On Sorting Multiple Bitonic Sequences

De-Lei Lee
Department of Computer Science
York University, 4700 Keele Street
North York, Ontario
CANADA, M3J 1P3
E-Mail: delei@cs.yorku.ca

Kenneth E. Batcher
Department of Mathematics
and Computer Science
Kent State University
Kent, OH 44242, USA
E-Mail: batcher@cs.kent.edu

ABSTRACT

Bitonic sorters sort a single bitonic sequence into an ascending sequence. A multi-bitonic sorter is presented here, which sorts k bitonic sequences of n keys each into an ascending sequence in at most $(\lceil\log_2(k+\lceil\frac{k}{2}\rceil)\rceil+1)(\lceil\log_2 n\rceil-1)+T(1,k)+\lceil\log_2 k\rceil+1$ time delay, where $T(1,k)$ is the time delay needed to sort k keys in order; and k is any integer not restricted to 1.

Key Words: Sorting networks, bitonic sequence, bitonic sorter, multi-bitonic sorter, odd-even merger, parallel processing.

1. INTRODUCTION

Bitonic sorters can be used as parallel merging networks. Parallel sorting networks are constructed from a series of bitonic sorters. In 1968, it was shown that a $2n$-key bitonic sorter can be constructed from 2 n-key bitonic sorters and n 2-key bitonic sorters [1]. In the past two decades, much effort has been devoted to adapt this parallel sorting method to a variety of parallel computer architectures for efficiency such as the shuffle-exchange machine [2] and the mesh-connected parallel computers [4,5]. Various properties of bitonic sorters have been investigated [3].

In 1989, a more general construction was shown: a (pq)-key bitonic sorter can be built from p q-key bitonic sorters and q p-key bitonic sorters [6]. This construction requires that bitonic sorters sort bitonic sequences of any size. Subsequently, Batcher gave a construction of ascending-descending sorters to sort special bitonic sequences of any size [7]. More recently, Liszka and Batcher showed a construction of bitonic sorters to sort general bitonic sequences of any size [8].

While previous work has been on sorting a single bitonic sequence, in this paper we present a multi-bitonic sorter (or k-bitonic sorter for short). The sorter described here sorts k independent bitonic sequences into an ordered sequence, where k is any integer not restricted to 1.

The paper is organized as follows. The next section contains some definitions. Section 3 examines a fundamental property of bitonic sequences, and presents a construction of multi-bitonic sorters. Conclusions are drawn in Section 4.

2. DEFINITIONS

An ascending sequence is defined as a sequence of keys $\langle x_1, x_2, \cdots, x_n\rangle$ such that $x_1 \leq x_2 \leq \cdots \leq x_n$. A descending sequence is defined as a sequence of keys $\langle x_1, x_2, \cdots, x_n\rangle$ such that $x_1 \geq x_2 \geq \cdots \geq x_n$.

A sequence $\langle x_1, x_2, \cdots, x_n\rangle$ of n keys is bitonic if $x_1 \leq x_2 \leq \cdots \leq x_p \geq x_{p+1} \geq \cdots \geq x_n$, for some p, $1 \leq p \leq n$. A sequence is bitonic if the sequence is obtained by splitting a bitonic sequence anywhere and interchanging the two parts. Note when a sequence is bitonic, so are all of its subsequences.

A bitonic sorter rearranges a bitonic sequence into an ascending sequence. By contrast, a k-bitonic sorter rearranges k independent bitonic sequences into an ascending sequence for any integer $k \geq 1$.

The basic building block used by both the bitonic sorter and the multi-bitonic sorter is a 2-by-2 comparison-exchange element (or a *comparator* for short), as shown in Figure 1. It receives two keys over its inputs A and B and presents their minimum on its L output and their maximum on its H output in one time delay.

Both bitonic sorters and multi-bitonic sorters comprise a number of comparators. The cost of a sorter is the number of comparators in the sorter. The time delay of a sorter is the number of comparators in the longest path through the sorter.

The zero-one principle [3] will be used in the following sections for correctness proofs: If a network with n input lines sorts all 2^n sequences of 0's and 1's into ascending order, it will sort any arbitrary sequence of n numbers into ascending order.

Thus, we need only to consider sequences consisting of 0's and 1's. A sequence of l 0's, followed by $n-l-t$ 1's, followed by t 0's

$$\underbrace{00\cdots0}_{l}\underbrace{11\cdots1}_{n-l-t}\underbrace{00\cdots0}_{t}, \tag{1}$$

Proceedings of the International Conference on Parallel Processing (ICPP'94)
0-8493-2493-9/0-6 $20.00

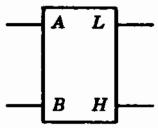

Figure 1: A comparison-exchange element.

where $0 \le l, t$ and $l + t \le n$, is a bitonic sequence of length n; and so is the sequence

$$\underbrace{11\cdots1}_{l}\underbrace{00\cdots0}_{n-l-t}\underbrace{11\cdots1}_{t}; \qquad (2)$$

and no other sequence is bitonic.

3. MULTI-BITONIC SORTERS

The following property of bitonic sequences is the key in our construction of multi-bitonic sorters. Proof of Theorem 1 below can be found in [8].

Theorem 1 *For any bitonic sequence* $\langle x_1, x_2, \cdots, x_n\rangle$, *let* z_1 *and* z_2 *be the numbers of 0's in the two subsequences* $\langle x_1, x_3, \cdots, x_{2\lceil n/2\rceil-1}\rangle$ *and* $\langle x_2, x_4, \cdots, x_{2\lfloor n/2\rfloor}\rangle$, *respectively. Then,*

$$-1 \le z_1 - z_2 \le 1, \ \textit{if } n \textit{ is even}; \qquad (3)$$

$$-1 \le z_1 - z_2 \le 2, \ \textit{if } n \textit{ is odd}. \qquad (4)$$

An N-key k-bitonic sorter is a sorter that rearranges k independent bitonic sequences of $\lceil N/k\rceil$ or $\lfloor N/k\rfloor$ keys into an ascending sequence of N keys. Let the k bitonic sequences be indexed from 1 to k, and the first m bitonic sequences have $n = \lceil N/k\rceil$ keys each, and the remaining $k - m$ ones have $n - 1$ keys each, where $m = N \bmod k$. We shall assume $m = k$ when N is a multiple of k. Let A_i denote the ith bitonic sequence: $A_i = \langle a_{i,1}, a_{i,2}, \cdots, a_{i,n}\rangle$, for $1 \le i \le m$; and $A_i = \langle a_{i,1}, a_{i,2}, \cdots, a_{i,n-1}\rangle$, for $m < i \le k$. For $1 \le i \le k$, let sequence $O_i = \langle a_{i,j} \mid j \text{ is odd}\rangle$ and sequence $E_i = \langle a_{i,j} \mid j \text{ is even}\rangle$; Both sequences O_i and E_i are subsequences of A_i, so they are bitonic.

A k-key k-bitonic sorter is a sorting network of order k, which can be constructed from comparators [1,7]. For any $N > k$, one can build an N-key k-bitonic sorter by repeatedly using the following rule:

1. For $1 \le i \le k$, presenting O_i, where i is odd, and E_i, where i is even, to a first k-bitonic sorter;
2. For $1 \le i \le k$, presenting E_i, where i is odd, and O_i, where i is even, to a second k-bitonic sorter;
3. Merging the outputs of the two k-bitonic sorters into an ascending sequence via a combining network.

In what follows, we describe the two small k-bitonic sorters and the combining network used in the construction. Let the notation $S(n, m)$ denote a $(k(n-1)+m)$-key k-bitonic sorter to rearrange m bitonic sequences each of size n followed by $k - m$ bitonic sequences each of size $n-1$, where $1 \le m \le k$. The two small k-bitonic sorters are derived as follows.

For odd n, our construction rule above assigns the first k-bitonic sorter with $\lceil \frac{m}{2}\rceil$ bitonic sequences of size $\lceil \frac{n}{2}\rceil$ and $k - \lfloor \frac{m}{2}\rfloor$ bitonic sequences of size $\lfloor \frac{n}{2}\rfloor$. So, $S(\lceil \frac{n}{2}\rceil, \lceil \frac{m}{2}\rceil)$ is the first k-bitonic sorter; and $S(\lceil \frac{n}{2}\rceil, \lfloor \frac{m}{2}\rfloor)$ is the second k-bitonic sorter. Similarly, for even n, the first and the second k-bitonic sorters are either $S(\frac{n}{2}, \lceil \frac{k+m}{2}\rceil)$ and $S(\frac{n}{2}, \lfloor \frac{k+m}{2}\rfloor)$, respectively, when m is even, or $S(\frac{n}{2}, \lfloor \frac{k+m}{2}\rfloor)$ and $S(\frac{n}{2}, \lceil \frac{k+m}{2}\rceil)$, respectively, when m is odd. Note that the numbers of keys assigned to each of the two small k-bitonic sorters differ by at most 1.

Before describing the construction of the combining network, we shall first show the property of the outputs of the two k-bitonic sorters.

Theorem 2 *Let the output of the first k-bitonic sorter be a sequence of* z_1 *0's followed by 1's and the output of the second k-bitonic sorter a sequence of* z_2 *0's followed by 1's. Then,*

$$|z_1 - z_2| \le k + \lceil \frac{m}{2}\rceil, \quad \textit{if } n \textit{ is odd}; \qquad (5)$$

$$|z_1 - z_2| \le k + \lceil \frac{k-m}{2}\rceil, \quad \textit{if } n \textit{ is even}. \qquad (6)$$

Proof. The bounds on $z_1 - z_2$ are determined by three kinds of bitonic sequences among the k bitonic sequences. Let $z_1 - z_2 = D_1 + D_2 + D_3$. The bounds on D_1 are determined by bitonic sequences with an even number of keys; The bounds on D_2 are determined by odd-indexed bitonic sequences with an odd number of keys; and the bounds on D_3 are determined by even-indexed bitonic sequences with an odd number of keys. Consider the following two cases separately.

1. n is odd. There are $k - m$ bitonic sequences of size $n - 1$, so $-(k - m) \le D_1 \le k - m$, according to inequality (3), since $n - 1$ is even. Each of the $\lceil \frac{m}{2}\rceil$ odd-indexed bitonic sequences of size n contributes to D_2 by the amount of -1, 0, 1, or 2, according to inequality (4). So, $-\lceil \frac{m}{2}\rceil \le D_2 \le 2\lceil \frac{m}{2}\rceil$. Each of the $\lfloor \frac{m}{2}\rfloor$ even-indexed bitonic sequences of size n contributes to D_3 by the amount of -2, -1, 0, or 1, according to (4). So, $-2\lfloor \frac{m}{2}\rfloor \le D_3 \le \lfloor \frac{m}{2}\rfloor$. Hence, $-(k + \lfloor \frac{m}{2}\rfloor) \le z_1 - z_2 \le k + \lceil \frac{m}{2}\rceil$, inequality (5).
2. n is even. Then, $-m \le D_1 \le m$, due to the m n-key bitonic sequences and (3). The number of odd-indexed bitonic sequences of size $n-1$ can be either $\lceil \frac{k-m}{2}\rceil$ or $\lfloor \frac{k-m}{2}\rfloor$, depending on the value of m.

Proceedings of the International Conference on Parallel Processing (ICPP'94)
0-8493-2493-9/0-6 $20.00 © 1994 IEEE

In the case of $\lceil \frac{k-m}{2} \rceil$, we have $-\lceil \frac{k-m}{2} \rceil \le D_2 \le 2\lceil \frac{k-m}{2} \rceil$, due to the $\lceil \frac{k-m}{2} \rceil$ odd-indexed $(n-1)$-key bitonic sequences and (4). It follows that $-2\lfloor \frac{k-m}{2} \rfloor \le D_3 \le \lfloor \frac{k-m}{2} \rfloor$, because of the $\lfloor \frac{k-m}{2} \rfloor$ even-indexed $(n-1)$-key bitonic sequences and (4). So, $-(k+\lfloor \frac{k-m}{2} \rfloor) \le z_1 - z_2 \le k + \lceil \frac{k-m}{2} \rceil$.

In the case of $\lfloor \frac{k-m}{2} \rfloor$, we have $-\lfloor \frac{k-m}{2} \rfloor \le D_2 \le 2\lfloor \frac{k-m}{2} \rfloor$, due to the $\lfloor \frac{k-m}{2} \rfloor$ odd-indexed $(n-1)$-key bitonic sequences. It follows that $-2\lceil \frac{k-m}{2} \rceil \le D_3 \le \lceil \frac{k-m}{2} \rceil$, because of the $\lceil \frac{k-m}{2} \rceil$ even-indexed $(n-1)$-key bitonic sequences. So, $-(k+\lceil \frac{k-m}{2} \rceil) \le z_1 - z_2 \le k + \lfloor \frac{k-m}{2} \rfloor$. Consequently, inequality (6) holds for both cases.

This concludes the proof. □

Construction of the combining network is based on Batcher's odd-even merger [1], which is given below in preparation for that of the combining network.

Let $\langle x_1, x_2, \cdots, x_p \rangle$ and $\langle y_1, y_2, \cdots, y_q \rangle$ be the two ascending sequences to be merged. A p-by-q odd-even merger is constructed recursively as follows:

1. If $pq = 1$, the odd-even merger is simply a comparator.
2. If $pq > 1$, merge the two odd sequences $\langle x_1, x_3, \cdots, x_{2\lceil p/2 \rceil -1} \rangle$ and $\langle y_1, y_3, \cdots, y_{2\lceil q/2 \rceil -1} \rangle$, obtaining the sorted result $\langle v_1, v_2, \cdots, v_{\lceil p/2 \rceil + \lceil q/2 \rceil} \rangle$; and merge the two even sequences $\langle x_2, x_4, \cdots, x_{2\lfloor p/2 \rfloor} \rangle$ and $\langle y_2, y_4, \cdots, y_{2\lfloor q/2 \rfloor} \rangle$, obtaining the sorted result $\langle w_1, w_2, \cdots, w_{\lfloor p/2 \rfloor + \lfloor q/2 \rfloor} \rangle$. Finally, apply the comparison-exchange operations
$$w_1 : v_2, w_2 : v_3, \cdots, w_q : v_{q+1} \tag{7}$$
to the sequence
$$\langle v_1, w_1, v_2, w_2, \cdots, v_q, w_q, \cdots, v_p \rangle \tag{8}$$

to obtain the final sorted sequence.

An odd-even merging network thus constructed is composed of two small odd-even mergers to produce the two sorted sequences $\langle v_1, v_2, \cdots, v_{\lceil p/2 \rceil + \lceil q/2 \rceil} \rangle$ and $\langle w_1, w_2, \cdots, w_{\lfloor p/2 \rfloor + \lfloor q/2 \rfloor} \rangle$, and a column of comparators to carry out the comparison-exchange operations in (7). It merges the two ascending sequences in $1 + \lceil \log_2 \max\{p, q\} \rceil$ time delay. Correctness proof of the odd-even merge can be found in [1].

Lemma 1 *Let $\langle x_1, x_2, \cdots, x_p \rangle$ and $\langle y_1, y_2, \cdots, y_q \rangle$ be two ascending sequences of z_x 0's followed by 1's and z_y 0's followed by 1's, respectively, and $p \ge q$. Then, these two ascending sequences can be merged into an ascending sequence in exact time delay of $\lceil \log_2 u \rceil$, if $0 \le z_x - z_y \le u$, where u is an integer $\le p$.*

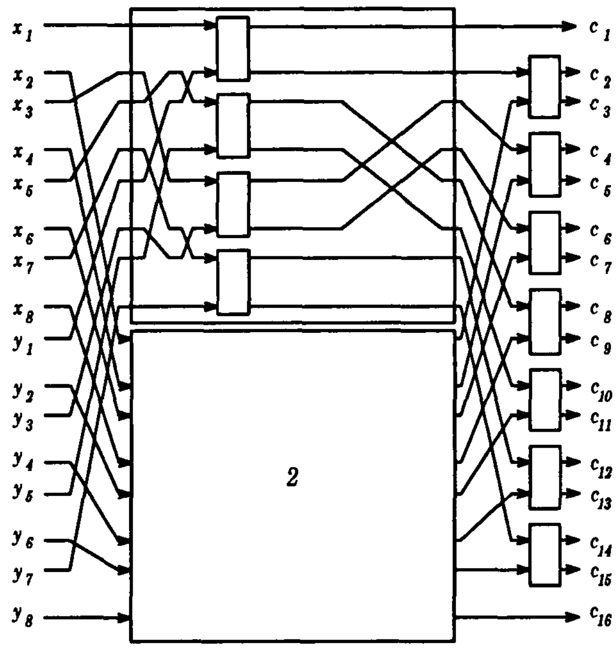

Figure 2: The combining network $C(8, 8, 2)$.

Proof. We show that the last $\lceil \log_2 u \rceil$ stages of a p-by-q odd-even merger can be used to merge the two sequences into one sorted sequence.

To see this, observe that the last stage of the odd-even merger can merge two ascending sequences $\langle x_1, x_2, \cdots, x_p \rangle$ and $\langle y_1, y_2, \cdots, y_q \rangle$ such that $0 \le z_x - z_y \le 2$ into an ascending sequence. The last two stages of the odd-even merger can merge two sequences $\langle x_1, x_2, \cdots, x_p \rangle$ and $\langle y_1, y_2, \cdots, y_q \rangle$ such that $0 \le z_x - z_y \le 4$ into an ascending sequence, etc. Thus, the last $\lceil \log_2 u \rceil$ stages of the odd-even merger can be used to merge the two ascending sequences given in the lemma into a sorted sequence. □

Lemma 2 *Let $\langle a_1, a_2, \cdots, a_p \rangle$ and $\langle b_1, b_2, \cdots, b_q \rangle$ be two ascending sequences of z_a 0's followed by 1's and z_b 0's followed by 1's, respectively, such that $-u \le z_a - z_b \le u$. Then, the first stage of a p-by-q odd-even merger generates two ascending sequences $\langle x_1, x_2, \cdots, x_{\max\{p,q\}} \rangle$ and $\langle y_1, y_2, \cdots, y_{\min\{p,q\}} \rangle$ of z_x 0's followed by 1's and z_y 0's followed by 1's, respectively, such that $0 \le z_x - z_y \le u$.*

Proof. By the recursive construction rule of the p-by-q odd-even merger, the first stage of the odd-even uses $\min\{p, q\}$ comparators to produce two sets $\{\min\{a_i, b_i\} \mid 1 \le i \le \min\{p, q\}\}$ and $\{\max\{a_i, b_i\} \mid 1 \le i \le \min\{p, q\}\}$; the $\langle x_1, x_2, \cdots, x_{\max\{p,q\}} \rangle$ sequence generated contains all elements in the first set plus all the elements not involved in the comparison, and $\langle y_1, y_2, \cdots, y_{\min\{p,q\}} \rangle$ contains all elements in the second set. So, the first set contains $\mid z_a - z_b \mid$ more 0's than does the second set. It follows that $0 \le z_x - z_y \le u$. □

Proceedings of the International Conference on Parallel Processing (ICPP'94)
0-8493-2493-9/0-6 $20.00 © 1994 IEEE

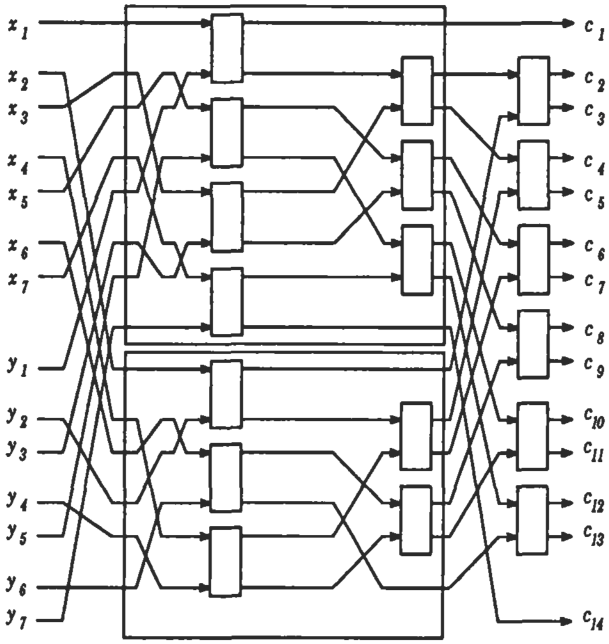

Figure 3: The combining network $C(7,7,3)$.

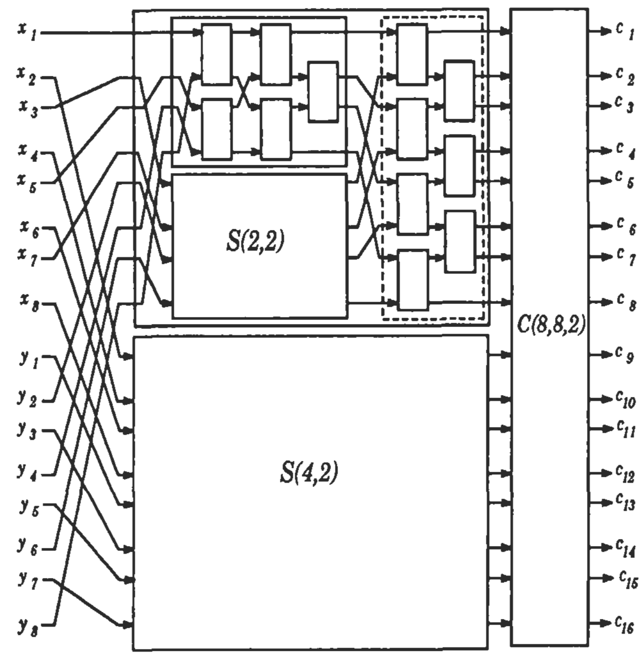

Figure 4: A 16-key 2-bitonic sorter.

Thus, a comparator network comprised of the first stage and the last $\lceil \log_2 u \rceil$ stages of a p-by-q odd-even merger can be used as a combining network to merge two ascending sequences $\langle a_1, a_2, \cdots, a_p \rangle$ and $\langle b_1, b_2, \cdots, b_q \rangle$ such that $-u \leq z_a - z_b \leq u$ into an ascending sequence. Let $C(p,q,u)$ denote such a combining network.

Figure 2 illustrates a combining network thus obtained, when $p = q = 8$, and $u \leq 2$, which comprises the first stage and the last stage of a 8-by-8 odd-even merging network.

Figure 3 illustrates a combining network for $p = q = 7$, and $u \leq 3$, composed of the first stage and the last two stages of a 7-by-7 odd-even merging network.

Theorem 3 *The outputs of the two k-bitonic sorters can be merged into an ascending sequence by a combining network in exact time delay of*

$$\lceil \log_2(k + \lceil \frac{m}{2} \rceil) \rceil + 1, \qquad \text{if } n \text{ is odd}; \quad (9)$$

$$\lceil \log_2(k + \lceil \frac{k-m}{2} \rceil) \rceil + 1, \quad \text{if } n \text{ is even}. \quad (10)$$

Proof. It follows from Lemmas 1, 2, and inequality (5) of Theorem 2 for odd n and inequality (6) of Theorem 2 for even n. □

Figure 4 depicts a 16-key 2-bitonic sorter, $S(8,2)$, where the combining network $C(8,8,2)$ following the two $S(4,2)$'s is identical to that of Figure 2. The two $S(2,2)$'s, one of which is shown in detail, are followed by the combining network in the dot-lined box.

Figure 5 illustrates a 14-key 2-bitonic sorter, $S(7,2)$, where the combining network $C(7,7,3)$ following the two $S(4,1)$'s is identical to that of Figure 3. The combining network following $S(2,1)$ and $S(2,2)$ is highlighted by the dot-lined box, and the detail of $S(2,1)$ is shown.

Let $T(n,m)$ be the time delay needed for $S(n,m)$ to sort k bitonic sequences into ascending order. Note that $T(n,0) = T(n-1,k)$. It follows from the iterative construction rule and Theorem 3 that

$$T(n,m) = \begin{cases} \max\{T(\lceil \frac{n}{2} \rceil, \lceil \frac{m}{2} \rceil), T(\lceil \frac{n}{2} \rceil, \lfloor \frac{m}{2} \rfloor)\} \\ \quad + \lceil \log_2(k + \lceil \frac{m}{2} \rceil) \rceil + 1 & \text{if } n \text{ is odd}; \\ \max\{T(\lceil \frac{n}{2} \rceil, \lceil \frac{k+m}{2} \rceil), T(\lceil \frac{n}{2} \rceil, \lfloor \frac{k+m}{2} \rfloor)\} \\ \quad + \lceil \log_2(k + \lceil \frac{k-m}{2} \rceil) \rceil + 1 & \text{if } n \text{ is even}. \end{cases}$$

So, it takes $(\lceil \log_2 k \rceil + 1)\log_2 n + T(1,k)$ time delay for the multi-bitonic sorter to rearrange k bitonic sequences of n keys each into an ascending sequence, when n is a power of two. The upper bound on the time delay of the multi-bitonic sorter for any n is given below.

Theorem 4 *The multi-bitonic sorter rearranges k bitonic sequences of n keys each into an ascending sequence in at most* $(\lceil \log_2(k + \lceil \frac{k}{2} \rceil) \rceil + 1)(\lceil \log_2 n \rceil - 1) + T(1,k) + \lceil \log_2 k \rceil + 1$ *time delay.*

Proof. Since $T(1,m) \leq T(1,k)$; $T(2,m) \leq T(2,k) = T(1,k) + \lceil \log_2 k \rceil + 1$; and $\max\{\lceil \log_2(k + \lceil \frac{m}{2} \rceil) \rceil, \lceil \log_2(k + \lceil \frac{k-m}{2} \rceil) \rceil\} \leq \lceil \log_2(k + \lceil \frac{k}{2} \rceil) \rceil$. □

Proceedings of the International Conference on Parallel Processing (ICPP'94)
0-8493-2493-9/0-6 $20.00 © 1994 IEEE

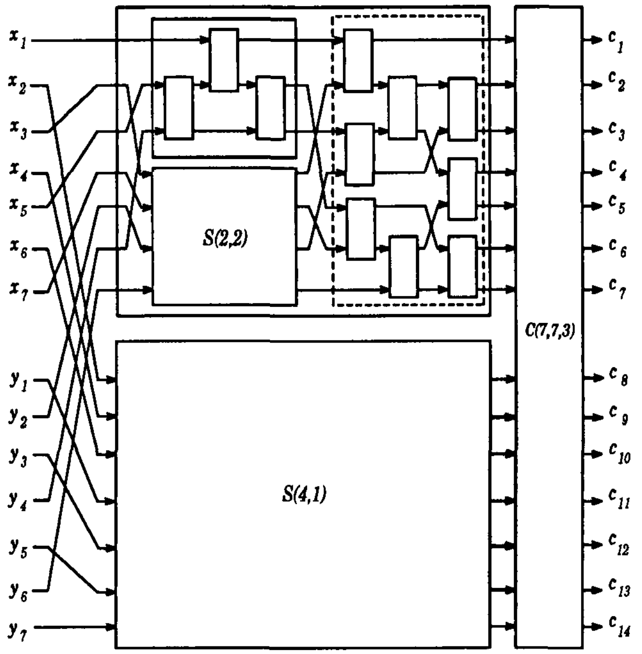

Figure 5: A 14-key 2-bitonic sorter.

The k bitonic sequences of n keys each could be sorted by first sorting each bitonic sequence into ascending order using k independent bitonic sorters; and then merging the k sorted keys into an ascending sequence using a k-way merging network of Lee and Batcher [9]. This would take at most $2\lceil \log_2 n \rceil - 1 + \lceil \log_2 k \rceil \lceil \log_2 n \rceil + T(1,k)$ time delay. The k-bitonic sorter takes at most $(\lceil \log_2(k + \lceil \frac{k}{2} \rceil) \rceil + 1)(\lceil \log_2 n \rceil - 1) + T(1,k) + \lceil \log_2 k \rceil + 1$ time delay; so, it can be faster by as much as $\lceil \log_2 n \rceil - 1$ time delay in some cases.

4. Conclusions

The multi-bitonic sort described in this paper sorts k bitonic sequences into an ascending sequence for any integer $k \geq 1$. This differs from the bitonic sort that sorts only a single bitonic sequence. The bitonic sort is a special case of the k-bitonic sort, when $k = 1$. In this case, the multi-bitonic sort is identical to the bitonic sort and retains the same complexity as that of the bitonic sort in terms of both comparators and time delay. The multi-bitonic sort, however, provides an important extension of the bitonic sort, allowing sorting more than one bitonic sequences in parallel.

Acknowledgment

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant NSERC-OGP0009196.

References

[1] K.E. Batcher, "Sorting Networks and Their Applications," *AFIPS Proc. Spring Joint Computer Conference*, pp. 307-314, 1968.

[2] H.S. Stone, "Parallel Processing with the Perfect Shuffle," *IEEE Trans. on Computers*, vol. 20, pp. 153-161, 1971.

[3] D.E. Knuth, *The art of Computer Programming, vol. 3: Sorting and Searching*, Addison-Wesley, 1973.

[4] C.D. Thompson and H.T. Kung, "Sorting on a Mesh-Connected Parallel Computer," *Communications of the ACM*, vol. 20, pp. 263-271, 1977.

[5] D. Nassimi and S. Sahni, "Bitonic Sort on a Mesh-Connected Parallel Computer," *IEEE Trans. on Computers*, vol. 27, pp. 2-7, 1979.

[6] T. Nakatani, S-T. Huang, B.W. Arden, and S. Tripathi, "K-Way Bitonic Sort," *IEEE Trans. on Computers*, vol. 38, pp. 283-288, 1989.

[7] K.E. Batcher, "On Bitonic Sorting Networks," *Proc. 19th International Conference on Parallel Processing*, vol. 1, pp. 376-379, 1990.

[8] K.J. Liszka and K.E. Batcher, "A Generalized Bitonic Sorting Network," *Proc. 22nd International Conference on Parallel Processing*, vol. 1, pp. 105-108, 1993.

[9] D.-L. Lee and K.E. Batcher, "A Multiway Merge Sorting Network," *IEEE Trans. on Parallel and Distributed Processing Systems*, 1994 (in press).

Proceedings of the International Conference on Parallel Processing (ICPP'94)
0-8493-2493-9/0-6 $20.00 © 1994 IEEE