

# DRAM Per-Row Activation Counting 기법의 시스템 성능 영향 분석

양성필<sup>§†</sup>, 강태운<sup>§</sup>, 구건재<sup>§</sup>

<sup>§</sup>고려대학교, <sup>†</sup>SK하이닉스

e-mail : yangs@korea.ac.kr, taewoon\_kang@korea.ac.kr, gunjaekoo@korea.ac.kr

## Performance Analysis of a Per-Row Activation Counting Mechanism in DRAM

Seongpil Yang<sup>§†</sup>, Taewoon Kang<sup>§</sup> and Gunjae Koo<sup>§</sup>

<sup>§</sup>Korea University, <sup>†</sup>SK Hynix

### Abstract

In this study we evaluate the performance overhead of per-row activation counting (PRAC) mechanism employed in DDR5 DRAM. PRAC is designed to handle the security attacks against DRAM cells by enabling refresh to target and adjacent rows based on access counts since the last activation. However, PRAC requires additional peripheral circuits that increase DRAM timing and refresh counts, thus PRAC may provoke significant performance overhead.

In order to investigate the performance hurdles by PRAC in DRAM, we analyze the performance impacts by adjusting DRAM timing parameters and refresh controls when PRAC is enabled in DRAM. We use the cycle-accurate DRAM simulator to evaluate the performance changes by various PRAC configurations. Our evaluation exhibits the performance overhead by PRAC is 8.5% on average when PRAC configurations are set up based on the JEDEC DDR5 standard.

### I. 서론

대규모 AI 모델과 같이 현재 널리 사용되는 주요 어플리케이션은 고용량 및 고성능 DRAM을 요구하고 있다. 이러한 요구에 맞추어 DRAM의 데이터 셀 크기를 줄여 하나의 DRAM 칩 저장 용량을 향상시키기 위한 연구가 활발하게 진행되고 있다. 또한 DRAM의 성능을 향상시키기 위하여 DRAM 동작 타이밍이 빠르게 이루어지도록 설계를 개선하고 있다. 이러한 DRAM 기술의 트렌드는 필연적으로 DRAM의 데이터 셀의 신뢰성 이슈를 초래하고 있다.

이와 함께 최근 수년간 DRAM의 구조적 특성으로 인해 발생할 수 있는 보안 취약점들이 보고되고 있다. 대표적으로 Row Hammer(RH) 공격은 DRAM의 특정 row을 반복적으로 활성화시킬 때 인접 row의 데이터가 변경되는 현상을 이용한 공격 방식으로서, 특정 데이터를 변경시킬 수 있다는 점에서 매우 치명적인 보안 위협으로 여겨지고 있다[6].

이를 해결하기 위해 다양한 메모리 관점의 대응 기법이 연구 및 적용되어 왔다. 대표적으로 DRAM 내부에서 특정 행(row)의 활성화 횟수를 추적해 대상 row 및 주변 row에 저장된 데이터를 보장하는 Target Row Refresh(TRR), DRAM에 패리티(parity) 셀을 추가해 오류를 보정하는 On-Die ECC, 그리고 그 외 다양한 방어 설계 기법들이 지속적으로 개발되어 RH 공격을 막기 위해 노력해왔다[9,10]. 그러나 점차 고도화되는 공격에는 메모리 차원의 대응만으로는 한계가 존재했다. 이러한 배경으로 인해 시스템 차원에

서의 대응 전략으로 Refresh Management (RFM)가 제안되었으며, 이는 메모리 컨트롤러가 특정 row의 접근 횟수를 감시하여 일정 임계치를 초과할 경우, 추가 리프레시를 수행하는 방식이다[1].

하지만 RFM은 단순 카운트 기반의 대응으로 인해 정밀도가 떨어지며, 오히려 불필요한 리프레시를 더 자주 해주어 리소스를 낭비할 수 있다는 단점을 가지고 있다. 이러한 한계를 극복하고 보다 정밀한 대응이 가능한 기법으로 Per-Row Activation Counting (PRAC)이 최신 DDR5 규격으로 등장하였다[1]. PRAC는 각 row에 대해 개별적인 활성화 카운트를 유지함으로써 실제 위협이 되는 row에만 선택적으로 리프레시를 적용할 수 있는 장점을 가진다.

본 논문은 PRAC 기법의 작동 원리와 그 한계를 분석하고, 이를 사이클 기반 DRAM 시뮬레이터로 구현하여 다양한 조건 하에서 평가함으로써 향후 메모리 보안 기법의 설계에 기여하고자 한다[5].

## II. 본론

### 2.1 PRAC 원리 및 작동 방식

PRAC는 DRAM 내 각 row 활성화 횟수를 추적해 일정 임계치가 넘으면 인접 row를 보호하는 방식이다. PRAC이 적용된 DRAM은 활성화 카운트 초기화 이후 각 row에서 Activate(ACT), Refresh(REF), Refresh management(RFM), Multi Purpose Command(MPC-Manual ECS)가 발생할 때마다 카운트가 증가하며, 설정된 임계치에 도달하면 DRAM은 ALERT\_n 신호를 통해 시스템에 경고를 보낸다. 이후 시스템은 RFM 명령을 통해 해당 row와 인접 row에 대한 리프레시를 수행한다[1,4]. 이 과정을 통해, 공격자가 임계치를 넘기기 직전에 접근을 차단하거나 우회 리프레시를 수행함으로써 RH로 인한 비트 플립을 방지할 수 있다. 또한 PRAC는 Adaptive Threshold 기능을 통해 임계치를 동적으로 조정할 수 있으며, 운영 환경에 따라 민감도를 조절할 수 있다[1].

### 2.2 PRAC의 성능적 제약과 보안적 취약성

PRAC는 모든 row를 카운트하여 정밀도를 높임으로써 RH 공격을 보다 효과적으로 방지할 수 있도록 설계되었지만, 구조적 특성상 몇 가지 성능 및 보안 측면의 제약을 수반한다. 우선, 모든 row에 대해 카운터를 유지관리해야 하기 때문에 하드웨어 자원이 추가로 소모되며 row 접근 시마다 카운터 업데이트로 인해 tRP, tRC, tRFM 등과 같은 DRAM 타이밍 파라미터를 늘려야 하므로 이는 전체 시스템 성능에 영향을 줄 수 있다[1]. 또한, 공격자가 실제로 RH 공격을 수행하지

않더라도 반복적인 row 접근을 유도하여 PRAC의 RFM 대응을 과도하게 유발시킬 경우, 시스템은 불필요한 리프레시로 인해 성능 저하 가능성이 있으며, 이로 인해 Denial-of-Service(DoS) 형태의 공격으로 악용될 수 있다.

따라서 PRAC는 정밀한 row 카운트 모니터링을 통해, 높은 정밀도와 신뢰성을 제공하지만, 시스템 성능에 영향을 미칠 수 있으므로 관련 연구가 필요하다.

### 2.3 PRAC가 시스템 성능에 영향을 미치는 요인

PRAC의 시스템 성능에 대한 영향은 여러 DRAM 타이밍 파라미터와 밀접한 관련이 있다. 앞서 언급한 tRP, tRC, tRFM과 같은 파라미터는 PRAC 동작으로 인해 더 늘어나야 하며, 이로 인해 메모리 접근 지연이 증가하고 전반적인 성능 저하를 유발할 수 있다.

또한, DRAM은 셀 데이터를 유지하기 위해 주기적인 리프레시가 필요하며, 온도에 따라 결정되는 리프레시 주기(tREFI)는 시스템 성능 및 PRAC의 효율성 과도 밀접하게 연결된다. tREFI가 짧을수록 단위 시간 내 더 많은 리프레시가 발생하고, 이에 따라 리프레시 처리 점유 비율이 증가하게 되어 시스템 성능에 부정적인 영향을 줄 수 있다. 반대로 tREFI가 길어지면 단일 row가 더 많은 활성화를 허용하게 되어, RH 공격 가능성이 증가하며 PRAC 동작이 더 자주 개입하게 되어 성능에 부정적인 영향을 줄 수 있다.

아울러 RH 임계치( $N_{RH}$ )는 PRAC가 공격으로 간주하는 기준선으로, 설정 값에 따라 감지 민감도 및 대응 빈도에 직접적인 영향을 미친다. 낮은  $N_{RH}$  설정은 보안성 향상에는 유리하지만, 과도한 대응으로 인한 성능 손실을 발생시킬 수 있다. 본 연구에서는 이러한 파라미터들이 PRAC 성능에 어떤 영향을 주는지 실험적으로 분석하고자 한다.

## III. 구현

### 3.1 Simulator 내 PRAC 구현 방식 및 구조 분석

실험에 앞서, 본 연구에서 사용된 DRAM 시뮬레이터인 Ramulator 2.0의 구조와 PRAC 구현 방식을 분석하였다[5]. Ramulator 2.0의 구성파일(yaml)은 그림 1과 같이 프론트엔드, 메모리 시스템 크게 2가지로 나눌 수 있고, 메모리 시스템에서 DRAM 타입, PRAC를 포함한 플러그인 및 컨트롤러를 설정한다.

PRAC 플러그인 내부를 살펴보면, 뱅크별로 정의한 Per-Bank 카운터 클래스를 통해 row 활성화를 모니터링 한다. ACT 명령이 발생할 때마다 각 row의 카운트가 증가하며, 설정 임계치를 초과하면 ALERT\_n 신호를 발생시키고 PRE\_RECOVERY 상태로 진입한

이후 RFM 명령을 통해 RECOVERY 단계로 전환된다. RECOVERY 완료 후 NORMAL상태로 복귀한다. abo\_threshold, abo\_delay\_acts, abo\_recovery\_refs, abo\_act\_ns 등의 값으로 설정 가능하며, 실험에서는  $N_{RH}$  값에 따라 이 파라미터들이 동적으로 조정된다.

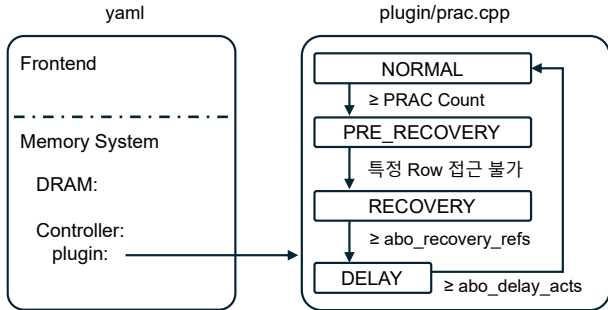


그림 1. Ramulator2.0 PRAC 구현 방식

### 3.2 평가 환경 및 조건

본 실험에서는 DRAM의 PRAC 기법이 시스템에 미치는 영향을 평가하기 위해 DRAM 시뮬레이터 Ramulator 2.0을 사용하였다[5]. 실험 환경은 최신 시스템을 참고하여 표1에 요약된 바와 같이 구성하였다. 프로세서는 4.8GHz의 클럭 속도를 가지며, 4-wide issue와 128개 엔트리 명령어 윈도우를 갖는 4코어 또는 1코어로 구성된다. Last-Level Cache는 64B cache line 크기를 가지며 총 용량은 8MB이다. 메모리 컨트롤러는 읽기/쓰기 요청 큐 각각에 대해 64개의 엔트리를 갖고, 스케줄링은 FR-FCFS 정책을 따르며, 주소 매핑은 RoBaRaCoCh 방식을 따른다. 메인 메모리는 DDR5-3200 구성으로, 1채널과 2랭크로 구성되어 있다.

표 1. 실험환경 구성

Processor	4.8GHz, 4 Cores/1 Core, 4-wide issue, 128-entry instr.
Last-Level Cache	64-byte cache line, 8MB
Memory Controller	64-entry read/write request queues; Scheduling policy: FRFCFS; Address mapping: RoBaRaCoCh
Main Memory (DRAM)	DDR5-3200, 1 channel, 2 ranks

워크로드는 SPEC CPU2006/2017의 메모리 집약적인 워크로드(H) 15개, 연산 집약적인 워크로드(L) 26개를 기반으로 워크로드 그룹을 생성한다. 이 그룹 구성은 HHHH, HHHL, HHLL, HLLL, LLLL 각 형태별로 5개씩 생성하되, 각 워크로드를 랜덤하게 조합하여 지속적으로 메모리 집약적인 형태, 혼합된 형태, 또는 연산 집약적인 형태와 같이 다양한 워크로드 조합을 실험할 수 있도록 구성하였다[7].

### 3.3 평가 방식 및 지표

시스템 성능 평가는 Weighted Speedup(WS) 지표를 사용하여 각 코어의 성능 저하 정도를 정량화 하였다.  $N_{RH}$  값을 1024부터 20까지 다양하게 변화시키며 PRAC의 민감도 조절에 따른 성능 오버헤드를 분석하였다. 참고사항으로 JEDEC 규격 기준 완화 임계치는 기본값, 기본값-10%, 20%, 30%까지 선택 가능하다.

본 실험은 단일 코어와 다중 코어 환경에서 수행되었으며, 트레이스 기반의 메모리 접근 시나리오를 활용하여 현실적인 워크로드 조건을 반영하였다. 이를 바탕으로 PRAC의 보호 효과와 성능 측면의 트레이드 오프를 함께 평가하였다.

WS 값은 Ramulator 2.0의 출력 결과 중 각 코어별 사이클 수를 기반으로, 단일 코어와 다중 코어 환경에서의 IPC를 계산하여 산출하였다. 정규화는 PRAC off 설정에서의 WS 값을 기준으로 수행하였다.

$$WS = \sum_{i=0}^3 \frac{IPC_i^{multi}}{IPC_i^{single}} \quad \text{Normalized WS} = \frac{WS_{\text{prac on}}}{WS_{\text{prac off}}}$$

또한, DDR5 및 LPDDR5과 같은 최신 DRAM 규격의 tREFI를 참고하여, tREFI별 성능 오버헤드 실험을 수행하였다[1,2].

### 3.4 실험 결과 및 유형별 성능 영향 분석

#### 1) PRAC on/off 간 성능 비교

PRAC 기법이 시스템 성능에 미치는 영향을 정량적으로 분석하기 위해, 동일한 환경에서 PRAC on/off에 따른 비교 실험을 진행하였다.

실험 결과, 그림2에서 확인할 수 있듯이, PRAC on 구성에서는  $N_{RH}$  1024부터 256까지 평균 성능 저하가 6.2%에서 6.6% 수준으로 거의 유사하게 나타났으며,  $N_{RH}$  128에서는 8.5% 수준으로 비교적 미미한 성능 감소가 관찰되었다. 그러나  $N_{RH}$  128 미만으로 작아질 경우 성능이 급격히 감소하였으며, 특히  $N_{RH}$  20에서는 약 33.4%까지 감소하는 것으로 나타났다. 이러한 결과는  $N_{RH}$  값이 작아질수록 PRAC의 개입 빈도가 증가하고 이에 따라 성능-보안 간의 트레이드오프가 존재함을 보여준다.

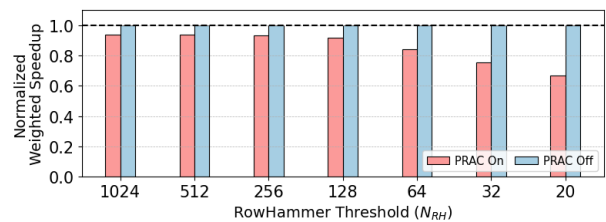


그림 2. PRAC on/off에 따른 성능 비교

이러한 점을 고려할 때, JEDEC DDR5 규격에서 선택 가능한 완화 설정 값은 기본값 대비 약 30% 이내이

며,  $N_{RH}$  128 이상 범위를 유지할 경우 PRAC 도입에 따른 성능 저하는 수용 가능하다고 판단된다.

### 2) 타이밍 파라미터별 성능 영향 비교

PRAC 동작은 각 row에 대해 활성화 횟수를 카운트하고 RFM 수행 시 대상 row 및 인접 row까지 함께 리프레시해야 한다. 이로 인해 일부 타이밍 파라미터가 변경되며, 이는 시스템 성능에 영향을 줄 수 있다. 이러한 타이밍 변화가 성능에 미치는 영향을 분석하기 위해 관련 파라미터들을 두 가지 그룹으로 분류하여 실험을 진행하였다.

첫 번째 그룹(T1)은 카운터 추가로 인한 타이밍 파라미터(tRP, tRC)이며, 두 번째 그룹(T2)은 대상 row 및 인접 row의 리프레시 동작과 관련된 tRFM1, tRFMsb로 구성된다. 실험은 총 네 가지 구성으로 수행되었으며, 각각 ①T1과 T2를 모두 적용한 Full timing(FT), ②T1만 적용한 구성, ③T2만 적용한 구성, ④모든 타이밍 딜레이를 제거한 Ideal 구성으로 나누어 비교하였다.

실험 결과, 그림3에 따르면 FT과 T1 구성의 성능은  $N_{RH}$  128 이상 구간에서 유사한 수준을 나타내며, 이는 해당 구간에서는 T1에 의한 성능 저하가 주요 원인임을 시사한다. 반면,  $N_{RH}$  128 미만으로 작아지면 T2 구성도 급격한 성능 하락을 보이는데, 이는 낮은 임계치로 인한 RECOVERY 동작이 자주 발생하고, 그에 따른 RFM 수행 빈도가 증가함에 따라 T2의 영향이 커진 결과로 해석된다.

또한, Ideal 구성은 모든 타이밍 딜레이를 제거한 상태임에도 불구하고,  $N_{RH}$  128 미만일 경우 급격한 성능 저하가 발생하였다. 이는 타이밍 파라미터와 무관하게, PRAC 자체의 오버헤드-예를 들어 RFM 빈도 증가나 스케줄링 간섭 등의 요인-에 기인한 것으로 판단된다.

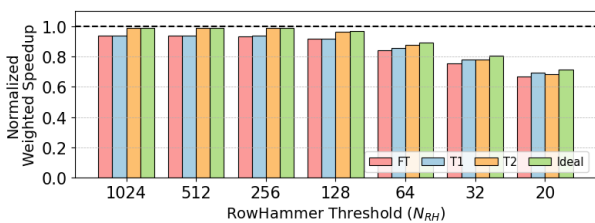


그림 3. 타이밍 파라미터별 성능 영향 비교

### 3) tREFI별 성능 영향 비교

DDR5 제품은 JEDEC 표준에 따라 32ms 내 총 8192개 row를 리프레시해야 하며, 이때 리프레시 명령 간격은 tREFI x1 (3.9us)로 정의된다. 동작 온도가 85°C 이상일 경우, tREFI x1/2 (1.95us)로 단축되어 동작한다[1]. 반면, 모바일 환경에서는 효율적인 전력 관리를 위해 LPDDR5에서 tREFI x1/8부터

x8까지 지원하며, LPDDR6에서 PRAC 기능도 추가될 예정이다[2,3]. 본 실험에서는 이러한 동작 환경을 반영하여 총 6가지 tREFI 설정(x1/8, x1/4, x1/2, x1, x4, x8)에 대해 성능 영향을 분석하였다.

실험 결과의 정규화 기준은 tREFI 자체의 영향으로 인한 결과 왜곡을 제거하기 위해, 각 tREFI 설정에 대한 PRAC off 결과를 기준값으로 설정하였다.

그림4의 실험 결과에 따르면,  $N_{RH}$  256 이상인 구간에서는 PRAC on/off 여부에 관계없이  $N_{RH}$ 에 따른 성능 변화는 거의 없는 것으로 확인하였다. 다만, 해당 구간 내에서도 tREFI 값에 따른 성능 차이는 존재하였으며, x1/2 이상의 설정에서는 평균 6.7% 감소 수준의 성능 감소가 발생한 반면, 보다 적극적인 리프레시가 이루어지는 x1/4과 x1/8에서는 각각 평균 7.7%, 10.9% 수준의 성능 저하가 확인되었다. 이러한 결과는 앞선 실험에서 확인한 타이밍 파라미터 T1과 연관되며, tREFI가 짧아질수록 단위 시간당 발생하는 tRFC가 증가하고, 여기에 PRAC on으로 인한 tRP 및 tRC 지연이 추가적으로 발생하여 성능에 영향을 미친 것으로 분석된다.

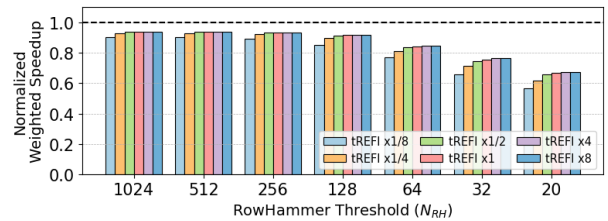


그림 4. tREFI별 성능 영향 비교

또한, 그림5에 나타난 바와 같이, 전체적인 tREFI x1/4 및 x1/8을 제외한 나머지 설정에서는 성능 감소율이 매우 유사하게 나타났으며, tREFI 값이 낮아질수록 경미하지만 꾸준한 성능 저하 경향이 관찰되었다. 이는 리프레시 간격이 짧아질수록 메모리 접근 기회의 감소와 뱅크 블로킹 현상이 누적되기 때문으로 해석할 수 있다.

본 실험을 통해 PRAC 기법이 시스템 성능에 미치는 영향에 대해  $N_{RH}$ , 타이밍 파라미터, tREFI 세 가지 측면에서 분석하였다. 실험 결과,  $N_{RH}$  128 이상 범위에서는 평균 8.5% 이내의 성능 저하로 PRAC의 적용 가능성이 확인되었으며,  $N_{RH}$ 가 낮아질수록 성능-보안 간

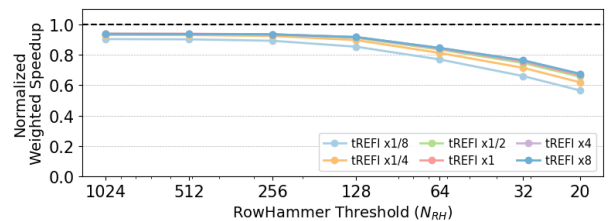


그림 5. tREFI별 성능 감소율

트레이드오프가 뚜렷하게 나타났다. 또한, tRP, tRC가 주요 성능 저하 요인으로 확인되었고, RFM은  $N_{RH}$ 가 낮을수록 성능에 점차 더 큰 영향을 미친다는 점도 실험을 통해 알 수 있었다. tREFI 관련 실험에서는 짧은 tREFI 일수록 성능 저하 경향을 확인할 수 있었는데, 이는 PRAC에 의한 타이밍 지연, 리프레시 간섭 증가와 बैं크 블로킹 누적이 복합적으로 작용한 결과로 해석된다. 이를 통해 PRAC의 성능 영향은 설정 값과 시스템 요인에 따라 달라지며, JEDEC 표준 범위 내에서는 수용 가능한 수준임을 실험적으로 확인하였다.

#### IV. 결론 및 향후 연구 방향

##### 4.1 연구 요약 및 종합 해석

본 연구는 DDR5 DRAM 환경에서의 RH 방어 기법인 PRAC의 성능 오버헤드를 정량적으로 분석하고, 실제 시스템 적용 가능성을 평가하였다. PRAC은 각 row별 활성화 횟수를 기반으로 동작하여 보안성을 강화하지만, 타이밍 지연, 리프레시 간섭 등의 부작용으로 인해 성능 저하가 발생할 수 있다. 실험 통해, PRAC의 성능 영향은 다양한 시스템 구성에 따라 상이하게 나타남을 확인하였으며, 특히  $N_{RH}$  128 이상에서는 성능 저하가 제한적이며, 실제 시스템에서 충분히 수용 가능한 수준으로 판단된다.

이러한 결과는 PRAC이 기존의 RH 대응 기법들 대비 비교적 적은 성능 저하로 충분한 보안 효과를 제공할 수 있음을 보여주며, 시스템이 보안 요구 수준과 성능 목표에 따라 PRAC 도입 여부를 유연하게 판단할 수 있다는 점에서 중요한 시사점을 갖는다.

##### 4.2 후속 연구 방향

본 연구는 Ramulator 2.0 기반의 시뮬레이션 환경에서 PRAC의 성능 오버헤드를 정량적으로 분석하였으나, 향후에는 실제 시스템과의 정확한 차이를 반영하기 위해 cycle-accurate 기반 CPU 시뮬레이터와의 연동을 통한 후속 연구가 필요하다[8]. 이러한 접근은 메모리 트래픽이 실제로 동적으로 생성되는 구조를 반영할 수 있어, 현실 기반 성능 평가가 가능하며, PRAC이 전체 시스템 동작에 미치는 영향을 보다 정밀하게 관찰할 수 있다.

또한, 본 연구는 워크로드 기반 평가에 중점을 두었으나 향후 연구에서는 의도적인 RH 공격이나 DoS 시나리오 등 다양한 위협 모델에서의 시스템 반응과 PRAC의 방어 효과 및 오버헤드 특성을 정량적으로 비교 분석함으로써, 보안성과 성능 사이의 최적 균형점을 보다 명확히 도출할 수 있을 것으로 기대된다.

## Acknowledgement

이 성과는 2025년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2021R1C1C1012172)

## 참고문헌

- [1] JEDEC, DDR5 SDRAM, JESD79-5C.01\_v1.31
- [2] JEDEC, LPDDR5 SDRAM, JESD209-5A
- [3] Murdock, Brett. LPDDR6: A Deep Dive Into the JEDEC Press Release. JEDEC Mobile / Client / AI Computing Forum, May 2024. JEDEC
- [4] Bennett, Tanj, et al. "Panopticon: A complete in-dram rowhammer mitigation." Workshop on DRAM Security (DRAMSec). Vol. 22. 2021.
- [5] Luo, Haocong, et al. "Ramulator 2.0: A modern, modular, and extensible dram simulator." IEEE Computer Architecture Letters 23.1 (2023): 112-116.
- [6] Kim, Yoongu, et al. "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors." ACM SIGARCH Computer Architecture News 42.3 (2014): 361-372.
- [7] Canpolat, Oğuzhan, et al. "Understanding the security benefits and overheads of emerging industry solutions to dram read disturbance." arXiv preprint arXiv:2406.19094 (2024).
- [8] Binkert, Nathan, et al. "The gem5 simulator." ACM SIGARCH computer architecture news 39.2 (2011): 1-7.
- [9] Bains, Kuljit, et al. Target Row Refresh Command for a Semiconductor Memory Device. U.S. Patent Application US20140059287A1, Intel Corporation, 27 Feb. 2014.
- [10] Kang, Uksong, et al. "Co-architecting controllers and DRAM to enhance DRAM process scaling." The memory forum. Vol. 14. 2014.