

# FPGA システムにおけるデータ圧縮を活用した Skewed DRAM キャッシュ

大谷 伸吾<sup>†</sup> 小川 愛理<sup>‡</sup> 吉瀬 謙二<sup>‡</sup>

<sup>†</sup>東京工業大学 情報工学科

<sup>‡</sup>東京工業大学 大学院情報理工学研究所

## 1 はじめに

近年、FPGA 評価ボードは内部 RAM とは別に、様々な外部メモリアンタフェースを備えるようになってきている。一般的に評価ボードには DRAM が搭載され、FPGA にソフトプロセッサを実装した際にはメインメモリとして使用することも出来る。しかしこれらは通常のコンピュータシステムと比べ容量が小さく、ボード備え付けで交換できない場合が多い。従って評価ボード上で動かすことが出来るアプリケーションがメモリ容量によって制約されてしまう。

我々は DRAM に入るデータを圧縮することでこの制約を緩和できないか検討した。データ圧縮を行うとデータ長が可変になり、その配置や管理が問題になる。そこで、我々は DRAM に圧縮キャッシュのデータ配置・管理の方法に着目した。これを元に、容量を擬似的に増大させることでメモリ制約を緩和する FPGA システムを提案し、実現に向けた検討を行う。本稿ではそのアプローチについて述べる。

## 2 圧縮キャッシュ

### 2.1 Skewed compressed cache

圧縮キャッシュにおいて問題になるのは、データが可変長になることである。エンタリは通常固定幅であるため、データ圧縮によって得られたエンタリの余剰部を使用するためには配置に関する情報が追加が必要になる。Skewed Compressed Cache[1](以下 SCC)はこの問題に着目した圧縮キャッシュである。連続するキャッシュラインは似た圧縮率を示すという局所性を利用し、複数個のキャッシュラインで SuperBlock という単位を構成する。それらを同一のタグで管理することで、なるべく少ない配置情報でエンタリの有効活用を図っている。

### 2.2 圧縮アルゴリズム

キャッシュに適した圧縮アルゴリズムは、ハードウェア化された際に以下の 2 つの重要な要件を満たす必要がある。

- 圧縮・解凍にかかるサイクル数が少ない
- 消費電力、回路規模が小さい

Skewed and Compressed DRAM Cache on FPGA Systems

Shingo OHYA<sup>†</sup>, Eri OGAWA<sup>‡</sup>, and Kenji KISE<sup>‡</sup>

<sup>†</sup>Department of Computer Science

Tokyo Institute of Technology

<sup>‡</sup>Graduate School of Information Science and Engineering

Tokyo Institute of Technology

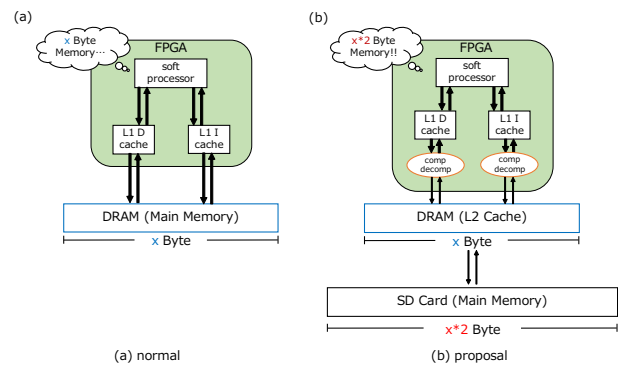


図 1: (a) ソフトプロセッサ搭載 FPGA システムの一般的な構成と (b) 提案する FPGA システムの構成

上記要件を満たす圧縮アルゴリズムとして SCC では C-Pack[2] を採用している。パターンマッチングと辞書式を組み合わせた圧縮手法で、ハードウェア化した際にパイプライン処理を行うことが出来るためレイテンシが少ない。また回路規模も小さくなるように設計されている。

## 3 提案システム

図 1 に一般的なソフトプロセッサ搭載システムと提案するシステムの構成図を示す。我々が提案するシステムでは、DRAM をメインメモリではなく大容量キャッシュとして利用し、SD カードをメインメモリとして扱う。この際、FPGA から DRAM に書き込まれるデータ(キャッシュライン)を圧縮することでその量を削減する。DRAM はキャッシュであるためデータの追い出しが起こりうるが、SD カードの一部に DRAM の 2 倍のアドレス空間を割り当て、こちらに追い出されたデータを保存することでこの問題を解決する。しかし、SD カードへの書き込みには膨大なサイクル数を必要とするため、頻繁に SD カードへのアクセスが起こるようなシステムは現実的でない。このようなアクセスを極力避けるためには、圧縮によって短くなったデータをエンタリに無駄なく格納する工夫が必要である。

そこで、我々は先に述べた SCC の手法を元に配置・管理を行い、C-Pack を圧縮アルゴリズムとして DRAM に適用し本システムの実現を目指す。

## 4 シミュレータへの実装と評価

我々は、前章で紹介したシステムを実機に実装する前段階として、シミュレーションを行った。

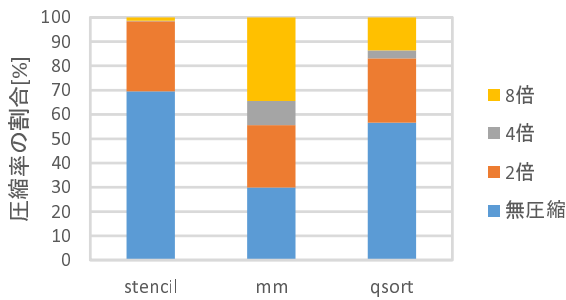


図 2: 各アプリケーションにおける圧縮率の割合

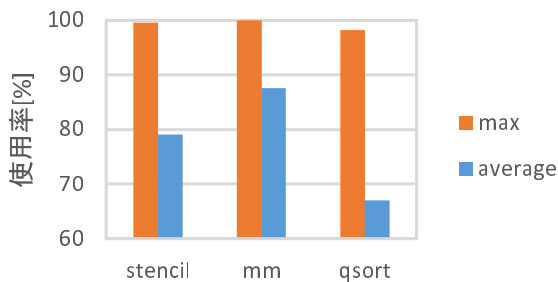


図 3: 各アプリケーションのエントリ使用率

今回、我々はシミュレータ環境として SimMips[3] を採用した。SimMips は、MIPS32 命令セットのプロセッサを含む C++ で記述されたシステムシミュレータである。メモレイテンシを考慮していないという欠点があるが、比較的シンプルな記述でありシステム全体のデータパスを監視しやすいという利点がある。我々は本シミュレータに L1 データキャッシュ 2KB、L1 命令キャッシュ 2KB、L2 キャッシュ 512KB、圧縮解凍ブロックを実装し、メインメモリを L2 キャッシュの 2 倍である 1MB として提案システムの動作をシミュレートした。

提案システムの初期検討として、メモリアクセスの多いアプリケーションであるステンシル計算 (256 × 256 行列)、行列積 (256 × 256 行列)、クイックソート (2<sup>17</sup> 要素) の 3 種類を評価した。なお初期データはすべて圧縮し、DRAM キャッシュに配置してから実行している。

## 5 考察

図 2 に、各アプリケーションを実行した際に DRAM に書き込まれる 64B キャッシュラインを圧縮した際の圧縮率の割合を示す。SCC では 2 の冪乗ごとに圧縮率を区別するため、そのように分類している。また、図 3 に各アプリケーションごとの L2 DRAM キャッシュのエントリ使用率、図 4 に各アプリケーションごとの L2 DRAM キャッシュのミス率を示す。

図 2 を見ると、mm ではキャッシュに格納されるデータの 7 割ほどが 2 倍以上の圧縮率となっており、図 3 では、平均で 9 割近い使用率を示している。また図 4 から、キャッシュミスの発生が少なく、SD カードへのアクセス回数を抑えているといえる。

stencil では無圧縮の占める割合が高いが、ミス率は

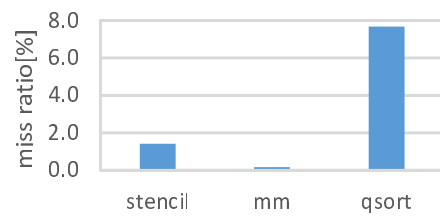


図 4: 各アプリケーションのキャッシュミス率

2%弱となっており、エントリも平均で 8 割程度使用しているという結果が得られた。

しかし一方で、図 4 からわかるように、qsort では他の 2 つのアプリケーションと比べミス率が大幅に増加してしまっている。さらに、ミス率と同様にエントリの平均使用率も他のアプリケーションと比べ低いという結果が得られている。

このように、アプリケーションに依存するものの、我々の提案するシステムは高い充填率で圧縮キャッシュラインをエントリに収め、SD カードへのアクセス回数を抑えているといえる。従って、メモリ容量を擬似的に増大させメモリ制約を緩和する、という目的を達成できると考える。

しかし、現在の実装では qsort のように、エントリの使用率が低いにもかかわらずミス率が高くなるケースが存在している。システムの有用性をさらに高めるためには、エントリをもっと有効に使えるよう、配置手法にも手を加えていくことが望ましいと思われる。また、本システムではキャッシュラインの圧縮率が高くなければ目的である容量増加を達成できない。今後様々な圧縮アルゴリズムを適用し、圧縮率がどう変化するかを、シミュレータで評価していくことが望ましいと考えられる。

## 6 まとめ

我々は、メモリ容量をデータ圧縮によって擬似的に増大させる FPGA システムを提案し、その初期検討として SimMips 上で評価を行った。その結果から、提案システムは既存システムと比べ容量の点において優れており、実現の可能性が十分にあると分かった。

今後の課題としては、圧縮解凍のレイテンシを考慮したシミュレーションを行うこと、前章で示した改善案を適用すること、それを元に実際に FPGA ボードにシステムを実装することが挙げられる。

## 参考文献

- [1] Somayeh Sardashti, André Seznez, and David A. Wood. Skewed compressed caches. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-47*, pages 331–342, Washington, DC, USA, 2014. IEEE Computer Society.
- [2] Xi Chen, Lei Yang, Robert P. Dick, Li Shang, and Haris Lekatsas. C-pack: A high-performance microprocessor cache compression algorithm. *IEEE Transaction on Very Large Scale Integration (VLSI) Systems.*, 18(8):1196–1208, aug 2010.
- [3] Naoki Fujieda, Takefumi Miyoshi, and Kenji Kise. Sim-mips: A mips system simulator. In *Proceedings of the 2009 Workshop on Computer architecture education*. New York City. Citeseer, 2009.