Periodic Multisorting Comparator Networks*

Marcin Kik
kik@im.pwr.wroc.pl

Institute of Mathematics, Wrocław University of Technology
ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Abstract. We present a family of periodic comparator networks that transform the input so that it consists of a few sorted subsequences. The depths of the networks range from 4 to $2\log n$ while the number of sorted subsequences ranges from $2\log n$ to 2. They work in time $c\log^2 n + O(\log n)$ with $4 \le c \le 12$, and the remaining constants are also suitable for practical applications. So far, known periodic sorting networks of a constant depth that run in time $O(\log^2 n)$ (a periodic version of AKS network [7]) are impractical because of complex structure and very large constant factor hidden by big "Oh".

Keywords: sorting, comparator networks, parallel algorithms.

1 Introduction

Comparator is a simple device capable of sorting two elements. Many comparators can be connected together to form a comparator network. This way we get the classical framework for sorting algorithms. Optimal arranging the comparators turned out to be a challenge. The main complexity measures of comparator networks are time complexity (depth or number of steps) and the number of comparators. The most famous sorting network is AKS network with asymptotically optimal depth $O(\log n)$ [1], however the big constant hidden by big "Oh" makes it impractical. The Batcher networks of depth $\approx \frac{1}{2}\log^2 n$ [2], seem to be very attractive for practical applications.

A *periodic* network is repeatedly used on the intermediate results until the output becomes sorted, thus the same comparators are reused many times. In this case, the time complexity is the depth of the network multiplied by the number of iterations. The main advantage of periodicity is the reduction of the amount of hardware (comparators) needed for the realization of the sorting algorithm, with a very simple control mechanism providing the output of one iteration as the input for the next iteration. Dowd et al, [3], reduced the number of comparators from $\Omega(n\log^2 n)$ to $\frac{1}{2}n\log n$, while keeping the sorting time $\log^2 n$, by the use of a periodic network of depth $\log n$. (The networks of depth d have at most $dn/2$ comparators.) There are some periodic sorting networks of a constant depth ([10], [5], [7]). In [7], constant depth networks with time complexity $O(\log^2 n)$ are

* research supported by KBN grant 7T11C 3220 in the years 2002, 2003

obtained by "periodification" of the AKS network, and more practical solutions with time complexity $O(\log^3 n)$, are obtained by "periodification" of the Batcher network. On the other hand there is not known any $\omega(\log n)$ lower bound on the time complexity of periodic sorting networks of constant depth. Closing the gap between the known upper bound of $O(\log^2 n)$ and the trivial general lower bound $\Omega(\log n)$ seems to be a very hard problem.

Periodic networks of constant depth can also be used for simpler tasks, such as merging sorted sequences [6], or resorting sequences with few values modified [4].

1.1 New Results

We assume that the values are stored in the registers and the only allowed operations are *compare-exchange* operations (*applications of comparators*) on the pairs of registers. Such an operation takes the two values stored in the pair of registers and stores the lower value in the first register and the greater value in the second register. (This interpretation differs from the one presented for instance in [8] but is more useful when periodic comparator networks are concerned.)

We present a family of periodic comparator networks $N_{m,k}$. The input size of $N_{m,k}$ is $n = 4m2^k$. The depth of $N_{m,k}$ is $2\lceil k/m \rceil + 2$. In Section 4 we prove the following theorem.

Theorem. *The periodic network $N_{m,k}$ transforms the input into $2m$ sorted subsequences of length $n/(2m)$ in time $4k^2 + 8km + O(k + m)$.*

For example, the network $N_{1,k}$ is a network of depth $\approx 2\log n$ that produces 2 sorted sequences in time $\approx 4\log^2 n + O(\log n)$. On the other hand, $N_{k,k}$ is a network of depth 4 that transforms the input into $\approx 2\log n$ sorted sequences in time $\approx 12\log^2 n + O(\log n)$. Due to the large constants in the known periodic constant depth networks sorting in time $O(\log^2 n)$, [7], it could be interesting alternative to use $N_{k,k}$ to produce very much ordered (although not completely sorted) output.

The output produced by $N_{m,k}$ can be finally sorted by a network merging $2m$ sequences. This can be performed by the very efficient multiway merge sorting networks [9]. It is an interesting problem to find efficient periodic network of constant depth that merges multiple sorted sequences. The periodic networks of constant depth that merge two sorted sequences in time $O(\log n)$ are already known [6].

As $N_{m,k}$ outputs multiple sorted sequences, we call it a *multisorting* network. Much simpler multisorting networks of constant depth exist if some additional operations are allowed (such as permutations of the elements in the registers between the iterations). However, we consider only the case restricted to the compare-exchange operations.

2 Preliminaries

By a *comparator network* we mean a set of *registers* $R_0, \dots, R_{n-1}$ together with a finite sequence of *layers of comparators*. Every moment a register R_i contains a single value (denoted by $v(R_i)$) from some totally ordered set, say $\mathbb{N}$. We say that the network stores a sequence $v(R_0), \dots, v(R_{n-1})$. A subset S of registers is *sorted* if for all R_i, R_j in S, $i < j$ implies that $v(R_i) \le v(R_j)$. A *comparator* is denoted by an ordered pair of registers (R_i, R_j). If $v(R_i) = x$ and $v(R_j) = y$ before an *application* of the comparator (R_i, R_j), then $v(R_i) = \min\{x, y\}$ and $v(R_j) = \max\{x, y\}$ after the application of (R_i, R_j). A set of comparators L forms a *layer* if each register is contained in at most one of the comparators of L. So all the comparators of a layer can be applied simultaneously. We call such application a *step*. The *depth* of the network is the number of its layers. An *input* is the initial value of the sequence $v(R_0), \dots, v(R_{n-1})$. An *output* of the network N is the sequence $v(R_0), \dots, v(R_{n-1})$ obtained after application of all its layers (*application of* N) on some initial input sequence. We can iterate the network's application, by applying it to the output of its previous application. We call such network a *periodic network*. The *time* complexity of the periodic network is the number of steps performed in all iterations.

3 Definition of the Network $N_{m,k}$

We define a periodic network $N_{m,k}$ for positive integers m and k. For the sake of simplicity we fix the values m and k and denote $N_{m,k}$ by N. Network N contains n registers $R_0, \dots, R_{n-1}$, where $n = 4m \cdot 2^k$. It will be useful to imagine that the registers are arranged in a three-dimensional matrix M of size $2 \times 2m \times 2^k$. For $0 \le x \le 1$, $0 \le y \le 2m - 1$ and $0 \le z \le 2^k - 1$, the element $M_{x,y,z}$ is a register R_i such that $i = x + 2y + 4mz$. For the intuitions, we assume that Z and Y coordinates are increasing downwards and rightwards respectively. By a *column* $C_{x,y}$ we mean a subset of registers $M_{x,y,z}$ with $0 \le z < 2^k$. $P_y = C_{0,y} \cup C_{1,y}$ is a *pair of columns*. An *Z-slice* is a subset of registers with the same Z coordinate.

Let $d = \lceil k/m \rceil$. We define the sets of comparators X, Y_0, Y_1, and Z_i, for $0 \le i < d$, as follows. (Comparators of X, Y_j and Z_i are called *X-comparators*, *Y-comparators* and *Z-comparators*, respectively.) The comparators of X, Y_0 and Y_1 act in each Z-slice separately (see Figure 1). Set X contains comparators $(M_{0,y,z}, M_{1,y,z})$, for all y and z. Let Y be an auxiliary set of all comparators $(M_{x,y,z}, M_{x,y',z})$ such that $y' = (y + 1) \bmod 2m$. Y_0 contains all comparators $(M_{x,y,z}, M_{x,y',z})$ from Y, such that y is even. Y_1 consists of these comparators from Y that are not in Y_0. Note that the layer Y_1 contains *nonstandard* comparators $(M_{x,2m-1,z}, M_{x,0,z})$ (i.e. comparators that place the greater value in the register with lower index).

In order to describe Z_i we define a matrix α of size $d \times 2m$ (with the rows indexed by the first coordinate) such that, for $0 \le i < d$ and $0 \le j < 2m$:

- if j is even then $\alpha_{i,j} = d \cdot j/2 + i$,
- if j is odd $\alpha_{i,j} = \alpha_{i,2m-1-j}$.

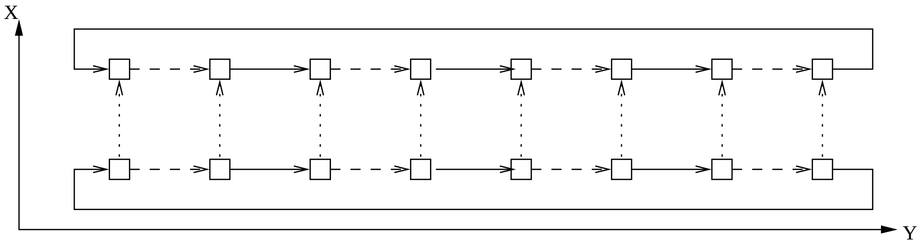

Fig. 1. Comparator connections within a single Z-slice. Dotted (respectively, dashed and solid) arrows represent comparators from X (respectively, Y_0 and Y_1).

For example, for $m = 4$ and $4 < k \leq 8$, α is the following matrix:

$$\begin{bmatrix} 0 & \mathbf{6} & 2 & \mathbf{4} & 4 & \mathbf{2} & 6 & \mathbf{0} \\ 1 & \mathbf{7} & 3 & \mathbf{5} & 5 & \mathbf{3} & 7 & \mathbf{1} \end{bmatrix}.$$

For $0 \leq i < d$, Z_i consists of comparators $(M_{1,y,z}, M_{0,y,z'})$ such that $0 \leq y < 2m$ and $z' = z + 2^{k-1-\alpha_{i,y}}$ provided that $0 \leq z,\ z' < 2^k$ and $k - 1 - \alpha_{i,y} \geq 0$. By a *height* of the comparator $(M_{x,y,z}, M_{x',y',z'})$ we mean $z' - z$. Note that each single Z-comparator is contained within a single pair of columns and all comparators of Z_i contained in the same pair of columns are are of the same height which is a power of two. All Z-comparators of height $2^{k-1}, 2^{k-2}, \ldots, 2^{k-d}$ (which are from $Z_0, Z_1, \ldots, Z_{d-1}$, respectively) are placed in the pairs of columns P_0 and P_{2m-1}. All Z-comparators of height $2^{k-1-d}, \ldots, 2^{k-2d}$ (from $Z_0, \ldots, Z_{d-1}$) are placed in P_2 and P_{2m-3}. And so on. Generally, for $0 \leq i < d$ and $0 \leq y < m$, the height of all comparators of Z_i contained in P_{2y} and in $P_{2m-1-2y}$ is $2^{k-1-dy-i}$.

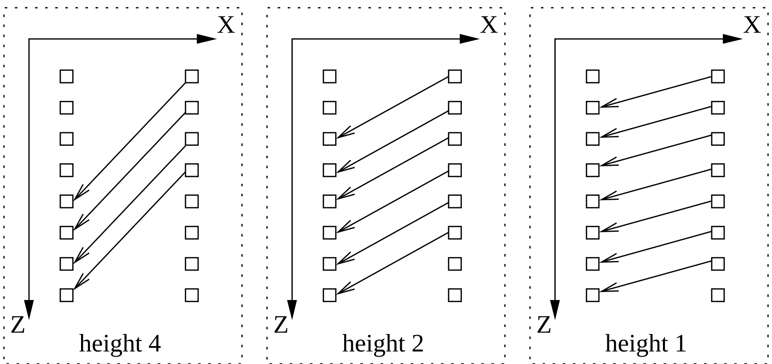

Fig. 2. Z-comparators of different heights within the pairs of columns, for $k = 3$.

The sequence of layers of the network N is $(L_0, \ldots, L_{2d+1})$ where $L_{2i} = X$, $L_{2i+1} = Z_i$, for $0 \leq i < d$, and $L_{2d} = Y_0$, $L_{2d+1} = Y_1$.

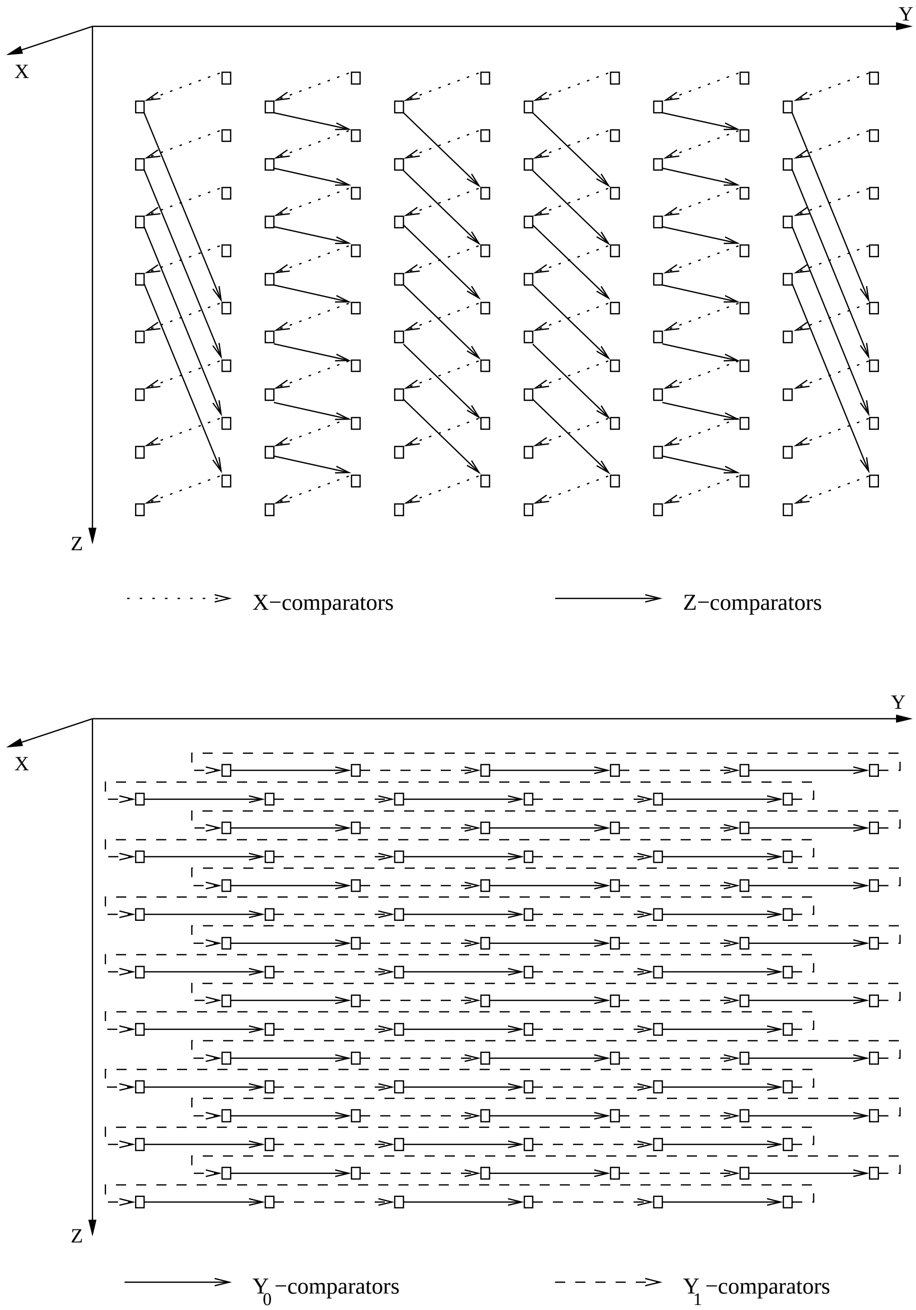

Fig. 3. Network $N_{3,3}$. For clarity, the Y-comparators are drawn separately.

A set of comparators K is *symmetric* if $(R_i, R_j) \in K$ implies $(R_{n-1-j}, R_{n-1-i}) \in K$. Note that all layers of N are symmetric.

Figure 3 shows a network $N_{k,m}$, for $k = m = 3$. As $m \geq k$, this network contains only one layer of Z-comparators Z_0.

4 Analysis of the Computation of $N_{m,k}$

The following theorem is a more detailed version of the theorem stated in the introduction.

Theorem 1. *After $T \leq 4k^2 + 8mk + 7k + 14m + 6\frac{k}{m} + 13$ steps of the periodic network $N_{m,k}$ all its pairs of columns are sorted.*

We denote $N_{m,k}$ by N. By the zero-one principle, [8], it is enough to show this property for the case when only zeroes and ones are stored in the registers. We replace zeroes by negative numbers and ones by positive numbers. These numbers can increase their absolute values between the applications of subsequent layers in periodic computation of N, but can not change their signs. We show that, after T steps, negative values preceed all positive values within each pair of columns.

Initially, let $v(R_0), \ldots v(R_{n-1})$ be arbitrary sequence of the values from $\{-1, 1\}$. We apply N to this sequence as a periodic network. We call the application of the layer Y_i (respectively, X,Z_i) an *Y-step* (respectively, *X-step*, *Z-step*).

To make the analysis more intuitive, we assume that each register stores (besides the value) an unique *element*. The *value of an element* e stored in R_i, (denoted $v(e)$) is equal to $v(R_i)$. If $v(e) > 0$ then e is *positive*. Otherwise e is *negative*. If just before the application of comparator $c = (R_i, R_j)$ we have $v(R_i) > v(R_j)$ then during the application of c the elements are exchanged between R_i and R_j. If c is from Y_0 or Y_1 then the elements are exchanged also if $v(R_i) = v(R_j)$. If e is a positive (respectively, negative) element contained in R_i or R_j, before the application of c, then e *wins* in c if, after the application of c, it ends up in R_j (respectively, R_i). Otherwise e *loses* in c.

We call the elements that are stored during the X-steps and Z-steps in the pairs of columns P_{2i}, for $0 \leq i < m$, *right-running* elements. The remaining elements are called *left-running*.

Let $k' = md$. (Recall that $d = \lceil k/m \rceil$.) Let $\delta = 1/(4k')$. Note that $k'\delta < 1$. By *critical comparators* we mean the comparators between P_{2m-1} and P_0 from the layer Y_1. We modify the computation of N as follows:

- After each Z-step, we increase the values of the positive right-running elements and decrease the values of the negative left-running elements by δ. (We call it *δ-increase.*)
- When a positive right-running (respectively, negative left-running) element e wins in a critical comparator, we increase $v(e)$ to $\lfloor v(e) + 1 \rfloor$ (respectively, decrease $v(e)$ to $\lceil v(e) - 1 \rceil$).

Note that once a positive (respectively, negative) element becomes right-running (respectively, left-running) it remains right-running (respectively, left-running) for ever. All the positive left-running and negative right-running elements have absolute value 1.

Lemma 1. *If, during the Z-step t, $|v(e)| = l + y'\delta$, where l and y' are nonnegative integers such that $l \geq 2$ and $0 \leq y' < k'$, then, during t, e can be processed only by comparators with height $2^{k-1-y'}$.*

Let e be a positive element. (A negative element behaves symmetrically.) Since $v(e) > 1$, e is a right-running element during step t. At the moment when e started being right-running, its value was equal 1. A right-running element can be δ-increased at most k' times between its subsequent wins in the critical comparators, and $k'\delta < 1$. Thus e reached the value 2 when it entered P_0 for the first time. Then its value was being increased by δ, after each Z-step (d times in each P_{2j}), and rounded up to the next integer during its wins in critical comparators. The lemma follows from the definition of α and Z_i: The heights of the Z-comparators from the subsequent Z-layers Z_i, for $0 \leq i < d$, in the subsequent pairs of columns P_{2j}, for $0 \leq j < m$, are the decreasing powers of two. □

We say that a register $M_{x,y,z}$ is *l-dense for v* if

- in the case $v > 0$: $v(M_{x,y,z+i\lceil 2^l \rceil}) \geq v$, for all $i \geq 0$ such that $z + i\lceil 2^l \rceil < 2^k$, and
- in the case $v < 0$: $v(M_{x,y,z-i\lceil 2^l \rceil}) \leq v$ for all $i \geq 0$ such that $z - i\lceil 2^l \rceil \geq 0$.

Note that, for $l < 0$,"l-dense" means "0-dense". An *element is l-dense for v* if it is stored in a register that is l-dense for v.

Lemma 2. *If $M_{x,y,z}$ is l-dense for $v > 0$ (respectively, $v < 0$), then, for $0 < v' \leq v$ (respectively, $v \leq v' < 0$), $M_{x,y,z}$ is l-dense for v'.*

If $M_{x,y,z}$ is l-dense for $v > 0$ (respectively, $v < 0$), then, for all $j \geq 0$ (respectively, $j \leq 0$), $M_{x,y,z+j\lceil 2^l \rceil}$ is l-dense for v.

If $M_{x,y,z}$ is l-dense for $v > 0$ (respectively, $v < 0$) and $M_{x,y,z+\lfloor 2^{l-1} \rfloor}$ (respectively, $M_{x,y,z-\lfloor 2^{l-1} \rfloor}$) is l-dense for v, then $M_{x,y,z}$ is $(l-1)$-dense for v.

The properties can be easily derived from the definition. □

Lemma 3. *Let L be any layer of N and $(M_{x,y,z}, M_{x',y',z'}) \in L$.*

If $M_{x,y,z}$ or $M_{x',y',z'}$ is l-dense for $v > 0$ (respectively, $v < 0$), just before an application of L, then $M_{x',y',z'}$ (respectively, $M_{x,y,z}$) is l-dense for v just after the application of L.

If $M_{x,y,z}$ and $M_{x',y',z'}$ are l-dense for v, just before the application of L, then $M_{x,y,z}$ and $M_{x',y',z'}$ are l-dense for v just after the application of L.

Proof. The lemma follows from the fact that, for each integer i such that $0 \leq z + i\lceil 2^l \rceil$, $z' + i\lceil 2^l \rceil < 2^k$, the comparator $(M_{x,y,z+i\lceil 2^l \rceil}, M_{x',y',z'+i\lceil 2^l \rceil})$ is also in L. □

Corollary 1. *If an element l-dense for v wins during an application of a layer L of N, then it remains l-dense for v. If it looses to another element l-dense for v, then it also remains l-dense for v. If it wins in critical comparator and $v > 0$ (respectively, $v < 0$), then it becomes l-dense for $\lfloor v+1 \rfloor$ (respectively, $\lceil v-1 \rceil$).*

If just before Z-step t, e is right-running positive (respectively, left-running negative) element l-dense for $v > 0$ (respectively, $v < 0$), and, during t, e looses to another element l-dense for v or wins, then it becomes l-dense for $v + \delta$ (respectively, $v - \delta$), after the δ-increase following t.

The following lemma states that each positive element e that was right-running for a long time is contained in a dense foot of the elements with the value $v(e)$ or greater, and an analogical property holds for left-running negative values.

Lemma 4. *Consider the configuration of N after the Z-step. For nonnegative integers l,s and y' such that $y' \leq k'$, for each element e:*

If $v(e) = l + 2 + s + y'\delta$, then e is $(k-l)$-dense for $l + 2 + y'\delta$ and, if $y' > l$, then e is $(k-l-1)$-dense for $l + 2 + y'\delta$.

If $v(e) = -(l + 2 + s + y'\delta)$, then e is $(k-l)$-dense for $-(l + 2 + y'\delta)$ and, if $y' > l$, then e is $(k-l-1)$-dense for $-(l + 2 + y'\delta)$.

Proof. We prove only the first part. The second part is analogical since all layers of N are symmetrical. The proof is on induction by l. Let $0 \leq l < k$. Let e be any element with $v(e) = l+2+s+y'\delta$, for some nonnegative integers s,y', where $y' \leq k'$. The element e was right-running during each of the last y' Z-steps. These steps were preceeded by a critical step t, that increased $v(e)$ to $l + 2 + s$. Let t_i (respectively, t'_i) be the $(i+1)$-st X-step (respectively, Z-step) after step t. Let M_{x_i,y_i,z_i} (respectively, $M_{x'_i,y_i,z'_i}$) be the register that stored e just after t_i (respectively, t'_i). Let v_i denote the value $l + 2 + i\delta$. During each step t_i and t'_i, all elements e' with $v(e') \geq v(e)$, in the pair of columns containing e, are $(k-l)$-dense for v_i. (For $l = 0$ it is obvious, since the "height" of N is 2^k, and, for $l > 0$, it follows from the induction hypothesis and Corollary 1, since e' was $(k-l)$-dense for $l+1$ already before t, and, hence, $(k-l)$-dense for v_0 just after t.)

Claim (Breaking Claim). For $0 \leq i \leq l$, just after the X-step t_i, the registers $M_{0,y_i,z_i+2^{k-i}}$ and $M_{1,y_i,z_i+2^{k-i}}$ are $(k-l)$-dense for v_i, if they exist.

We prove the claim by induction on i. For $i = 0$ it is obvious. (M_{0,y_i,z_i+2^k} and M_{1,y_i,z_i+2^k} do not exist.)

Let $0 < i \leq l$. Consider the configuration just after step t_{i-1}. (See Figure 4.) Since t_{i-1} was an X-step, $v(M_{1,y_{i-1},z_{i-1}}) \geq v(e)$ and, hence, $M_{1,y_{i-1},z_{i-1}}$ is $(k-l)$-dense for v_{i-1}. Thus, $M_{1,y_{i-1},z_{i-1}+2^{k-i}}$ is $(k-l)$-dense for v_{i-1}, since 2^{k-i} is multiple of 2^{k-l}. By the induction hypothesis of the claim, $M_{0,y_{i-1},z_{i-1}+2^{k-i+1}}$ and $M_{1,y_{i-1},z_{i-1}+2^{k-i+1}}$ are $(k-l)$-dense for v_{i-1}. Just after the step t'_{i-1}, $M_{1,y_{i-1},z_{i-1}+2^{k-i}}$, and $M_{1,y_{i-1},z_{i-1}+2^{k-i+1}}$ remain $(k-l)$-dense for v_{i-1}, since they were compared to the registers $M_{0,y_{i-1},z_{i-1}+2^{k-i+1}}$ and $M_{0,y_{i-1},z_{i-1}+2^{k-i+2}}$

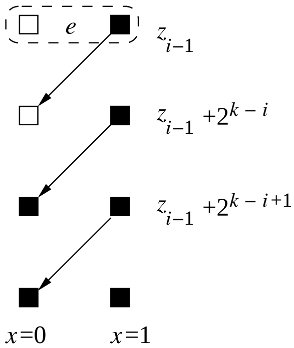

Fig. 4. The configuration after t_{i-1} in $P_{y_{i-1}}$ in the registers with Z-coordinates $z_{i-1}+j2^{k-i}$, for $0 \leq j < 4$. (Black registers are $(k-l)$-dense for v_{i-1}. Arrows denote the comparators from t'_{i-1}.)

that were $(k-l)$-dense for v_{i-1}. $M_{0,y_{i-1},z_{i-1}+2^{k-i+1}}$ remains $(k-l)$-dense for v_{i-1}. $M_{0,y_{i-1},z_{i-1}+2^{k-i}}$ also becomes (or remains) $(k-l)$-dense for v_{i-1}, since it was compared to $M_{1,y_{i-1},z_{i-1}}$. Thus, just after the Z-step t'_{i-1}, for $x \in \{0,1\}$, the registers $M'_x = M_{x,y_{i-1},z'_{i-1}+2^{k-i}}$ are $(k-l)$-dense for v_{i-1} (and for v_i, after the δ-increase). (Either $z'_{i-1} = z_{i-1}$ and $M'_x = M_{x,y_{i-1},z_{i-1}+2^{k-i}}$, or $z'_{i-1} = z_{i-1}+2^{k-i}$ and $M'_x = M_{x,y_{i-1},z_{i-1}+2^{k-i+1}}$.) If $i \bmod d = 0$ then, during the next two Y-steps, the elements from both M'_0 and M'_1 together with the element e are moved "horizontally" to $P_{2i/d}$ (wining by the way). Thus, by Corollary 1, just before and after the X-step t_i, for $x \in \{0,1\}$, the registers $M_{x,y_i,z_i+2^{k-i}}$ are $(k-l)$-dense for v_i. This completes the proof of the claim.

The next claim shows how the values v_l or greater form twice more condensed foot below e.

Claim (Condensing Claim). After the Z-step t'_l, e is $(k-l-1)$-dense for v_l (and for v_{l+1}, after the δ-increase).

Consider the configuration just after X-step t_l. The registers M_{x_l,y_l,z_l} and, by the Breaking Claim, $M_{0,y_l,z_l+2^{k-l}}$ and $M_{1,y_l,z_l+2^{k-l}}$ are $(k-l)$-dense for v_l. Since the last step was an X-step, M_{1,y_l,z_l} is $(k-l)$-dense for v_l.

Consider the following scenarios of the Z-step t'_l (see Figure 5):

1. e remains in M_{0,y_l,z_l}: Then the register $M_{0,y_l,z_l+2^{k-l-1}}$ becomes $(k-l)$-dense for v_l, by Lemma 3, since M_{1,y_l,z_l} was $(k-l)$-dense for v_l just before t'_l. Thus e becomes $(k-l-1)$-dense for v_l, by Lemma 2.
2. e is moved from M_{1,y_l,z_l} to $M_{0,y_l,z_l+2^{k-l-1}}$: Then by Corollary 1, e remains $(k-l)$-dense for v_l, and the register $M_{0,y_l,z_l+2^{k-l}}$ remains $(k-l)$-dense for v_l. Thus e becomes $(k-l-1)$-dense for v_l, by Lemma 2.
3. e remains in M_{1,y_l,z_l}: Then $v(e) \leq v(M_{0,y_l,z_l+2^{k-l-1}}) \leq v(M_{1,y_l,z_l+2^{k-l-1}})$ just before t'_l. (The second inequality is forced by the X-step t_l). Hence, for $x \in \{0,1\}$, $R'_x = M_{x,y_l,z_l+2^{k-l-1}}$ was $(k-l)$-dense for v_l just before t'_l. During

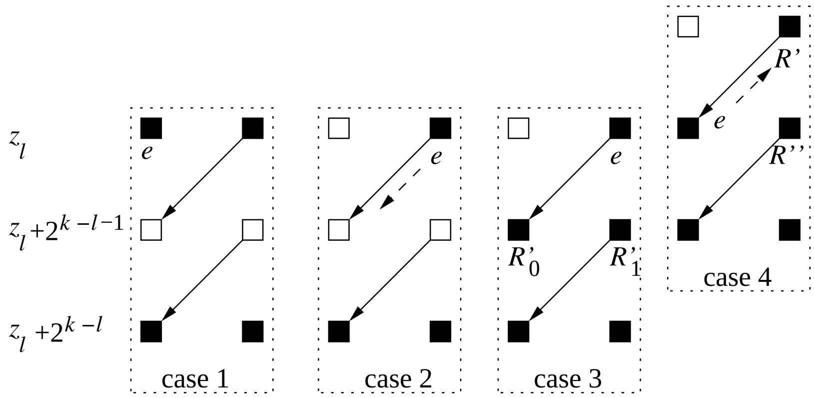

Fig. 5. The scenarios of t'_l.

t'_l the register R'_1 is compared to $M_{0,y_l,z_l+2^{k-l}}$. So R'_1 remains $(k-l)$-dense for v_l. Since e was compared to R'_0, it also remains $(k-l)$-dense for v_l. By Lemma 2, e is $(k-l-1)$-dense for v_l just after t'_l.

4. e is moved from M_{0,y_l,z_l} to $R' = M_{1,y_l,z_l-2^{k-l-1}}$: During t'_l, R' was compared to M_{x_l,y_l,z_l} and $R'' = M_{1,y_l,z_l}$ was compared to $M_{0,y_l,z_l+2^{k-l-1}}$ that was $(k-l)$-dense for v_l just before t'_l, by the Breaking Claim applied to the element in R'. Thus, by Lemma 3, the registers R' and R'' remain $(k-l)$-dense for v_l just after t'_l. By Lemma 2, R' is $(k-l-1)$-dense for v_l just after t'_l.

Since there are no other scenarios for e and the subsequent δ-increase is the same for all positive elements in P_{y_l}, the proof of the claim is completed.

By Corollary 1, the element e remains $(k-l-1)$-dense for v_i, for $i > l$, since other elements in its pair of columns with values $v(e)$ or greater are now also $(k-l-1)$-dense for v_i, and during Y-steps e is wining (right-running).

For $l \geq k$, "$(k-l)$-dense for v" means "0-dense for v". The element e with $v(e) = k+1+k\delta$ is 0-dense for $k+1+k\delta$. All the positive elements below it increase their values at the same rate as e. Thus, when $v(e)$ reaches $k+2$, it becomes 0-dense for $k+2$. By repeating this reasoning for the values $k+2$ and greater we complete the proof of the Lemma 4. □

By Lemma 4, whenever any element e reaches the value $k+2$ (in the pair of columns P_0) it is 0-dense for $k+2$. Then, by the Breaking Claim, after the X-step after e reaches the value $k+2+k\delta$, e is stored in a register $M_{x,y,z}$ such that $M_{0,y,z+1}$ is also 0-dense for $k+2+k\delta$. Hence, all the elements following e in its pair of columns are 0-dense for $k+2+k\delta$. By Corollary 1, this property of e remains valid forever. Since the network is symmetric, we have the following corollary:

Corollary 2. *Consider a configuration in a pair of columns P_y just after an X-step.*

If, for some register $R_i \in P_y$, $v(R_i) \geq k + 2 + k\delta$, then, for all $R_j \in P_y$ such that $j \geq i$, we have $v(R_j) \geq k + 2 + k\delta$.

If, for some register $R_i \in P_y$, $v(R_i) \leq -(k + 2 + k\delta)$, then, for all $R_j \in P_y$ such that $j \leq i$, we have $v(R_j) \leq -(k + 2 + k\delta)$.

Now, it is enough to show that, after the last X-step of the first T steps, all right-running positive and all left-running negative elements have the absolute values $k+2+k\delta$ or greater. Then in each pair of columns containing right-running elements, the -1s are above the positive values, and in each pair of columns containing left-running elements, the 1s are below the negative elements.

Lemma 5. *If, after m Y-steps, and the next $k'(k+1)+k$ Z-steps, and the next X-step, e is a left-running positive (respectively, right-running negative) element, then e remains left-running (respectively, right-running) forever.*

Let e be positive. (The proof for e negative is analogical). During each of the first m Y-steps, e was compared with the positive right-running elements. For $t \geq 0$, let y_t be such that e was in P_{y_t} just after the $(t+1)$st Y-step. For $0 \leq i < m$, let S_i (respectively, S'_i) denote the set of positive elements that were in P_{y_i} (respectively, $P_{(y_i+1) \bmod 2m}$) just after $(i+1)$st Y-step. Let S'' be the set of negative elements in $P_{y_{m-1}}$ just after the mth Y-step. For $0 \leq i < m$, $|S_{m-1}| = 2 \cdot 2^k - |S''| \leq |S'_i|$, since $S_{m-1} \subseteq S_i$ and $|S_i| \leq |S'_i|$. Note that, for all $t \geq m$, during the $(t+1)$st Y-step, the pair of columns containing (left-running) S'' is compared to the pair of columns containing (right-running) $S'_{t \bmod m}$.

After the next $k'(k+1)+k$ Z-steps all the elements of S'' have values $-(k+2+k\delta)$ or less, and, for $0 \leq i < k$, the elements of S'_i have values $k+2+k\delta$ or greater (they have walked at least $k+1$ times through the critical comparators and then increased their values by δ at least k times during Z-steps). Let t' be the next X-step. Let t be any Y-step after t' such that e is still in the same pair of columns as S''. Before the step t, the elements in S'' and each S'_i were processed by an X-step after their absolute values had reached $k+2+k\delta$. Hence, by the Corollary 2, just before the Y-step t, all the final $|S'_i|$ registers of the pair of columns containing S'_i store the values $k+2+k\delta$ or greater and the pair of columns containing S'' has all the initial $|S''|$ registers filled with the values $-(k+2+k\delta)$ or less. Thus, e is stored in one of its remaining $2 \cdot 2^k - |S''|$ final registers and, during the Y-step t, e is compared with a value $k+2+k\delta$ or greater and it must remain left-running. □

The depth of N is $2d+2$. Each iteration of N performs two Y-steps as its last steps. Thus the first m Y-steps are performed during the first $(2d+2)\lceil m/2 \rceil$ steps. Each iteration of N performs d Z-steps. Thus, the next $k'(k+1)+k$ Z-steps are performed during the next $(2d+2)\lceil (k'(k+1)+k)/d \rceil$ steps. After the next X-step, t', by Lemma 5, the set S of positive right-running and negative left-running elements remains fixed. After the next $\lceil (k'(k+1)+k)/d \rceil$ iterations absolute values of elements in S are $k+2+k\delta$ or greater. (t' was the first step of these iterations.) After the first X-step of the next iteration, by Corollary 2, in all pairs of columns the negative values preceed the positive values. We can now replace negative values with zeroes, positive values with ones, and, by the zero-one principle, we have all the pairs of columns sorted. (Note that, by the definition of N, once all the pairs of columns are sorted, they remain sorted for ever.)

We can estimate the number of steps by $T \leq (2d+2)(\lceil m/2 \rceil + 2\lceil (k'(k+1)+k)/d \rceil) + 1$. Recall that $d = \lceil k/m \rceil$. It can be verified that $T \leq 4k^2 + 8mk + 7k + 14m + 6\frac{k}{m} + 13$. This completes the proof of Theorem 1.

Remarks: Note that the network $N_{1,k}$ can be simplified to a periodic sorting network of depth $2 \log n$, by removing the Y-steps and merging P_0 with P_1. However, better networks exist, [3], with depth $\log n$ that sort in $\log n$ iterations. Note also that the arrangement of the registers in the matrix M can be arbitrary. We can select the one that is most suitable for the subsequent merging.

Acknowledgments

I would like to thank Mirosław Kutyłowski for his useful suggestions and comments on this paper.

References

1. M. Ajtai, J. Komlós and E. Szemerédi. Sorting in $c \log n$ parallel steps. Combinatorica, Vol. 3, pages 1–19, 1983.
2. K. E. Batcher. Sorting networks and their applications. Proceedings of 32nd AFIPS, pages 307–314, 1968.
3. M. Dowd, Y. Perl, L. Rudolph, and M. Saks. The periodic balanced sorting network. Journal of the ACM, Vol. 36, pages 738–757, 1989.
4. M. Kik. Periodic correction networks. Proceedings of the Euro-Par 2000, Springer Verlag, LNCS 1900, pages 471-478, 2000.
5. M. Kik, M. Kutyłowski and G. Stachowiak. Periodic constant depth sorting network. Proceedings of the 11th STACS, Springer Verlag, LNCS 775, pages 201–212, 1994.
6. M. Kutyłowski, K. Loryś and B. Oesterdiekhoff. Periodic merging networks. Proceedings of the 7th ISAAC, pages 336–345, 1996.
7. M. Kutyłowski, K. Loryś, B. Oesterdiekhoff, and R. Wanka. Fast and feasible periodic sorting networks. Proceedings of the 55th IEEE-FOCS, 1994.
8. D. E. Knuth. The art of Computer Programming. Volume 3: Sorting and Searching. Addison-Wesley, 1973.
9. De-Lei Lee and K. E. Batcher. A multiway merge sorting network. IEEE Transactions on Parallel and Distributed Systems 6, pages 211-215, 1995.
10. U. Schwiegelshohn. A short-periodic two-dimensional systolic sorting algorithm. IEEE International Conference on Systolic Arrays, pages 257–264, 1988.