

# A case for Refresh Pausing in DRAM memory systems

著者： Prashant Nair, Chia-Chen Chou, Moinuddin Qureshi.

出典： *High Performance Computer Architecture, 2013 IEEE 19th International Symposium on. 23-27*

発表者： 1553002 石原雅也

## 1 はじめに

ここ数十年の間、メインメモリとして使用されているDRAM(Dynamic Random Access Memory)は、SRAMなどと比べ構造が単純なため、安価で大容量なメモリとして用いられている。一方でDRAMは、構造上常にリフレッシュを続けなければならない、これによってリードレイテンシの増加やパフォーマンスの低下などが引き起こされることが課題となっている。

本稿では、リードレイテンシを抑えパフォーマンス向上をはかるために新しいDRAMのリフレッシュ方法を提案する。また、従来のDRAMのリフレッシュ方法とのパフォーマンスの比較評価を行い、新しいリフレッシュ方法の有効性を証明する。

## 2 研究背景

### 2.1 DRAMの構造

DRAMチップには数百万個のセルが配置されており、DRAMセルは1個のトランジスタと1個のコンデンサから構成されている。このコンデンサは電荷を一時的に蓄える働きがあり、そこで蓄えられた電荷によってデータを記憶する仕組みになっている。電荷があれば“1”、なければ“0”となる。ただ、コンデンサに蓄えられた電荷は、放っておくと時間とともに漏れてなくなってしまうため必然的にデータも失われてしまう。そこでDRAMは、データの整合性を維持するために、電荷が完全にならないうちに充電を繰り返し復元することによって記憶を保持している。

### 2.2 従来のリフレッシュの方法

DRAMチップには、たくさんのDRAMセルの集まったメモリバンクが存在する。バンクは、いくつかの行で構成されており、リフレッシュは基本的にこの行単位で行われていく。このリフレッシュの方法として、バーストリフレッシュと分散リフレッシュが挙げられる。

#### 2.2.1 バーストリフレッシュ

バーストリフレッシュは、リフレッシュ時間内(JEDECが規定した64ms)[1]に、全バンクの全ワードに対して一

度にリフレッシュを行う方法である(図1)。リフレッシュ時間からリフレッシュに要した時間を引いた期間だけ、リフレッシュによる中断などのない通常のメモリ動作を行うことができる。

リフレッシュパルス

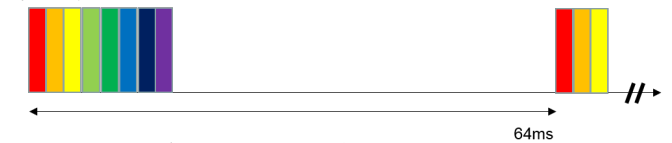


図1 バーストリフレッシュ

#### 2.2.2 分散リフレッシュ

一方で、分散リフレッシュは、規定されたりフレッシュ時間を周期ごとに分け(リフレッシュパルス)、全てのバンクで一行ずつ一緒にリフレッシュしていく方法になる(図2)。これは、リフレッシュパルス間に生じる時間に通常のメモリ動作を行えるため、リクエストとの競合を減少させることができる。

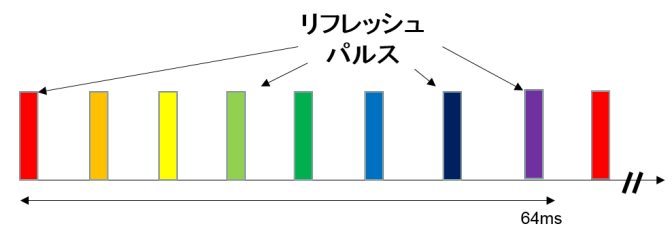


図2 分散リフレッシュ

## 3 新しいリフレッシュの方法

リフレッシュにあたり、リードレイテンシの増加とパフォーマンスの低下は常に存在する。本論文ではリフレッシュにより生じるリードレイテンシの増加とパフォーマンスの低下を抑制する方法として、リフレッシュの一時停止を提案している。これは、周期的に行われているリフレッシュに割り込んでリクエストを消化する方法で、リフレッシュ中にリクエストが来ると、その時リフレッシュを行っていた行が終了したのち、次の行へと進まずに行のアドレスを記録したあと一旦停止して、その間にリクエストを実行する。その後、記録されたアドレスを元に次の行からリフレッシュを再開する。こうすることで、リフレッシュの周期が終わるまでに生じるレイテンシを短縮する

ことができるようになる。ただ、好きなだけ一時停止が行えてしまうと、頻繁に一時停止が起きることでリフレッシュの停滞が起り、データの損失をまねいてしまう可能性がある。そのため、規定時間を超えた場合、一時停止ができない強制的なリフレッシュを行う。

## 4 実装方法

今回の論文では、ハードウェアへの変更を最小限にするため、メモリーコントローラーと DRAM へ変更を行った。

DRAM には、アドレスをインクリメントする、つまり次にリフレッシュする行を指す RAC(行アドレスカウンタ)と、リフレッシュイネーブルパルスを受信して RAC をインクリメントするインクリメンタが存在する。この RAC によるインクリメントを、単純な AND ゲートを用いて停止させることで、リフレッシュを一時的に停止させている。

メモリーコントローラは、リフレッシュイネーブルの信号を用いてアサートされることで DRAM チップをリフレッシュすることができる。この信号経路を利用し、リフレッシュイネーブルのアクティブローによってリフレッシュを中断させる。一時停止している場合、リフレッシュイネーブルはディアサートし、リードをサービスする。リードが完了したのち、リフレッシュイネーブルは再びアサートされ、リフレッシュは再開される。また、メモリーコントローラ内のスケジューラによって常にリフレッシュの状況を追跡し、一時停止される前に行われていたリフレッシュを記録することで、リフレッシュのスケジュール管理を行っている。

## 5 評価

本論文では、リフレッシュの一時停止の有用性を検証するため、シミュレータに uSIMM[2]、ワークロードに MSC Suite[3](共に MSC 出典)を用いて、表 1 の内容で、リードレイテンシ(図 3)、パフォーマンス(図 4)について従来のリフレッシュ方法との比較評価を行った。また、チップ密度ごと(図 5)についても評価を行った。

表 1 使用したメモリスシステム

Number of Cores	4
Last Level Cache	1MB
DRAM (DDR3)	8 Chips/Rank, 8Gb/Chip
Channels, Ranks, Banks	4,2,8
Refresh (Baseline)	Distributed (JEDEC)

リードレイテンシは今回使用した全てのワークロードで短縮される結果となった。また、平均して約 7% の短縮を可能とした。パフォーマンスについても、従来のリフレッシュ方法と比べ全てのワークロードで向上するという結果となった。また、平均して約 5% のパフォーマンスの向上を可能とした。チップ密度ごとにパフォーマンスを比

べた場合、密度増加に伴いパフォーマンスは向上する結果となった。これによりチップ密度が増加した場合も有用であることがわかった。

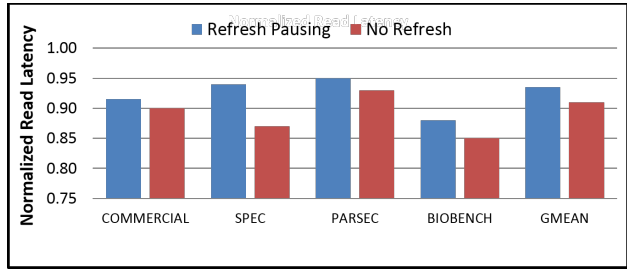


図 3 リードレイテンシについての評価

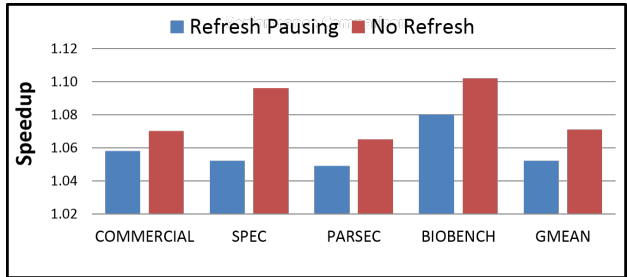


図 4 パフォーマンスについての評価

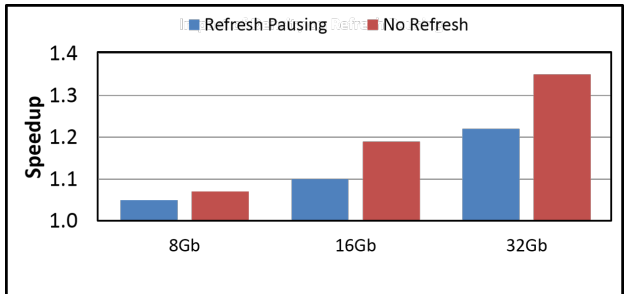


図 5 メモリ密度ごとのパフォーマンス評価

## 6 まとめ

DRAM はリフレッシュが必要である。しかし、リフレッシュにあたりリードレイテンシの増加とパフォーマンスの低下は常に存在する。その問題を改善するために、リフレッシュの一時停止を提案した。これにより、従来の方法に比べ約 7% のリードレイテンシの短縮、約 5% のパフォーマンスの向上が可能となった。

## 参考文献

- [1] JESD79-3F, JEDEC Committee JC-42.3 Std. DDR3, 2010.
- [2] N. Chatterjee, R. Balasubramonian, M. Shevgoor, S. Pugsley, A. Udipi, A. Shafiee, K. Sudan, M. Awasthi, and Z. Chishti, "USIMM: the Utah Simulated Memory Module," University of Utah, Tech. Rep., 2012, uUCS-12-002.
- [3] (2012) Memory scheduling championship (msc). [Online]. Available: <http://www.cs.utah.edu/rajeev/jwac12/>