

## 2-ポートメモリと並列処理を用いたハードウェアソータ

安田尚樹<sup>†</sup>, 宮崎 敏明<sup>†</sup>  
<sup>†</sup>会津大学コンピュータ理工学部

### 1 はじめに

様々なアプリケーションで必要とされている大量データの高速ソーティングを目的としたハードウェアソータが幾つか提案されている[1][2]。本稿では、文献[1]に提案されているマージソータを基本としたハードウェアソータにデータの読み書きが同時に行える 2-ポートメモリを導入すると共に、2個のマージソータ回路を並列動作させることにより、ソーティングのスループットの改善を試みたので報告する。

### 2 提案システム

図1に提案アーキテクチャの概要を示す。文献[1]で提案されているアーキテクチャと同様に、16入力 16出力のソーティングネットワーク回路と8入力マージソータ回路からなる。図2は、ソーティングネットワーク回路の詳細である。ソーティングネットワーク回路は、本来組み合わせ回路で実現できるが、動作クロックを上げるために、11段のパイプライン化を図っている。1データの bit 幅は32bit である。また、マージソータ回路は、8ポート(8-way)マージソートツリーを持つ。さらに、当該マージソートツリーは、1ポート当たり4データ(128bit)を同時に入力でき、同時に4データごとソート結果を出力する。また、マージソータ回路の各入力には入力 FIFO バッファ(以下単に、入力バッファ)が接続されている。入力バッファの入力側は、マルチプレクサを介してソーティングネットワーク回路に接続されており、16データごとにソートされた計 512bit(=16×32bit)を一気に1ブロックとして入力できる。一方、出力側は、マージソータ回路に4個ずつデータを供給するために、512bit のシフトレジスタを設け、128bit(=32bit×4データ)ごとシフトしつつ、マージソータ回路へデータを供給する。

上記に加え、2-ポート DRAM メモリを導入し、ソーティングネットワーク回路へのデータ読み出しと、ソート結果を一時格納するために設けた出力 FIFO バッファ(以下単に、出力バッファ)からのデータ書き込みを同時に行えるようにしている。

ソーティングネットワーク回路の 512bit の出力は、マルチプレクサを介して、8本の入力バッファの内、最もデータ蓄積量が少ない入力バッファに入力される様に制御され、マージソートツリーの8入

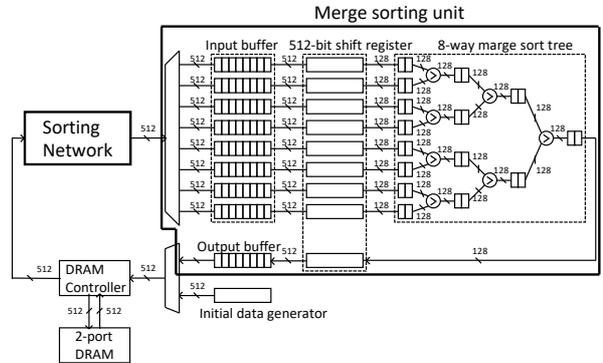


図1 提案アーキテクチャ概要

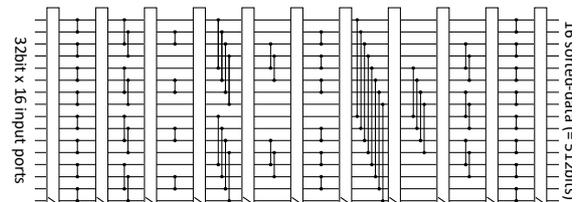


図2 パイプライン化ソーティングネットワーク回路

力ポートに均等にデータが入力されるようにしている。各回路の具体的な動作は、文献[1]と同様である。

なお、直接データのソーティングには関係しないが、評価用に 2-ポート DRAM メモリへの初期データの生成用に、初期データ生成回路を設けた。本初期データ生成回路は、最大値を与えると、その最大値から0に向かって、1減じた 32bit 整数値を出力する。

#### 2.1 マージソータ回路を2つ設けた構成

スループットを向上させるために、図1に示したマージソータ回路を2つ設け、それらを同時に動作させることでスループットの向上を図る試みも行った。図3に、その回路構成を示す。ソーティングネットワーク回路により得られたソートされた 16データ(512bit)は、マルチプレクサを介して、マージソータ回路#1またはマージソータ回路#2に交互に入力される。

### 3 評価

提案したハードウェアソータの性能を確認するために、入力データ数を変化させ、その時間を計測した。入力データは全て、最大値から0まで、1ずつ減じたデータが降順に並んだものであり、それを昇順にソートする時間を計測した。提案ハードウェア

A Hardware Sorter Using 2-port Memory and Parallel Processing  
 Naoki Yasuta<sup>†</sup>, and Toshiaki Miyazaki<sup>†</sup>  
<sup>†</sup>School of Computer Science and Engineering, the University of Aizu, Japan

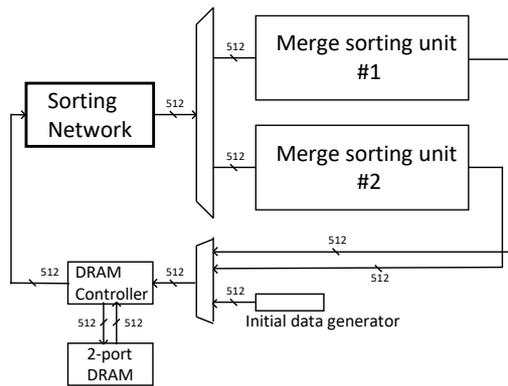


図3 マージソート回路を2つ設けた構成

ソータは、最終的に FPGA に実装することを想定している。今回は、Verilog-HDL で回路を記述し、RTL シミュレータによりソート終了までにかかるクロック数を計測し、クロック周波数が 100MHz で動作するものとして、終了時間を算出した。また、比較のために、ソフトウェアによるマージソートおよびクイックソートを PC(Intel core i7-3770@3.40GHz, Memory: 8GB, OS: Ubuntu 16.10) 上で実行し、その実行時間も実測した。

結果を表 1 に示す。表中、[A]および[B]は、それぞれマージソートおよびクイックソートを PC 上で実行した時間である。[C]は、文献[1]の回路、すなわち 8-way マージソート回路を 1つ使用したハードウェアソータの実行結果である。[D]は、図 3 に示した 8-way マージソート回路を 2つ使用した構成であるが、2-ポート RDAM メモリを使用していない。[E]は、図 3 に示した回路構成の実行結果である。表中”-“と記した部分は、原稿執筆段階で、当該条件での計測が終了していないことを表す。[B]/[D]は、ソフトウェアによるクイックソートと 8-way マージソート回路を 2つ使用した構成との速度比である。また、[C]/[D]は、8-way マージソート回路を 1つ使用した文献[1]の構成と 2つ使用した提案構成との速度比である。

表 1 からわかるように、データ数によらず、提案

回路は、ソフトウェアによるクイックソートに比べ、13 倍程度の高速化が実現できている。また、マージソート回路を 2つ使用した[D]の結果は、1つの場合[C]に比べ、データ数 256M の時、1.6 倍高速であり、データ数が多くなるほど、2つのマージソート回路による並列処理の効果が現れていることがわかる。2-ポート DRAM メモリを導入した図 3 の構成[E]は、それを使用していない[D]と比べ、データ数 8K までは変化がない。これは、ボトルネックが DRAM へのアクセス（読み出し・書き込み）ではなく、マージツリーの出力バッファにあることが原因である。しかし、大規模データをソートする場合、ボトルネックは、DRAM メモリへのアクセスに移行するので、2-ポート DRAM メモリの導入効果が期待できる。大規模データに対しての計測を続ける。

#### 4 おわりに

本稿では、大量データを高速にソートするハードウェアソータを提案した。2個のマージソート回路を並列動作させることにより、大規模データの高速化が期待できることがわかった。

今後は、2-ポート DRAM メモリの導入効果の計測を続けると共に、ハードウェアソータのさらなる高速化を目指し、文献[2]で提案されている 8 データ同時出力が可能な並列マージソートツリーの導入や、より多くのマージソート回路を並列動作させる構成を検討する。

#### 参考文献

- [1] 小林諒平, 吉瀬謙二, “FPGA を用いた世界最速のソーティングハードウェアの実現に向けた試み,” 信学技法 RECONF2015-12(2015-6), pp.65-70, 2015.
- [2] Wei Song, Dirk Koch, Mikel Lujan, and Jim Garside, “Parallel Hardware Merge Sorter,” IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2016, DOI: 10.1109/FCCM.2016.34

表1 評価結果 (単位:m 秒)

# of input data	[A] Merge sort (SW)	[B] Quick sort (SW)	[C] One 8-way merge sort (Ref[1])	[D] Two 8-way merge sort	[E] [D] + 2-port DRAM	[B]/[D]	[C]/[D]
128	0.01000	0.01000	0.00097	0.00097	0.00097	10.3	1.0
1K	0.09900	0.08420	0.00951	0.00883	0.00883	9.5	1.1
8K	1.00200	0.79540	0.07517	0.06106	0.06106	13.0	1.2
64K	9.71400	7.29850	0.72470	0.55163	-	13.2	1.3
512K	91.89100	70.77390	7.03631	5.13893	-	13.8	1.4
4M	831.38300	622.65910	66.67784	47.51040	-	13.1	1.4
32M	7508.41360	5701.87060	617.18401	431.25539	-	13.2	1.4
256M	65449.07610	50660.97350	5608.38919	3405.15913	-	14.9	1.6