Lower Bounds On Bitonic Sorting Networks

Tamir Levi* Ami Litman†

March 26, 2009

Abstract

This paper points to an abnormal phenomena of comparator networks. For most key processing problems (such as sorting, merging or insertion) the smaller the input size the easier the problem. Surprisingly, this is not the case for Bitonic sorting. Namely, the minimal depth of a comparator network that sorts all Bitonic sequences of \( n \) keys is not monotonic in \( n \).

We show that for any \( n \), not a power of two, the depth of a Bitonic sorter of \( n \) keys is at least \( \lfloor \log(n) \rfloor + 1 \). This contrast with Batcher’s seminal construction of a Bitonic sorter of \( 2^j \) keys and depth \( j \). That is, in order to reduce the number of keys from \( n = 2^j \) to \( n' > 2^{j-1} \) one must increase the depth of the network.

Keywords: Bitonic Sorting, Merging, Comparator Networks, Min Max Networks

1 Introduction

This paper points to an abnormal phenomena of comparator networks. For most key processing problems (such as sorting, merging or insertion) the smaller the input size the easier the problem. Surprisingly, this is not the case for Bitonic sorting. Namely, the minimal depth of a comparator network that sorts all Bitonic sequences of \( n \) keys is not monotonic in \( n \).

Let \( D(n) \) denote the minimal depth of a Bitonic sorter of \( n \) keys. This paper shows that for any \( n \), not a power of two, \( D(n) \geq \lfloor \log(n) \rfloor + 1 \). Due to Batcher’s seminal construction \( D(2^j) = j \). That is, \( D(n) < D(n') \) for any \( n = 2^j > n' > 2^{j-1} \).

For some values of \( n \) (e.g., 3 and 5) our bound is tight; however, we do not know if this bound is always tight. We also do not know if \( D(n) = \log(n) + O(1) \).

2 Preliminaries

Our work concerns the well-known concept of a comparator network [3]. To be self contained, we provide the following definitions. A comparator is a combinational device that receives, via two incoming edges, two keys and sorts them. It has two outgoing edges; on one of them, called the MIN-edge, it sends the minimal key and on the other outgoing edge, called the MAX-edge, it sends the maximal key. A comparator network is an acyclic network of comparators. See Figure 1. In this figure, a solid arrowhead denotes a MAX-edge and a hollow arrowhead denotes a Min-edge. Keys enter a comparator network via its input ports and exit the network via its output ports. These ports are depicted by solid circles. The network specifies, in some form, how the input is fed to the input ports and how the output is assembled.

*Faculty of Computer Science, Technion, Haifa 32000, Israel. leyt@cs.technion.ac.il
†Faculty of Computer Science, Technion, Haifa 32000, Israel. litman@cs.technion.ac.il
from the output ports. The fan-out of an input port in a comparator network is exactly one. Hence, a comparator network has the same number of input ports and output ports; this number is referred to as the **width** of the network. The **depth** of the network is the maximal number of comparators on a directed path in the network. For example, the network of Figure 1 is of width five and depth four.

![Figure 1: A Bitonic sorter of width 5 and depth 4](image)

As said, comparator networks processes **keys** which are members of some ordered set, $\mathbb{K}$. The exact nature of keys is usually not important but for definiteness we choose $\mathbb{K} = \mathbb{Q}$. This paper focuses on comparator networks that sort Bitonic sequences. A sequence of keys is **Bitonic**\(^1\) if it is a rotation of a concatenation of two sequences – an ascending sequence followed by a descending one. A **Bitonic sorter of width** $n$ is a comparator network that sort any Bitonic sequence of $n$ keys. The network of Figure 1 is a Bitonic sorter. Verifying the last statement via a variant of the 0-1 Principle [3] requires to test all 20 binary sequences of width five that are Bitonic and non-constant. However, there is a better way to accomplish the same goal. By Lemma 2, it suffices to verify that the network in question sorts the following five sequences: $\langle 1, 2, 3, 4, 5 \rangle$, $\langle 2, 3, 4, 5, 1 \rangle$, $\langle 3, 4, 5, 1, 2 \rangle$, $\langle 4, 5, 1, 2, 3 \rangle$, $\langle 5, 1, 2, 3, 4 \rangle$.

The Bitonic sorter of Figure 1 demonstrates that $D(5) \leq 4$. Our lower bound implies that $D(5) \geq \lceil \log(5) \rceil + 1 = 4$; hence, for $n = 5$ our lower bound is tight.

3 **Related work**

We summarize the little that is known on the function $D$. By a straightforward reachability argument, for every $n$,

$$D(n) \geq \lceil \log(n) \rceil \quad (1)$$

Batcher’s [1] well-known Bitonic sorter is based on a network of depth one that splits every Bitonic sequence of width $2n$ into two Bitonic sequences of width $n$. The first contains the lower keys and the second contains the higher keys; therefore,

$$D(2 \cdot j) \leq D(j) + 1 \quad (2)$$

This, the fact that $D(2) = 1$, and Inequality (1) imply:

$$D(2^j) = j \quad (3)$$

Nakatani et al. [5] has presented an elegant procedure to produce a larger Bitonic sorter from smaller ones as follows. Let $B'$ and $B''$ be two Bitonic sorters of width $n'$ and $n''$ and of depth $d'$ and $d''$, respectively. They have shown how to combine several copies of $B'$ and $B''$ into a Bitonic sorter of width $n' \cdot n''$ and of depth $d' + d''$. This implies that

$$D(i \cdot j) \leq D(i) + D(j) \quad (4)$$

\(^1\)The term ‘Bitonic’ was coined by Batcher [1] and we follow his terminology. We caution the reader that some authors use the same term with other meanings.
The only technique that constructs Bitonic sorters for any width is due to Batcher and Liszka [6]. Their construction partitions a Bitonic sequence of length \( n \) into two Bitonic subsequences of length \( \lceil \frac{n}{2} \rceil \) and \( \lfloor \frac{n}{2} \rfloor \) and sorts each of them recursively. The two resulting sorted sequences are then merged into a single sorted sequence. This merge requires a depth one network when \( n \) is even and a depth two network when \( n \) is odd. Therefore,

\[
\mathcal{D}(2n + 1) \leq \max(\mathcal{D}(n), \mathcal{D}(n + 1)) + 2
\]

(5)

This implies that for every \( n \),

\[
\mathcal{D}(n) \leq 2 \lceil \log(n) \rceil - 1
\]

(6)

Levy and Litman [4] studied another class of key processing networks – the Min-Max networks. A Min-Max network is built of 2-input Min-gates and Max-gates. There is no fanout restriction on the gates or the input ports of a Min-Max network. They construct, for every \( n \), a Min-Max network that is a Bitonic sorter of width \( n \) whose depth is at most \( \lceil \log(n) \rceil + 3 \). As said, it is unknown if the comparator model has a Bitonic sorter of a comparable depth – namely, of depth \( \lceil \log(n) \rceil + \mathcal{O}(1) \). They also show that the above abnormality exists (in a slightly different form) in Min-Max networks.

4 Lower Bound of Bitonic sorters

In this section we establish the main result of this paper. Namely, that:

**Theorem 1** For every \( n \), not a power of two, \( \mathcal{D}^n > \lceil \log(n) \rceil \).

To this end, we propose a new manner to organize keys. Traditionally, keys are organized as sequences. Such sequences are fed into and produced by comparator networks. However, in many cases it is preferable to consider the keys to have a different structure. For example, in the work of Nakatani [5] keys are arranged in a matrix. In the context of Bitonic sequences we prefer to consider the keys to be arranged on a circle. To study arbitrary arrangement of keys, we pick a finite set, \( X \), called the index set. A vector \( v \) over \( X \) is a function \( v \colon X \to \mathbb{K} \). We denote the elements of \( v \) by \( v_x \), rather than \( v(x) \), and the term ‘index’ is due to this notation. For example, a vector over \( \{1, 2, \ldots, n\} \) is the familiar sequence of \( n \) keys while a vector over \( \{1, 2, \ldots, n\} \times \{1, 2, \ldots, m\} \) is an \( n \times m \) matrix of keys.

To allow networks to receive and produce such vectors, a network \( N \) is associated with an input index set denoted \( \mathcal{I}(N) \) and with an output index set denoted \( \mathcal{O}(N) \). Such a network receives vectors over \( \mathcal{I}(N) \) and produces vectors over \( \mathcal{O}(N) \). The former vectors are called input vector of \( N \). There are two bijections that specify how the input is fed into the network and how the output is collected from the network; one is between the input ports and \( \mathcal{I}(N) \) and the other is between the output ports and \( \mathcal{O}(N) \).

A sequential key structure is, by nature, unsymmetrical; it has a first member and a last member. In the context of Bitonic sequences, this lack of symmetry is artificial, since Bitonic sequences (by definition) are closed under rotations. Hence, it is preferred that the keys are organized in a structure that is of a circular nature – a structure that has no first element or last element. To formalize this idea we pick, for every \( n \), some directed graph \( G^n = (C^n, E^n) \) which is a cycle of \( n \) vertices. The exact nature of the vertices is not important. The set \( C^n \) serves as the index set for Bitonic vectors. Namely, a circular vector of width \( n \) is a function \( v : C^n \to \mathbb{K} \). In contrast to a sequence, a circular vector has no first or last elements – all elements play the same role in this structure. Figure 2 depicts three such circular vectors. By our convention, a Bitonic sorter \( B \) receives circulars vectors over \( C^n \); hence, \( \mathcal{I}(B) = C^n \) and \( \mathcal{O}(B) = \{1, 2, \ldots, n - 1, n\} \).

A circular vector \( v : C^n \to \mathbb{K} \) is Bitonic if there are two vertices \( u', u'' \in C^n \) such that \( v \) is weakly increasing on the path from \( u' \) to \( u'' \) (including the two endpoints), and \( v \) is weakly decreasing on the path from \( u'' \) to \( u' \) (including the two endpoints). All three circular vectors of Figure 2 are Bitonic. A circular vector \( v \) is Unitonic if there is simple path that covers all of \( C^n \) and \( v \) is weakly increasing on
Observation 1: Clearly, permutation $p$ which is replaced with $j$ is a permutation if every key of the interval $[1, n]$ appears in $v$; hence, every such key appears exactly once in $v$. For example, vector (c) of Figure 2 is a permutation. The concept of Unitonic permutations is important due to the following lemma of [2].

Lemma 2 ([2]) A comparator network is a Bitonic sorter if and only if it sorts all Unitonic permutations of the appropriate width.

Recall that we assume that $\mathbb{K} = \mathbb{Q}$. Let $v'$ and $v''$ be two vectors over an index set $X$. The distance between $v'$ and $v''$ is defined by $\delta(v', v'') = \max_{x \in X} |v'_x - v''_x|$. Let $e$ be an edge and $v$ be an input vector of a comparator network $N$. We define $v(e)$ to be the key transmitted on $e$ when $v$ is fed into $N$. For a set $V$ of input vectors of $N$, define $V(e) = \{v(e) | v \in V\}$. Let $\mathcal{P}^n$, $\mathcal{B}^n$, and $\mathcal{U}^n$ denote the set of permutations, Bitonic permutations and Unitonic permutations, all of width $n$, respectively. The following lemma is due to Alekseyev [3, Section 5.3.4 Exercise 25].

Lemma 3 Let $e$ be an edge of a comparator network of width $n$. Then $\mathcal{P}^n(e)$ is an interval.

Lemma 3 follows from the following two lemmas. The first lemma can be established by a simple induction.

Lemma 4 Let $x$ and $y$ be two input vectors of a comparator network $N$ and let $e$ be an edge of $N$. Then $|v(x) - v(y)| \leq \delta(x, y)$.

For the next lemma, we use the following terminology. Two vectors, $v'$ and $v''$, are $1$-neighbors if they are over the same index set and $\delta(v', v'') = 1$. For a set $V$ of vectors over the same index set, we define the binary relation ‘$V$-neighbor’ as the transitive closure of the ‘$1$-neighbor’ relation. Namely, two vectors, $v'$ and $v''$, are $V$-neighbors if there is a sequence of vectors, all members of $V$, in which the first is $v'$, the last is $v''$ and every two consecutive vectors are $1$-neighbors. Clearly, the ‘$V$-neighbor’ relation is an equivalent relation. The following observation is straightforward.

Lemma 5 Every two members of $\mathcal{P}^n$ are $\mathcal{P}^n$-neighbors.

As said, Lemma 3 easily follows from Lemmas 4 and 5. We next prove a variant of Lemma 3 in which $\mathcal{P}^n$ is replaced with $\mathcal{B}^n$. To this end, it suffices to prove a variant of Lemma 5 in which $\mathcal{P}^n$ is replaced with $\mathcal{B}^n$. We use the following terminology. Let $k'$ and $k''$ be two keys that appear in a permutation $p$. Define $p^{k'\leftrightarrow k''}$ to be the permutation derived from $p$ by swapping the keys $k'$ and $k''$. Clearly, $\delta(p, p^{k'\leftrightarrow k''}) = |k' - k''|$. Assume $p$ is a permutation over the index set $\mathcal{C}^n$. We say that $k'$ and $k''$ are adjacent in $p$ if they are associated with consecutive vertices of $\mathcal{C}^n$.

Let $b \in \mathcal{B}^n$. Then following two observations are straightforward:

Observation 1: $b^{1\rightarrow 2}, b^{n-1\rightarrow n} \in \mathcal{B}^n$.

Observation 2: Let $k \in [1, n-1]$ such that $k$ and $k + 1$ are not adjacent in $b$. Then $b^{k\rightarrow k+1} \in \mathcal{B}^n$. 
Observation 3: For every \( b \in B^n \) there is a \( u \in U^n \) such that \( b \) and \( u \) are \( B^n \)-neighbors and \( b_x = u_x = 1 \) for some index \( x \in C^n \).

Recall that ‘\( B^n \)-neighbor’ is an equivalent relation. By Observations (1) and (3):

Observation 4: Every two members of \( U^n \) are \( B^n \)-neighbors.

Observations (3) and (4) imply the desired variant of Lemma 5.

Lemma 6 Every two members of \( B^n \) are \( B^n \)-neighbors.

Lemmas 4 and 6 imply the required variant of Alekseyev’s result.

Lemma 7 Let \( N \) be a comparator network such that \( \mathcal{I}(N) = C^n \) and let \( e \) be an edge of \( N \). Then \( B^n(e) \) is an interval.

Note that Lemma 7 does not require the network in question to be a Bitonic sorter.

Let \( e \) be an edge of a comparator network \( N \). Define \( \mathcal{I}(e) \subseteq \mathcal{I}(N) \) to be the set of indexes associated with input ports that have a path to \( e \). Similarly, define \( \mathcal{O}(e) \subseteq \mathcal{O}(N) \) to be the set of indexes associated with output ports that are reachable from \( e \). The input depth of \( e \), denoted \( d^I(e) \), is the maximal number of comparators along a path from an input port to \( e \), not including the comparator at the end of \( e \) (if there is such a comparator). Similarly, the output depth of \( e \), denoted \( d^O(e) \), is the maximal number of comparators along a path from \( e \) to an output port, not including the comparator from which \( e \) exits (if there is such a comparator). Clearly,

\[
\mathcal{I}(N) \leq 2d^I(e) \quad \text{and} \quad \mathcal{O}(N) \leq 2d^O(e)
\]

Let \( d(N) \) denote the depth of \( N \), that is, the maximal number of comparators on a directed path in \( N \). Let \( e \) be a comparator of a Bitonic sorter. We say that \( c \) is degenerate if its incoming edges can be named \( e' \) and \( e'' \) such that \( b'(e) \leq b''(e) \), for every two input vectors, \( b' \) and \( b'' \), which are Bitonic.

Lemma 8 Let \( N \) be a Bitonic sorter with no degenerate comparators. Then \( \mathcal{O}(e) \) is an interval, for every edge \( e \).

Proof: The proof is by induction on \( d^O(e) \). The case where \( d^O(e) = 0 \) is trivial. Assume otherwise and note the following fact. For two intervals of integers, \( I' \) and \( I'' \), \( I' \cup I'' \) is an interval if and only if there exist \( i' \in I' \) and \( i'' \in I'' \) such that \( |i' - i''| \leq 1 \).

Let \( e' \) and \( e'' \) be two edges that enter a comparator \( c \) and let \( f' \) and \( f'' \) be the two outgoing edges of \( c \). By Lemma 7, \( B^n(e') \) and \( B^n(e'') \) are intervals. Since \( c \) is not degenerate, it follows that \( B^n(e') \cup B^n(e'') \) is an interval. Clearly, \( B^n(f') \) and \( B^n(f'') \) are intervals. By the above fact, there are \( i' \in B^n(f') \) and \( i'' \in B^n(f'') \) such that \( |i' - i''| \leq 1 \). Clearly, \( B^n(f') \cup B^n(f'') \) is an interval. By the induction hypothesis, \( \mathcal{O}(f') \) and \( \mathcal{O}(f'') \) are intervals. Again by the above fact, \( \mathcal{O}(f') \cup \mathcal{O}(f'') \) is an interval. Clearly, \( \mathcal{O}(e') = \mathcal{O}(e'') = \mathcal{O}(f') \cup \mathcal{O}(f'') \).

Note that Lemma 7 holds for every comparator network with no concern to the network’s functionality. Lemma 8, on the other hand, refers to Bitonic sorters and does not hold in general. Recall that our goal is to prove Theorem 1; that is, to show that \( D^n > \lfloor \log(n) \rfloor \). We need Lemma 8 to this end but we use it much later.

A major tool in our analysis is the following concept of span. Let \( e \) be an edge of a comparator network \( N \). The span of \( e \), denoted \( S(e) \), is defined by \( S(e) \triangleq |\mathcal{I}(e)| \cdot |\mathcal{O}(e)| \). The following lemma is straightforward.
Lemma 9 Let \( e \) be an edge of a comparator network. Then a directed path having \( \lceil \log(\mathcal{S}(e)) \rceil \) comparators passes through \( e \).

Lemma 10 Let \( e \) be an edge of a Bitonic sorter of width \( n \). Then \( \mathcal{S}(e) \geq n \).

\textbf{Proof:} Let \( I = \mathcal{I}(e) \). The set \( I \) partitions the circle \( C^n \) into \( |I| \) disjoint segments; each segment starts at a member of \( I \) and ends just before the next member of \( I \). Clearly, the length of one of these segments is at least \( \lceil n/|I| \rceil \). Let \( s \) be such a segment.

Let \( U \subset U^n \) be the set of Unitonic permutations in which the maximal key, \( n \), is located in \( s \). Every comparator which leads to \( e \) performs the same routing for all members of \( U \). That is, there is an index \( x \in C^n \) such that \( u(e) = u_x \), for every \( u \in U \). This implies that \( u'(e) \neq u''(e) \) for every \( u' \) and \( u'' \), distinct members of \( U \). Hence \( |U^n(e)| \geq |U(e)| = |U| = |s| \geq n/|I| \). This clearly implies that \( |\mathcal{O}(e)| \geq n/|I| \). Hence, \( \mathcal{S}(e) = |\mathcal{O}(e)| \cdot |I| \geq n. \)

Lemmas 9 and 10 imply the weak inequality, \( \mathcal{D}(n) \geq \lceil \log(n) \rceil \). In order to prove the strong inequality, \( \mathcal{D}(n) > \lceil \log(n) \rceil \), it remains to show the following lemma.

Lemma 11 Let \( B \) be a Bitonic sorter whose width is not a power of two and assume \( B \) has no degenerate comparators. Then \( B \) has two edges \( e' \) and \( e'' \) such that \( \mathcal{S}(e'') \geq 2 \cdot \mathcal{S}(e') \).

\textbf{Proof:} Let \( p \) be the output port indexed by 1. Consider the subgraph \( G \) of \( B \) composed of all comparators, edges and input ports that have a path to \( p \). Since the width of \( B \) is not a power of two, \( G \) is not a balanced tree. Hence, \( G \) has (at least one) comparator \( c \) with incoming edges \( e' \) and \( e'' \) and outgoing edges \( f' \) and \( f'' \), for which one (or both) of the followings conditions hold:

1. \( d^I(e') \neq d^I(e'') \).
2. \( f' \) and \( f'' \) belong to \( G \).

We refer to such a comparator as a \textit{bad} comparator. Let \( \bar{c} \) be a minimal bad comparator; that is, \( \bar{c} \) is not reachable from any other bad comparator.

First assume that Condition (1) holds w.r.t. \( \bar{c} \). Assume, without loss of generality, that \( d^I(e'') > d^I(e') \). The minimality of \( \bar{c} \) implies that \( |\mathcal{I}(e'')| = 2^{d^I(e'')} \geq 2 \cdot 2^{d^I(e')} = 2 \cdot |\mathcal{I}(e')| \). Clearly, \( |\mathcal{O}(e')| = |\mathcal{O}(e'')| \). This implies that \( \mathcal{S}(e'') \geq 2 \cdot \mathcal{S}(e') \).

Next assume that Condition (1) does not hold; hence, Condition (2) holds. By Lemma 8, \( \mathcal{O}(f') \) and \( \mathcal{O}(f'') \) are intervals. By construction, \( 1 \in \mathcal{O}(f') \) and \( 1 \in \mathcal{O}(f'') \); hence, one interval is a subset of the other. Say, \( \mathcal{O}(f') \subseteq \mathcal{O}(f'') \). By the minimality of \( \bar{c} \) and the fact that Condition (1) does not hold, we get that \( |\mathcal{I}(f'')| = 2 \cdot |\mathcal{I}(e'')| \). However \( \mathcal{O}(f'') = \mathcal{O}(e'') \). This implies that \( \mathcal{S}(f'') = 2 \cdot \mathcal{S}(e'') \).

Clearly, a degenerate comparator of a Bitonic sorter can be removed without disturbing the network’s functionality and without increasing its depth. This fact, together with Lemmas 9,10 and 11, imply Theorem 1.

\textbf{References}


