분리형 로우 버퍼를 활용한 메모리 읽기 및 쓰기와 병렬 처리 가능한 DRAM 리프레시 기법

이민규, 정기석*
*한양대학교

hanlovelan@hanyang.ac.kr, *kchung@hanyang.ac.kr

Parallelizing Refresh with Read/Write Operations using Split Row Buffer

Lee, Min-Kyu and Chung, Ki-Seok*
Hanyang University, Seoul, Korea

요 약

현대의 컴퓨터는 Dynamic Random Access Memory (DRAM)을 주기억장치로 주로 사용한다. DRAM 은 하나의 캐패시터와 하나의 트렌지스터로 구성된 메모리 셀에 전하를 저장함으로써 기억장치로 동작한다. 메모리 셀에 저장된 전하는 시간에 따라 누설되어 데이터는 손실이 유발될 수 있다. 따라서, DRAM 은 이러한 데이터 손실을 막기 위해 주기적으로 메모리 셀의 전하를 다시 충전하는 리프레시 (Refresh)를 한다. DRAM 에서 리프레시를 하는 동안에는 DRAM 의 데이터를 사용하지 못하기 때문에 리프레시로 인한 시스템 성능 저하가 발생한다. 따라서, 본 논문에서는 이러한 리프레시에 의한 시스템 성능 저하를 줄이기 위해 로우 액세스 버퍼와 컬럼 액세스 버퍼로 분리된 로우 버퍼 DRAM (Split Row Buffer DRAM)를 활용한다. 분리형 로우 버퍼 DRAM 에서는 액티베이트 (Activate)와 프리차지 (Precharge)와 같은 로우 레벨 동작과 읽기나 쓰기 같은 컬럼 레벨 동작을 동시에 수행 가능한 점을 활용하여, 본 논문에서는 로우 액세스 버퍼에서 리프레시 수행 중에 컬럼 액세스 버퍼에서 읽기와 쓰기가 가능한 리프레시 기법을 제한함으로써 리프레시로 인한 시스템 성능 저하를 줄인다. 실험 결과, 제안한 리프레시 기법은 평균적으로는 7.2% 시스템 성능 향상을 보였다.

Ⅰ. 서 론

Dynamic Random-Access Memory (DRAM)은 주로 시스템의 주기억장치로 사용된다. 이러한 DRAM 은 하나의 캐패시터와 트랜지스터로 구성된 메모리 셀에 데이터를 저장한다. DRAM 메모리 셀에서는 지속적으로 누설 전류가 발생하기 때문에 메모리 셀에 저장된 데이터는 손실 위험이 있다. 이러한 데이터 손실 위험을 막기 위해 DRAM 은 메모리 셀의 전하를 다시 충전하는 리프레시 (Refresh)를 한다. DRAM 리프레시는 DRAM 의 메모리 셀의 값을 읽고 다시 쓰는 동작으로 DRAM 리프레시가 이루어지는 동안에는 DRAM 에 저장된 데이터를 읽거나 쓰기를 못한다. 따라서 DRAM 리프레시는 메모리 명령의 대기 시간을 증가시켜 시스템 성능을 저하 시킨다.

DRAM 은 데이터를 보존하기 위해서 주기적으로 리프레시가 필요하다. DRAM 의 리프레시 시간은 리프레시가 되는 메모리 셀의 수에 따라 비례한다. DRAM 의 모든 메모리 셀을 한번에 리프레시 하기 위해서는 긴 시간이 필요하다. 이러한 긴 리프레시 시간으로 인한 시스템 성능 저하는 심각하기 때문에 일반적으로 DRAM 의 모든 메모리 셀은 한번에 리프레시가 되지 않고 8K 리프레시 명령에 의해 부분적으로 리프레시가 된다[1]. 하지만 DRAM 의 용량이 증가하면서 리프레시 되는 메모리 셀의 수도 증가한다. 따라서 DRAM 의 용량이 증가하면서 리프레시 시간도 점차 증가하고 있다. 이렇게 증가된 리프레시 시간으로 인한 시스템 성능 저하는 점차 심화된다.

분리형 로우 버퍼 DRAM (Split Row Buffer DRAM)은 서로 다른 로우 주소를 갖는 메모리 명령들이 연속적으로 접속 할 때 발생하는 로우 버퍼 미스 패널티 (Row Buffer Miss Penalty)를 줄이기 위해 제안된 구조이다[2]. 분리형 로우 버퍼 DRAM 은 액티베이트 (Activate)와 프리차지 (Precharge)와 같은 로우 레벨 액세스 명령과 읽기 및 쓰기와 같은 컬럼 레벨 액세스 명령이 병렬로 수행 가능한 구조이다. 리프레시는 메모리 셀의 전하를 재충전하는 동작으로 액티베이트와 프리차지가 연속적으로 동작하는 것과 같다. 따라서 본 논문에서는 이러한 특징을 활용하여 리프레시와 읽기나 쓰기를 병렬로 수행함으로써 리프레시로 인한 시스템 성능 저하를 줄이는 리프레시 기법을 제안한다.

Ⅱ. 본론

분리형 로우 버퍼 DRAM 에서는 리프레시와 읽기 및 쓰기 가 동시에 수행하는 것이 가능하다. 하지만 리프레시 중에 새로운 로우의 액티베이트가 불가능하기 때문에 현재 컬럼 액세스 버퍼에 있는 로우 데이터와 동일한 로우 주소를 갖는 메모리 명령을 제외하고는

실행이 불가능하다. 따라서 컬럼 액세스 버퍼와 동일한 로우 주소를 갖는 메모리 명령이 모두 실행된 이후 다른 메모리 명령이 실행 되기 위해서는 로우 액세스 버퍼에서 리프레시가 완료 될 때까지 대기해야 된다. 리프레시로 인한 메모리 명령 대기 시간을 줄이기 위해서는 리프레시 시간을 줄여야 한다. JEDEC 에서는 리프레시로 인한 성능 저하를 줄이기 위해 리프레시 단위를 줄이는 대신 리프레시 실행 빈도를 늘려 한 번 실행되는 리프레시 시간을 줄이는 Fine-Granularity Refresh (FGR)을 DDR4 DRAM 에 추가하였다[3].

본 논문에서 제안하고 있는 리프레시 기법에서는 DDR4 DRAM 에서 사용하고 있는 FGR 기법을 분리형 로우 버퍼 구조에 적용하였다. 즉 FGR 기법을 통해 로우 액세스 버퍼에서 리프레시 시간을 줄이고 컬럼 액세스 버퍼에서 더 이상 실행 가능한 메모리 명령이 없어 컬럼 액세스 버퍼가 스톨 (Stall)되는 시간을 줄인다. 본 논문에서 제안하고 있는 리프레시 기법은 한 번 수행하는 리프레시 시간을 줄이고, 반면 리프레시 빈도를 늘려, 전체 리프레시 동안에 실행 가능한 메모리 명령을 늘린다. 본 리프레시 기법은 리프레시와 읽기 및 쓰기를 효과적으로 병렬적으로 실행할 수 있게 함으로써 리프레시로 인한 성능 저하를 줄인다.

본 논문에서 제안한 리프레시 기법의 성능을 평가하기 위해 이벤트 기반 컴퓨터 시뮬레이터인 gem5[4]와 DRAM 시뮬레이터인 DRAMSim2[5]를 통합하여 사용했다. 실험에서 사용한 DRAM AC 파라미터는 [6]에서 주어진 값을 사용했으며, 리프레시 시간은 32Gb DRAM 을 기준으로 980ns 이다. 표 1 은 실험에서 사용한 다양한 파라미터 값을 나타낸다. 이 실험은 Alpha ISA 로 컴파일된 SPEC CPU 2006 벤치마크[7]를 사용했다.

표 1. 시뮬레이터 파라미터

파라미터	값
Processor	1 core, 2Ghz, Out-of-Order
L1 Caches (Per core)	64KB instruction cache, 64KB data cache, LRU
L2 Cache (shared)	1MB Cache, 8-way, LRU
DRAM	1 채널, 1 랭크, 8 뱅크, DDR3-1333[4]

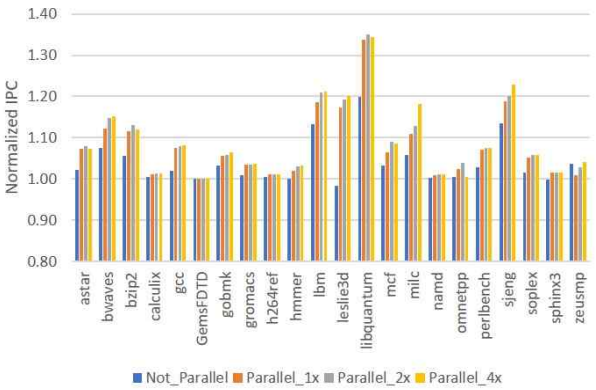

그림 1. 제안한 리프레시 기법 성능 비교

그림 1 은 본 논문에서 제안하고 있는 리프레시 기법의 성능을 일반적인 DRAM 을 갖는 시스템과 정규화한 IPC (Normalized Instruction Per Cycle)를 비교한 그래프이다. Not_Parallel 은 분리형 로우 버퍼 DRAM 에서 리프레시와 읽기 및 쓰기가 병렬로 동작하지 않으며, Parallel 1x, 2x, 와 4x 는 분리형 로우 버퍼 DRAM 에서 리프레시와 읽기 및 쓰기가 병렬로 동작하며 각각의 리프레시 주기는 FGR 처럼 1 배, 2 배, 와 4 배이다. 제안하고 있는 Not_Parallel / Prallel_1x / Parallel_2x / Parallel_4x 은 일반적인 DRAM 의 시스템 성능에 비해 각각 3.85%, 6.12%, 6.62%, 7.18% 성능 향상이 있다. 리프레시 빈도가 증가할수록 개별의 리프레시 시간은 감소하기 때문에 분리형 로우 버퍼 DRAM 의 컬럼 액세스 버퍼로 실행가능한 메모리 명령이 없이 스톨 되는 시간이 줄어들고, 전체 리프레시 시간 동안에 실행 가능한 메모리 명령이 늘어나 리프레시로 인한 시스템 저하가 줄어든다. 따라서 본 논문에서 제안하고 있는 리프레시 기법은 리프레시와 읽기 및 쓰기가 효과적으로 병렬로 동작해 높은 성능 향상을 보였다.

Ⅲ. 결론

DRAM 의 용량 증가에 따라 리프레시 시간은 비례해서 증가하고, 이에 따른 시스템 성능 저하 역시 증가한다. 본 논문에서는 이 성능 저하를 최소화하기 위해 분리형 로우 버퍼 DRAM 을 사용한 리프레시 기법을 제안했다. 분리형 로우 버퍼 DRAM 은 리프레시와 읽기 및 쓰기가 병렬로 처리 가능하다. 하지만 분리형 로우 버퍼 DRAM 구조는 리프레시 도중에 새로운 로우를 액티베이트 하지 못하기 때문에 실행 가능한 메모리 접근 명령은 제한적이다. 이러한 문제를 해결하기 위해 본 논문에서 제안하는 리프레시 기법은 분리형 로우 버퍼 DRAM 에 DDR4 DRAM 에서 채택된 FGR 를 적용하여 리프레시 시간을 줄이고 동작 빈도는 늘렸다. 본 논문에서 제안한 리프레시 기법은 분리형 로우 버퍼 DRAM 에서 리프레시와 읽기 및 쓰기를 효과적으로 병렬로 동작해, 시스템 성능을 평균 7.2% 향상시켰다.

ACKNOWLEDGMENT

이 논문은 2018 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.R7119-16-1009, 하베스트 에너지 기반 IoT 디바이스를 위 한 지능형 반도체 핵심기술 개발)

참 고 문 헌

[1] JESD79-3: 'DDR3 SDRAM STANDARD', 2012

[2] Lee, M. K., Chung, K. S.: 'High performance DRAM architecture with split row buffer', Electron. Lett., 2016, 52, (22), pp. 1844-1845

[3] JESD79-4B: 'DDR4 SDRAM STANDARD', 2017

[4] Binkert, N., Beckmann, B., Black, G., et al.: 'The Gem5 simulator', ACM SIGARCH Comput. Archit. News, 2011, 39, (2), pp. 1-7

[5] Rosenfeld, P., Cooper-Balis, E., Jacob, B.: 'DRAMSim2: A cycle accurate memory system simulator', IEEE Comput. Archit. Lett., 2011, 10, (1), pp. 16-19

[6] '4Gb: x4, x8, x16 DDR3 SDRAM', www.micron.com/products/datasheets, accessed 16 July 2017

[7] 'SPEC CPU2006', http://www.spec.org/cpu2006, accessed 16 July 2017