2014 年 4 月 2 日 情報システム基盤学専攻修士論文中間発表会発表資料

MissMap を利用した DRAM キャッシュの省電力化に関する研究

所属： 高性能コンピューティング学講座 本多研究室
発表者： 1353006 金東賢
主任指導教員： 本多弘樹

1 はじめに

近年、回路ブロックを立体的に配置する三次元積層技術を前提としたプロセッサ・アーキテクチャに関する研究が盛んである。その中でも、異なる製造プロセスで作成されるプロセッサ・ダイと DRAM ダイを積層し、DRAM をキャッシュメモリとして使用する DRAM キャッシュは大容量のオンチップメモリを実現できるため、注目されている。しかし、DRAM はデータ保持にリフレッシュ動作が必要で、SRAM に比べ待機電力が増加してしまうことが問題である。

そこで本研究では、将来の CPU アーキテクチャに DRAM キャッシュが使用された場合、平均メモリアクセス時間の低減に使用されると考えられる MissMap 機構 [1] による消費電力の削減手法を提案する。MissMap 機構がキャッシュの使用領域を把握できる特性を持つことを利用し、データが格納されていない領域のリフレッシュ動作を停止することで消費電力を削減する。

2 背景

2.1 DRAM キャッシュの問題点

現在、CPU のキャッシュメモリには SRAM が使用されており、特徴としてアクセス速度は高速であるが、実装密度を上げにくい。これに対し、DRAM はアクセス速度は低速だが、高い実装密度が得られる特徴がある。このため、DRAM キャッシュを実装する場合、SRAM を利用したキャッシュと比べ、容量性によるキャッシュミスを削減出来るメリットがある。一方で、アクセスレイテンシが増加してしまうデメリットがあり、特に、キャッシュミスした際のレイテンシは深刻である。このため、平均メモリアクセス時間の増加を隠蔽できる程のキャッシュミス率の削減が達成できない場合、SRAM を使用したキャッシュに比べ、性能が低下してしまう。

2.2 MissMap 機構

DRAM キャッシュを使用した際の平均メモリアクセス時間の削減手法として、DRAM キャッシュにラインが無いと判断できる場合は DRAM キャッシュへのアクセスを

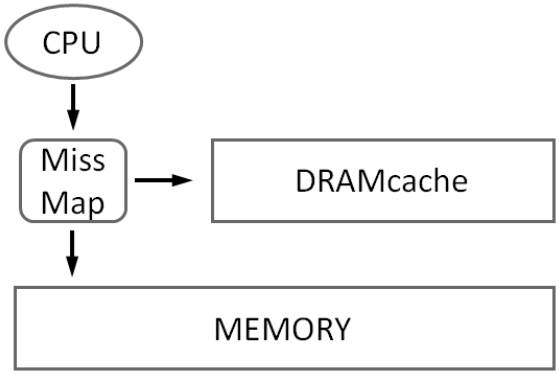

図 1: MissMap 機構

行わず、直接メインメモリへアクセスを行う MissMap 機構が提案されている。MissMap 機構を図 1 に示す。

MissMap はラインの格納情報の一部を格納することでラインの格納状態を把握する。DRAM キャッシュへアクセスの際は、まずアクセスしたアドレス情報と MissMap 機構のライン格納情報を比較し、キャッシュミスすると判断できる場合は、キャッシュアクセスをスキップし、直接メインメモリへアクセスすることでキャッシュミスペナルティを削減する。

MissMap 機構の動作例を図 2 に示す。図 2 の右側は DRAM キャッシュ、左側はそれに対応する MissMap 機構を示しており、インデックス情報に対するフラグを格納する。ここでは例として、DRAM キャッシュのインデックス 001 にラインが格納され、対応する MissMap のインデックスのフラグが 1 になった場合を示している。

CPU が DRAM キャッシュへアクセスする際、まず MissMap のインデックスを参照し、フラグが 1 の場合は DRAM キャッシュへアクセス、0 の場合はメインメモリ

MissMap	
flag	Index
0	000
1	001
0	010
0	011

DRAMCache		
Index	Tag	
000		
001		
010		
011		

図 2: MissMap 機構の動作

へ直接アクセスすることでアクセスレイテンシを低減する。図 2 では MissMap 機構のインデックス 001 のフラグが 1 であるので、DRAM キャッシュへのアクセスしたラインが 001 の場合のみ DRAM キャッシュへアクセスを行い、キャッシュミスした場合はメインメモリへアクセスを行う。

このように、フラグが 0 のインデックスにはラインが格納されていないと判断できるため、確実にキャッシュミスすると判断できる場合の無駄な DRAM キャッシュアクセスを削減できる。

3 提案手法

本研究では、DRAM キャッシュにおいて、ラインの格納されていない領域に対するリフレッシュ動作を停止することで、無駄な消費電力を削減する手法を提案する。

ラインの格納されていない領域の把握に MissMap 機構を利用する。MissMap 機構を使うメリットとして MissMap 機構はインデックス情報を保持しているため、データの格納領域の把握に新たなハードウェアの追加が必要無いことと、キャッシュの領域をインデックスごとにまとめて扱うことが出来、リフレッシュ動作を停止するためのハードウェア容量の増加を防ぐことができることが挙げられる。

理想的には、それぞれのラインに対し、個別にリフレッシュ動作が必要かどうかを判定することが望まれるが、キャッシュ容量が膨大になった場合、かえってリフレッシュ動作を制御するハードウェアや電力が増加してしまう。また、メモリアクセスには空間的局所性があるため、ある程度領域を区切ってリフレッシュ動作が必要か判断するほうが効率が良い。図 2 を例として説明すると、フラグが 0 の領域と対応しているインデックス 000、010、011 の領域のリフレッシュ動作を停止することで消費電力の削減を行うことになる。

4 関連研究

DRAM のリフレッシュ動作の削減手法として、Partial Array Self-Refresh(PASR) が提案されている [2]。この手法では、DRAM のどの領域にデータを格納するか選択し、データの格納されていない領域に対してリフレッシュ動作を停止することで、消費電力の削減を行う。この手法では、データ格納領域を決定しなければならないが、提案手法では、データ格納領域はインデックス情報から判断できるため、DRAM をキャッシュとした場合には有用な手法であると考えられる。

5 進捗状況

予備評価として、アプリケーションごとに適切なキャッシュ容量を推定するためにキャッシュの容量を変化させた場合のキャッシュミスの変化について調査した。コア数は 1、L1 キャッシュの連想度を 2、i-cache=64KB、d-cache=32KB、L2 キャッシュの連想度を 8 とし、アプリケーションを一つ実行した際の一億命令当たりのキャッシュミス回数を測定した。測定結果を図 3 に示す。

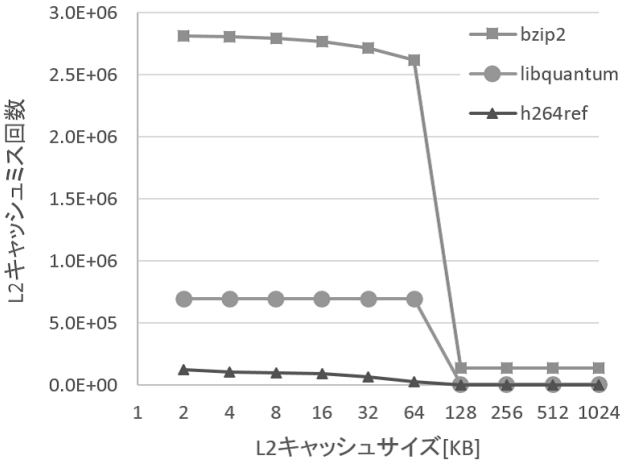

図 3: L2 サイズとキャッシュミス回数の関係

測定結果より、測定に使用したアプリケーションは L2 キャッシュ容量が 128 KB から大幅にキャッシュミス回数が減少しており、ワーキングサイズは 128 KB 程度であると考えられる。このようなアプリケーションに対しては、128 KB 程度のキャッシュ容量を用意することで高い IPC が得られると考えられる。また、128 KB より小さなワーキングサイズをもつアプリケーションに対しては、提案手法により、使用されていない領域に対し、リフレッシュ動作を停止することでキャッシュ容量に合わせて消費電力を削減できるのではないかと考えられる。

6 おわりに

DRAM をキャッシュとする DRAM キャッシュの省電力手法を提案した。本研究では、DRAM キャッシュに特有の MissMap 機構が DRAM キャッシュのラインの格納情報を保持していることを利用し、ラインが格納されていない領域のリフレッシュを停止することで消費電力の削減を目指す。

今後の方針として、CPU シミュレータに MissMap 機構を実装し、提案手法の有用性を調査すると共に、MissMap を実装したときの最適なキャッシュ構成を検討する。

参考文献

[1] Moinuddin K. Qureshi, Gabriel H, Loh, “Fundamental Latency Trade-offs in Architecting DRAM Caches,” MICRO, pp.235-246, 2012.

[2] Y. Riho, K. Nakazato, “Partial Access Mode: New Method for Reducing Power Consumption of Dynamic Random Access Memory,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., pp.1-9, 2013.