February, 1983

THE BITONIC AND ODD-EVEN NETWORKS ARE MORE THAN MERGING

Yehoshua Perl

DCS-TR-123

Department of Computer Science
Hill Center for the Mathematical Sciences
Busch Campus, Rutgers University
New Brunswick, New Jersey 08903

and

Bar-Ilan Univeristy
Ramat-Gan, Israel

ABSTRACT

The known bitonic and odd-even merging networks are reinvestigated. For both networks the following results are obtained:

The input vectors sorted by the network are characterized. Those vectors are recursively balanced for some definition of balance. The set of those vectors is much larger than the set of vectors known to be sorted by the network. The output vectors obtainable by applying the network to an arbitrary input vector are also characterized. Those vectors satisfy recursive dominance for some definition of dominance.

1. INTRODUCTION

The bitonic and odd−even networks were introduced by Batcher [B] as merging networks. Both networks require O(lg n) time to merge two sorted lists of n/2 elements into a sorted list of $n=2^k$ elements. The bitonic (odd−even) merging network is the basic component in the bitonic (odd−even) sorting network which requires $O(\lg^2 n)$ time. Stone [S] implemented the bitonic sorting network on the shuffle exchange interrconnection model such that only one layer of comparators is required. Hong and Sedgewick [HS] consider the bitonic and odd−even networks as (m,n) merging networks for $m \neq n$. For a review on merging networks and sorting networks see Knuth [K].

A bitonic sequence is obtained by concatenating in any order two monotonic sequences one nonincreasing and one nondecreasing. Two examples of bitonic sequences are: (1, 3, 5, 7, 6, 4, 2) and (8, 6, 4, 3, 2, 1, 5, 7).

The basic unit of the network is, as usual, a two input, two output comparator transforming the two input elements in arbitrary order into nondecreasing order. Let x be a bitonic input vector of $n=2^k$ elements for the bitonic network. Assume the index set of x is {0,1, ..., n−1}. The first layer of the 2^k−bitonic network consists of n/2 comparators comparing x(i) and x(n/2+i), $0 \le i < n/2$. The first layer partitions the elements of the bitonic vector x into two bitonic sequences of the n/2 smaller elements in the lower half and the n/2 larger elements in the upper half. These two bitonic sequences are further sorted by applying two 2^{k-1}−bitonic networks to each half of the vector. A 16−bitonic network appears in Figure 1 where horizontal lines represent the input lines and vertical lines represent comparisons between the elements on the corresponding input lines. Note that the bitonic network is a merging network since the two monotinic sequences concatenated into a bitonic sequence are merged into one sorted sequence.

The input for the odd−even network is a vector of $n=2^k$ elements such that each of its two halves contains a nondecreasing order sequence. The 2^k−odd−even network consists of two 2^{k-1}−odd−even networks applied separately to the even elements and to the odd elements of the vector, followed by an extra layer comparing the $(2i-1)^{th}$ and the $2i^{th}$ elements, $1 \le i < n/2$. A 16−odd−even network appears in Figure 2.

In this work we reinvestigate the bitonic and odd−even networks from a different point of view. We ask the following two questions:

1. What are the input vectors which are sorted by the network?
2. What is the effect of applying the network to an arbitrary input vector?

By the zero−one principle [K] it is sufficient to analyze sorting networks for binary input. However our analysis for a general input gives more insight.

In Section 2 we characterize the input vectors sorted by the bitonic network. This set of vectors is shown to be much larger than the set of bitonic vectors. In Section 3 we

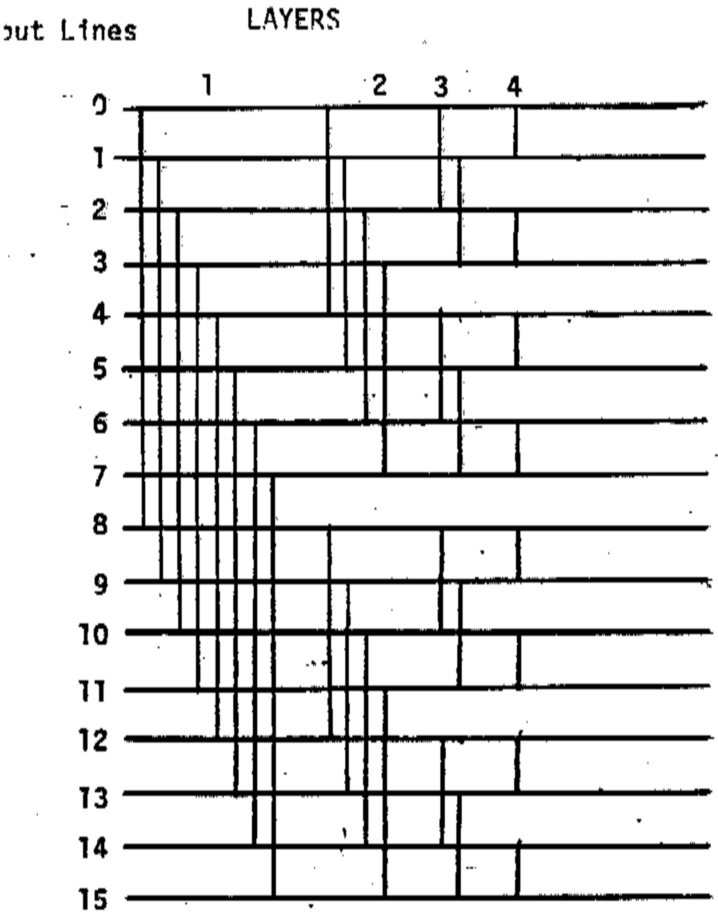

FIGURE 1

The bitonic network

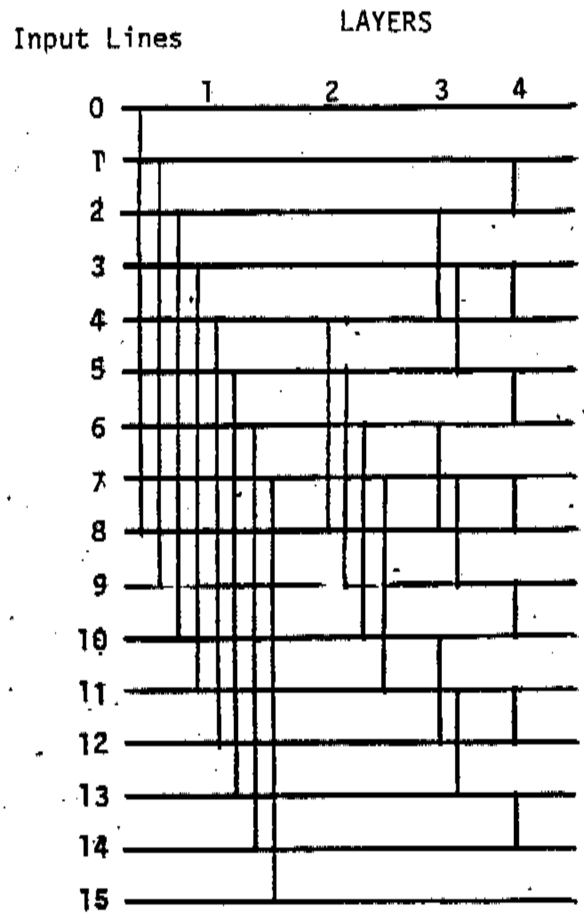

FIGURE 2

The odd-even network

characterize the output vectors obtainable by applying the bitonic network to an arbitrary input.

The vectors sorted by the odd-even network are characterized in Section 4. This set of vectors is shown to be much larger than set of vectors which were known to be sorted by the odd-even network. However, this set is smaller than the set of vectors sorted by the bitonic network. In Section 5 we characterize the output vectors obtainable by applying the odd-even network to an arbitrary input.

No practical applications of these characterizations are known to us. However the insight provided by such characterizations may help in the design and analysis of new sorting networks. For example, if this insight had been known before, it could have helped in the difficult analysis of the balanced sorting network [DPRS].

2. THE BITONIC NETWORK SORTS RECURSIVELY MODULO BALANCED VECTORS

We consider a vector of $n=2^k$ elements and assume its index set is $\{0,1,2,...,n-1\}$. The even chain (odd chain) of a vector is the subvector of the elements with even (odd) indices. The even chain and the odd chain are the level 1 chains of the vector. The vector itself is the level 0 chain. In general, the set of level i chains of a vector, $0 \le i \le k$, consists of 2^i subvectors of 2^{k-i} elements each, containing the elements whose indices are equal modulo 2^i. Note that if we represent the indices in their binary representation, then the indices of a level i chain have the same rightmost i bits.

Each level i chain, $0 \le i < k$, consists actually of two level i+1 chains. The level i+1 chain containing the first (second) element of the level i chain is the even (odd) chain of the level i chain. The level i chains, $0 \le i < 4$, of a vector of 16 elements are depicted in Figure 3.

A vector is modulo balanced if, when the elements are arranged in nondecreasing order in pairs, one element from each pair lies in the even chain of the vector and the other element lies in the odd chain of the vector. Equivalently, a vector is modulo balanced if for each i, $1 \le i < n/2$, the i^{th} largest element of the even (odd) chain is larger than or equal to the $(i+1)^{th}$ largest element of the odd (even) chain.

A vector is recursively modulo balanced if it is modulo balanced and both its even and odd chains are recursively modulo balanced. Clearly any vector of 2 elements is recursively modulo balanced. Note that in a recursively modulo balanced vector all the level i chains, $0 \le i < k$, of the vector are balanced. A recursively modulo balanced vector appears in Figure 3.

A similar definition of a recursively balanced vector, according to a different kind of chains, appears in the analysis of the balanced sorting network [DPRS]. The term modulo balanced was chosen to differentiate between the two definitions. However for convenience we shall use from now on the terms balanced and recursively balanced for modulo balanced and recursively modulo balanced, respectively.

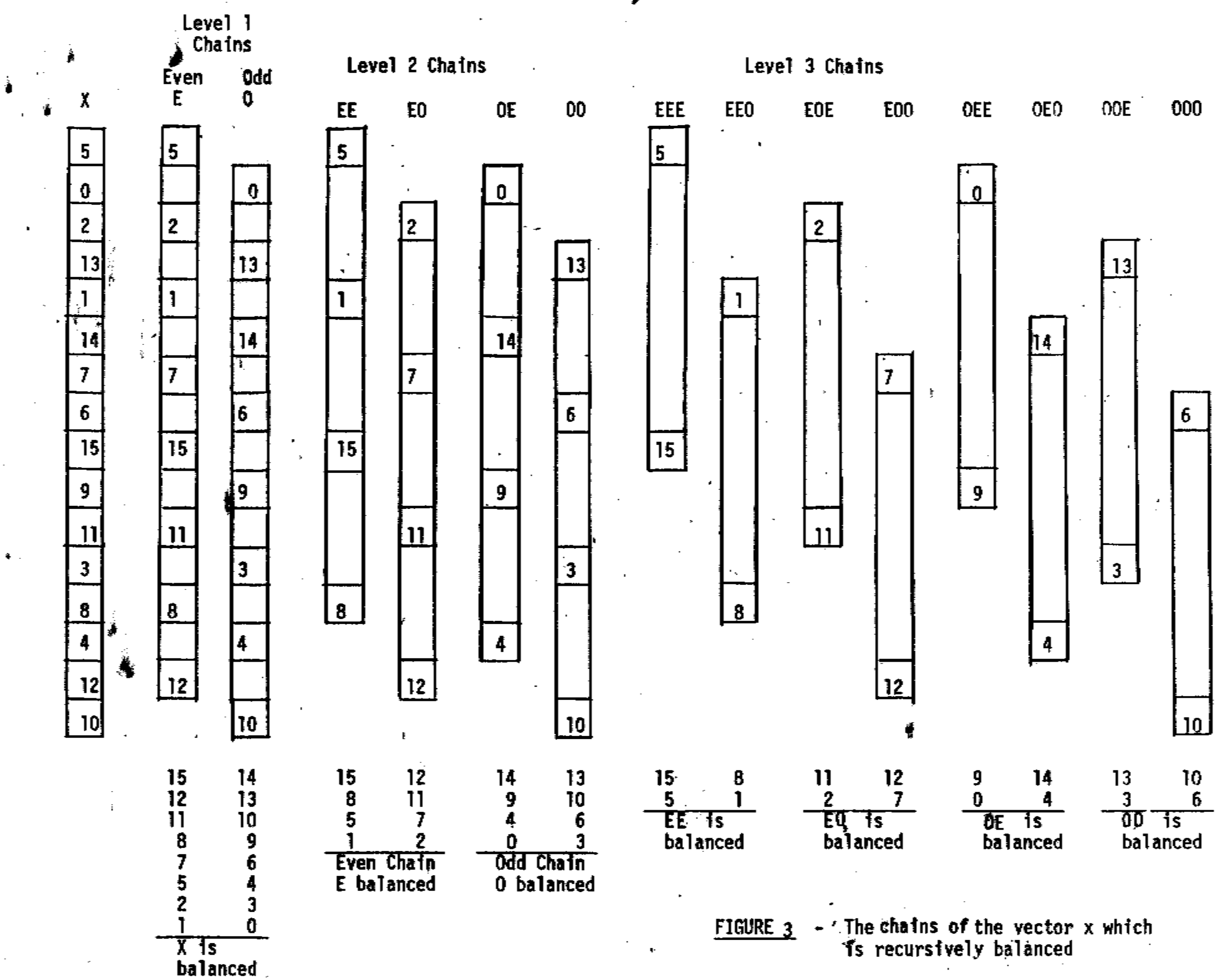

FIGURE 3 - The chains of the vector x which is recursively balanced

In the following proofs we use the following straightforward observations.

Proposition 1: In the first k−1 layers of the 2^k−bitonic (odd−even) network there is no comparison between the elements of the even chain and the elements of the odd chain of the vector.

Proposition 2: The first k−1 layers of the 2^k−bitonic (odd−even) network consist of two independent 2^{k-1}−bitonic (odd−even) networks applied separately to the even and odd chains of the input vector.

Lemma 3: The 2^k−bitonic network sorts any recursively balanced vector of $n=2^k$ elements.

Proof: By induction on k. The lemma is true for k=1. Assume the lemma is true for k−1 and prove it for k.

Let x be a recursively balanced input vector. Let y be the output vector obtained from x after applying the first k−1 layers of the bitonic network. By definition the even and odd chains of x are both recursively balanced. Thus, by Proposition 2 and the induction hypothesis the even and odd chains of y are both sorted. By Proposition 1 the sets of elements of the even (odd) chains of x and y are equal. Hence y is balanced since x is balanced. These two observations imply that:

$$y_0, y_1 \le y_2, y_3 \le \ldots \le y_{n-2}, y_{n-1}.$$

Hence the last layer which compares y_{2i} and y_{2i+1}, $0 \le i < n/2$ yields a sorted output ∎

This proof is quite simple. However it does not provide the full insight for the operation of the bitonic network on a recursively balanced vector. Such an insight is obtained by the next proposition which is a little more difficult to prove.

Proposition 4: The application of the first layer of the bitonic network to a recursively balanced vector of 2^k elements yields a vector in which every element in the lower half is less than or equal to every element in the upper half and both halves are recursively balanced.

Proposition 4 immediately implies Lemma 3. The proof of the proposition is left to the reader.

Lemma 5: The 2^k−bitonic network, k≥2, does not sort a vector of $n=2^k$ elements which is not recursively balanced.

Proof: By induction on k, k≥2. For k=2 the proof is simple. Assume the lemma is true for k−1 and prove it for k.

Let x be a vector of 2^k element which is not recursively balanced. By definition either x is not balanced or at least one of its level 1 chains is not recursively balanced. Let y be the vector obtained after the first $k-1$ layers are applied to x.

Assume first that x is not balanced. Then if the elements of x are arranged in nondecreasing order in pairs, then there exists a pair (a,b) whose both elements belong to one of the level 1 chains, say the even chain. By Proposition 1 the elements a and b belong to the even chain of y. The last layer of the bitonic network compares y_{2i} with y_{2i+1}, $0 \le i < n/2$. Thus the elements a and b are not a pair in the arrangement of the output vector into pairs and thus the vector is not sorted.

Assume now that x is balanced but one of its level 1 chains, say the even chain is not recursively balanced. Then by Proposition 1 and the induction hypothesis the even chain of y is not sorted. That is there is at least one pair of the even chain of y which is out of order. After the application of the last layer to the vector y this pair is still out of order. Hence the output vector is not sorted. ■

Lemma 3 and 5 imply the following characterization.

Theorem 6: The 2^k-bitonic network sorts a vector of 2^k elements if and only if it is recursively balanced.

A new proof for the known Theorem 8 that the bitonic network sorts a bitonic vector is implied by Lemma 3 and the following lemma.

Lemma 7: A bitonic vector x of $n = 2^k$ elements is recursively balanced.

Proof: By induction on k. For $k = 2$ the proof is trivial. Assume the lemma is true for $k-1$ and prove it for k.

The even chain and the odd chain of a bitonic vector are bitonic subvectors. Thus by the induction hypothesis the odd chain and the even chain of x are recursively balanced. Thus it is left to show that x is balanced. We consider the case that the bitonic vector x is first increasing and then decreasing. The proof for the other case is similar. The two smallest elements of x are either x_0, x_1 or x_{t-2}, x_{t-1} or x_0, x_{t-1}. In each case one element of the smallest pair of x belongs to the even chain while the other element belongs to the odd chain. The same applies for the following pairs since the rest of x is a bitonic vector. Hence x is balanced. ■

Lemmas 3 and 7 imply:

Theorem 8: The 2^k-bitonic network merges two sorted lists, one nondecreasing of length m and the other nonincreasing of length n if $m + n = 2^k$.

Theorem 8 implies that any two sorted lists of length m and n can be merged by the 2^k-bitonic network for the lowest k such that $m + n \le 2^k$, by arranging the input vector to start with the first list of m elements in nonincreasing order, followed by the second list of n elements in nondecreasing order, followed by $2^k-(m+n)$ ∞ values. It is known (see e.g. [K] and [HS]) that an (m,n) merging network can be obtained from the above bitonic network by removing all comparators "incident" with the input lines associated with ∞ values.

The bitonic network was known to sort any bitonic input. The first monotonic subsequence of i elements, $0 \le i \le n$, in a bitonic vector is either increasing or decreasing. Thus the number of bitonic permutations of order n is

$$2 \sum_{i=0}^{n} \binom{n}{i} = 2 \cdot 2^n = 2^{n+1}$$

The following theorem shows that the bitonic network sorts actually many more permutations of order n.

Theorem 9: The number of recursively balanced permutations of order $n=2^k$, is $n^{n/2}$.

Proof: Let f(n) denote the number of recursively balanced permutations of order n. A permutation is recursively balanced if it is balanced and its even and odd chains are both recursively balanced. If we arrange the elements in increasing order in pairs, then the permutation is balanced if exactly one element from each pair is in the even chain. Thus there are $2^{n/2}$ ways the elements of the even chain can be selected. Once this is done there are f(n/2) ways to arrange the even chain and f(n/2) ways to arrange the odd chain since both chains are recursively balanced.

Hence

$$f(1) = 1$$
$$f(n) = 2^{n/2} \, [f(n/2)]^2$$

It is easy to show that $f(n) = n^{n/2}$. ■

3. THE OUTPUT OF THE BITONIC NETWORK SATISFIES RECURSIVE DOMINANCE

In this section we characterize the output vectors of the bitonic network. Let x be a vector of $n=2^k$ elements in which the level k−1 chains are sorted, i.e. $x_i \le x_{i+n/2}$, $0 \le i < n/2$. Then we say that x satisfies dominance or that the upper half of x dominates the lower half of x. The vector x satisfies recursive dominance if it satisfies dominance and its lower and upper halves each satisfy recursive dominance. E.g. the vector (1,4,2,7,3,6,5,8) satisfies recursive dominance.

Lemma 10: Let x be a vector which satisfies dominance. Let y be the vector obtained from x by applying the two comparators between x_i and x_j and between $x_{i+n/2}$ and $x_{j+n/2}$ for i and j such that $0 \le i < j < n/2$. Then y satifies dominance.

Proof: We consider the case $x_i \le x_j$, and $x_{i+n/2} \ge x_{j+n/2}$. The proof for the other cases is similar and is left to the reader.

In this case $y_i=x_i$, $y_j=x_j$, $y_{i+n/2}=x_{j+n/2}$, $y_{j+n/2}=x_{i+n/2}$ and $y_m=x_m$ for all other indices.

$$y_i=x_i\le x_j\le x_{j+n/2}=y_{i+n/2}$$

$$y_j=x_j\le x_{j+n/2}\le x_{i+n/2}=y_{j+n/2}$$

Hence y satisfies dominance. ∎

Lemma 11: The output vector y of the 2^k-bitonic network satisfies recursive dominance for any input vector x.

Proof: By induction on k. For k=1 the proof is trivial. Assume the lemma is true for k-1 and prove it for k.

The vector z obtained by the application of the first layer of the bitonic network to the input x satisfies dominance. The next k-1 layers of the 2^k-bitonic network consist of two 2^{k-1}-bitonic networks applied separately to the lower and upper halves of z, respectively. Thus by the induction hypothesis the lower and upper halves of y each satisfy recursive dominance. Lemma 10 implies that y satisfies dominance since z satisfies dominance. Hence y satisfies recursive dominance. ∎

Theorem 12: A vector y can be an output of the bitonic network if and only if it satisfies recursive dominance.

Proof: Lemma 11 implies one side of the proof. For the other side note that if the bitonic network is applied to any vector y which satisfies recursive dominance it yields the vector y itself. ∎

It is interesting to note that the same characterization was found by S.A. Cook for the balanced merging network [DPRS]. However one difference between the two networks is that if the balanced merging network is applied to a vector which satisfies recursive dominance it yields a different vector and not the same vector as for the bitonic network.

4. THE ODD-EVEN NETWORK SORTS RECURSIVELY SHIFT-BALANCED VECTORS

As before we consider a vector of $n=2^k$ elements. A vector is shift-balanced if its smallest element belongs to the even chain, its largest element belongs to the odd chain and when the rest of the elements of the vector are arranged in nondecreasing order in pairs, one element from each pairs lies in the even chain and the other element lies in the odd chain. Equivalently, a vector is shift-balanced if the i^{th} largest element of the odd chain, $1\le i\le n/2$, is not smaller than the i^{th} largest element of the even chain and the i^{th} largest element of the even chain, $1\le i\le n/2-2$ is not smaller than the $(i+2)^{th}$ largest element of the odd chain.

A vector is recursively shift-balanced if it is shift-balanced and both its even and odd

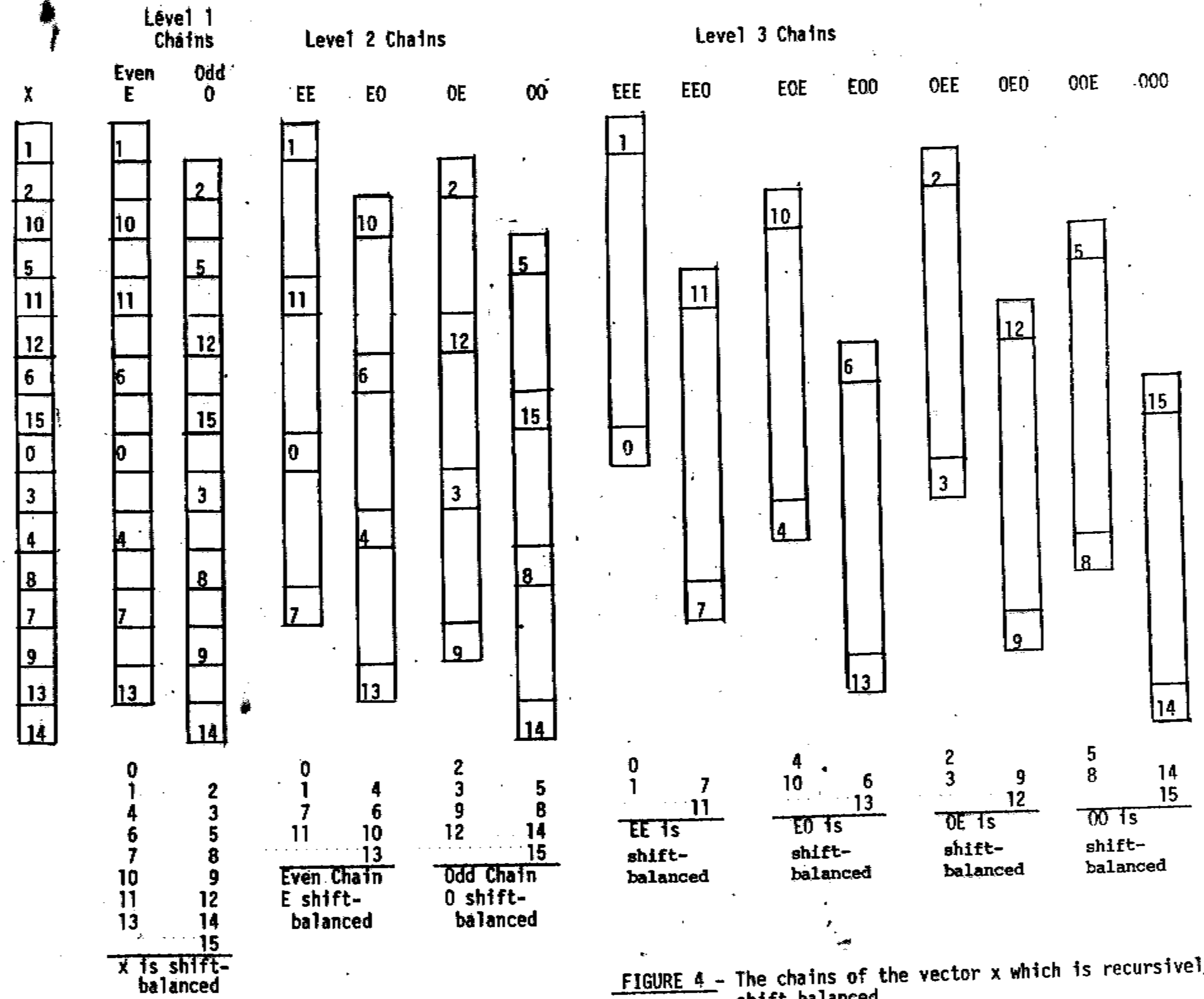

FIGURE 4 - The chains of the vector x which is recursively shift-balanced

chains are recursively shift-balanced, where <u>any</u> vector of two elements is <u>called</u> recursively shift-balanced. Note that in a recursively shift-balanced vector all level i chains $0 \le i \le k-2$ are shift-balanced. A recursively shift-balanced vector appears in Figure 4.

Recall that Propositions 1 and 2 are stated for the odd-even network too.

<u>Lemma 13</u>: The 2^k-odd-even network sorts any recursively shift-balanced vector of $n=2^k$ elements.

<u>Proof</u>: By induction on k. The case k=1 is trivial. Assume the lemma is true for k-1 and prove it for k.

Let x be a recursively shift-balanced vector. Let y and z be the vectors obtained by applying to x the first k-1 and k layers of the odd-even network, respectively. The even and odd chains of x are recursively shift-balanced. By Proposition 2 and the induction hypothesis the even and odd chains of y are sorted. By Propositiopn 1 the sets of the elements of the even (odd) chains of x and y are equal. Thus y is shift-balanced and y_o and y_{n-1} are the smallest and largest elements of y, respectively. The last layer of the odd-even network compares y_{2i-1} and y_{2i}, $1 \le i < n/2$. Thus by the definition of shift-balanced z is sorted. ∎

<u>Lemma 14</u>: The 2^k-odd-even network, $k \ge 2$, does not sort a vector which is not recursively shift-balanced.

<u>Proof</u>: By induction on k. For k=2 the proof is trivial. Assume the lemma is true for k-1 and prove it for k.

Let x be a vector of 2^k elements which is not recursively shift-balanced. Let y and z be the vectors obtained by applying to x the first k-1 and k layers of the odd-even network, respectively. By definition either x is not shift-balanced or at least one of its level 1 chains is not recursively shift-balanced.

Assume first x is not shift-balanced. By the definition we have to consider three possibilities. If the smallest element of x does not belong to the even chain the x can not be sorted by the odd-even network since the only comparison of x_o is with $x_{n/2}$ in the first layer. The proof for the case that the largest element of x does not belong to the odd chain is similar. The third possibility is that when arranging the rest of the elements of x, in nondecreasing order, in pairs there exist a pair (a,b) whose both elements belong to the same chain, say the even chain. By Proposition 1 the pair (a,b) belongs to the even chain of y, i.e. $a=y_{2p}$ $b=y_{2q}$, $1 \le p,q < n/2$. The last layer of the odd-even network compares y_{2i-1} and y_{2i}, $1 \le i < n/2$. Thus the elements a and b can not be the $(2i-1)^{th}$ and $2i^{th}$ elements of z for some i, $1 \le i < n/2$. Hence z is not sorted.

Assume now that at least one of the level 1 chains, say the even chain, is not recursively shift-balanced. Proposition 2 and the induction hypothesis imply that the even chain of y is not sorted. That is, there is at least one pair of the even chain of y which is out of order. After applying the last layer of the odd-even network this pair is still out of order. Hence the output z is not sorted. ■

Lemmas 13 and 14 imply the following characterization.

<u>Theorem</u> <u>15</u>: The 2^k-odd-even network sorts a vector of 2^k elements, $k \geq 2$, if and only if it is recursively shift-balanced.

A new proof for the known Theorem 17 that the odd-even network is a merging network follows from Lemma 13 and the following lemma.

<u>Lemma</u> <u>16</u>: A sequence x obtained by concatenating two nondecreasing sequences of 2^{k-1} elements is recursively shift-balanced.

<u>Proof</u>: By induction on k. For or k=2 the proof is trivial. Assume the lemma is true for k-1 and prove it for k.

The even chain and the odd chain of x are each obtained by concatenating two nondecreasing sequences of 2^{k-2} elements. By the induction hypothesis the even and odd chains of x are recursively shift-balanced.

To complete the proof it is left to show that x is shift-balanced. Clearly the smallest (largest) element of x belongs to the even(odd) chain of x. If the rest of the elements of x are arranged in nondecreasing order in pairs then one element of the first pair belongs to the even chain and the other element belongs to the odd chain. The same applies to all following pairs. Hence, x is shift-balanced and recursively shift-balanced. ■

Lemma 13 and 16 imply:

<u>Theorem</u> <u>17</u>: The 2^k-odd-even network merges two sorted lists of 2^{k-1} elements.

This known theorem implies that the odd-even network was known to sort $\binom{n}{n/2} = 0(2^n/n^{1/2})$ permutations of order $n=2^k$. The following theorem shows that the odd-even network sorts actually many more permutations of order n.

<u>Theorem</u> <u>18</u>: The number of recursively shift-balanced permutations of order $n=2^k, k \geq 1$, is $2(n/2)^{n/2}$.

<u>Proof</u>: Let g(n) denote the number of recursively shift-balanced permutations of order n. A permutation is recursively shift-balanced if it is shift-balanced and its even and odd chains are both recursively shift-balanced. Thus the smallest element belongs to the even chain, the largest element belongs to the odd chain and if the rest of the elements are arranged in increasing order in pairs then exactly one element of each pair belongs to the even chain. Thus there are $2^{n/2-1}$ ways to select the elements of the even chain. Once this is done there are g(n/2) ways to arrange the even chain and g(n/2) ways to arrange the

odd chain since both chains are recursively shift-balanced.

Hence

$$g(2) = 2$$
$$g(n) = 2^{n/2-1} \, [g(n/2)]^2$$

It is easy to show that $g(n)=2(n/2)^{n/2}$. ∎

Note that the bitonic networks sorts more permutations than the odd-even network.

$$f(n)/g(n) = n^{n/2}/(2(n/2)^{n/2}) = 2^{n/2-1}$$

This is expected since the bitonic network contains more comparators than the odd-even network and thus it is more powerful.

5. THE OUTPUT VECTORS OF THE ODD-EVEN NETWORK

A vector y of $n=2^k$ elements satisfies pair dominance if $y(2i-1) \le y(2i)$, $1 \le i < n/2$. Note that pair dominance has no requirements for the extreme elements y(0) and y(n−1). The vector obtained by applying the last layer of the odd-even network clearly satisfies pair dominance.

A vector y of $n=2^k$ elements satisfies double dominance if it satisfies both dominance and pair dominance. A vector y of $n=2^k$ elements satisfies recursively reachable double dominance if there exists a vector z obtainable from y by performing exchanges of y(2i−1) and y(2i) for some indices i, $1 \le i < n/2$, such that both the even and odd chains of z satisfy double dominance and recursively reachable double dominance.

Lemma 19: For any input vector x of $n=2^k$ elements, the output vector y of the 2^k-odd-even network applied to x satisfies double dominance and recursively reachable double dominance.

Proof: By induction on k. The proof for k=2 is trivial. Assume the lemma is true for k−1 and prove it for k.

Let z denote the vector obtained by applying to x the first k−1 layers of the odd-even network. By proposition 2 and the induction hypothesis both the even and odd chains of z satisfy double dominance and recursively reachable double dominance. The vector z satisfies dominance since its even and odd chain satisfy dominance.

The last layer of the odd-even network compares z(2i−1) and z(2i), $1 \le i < n/2$. Thus y satisfies pair dominance.

Also

$$y(0) = z(0) \le z(n/2) \le \max(z(n/2), z(n/2-1)) = y(n/2)$$

$$y(n/2-1) = \min(z(n/2-1), z(n/2)) \le z(n/2-1) \le z(n-1) = y(n-1)$$

By Lemma 10 $y(i) \le y(n/2+i)$, $1 \le i \le n/2-2$, since z satisfies dominance. Hence y satisfies dominance and thus double dominance.

The vector y satisfies also recursively reachable double dominance since the above vector z can be obtained from y by exhcanging back the pairs exchanged by the last layer and the even and odd chain of z both satisfy double dominance and recursively reachable double dominance. ∎

Lemma 20: Let y be a vector of $n=2^k$ elements which satisfies double dominance and recursively reachable double dominance then there exists a vector x such that y is obtained by applying to x the 2^k-odd-even network.

Proof: By induction on k. The proof for k=2 is trivial. Assume the lemma is true for k−1 and prove it for k.

The vector y satisfies recursively reachable double dominance. Thus there exists a vector z obtainable from y by performing exchanges of $y(2i-1)$ and $y(2i)$ for some indices i, $1 \le i < n/2$, such that both the even chain z_0 and the odd chain z_1 of z satisfy double dominance and recursively reachable double dominance. By the induction hypothesis there exists a vector $x_0(x_1)$ of 2^{k-1} elements such that applying the 2^{k-1}-odd-even network to $x_0(x_1)$ yields $z_0(z_1)$. Let x be the vector whose even chain is x_0 and odd chain is x_1. By Proposition 2 applying the first k−1 layers of the 2^k-odd-even network to x yield z. Applying the last layer of the 2^k-odd-even network to z yields y since y satisfies pair dominance. Hence applying the 2^k-odd-even network to x yields y. ∎

Lemmas 19 and 20 imply:

Theorem 21: A vector y of 2^k elements can be an output of the odd-even network if and only if it satisfies double dominance and recursively reachable double dominance.

There are some vectors which satisfy double dominance and recursively reachable double dominance for which no exchange occurs while applying to them the odd-even network. Thus as for the bitonic network, repeated application of the odd-even network does not necessarily yield a sorted vector. On the other hand applying the 2^k-balanced merging network [DPRS] k times sorts any vector of 2^k elements.

REFERENCES

[Ba] K. Batcher, "Sorting networks and their applications," AFIPS Spring Joint Comp. Conf. 32 (1968) 307–314.

[DPRS] M. Dowd, Y. Perl, L. Rudolph and M. Saks, "The balanced sorting network," Rutgers University LCSR Technical Report.

[HS] Z. Hong and R. Sedgewick, "Notes on Merging Networks," Proceedings of the Fourteen Annual ACM Symposium on Theory of Computing, San Francisco 1982, 296–302.

[K] D.E. Knuth, The art of computer programming, Vol. 3: Sorting and searching, Addison Wesley, Reading, MA, 1973.

[S] H.S. Stone, "Parallel processing with the perfect shuffle," IEEE Transaction on Computing, C-20,2 (1971)