

7G-05

フィードバックレコード数の最適なマージネットワークを用いたマージツリー

齊藤 誠† 吉瀬 謙二††

†東京工業大学 工学部情報工学科

††東京工業大学 情報理工学院

†saitoh@arch.cs.titech.ac.jp ††kise@c.titech.ac.jp

1 はじめに

ソートは重要な計算カーネルであり、その高速処理を可能にするため、FPGAを用いたソーティングアクセラレータが研究されている。このアクセラレータは、多数のソートされたレコード列を1つにマージするモジュールであるマージツリーを用いて、マージベースの処理を行うものが主流である。また、マージツリーには、2つのソートされたレコード列を1つにマージするモジュールであるマージロジックを繋げることで構成されたものがあり、現在最高性能クラスのソーティングアクセラレータで採用されている[1]。そして我々は過去に、ソーティングアクセラレータのさらなる高性能化のため、既存のマージネットワーク[1][2]に最適化を施し、マージネットワーク単体の高性能化を図った[3]。

ここでレコードとは、ソートを行う対象を指している。レコードは、大小比較に使用する値であるキーと、キーに付随する値、回路の制御に使う値の3種類の値を1つずつ含む。以降、レコードのビット幅、キーのビット幅をそれぞれ記号 R, K で表し、キーに付随する値は32bit、制御に使う値は2bitとする。従って、 $R = K + 34$ で求められる。制御に使う値とは、Finish bit と Valid bit である。Finish bit は、あるマージ対象のレコード列のキューが尽きた後にそのキューから取り出されるレコードでアサートされ、Valid bit は、レコードが有効である時にアサートされる。これらはキーのMSBに隣接する形で配置される。

本稿では、[3]で最適化されたマージネットワークについて、それ単体では確認されている性能の向上が、マージツリーでも現れるかを検証する。そのため、まずマージツリーとマージロジック、マージネットワークの説明と、それぞれの実装方法の提案を行う。また、これに基づいて最適化前後のマージネットワークを使ったマージツリーを実装し、その論理合成結果を通して必要な資源量と動作周波数、及びスループットを評価する。

2 提案手法

マージツリー、マージロジック、マージネットワークの説明と実装方法の提案を行う。マージネットワークについては、[3]で行われた最適化の説明を併せて行う。

2.1 マージツリー

マージツリーとは、ソート済みの複数のレコード列を入力し、それらをマージした1つのレコード列を出力するモジュールである。1つのツリーを通すことによってマージできるレコード列の数を、記号 W で表す。

本稿では、マージツリーを図1のブロック図の通りに、節2.2で説明するマージロジックをツリー状に接続したものとして実装する。図1は、 $W=8$ のマージツリーであるため、マージツリーに入力される8つのレコード列が、左端の4つのマージロジックの $in1, in2$ にそれぞれ入力される。マージロジックは $in1, in2$ のレコード列をマージしたものを out か

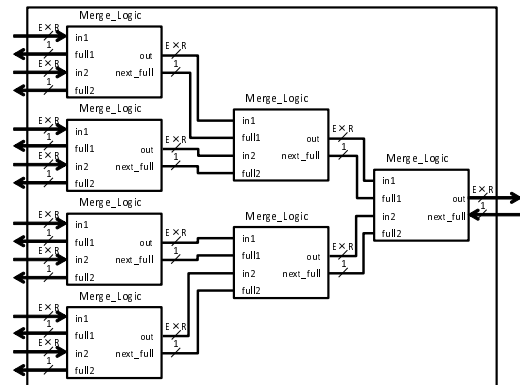


図 1: $W=8$ のマージツリーのブロック図

ら出力し、そしてこの出力がまた次に繋がれたマージロジックに入力されるため、左から右に向かってレコード列が8つから4つ、2つ、1つへとマージされる。このツリーが1サイクルにマージできるレコード数は、図1右端の、ツリーの根となるマージロジックが1サイクルにマージできるレコード数と同じである。また、マージツリーに入力される全てのレコードは、Valid bit がアサートされているものとする。

2.2 マージロジック

マージロジックとは、2つのソート済みのレコード列を1つのレコード列にマージするモジュールである。マージロジックは1サイクルに複数のレコードのマージが可能であり、この、1サイクルにマージできるレコードの数を以降は記号 E で表す。

本稿では、マージロジックを図2のブロック図の通りに実装する。図2中(a)では、マージロジックに入力される2つのソート済みレコード列が、そのままFIFOの in に入力されている。また、FIFOのpushには、マージロジックに入力されるレコード列の先頭と末尾のValid bitの論理和を取ったものが入力されており、有効なレコードが1つでもある場合にFIFOにレコード列がエンキューされるようになっている。図2中(b)では、一定の規則に従って E 個のレコードが片方のFIFOからデキューされ、節2.3で説明するマージネットワークに入力される。この時、マージネットワークは、1つにマージされたレコード列を出力する[3]ため、それを用いてマージロジックの出力とする。

マージロジックは、2つあるFIFOが片方でも空である場合、そのサイクルではマージが行えない。しかしこの場合でも、空のFIFOは予想できない値のレコードを有効なレコードとして出力してしまう可能性があるため、これを無効なレコードとして処理できるようにしなければならない。この解決方法として、Mask Validは、2つあるFIFOの片方でも空である場合にValid Bitをデアサートする。Valid Bitをデアサートすることで、任意のレコードをバブルとして機能させることができるためである。Valid Bitがデアサートされたレコードは、マージネットワークから可能な限り早く出力される性質を持つ。また、その出力は次に接続されたマージロジックの入力もしくはマージツリーの出力となるが、前述

An Efficient Merge Tree Using Merge Networks With Optimized Feedback Records

Makoto SAITOH†, and Kenji KISE††

†Department of Computer Science, Tokyo Institute of Technology ††School of Computing, Tokyo Institute of Technology

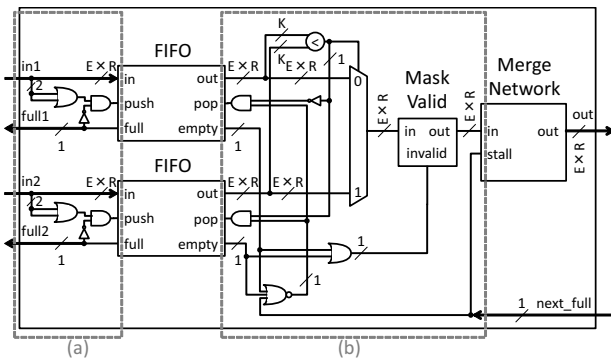


図 2: マージロジックのブロック図

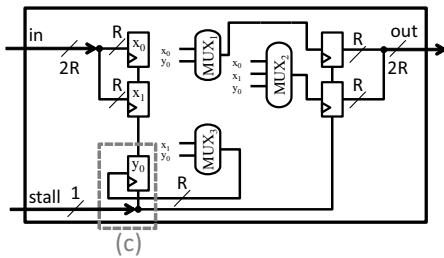


図 3: E=2 のマージネットワークのブロック図

の通りマージロジックの FIFO にはエンキューされず、マージツリーの出力でも同様に出力を無視できるため、バブルとして機能する。

2.3 マージネットワーク

マージネットワークを初めて提案したのは [1] であり、その後、[1] を高速化した [2] と、更に [2] を最適化した [3] が提案されているため、計 3 種類のマージネットワークが存在する。この節では、[3] の説明を行う。

マージネットワークとは、マージロジックに入力されたソート済みの 2 つのレコード列をマージロジックと協調することで 1 つのレコード列にマージする、マージロジックのカーネルとなるモジュールである。本稿ではマージネットワークを、[3] で提案されたものにマージツリーで使うための制御線を加えた、図 3 の通りに実装する。図では一例として E=2 の構成を示している。マージネットワークには、図 3 中 (c) に示す、レコードを格納する特別なレジスタ、フィードバックレジスタが E-1 レコード分存在する。そしてこのフィードバックレジスタのレコードと、入力されるレコードを合わせた計 2E-1 個のレコードをソートし、小さい E 個を出力、大きい E-1 個をフィードバックレジスタにフィードバックする、という動作を毎サイクル行う。[1] と [2] では、ここで E 個のレコードのフィードバックが行われていたが、必要な E-1 個のレコードのみをフィードバックするように最適化されている [3]。この最適化により、[3] は [2] と比較して、マージネットワーク単体の実装の資源量削減と動作周波数の向上を達成している。

また、マージネットワークでは、Finish bit と Valid bit とキーを接続したものを単にキーとして扱う。これにより、Valid bit がアサートされているレコードは先に出力され、Finish bit がアサートされているレコードは後に出力されるため、マージの動作に合致する動作を自然に実行することができる。

3 評価

2 章で示したマージツリーの性能評価を行う。[3] で提案されたマージネットワークの最適化について、記号 O を使い、最適化適用前のマージネットワークを O=0、適用後のものを O=1 と表す。評価対象は、この O と前述の E, W, K の 4 つのパラメータについて、それぞれパラメータを変化させたマージツリーを 2 章の通り実装したものである。(E, W) はセットとして (4,4), (8,8) の 2 通り、K は 16, 32, 64 の 3 通り、O は 0, 1 の 2 通りに変化させる。評価項目は資源量と動

表 1: 各パラメータと論理合成結果。E:1 サイクルにマージするレコード数、W:ウェイト数、K:キー幅、O:最適化

| (E,W) | K | O | 動作周波数 (MHz) | Slice | Logic LUTs | FFs | SRLs |
|-------|----|---|-------------|-------|------------|-------|------|
| (4,4) | 16 | 0 | 305 | 1314 | 2834 | 1858 | 654 |
| (4,4) | 16 | 1 | 315 | 1238 | 2420 | 1713 | 654 |
| (4,4) | 32 | 0 | 280 | 1724 | 3965 | 2422 | 840 |
| (4,4) | 32 | 1 | 290 | 1573 | 3373 | 2238 | 852 |
| (4,4) | 64 | 0 | 260 | 2611 | 6262 | 3618 | 1318 |
| (4,4) | 64 | 1 | 265 | 2530 | 5277 | 3326 | 1316 |
| (8,8) | 16 | 0 | 190 | 8511 | 23252 | 8672 | 3324 |
| (8,8) | 16 | 1 | 190 | 8052 | 21241 | 8254 | 3387 |
| (8,8) | 32 | 0 | 170 | 11630 | 33391 | 11350 | 4284 |
| (8,8) | 32 | 1 | 185 | 10575 | 29987 | 10883 | 4303 |
| (8,8) | 64 | 0 | 155 | 17821 | 52181 | 16856 | 6650 |
| (8,8) | 64 | 1 | 155 | 16896 | 47336 | 16148 | 6676 |

作周波数の 2 つである。XC7VX485T FPGA をターゲットとして、Vivado2016.4 で各実装を論理合成した結果で評価を行う。

表 1 に論理合成の結果を示す。必要な資源量については、Slice, Logic LUTs, FFs のそれぞれについて、O=0 の結果よりも O=1 の結果の方が減っていることが確認できる。SRLs については O=0 と O=1 の結果で殆ど差が無いが、これは SRL がマージロジックの FIFO に使われており、使用量がマージネットワークの最適化に依らないためであると考えられる。以上より、マージネットワークがフィードバックするレコード数の最適化によって、それを使ったマージツリーにおいて必要な資源量を削減できることが確認できる。動作周波数については、(E,W) が (4,4) か (8,8) の際には、O=1 の結果を O=0 の結果と比較すると、向上するか、もしくは変化していないことが分かる。

最後に、先行研究と性能を比較するため、マージツリーのスループットを求める。スループットは、E と W と動作周波数 f を使い、 $R=66$ 、つまり $K=32$ の時、 $E \times W \times f \times 8 \text{Byte/sec}$ と表せる。今回のマージツリーでは、 $W=8, K=32$ の条件下で得られる最大スループットが $W=8, K=32, E=8, O=1$ の構成時の 11.84GByte/sec に達している。これは、現在最高性能クラスの先行研究 [4] において $W=8, K=32$ の構成時のスループットである、 10.19GByte/sec を超えている。

4 結論

ソーティングアクセラレータの性能向上のために [3] で最適化されたマージネットワークを用いて、マージツリーの実装と評価を行った。その結果、最適化されたマージネットワークを使ったマージツリーは、最適化されていないマージネットワークを使ったもの比べて必要な資源量 (Slice 数) が削減され、動作周波数は向上することが確認できた。また、 $W=8, K=32$ の構成時に、[4] よりも高いスループットが出ることを確認した。以上より、[3] で提案された最適化は、マージツリーの性能を向上させることができると分かった。

参考文献

- [1] Jared Casper and Kunle Olukotun. Hardware Acceleration of Database Operations. In *Proceedings of the 2014 ACM/SIGDA International Symposium on Field-programmable Gate Arrays*, FPGA '14, pages 151–160, 2014.
- [2] Susumu Mashimo, Thiem Van Chu, and Kenji Kise. Cost-Effective and High-Throughput Merge Network Architecture for the Fastest FPGA Sorting Accelerator. In *International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies*, HEART '16, pages 7–12, 2016.
- [3] 齊藤 誠, 眞下 達, Thiem Van Chu, 吉瀬 謙二. FPGA ソーティングアクセラレータのためのマージネットワークの改良. In *信学技報*, RECONF '16, pages 13–18, 2016.
- [4] Wei Song, Dirk Koch, Mikel Luján, and Jim Garside. Parallel Hardware Merge Sorter. In *IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, FCCM '16, pages 95–102, 2016.