서 상술한 프리로더 캐시를 포함하는 프리로더가 아닌 프리로더 캐시를 포함하지 않는 프리패처를 기준으로 보폭을 결정하고 데이터를 L1 캐시로 프리패치하는 방법을 설명하겠다.
- [0052] 다만, 이하에서 설명하는 방법은 프리로더 캐시를 포함하는 프리로더에도 적용될 수 있음은 통상의 기술자에게 자명하다. 이때, 프리로더가 아래의 방법을 수행하는 경우, 도 6 및 도 7에서 설명되듯이 프리패치 데이터는 L1 캐시가 아닌 프리로더 캐시에 저장될 수 있다.
- [0053] 도 3을 참조하면, 프리패치 메모리 주소를 예측하기 위해 보폭(stride)을 결정하는 방법이 도시된다. 구체적으로, 도 3의 (A)는 인트라 워프 패턴 보폭(intra-warp pattern stride)을 결정하고 데이터를 프리패치하는 방법이 도시된다. 도 3의 (B)는 인터 워프 패턴 보폭(inter-warp pattern stride)을 결정하고 데이터를 프리패치하는 방법이 도시된다.
- [0054] 도 3에서, 블록(301)은 이전 요구 메모리 요청에 따른 이전 요구 메모리 주소를 나타낼 수 있다. 블록(303)은 프리패치 메모리 요청(prefetch memory request)에 포함된 프리패치 메모리 주소를 나타낼 수 있다. 블록(305)은, 다음 연산을 위한 요구 메모리 요청에 포함된 요구 메모리 주소를 나타낼 수 있다.
- [0055] 이전 요구 메모리 요청에 따른 이전 요구 메모리 주소는 이전 연산에 이용된 데이터가 저장된 메모리 주소를 의미할 수 있다. 프리패치 메모리 요청에 포함된 프리패치 메모리 주소는 프리패처가 다음 연산에 이용될 것이라고 예측한 데이터가 저장된 메모리 주소를 의미할 수 있다. 다음 연산을 위한 요구 메모리 요청에 포함된 요구 메모리 주소는 LDST 유닛에 의해 요청된 다음 연산에 필요한 데이터가 저장된 메모리 주소를 의미할 수 있다.
- [0056] 인트라 워프 패턴 보폭은 동일한 워프로부터 생성된 요구 메모리 요청들 간의 보폭을 이용하여 프리패치 메모리 주소를 예측하는 알고리즘일 수 있다. 프리패처는 동일한 워프로부터 생성된 이전 요구 메모리 요청들에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다. 프리패처는 샘플링된 이전 요구 메모리 주소들 간의 보폭에 기반하여 프리패치 메모리 주소를 결정할 수 있다. 프리패처는 결정된 프리패치 메모리 주소에 저장된 데이터인 프리패치 데이터를 L1 캐시로 로딩하여 저장할 수 있다. 즉, 프리패처는 프리패치 데이터에 대해 프리패칭을 수행할 수 있다. 이후에, 다음 연산을 위한 요구 메모리 요청이 있는 경우, 프리패처는 요구 메모리 요청에 포함된 요구 메모리 주소와 L1 캐시에 저장된 프리패치 메모리 주소를 비교할 수 있다. 요구 메모리 주소와 프리패치 메모리 주소가 일치하는 경우, 프리패처는 L1 캐시에서 LDST 유닛으로 프리패치 데이터를 전달할 수 있다.
- [0057] 예를 들어, 도 3의 (A)를 참조하면, 프리패처는 동일한 워프 #00에서 생성된 이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다. 프리패처는 이전 요구 메모리 주소들 0x0100 및 0x0200 간의 보폭을 계산할 수 있다(도 3의 (A)에서 ①). 이때, 보폭은 0x0100 및 0x0200 간의 차이인 0x0100일 수 있다. 프리패처는 보폭을 이용하여 다음 연산에 필요한 프리패치 데이터가 저장된 프리패치 메모리 주소를 0x0300으로 예측할 수 있다(도 3의 (A)에서 ②). 프리패처는 0x0300에 위치한 프리패치 데이터를 L1 캐시에 로딩하여 저장할 수 있다. 다음 연산을 위해 워프 #00으로부터 요구 메모리 주소가 0x0300인 요구 메모리 요청이 생성되는 경우, LDST 유닛은 L1 캐시로 요구 메모리 요청을 전송할 수 있다. 이 때, L1 캐시에 저장되어 있고, 프리패치 메모리 주소 0x0300에 대응하는 프리패치 데이터는 LDST 유닛에 전송될 수 있다. 반면에, 요구 메모리 주소와 프리패치 메모리 주소가 상이한 경우, 요구 메모리 요청은 하위 캐시(예를 들어, L2 캐시)로 전달될 수 있다.
- [0058] 인터 워프 패턴 보폭은 상이한 워프로부터 생성된 요구 메모리 요청들 간의 보폭을 이용하여 프리패치 메모리 주소를 예측하는 알고리즘 일수 있다. 프리패처는 상이한 워프로부터 생성된 이전 요구 메모리 요청들에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다. 샘플링된 이전 요구 메모리 주소들을 이용하여 보폭을 결정하는 단계부터는 인트라 워프 패턴 보폭과 동일하므로 설명을 생략하도록 하겠다.
- [0059] 예를 들어, 도 3의 (B)를 참조하면, 프리패처는 상이한 워프 #00 및 워프 #01 각각으로부터 이전 요구 메모리 주소 0x0100 및 0x0200을 샘플링할 수 있다. 프리패처는 이전 요구 메모리 주소들 0x0100 및 0x0200을 이용하여 보폭을 0x0100으로 결정할 수 있다(도 3의 (B)에서 ①). 프리패처는 보폭을 이용하여 다음 연산을 위해 필요한 워프 #02가 필요로 하는 프리패치 데이터가 위치한 프리패치 메모리 주소를 0x0300으로 예측할 수 있다(도 3의

(B)에서 ②). 프리페처는 0x0300에 위치한 프리페치 데이터를 L1 캐시로 로딩하여 저장할 수 있다. 다음 연산을 위해 워프 #02으로부터 요구 메모리 주소가 0x0300인 요구 메모리 요청이 생성되는 경우, LDST 유닛은 L1 캐시로 요구 메모리 요청을 전송할 수 있다. 이때, L1 캐시에 저장되어 있고, 프리페치 메모리 주소 0x0300에 대응하는 프리페치 데이터는 LDST 유닛에 전송될 수 있다. 반면에, 요구 메모리 주소와 프리페치 메모리 주소가 상이한 경우, 요구 메모리 요청은 하위 캐시(예를 들어, L2 캐시)로 전달될 수 있다.

[0060] 이하에서는, L1 캐시에 데이터를 프리페치했을 때 발생할 수 있는 시나리오에 대해서 설명하도록 하겠다.

[0062] 도 4a 및 도 4b는 복수의 프리페칭 시나리오를 설명하기 위한 도면이다.

[0063] 도 4a 및 도 4b를 참조하면, GPU(400)는 스트리밍 멀티프로세서(410), 인터커넥션(420), L2 캐시(430) 및 글로벌 메모리(440)를 포함할 수 있다. 스트리밍 멀티프로세서(410)는 레지스터 파일(411), LDST 유닛(413), 프리페처(415) 및 L1 캐시(417)를 포함할 수 있다.

[0064] 도 4a의 (A)를 참조하면, ①에서, 프리페처(415)는 다음 연산에 이용될 프리페치 데이터의 주소인 프리페치 메모리 주소를 예측할 수 있다. 프리페처(415)는 프리페치 메모리 주소를 포함하는 프리페치 메모리 요청을 생성하고, 프리페치 메모리 요청을 L1 캐시에 전달할 수 있다. 프리페치 메모리 요청은 L1 캐시에 프리페치 메모리 주소에 대응하는 프리페치 데이터가 로딩되어 있는지 여부를 확인할 수 있다. 프리페치 데이터가 로딩되어 있지 않은 경우 캐시 미스(cache miss)가 발생할 수 있다. 캐시 미스가 발생한 경우, 프리페치 메모리 요청은 인터커넥션(420)을 통해 하위 캐시인 L2 캐시(430) 및/또는 글로벌 메모리(440)로 전달될 수 있다.

[0065] ②에서, 프리페처 메모리 주소에 대응하는 프리페치 데이터가 L2 캐시(430) 및/또는 글로벌 메모리(440)에 존재하는 경우, 프리페치 메모리 요청에 따른 프리페치 메모리 응답(prefetch memory response)이 생성될 수 있다. 프리페치 메모리 응답은 인터커넥션(420)을 통해 L1 캐시(417)로 전달될 수 있다. 프리페치 메모리 응답은 프리페처 메모리 주소에 대응하는 프리페치 데이터를 포함할 수 있다. 프리페치 데이터 및 해당 프리페치 데이터에 대응하는 프리페치 메모리 주소는 L1 캐시(417)에 저장될 수 있다.

[0066] ③에서, LDST 유닛(413)은 L1 캐시(417)에 저장된 데이터를 레지스터 파일(411)로 로딩하기 위해 요구 메모리 요청을 생성할 수 있다. 요구 메모리 요청이 L1 캐시(417)에 전달되면, 요구 메모리 요청에 포함된 요구 메모리 주소는 L1 캐시(417)에 저장된 프리페치 메모리 주소와 비교될 수 있다. 요구 메모리 주소와 프리페치 메모리 주소가 일치하는 경우, 캐시 히트(cache hit)가 발생할 수 있다.

[0067] ④에서, 캐시 히트가 발생한 경우, 프리페치 데이터는 L1 캐시(417)로부터 LDST 유닛(413)을 통해 레지스터 파일(411)로 전송될 수 있다. 캐시 히트가 발생한 경우, 요구 메모리 요청은 L1 캐시(417)의 하위 캐시로 전달되지 않을 수 있다. 캐시 미스가 발생한 경우, 요구 메모리 요청은 L1 캐시(417)의 하위 캐시인 L2 캐시(430) 및/또는 글로벌 메모리(440)으로 전달될 수 있다.

[0068] 즉, 프리페치 메모리 요청이 발생한 후 요구 메모리 요청이 발생하여 프리페치 데이터가 미리 L1 캐시(417)로 로딩될 수 있다. 이때, 요구 메모리 요청이 L1 캐시(417)의 하위 캐시로 전달될 필요가 없어 메모리 지연이 감소되어 높은 성능 향상을 획득할 수 있다.

[0069] 도 4a의 (B)를 참조하면, ①에서, 프리페처(415)는 다음 연산에 이용될 프리페치 데이터의 주소인 프리페치 메모리 주소를 예측할 수 있다. 프리페처(415)는 프리페치 메모리 주소를 포함하는 프리페치 메모리 요청을 생성하고, 프리페치 메모리 요청을 L1 캐시에 전달할 수 있다. 이때, 프리페치 데이터가 L1 캐시(417)에 이미 로딩되어 있어 캐시 히트가 발생할 수 있다. 프리페치 메모리 요청은 캐시 히트가 발생하여 삭제될 수 있다.

[0070] ②에서, LDST 유닛(413)은 L1 캐시에 저장된 데이터를 레지스터 파일(411)로 로딩하기 위해 요구 메모리 요청을 생성할 수 있다. 요구 메모리 요청이 L1 캐시(417)에 전달되면, 요구 메모리 요청에 포함된 요구 메모리 주소는 L1 캐시(417)에 저장된 프리페치 메모리 주소와 비교될 수 있다. 요구 메모리 주소와 프리페치 메모리 주소가 일치하는 경우, 캐시 히트(cache hit)가 발생할 수 있다.

[0071] ③에서, 캐시 히트가 발생한 경우, 프리페치 데이터는 L1 캐시(417)로부터 LDST 유닛(413)을 통해 레지스터 파일(411)로 전송될 수 있다.

[0072] 다만, 도 4a의 (B)는 이미 L1 캐시(417)에 데이터가 로딩되어 있으므로, 프리페칭에 의한 효과는 존재하지 않는다.

- [0073] 도 4b의 (C)를 참조하면, ① 및 ② 단계는 도 4a의 (A)에서 설명한 바 설명을 생략하겠다.
- [0074] GPU은 수 천개의 쓰레드를 동시에 수행하여 L1 캐시(417)를 공유하는 구조일 수 있다. 따라서, L1 캐시(417)의 크기는 제한적일 수 있다.
- [0075] 따라서, ③에서, ① 및 ② 단계를 통해 L1 캐시(417)에 저장된 프리패치 데이터가 적출(eviction)될 수 있다. 예를 들어, L1 캐시(417)에 요구 메모리 요청이 도달하기 전에 다른 워프(또는, 쓰레드)의 요구 메모리 요청에 의해 프리패치 데이터가 적출되고, 다른 워프(또는, 쓰레드)가 필요로 하는 데이터가 쓰여 질 수 있다.
- [0076] ④에서, 요구 메모리 요청이 생성되었지만, 이미 프리패치 데이터가 적출된 이후일 수 있다. 따라서, 요구 메모리 요청이 L1 캐시(417)에 전달되어도 프리 패치 데이터는 이미 적출되었으므로 캐시 미스가 발생할 수 있다. 따라서, 요구 메모리 요청은 인터커넥션(420)을 통해 L2 캐시(430) 또는 글로벌 메모리(440)로 전달될 수 있다.
- [0077] ⑤에서, 요구 메모리 요청에 따른 요구 메모리 응답으로 데이터가 L2 캐시(430) 또는 글로벌 메모리(440)로부터 LDST 유닛(413)을 통해 레지스터 파일(411)로 전송될 수 있다.
- [0078] 다시 말해, 프리패치 데이터가 정상적으로 L1 캐시(417)에 로딩되었음에도 적출됨으로써 다시 L2 캐시(430) 또는 글로벌 메모리(440)로부터 데이터를 로딩하기 위해 수백 사이클이 소요되는 문제가 있을 수 있다. 따라서, L1 캐시(417), 인터커넥션(420), L2 캐시(430) 및 글로벌 메모리(440)의 부하가 증가할 수 있다. 결국, 메모리 지연이 급격히 증가하여 GPU의 성능이 감소할 수 있다.
- [0079] 도 4b의 (D)를 참조하면, ①에서, 프리패치(415)가 다음 연산에 필요한 프리패치 데이터가 위치한 프리패치 메모리 주소를 예측하였으나, 예측이 잘못될 수 있다. 즉, 도 4a의 (A)에서 설명한 ① 및 ② 단계를 통해 프리패치 데이터를 L1 캐시(417)에 로딩할 수 있지만, 예측이 틀려 프리패치 데이터가 불필요하게 L1 캐시(417)의 공간만을 점유할 수 있다.
- [0080] ③에서, LDST 유닛(413)이 L1 캐시(417)에 저장된 데이터를 레지스터 파일(411)로 로딩하기 위해 요구 메모리 요청을 생성했으나, 프리패치 메모리 주소의 예측이 틀려 캐시 미스가 발생할 수 있다. 캐시 미스가 발생한 이후의 단계는 도 4b의 (C)에서 상술한바 설명을 생략하도록 하겠다.
- [0081] 결과적으로, 예측이 틀린 경우에도 프리패치 데이터가 적출된 경우와 마찬가지로 메모리 지연이 급격히 증가하여 GPU의 성능이 감소하는 문제가 있을 수 있다. 다만, GPU에서 실행되는 애플리케이션은 대체로 일정한 보폭을 갖기 때문에 예측이 잘못되는 경우는 제한적으로 발생할 수 있다.
- [0082] 이하에서는, 별도의 프리로더 캐시를 포함하는 프리로더를 이용하여 상술한 문제를 해결하는 방법에 대해서 설명하겠다.
- [0084] 도 5는 본 개시의 일실시예에 따른 프리로더를 포함하는 전자 장치의 동작을 설명하기 위한 플로우 차트이다.
- [0085] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(501) 내지 단계(509)는 프로세서에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 프리로더에 의해 수행될 수도 있다.
- [0086] 단계(501)에서, 프로세서는 이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다.
- [0087] 프로세서는 동일한 워프의 이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다. 또는, 프로세서는 상이한 워프의 이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다.
- [0088] 단계(502)에서, 프로세서는 샘플링된 이전 요구 메모리 주소들 간의 보폭에 기반하여 프리패치 메모리 요청을 생성할 수 있다.
- [0089] 프로세서는 동일한 워프의 이전 요구 메모리 주소들 간의 보폭에 기반하여 인트라 워프 패턴 보폭을 결정할 수 있다. 프로세서는 상이한 워프의 이전 요구 메모리 주소들 간의 보폭에 기반하여 인터 워프 패턴 보폭을 결정할 수 있다. 프로세서는 인트라 워프 패턴 보폭 및/또는 인터 워프 패턴 보폭에 기반하여 다음 연산에 필요한 데이터가 저장된 프리패치 메모리 주소를 예측할 수 있다. 프리 패치 메모리 요청은 예측된 프리패치 메모리 주소를 포함할 수 있다.

- [0090] 프로세서가 보폭을 결정한 결과, 보폭이 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭을 모두 포함할 수 있다. 이때, 인트라 워프 패턴 보폭의 우선순위가 인터 워프 패턴 보폭보다 높을 수 있다. 따라서, 보폭을 결정한 결과, 보폭이 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭을 모두 포함하고, 인트라 워프 패턴 보폭 및 인터 워프 패턴 보폭이 상이한 경우, 프로세서는 인트라 워프 패턴 보폭에 기반하여 프리패치 메모리 요청을 생성할 수 있다.
- [0091] 단계(503)에서, 프리패치 메모리 요청에 따른 프리패치 메모리 응답을 프리로더에 전달할 수 있다.
- [0092] 프로세서는 프리패치 메모리 요청을 프리로더에 포함되는 프리로더 캐시의 하위 캐시로 전달할 수 있다. 프리로더 캐시의 하위 캐시는 L1 캐시, L2 캐시 및 글로벌 메모리일 수 있다. 프리패치 메모리 요청을 수신한 하위 캐시에 프리패치 데이터가 저장되어 있는 경우, 프리패치 메모리 응답이 생성될 수 있다. 프로세서는 생성된 프리패치 메모리 응답을 프리로더에 전송할 수 있다. 프리패치 메모리 응답은 프리패치 데이터를 포함할 수 있다. 예를 들어, 프리패치 데이터가 L2 캐시에 존재하는 경우, L2 캐시에서 프리패치 메모리 응답이 생성될 수 있다. 프리패치 메모리 응답은 L2 캐시로부터 프리로더로 전송될 수 있고, 프리패치 메모리 응답은 프리패치 데이터를 포함할 수 있다.
- [0093] 단계(504)에서, 프로세서는 프리패치 메모리 요청에 포함되는 프리패치 데이터 및 프리패치 데이터에 대응하는 프리패치 메모리 주소를 프리로더 캐시에 저장할 수 있다.
- [0094] 프로세서는 요구 메모리 요청이 프리로더에 수신되었을 때 프리패치 메모리 주소가 제대로 예측되었는지 여부를 확인하기 위해 프리패치 메모리 주소를 저장할 수 있다. 요구 메모리 요청은 요구 메모리 주소를 포함할 수 있다. 요구 메모리 주소는 다음 연산을 위해 명령어가 필요한 데이터가 저장된 주소를 의미할 수 있다. 따라서, 프리패치 데이터가 정확히 프리로더 캐시에 로딩된 경우, 프리패치 메모리 주소와 요구 메모리 주소는 일치할 수 있다.
- [0095] 단계(505)에서, 프로세서는 LDST 유닛에서 생성된 다음 연산을 위한 요구 메모리 요청을 프리로더로 전달할 수 있다.
- [0096] 단계(506)에서, 프로세서는 요구 메모리 주소와 프리로더 캐시에 저장된 프리패치 메모리 주소가 일치하는 지 여부를 확인할 수 있다.
- [0097] 단계(507)에서, 요구 메모리 주소와 프리로더 캐시에 저장된 프리패치 메모리 주소가 일치하지 않는 경우, 프로세서는 요구 메모리 요청을 하위 캐시로 전달할 수 있다.
- [0098] 다시 말해, 프로세서는 요구 메모리 요청을 프리로더에 포함된 캐시의 하위 캐시로 요구 메모리 요청을 전달할 수 있다. 여기서, 하위 캐시는 L1 캐시, L2 캐시 및 글로벌 메모리를 포함할 수 있다.
- [0099] 단계(508)에서, 프로세서는 하위 캐시에 저장된 데이터를 LDST 유닛에 전송할 수 있다.
- [0100] 요구 메모리 요청을 수신한 하위 캐시가 요구 메모리 주소에 대응하는 데이터를 포함하고 있는 경우, 프로세서는 요구 메모리 응답을 LDST 유닛에 전송할 수 있다. 요구 메모리 응답은 요구 메모리 주소에 대응하는 데이터를 포함할 수 있다.
- [0101] 단계(509)에서, 요구 메모리 주소와 프리로더 캐시에 저장된 프리패치 메모리 주소가 일치하는 경우, 프로세서는 프리로더 캐시에 저장된 프리패치 데이터를 LDST 유닛으로 전송할 수 있다.
- [0102] 다시 말해, 프로세서는 요구 메모리 요청에 대한 응답으로 프리로더 캐시에 저장된 프리패치 데이터를 LDST 유닛으로 전송할 수 있다.
- [0103] 결국, 다음 연산에 필요하다고 예측된 데이터가 L1 캐시가 아닌 프리로더 캐시에 저장됨으로써 도 4b의 (C)와 같은 데이터의 적출을 방지할 수 있다.
- [0104] 이하에서는 상술한 프리로더를 포함하는 프로세서의 동작을 도면을 이용하여 설명하도록 하겠다.
- [0106] 도 6은 본 개시의 일실시예에 따른 GPU의 동작을 설명하기 위한 플로우 차트이다.
- [0107] 도 6을 참조하면, GPU(600)는 스트리밍 멀티프로세서(610), 인터커백션(620), L2 캐시(630) 및 글로벌 메모리(640)를 포함할 수 있다. 스트리밍 멀티프로세서(610)는 레지스터 파일(611), LDST 유닛(613), 프리로더(615)

및 L1 캐시(617)를 포함할 수 있다. 프리로더(615)는 내부 저장 장치인 캐시(미도시)를 포함할 수 있다. 이하에서는 프리로더(615)에 포함된 캐시(미도시)를 프리로더 캐시(preloader cache)라고 하겠다.

[0108] 프리로더(615)는 이전 요구 메모리 요청에 포함된 이전 요구 메모리 주소들을 샘플링할 수 있다. 프리로더(615)는 샘플링된 이전 요구 메모리 주소들을 이용하여 보폭을 결정할 수 있다. 프리로더(615)는 샘플링된 이전 요구 메모리 주소들을 보폭이 이용하여 인터 워프 패턴 보폭 및/또는 인트라 워프 패턴 보폭인지 여부를 결정할 수 있다. 프리로더(615)는 결정한 보폭에 기반하여 다음 연산에 이용될 데이터가 저장된 주소인 프리패치 메모리 주소를 예측할 수 있다. 다시 말해, 프리로더(615)는 LDST 유닛에 의해 생성될 요구 메모리 요청에 포함될 요구 메모리 주소를 예측할 수 있다.

[0109] 이후, 프리로더(615)는 프리패치 메모리 주소를 포함하는 프리패치 메모리 요청을 생성할 수 있다. 프리패치 메모리 요청은 L1 캐시(617)로 전달될 수 있다. L1 캐시에 프리패치 메모리 주소에 대응하는 프리패치 데이터가 존재하는 경우 캐시 히트가 발생할 수 있다. 캐시 히트가 발생한 경우, 프리패치 메모리 요청에 따른 프리패치 메모리 응답이 L1 캐시(617)로부터 프리로더(615)에 전송될 수 있다. 이때, 프리패치 메모리 응답은 프리패치 데이터를 포함할 수 있다. 프리로더(615)는 프리패치 메모리 응답을 수신하여 프리패치 데이터를 프리로더 캐시에 저장할 수 있다.

[0110] 반면에, L1 캐시(617)에 프리패치 메모리 주소에 대응하는 프리패치 데이터가 존재하지 않는 경우, 캐시 미스가 발생할 수 있다. 이때, 프리패치 메모리 요청은 인터커넥션(620)을 통해 하위 캐시로 전달될 수 있다. 프리패치 메모리 요청을 수신한 하위 캐시에 프리패치 메모리 주소에 대응하는 프리패치 데이터가 존재하는 경우, 캐시 히트가 발생하고 해당 하위 캐시에서 프리 패치 메모리 응답이 생성될 수 있다. 이때, 프리패치 메모리 응답은 프리로더(615)까지 전송될 수 있다. 프리로더(615)는 프리패치 메모리 응답에 포함된 프리패치 데이터를 프리로더 캐시에 저장할 수 있다.

[0111] 예를 들어, L2 캐시에도 프리패치 데이터가 존재하지 않는 경우, 캐시 미스가 발생하고 프리패치 메모리 요청은 글로벌 메모리(640)까지 전달될 수 있다. 글로벌 메모리(640)에 프리패치 데이터가 존재하는 경우, 프리패치 메모리 응답이 생성될 수 있다. 생성된 프리패치 메모리 응답은 프리로더(615)까지 전송될 수 있다.

[0112] 그 다음, LDST 유닛(613)에서 다음 연산을 위해 필요한 데이터를 레지스터 파일(611)에 로딩하기 위해 요구 메모리 요청을 생성할 수 있다. 요구 메모리 요청은 다음 연산을 위해 필요한 데이터가 저장된 요구 메모리 주소를 포함할 수 있다.

[0113] 요구 메모리 요청은 프리로더(615)로 전달될 수 있다. 프리로더 캐시에 저장된 프리패치 데이터의 프리패치 데이터 주소와 요구 메모리 요청의 요구 메모리 주소를 비교하여 동일한 경우, 프리패치 데이터는 LDST 유닛(613)을 통해 레지스터 파일(611)에 로딩될 수 있다.

[0114] 결국, 요구 메모리 요청이 L1 캐시(617) 또는 그 이하의 캐시까지 접근하지 않더라도 프리패치 데이터가 레지스터 파일(611)로 로딩될 수 있다. 다시 말해, 최소의 메모리 지연만으로 필요한 데이터를 레지스터 파일(611)에 로딩할 수 있다.

[0115] 이하에서는, 프리로더(615)의 내부 동작을 설명하도록 하겠다.

[0117] 도 7은 본 개시의 일실시예에 따른 프리로더의 동작을 설명하기 위한 도면이다.

[0118] 도 7을 참조하면, LDST 유닛(710), 프리로더(720) 및 L1 캐시(730)가 도시된다. 프리로더(720)는 프리패치 주소 생성기(721) 및 프리로더 캐시(723)를 포함할 수 있다. 프리로더 캐시(723)는 프리로더(720)의 내부 저장 장치일 수 있다. 프리로더(720)는 LDST 유닛(710)과 연결될 수 있다. 따라서, LDST 유닛(710)에서 생성되는 모든 요구 메모리 요청은 프리로더(720)를 통과할 수 있다. 최근 GPU는 128 바이트 크기의 캐시 라인을 4개의 32 바이트 크기의 섹터(sector)로 분할하는 섹터 L1 캐시를 이용할 수 있다. 섹터 L1 캐시는 128 바이트 크기의 캐시 라인을 모두 로딩하는 대신 섹터 측면에서 데이터를 로딩할 수 있다. L1 캐시(730)는 섹터 L1 캐시일 수 있다.

[0119] 프리로더 캐시(723)는 엔트리의 상태, 프리패치 메모리 주소 및 프리패치 데이터를 저장하기 위해 각 워프마다 하나의 엔트리를 포함할 수 있다. 따라서, 프리로더 캐시(723)는 총 64개의 엔트리를 포함할 수 있다. 다시 말해, 프리로더 캐시(723)는 총 64개의 행을 가질 수 있다. 엔트리를 각각은 워프 ID로 인덱싱될 수 있다. 예를 들어, 첫번째 엔트리는 워프 #00로 인덱싱 될 수 있고, 두번째 엔트리는 워프 #01로 인덱싱 될 수 있다. 엔트리들 각각은 엔트리의 상태를 나타내는 상태 비트 4 비트, 프리패치 메모리 주소를 나타내는 106 비트 및 프리패

치 데이터를 나타내는 1024 비트를 포함할 수 있다.

- [0120] 프리로더(720)는 섹터 크기의 데이터를 프리페칭하기 위해 복수의 서브 프리페치 메모리 요청(sub-prefetch memory request)들을 생성할 수 있다. 이하에서는 설명의 편의를 위해 프리페치 메모리 요청이 4개의 서브 프리페치 메모리 요청들로 구성된다고 가정하겠다. 따라서, 본 개시는 프리페치 메모리 요청이 4개인 경우에 한정되지 않음은 통상의 기술자에게 자명하다. 이때, 4개의 서브 프리페치 메모리 요청들 각각은 4개의 상이한 캐시 라인에 위치한 섹터의 데이터에 접근할 수 있다.
- [0121] 도 7의 (A)는 프리페치 메모리 요청을 생성하는 방법을 설명하기 위한 도면이다. 이하에서, ① 내지 ⑦은 설명의 편의를 위한 숫자일 뿐, 동작이 순차적으로 수행됨을 의미하는 것은 아니다.
- [0122] ①에서, 프리페치 주소 생성기(721)는 프리페치 메모리 요청을 생성할 수 있다. 프리페치 메모리 요청은 프리페치 메모리 주소를 포함하고 L1 캐시(730)으로 전달될 수 있다. 프리페치 주소 생성기(721)는 각 워프에 대해 서브 프리페치 메모리 요청을 4개까지 생성할 수 있다. 각 서브 프리페치 메모리 요청은 L1 캐시로 전달되는 하나의 섹터에 대한 요청일 수 있다. 다시 말해, 프리페치 메모리 요청은 4개의 서브 프리페치 메모리 요청으로 구성될 수 있다. 서브 프리페치 메모리 요청들 각각은 주소를 포함할 수 있다. 프리페치 메모리 주소는 4개의 서브 프리페치 메모리 요청들에 포함되는 주소로 구성될 수 있다.
- [0123] ②에서, 요구 메모리 요청에 포함된 요구 메모리 주소와 프리페치 메모리 요청에 포함된 프리페치 메모리 주소가 일치하는지 여부를 확인하기 위해 프리로더(720)는 프리페치 메모리 주소를 프리로더 캐시(723)에 저장할 수 있다. 이때, 프리페치 메모리 주소 전부를 저장하기보다 프리페치 메모리 주소를 구성하는 4개의 서브 프리페치 메모리 요청의 주소 일부만을 저장할 수 있다.
- [0124] 즉, ③에서, 4개의 서브 프리페치 메모리 요청의 주소를 모두 저장하기 보다 제1 서브 프리페치 메모리 요청에 포함되는 주소 및 다른 서브 프리페치 메모리 요청에 포함되는 주소의 오프셋이 저장할 수 있다. 오프셋은 20 비트로 제한되지만, LSB(least significant bit) 6 비트는 섹터 오프셋으로 이를 제외한 14비트만이 저장될 수 있다.
- [0125] 따라서, ⑥에서, 제1 서브 프리페치 메모리 요청에 포함되는 주소 64 비트 및 다른 서브 프리페치 메모리 요청들 각각의 오프셋 14비트가 결합될 수 있다.
- [0126] 결국, ⑦에서, 프리페치 메모리 주소는 106 비트로 변환되어 프리로더 캐시(723)에 저장될 수 있다.
- [0127] ④에서, 프리로더 캐시(723)에 상술한 106비트로 변환된 주소를 저장하기 위해 프리로더(720)는 워프 ID를 인덱스로 이용할 수 있다. 프리로더(720)는 하나의 워프에 대해 하나의 프리페치 메모리 요청을 생성할 수 있으므로, 프리로더 캐시(723)는 엔트리의 상태, 변환된 주소 및 프리페치 데이터를 저장하기 위해 64개의 엔트리를 가질 수 있다. 엔트리들 각각은 워프 ID로 인덱싱되므로, 프리로더(720)는 변환된 주소를 저장하기 위해 워프 ID를 인덱스로 이용할 수 있다.
- [0128] ⑤에서, 프리페치 메모리 요청이 L1 캐시(730)으로 전달된 경우, 엔트리의 상태를 나타내는 상태 비트가 업데이트 될 수 있다. 상태 비트는 프리페치 메모리 요청이 생성되었음을 나타내도록 업데이트 될 수 있다. 엔트리의 상태가 업데이트되는 방법은 도 8에서 더 설명하도록 하겠다.
- [0129] 도 7의 (B)는 LDST 유닛(710)으로부터 프리로더(720)로 요구 메모리 요청이 전달되었을 때를 설명하기 위한 도면이다. ① 내지 ⑥은 설명의 편의를 위한 숫자일 뿐, 동작이 순차적으로 수행됨을 의미하는 것은 아니다.
- [0130] 요구 메모리 요청은 4개의 요구 서브 메모리 요청들(demand sub-memory requests)으로 구성될 수 있다. 이하에서는 설명의 편의를 위해 요구 메모리 요청은 4개의 요구 서브 메모리 요청들로 구성된다고 가정하겠다. 다만, 본 개시는 요구 서브 메모리 요청들이 4개인 경우에 한정되지 않음은 통상의 기술자에게 자명하다. 요구 서브 메모리 요청들 각각은 요구 주소(demand address)를 포함할 수 있다. 요구 메모리 요청은 요구 메모리 주소를 포함할 수 있다. 요구 메모리 주소는 요구 서브 메모리 요청들의 요구 주소로 구성될 수 있다.
- [0131] ①에서, 요구 메모리 요청이 프리로더(720)에 도착하면, 프리로더(720)는 워프 ID를 이용하여 프리로더 캐시(723)에서 엔트리를 선택할 수 있다. 요구 메모리 요청은 특정 워프의 다음 연산에 대한 것일 수 있다. 따라서, 프리로더(720)는 이 특정 워프의 워프 ID를 이용하여 프리로더 캐시(723)에서 엔트리를 선택할 수 있다.
- [0132] ②에서, 요구 서브 메모리 요청들의 오프셋은 도 7의 (A)에서 프리페치 서브 메모리 주소와 동일한 방식으로 계산될 수 있다. 즉, 제1 요구 서브 메모리 요청을 제외한 다른 요구 서브 메모리 요청들의 요구 주소에서 오프셋

은 20 비트로 제한되지만, LSB 6비트를 제외한 14 비트만이 오프셋으로 계산될 수 있다.

- [0133] ③에서, 제1 요구 서브 메모리의 요구 주소 64 비트 및 다른 요구 서브 메모리 요청들 각각의 오프셋 14비트가 결합될 수 있다.
- [0134] ④에서, 프리로더 캐시(723)의 엔트리에 저장된 프리패치 메모리 주소가 변환된 주소 및 ③에서 결합된 주소가 비교될 수 있다. 다시 말해, 프리패치 메모리 주소와 요구 메모리 주소가 비교될 수 있다.
- [0135] ⑤에서, 비교 결과 및 선택된 엔트리의 상태에 기반하여, 상태 비트가 업데이트 될 수 있다. 그리고, 비교 결과 및 선택된 엔트리에 기반하여 요구 메모리 요청이 L1 캐시(730)에 전달될 수 있다.
- [0136] 비교 결과 주소가 동일하고 엔트리의 상태가 이미 프리패치 요청이 생성된 상태라면, 요구 메모리 요청은 삭제 되고 L1 캐시(730)로 전달되지 않을 수 있다. 다시 말해, 비교 결과 주소가 동일하고 엔트리의 상태가 이미 프리패치 요청이 생성된 상태는 프리로더 캐시(723)에 저장된 프리패치 메모리 주소가 정확히 예측되었음 의미할 수 있다.
- [0137] 따라서, 이 경우 ⑥에서, 프리로더 캐시(723)에서 프리패치 데이터를 LDST 유닛(710)으로 라이트 백 연산(writeback operation)을 통해 전달하면 되고, L1 캐시(730)로 요구 메모리 요청을 전달할 필요가 없을 수 있다.
- [0138] 다만, ⑤에서 비교 결과 주소가 다른 경우, 라이트 백 연산이 수행되지 않고 요구 메모리 요청이 L1 캐시(730)로 전달될 수 있다.
- [0139] 상술한 동작 이후, 엔트리의 상태는 업데이트될 수 있다. 엔트리의 상태가 업데이트되는 방법에 대해서는 도 8에서 설명하도록 하겠다.
- [0140] 도 7의 (C)는 프리패치 메모리 응답을 관리하는 방법을 설명하기 위한 도면이다.
- [0141] 프리패치 메모리 요청이 프리로더 캐시(723)의 하위 캐시로 전달되었고, 그에 대한 응답으로 하위 캐시로부터 프리패치 메모리 응답이 프리로더(720)에 전달될 수 있다. 프리패치 메모리 응답은 프리패치 데이터를 포함할 수 있다. 이때, 프리패치 데이터는 프리로더 캐시(723)에 저장되거나 LDST 유닛(710)으로 우회될 수 있다.
- [0142] 구체적으로 ①에서, 프리로더(720)는 프리패치 메모리 요청의 자세한 정보를 가지고 있는 엔트리를 선택하기 위해 워프 ID를 이용할 수 있다. 프리로더(720)는 워프 ID를 이용하여 해당 워프 ID로 인덱싱된 엔트리를 프리로더 캐시(723)에서 선택할 수 있다.
- [0143] ②에서, 엔트리의 상태가 요구 메모리 요청이 이미 생성된 상태이고 엔트리에 저장된 프리패치 메모리 주소 및 요구 메모리 요청에 포함된 요구 메모리 주소가 일치하는지 여부를 판단할 수 있다.
- [0144] ③에서, 엔트리의 상태가 요구 메모리 요청이 이미 생성된 상태이고 엔트리에 저장된 프리패치 메모리 주소 및 요구 메모리 요청에 포함된 요구 메모리 주소가 일치하는 경우, 프리패치 메모리 응답은 디멀티플렉서(demultiplexer)를 이용하여 LDST 유닛으로 라이트 백 될 수 있다. 즉, 이미 요구 메모리 요청이 생성된 경우, 프리패치 데이터는 프리로더 캐시(723)에 저장되지 않고 LDST 유닛으로 우회할 수 있다.
- [0145] ④에서, 엔트리의 상태가 요구 메모리 요청이 이미 생성된 상태가 아니거나 엔트리에 저장된 프리패치 메모리 주소 및 요구 메모리 요청에 포함된 요구 메모리 주소가 일치하지 않는 경우, 프리패치 메모리 응답에 포함된 프리패치 데이터는 프리로더 캐시(723)에 저장될 수 있다. 이때, 저장된 프리패치 데이터는 추후 LDST 유닛에 의해 요구 메모리 요청이 생성되었을 때 도 7의 (B)에서 설명한 방법에 따라 LDST 유닛으로 전송될 수 있다.
- [0146] 이하에서는, 엔트리의 상태가 프리로더(720)의 상태에 따라 업데이트되는 방법에 대해서 설명하도록 하겠다.
- [0148] 도 8은 본 개시의 일실시예에 따른 엔트리의 상태 업데이트를 설명하기 위한 도면이다.
- [0149] 프리로더 캐시에 포함되는 엔트리들 각각은 개별적으로 5개의 상태를 가질 수 있다. 엔트리의 상태는 상태 비트로 표현될 수 있다. 상태 비트는 복수의 비트들을 포함할 수 있다. 상태 비트 중에서 하위 2 비트가 엔트리의 상태를 나타낼 수 있다. 상태 비트 중에서 하위 2 비트를 제외한 상위 비트는 복수의 서브 프리패치 메모리 요청들을 관리하는데 이용될 수 있다. 복수의 비트들 중에서 하위 2비트를 제외한 상위 비트의 개수는 복수의 서브 프리패치 메모리 요청들의 개수에 기반할 수 있다. 다시 말해, 하위 2비트를 제외한 상위 비트가 N개인

경우, 2^N 개의 서브 프리패치 메모리 요청들이 관리될 수 있다. 하위 2비트를 제외한 상위 비트는 복수의 서브 프리패치 메모리 요청에 대응하는 복수의 서브 프리패치 메모리 응답이 프리로더에 도착한 개수에 따라 업데이트될 수 있다.

- [0150] 이하에서는, 설명의 편의를 위해 상태 비트가 4비트인 경우를 가정하여 설명하겠다. 이때, 하위 2 비트는 엔트리의 상태를 나타낼 수 있고, 상위 2비트는 2^2 개의 서브 프리패치 메모리 요청들을 관리하는데 이용될 수 있다. 다만, 이하의 설명은 상태 비트가 4비트인 경우에 한정되지 않음은 통상의 기술자에게 자명하다.
- [0151] 엔트리의 상태는 항상 Invalid 상태에서 시작할 수 있고, Invalid 상태에서 상태 비트는 [0 0 0 0]으로 표현될 수 있다.
- [0152] ①에서, 프리패치 메모리 요청이 생성되고 106 비트의 주소가 엔트리에 저장되는 경우, 엔트리의 상태는 Invalid 상태에서 Pref. Gen. 상태로 업데이트될 수 있다. Pref. Gen. 상태는 [0 0 0 1]로 표현될 수 있다.
- [0153] Pref. Gen. 상태는 프리패치 메모리 요청이 생성되었음을 나타내는 상태일 수 있다.
- [0154] 엔트리의 상태는 Pref. Gen. 상태에서 3 가지의 상태 중 어느 하나로 업데이트될 수 있다.
- [0155] ②에서, 요구 메모리 요청이 생성되었고, 엔트리에 저장된 프리패치 데이터 주소와 요구 메모리 요청에 포함된 요구 메모리 주소가 일치하는 경우, 엔트리의 상태는 Pref. Gen. 상태에서 Dem. Gen. 으로 업데이트될 수 있다. Dem. Gen. 상태는 [# # 1 1]로 표현될 수 있다.
- [0156] 여기서, 상위 2 비트는 서브 프리패치 메모리 요청에 대응하는 서브 프리패치 메모리 응답의 개수에 따라 업데이트될 수 있다. 예를 들어, 4개의 요구 서브 메모리 요청에 대응하여 2개의 요구 서브 메모리 응답만이 프리로더에 도착한 경우, 상위 2비트는 [1 0]으로 표현될 수 있다.
- [0157] ③에서, 요구 메모리 요청이 생성되었고, 엔트리에 저장된 프리패치 데이터 주소와 요구 메모리 요청에 포함된 요구 메모리 주소가 일치하지 않는 경우, 엔트리의 상태는 Pref. Gen. 상태에서 Miss Match 상태로 업데이트될 수 있다. Miss Match 상태는 [# # 1 0]로 표현될 수 있다.
- [0158] ④에서, 요구 메모리 요청이 생성되기 전에 프리패치 메모리 요청에 따른 프리패치 메모리 응답이 도착한 경우, 엔트리의 상태는 Pref. Gen. 상태에서 Pre. Data Ack. 상태로 업데이트될 수 있다. Pre. Data Ack. 상태는 [# # 0 1]로 표현될 수 있다.
- [0159] 이때, Dem. Gen. 상태, Miss Match 상태 및 Pre. Data Ack. 상태의 상위 2 비트는 복수의 서브 프리패치 메모리 요청에 대한 응답으로 프리로더에 도착한 서브 프리패치 메모리 응답의 수를 나타낼 수 있다.
- [0160] Miss Match 상태는 프리패치 메모리 주소와 요구 메모리 주소가 일치하지 않음을 나타내는 상태일 수 있다.
- [0161] 따라서, ⑤에서, 서브 프리패치 메모리 요청에 따른 서브 프리패치 메모리 응답이 전부 도착하면, 엔트리의 상태는 다시 Miss Match 상태에서 Invalid 상태로 업데이트 될 수 있다. 또한, Miss Match 상태에서 요구 메모리 요청은 데이터를 로딩하기 위해 하위 캐시로 전달될 수 있다.
- [0162] Dem. Gen. 상태는 프리패치 메모리 주소와 요구 메모리 주소가 일치하는 상태를 나타낼 수 있다. 이때, 요구 메모리 주요청은 L1 캐시에 전달되지 않도록 제거될 수 있다.
- [0163] ⑥에서, 서브 프리패치 메모리 요청에 따른 서브 프리패치 메모리 응답이 전부 프리로더에 도착하면, 엔트리의 상태는 Dem. Gen. 상태에서 Invalid 상태로 업데이트될 수 있다.
- [0164] Pre. Data Ack. 상태는 요구 메모리 요청이 생성되기 전에 서브 프리패치 메모리 요청에 따른 서브 프리패치 메모리 응답이 프리로더에 도착한 상태를 나타낼 수 있다. 따라서, 프리로더로 로딩된 데이터는 프리로더 캐시(즉, 엔트리)에 저장될 수 있다.
- [0165] ⑦에서, 요구 메모리 요청이 생성되고, 요구 메모리 주소와 프리로더 캐시에 저장된 프리패치 메모리 주소가 일치하는 경우, 프리로더 캐시에 저장된 프리패치 데이터는 LDST 유닛으로 전달될 수 있다. 그리고 나서, 엔트리의 상태는, Pre. Data Ack. 상태에서 Dem. Gen. 상태로 업데이트될 수 있다.
- [0166] 만약 ⑧에서, 요구 메모리 주소와 프리로더 캐시에 저장된 프리패치 메모리 주소가 일치하지 않는 경우, 엔트리의 상태는 Pre. Data Ack. 상태에서 Miss Match 상태로 업데이트될 수 있다. 그리고, 요구 메모리 요청은 하위 캐시로 전파될 수 있다.

- [0168] 도 9는 본 개시의 일실시예에 따른 전자 장치의 동작을 설명하기 위한 플로우차트이다.
- [0169] 이하 실시예에서 각 동작들은 순차적으로 수행될 수도 있으나, 반드시 순차적으로 수행되는 것은 아니다. 예를 들어, 각 동작들의 순서가 변경될 수도 있으며, 적어도 두 동작들이 병렬적으로 수행될 수도 있다. 단계(901) 내지 단계(907)는 프로세서에 의해 수행될 수 있으나 실시예가 이에 한정되지 않으며 프리로더에 의해 수행될 수도 있다.
- [0170] 단계(901)에서, 프로세서는 샘플링된 이전 요구 메모리 주소들의 보폭에 기반하여 프리패치 메모리 요청을 생성할 수 있다.
- [0171] 단계(903)에서, 프로세서는 프리패치 메모리 요청에 대한 응답인 프리패치 메모리 응답에 포함되는 프리패치 데이터 및 프리패치 데이터에 대응하는 프리패치 메모리 주소를 프리로더 캐시에 저장할 수 있다.
- [0172] 단계(905)에서, 프로세서는 LDST 유닛으로부터 생성된 다음 연산을 위한 요구 메모리 요청에 포함된 요구 메모리 주소 및 프리로더 캐시에 저장된 프리패치 메모리 주소가 일치하는지 여부를 판단할 수 있다.
- [0173] 단계(907)에서, 프로세서는 요구 메모리 주소와 프리패치 메모리 주소가 일치하는 경우, 프리패치 데이터를 LDST 유닛으로 전송하고, 요구 메모리 주소와 프리패치 메모리 주소가 일치하지 않는 경우, 요구 메모리 요청을 프리로더 캐시의 하위 캐시로 전달할 수 있다.
- [0175] 한편, 본 발명에 따른 방법은 컴퓨터에서 실행될 수 있는 프로그램으로 작성되어 마그네틱 저장매체, 광학적 판독매체, 디지털 저장매체 등 다양한 기록 매체로도 구현될 수 있다.
- [0176] 본 명세서에 설명된 각종 기술들의 구현들은 디지털 전자 회로조직으로, 또는 컴퓨터 하드웨어, 펌웨어, 소프트웨어로, 또는 그들의 조합들로 구현될 수 있다. 구현들은 데이터 처리 장치, 예를 들어 프로그램가능 프로세서, 컴퓨터, 또는 다수의 컴퓨터들의 동작에 의한 처리를 위해, 또는 이 동작을 제어하기 위해, 컴퓨터 프로그램 제품, 즉 정보 캐리어, 예를 들어 기계 판독가능 저장 장치(컴퓨터 판독가능 매체) 또는 전파 신호에서 유형적으로 구체화된 컴퓨터 프로그램으로서 구현될 수 있다. 상술한 컴퓨터 프로그램(들)과 같은 컴퓨터 프로그램은 컴파일된 또는 인터프리트된 언어들을 포함하는 임의의 형태의 프로그래밍 언어로 기록될 수 있고, 독립형 프로그램으로서 또는 모듈, 구성요소, 서브루틴, 또는 컴퓨팅 환경에서의 사용에 적절한 다른 유닛으로서 포함하는 임의의 형태로 전개될 수 있다. 컴퓨터 프로그램은 하나의 사이트에서 하나의 컴퓨터 또는 다수의 컴퓨터들 상에서 처리되도록 또는 다수의 사이트들에 걸쳐 분배되고 통신 네트워크에 의해 상호 연결되도록 전개될 수 있다.
- [0177] 컴퓨터 프로그램의 처리에 적절한 프로세서들은 예로서, 범용 및 특수 목적 마이크로프로세서들 둘 다, 및 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서들을 포함한다. 일반적으로, 프로세서는 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 다로부터 명령어들 및 데이터를 수신할 것이다. 컴퓨터의 요소들은 명령어들을 실행하는 적어도 하나의 프로세서 및 명령어들 및 데이터를 저장하는 하나 이상의 메모리 장치들을 포함할 수 있다. 일반적으로, 컴퓨터는 데이터를 저장하는 하나 이상의 대량 저장 장치들, 예를 들어 자기, 자기-광 디스크들, 또는 광 디스크들을 포함할 수 있거나, 이것들로부터 데이터를 수신하거나 이것들에 데이터를 송신하거나 또는 양쪽으로 되도록 결합될 수도 있다. 컴퓨터 프로그램 명령어들 및 데이터를 구체화하는데 적절한 정보 캐리어들은 예로서 반도체 메모리 장치들, 예를 들어, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM(Compact Disk Read Only Memory), DVD(Digital Video Disk)와 같은 광 기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-Optical Media), 롬(ROM, Read Only Memory), 램(RAM, Random Access Memory), 플래시 메모리, EPROM(Erasable Programmable ROM), EEPROM(Electrically Erasable Programmable ROM) 등을 포함한다. 프로세서 및 메모리는 특수 목적 논리 회로조직에 의해 보충되거나, 이에 포함될 수 있다.
- [0178] 또한, 컴퓨터 판독가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용매체일 수 있고, 컴퓨터 저장매체 및 전송매체를 모두 포함할 수 있다.
- [0179] 본 명세서는 다수의 특정한 구현물의 세부사항들을 포함하지만, 이들은 어떠한 발명이나 청구 가능한 것의 범위에 대해서도 제한적인 것으로서 이해되어서는 안되며, 오히려 특정한 발명의 특정한 실시형태에 특유할 수 있는 특징들에 대한 설명으로서 이해되어야 한다. 개별적인 실시형태의 문맥에서 본 명세서에 기술된 특정한 특징들은 단일 실시형태에서 조합하여 구현될 수도 있다. 반대로, 단일 실시형태의 문맥에서 기술한 다양한 특징들 역

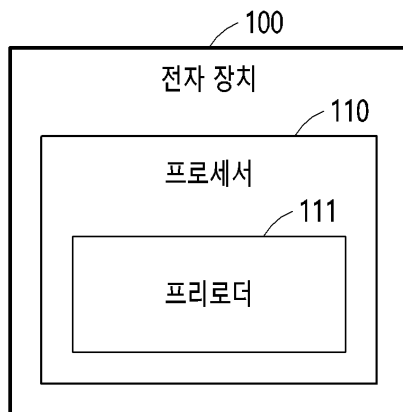
시 개별적으로 혹은 어떠한 적절한 하위 조합으로도 복수의 실시형태에서 구현 가능하다. 나아가, 특징들이 특정한 조합으로 동작하고 초기에 그와 같이 청구된 바와 같이 묘사될 수 있지만, 청구된 조합으로부터의 하나 이상의 특징들은 일부 경우에 그 조합으로부터 배제될 수 있으며, 그 청구된 조합은 하위 조합이나 하위 조합의 변형물로 변경될 수 있다.

[0180] 마찬가지로, 특정한 순서로 도면에서 동작들을 묘사하고 있지만, 이는 바람직한 결과를 얻기 위하여 도시된 그 특정한 순서나 순차적인 순서대로 그러한 동작들을 수행하여야 한다거나 모든 도시된 동작들이 수행되어야 하는 것으로 이해되어서는 안 된다. 특정한 경우, 멀티태스킹과 병렬 프로세싱이 유리할 수 있다. 또한, 상술한 실시형태의 다양한 장치 컴포넌트의 분리는 그러한 분리를 모든 실시형태에서 요구하는 것으로 이해되어서는 안되며, 설명한 프로그램 컴포넌트와 장치들은 일반적으로 단일의 소프트웨어 제품으로 함께 통합되거나 다중 소프트웨어 제품에 패키징 될 수 있다는 점을 이해하여야 한다.

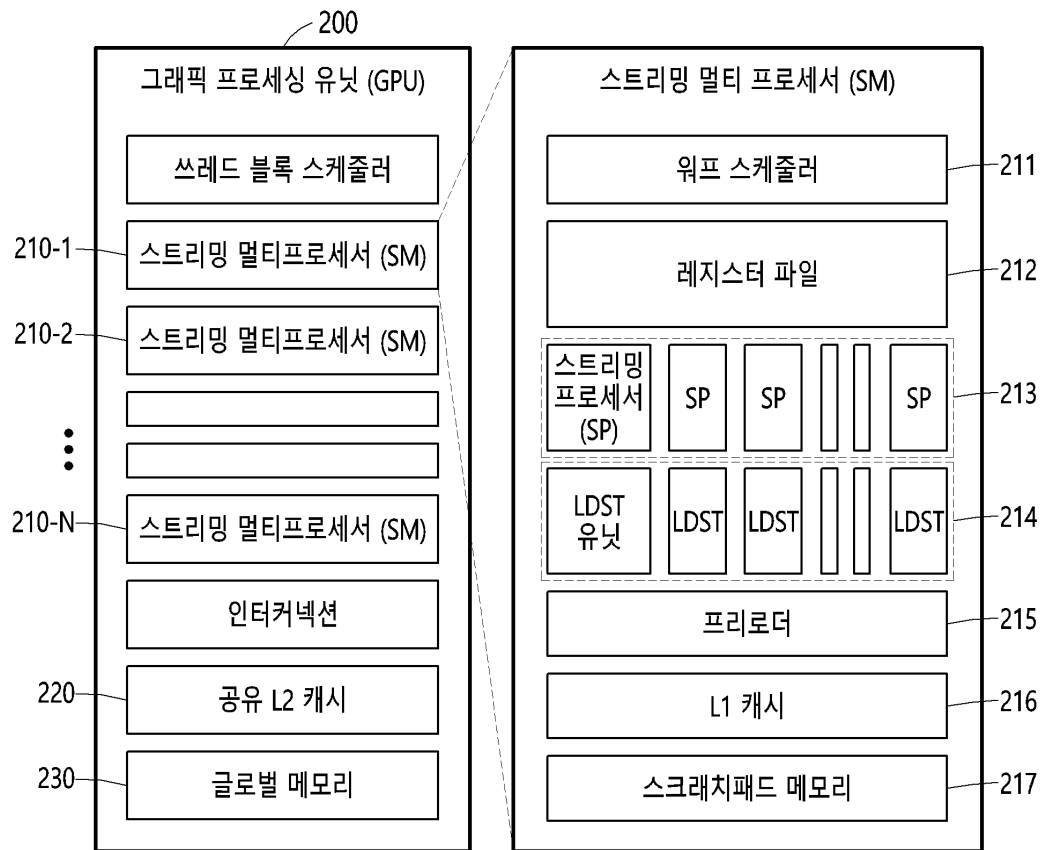
[0181] 한편, 본 명세서와 도면에 개시된 본 발명의 실시 예들은 이해를 돕기 위해 특정 예를 제시한 것에 지나지 않으며, 본 발명의 범위를 한정하고자 하는 것은 아니다. 여기에 개시된 실시 예들 이외에도 본 발명의 기술적 사상에 바탕을 둔 다른 변형 예들이 실시 가능하다는 것은, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 자명한 것이다.

도면

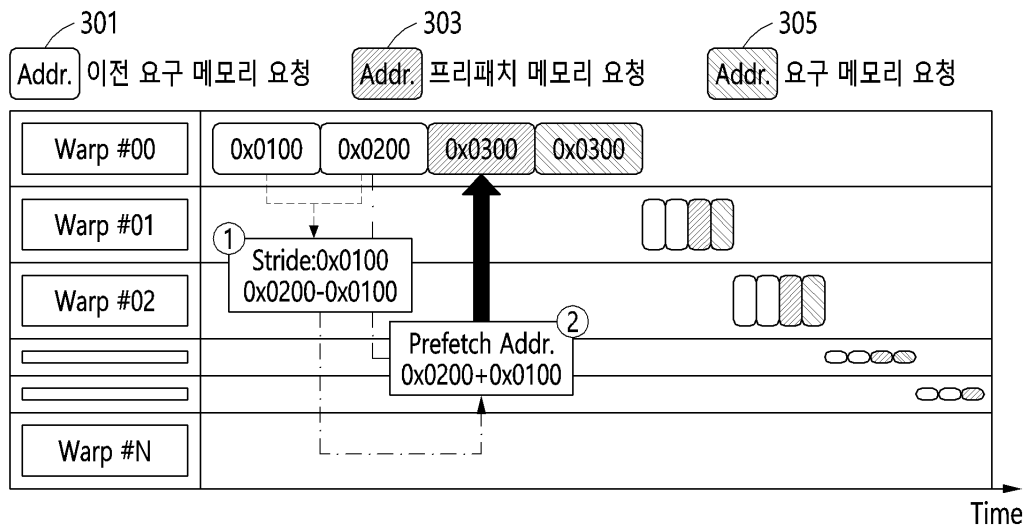
도면1



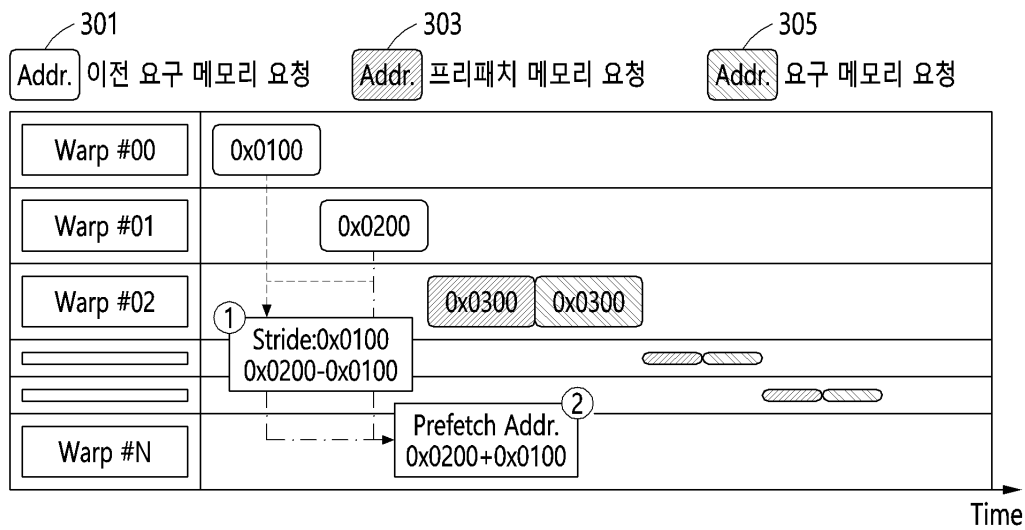
도면2



도면3

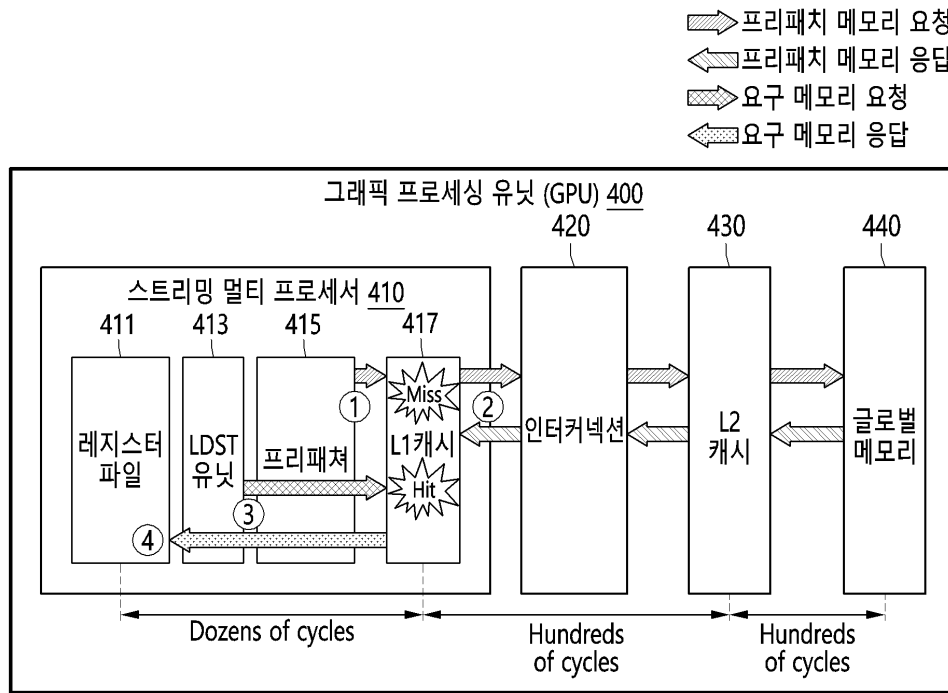


(A)

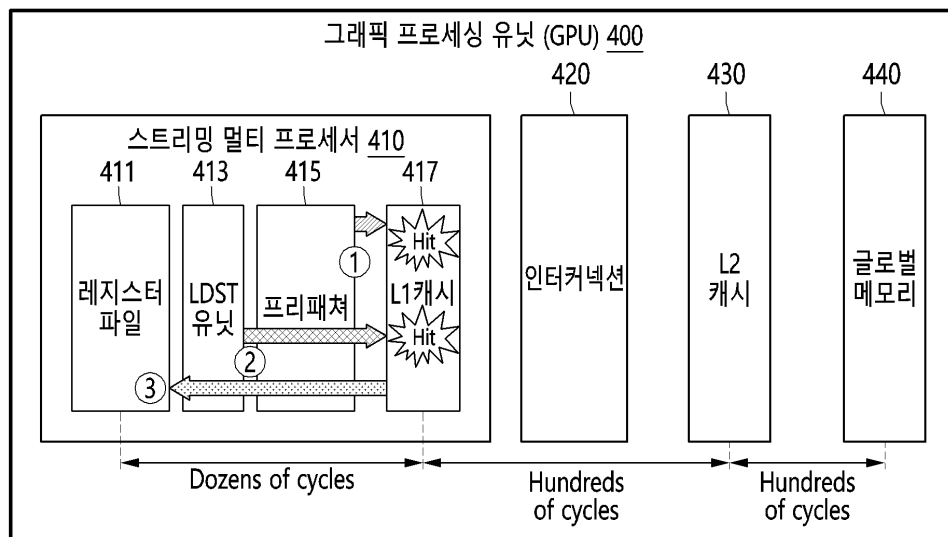


(B)

도면4a

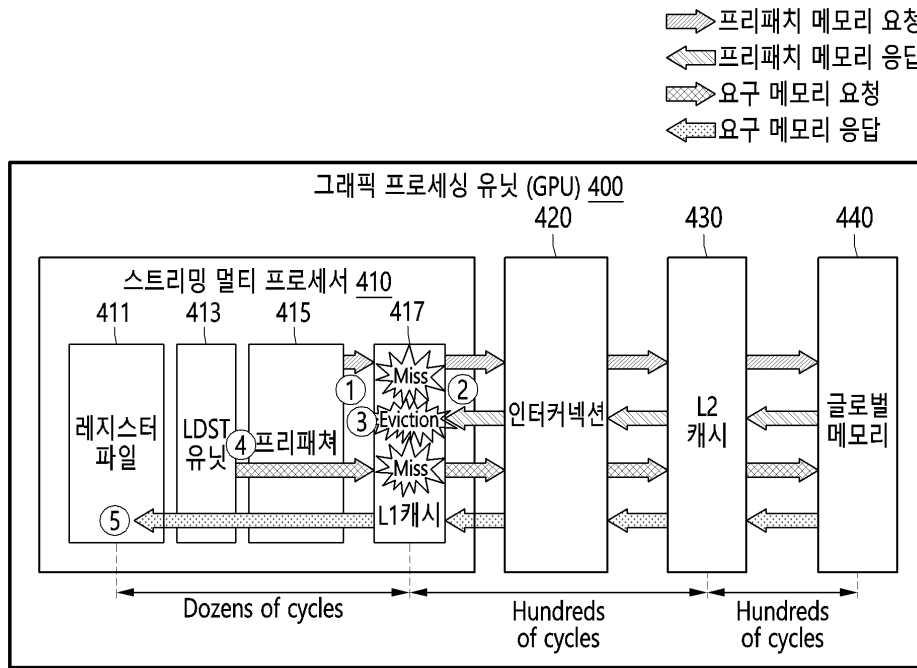


(A)

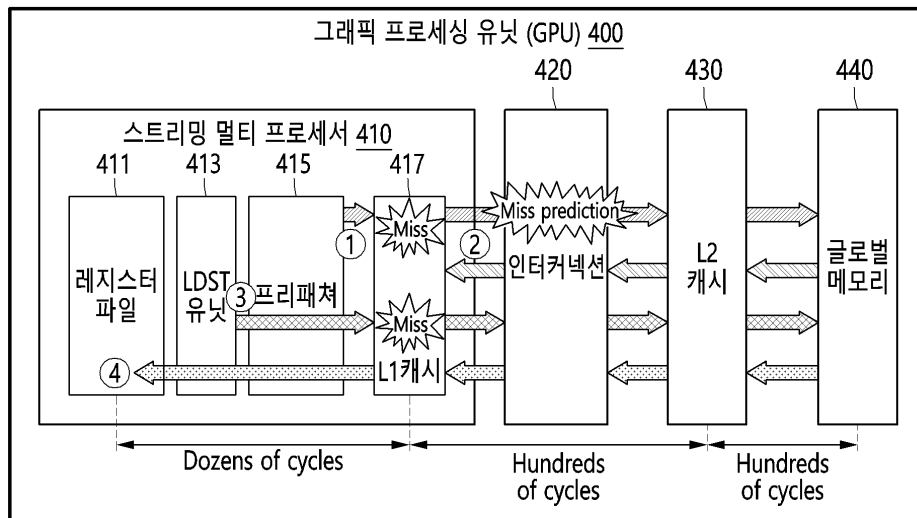


(B)

도면4b

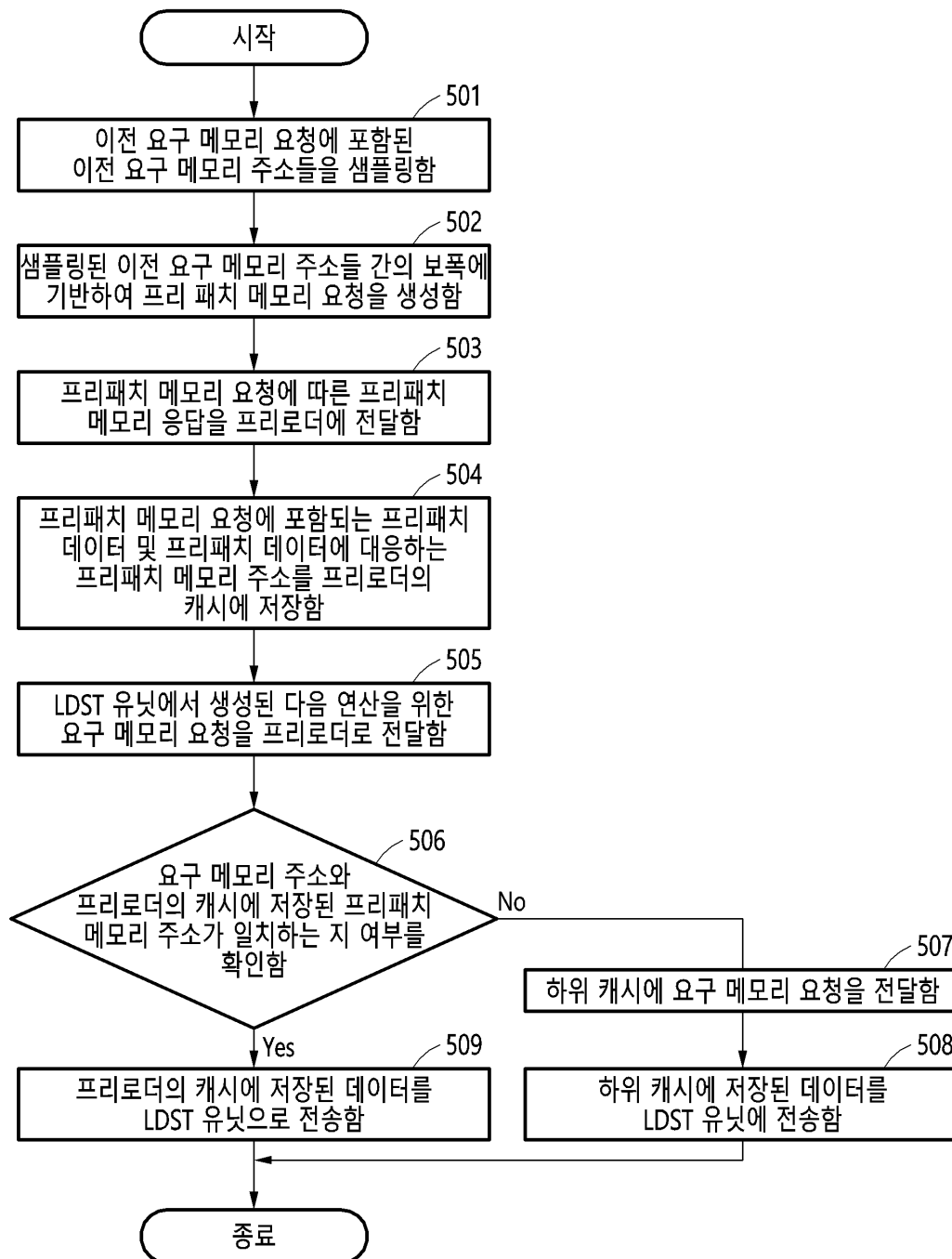


(C)

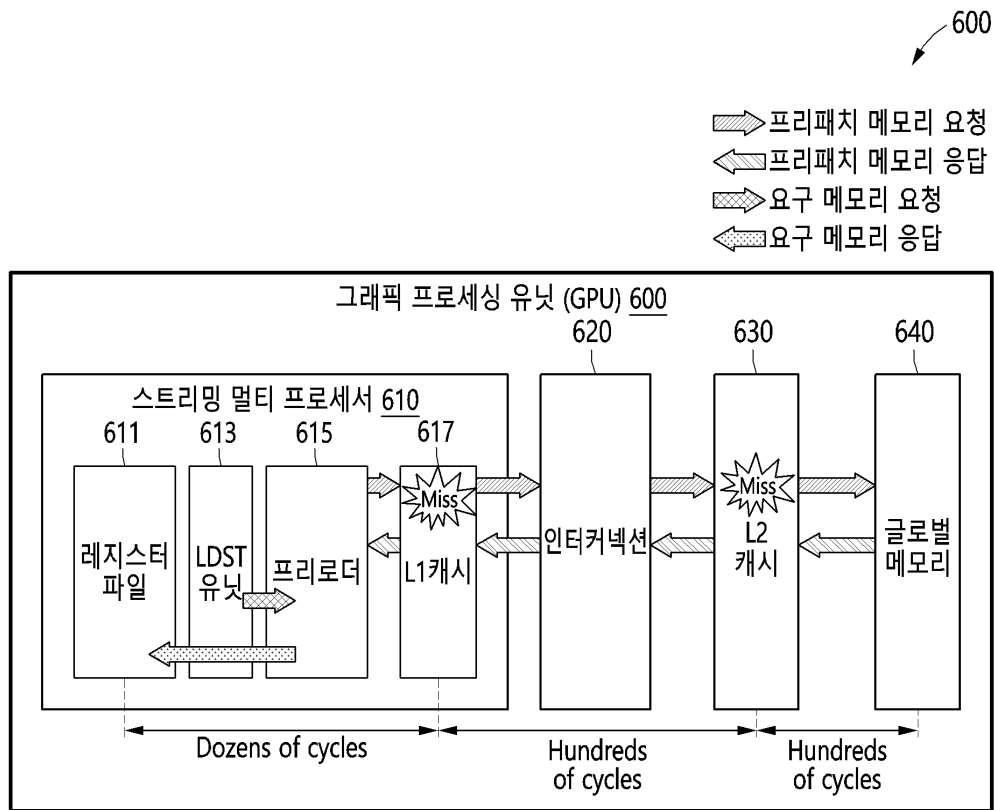


(D)

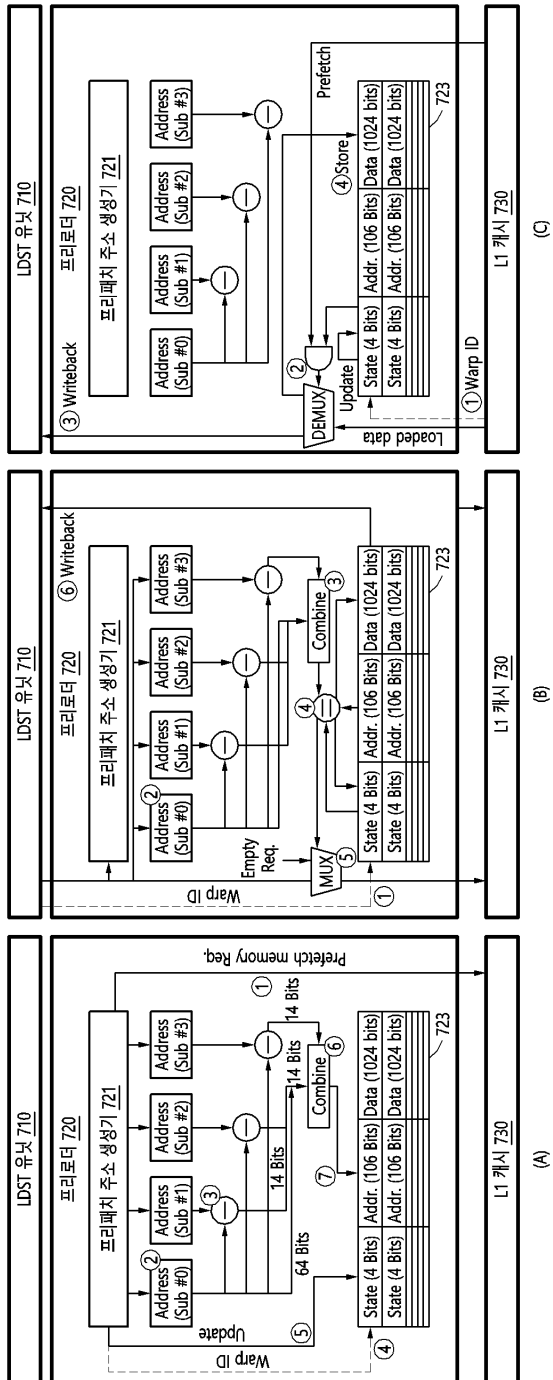
도면5



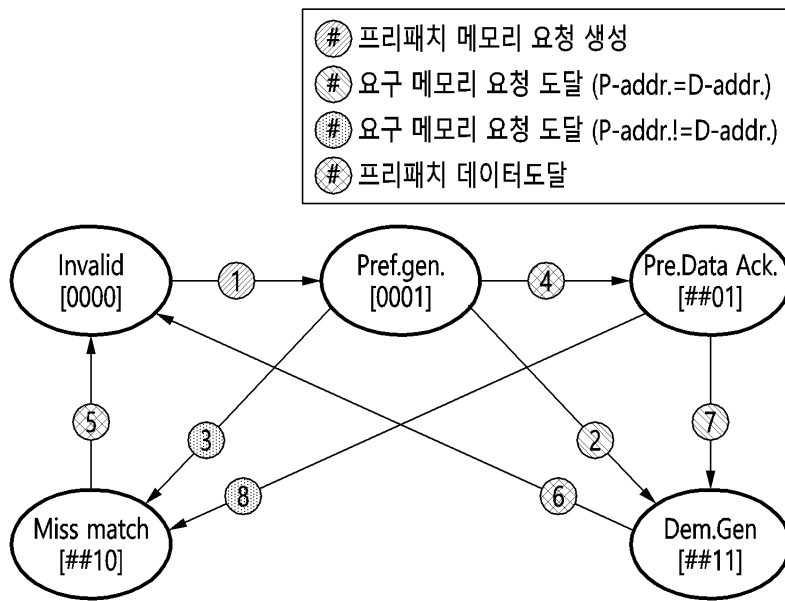
도면6



도면7



도면8



도면9

