JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 24 No. 11, pp. 1-9, November 2019
https://doi.org/10.9708/jksci.2019.24.11.001

Small Active Command Design for High Density DRAMs

Kwangho Lee*, Jongmin Lee*

*Student, Dept. of Computer Engineering, Won-Kwang University, Iksan, Korea
*Professor, Dept. of Computer Engineering, Won-Kwang University, Iksan, Korea

[Abstract]

In this paper, we propose a Small Active Command scheme which reduces the power consumption of the command bus to DRAM. To do this, we target the ACTIVE command, which consists of multiple packets, containing the row address that occupies the largest size among the addresses delivered to the DRAM. The proposed scheme identifies frequently referenced row addresses as Hot pages first, and delivers index numbers of small caches (tables) located in the memory controller and DRAM. I-ACTIVE and I-PRECHARGE commands using unused bits of existing DRAM commands are added for index number transfer and cache synchronization management. Experimental results show that the proposed method reduces the command bus power consumption by 20% and 8.1% on average in the close-page and open-page policies, respectively.

▸ **Key words**: DRAMs, Low-power, Index-based Activation, Memories, Simulation

[요 약]

본 논문에서는 DRAM으로 전송되는 커맨드 버스의 전력 소모량을 감소시킬 수 있는 Small Active Command 기법을 제안한다. 이를 위해, DRAM으로 전달되는 주소 중 가장 큰 크기를 차지하는 Row 주소를 포함하고 다중패킷으로 구성된 ACTIVE 커맨드를 대상으로 한다. 제안된 Small Active Command 기법은 자주 참조되는 Row 주소를 Hot 페이지로 식별하고 메모리 컨트롤러와 DRAM에 적재된 작은 캐시(테이블)의 인덱스 번호를 Row 주소를 대신하여 단일 패킷으로 전달한다. 제안된 기법에서는 인덱스 번호 전달과 캐시 동기화 관리를 위해 기존 DRAM커맨드의 사용하지 않는 비트를 활용한 I-ACTIVE와 I-PRECHARGE 커맨드를 추가하였다. 시뮬레이션을 이용한 실험 결과 제안된 방식은 Close-page 정책과 Open-page 정책에서 각각 평균적으로 20%, 8.1%의 커맨드 버스 전력 소모량을 감소시켰다.

▸ **주제어**: DRAM, 저전력, 색인기반 명령어, 메모리, 시뮬레이션

• First Author: Kwangho Lee, Corresponding Author: Jongmin Lee
*Kwangho Lee (lkh002@wku.ac.kr), Dept. of Computer Engineering, Won-Kwang University
*Jongmin Lee (square55@wku.ac.kr), Dept. of Computer Engineering, Won-Kwang University
• Received: 2019. 10. 31, Revised: 2019. 11. 19, Accepted: 2019. 11. 20.

Copyright © 2019 The Korea Society of Computer and Information
http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945
www.kci.go.kr

I. Introduction

DRAM(Dynamic Random Access Memory)은 IT1C로 메모리 셀을 구현할 수 있는 고집적성과 빠른 접근 시간의 특성으로 지난 수십 년간 메인 메모리를 구현하기 위한 기억 장치로 주요하게 사용되어 왔다. 컴퓨팅 환경의 고성능화와 CMOS 공정 기술의 발전으로 DRAM의 크기가 계속 증가하는 추세이다. 따라서 DRAM 내부에 저장된 데이터의 위치를 가리키기 위한 메모리 주소의 크기가 증가하게 되고 이는 DRAM 커맨드를 전송하기 위한 버스 폭(bus width)의 확장을 필요로 한다. 하지만, 버스 폭을 증가시키면 메모리 컨트롤러와 DRAM 간의 고속 인터페이스에 부정적인 영향을 미칠 수 있다. 주소 버스가 확장됨에 따라 신호 무결성(signal integrity)을 유지하기 위한 타이밍 마진(margin)이 감소하고 제조비용뿐만 아니라 버스 폭 증가로 인한 전력소비 역시 증가한다. 이를 방지하기 위해 LPDDR4(Low-power DRAM)[1]와 HBM(High Bandwidth Memory)[2]과 같은 최근의 고용량 DRAM은 제한된 버스 폭의 크기를 유지하면서 패킷화(packetized manner) 된 방식으로 DRAM 커맨드를 여러 사이클에 걸쳐 전송한다.

DRAM 커맨드 중 가장 많은 주소정보를 포함하는 커맨드는 ACTIVE 커맨드이다. ACTIVE 커맨드는 DRAM의 구조적인 특성으로 인해 필요한 커맨드로 메모리에 데이터를 읽거나 쓰는 동작을 수행하기 전, Row 페이지(LPDDR4: 1KB-2KB)를 미리 활성화하기 위해 DRAM으로 전송된다. 일반적으로 ACTIVE 커맨드는 최대 16비트 크기의 Row 주소를 담고 있다. LPDDR4와 HBM은 모두 하나의 ACTIVE 커맨드 전송을 위해 공통적으로 두 개의 패킷을 구성하여 전송하고 있지만 전송하는 방식에 차이가 있다. LPDDR4는 클럭(clock)의 상승 엣지(rising edge)에만 패킷을 전송하는 방식으로 4 사이클에 걸쳐 ACTIVE 커맨드를 전송하고 HBM은 DDR(Double Data Rate) 방식으로 2사이클에 걸쳐 ACTIVE 커맨드를 전송한다. 위의 두 DRAM에서 16비트 Row 주소가 현재 DRAM용량에서 모두 사용되고 있기 때문에 향후 초고용량 메모리를 구현하기 위한 차세대 DRAM에서는 다수의 패킷이 필요할 수 있다.

본 논문에서는 다중 패킷으로 ACTIVE 커맨드를 전송하는 기존의 방법을 대신하여 하나의 패킷으로 ACTIVE 커맨드를 구현할 수 있는 Small Active Command 기법을 제안한다. 제안된 기법은 I-ACTIVE라고 명명된 작은 크기의 ACTIVE 커맨드 전송을 통해 메모리 접근 시간을 줄이고 커맨드 버스(Command bus)에서 소비되는 전력을 감소시킨다. 이를 위해 제안된 방법은 페이지 식별(page identification) 알고리즘을 통해 자주 사용되는 Row 페이지를 Hot 페이지로 먼저 분류한다. 선택된 페이지의 Row 주소는 메모리 컨트롤러와 DRAM에 위치한 작은 캐시에 저장된다. 인덱스 번호(index number)로 식별되는 캐시의 각 블록은 해당 Row 주소를 저장한다. 그 후, DRAM의 Row가 활성화 되어야 할 때, I-ACTIVE는 Row 주소 대신 캐시의 인덱스 번호만을 포함하여 작은 크기의 I-ACTIVE 커맨드를 DRAM으로 전송한다. DRAM은 I-ACTIVE커맨드를 이용하여 해당 위치의 캐시에 저장된 Row 주소를 이용하여 ACTIVE 커맨드를 위한 동작을 수행한다. 본 논문에서 제안하는 Small Active Command 기법은 기존 DRAM의 표준 JEDEC(Joint Electron Device Engineering Council) 스펙(specification)을 위반하지 않기 때문에 기존의 ACTIVE 커맨드와 동시에 사용이 가능하다. 사이클 단위의 정확도를 지원하는 시뮬레이터를 이용한 실험을 통해 제안된 기술이 성능 감소 없이 커맨드 버스의 전력 소비를 크게 감소시키는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 먼저 다음 장에서는 DRAM의 저전력 기술과 관련 있는 기존 연구들을 다룬다. 3장에서는 본 논문에서 제안하는 Small Active Command 설계와 구현에 관한 내용을 제시한다. 4장에서는 제안된 기법에 대한 실험 방법과 결과에 대한 내용을 기술한다. 마지막으로 5장에서는 본 논문의 결론과 향후 계획을 기술한다.

II. Preliminaries

1. Related works

1.1 Low-power DRAMs

DRAM의 전력소비는 시스템 전체 전력 소비에 큰 영향을 미치기 때문에 이에 대한 많은 연구가 수행되었다 [3-5]. 기존 연구의 대부분은 DRAM에서 지원하는 저전력 모드를 최대한 효과적으로 활용하여 DRAM에서 소비되는 전력 소비를 감소시키는 것을 목표로 한다. 최근의 연구는 DRAM의 오버페치 문제에 중점을 두었다. 기존 DRAM을 구성하는 셀 어레이(cell array) 구조를 더 작은 그룹으로 재구성하여 ACTIVE 커맨드에 의해서 활성화되는 Row 페이지의 크기를 줄이고 에너지 낭비를 줄이기 위한 연구가 수행되었다 [6-8]. 본 연구에서 제안하는 기법은 작은 크기의 I-ACTIVE 커맨드를 이용하여 커맨드 버스의 전력 소비를 감소시키는 것을 목표로 한다는 점에서 차이점이 있다.

www.kci.go.kr

1.2 Hot/Cold page identification

페이지 식별(page identification) 연구는 메모리에 저장된 데이터가 모두 고르게 참조되지는 않는 특성을 이용하여 페이지 이주 및 관리를 통해 시스템의 성능 향상 및 전력 소비 감소를 목표로 한다. 접근 시간(latency), 대역폭(bandwidth) 및 유지시간(retention time)에서 서로 다른 특성을 가지는 메모리 소자를 조합하여 구성된 하이브리드(hybrid) 메모리 시스템에서 활발한 연구가 진행되었다. [9-12]. 일반적으로 페이지 식별 기법을 통해 자주 참조되는 페이지를 Hot으로 드물게 참조되는 페이지를 Cold로 분류할 수 있다. 관련 연구[9]는 PCM-DRAM 구조의 하이브리드 메모리 시스템을 대상으로 하여 Hot/Cold 페이지가 저장되는 메모리 위치를 관리한다. PCM(Phase Change Memory)은 DRAM보다 큰 용량의 메모리 설계가 가능하지만 셀 당 쓰기 횟수가 제한되어 있어 메모리의 수명이 짧다. 관련 연구[10]은 NVM-DRAM 구조의 시스템을 대상으로 한다 PCM과 유사하게 NVM(Non Voltage Memory)도 DRAM보다 큰 용량의 메모리 설계가 가능하지만 DRAM보다 접근 속도가 느리다는 단점이 있다. 이러한 특성을 이용하여 자주 참조되는 Hot 페이지를 DRAM에 할당하고 Cold 페이지를 NVM에 할당하여 성능 차이를 극복한다.

자주 참조되는 페이지를 효율적으로 찾기 위한 다양한 연구도 수행되었다. 관련 연구[11] 에서는 멀티 큐(multi queue) 구조를 이용하여 각각의 큐에 랭킹을 부여하고 참조 횟수에 따라 페이지를 이동하여 Hot/Cold 페이지를 구분하는 방법을 제안하였다. 관련 연구[9]는 커널레벨(kernel-level)에서 관리하는 페이지 테이블 엔트리(page table entry)의 dirty bit를 이용하여 각 페이지의 히스토리 정보를 기록하고 인접한 페이지 주소의 참조 횟수를 평균하여 Hot/Cold 페이지를 구분한다. 관련 연구[10]은 LSTM 신경망을 이용하여 할당된 페이지의 참조 횟수를 예측하여 Hot/Cold 페이지를 구분하는 방법을 제안하였다. 기존의 하이브리드 메모리 시스템에서 사용된 히스토리 기반 페이지 스케줄러를 이용하여 페이지를 할당한 후 학습된 LSTM을 사용하여 분류된 페이지의 추후 참조 횟수를 예측하여 적합한 메모리에 페이지를 이주하는 방법이다.

본 연구에서 제안하는 페이지 분류 기법은 페이지의 ACTIVE 커맨드 발생 횟수와 히스토리 정보를 저장하는 간단한 구조의 작은 캐시를 이용하여 Hot/Cold 페이지를 분류하는 점에서 기존의 연구와 차이가 있다.

2. Background

본 절에서는 DRAM의 기본 동작과 커맨드에 대해 다룬다. DRAM의 기본적인 동작은 Activation, Precharge, Read/Write, Refresh로 구분할 수 있다. Activation은 메모리에 Read/Write를 수행하기 전 반드시 선행되어야 하는 동작으로 해당 주소의 Row 페이지를 sense amp(혹은 row buffer)로 저장하여 버퍼와 같이 메모리 셀의 데이터를 저장하는 동작을 수행한다. 그 후, row buffer에 활성화되어 있는 Row를 통해 Read/Write를 수행 할 수 있다. 하나의 Row는 많은 수의 메모리 셀을 포함하고 있고 Activation 동작을 수행하는 비용이 비싸기 때문에 두 개의 row buffer 관리 정책을 사용하여 Activation 동작을 위한 지연 시간 및 전력 소비 부담(overhead)을 완화한다.

먼저, Open-page 정책은 다른 Row가 활성화 될 때까지 현재 row buffer에 보관된 Row를 최대한 유지한다. 활성화된 Row에 접근하는 Read 및 Write 요청으로 row buffer hit가 발생하면 추가적인 Activation을 위해 필요한 지연시간 및 전력 소비를 제거할 수 있다. 따라서 Open-page 정책은 응용프로그램에서 메모리 접근 패턴이 높은 데이터 지역성(data locality)을 보이는 경우 유용하다. 그러나 뒤따르는 메모리 접근이 현재 활성화 되어 있는 Row가 아닐 경우 Open-page 정책은 효과적이지 않다. 다른 Row로 Read/Write가 필요한 경우 현재 활성화된 Row를 닫기 위해 Precharge와 Activation이 필요하다. Precharge 동작은 row buffer에 보관되어 있는 Row를 다시 메모리 셀로 저장하는 동작을 수행한다. 메모리 주소를 재구성(remapping)하여 현재 활성화 되어 있는 Row의 재사용성을 높이려는 연구가 진행되었다 [13-14]. Close-page 정책은 다음 메모리 요청을 준비하기 위해 Read/Write 후 즉시 Precharge를 수행한다. 다른 Row 메모리 접근 요청이 발생할 경우 Open-page 정책에서는 Precharge가 필요한데 반해, Close-page 정책은 뒤따르는 메모리 접근이 이전과 다른 Row일 경우 지연시간 측면에서 이점이 있다. 고성능 멀티코어 프로세서는 다수의 응용프로그램이 동시에 수행되며 메모리 접근 패턴이 낮은 지역성을 보이는 경향을 보이기 때문에 Close-page 정책이 유리하다.

Command	CS	CA0	CA1	CA2	CA3	CA4	CA5
Active-1	H	H	L	R12	R13	R14	R15
	L	BA0	BA1	BA2	V	R10	R11
Active-2	H	H	H	R6	R7	R8	R9
	L	R0	R1	R2	R3	R4	R5
Precharge	H	L	L	L	L	H	AB
	L	BA0	BA1	BA2	V	V	V

Fig. 1. DRAM Command Truth Table of LPDDR4 (ACTIVE, PRECHARGE) [1]

www.kci.go.kr

메모리 컨트롤러에서 DRAM으로 전송되는 명령어는 JEDEC에서 발표한 커맨드 규칙을 따르는데 Fig. 1은 본 논문의 주요 대상인 LPDDR4 DRAM의 커맨드 진리표(truth table)를 나타낸다. 본 논문과 밀접하게 관련 있는 ACTIVE와 PRECHARGE 커맨드를 Fig. 1에 표시하였다. LPDDR4에서 ACTIVE 커맨드를 통해 전송되는 Row 주소의 크기는 최대 16비트(R0-R15)이고 ACTIVE-1과 ACTIVE-2로 두 개의 패킷으로 나누어 클럭의 상승 엣지(rising edge) 타이밍에 총 4 클럭 사이클에 걸쳐 전송된다. ACTIVE-1은 대상 뱅크(bank) 주소와 상위 Row 주소를 포함하고 있으며 ACTIVE-2 커맨드는 하위 Row 주소를 포함한다. V(valid)로 표시된 부분은 커맨드를 해석(decode)할 때 사용되지 않는 부분으로 0 혹은 1의 값을 나타낸다. PRECHARGE 커맨드는 AB(CA5)신호를 통해 all bank precharge와 per bank precharge를 구별한다. per bank precharge는 해당 뱅크의 활성화된 Row를 닫고 all bank precharge는 전송된 뱅크 주소와 상관없이 DRAM의 모든 활성화된 Row를 닫는다. 본 논문에서는 위의 ACTIVE와 PRECHARGE 커맨드를 이용하여 다중 패킷에 걸쳐 전송되는 ACTIVE 커맨드를 작은 크기로 개선하기 위한 기법을 제안한다.

III. The Proposed Scheme

1. Overview

본 논문에서 제안하는 Small Active Command 기법은 단일 패킷 ACTIVE 커맨드를 제안하여 메모리 접근속도와 커맨드 버스의 전력 소비를 감소시키는 것을 목표로 한다. 이를 위해 자주 활성화 되는 Row 페이지를 식별한 후 메모리 컨트롤러와 DRAM에 위치한 작은 캐시에 해당 정보를 저장하고 캐시의 인덱스를 이용하여 기존의 다중 패킷 ACTIVE 커맨드로 전달되는 주소 정보의 크기를 감소시킨다.

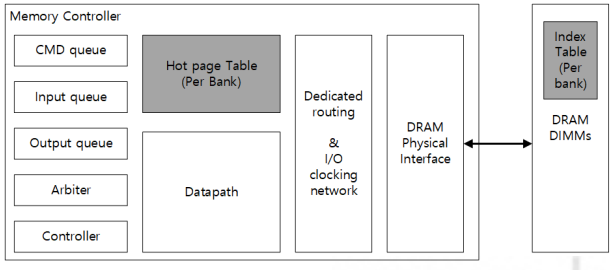

Fig. 2. Overall Structure of Small Active Command

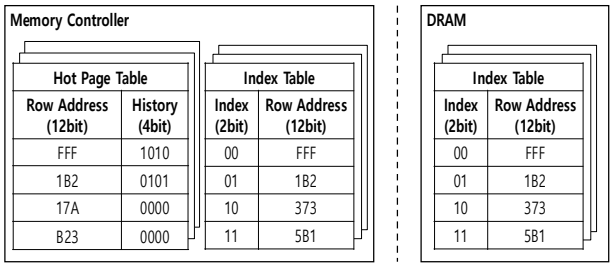

Fig. 3. Table Structure of Hot Page Table and Index Table

Fig. 2는 Small Active Command 기법이 적용된 DRAM 시스템의 전체적인 구조를 나타낸다. 기존의 메모리 컨트롤러와 DRAM에 추가된 모듈을 회색으로 표시하였다. Hot Page Table은 DRAM에서 자주 참조되는 Row 페이지(Hot)를 식별하기 위해 필요한 Activation 히스토리 정보를 저장한다. Index Table은 Hot으로 분류된 Row 주소와 캐시 인덱스 번호를 저장한다. Index Table은 메모리 컨트롤러와 DRAM 양쪽에 위치하는데 DRAM에 위치한 Index Table은 캐시 인덱스 전송 후 Row 주소로 변환하는데 사용되며 메모리 컨트롤러에 위치한 테이블과 동기화되어 관리된다. 메모리 컨트롤러와 DRAM간 Index Table 동기화 및 ACTIVE 커맨드에 Row 주소를 대신하여 캐시 인덱스를 전송하기 위해 Small Active Command 기법에서는 새로운 DRAM 커맨드를 추가하였다. 추가된 테이블, 커맨드 진리표 및 Hot 페이지 식별 알고리즘에 대해서는 이어지는 절에서 자세히 다룬다.

2. Table Structure

Small Active Command 기법을 구현하기 위해 두 종류의 테이블(캐시)을 추가하였다. Hot Page Table은 Activation 동작의 히스토리 정보를 저장하여 자주 참조되는 페이지를 식별하기 위해 사용되고 Index Table은 캐시의 인덱스 번호와 해당 Row 주소 일부를 저장하기 위해 사용된다. Fig. 3은 두 테이블의 구조를 나타낸다. 먼저, Hot Page Table 4개의 엔트리(entry)로 구성되어 있으며 각 엔트리는 Row 주소 R0-R11을 저장하기 위한 12-비트와 참조 히스토리 정보를 저장하기 위한 4-비로 구성된다. Index Table 역시 4개의 엔트리로 구성되어 있으며 각 엔트리는 인덱스 번호를 저장하기 위한 2-bit와 Row 주소를 저장하기 위한 12비트로 구성된다. Hot Page Table은 메모리 컨트롤러에만 위치하고 있으며 Index Table은 인덱스 번호의 전송 및 해당 인덱스를 이용한 Row 주소 변환을 위해 메모리 컨트롤러와 DRAM 양쪽에 모두 존재한다. 각 테이블은 뱅크별로 독립적인 동작을 수행하기 때문에

www.kci.go.kr

DRAM 뱅크의 개수(LPDDR4: 8개)만큼 테이블이 생성된다.

3. I-ACTIVE, I-PRECHARGE Command

본 절에서는 Small Active Command 기법의 핵심 동작을 구현하기 위해 추가된 DRAM 커맨드에 대해 다룬다. Fig. 4는 새롭게 추가된 I-ACTIVE와 I-PRECHARGE 커맨드의 진리표를 나타낸다. 각 커맨드는 Fig. 1에 표시한 기존 커맨드에서 두 번째 클럭의 CA3 valid(V) 비트를 이용하여 구별되며 해당 비트가 H이면 각각 I-ACTIVE와 I-PRECHARGE 커맨드를 의미하고 L이면 기존의 ACTIVE-1과 PRECHARGE 커맨드를 의미한다. I-ACTIVE 커맨드는 기존의 ACTIVE 커맨드를 대신하여 Index Table에 저장된 Row를 활성화시키기 위해 사용하는 커맨드로 첫 번째 클럭에 Row의 상위 주소(R12-R15)를 전달하고 두 번째 클럭에 뱅크(bank) 주소와 남은 두 비트(CA4, CA5)를 이용하여 인덱스 정보를 DRAM에 전달한다. DRAM 내부 동작을 살펴보면, 첫 번째 클럭에서 ACTIVE-1 커맨드를 해석(CA0-CA1)한 후 부분-어레이(sub-array)의 wordline을 활성화하기 위해 Row의 상위 주소가 사용되기 때문에 I-ACTIVE 커맨드는 기존의 ACTIVE-1 커맨드와 첫번째 클럭의 전송 정보를 같게 유지하였다. I-ACTIVE 커맨드는 기존의 ACTIVE-1 커맨드와 달리 후속 커맨드가 필요하지 않으며 하나의 패킷을 이용하여 두 사이클에 걸쳐 Activation 동작에 필요한 정보를 전달할 수 있다.

I-PRECHARGE 커맨드는 메모리컨트롤러와 DRAM에 위치한 Index Table 간의 동기화 동작을 수행한다. Hot 페이지로 분류된 Row 페이지는 기존의 ACTIVE 커맨드를 이용하여 활성화 된 후 I-PRECHARGE 커맨드로 페이지 닫기 동작을 수행하기 전 일부 Row 주소(R0-R11)가 DRAM에 위치한 Index Table에 저장된다. I-PRECHARGE 커맨드에 담을 수 있는 정보의 양(2-비트)이 제한적이기 때문에 per bank precharge의 경우에만 인덱스 정보에 대한 동기화를 수행한다.

Command	CS	CA0	CA1	CA2	CA3	CA4	CA5
I-Active	H	H	L	R12	R13	R14	R15
	L	BA0	BA1	BA2	H	I	I
I-Precharge	H	L	L	L	L	H	AB
	L	BA0	BA1	BA2	H	I	I

Fig. 4. I-ACTIVE, I-PRECHARGE Command Truth Table

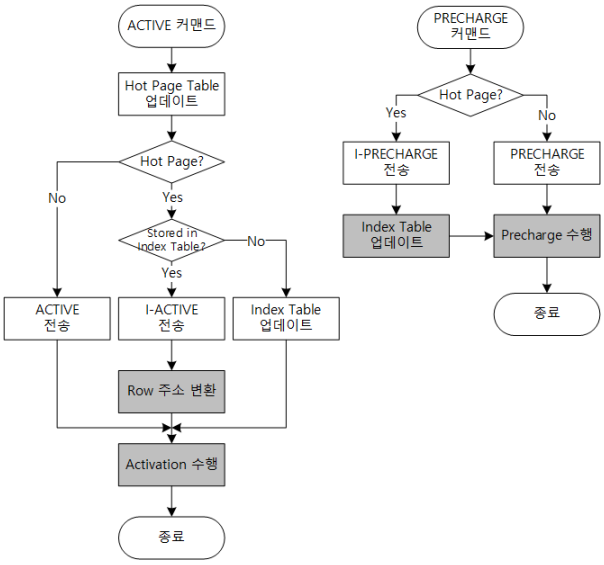

Fig. 5. Sequential Diagram of Small Active Command

본 논문에서 제안하는 Small Active Command 기법은 기존 ACTIVE-1과 PRECHARGE 커맨드에서 사용하지 않는 V(valid) 비트를 이용하여 새로운 커맨드를 추가하였기 때문에 LPDDR4의 커맨드 규칙을 위반하지 않는다.

Fig. 5는 I-ACTIVE와 I-PRECHARGE 커맨드를 포함한 Small Active Command 기법의 전반적인 동작순서를 나타낸다. 메모리 컨트롤러에서의 동작은 흰색으로 DRAM에서의 동작은 회색으로 표시하였다. ACTIVE 커맨드가 Hot Page Table에서 Hot 페이지로 식별되면 해당 Row 페이지를 메모리 컨트롤러의 Index Table에 저장하고 기존의 ACTIVE 커맨드를 전송하여 DRAM에서 Activation 동작을 수행한다. 해당 Row의 사용이 끝나면 I-PRECHARGE 커맨드를 사용하여 DRAM의 Index Table에 해당 Row 주소를 저장한다. 이후 Inex Table에 저장된 Row가 다시 참조될 경우, 메모리 컨트롤러에 위치한 Index Table의 Row 주소를 참조하여 DRAM으로 I-ACTIVE 커맨드를 통해 인덱스를 전달한다. DRAM은 해당 인덱스 번호를 참조하여 Row 주소를 변환하고 Activation 동작을 수행한다.

4. Hot Page Identification Algorithm

페이지 식별 알고리즘을 사용하여 Row 페이지는 참조되는 빈도에 따라 Hot 페이지와 Cold 페이지로 분류된다. 페이지 식별을 수행할 수 있는 다양한 알고리즘이 존재하지만 [9-12] 본 논문에서는 간단한 시프트(Shift) 연산을 통해 Row 페이지를 식별할 수 있는 알고리즘을 제안한다.

www.kci.go.kr

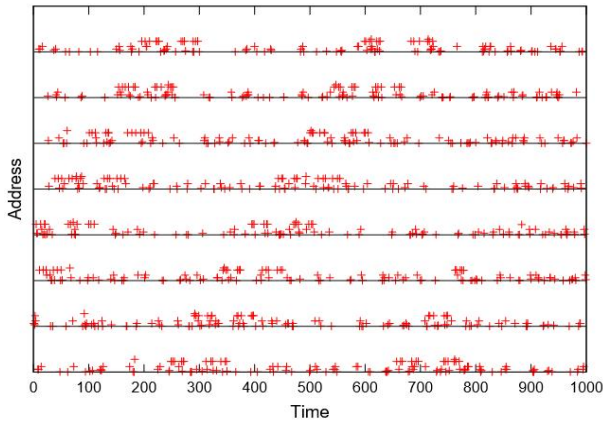

Fig. 6. Memory Access Pattern of mix4

Fig. 6은 SPEC2006 벤치마크를 랜덤하게 그룹화(mix4)하고 일정 구간 동안 수행하여 추출한 메모리 접근 패턴을 나타낸다. 실험에 대한 자세한 사항은 다음 장에서 다루도록 하겠다. Fig. 6의 X-축은 시간을 나타내고 Y-축은 DRAM의 각 뱅크로 접근된 ACTIVE 커맨드의 Row 주소를 나타낸다. 각각의 점으로 뱅크에서 Activation 동작을 수행한 Row 주소를 표시하였다. 메모리 데이터 지역성의 특성으로 일부 Row 주소가 계속적으로 접근되거나 일정 구간 자주 접근되는 것을 확인할 수 있다. 이러한 Row 페이지를 Hot 페이지로 정의하고 해당 Row 페이지를 분류할 수 있는 방법을 제시한다.

Small Active Command 기법은 Hot 페이지를 분류하기 위해 ACTIVE 커맨드의 히스토리 정보를 이용한 분류 알고리즘을 사용한다. Fig. 3에 보인 것과 같이 Hot Page Table 엔트리의 History 필드(4-비트)가 페이지 식별에 사용된다. 먼저, ACTIVE 커맨드 발생 시 해당 뱅크의 Hot Page Table에 저장된 History 필드를 모두 오른쪽으로 시프트한다. 이후 해당 Row 주소가 Hot Page Table에 저장되어 있는 경우 History 필드의 최상위 비트를 1로 변경한다. 만약, 해당 Row 주소가 Hot Page Table에 없다면, 새로운 Row 주소를 LRU방식으로 엔트리에 저장하고 History 필드의 최상위 비트를 1로 변경한다. 따라서 History 필드의 상위 비트는 최근에 참조가 되었음을 나타낸다. Hot Page Table의 엔트리 중 History 필드의 값이 10(10102)이상인 경우 Hot 페이지로 결정한다. Hot 페이지로 결정된 ACTIVE 커맨드의 일부 Row 주소(R0-R11)는 Index Table에 저장된다.

제안된 페이지 식별 알고리즘은 DRAM으로 ACTIVE 커맨드가 전송되기 전 메모리 컨트롤러의 CMD queue에서 대기하는 동안 수행되기 때문에 알고리즘 수행으로 인한 성능감소를 유발하지 않는다.

IV. Experiments

1. Experimental Methodology

본 논문에서 제안한 Small Active Command 기법의 시스템 성능 변화 및 커맨드 버스의 전력감소를 측정하기 위해 GEM5[17]와 CACTI[18] 시뮬레이터를 이용한 시뮬레이션 기반의 실험을 수행하였다. Table 1은 실험에 사용한 ARM 프로세서와 LPDDR4 DRAM의 파라미터를 나타낸다.

Table 1. System Environment

Feature	Value
Processor	ARM (4-Cores)
Clock	2GHz
L1 Data $	32KB
L1 Instruction $	32KB
L2 Unified $	128KB
L3 Unified $	2MB
DRAM	LPDDR4
DRAM Size	2GB
Row Policy	Open, Close
# of Bank	8
# of Rank	2
# of Channel	2

Table 2는 실험에서 사용한 LPDDR4 DRAM의 타이밍 파라미터를 나타낸다. 타이밍 파라미터는 Micron사에서 제공하는 LPDDR4 스펙[1]을 참조하여 설정하였다. 또한, 멀티코어 환경에서 다수의 어플리케이션이 DRAM에 경쟁적으로 접근하는 상황을 구현하기 위해 SPEC2006 벤치마크[17]의 응용프로그램을 랜덤하게 선택(4개)한 후 10개의 mix셋(mix1-mix10)을 생성하여 4개의 프로세서 코어에 병렬적으로 시뮬레이션을 수행하였다. GEM5 시뮬레이션에서 각 mix셋은 10억개의 프로세서 명령어를 fast-forward하고 100만개의 프로세서 명령어를 수행한 결과를 추출하였다.

Table 2. LPDDR4 DRAM Timing Parameters

Feature	Value
tCK	1.25ns
tRCD	18ns
tCL	0.675ns
tRAS	42ns
tWR	18ns
tRTP	7.5ns
tRP	18ns
tBURST	5ns
tRFC	130ns
tREFI	3.9us
tXP	7.5ns
tWTR	10ns
tRTW	2.5ns
tCS	1.75ns
tRRD	10ns
tFAW	40ns

www.kci.go.kr

2. Experimental Results

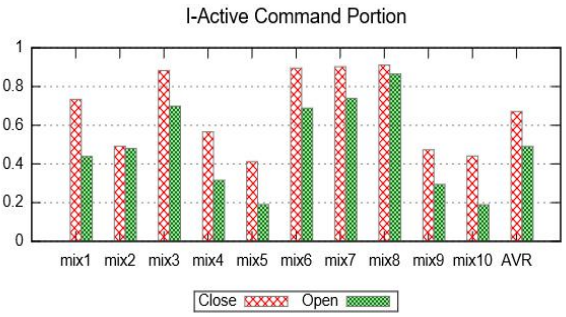

Fig. 7. I-Active Command Portion of Mix Set

Small Active Command 기법의 Hot 페이지 식별 성능 평가를 위해 각 벤치마크 mix셋을 수행 후, 기본 수행(Default)의 ACTIVE 커맨드 대비 I-ACTIVE 커맨드의 비율을 정규화(normalized)하여 Fig. 7에 표시하였다. X-축은 각각의 mix set을 나타내고 Y-축은 I-Active 커맨드의 비율을 나타낸다. 각 막대그래프의 왼쪽은 Close-page 정책을 나타내고 오른쪽은 Open-page 정책을 나타낸다. Open-page 정책에서 I-ACTIVE 커맨드 비율은 최대 86%(mix8), 최소 18.9%(mix10), 평균 48.9%를 보이며 Close-page 정책에서는 최대 91%(mix8), 최소 41.2%(mix5), 평균 67%를 보인다. mix8의 경우 일부 페이지를 계속 참조하는 메모리 접근 패턴을 보이기 때문에 I-ACTIVE의 커맨드 비율이 높은 것을 확인할 수 있지만, mix5,9,10의 경우 할당된 페이지의 수가 많고 다양한 패턴으로 메모리를 참조하여 제안한 페이지 식별 알고리즘의 성능이 상대적으로 낮게 평가되었다.

Small Active Command 기법은 ACTIVE 커맨드를 하나의 패킷으로 전송할 수 있기 때문에 ACTIVE 커맨드의 지연시간(latency)을 감소시킨다. Table 2에 표시한 DRAM의 주요 파라미터 중 ACTIVE 커맨드와 관련 있는 것은 tRAS, tRRD, tRCD이다. Equation (1)을 이용하여 각 파라미터의 평균적인 지연시간을 계산할 수 있다.

$$\text{Reduced parameter} = \textit{Hot page rate} * (\textit{parameter} - \textit{2cycle}) + \textit{Cold page rate} * \textit{parameter} \quad (1)$$

Hot page rate는 Hot 페이지로 분류되어 I-ACTIVE 커맨드를 전송하는 비율을 나타내고 Cold page는 기존의 ACTIVE 커맨드를 전송하는 비율을 나타낸다.

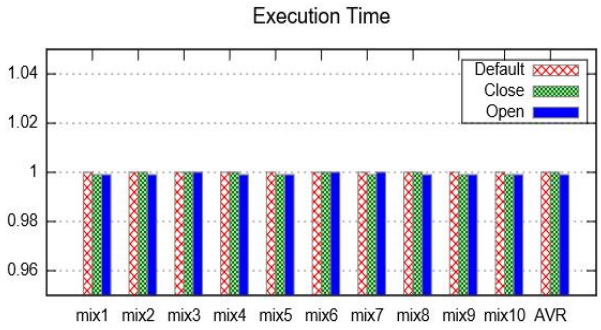

Fig. 8. Execution Time of Mix Set

Fig. 8은 Small Active Command 기법을 적용하여 수행한 벤치마크의 실행시간(execution time)을 나타낸다. Y-축은 Default를 기준으로 정규화한 실행 시간을 나타낸다. Small Active Command 기법을 적용하여 ACTIVE 커맨드의 지연시간을 감소시켰지만 벤치마크의 수행 시간 감소량이 적은 것을 확인할 수 있다(0.1%미만). 이는 DRAM의 동작 특성으로 인한 것으로 Table 2에 표시한 DRAM 파라미터 중 tFAW가 해당 시간 동안 동일 랭크에서 발생할 수 있는 ACTIVE 커맨드를 4개로 제한하기 때문이다. 즉, I-ACTIVE의 커맨드로 지연시간이 2 사이클 감소하지만 tFAW 파라미터가 4 개의 ACTIVE 커맨드 이후 다음 ACTIVE 커맨드를 tFAW 주기 동안 제한하기 때문에 실행시간 감소폭이 작게 측정되었다.

Table 3. CACTI Simulation Results of Caches

	Hot Page Table	Index Table
Access Time (ns)	0.43	0.37
Access Energy (pJ)	0.31	0.22
Area (μm^2)	698.34	291.33

DRAM 커맨드 버스의 전력 감소를 평가하기 위해 먼저 메모리 컨트롤러와 DRAM에 추가된 Hot Page Table과 Index Table을 CACTI[16] 시뮬레이터를 이용하여 Access Time, Access Energy, Area를 측정하여 결과를 Table 3에 표시하였다. CACTI 시뮬레이터는 LPDDR4 DRAM의 14nm 공정을 지원하지 않아 22nm 공정으로 실험을 수행하였다. 두 테이블의 Access Energy를 Gem5에서 측정한 접근 횟수와 곱하여 Small Active Command 기법으로 인한 추가적인 에너지를 계산하였다.

www.kci.go.kr

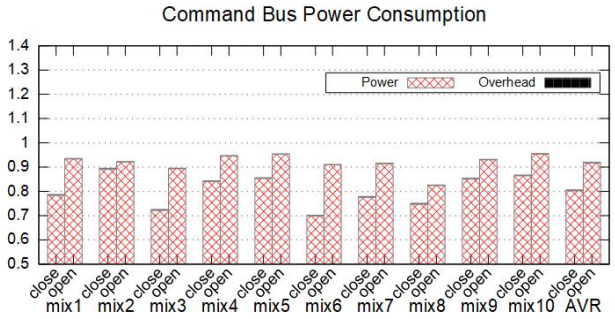

Fig. 9. Command Bus Power Consumption of Mix Set

Fig. 9는 Small Active Command 기법으로 인한 오버헤드(overhead)를 포함한 커맨드 버스의 평균적인 전력소비를 나타낸다. Y-축은 Default를 기준으로 정규화한 전력 소모량을 나타낸다. 커맨드 버스의 전력 소비 계산을 위해 관련연구[18-19]의 Switching Activity 기반의 Dynamic Power Equation을 사용하였다. 각 막대그래프의 왼쪽은 Close-page 정책, 오른쪽은 Open-page 정책을 나타낸다. Hot Page Table과 Index Table로 인한 전력 소비는 0.1%~0.3% 으로 버스 전력소비와 비교하여 매우 낮은 수준인 것으로 나타났다 (각 믹스셋 10μW미만). Close-page 정책에서 파워 소모량 감소율은 최대 30.1%(mix6), 최소 10.7%(mix2), 평균 20% 감소하였다. Open-page 정책에서는 최대 17.5%(mix8), 최소 4.6%(mix5,10) 평균 8.1% 감소하였다. Small Active Command 기법은 ACTIVE 커맨드의 비율이 높은 Close-page 정책에서 보다 효과적인 것을 확인하였다.

V. Conclusions

DRAM(Dynamic Random Access Memory)은 메인 메모리를 구현하기 위한 주요 장치로 사용되고 있다. 컴퓨팅 환경의 고성능 요구와 CMOS 공정 기술의 발전에 따라 DRAM의 크기가 대용량화되어 가고 이에 따라 DRAM으로 전송되어야 하는 메모리 주소 정보의 양이 증가한다. 본 논문에서는 ACTIVE 커맨드를 단일 패킷으로 Row 주소를 전송할 수 있는 새로운 주소 지정 체계를 제안하였다. Gem5 시뮬레이터를 이용한 실험 결과 Small Active Command 기법을 적용하였을 때 Close-page 정책과 Open-page 정책에서 각각 20%, 8.1% 평균적인 커맨드 버스의 전력소비를 감소하는 것을 확인하였다. 본 논문에서는 LPDDR4의 ACTIVATE, PRECHARGE 커맨드의 valid 비트를 이용하여 새로운 커맨드를 추가하였지만 다른 종류의 장치에서는 스펙에 따라 커맨드를 다르게 추가해야하는 제약이 존재한다. 향후의 계획은 DRAM에서 빈번하게 사용하는 REFRESH 커맨드를 대상으로 커맨드 버스에서 소비되는 전력을 감소시킬 수 있는 방안에 대한 연구이다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2018R1C1B5046647).

REFERENCES

[1] JEDEC Solid State Technology Association, Lower Double Data Rate 4 (LPDDR4), August 2014, https://www.jedec.org

[2] JEDEC Solid State Technology Association, High Bandwidth Memory (HBM), October 2013, https://www.jedec.org

[3] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt, "Understanding and designing new server architectures for emerging warehouse-computing environments," Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA), pp. 315-326, Beijing, China, 2008.

[4] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: Eliminating server idle power," Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XIV), pp. 205-216, Washington DC, USA, 2009.

[5] Y. Lee, S. Kim, S. Hong, and J. Lee, "Skinflint DRAM System: Minimizing DRAM Chip Writes for Low Power," Proceedings of the 19th International Symposium on High Performance Computer Architecture (HPCA), pp. 25-34, Shenzhen, China, 2013.

[6] J. Ahn, N. P. Jouppi, C. Kozyrakis, J. Leverich, and R. S. Schreiber, "Future Scaling of Processor-memory Interfaces," Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC), pp. 1-12, Portland, USA, 2009.

[7] A. N. Udipi, N. Muralimanohar, N. Chatterjee, R. Balasubramonian, A. Davis, and N. P. Jouppi, "Rethinking DRAM Design and Organization for Energy-constrained Multi-cores," Proceedings of the 37th annual International Symposium on Computer Architecture (ISCA), pp. 175-186, Saint-Malo, France, 2010.

www.kci.go.kr

[8] H. Zheng, J. Lin, Z. Zhang, E. Gorbatov, H. David, and Z. Zhu, "Mini-rank: Adaptive DRAM Architecture for Improving Memory Power Efficiency," Proceedings of the 41st IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 210-221, Lake Como, Italy, 2008.

[9] D. Shin, S. Park, S. Kim, and K. Park, "Adaptive Page Grouping for Energy Efficiency in Hybrid PRAM-DRAM Main Memory," Proceedings of the 2012 ACM Research in Applied Computation Symposium, pp. 395-402, San Antonio, USA, 2012.

[10] Thaleia Dimitra Doudali, Sergey Blagodurov, Abhinav Vishnu, Sudhanva Gurumurthi, and Ada Gavrilovska, "Kleio: A Hybrid Memory Page Scheduler with Machine Intelligence." In Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing (HPDC), pp. 37-48, New York, USA, 2019

[11] L. Ramos, E. Gorbatov, and R. Bianchini, "Page Placement in Hybrid Memory Systems," Proceedings of the International Conference on Supercomputing (ICS), pp. 85-95, Tucson, USA, 2011.

[12] I. Shin, "Hot/cold clustering for page mapping in NAND flash memory," in IEEE Transactions on Consumer Electronics, Vol. 57, No. 4, pp. 1728-1731, November 2011. DOI:10.1109/TCE.2011.6131147

[13] B. Jacob, S. W. Ng, and D. Wang, "Memory Systems: Cache, DRAM, Disk," Morgan Kaufmann Publishers, 2007.

[14] Z. Zhang, Z. Zhu, and X. Zhang, "A Permutation-based Page Interleaving Scheme to Reduce Row-buffer Conflicts and Exploit Data Locality," Proceedings of the 33rd IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 32-41, Monterey, USA, 2000.

[15] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The Gem5 Simulator," SIGARCH Computer Architecture News, Vol. 39, No 2, May 2011.

[16] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "CACTI 6.0: A Tool to Model Large Caches," HP Laboratories, 2009.

[17] J. L. Henning, "SPEC CPU2006 Benchmark Descriptions," ACM SIGARCH Computer Architecture News, pp. 1-17, September 2006. https://www.spec.org

[18] K. Ning and D. Kaeli, "Bus Power Estimation and Power-Efficient Bus Arbitration for System-on-a-Chip Embedded Systems," Proceedings of the 4th International Conference on Power-Aware Computer Systems (PACS), pp. 95-106, Portland, USA, 2004.

[19] D. Liu and C. Svensson, "Power Consumption Estimation in CMOS VLSI Chips," IEEE Journal of Solid-State Circuits, Vol. 29, Issue 6, pp. 663-670, 1994. DOI:10.1109/4.293111

www.kci.go.kr

Authors

Kwangho Lee received the B.S. degree in computer engineering from Wonkwang University, Korea, in 2019. He is currently a graduate student at Wonkwang University, Korea. He is interested in embedded systems/software, operating systems, computer architectures.

Jongmin Lee received the B.S. degree in computer science and engineering from Dankook University, Korea, in 2008. He received the integrated master's Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2015. Dr. Lee joined the faculty of the Department of Computer Engineering at WonKwang University, Iksan, Korea, in 2018. He is currently an assistant Professor in the Department of Computer Engineering, WonKang University. He is interested in computer architectures, memory systems, and embedded systems/software.