JULY 9–13, 2023

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

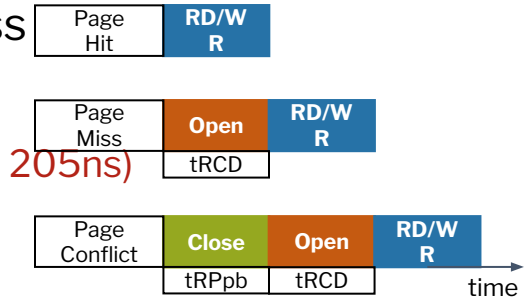
PBR Bank Conflict Avoidance Scheduling to enhance DRAM bank utilization

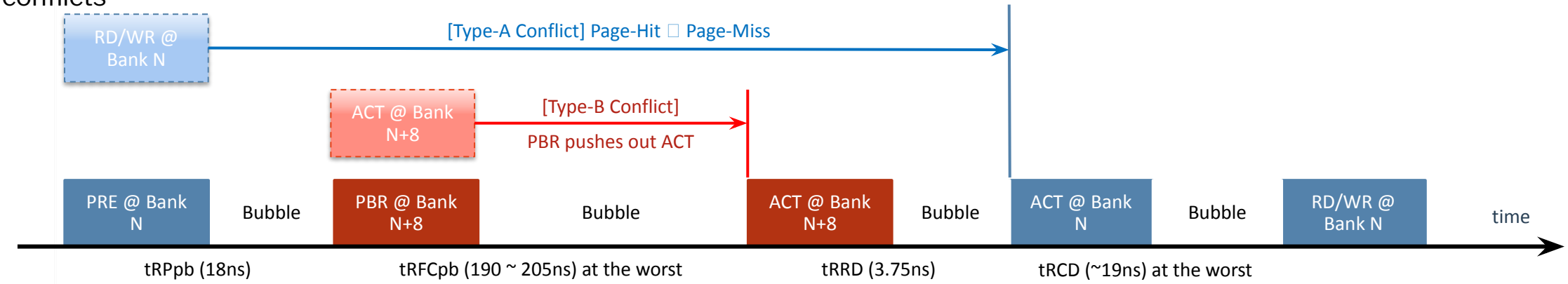
Hyunjoon Kang, Heeju Kim, Taehun Kim, Manhwee Jo, and Wooil Kim

Innovative AP Development Team, System LSI Business, Samsung Electronics Co., Ltd

Motivation

- Performance degradation by Per-Bank Refresh (PBR) interventions
  - PBR (Per-Bank Refresh)
    - Enhanced DRAM refresh scheme (compared to All Bank Refresh) refreshing DRAM cells at the individual bank level
    - DRAM performance can be degraded if a DRAM operation requires the bank that PBR operations are in progress
  - PBR bank can be conflicted with the bank where normal DRAM traffic is about to access
    - [Conflict Type-A] Utilization degradation □ Page-Hit is changed to Page-Miss in DRAM
    - [Conflict Type-B] Latency degradation □ Unloaded latency can be increased up to tRFCpb (190ns ~ 205ns)
  - PBR bank confliction can occur more frequently at higher refresh rate
    - Automotive can be exposed to high ambient temperature, where DRAM requires high refresh rate, leading to more PBR conflicts





Implementation of PBR Bank Selection. Design Flow

. If no PBR conflicts are expected,

- Select bank by **in-order** (B0 □ B1 □ B2 □ … □ B7)

. If PBR conflicts are expected,

- Select bank by **priority order** on the right
- Remove the PBR conflicts degrading DRAM performance

. Priority 1

- Select bank
  : **Not in request queue** with DRAM **page closed**

. Priority 2

- Select bank
  : **In request queue** with DRAM **page closed**
  : of **Lower scheduling urgency** and **Younger age** request

. Priority 3

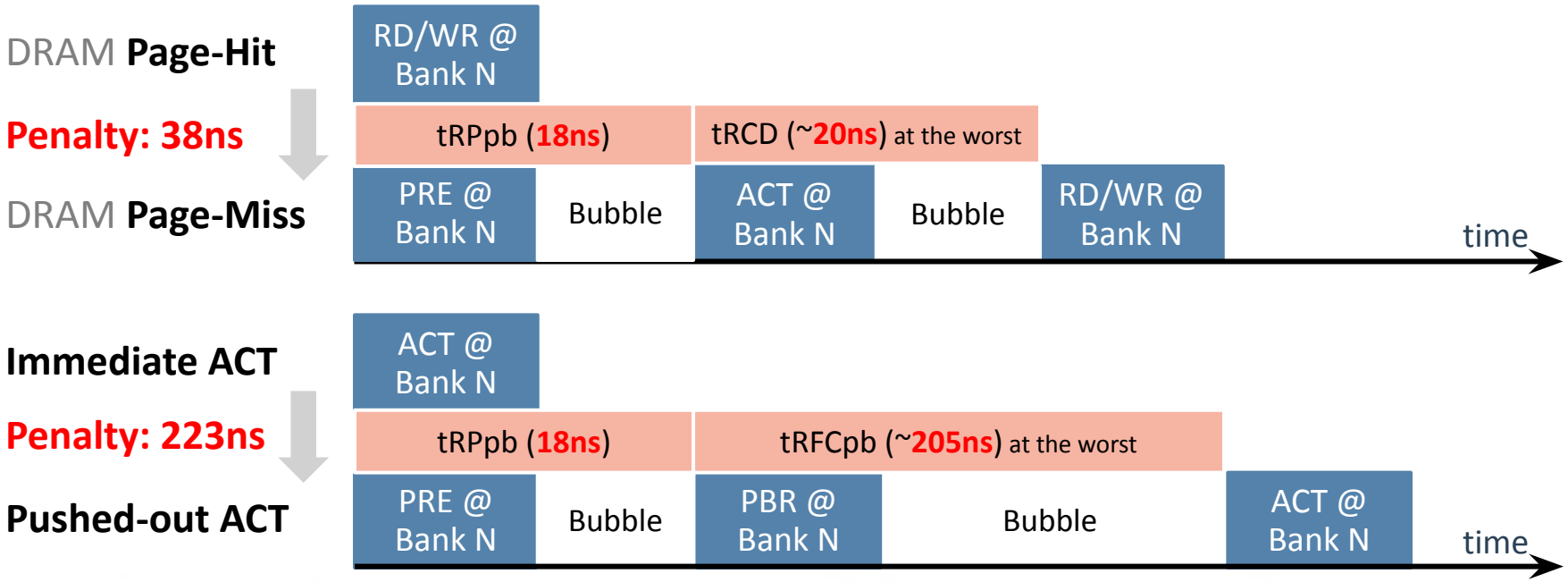
- Select bank
  : **In request queue** with DRAM **page conflicted without page-hit**
  : of **Lower scheduling urgency** and **Younger age** request

. Priority 4

- **Postpone** (one~eight) **PBR**
- And release it following Priority1~3

[JEDEC] DRAM Performance Penalty

- Performance degradation by DRAM Core Timing Parameter
  - tRCD (~20ns): RAS-to-CAS delay
  - tRPpb (18ns): Row precharge time (single bank)
  - tRFCpb (~205ns): Refresh cycle time (Per bank)



Evaluation

- Enhancement of DRAM utilization for different benchmarks / refresh rates
  - DRAM utilization enhancement up to 8.54% (PBR Bank is selected by Priority 1 ~ Priority 4)

	Memory Utilization (gbv4)		
DRAM Temp PBR Bank Selection	1x	1/2x	1/4x
in-order	74.41 [%]	71.70 [%]	62.70 [%]
Priority-order	75.65 [%]	72.69 [%]	64.66 [%]
Gain	1.67 [%]	1.38 [%]	3.13 [%]

	Memory Utilization (aztf210_peak)			Memory Utilization (aztf210_sustain)		
DRAM Temp PBR Bank Selection	1x	1/2x	1/4x	1x	1/2x	1/4x
in-order	54.15 [%]	48.10 [%]	34.88 [%]	54.04 [%]	48.15 [%]	34.78 [%]
Priority-order	54.29 [%]	48.72 [%]	37.72 [%]	54.20 [%]	48.62 [%]	37.75 [%]
Gain	0.26 [%]	1.29 [%]	8.14 [%]	0.30 [%]	0.98 [%]	8.54 [%]

	Memory Utilization (mhtf131_peak)			Memory Utilization (mhtf131_sustain)		
DRAM Temp PBR Bank Selection	1x	1/2x	1/4x	1x	1/2x	1/4x
in-order	67.58 [%]	62.28 [%]	49.11 [%]	67.65 [%]	62.24 [%]	49.49 [%]
Priority-order	67.97 [%]	63.39 [%]	52.37 [%]	68.16 [%]	63.50 [%]	52.70 [%]
Gain	0.58 [%]	1.78 [%]	6.64 [%]	0.75 [%]	2.02 [%]	6.49 [%]

Summary

Background

- Performance degradation by Per-Bank Refresh (PBR) interventions
  - ✔ Two concepts of PBR bank conflict affect on both DRAM utilization and latency
  - ✔ PBR needs to be adjusted in its bank selection to minimize DRAM performance degradation

Suggestions

- We can select PBR Bank in accordance with the suggested priorities
  - ✔ Priority 1: Bank not in DRAM controller's request queue w/ DRAM page closed
  - ✔ Priority 2: Bank in DRAM controller's request queue w/ DRAM page closed
  - ✔ Priority 3: Bank in DRAM controller's request queue w/ DRAM page conflicted
  - ✔ Priority 4: Postpone PBR if no Bank is selected above, and release it I.A.W Priority 1~3

Conclusion

- We found that PBR conflict was decreased, resulting in DRAM utilization gain by 8.5% at 0.25x refresh rate with the suggested PBR Bank selection algorithm (PBCAS) in GPU sustainable operating scenario.

Biography (Hyunjoon Kang)

- Hyunjoon Kang is a senior engineer in the System LSI Division of Samsung Electronics. He has been architecting and designing AP DRAM controllers since 2016. His ongoing interest is how to control the DRAM controller to improve AP's performance/power when operating LPDDR DRAM in refresh/low power perspective.