Artificial Intelligence and Information-Control Systems of Robots
I. Plander (editor)
© Elsevier Science Publishers B.V. (North-Holland), 1984

Area-Time Complexity for VLSI

Clark D. Thompson

Division of Computer Science
573 Evans Hall
U.C.Berkeley, CA 94720

A simplified model of VLSI computation is used to derive lower bounds on the area-time^2 [2] complexity of computation. Circuits for sorting and discrete Fourier transformation are proven to require area A and time T such that $AT^2 = \Omega(N^2 \lg^2 N)$.
Lower bounds on circuit area are also considered. This paper shows how lower bounds on circuit area can be obtained from AT^2 proofs.

1. MODEL OF VLSI COMPUTATION

In our formal model, we orient chips in the Cartesian plane. Labelling the axes l for length and w for width, we say that a chip is the L by W square defined by $0 \leq l \leq L$ and $0 \leq w \leq W$. The axes can be scaled so that at most k wires can be cut by any length-k segment in the plane of the chip, if $k > \mu$. Here μ is the small fixed constant representing the number of available wiring layers.

The area A of a chip is most naturally expressed as the product of its length L and width W. An alternative definition, considered briefly in Section 5, is to count only the area that is "occupied" by gates or wires. Our lower bounds are valid for either of these definitions of circuit area.

We introduce time T into our model as follows. A chip that has been in operation T time units is viewed as having swept out an area-time volume of L.W.T. That is, we add a third axis t to our Cartesian coordinate system in order to represent time [8] . The t-axis is scaled to correspond to the digital bandwidth available in the VLSI technology under consideration. One unit of time is just long enough to send one bit down a wire.

Achievable digital bandwidths currently range from a low of about 10^8 bits/second/wire (in CMOS) to a high of perhaps 10^{11} bits/second/wire (in IBM`s Josephson junction Current Injection Logic [9]). Thus our unit of time for VLSI circuits is 10^{-8} to 10^{-11} seconds.

Considering, again, a length-k cut-segment in the plane of the chip: if this cut-segment remains in place for T time units, its information capacity is at most kT. This information capacity is numerically equal to the area of the surface swept out by the cut-segment in our area-time coordinate system.

Consider, now, a cut-surface perpendicular to the time axis. Such a cut separates a region of the chip at one time from itself, an instant later. The information flowing across this cut is the "state" of that region of the circuit. By the argument below, this information is upper-bounded by the area of the cut-surface. Thus, whether a cut-surface is perpendicular or parallel to the t-axis, the flow of information across that surface is bounded by the area of the surface.

Why do we assert that the storage capacity of a VLSI circuit is bounded by its area? The reason is that wires are the smallest features on a chip. Gates are formed of special types of wire crossings. For example, a metal wire over a polysilicon wire forms part of a MOS gate. There can be no more gates, and thus no more bits of state, than wire crossings.

Stating our information capacity assumption formally, we write

Assumption A-1. At most a bits of information can cross a rectangular cut-surface of area a, if the cut-surface is either parallel or perpendicular to the time axis, and if each dimension of the cut-surface is longer than the fixed constant μ.

(The reason we do not permit cut-surfaces to be skew to the time axis is that we are not sure how to bound information capacity in such cases. Without digging into the matter too deeply, it appears that the capacity of a skewed cut-surface of area a might be as much as $\sqrt{3}a$. Since we have no need of skewed cut-surfaces, we have chosen to define the scaling factors in the model as simply as possible.)

We now discuss circuit input and output, in order to define how information is produced and consumed in our model.

A circuit is said to perform a computation in a when- and where-oblivious [18] fashion if its input and output events occur at predetermined times and places, regardless of the input values being read. In this paper we ignore the more complex (and somewhat rarer) case of non-oblivious circuits.

We further restrict our attention to so-called semelective circuits [23] (coined from Latin: semel=once, lectus=read), in which each input bit is read exactly once. This restriction is almost universally observed in practice.

In a final idealization of the input/output behavior of real circuits, we associate each I/O event with a single point in the area-time coordinates of a circuit. Thus, the m-th bit of the n-th input word might be read at coordinate $(l_{n,m}, w_{n,m}, t_{n,m})$. Information about $x_{n,m}$ may then percolate throughout the chip as the computation proceeds.

Using Assumption A-1, we prove lower bounds on area and time by requiring that the circuit always produce the correct output values. This means that enough information must be present at the area-time coordinate of an output event to determine its value. As with inputs, we assume that output events are points in area-time space.

In actual practice, however, circuit inputs and outputs are not points in area-time. An input value is typically available for at

Artificial Intelligence and Information-Control Systems of Robots
I. Plander (editor)

Area-Time Complexity for VLSI

Clark D. Thompson

Division of Computer Science
573 Evans Hall
U.C.Berkeley, CA 94720

A simplified model of VLSI computation is used to derive lower bounds on the area-time^2 [2] complexity of computation. Circuits for sorting and discrete Fourier transformation are proven to require area A and time T such that $AT^2 = \Omega(N^2 \lg^2 N)$.
Lower bounds on circuit area are also considered. This paper shows how lower bounds on circuit area can be obtained from AT^2 proofs.

1. MODEL OF VLSI COMPUTATION

In our formal model, we orient chips in the Cartesian plane. Labelling the axes l for length and w for width, we say that a chip is the L by W square defined by $0 \leq l \leq L$ and $0 \leq w \leq W$. The axes can be scaled so that at most k wires can be cut by any length-k segment in the plane of the chip, if $k > \mu$. Here μ is the small fixed constant representing the number of available wiring layers.

The area A of a chip is most naturally expressed as the product of its length L and width W. An alternative definition, considered briefly in Section 5, is to count only the area that is "occupied" by gates or wires. Our lower bounds are valid for either of these definitions of circuit area.

We introduce time T into our model as follows. A chip that has been in operation T time units is viewed as having swept out an area-time volume of L.W.T. That is, we add a third axis t to our Cartesian coordinate system in order to represent time [8] . The t-axis is scaled to correspond to the digital bandwidth available in the VLSI technology under consideration. One unit of time is just long enough to send one bit down a wire.

Achievable digital bandwidths currently range from a low of about 10^8 bits/second/wire (in CMOS) to a high of perhaps 10^{11} bits/second/wire (in IBM`s Josephson junction Current Injection Logic [9]). Thus our unit of time for VLSI circuits is 10^{-8} to 10^{-11} seconds.

Considering, again, a length-k cut-segment in the plane of the chip: if this cut-segment remains in place for T time units, its information capacity is at most kT. This information capacity is numerically equal to the area of the surface swept out by the cut-segment in our area-time coordinate system.

Consider, now, a cut-surface perpendicular to the time axis. Such a cut separates a region of the chip at one time from itself, an instant later. The information flowing across this cut is the "state" of that region of the circuit. By the argument below, this information is upper-bounded by the area of the cut-surface. Thus, whether a cut-surface is perpendicular or parallel to the t-axis, the flow of information across that surface is bounded by the area of the surface.

Why do we assert that the storage capacity of a VLSI circuit is bounded by its area? The reason is that wires are the smallest features on a chip. Gates are formed of special types of wire crossings. For example, a metal wire over a polysilicon wire forms part of a MOS gate. There can be no more gates, and thus no more bits of state, than wire crossings.

Stating our information capacity assumption formally, we write

Assumption A-1. At most a bits of information can cross a rectangular cut-surface of area a, if the cut-surface is either parallel or perpendicular to the time axis, and if each dimension of the cut-surface is longer than the fixed constant μ.

(The reason we do not permit cut-surfaces to be skew to the time axis is that we are not sure how to bound information capacity in such cases. Without digging into the matter too deeply, it appears that the capacity of a skewed cut-surface of area a might be as much as $\sqrt{3}a$. Since we have no need of skewed cut-surfaces, we have chosen to define the scaling factors in the model as simply as possible.)

We now discuss circuit input and output, in order to define how information is produced and consumed in our model.

A circuit is said to perform a computation in a when- and where-oblivious [18] fashion if its input and output events occur at predetermined times and places, regardless of the input values being read. In this paper we ignore the more complex (and somewhat rarer) case of non-oblivious circuits.

We further restrict our attention to so-called semelective circuits [23] (coined from Latin: semel=once, lectus=read), in which each input bit is read exactly once. This restriction is almost universally observed in practice.

In a final idealization of the input/output behavior of real circuits, we associate each I/O event with a single point in the area-time coordinates of a circuit. Thus, the m-th bit of the n-th input word might be read at coordinate $(l_{n,m}, w_{n,m}, t_{n,m})$. Information about $x_{n,m}$ may then percolate throughout the chip as the computation proceeds.

Using Assumption A-1, we prove lower bounds on area and time by requiring that the circuit always produce the correct output values. This means that enough information must be present at the area-time coordinate of an output event to determine its value. As with inputs, we assume that output events are points in area-time space.

In actual practice, however, circuit inputs and outputs are not points in area-time. An input value is typically available for at least 10 time units, anywhere on an I/O pad of area 10^3 or 10^4. Similarly, a circuit must hold its output values stable over at least this much area-time volume. This obsevation does not invalidate our model. In fact, Assumption A-1 must hold a fortiori for I/O events because they have such little bandwidth per unit area.

Note that it is quite possible for an I/O event to occur in the plane of an area-time cut-surface, in some application of Assumption A-1. In such cases, we are justified in placing such events on either "side" of the cut-surface. Our reasoning is that, since the information involved in an I/O event must be constant over all points in a non-zero volume of area-time, Assumption A-1 must hold for any and all infinitesimal adjustments in the I/O events`area-time coordinates.

Formally stating our assumption about when- and where-oblivious, semelective, pointcoordinate I/O, we write

Assumption A-2. Each input event $x_{n,m}$ is associated with a fixed point ($l_{n,m}$, $w_{n,m}$, $t_{n,m}$) in the area-time coordinates of a VLSI circuit. Output events are also points in area-time space. If any I/O events lie in the plane of a cut-surface, they may be assigned arbitrarily to either "side" of the plane: the information flow bound of Assumption A-1 must hold for any such assignment.

Note that our model assumptions do not require that information flow in the normal, forward-in-time, fashion. Indeed, an output event can occur before the input event that determines its value. Thus our model is intrinsically nondeterministic. We require only that our circuit`s output "guesses" are proved correct, sooner or later, by information flowing from its inputs.

Our two assumptions are, however, strong enouht to prove tight lower bounds on area-time complexity. The following definitions and lemmas illustrate our proof technique.

Definition. A circuit has "temporal information flow" I if a cut-surface can be found, perpendicular to the t-axis, across which I bits must flow.

Lemma 1-1. The area A of any circuit with temporal information flow I must obey $A > I$.

Proof. The area A of the chip must be at least as large as the area of the cut-surface. By Assumption A-1, the cut-surface area is bounded from below by I. □

Definition. A circuit has "cross-chip information flow" I if two cut-surfaces can be found, one perpendicular to the l-axis and one perpendicular to the w-axis, across which I bits must flow.

Lemma 1-2. $AT^2 \geq I^2$ for any circuit with cross-chip information flow I.

Proof. Consider the cut-surface perpendicular to the l-axis. It has area at most WT and thus, by Assumption A-1, $WT \geq I$. Similarly, $LT \geq I$. Multiplying the two inequalities together, we obtain the Lemma. □

2. NOTATION

The following letters have special meaning:

A - circuit area, see § 1; B - the matrix defining a discrete Fourier transform, see § 4; L - circuit length, see § 1; μ - number of layers of wiring, see § 1; M - the number of bits in an input word, see § 3; N - the number of words of input, see § 3; q - the modulus of the arithmetic operations for the DFT, see § 4; T - circuit time, see § 1; W circuit width, see § 1; X - the vector of random input bits, see § 3 and § 4; $\check{X}$ - a particular value for X; $x_{n,m}$ - the m-th least-significant bit of the n-th input word, see § 3 and §4; Y - the vector of output bits, see § 3 and § 4; $\check{Y}$ - a paricular value for Y; and $y_{n,m}$ - the m-th least-significant bit of the n-th output word, see § 3 and § 4.

In addition, we use the following standard functional notation.

"probability distribution" for a discrete random variable X, Pr $[X = \check{X}]$, constrained by

$$(\forall\check{X}\ 0 \leq \Pr\ [X = \check{X}] \leq 1) \text{ and } (\sum_{\check{X}} \Pr\ [X = \check{X}] = 1)$$

"entropy of X":

$$H(X) = \sum_{\check{X}} - \Pr[X = \check{X}]\ \lg \Pr\ [X = \check{X}]$$

"worst-case entropy of X":

$$H_w(X) = \lg \left|\{\check{X}\colon \Pr[X = \check{X}] > 0\}\right|$$

The entropy function H(X) can be interpreted as the average length of a description of an event in X. The worst-case entropy function $H_w(X)$ is, by contrast, the length of the longest descriptor for an event in X. The latter definition is more useful for our purposes, since we study circuits with when-oblivious I/O. Hence we seek worst-case bounds on information and time.

3. LOWER BOUNDS FOR SORTING

The problem of sorting is to arrange a sequence of input values into increasing order. We assume that the inputs are expressed as M-bit integers, that 2^M grows proportionately with the number N of inputs, and that each input bit position actually represents one bit of (worst-case) information. More formally, we assume:

Definition S-1: The input to a sorting circuit is a vector X of NM boolean variables $x_{n,m} \in \{0,1\}$ for $1 \leq n \leq N$, $1 \leq m \leq M$. Each value of X is possible: $\Pr(X = \check{X}) > 0$.

Definition S-2. $M \geq (1+\epsilon)\lg N$, for some fixed $\epsilon > 0$.

Definition S-3. The inputs X are interpreted as N words of M bits, encoded in straight binary. We refer to the n-th word as $x_{n,*}$; its value is given by

$$x_{n,*} = \sum_{m} x_{n,m} 2^{M-m}$$

Let π be any permutation of $\{1,2, \ldots, N\}$ that brings the $x_{n,*}$ into nondecreasing order: $x_{\pi(n),*} \leq x_{\pi(n+1),*}$, for $1 \leq n < N$. Then the value of each bit in the output vector Y is defined by $y_{n,m} = x_{\pi(n),m}$.

Definitions S-1 and S-2 can be used to express the entropy of the input vector as a function of N alone:

$$H(X) \geq (1 + \epsilon)(N \lg N)$$

We do not know whether definition S-2 is strictly necessary to prove strong lower bounds on the sorting problem. However, we have found counterexamples to our lower bound theorems if $M < .5 \lg N$. It seems that, if M is too small, the sorting problem reduces to a much easier "counting problem." This issue is discussed again at the end of this section.

We need to make one more definition before proving our first AT^2 theorem:

Definition. A chip has "bit-serial I/O" if all M bits of each input word $x_{n,*}$ enter the chip at the same place (l_n, w_n). Similarly, one place is associated with each output word $y_{n,m}$.

Note that the time $t_{n,m}$ at a bit-serial circuit receives the m-th bit of input $x_{n,*}$ is, in general, distinct from the time at which it gets the $(m + 1)$st bit of this word. For lack of space the following theorem, and the subsequent ones, are presented without proofs.

Theorem 3-1. $AT^2 = \Omega(N^2 \lg^2 N)$ for any sorting chip with bit-serial I/O.

For sorting chips not obeying the bit-serial I/O restriction, we offer the following theorem due to Vuillemin [31] .

Theorem 3-2. $AT^2 = \Omega(N^2)$ for any chip sorting N numbers.

To prove the above theorem we have to take advantage of the fact that our cut-surface was perpendicular to the time axis. Thus we demonstrate there is a temporal information flow of N/2 bits, as needed to apply Lemma 1-1:

Theorem 3-3. $A = \Omega(N)$ for any chip sorting N numbers.

Unfortunately, Lemma 1-1 does not apply to the $\Omega(N \lg N)$ information flow observed in the proof of Theorem 3-1. For that flow, it was important that all the bits of each input word lie on one side or the other of a cut-surface. The bit-serial I/O assumption was designed to assure that this is the case.

This suggests an interesting question: Is the bit-serial I/O assumption necessary to prove an $\Omega(N \lg N)$ information flow? If not then $AT^2 = \Omega(N \lg N)$ for unrestricted inputs, as well as $A = \Omega(N$

lg N). A sketch of these results was recently shown the author by Tom Leighton of M.I.T. [16] . The only defect of Leighton`s proof technique is that it probably requires word lengths M of at least 4 lg N.

Thus an open question remains: What is the minimum word length for which the AT^2 complexity of sorting is Ω(N lg N)? The interesting range of word length is between .5 lg N and lg N, since sorting is certainly "easy" when $M < .5$ lg N:

Theorem 3-4. A circuit can be built to sort N m-bit words, using area A = O(NM) and time T = $O(2^M \lg N)$.

4. DISCRETE FOURIER TRANSFORMATION

The discrete Fourier transform (or DFT) on N elements, computed over a finite ring of modulus q, can be defined in the following fashion. Note the similarities to our definition of the sorting problem.

Definition D-1. The input to a DFT circuit is a vector X of NM boolean variables $x_{n,m} \in \{0,1\}$ for $1 \leq n \leq N$, $1 \leq m \leq M$. Each value of X is possible: $\Pr\{X = \check{X}\} > 0$.

Definition D-2. $M = \lceil \lg q \rceil$. Also, the prime factorization $p_1^{r_1}\ p_2^{r_2}$ $\ldots, p_k^{r_k}$ of the ring modulus q must be such that N is evenly divisible by the least common divisor of $p_1 - 1, p_2 - 1, \ldots, p_k - 1$.

Definition D-3. Elements of the ring are coded as binary integers from the set $\{0, 1, \ldots, q-1\}$. Ring multiplication and addition are then performed as integer multiplication and addition modulo q. Writing $x_{n,*}$ for the value of the n-th input word,

$$x_{n,*} = \sum_m x_{n,m}\ 2^{M-m}$$

and, similarly, writing $y_{n,*}$ for the n-th output word, the DFT is a simple matrix-vector multiplication Y = BX. The ij-th element of the matrix B is defined by

$$B\left[i,j\right] = \sqrt[N]{1}^{ij}$$

The restriction on q`s factorization in Definition D-2 is necessary and sufficient to ensure that $\sqrt[N]{1}$ and N^{-1} exist in the ring. The matrix B is thus well-defined, and our DFT has the usual properties of invertibility, orthogonality, and cyclic convolvability [2] .

Definition D-2 also implies that $q > N$, and thus that $M \geq \lceil \lg N \rceil$. As in the sorting problem, the entropy of the input vector is $H_W(X) = \Omega$ (N lg N).

We approach our AT^2 theorem by means of the following lemma.

Lemma 4-1. [30], [28] . Let B_R be the square submatrix formed by selecting the first $\lfloor N/2 \rfloor$ rows and any $\lfloor N/2 \rfloor$ columns of the matrix

B of Definition D-3. If the ring modulus q is prime, then B_R is invertible.

Theorem 4-2. [24] . $AT^2 = \Omega(N^2 lg^2 N)$ for the bit-serial computation of the DFT.

We do not know of any lower bound on information flow in a DFT computation that does not assume the inputs are presented in bit-serial fashion. However, Theorem 4-2 is strong enough to prove the following result:

Corollary 4-3. [25] . The area of a shuffle-exchange graph is bounded from below by $A = \Omega(N^2/lg^2 N)$.

5. CONCLUSION

In this paper, we have shown that the finite width and bandwidth of wires in VLSI circuits imply lower bounds on area and on the area-time^2 product. Using the bit-serial I/O assumption, we were able to prove that $AT^2 = \Omega(N^2 \log^2 N)$ for both sorting and Fourier transformation. These bounds are tight, since circuits can be constructed to match this performance [26 , 27] . Some of these circuits occupy only O(N lg N) area, showing that a lower bound of $A = \Omega(N lg N)$ would be the best-possible result for these problems. Unfortunately, no such lower bounds are available in the published literature.

We are also unable to prove that $AT^2 = \Omega(N^2 lg^2 N)$ for circuits which don`t employ bit-serial I/O. Tom Leighton appears to have proved this theorem for the sorting problem [16] . Using the techniques of this paper, a simple corollary of Leighton`s proof is that all sorting circuits have $A = \Omega(N lg N)$. We conjecture that analogous bounds could be obtained for Fourier transformation.

Another open problem, mentioned in Section 4, is to understand the effect of word length on the area-time complexity of sorting. Word lengths between (lg N)/2 and (lg N) bits are the most interesting. Circuits based on an enumeration strategy [20] work well on shorter words, and we have reasonably-tight bounds for longer words.

Research could of course be done to determine the area-time complexity of problems other than sorting and Fourier transformation. Ullman`s recent book [29] is a good summary of the current state-of-the-art in VLSI complexity theory. In brief, tight or nearly-tight bounds have been proven for integer arithmetic, [1, 5, 32, 12] matrix multiplication, [31, 23, 4] boolean predicates and language recognition problems, [6, 18, 22] as well as for transitive closure and other graph-theoretic calculations [10, 11] .

Another research direction is to determine the effect of varying model assumptions. For example, one might assume that information is transmitted at a maximum velocity, [7] that the circuit is allowed to have some forms of nondeterministic or random behavior, [19, 3] that circuits are three-dimensional, [21] or that circuit I/O is not semelective [13]. Also, one might consider different parameters of circuit performance, such as "occupied area" [25] or energy consumption [17, 14] .

We can immediately extend the results of this paper in a couple of the directions mentioned above. Bounds on volume-time complexity can be obtained in a fashion analogous to the AT^2proofs. We introduce a fourth spatial coordinate h, for height, to our area-time system. (This model will only be appropriate when the height H of a chip is allowed to grow with the size of the problem.) Assuming that wires have a finite cross-sectional area, the information capacity of a space-time cut is proportional to the volume of the cut. Now we observe that the cross-chip information flow of our proofs will occur across three surfaces, one perpendicular to each **of** the three spatial axes. We then write three inequalities: $HLT \geq I$, $HWT \geq I$, $LWT \geq I$. Multiplying the three together we find that $V^2T^3 \geq I^3$, where V is the volume of the chip.

A second extension is to a different notion of circuit area. The "bounding rectangle" definition of area used in this paper is something of an overstimate of the actual amount of wires and gates required by a circuit. A circuit with a nonconvex perimeter could have an arbitrarily small ratio between its "occupied area" A_O and its "bounding rectangle" A. This objection to our definition of A is, however, not of fundamental importance. Any circuit with holes or a nonconvex perimeter can be warped into a rectangular shape, stretching its wires only by an additive constant factor proportional to the size of the largest unwarpable "feature" of the chip. Thus, up to constant factors, nonconvex and hole-y circuits have no speed or area advantage. Rectangular (or even square) circuits are sufficiently general to cover the VLSI design space.

REFERENCES

[1] H. Abelson and P. Andreae, "Information Transfer and Area-Time Tradeoffs for VLSI Multiplication," Comm. ACM, vol. 23, no.1, pp.20-23, January 1980.

[2] Ramesh C. Agarwal and Sidney C. Burrus, "Number Theoretic Transforms to Implement Fast Digital Convolutions," Proc. IEEE, vol. 63, no. 4, pp. 550-560, April 1975.

[3] A. V. Aho, J. D. Ullman, and M. Yannakakis, "On Notions of Information Transfer in VLSI Circuits," in Proc. 15th Annual ACM Symp. on Theory of Computing, pp. 133-139, April 1983.

[4] Dana Angluin, "VLSI: On the Merits of Batching," unpublished manuscript, Yale University, April 1982.

[5] R. Brent and H. T. Kung, "The Area-Time Complexity of Binary Multiplication," JACM, vol 28, no.3, pp. 521-534, July 1981.

[6] R. P. Brent and L. Goldshlager, "Some Area-Time Tradeoffs for VLSI," SIAM J. Comput., vol. 11, no. 4, pp. 737-747, November 1982.

[7] B. Chazelle and L. Monier, "Towards More Realistic Models of Computation for VLSI", in Proc. 11th Annual ACM Symp. on Theory of Computing, pp. 209-213, April 1979.

[8] Abbas El Gamal and King F. Pang, "VLSI Complexity of Functions with Certain Local Properties," extended abstract, November 1983.

[9] T.R.Gheewala, "Design of 2.5-Micrometer Josephson Current Injection Logic (CIL)," IBM J. Res. Develop., vol. 24, no. 2, pp. 130-142, March 1980.

[10] Susanne E. Hambrusch and Janos Simon, "Solving Undirected Graph Problems on VLSI," CS-81-23, Computer Science Dept., Pennsylvania State Univ., Univ. Park, PA 16802, December 1981.

[11] J. Ja`Ja`, "The VLSI Complexity of Graph Problems," CS-81-25, Computer Science Dept., Pennsylvania State Univ., Univ. Park, PA 16802, October 1981.

[12] J. Ja`Ja`and V.K.Prasanna Kumar, "Information Transfer in Distributed Computing with Applications to VLSI," CS-82-17, Computer Science Dept., Pennsylvania State Univ., Univ. Park, PA 16802, August 1982.

[13] Zvi M. Kedem and Alessandro Zorat, "Replication of Inputs May Save Computational Resources in VLSI," in Proc. 22nd Symp. on the Foundations of Computer Science, IEEE Computer Society, October 1981.

[14] Gloria Kissin, "Measuring Energy Consumption in VLSI Circuits: a Foundation," in Proc. 14th Annual ACM Symp. on Theory of Computing, pp. 99-104, May 1982.

[15] F. T. Leighton, "Layouts for the Shuffle-Exchange Graph and Lower Bound Techniques for VLSI," Ph. D. Dissertation, MIT/LCS/TR-724, M.I.T. Lab for Computer Science, June 1982.

[16] F. T. Leighton, private communication, November 1983.

[17] Thomas Lengauer and Kurt Mehlhorn, "On the Complexity of VLSI Computations", in VLSI Systems and Computations, ed. H.T.Kung, Bob Sproull, Guy Steele, pp. 89-99, Computer Science Press, October 1981.

[18] Richard J. Lipton and Robert Sedgewick, "Lower Bounds for VLSI," in Proc. 13th Annual ACM Symp. on Theory of Computing, pp. 300 -307, May 1981.

[19] Kurt Mehlhorn and Erik M. Schmidt, "Las Vegas is better than Determinism in VLSI," in Proc. 14th Annual ACM Symp. on Theory of Computing, pp. 330-337, May 1982.

[20] D.E. Muller and F.P.Preparata, "Bounds to Complexities of Networks for Sorting and for Switching," JACM, vol.22, no. 2, pp. 195-201, April 1975.

[21] Arnold L.Rosenberg, "Three-Dimensional VLSI, I: a case study," in VLSI Systems and Computations, ed. H.T.Kung, Bob Sproull, Guy Steele, pp. 69-79, October 1981.

[22] J. Savage, "Planar Circuit Complexity and the Performance of VLSI Algorithms," in VLSI Systems and Computations, ed. H.T. Kung, Bob Sproull, Guy Steele, pp. 61-68, Computer Science Press, October 1981.

[23] J. Savage, "Area-Time Tradeoffs for Matrix Multiplication and Related Problems in VLSI Models," Journal of Comput. and Syst. Sci., vol. 22, no.2, pp. 230-242, April 1981.

[24] James B. Saxe, private communication, October 1978.

[25] C.D.Thompson, "A Complexity Theory for VLSI," Ph.D. Dissertation, CMU-CS-80-140, Computer Science Dept., Carnegie-Mellon University, August 1980.

[26] C.D.Thompson, "Fourier Transforms in VLSI," IEEE Trans. Comput., October 1983.

[27] C.D.Thompson, "The VLSI Complexity of Sorting," IEEE Trans. Comput., December 1983.

[28] Martin Tompa, "Time-Space Tradeoffs for Computing Functions, Using Connectivity Properties of Their Circuits," in Proc. 10th Annual ACM Symp. on Theory of Computing, pp. 196-204, May 1978.

[29] J. Ullman, Computational Aspects of VLSI, Computer Science Press, 1984.

[30] Leslie G. Valiant, "Graph-Theoretic Properties in Computational Complexity", Journal of Computation and Systems Sciences, vol. 13, pp. 278-285, 1976.

[31] J. Vuillemin, "A Combinational Limit to the Computing Power of VLSI Circuits," IEEE Trans. Comput., vol. C-32, no. 3, pp. 294-300, March 1983.

[32] A.C. Yao, "The Entropic Limits of VLSI Computations," in Proc. 13th Annual ACM Symp. on Theory of Computing, pp. 308-311, May 1981.