Computers and Artificial Intelligence, 4 (1985), No. 6, 535—544

TIGHT CHIP AREA BOUNDS FOR SORTING

Pavol ĎURIŠ
Computing Centre, Slovak Academy of Sciences, Dúbravská cesta, 842 35 Bratislava, Czechoslovakia

Clark D. THOMPSON[1]/
University of California, Computer Science Division, 573 Evans Hall, Berkeley, CA 94720, USA

Ondrej SÝKORA, Imrich VRŤO
Institute of Technical Cybernetics, Slovak Academy of Sciences, Dúbravská cesta, 842 37 Bratislava, Czechoslovakia

Abstract. Tight lower and upper bounds on the area of VLSI sorting circuits are proved. The area required to sort n k-bit numbers is shown to be dependent on the relative sizes of k and n. For example, $A = \Theta(n \log n)$ when $k \geq 2 \log n$, but $A = \Theta(2^k (\log n - k))$ when $k < \log n$.

Оптимальные нижние и верхние границы площади СБИС для сортировки

П. Дюриш, К. Д. Томпсон, О. Сикора, И. Вртё

Резюме. Предлагаются оптимальные нижние и верхние границы площади СБИС кристаллов для сортировки. Показано, что площадь для сортировки n k-разрядных двоичных чисел зависит от относительной величины n и k. Например, $A = \Theta(n \log n)$, если $k \geq 2 \log n$, но $A = \Theta(2^k (\log n - k))$, если $k < \log n$.

1. INTRODUCTION

Complexity theory for VLSI differs from classical complexity theories in its treatment of circuit size, memory space and circuit area. Classical circuit complexity theory counts only the number of gates required to compute a function (circuit size). Classical algorithmic complexity theory counts only the amount of storage required to compute a function (memory space). VLSI complexity theory combines these two measures, gate count and memory space, as well as adding a third component — wire area. In VLSI complexity theory, as in present day engineering practice, the area occupied by a circuit is the sum of three components: gate count, memory space and wire area. A circuit is infeasible if any of these three components takes up too much "silicon real estate". Several recent papers have

[1]/ Supported in part by National Science Foundation grant ECS 84-06408.

developed VLSI complexity-theoretic techniques for proving lower bounds on VLSI circuit area. These papers make two assumptions about circuit input and output (I/O): circuits read their inputs exactly once (semelective I/O) and circuits allways read their inputs and give outputs in the same order (when oblivious I/O). Using these assumptions, any circuit sorting n k-bit numbers can be proved [1] to occupy $\Omega(n)$ area, if $k \geq 2 \log n$. In [2, 5] this result was extended for $k > \log n$. By adding the assumption of bit-serial I/O, THOMPSON [3] shows that $A = \Omega\ (n \log n)$ for $k \geq \varepsilon \log n$, where $\varepsilon > 1$ is an arbitrary constant. THOMPSON expresses the hypothesis that the same result could be proved without the condition of bit-serial I/O.

LEIGHTON [4] proves this hypothesis for $\varepsilon = 2$ and remarks that it could be proved for any $\varepsilon > 1$. Further, LEIGHTON appears to believe that the same result is true for $k = \log n$.

Let us assume $X = (x_1, x_2, \ldots, x_n)$, $Y = (y_1, y_2, \ldots, y_n)$ are input and output sequences of a sorting circuit respectively, where x_i, y_i are k-bit binary numbers, Y is a nondecreasing sequence and n is a power of two.

In this paper we completely solve the problem of proving tight bounds on area for sorting circuits:

$$A = \begin{cases} \Theta(n \log n) & \text{if } k \geq 2 \log n \\ \Theta(n(k - \log n)) & \text{if } 2 \log n > k > \log n \\ \Theta(n) & \text{if } k = \log n \\ \Theta(2^k(\log n - k)) & \text{if } k < \log n \end{cases}$$

We use a simple model of VLSI computation: expressed by the following three assumptions.

1. Semelective input: each input variable is read only once.
2. When-oblivious input and output: the timing of I/O events is data-independent.
3. Unit-area bits: each bit of memory occupies one unit of area.

We prefer the term "when-oblivious" [4] to "when-determinate" [2] for assumption 2, because a non-oblivious, i.e. data-dependent chip is a more realistic concept than a non-deterministic, i.e. oracle-driven chip. For more detailed information about models see [2].

The next section contains our lower bound proofs. Our upper bound proofs appear in Section 3, in the form of circuit constructions.

Finally, we close the paper with conclusions and open problems.

2. LOWER BOUNDS FOR SORTING

For the sake of clarity we represent input and output of the sorting circuit by rectangles divided into n columns and k rows (see Fig. 1), whereby the i-th bit of the j-th number is assigned to the pixel in the i-th row of the j-th column of the rectangles. Let us assume that the rows (columns) are numbered from the bottom to the top (from the left to the right) by numbers $0, 1, \ldots, k-1$, $(1, 2, \ldots, n)$. About Figures 2 and 3 let us assume that the bits in the shaded area are 1's. Otherwise the bits are 0's.

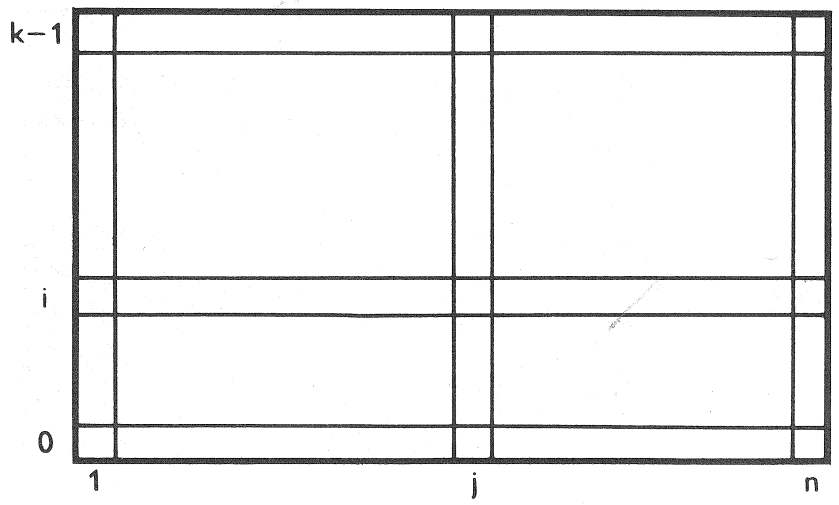

Fig. 1.

We show an auxiliary lemma before proving the main results.

Lemma 1. Any sorting circuit satisfies the following property: for each $r > p$ and for each q, s it holds that the p-th bit of the q-th output number depends on the r-th bit of the s-th input number.

Proof. Let us prove the property by contradiction. Let the r-th bit of the s-th input number be not yet input, whereby $r > p$. Recall now that the order in which the bits are input and output is data-independent because of when-obliviousness.

(i) If the r-th bit of the s-th input number in Fig. 2 (i.e. the pixel with a question mark) is 0 (1), then the p-th bit of the q-th output number must be 0 (1 resp.).
(ii) If $s > q$, let us consider the input as in Fig. 2 with interchanged columns s and q.
(iii) If the r-th bit of the s-th input number is 0 (1), then the p-th bit of the q-th output number must be 1 (0 resp.).
(iv) If $s = q > 1$, let us consider the input as in Fig. 3. If the r-th bit of the s-th input number is 0 (1), then the p-th bit of the q-th output number must be 1 (0 resp.).
(v) If $s = q = 1$, let us consider the input as in Fig. 3 with the r-th bits of all input numbers equal to 0.
(vi) If the r-th bit of the s-th input number is 0 (1), then the p-th bit of the q-th output number must be 0 (1 resp.).

Therefore the p-th bit of the q-th output number cannot be determined earlier than the p-th bit of the s-th input number.

Now we state the main result:

Theorem 1. Any circuit sorting n k-bit numbers requires the following area:

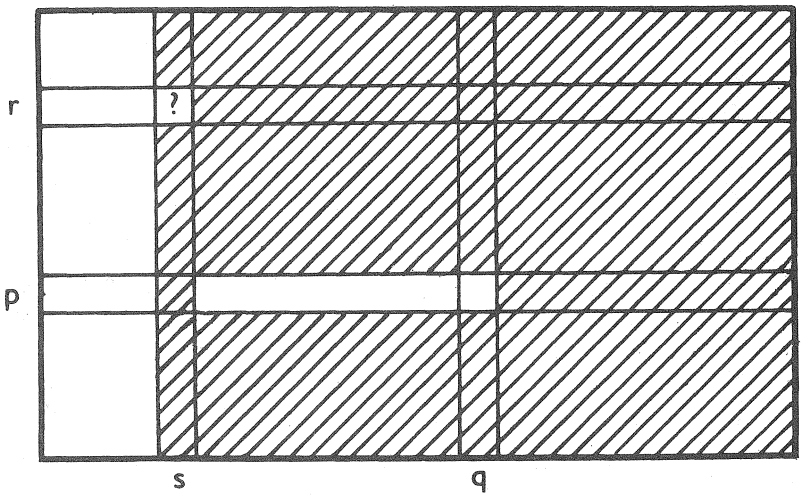

Fig. 2.

$$A = \begin{cases} \Omega(n \log n) & \text{if} \quad k \geq 2 \log n \\ \Omega(n(k - \log n)) & \text{if} \quad 2 \log n > k > \log n \\ \Omega(n) & \text{if} \quad k = \log n \\ \Omega(2^k (\log n - k)) & \text{if} \quad k < \log n \end{cases}$$

Proof. The proof consists of four cases according to the relative sizes of k and n.

Case 1. $\log n < k \leq 2 \log n$

Let $m = 2^{k - \log n}$ and let $(a_1, a_2, \ldots, a_n) = (0, 0, \ldots, 0, 1, 1, \ldots, 1, \ldots, m-1, m-1, \ldots, m-1)$ be such a sequence where each number $j \in \{0, 1, \ldots, m-1\}$ is replicated exactly n/m-times.

Let W be the set of all sequences (of length n) of the following form: $(a_{i_1}, m + a_{i_2}, \ldots, (n-1)m + a_{i_n})$, where $i_1, i_2, \ldots, i_n$ is an arbitrary permutation of $1, 2, \ldots, n$. Let V be the set of all sequences of length n of the form: $(t_1 m + a_1, t_2 m + a_2, \ldots, t_n m + a_n)$, where $t_1, t_2, \ldots, t_n$ is an arbitrary permutation of $2, \ldots, n$. It is evident that for each sequence $w = (w_1, w_2, \ldots, w_n) \in W$ a sequence $v = (v_1, v_2, \ldots, v_n) \in V$ exists such that w is equal to sorted v. Further, if w_i and w'_i are the i-th elements of distinct sequences w and $w' \in W$ resp., and w_i and w'_i are distinct, then they differ in the l-th bit of their binary representation for some l, $0 \leq l \leq k - \log n - 1$ because $w_i = (i-1)m + c_i$ and $w'_i = (i-1)m + c'_i$, where $0 \leq c_i, c'_i \leq 2^{k - \log n} - 1 = m - 1$.

Now we show that any sorting circuit must have at least as many states as the number of distinct sequences in W. According to Lemma 1 there exists a time T when the h-th bits of all input numbers are input for $h = k - \log n, \ldots, k-1$ and none of the l-th bits of output numbers for $l = 0, 1, \ldots, k - \log n - 1$ is output. Let $w, w' \in W$ be distinct and let $v, v' \in V$ be such that w and w' are sorted v and v' respectively. Let s and s' be the states in which the sorting circuit is in the time T if it starts computation with v and v' respectively. As in the

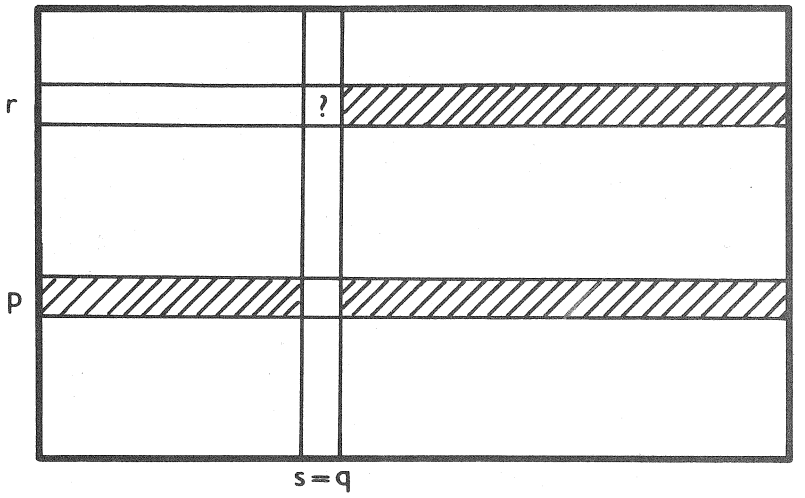

Fig. 3.

time T only some of the 0, 1, ..., $(k-\log n-1)$-th bits of input numbers are not input and as in these bits $v_i = v'_i$ for each i (see the form of sequences from V) one gets that if s and s' are the same states, then the circuit works equally for v and v'. Therefore $w_j = w'_j$ in the 0, 1, ..., $(k-\log n-1)$-th bits for each $j = 1, 2, \ldots, n$ (see the selection of time T), which is the contradiction with the fact that w and w' are distinct in some element in the l-th bit, where $0 \leq l \leq k-\log n-1$. As the circuit area is proportional to the logarithm of the number of states and the number of states is not less than the cardinality of W, one gets

$$A = \Omega(\log(m!)^{n/m}) = \Omega(n(k-\log n)).$$

Case 2. $2 \leq k \leq \log n$

Let U be a set of sequences (of length n) of form (0, 0, ..., 0, 1, 1, ..., 1, ..., 2^k-2, 2^k-2, ..., 2^k-2, 2^k-1, 2^k-1, ..., 2^k-1), where the even numbers occur as many times as the odd numbers (i.e. $n/2-$times) and for each $j = 0, 1, \ldots, 2^{k-1}-1$ it holds that the length of the maximal subsequence of the form ($2j$, $2j$, ..., $2j$, $2j+1$, $2j+1$, ..., $2j+1$) is exactly $n/2^{k-1}$. Let Z be a set of sequences (of length n) such that the first $n/2$ elements are even numbers and other elements are odd numbers and after sorting of each such sequence one gets some sequence from U. If $u, u' \in U$ are distinct, then according to the fact that in each sequence from Y the size of each maximal subsequence of the form ($2j$, $2j$, ..., $2j$, $2j+1$, $2j+1$, ..., $2j+1$) is exactly $n/2^{k-1}$, $j = 0, 1, \ldots, 2^{k-1}-1$, there exists i such that u_i differs from u'_i just in the 0-th bit. Similarly as in the first case one can show that any sorting circuit has to have as many states as there exist mutually distinct sequences from U. According to Lemma 1 there exists time T when all bits except the 0-th bits of all input numbers are input and at the same time the 0-th bit of no output number is determined.

Let us consider the inputs from the set Z. If $z, z' \in Z$, then the 0-th bits of z_i and z'_i are

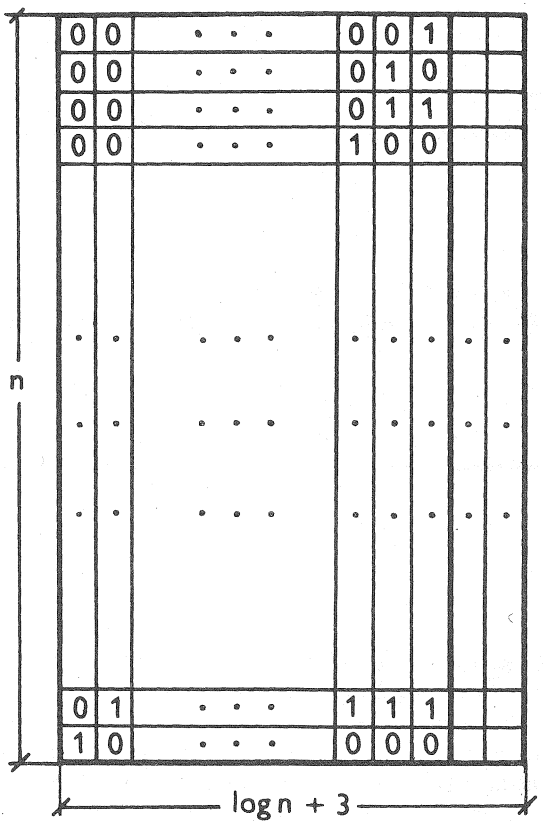

Fig. 4.

equal for each i and if $u, u' \in U$ are distinct, then there exists i such that u_i and u'_i differ in the 0-th bit.

The cardinality of U can be estimated as follows: it is evident that for a sequence $(p_0, p_1, \ldots, p_{2^{k-1}})$, where $0 \leqslant p_i \leqslant n/2^{k-1}$ for each i and $p_i + p_{2^{k-2}+i} = n/2^{k-1}$ for each $i = 0, 1, \ldots, 2^{k-2}-1$, there exists a sequence $u \in U$ for which p_i is the number of occurrences of the number $2i$ in the sequence u (the equality $p_i + p_{2^{k-2}+i} = n/2^{k-1}$ guarantees that the numbers of occurrences of odd and even numbers in u coincide). As there exist at least $(n/2^{k-1}+1)^{2^{k-2}}$ sequences of the form $(p_0, p_1, \ldots, p_{2^{k-1}-1})$ (since the first 2^{k-2} p's can be chosen arbitrarily), the number of distinct sequences in U is at least $(n/2^{k-1}+1)^{2^{k-2}}$. From this it follows, similarly as in case 1, that: $A = \Omega(\log(n/2^{k-1}+1)^{2^{k-2}}) = \Omega(2^k \log(n-k))$.

Case 3. $k > 2\log n$

In this case the proof is the same as for $k = 2\log n$ in case 1, whereby the l-th bits of all input numbers are equal to 0 for $l = 2\log n, 2\log n + 1, \ldots, k-1$.

Case 4. $k = 1$

Similarly as in Lemma 1 one can prove that any output number cannot be determined formerly as all the input numbers are input.

4. AREA-OPTIMAL CIRCUITS FOR SORTING

Constructed sorting circuits are realized as sequential machines. We will show that the required area coincides with lower bounds.

4.1. A radix sorter

This construction requires for sorting of n k-bit numbers $\mathcal{O}(n \log n)$ area. Let us assume there is a table in the sorting circuit (see Fig. 4) which has n rows and $\log n + 3$ columns. In the columns 1, 2, ..., $\log n + 1$ of the table indices of input numbers are stored. At the very beginning of the computation there is index j for $j = 1, 2, \ldots, n$, in the j-th row. In column $\log n + 2$ FLAG-bits are stored which are zeros at the beginning of the computation. Column $\log n + 3$ serves for storing bits of the same rank of input numbers.

Sorting is performed in k-stages. The i-th stage consists of four operations.

1. Input the k-i-th bits of all input numbers into the $(\log n + 3)$-th column in such a way that the bit of the j-th number and index j are in the same row.
2. If there exists a couple of the neighbour rows in the table such that the upper neighbour row contains 1 in the $(\log n + 3)$-th column and the lower neighbour row contains 0 in the same column, and at the same time the lower neighbour has FLAG-bit equal to 0, then these rows except FLAG-bits are changed.
3. For each row such that in the $(\log n + 3)$-th column it contains 1 and in the upper neighbour row in the same column there is 0, the corresponding FLAG-bit is set up to 1.
4. Output of bits from the $(\log n + 3)$-th column.

It is evident that the realization of the table and of the above-described operations requires $\mathcal{O}(n \log n)$ area.

4.2. A small sorting circuit

The second construction, which is designed for sorting of the numbers shorter than $2 \log n$ bits, is more complicated. Let us assume that sorted numbers are kept in the circuit in the form of a string. Every item of the string is composed of two numbers of a variable length DELTA and COUNT. DELTA represents the difference between the number and its predecessor, COUNT indicates the multiplicity of its occurrences. After input of the number into the circuit, the DELTA-values are step by step subtracted from the input number until the end of the string is reached or the result of the subtraction is not positive. If the end of the string is reached a new item with the value of difference as DELTA and COUNT = 1 is added to the end of the string. If the difference is equal to zero, the number COUNT of the corresponding item is incremented by 1. If the difference is equal to a negative number, a new item with the difference value attained before the last subtraction and with COUNT = 1 is inserted before the item whose DELTA-value was for the last time

subtracted. Theorem 2, below, bounds the area required by small sorting circuit.

Before stating the theorem we prove a useful

Lemma 2. Let s, m, q, p_1, p_2, ..., p_s be positive real numbers such that

$$\sum_{1\leq i\leq m} p_i \leq q \quad \text{and} \quad 1\leq s\leq m\leq q.$$

Then

$$\prod_{1\leq i\leq s} p_i \leq (2q/m)^m.$$

Proof. If $s<m$, we write $p_{s+1}=p_{s+2}=\ldots=p_m=1$. Then for all $s\leq m$, we have

$$\sum_{1\leq i\leq m} p_i \leq q+m\leq 2q \quad \text{and} \quad \prod_{1\leq i\leq m} p_i = \prod_{1\leq i\leq s} p_i.$$

Since the geometric mean of a sequence is at most equal to its arithmetic mean (Jensen's inequality),

$$\prod_{1\leq i\leq m} p_i^{1/m} \leq (1/m) \sum_{1\leq i\leq m} p_i \leq 2q/m.$$

Theorem 2. Small sorting circuit requires $\mathcal{O}(\min(n, 2^k)(1+|k-\log n|))$ area to sort n k-bit numbers.

Proof. Sorting circuit represents a sequence $X=(x_1, x_2, \ldots, x_n)$ as a string of DELTA and COUNT-values

$$S(X) = d_1 \# c_1 \# d_2 \# c_2 \# \ldots \# d_r \# c_r.$$

In this string, d_1 is equal to $\min(x_1, \ldots, x_n)$, and c_1 is the number of times the minimum element appears in X. In general, d_i and c_i are defined so that $\sum_{1\leq j\leq i} d_j$ is the value of the j-th smallest element of X, and c_i is the number of times this element appears in X. Trivially, we have $\sum_{1\leq i\leq r} d_i \leq 2^k$ and $\sum_{1\leq i\leq r} c_i \leq n$. Furthermore, all d_i and c_i are positive integers, and $r\leq \min(n, 2^k)$.

Let us assume that any m (DELTA or COUNT-value) from $S(X)$ is binary encoded so that 1 (0) is represented by 11 (01) and terminator # by 00. Then any m with the terminator from $S(X)$ can be represented by $2\lceil \log(m+1)\rceil + 2$ bits. Therefore the length of the entire string $S(X)$ is equal to

$$2\sum_{1\leq i\leq r} (\lceil \log(d_i+1)\rceil + \lceil \log(c_i+1)\rceil + 2) \leq 2\left(\sum_{1\leq i\leq r} \log d_i + \right.$$

$$\left. + \sum_{1\leq i\leq r} \log c_i + 6r\right) \leq 2\left(6r + \log \prod_{1\leq i\leq r} d_i + \log \prod_{1\leq i\leq r} c_i\right).$$

If $2^k\leq n$, by Lemma 2 the length of $S(X)$ is less than or equal to

$$2(6\cdot 2^k + 2^k \log(2^{k+1}/2^k) + 2^k \log(2n/2^k)) = \mathcal{O}(2^k(\log n - k + 1)).$$

Similarly, if $n\leq 2^k$, the length of $S(X)$ is less than or equal to

$$2(6n + n \log(2^{k+1}/n) + n \log(2n/n)) = \mathcal{O}(n(k - \log n + 1)).$$

Combining these bounds, we obtain that the length of $S(X)$ is equal to $\mathcal{O}(\min(n, 2^k)(|k - \log n| + 1))$.

It is evident that a shift register storing the string $S(X)$ requires $\mathcal{O}(\min(n, 2^k)(|k - \log n| + 1))$ area which subsumes the area required for realization of the described operations and control.

5. CONCLUSIONS

We have completely determined the area complexity of when-oblivious VLSI circuits for sorting. Our area bounds are a function of input wordlength k and number of inputs n.

Although our method was developed for VLSI models, it could be applied to tape-bounded Turing machine models or to memory-bounded stored-program random-access machines [6]. The converse is also true: lower bounds on space in classical models can be interpreted as lower bounds on VLSI area. Note, however, that such classically-derived lower bounds for VLSI may be suboptimal because they ignore the area contributions of wires and gates. Nonetheless, for the problems studied in this paper, memory-based lower bounds are seen to be sufficient.

We show our lower bounds on VLSI area are optimal by describing minimal-area VLSI circuits.

We also note that SIEGEL [7] and BILARDI [8] have recently attained the tight chip area bounds presented in this paper (except for the upper bound in case $k \sim \log n$ in [8]).

Although we have completely determined the minimum area requirements of when-oblivious sorting, a couple of interesting open questions remain:

1. Are our upper bounds still optimal if the when-oblivious restriction is dropped? We think so, even though we are unable to remove this assumption from our lower bound proofs.
2. What is the optimal time complexity for minimal-area sorters? For example, available lower bounds on the area·time^2 product of VLSI sorting circuits [4] cover only the case $k = c \log n$, $c > 1$. For other k then we do not know how to simultaneously minimize both area and time.

REFERENCES

[1] CHAZELLE, B.—MONIER, L.: A model of computation for VLSI with related complexity results. In: Proc. of 13th Annual ACM Symposium on Theory of Computing. Milwaukee 1981, pp. 318—325.

[2] ULLMAN, J.: Computational Aspects of VLSI. Computer Science Press, Rockville 1984, pp. 42—43.

[3] THOMPSON, C. D.: Area-time complexity for VLSI. In: Proc. of 3rd Conf. on AI and Inf. Control Systems of Robots, North-Holland Publ. Comp., Bratislava—Smolenice 1984, pp. 373—382.

[4] LEIGHTON, F. T.: Tight bounds on the complexity of parallel sorting. In: Proc. of 16th Annual ACM Symposium on Theory of Computing, Washington 1984, pp. 71—80.

[5] SAVAGE, J. E.: Planar-circuit complexity and the performance of VLSI algorithms. In: VLSI

Systems and Computations, H.T. Kung, B. Sproull, G. Steel (Eds.), Computer Science Press, 1981, pp. 61—68.

[6] AHO, A. V.—HOPCROFT, J. E.—ULLMAN, J. D.: The design and analysis of computer algorithms. Addison-Wesley, Reading 1984, pp. 15—19.

[7] SIEGEL, A. R.: Minimum storage sorting networks. IEEE TC, Vol. C-34, 1985, No. 4, pp. 355—361.

[8] BILARDI, G.: The Area-Time Complexity of Sorting. ACT-52 (PhD. dissertation), Coordinated Science Laboratory, University of Illinois at Urbana, Champaign 1984, 182 pp.

Received February 25, 1985