February 1986

UILU-ENG-86-2204
ACT-67

COORDINATED SCIENCE LABORATORY
College of Engineering
Applied Computation Theory

ON PROBLEM TRANSFORMABILITY IN VLSI

Scot Hornick
Majid Sarrafzadeh

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release, distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-86-2204 (ACT-67)	5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Laboratory, Univ. of Illinois 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Semiconductor Research Corporation
6c. ADDRESS (City, State and ZIP Code) 1101 W. Springfield Avenue Urbana, Illinois 61801	7b. ADDRESS (City, State and ZIP Code) P.O. Box 12053 Research Triangle Park, NC 27709
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Semiconductor Research Corporation 8b. OFFICE SYMBOL (If applicable) N/A	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER SRC-RSCH-84-06-049-6
8c. ADDRESS (City, State and ZIP Code) P.O. Box 12053 Research Triangle Park, NC 27709	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: N/A PROJECT NO.: N/A TASK NO.: N/A WORK UNIT NO.: N/A
11. TITLE (Include Security Classification) On Problem Transformability in VLSI	
12. PERSONAL AUTHOR(S) Hornick, Scot and Sarrafzadeh, Majid	
13a. TYPE OF REPORT Interim Technical, final 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) February 1986 15. PAGE COUNT 21
16. SUPPLEMENTARY NOTATION N/A	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	VLSI model of computation, area-time tradeoff, lower bound, problem transformation, computational prototype

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The two basic performance parameters that capture the complexity of any VLSI chip are the area of the chip, A, and the computation time, T. A systematic approach for establishing lower bounds on A is presented. This approach relates A to the bisection flow, ϕ. A theory of problem transformation based on ϕ, which captures both AT^2 and A complexity, is developed. A fundamental problem, namely, element uniqueness, is chosen as a computational prototype. It is shown under general I/O protocol assumptions that any chip that decides if a list of n elements (each with $(1+\varepsilon)\log n$ bits) are unique must have $\phi = \Omega(n\log n)$, and thus, $AT^2 = \Omega(n^2\log^2 n)$, and $A = \Omega(n\log n)$. A theory of VLSI transformability reveals the inherent AT^2 and A complexity of a large class of related problems.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) / 22c. OFFICE SYMBOL

On Problem Transformability in VLSI*

Scot Hornick and Majid Sarrafzadeh

Coordinated Science Laboratory and

Department of Electrical and Computer Engineering

University of Illinois

Urbana, IL 61801

Abstract: The two basic performance parameters that capture the complexity of any VLSI chip are the area of the chip, A, and the computation time, T. A systematic approach for establishing lower bounds on A is presented. This approach relates A to the bisection flow, ϕ. A theory of problem transformation based on ϕ, which captures both AT^2 and A complexity, is developed. A fundamental problem, namely, element uniqueness, is chosen as a computational prototype. It is shown under general I/O protocol assumptions that any chip that decides if a list of n elements (each with $(1+\epsilon)\log n$ bits) are unique must have $\phi = \Omega(n \log n)$, and thus, $AT^2 = \Omega(n^2 \log^2 n)$, and $A = \Omega(n \log n)$. A theory of VLSI transformability reveals the inherent AT^2 and A complexity of a large class of related problems.

Key words: VLSI model of computation, area-time tradeoff, lower bound, problem transformation, computational prototype.

---

*This work was supported in part by the Semiconductor Research Corporation under contract RSCH 84-06-049-6.

1. Introduction

In the study of complexity theory, a fundamental problem -- normally referred to as a *computational prototype* -- is chosen as the representative of a class of related problems. Establishing a lower bound on some significant performance parameter of a computational prototype has always been a difficult task, but once it is accomplished, the same bound for the rest of the problems in the class is established by means of *problem transformation*. Employment of a computational prototype is now classical; the most well-known examples are satisfiability in the theory of NP-completeness [GJ] and element uniqueness in the RAM model [PS].

Recent improvements in fabrication technology have made VLSI an attractive computation environment. The new challenge is the exploitation of the properties of VLSI to build efficient and effective computational structures. In the VLSI model of computation as formulated by [T,BK,AA], the fundamental complexity measures are A, the area of the VLSI chip, and T, its computation time. VLSI computation theory addresses the problem of using these two resources in an optimal (or efficient) manner. In order to establish criteria of optimality, research is often directed at proving lower bounds on area, time, or various functions that capture an area-time tradeoff, e.g., AT^2. Standard techniques exist for proving lower bounds on T and AT^2; they are based on bounded fan-in arguments (in the case of T) and on information flow arguments (in the case of AT^2) [T,BP]. In this paper, we will present a standard technique for proving lower bounds on A. This technique is very similar to Thompson's bisection flow technique. Indeed, we will show that a lower bound on the bisection flow for a particular computation immediately implies a lower bound on the area of any chip that performs the computation (subject to appropriate input/output protocol constraints).

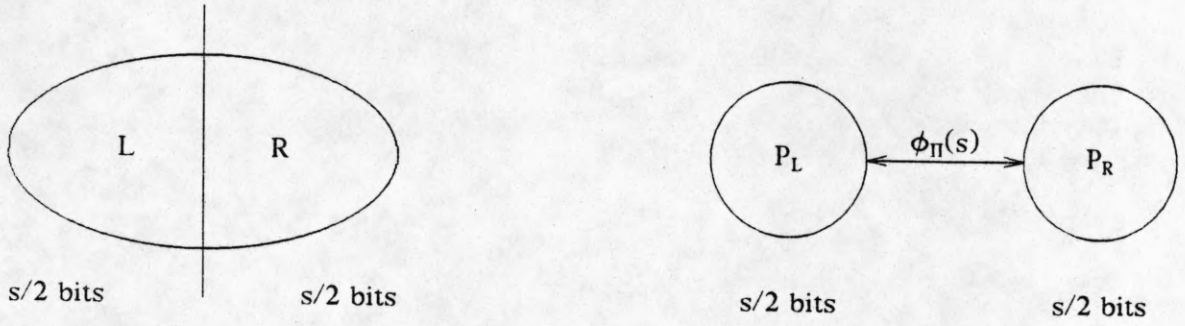
To establish a lower bound on the bisection flow for a problem Π, there are two ways to proceed. The traditional approach is to essentially start from scratch, without taking advantage of previously derived lower bounds. A different approach is to utilize facts already known about another problem and show, by means of problem transformation, that Π is at least as hard as this

problem. Until now, the first technique has been used almost exclusively; the second approach has been used only in trivial situations, for example, to observe that inverting an arbitrary matrix is at least as hard as inverting a triangular matrix. Our goal is to establish a framework in which the second technique, that is, problem transformation, can be efficiently employed. This framework can be used to establish nontrivial lower bounds for a large class of related problems.

This paper is organized as follows. In section 2, we modify the bisection flow technique of Thompson to lower bound A instead of $\sqrt{A}T$. We investigate the duality of area and time in these lower bounds and show how, under this duality, a $\sqrt{A}T$ (i.e., AT^2) lower bound, obtained by bisection flow arguments, implies an A lower bound. In Section 3, we develop a theory of problem transformation in VLSI that is based on the bisection flow. A computational prototype, namely, element uniqueness, is introduced and nontrivial lower bounds on the bisection flow for this problem are established. Finally, in Section 4, these results are integrated to establish nontrivial AT^2 and A lower bounds for a large class of problems.

2. Lower Bounds Using Bisection Flow

Thompson, in his seminal thesis [T], proposed a now classical technique for analyzing VLSI complexity, as follows. Consider a problem $\Pi(s)$, where s is the input size, and a chip C_Π with area A that is capable of solving Π in time T. Let l be a cut that partitions C_Π into a left side (L) and a right side (R), such that each side reads (almost) half of the inputs, i.e., $s/2 - o(s)$ bits, as shown in Figure 1a. The general framework is one in which two processors, P_L and P_R, associated respectively with L and R cooperate to solve $\Pi(s)$ (see Figure 1b). We denote by $\phi_\Pi(s)$ the number of bits that P_L and P_R communicate to solve $\Pi(s)$. As Ullman noted [U], the history of the computation performed by C_Π can be modeled with an area-time solid, as shown in Figure 2. The communication channel between P_L and P_R is represented by the rectangle F (indicated by the dashed line) that transects the longer of the two area dimensions. Thus, F has sides of length T and (at most) $\sqrt{A}$; so A_F, the area of F, is at most $\sqrt{A}T$. If $\phi_\Pi(s)$ bits must flow across this channel, then $A_F = \Omega(\phi_\Pi(s))$. Hence, we obtain:



a. Bisection of a chip

b. A spatial two-processor system

Figure 1

$$\sqrt{A}T = \Omega(\phi_\Pi(s)), \tag{1}$$

or, equivalently,

$$AT^2 = \Omega(\phi_\Pi^2(s)). \tag{2}$$

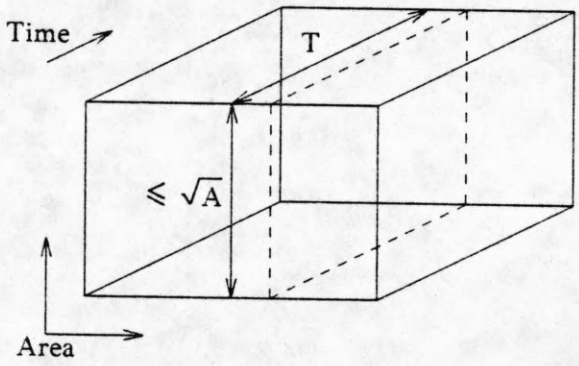


Figure 2. Area-time solid with spatial bisection of inputs

Lower bounds on chip area have been obtained for a number of specific problems, e.g., [BK,BP,L,S,DSVT]. However, unlike AT^2 lower bounds, for which Thompson's thesis gives us a standard proof technique, A lower bounds are usually proven with involved ad hoc arguments. Until now, general results on area were known only for 0/1 output functions [Y2] and for transitive functions [V]. Here, we generalize both of these results and present a new methodology for proving area lower bounds.

Again, we consider the area-time solid that models the computational history of C_Π. Suppose there is a time t_l at which C_Π has read (almost) half of the inputs, i.e., $s/2 - o(s)$ bits. Let F (indicated by the dashed line) be the rectangular intersection of the plane $t = t_l$ with the area-time solid, as shown in Figure 3. Clearly, $A_F = A$.

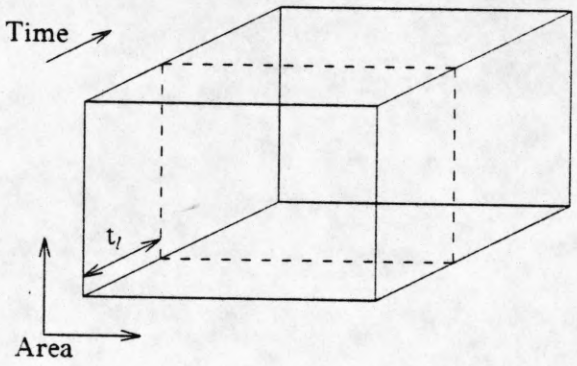


Figure 3. Area-time solid with temporal bisection of inputs

This bisection also yields a two-processor system. Here, P_B and P_E, associated respectively with the beginning ($0 \leqslant t \leqslant t_l$) and end ($t_l < t \leqslant T$) of the computation of C_Π, cooperate to solve $\Pi(s)$ (see Figure 4). We denote by $\psi_\Pi(s)$ the number of bits that P_B and P_E communicate to solve $\Pi(s)$. Because the electrical circuitry of the chip must be causal, information cannot flow backwards in time, and so this communication is strictly one-way, from P_B to P_E.

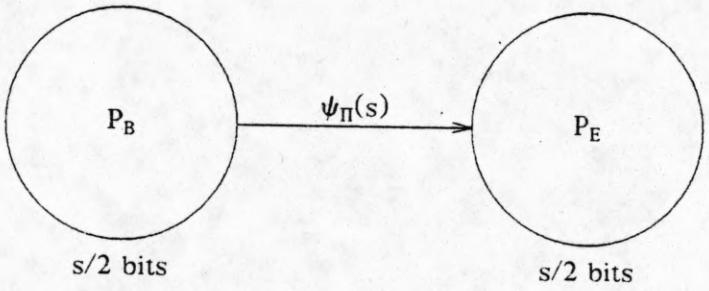


Figure 4. A temporal two-processor system

We can now state the following theorem relating A to $\psi_\Pi(s)$.

Theorem 1: Any chip that solves $\Pi(s)$ must have area satisfying

$$A = \Omega(\psi_\Pi(s)). \tag{3}$$

Proof: As above, let us first assume that there is in fact a time t_l when $s/2 - o(s)$ bits have been read. Then the rectangle F (see Figure 3) represents the communication channel from P_B to P_E. All information that crosses F must be encoded in the chip's state (i.e., stored in its memory) at time t_l. Since the storage of a bit requires some constant amount of area under any realistic assumptions, $A = A_F = \Omega(\psi_\Pi(s))$.

Now, if there is no such time t_l, then at some instant $\Omega(s)$ bits must be read simultaneously. This requires the existence of $\Omega(s)$ input ports, which would occupy $\Omega(s)$ area. Thus, $A = \Omega(s)$ in this case. But $\psi_\Pi(s) \leqslant s/2$, since P_B can simply send all of its inputs to P_E. Therefore, in this case, we also have $A = \Omega(\psi_\Pi(s))$. □

The above theorem gives us a convenient relationship between the area complexity of a VLSI chip and the one-way communication complexity of a two-processor system. However, because two-way communication complexity is the measure of interest in the proof of AT^2 lower bounds, it is convenient to relate area to this measure also. If we denote by $\psi_\Pi^{\leftrightarrow}(s)$ the number of bits that P_B and P_E must communicate to solve $\Pi(s)$ when two-way communication is allowed, then obviously $\psi_\Pi^{\leftrightarrow}(s) \leqslant \psi_\Pi(s)$. Thus, we have the following corollary.

Corollary 1: Any chip that solves $\Pi(s)$ must have area satisfying

$$A = \Omega(\psi_\Pi^{\leftrightarrow}(s)). \tag{4}$$

Although this bound may in general be quite weak, we will find it sufficiently tight for many problems.

Input/output protocol constraints are often established in VLSI computation theory. Such constraints reflect realistic assumptions regarding the physical structure of VLSI chips and the computing environments in which these chips might be used, they simplify the combinatorics involved in the lower bound arguments, and they avoid redundant solutions. Here, we will investigate the

dual roles played by area and time in the proof of these lower bounds. In particular, we will show how a spatial constraint on the input/output protocol, which may be used to bound $\phi_\Pi(s)$, corresponds to a temporal constraint, which may be used to bound $\psi^*_\Pi(s)$. The fundamental observation here is that ϕ (ψ^*) depends only on the distribution of input/output variables between P_L and P_R (P_B and P_E), and that the class of allowed distributions is governed by the spatial (temporal) input/output protocol constraints.

We begin by summarizing typical input/output protocol constraints. For the purpose of this discussion, we will assume that the input is organized as n words, each with k bits. First, we have spatial constraints:

(A1) Unilocal: Each input/output bit is available at only one port (but perhaps at several time instances);

(A2) Place-determinate: Input/output data are available at a prespecified (instance-independent) place;

(A3) Word-local: For any cut l partitioning the chip, o(n) input (output) words enter (exit) the chip on both sides of l;

(A4) Bit-local: For any cut l partitioning the chip, o(k) input (output) bit positions enter (exit) the chip on both sides of l.

Second, we have temporal constraints:

(B1) Semellective: Each input/output bit is available at only one time instance (but perhaps at several ports);

(B2) Time-determinate: Input/output data are available at a prespecified (instance-independent) time;

(B3) Word-serial: At any time instance, at most one input (output) word has some, but not all, of its bits already read (written);

(B4) Word-parallel: At any time instance, for all but at most one l, either all or none of the lth significant bits of the input (output) words are already read (written).

When A1 and A2 (B1 and B2) are the only protocol constraints extant, the protocol is said to be non-word-local (non-word-serial).

Now, we will discuss the manner in which these constraints restrict the class of distributions of input variables allowed in the two-processor system. Constraint A1 ensures that any particular input/output bit resides in either P_L or P_R, but not both. Correspondingly, constraint B1 ensures that any particular input/output bit resides in either P_B or P_E, but not both. Constraint A2 (or B2) ensures that, for all problem instances of a given input size, any particular input/output bit resides always in the same processor. Constraint A3 distributes the input/output bits between P_L and P_R essentially by word (possibly with o(n) words fragmented across processors). Constraint B3 corresponds to A3 but is somewhat stronger. It distributes the input/output bits between P_B and P_E also by word (with o(1) words fragmented across processors). Constraint A4 distributes the input/output bits between P_L and P_R essentially by their position in their respective words (possibly with o(k) positions fragmented across processors). Constraint B4, similar but stronger, distributes the input/output bits between P_B and P_E also by bit position (with o(1) positions fragmented across processors). Because of this correspondence (see Figure 5), any theorem lower bounding ϕ (and hence $\sqrt{AT}$) that is predicated on some combination of A1-A4 immediately yields a theorem lower bounding ψ (and hence A) that is predicated on a corresponding combination of B1-B4. Hereafter, we will use the notation of the spatial two-processor system to establish lower bounds on ϕ. From the previous discussion, it is clear that the same arguments can be used to establish lower bounds on ψ.

spatial constraints (for $\sqrt{AT}$ bounds)	temporal constraints (for A bounds)
unilocal	semellective
place-determinate	time-determinate
word-local	word-serial
bit-local	word-parallel

Figure 5. Table summarizing correspondence between spatial and temporal constraints

3. Transformability in VLSI

In this section, we will develop a general theory for establishing lower bounds on AT^2 and A. In Section 2, it was shown that the bisection flow fully captures both the AT^2 and the A measure of complexity.

Following the notation of Preparata-Shamos [PS], consider two problems $\Pi_1(s_1)$ and $\Pi_2(s_2)$, and assume that a two-processor system $P_{\Pi_1(s_1)}$ is available that solves $\Pi_1(s_1)$. Problem $\Pi_2(s_2)$ can be solved as follows.

1) The input to problem $\Pi_2(s_2)$ is converted into a suitable input to problem $\Pi_1(s_1)$.

2) $P_{\Pi_1(s_1)}$ is used to solve $\Pi_1(s_1)$.

3) The output of $\Pi_1(s_1)$ is transformed into a solution to problem $\Pi_2(s_2)$.

Thus, it is said that problem $\Pi_2(s_2)$ has been transformed to problem $\Pi_1(s_1)$. If steps 1 and 3 (above) can be done by transmitting $\phi_{2,1}(s_2)$ bits between the two processors in $P_{\Pi_1(s_1)}$, then $\Pi_2(s_2)$ is said to be $\phi_{2,1}(s_2)$ - transformable to $\Pi_1(s_1)$, we write: $\Pi_2(s_2) \overset{\phi_{2,1}(s_2)}{\rightarrow} \Pi_1(s_1)$.

Proposition: If problem $\Pi_2(s_2)$ is known to require $\phi_{\Pi_2}(s_2)$ bits of information flow and $\Pi_2(s_2)$ is $\phi_{2,1}(s_2)$-transformable to $\Pi_1(s_1)$, then $\Pi_1(s_1)$ requires information flow of at least $\phi_{\Pi_2}(s_2) - \phi_{2,1}(s_2)$ bits in the two-processor system associated with $\Pi_1(s_1)$.

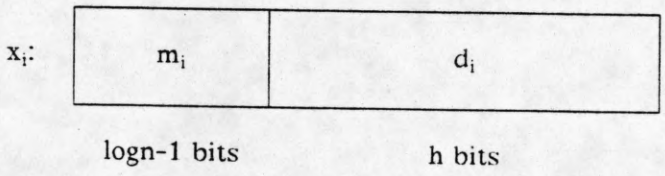
Now we need to search for a problem $\Pi(s)$ for which we can establish a lower bound of $\phi_\Pi(s)$ on the information flow and a transformation $\Pi(s) \overset{o(\phi_\Pi(s))}{\rightarrow} \Pi'(s')$, for many related problems $\Pi'(s')$. $\Pi(s)$ then serves as a computational prototype for this class of related problems. A good computational prototype for a complexity class must be a simple problem, which makes it difficult to establish a lower bound on its bisection flow complexity. Indeed, this is the case for computational prototypes in other models of computation (e.g., satisfiability in the theory of NP-completeness). In this paper, we choose element uniqueness (EU) as a computational prototype.

EU(n,h): Given n inputs $(x_1, \ldots, x_n)$, each of which is represented with h+logn-1 bits, decide if they are all unique ($h \geq 1$, otherwise the problem is trivial). By convention, if they are all unique, then the output (one bit) is 1, otherwise the output is 0.

The following framework will be used to establish a lower bound on the communication complexity of any two-processor system that solves EU(n,h). Consider a decision problem $\Pi(s)$, where s is the number of inputs and let $P_\Pi(s)$ be a two-processor system that solves $\Pi(s)$. Consider a matrix $M_\Pi(s)$, called the *result matrix* of $\Pi(s)$, with all $2^{s/2}$ possible values of inputs in P_L as its row indices and all $2^{s/2}$ possible values of inputs in P_R as its column indices. The (i,j)-th entry of the matrix is the output of $\Pi(s)$ when its input corresponds to the values of i and j. It has been shown by [Y1, MS]:

$$\phi_\Pi(s) = \Omega[\log \text{ rank } (M_\Pi(s))] \tag{5}$$

In what follows, we show that $\phi_{EU}(n,h) = \Omega(nh)$ under the word-local protocol (Lemma 1) and also under the bit-local protocol (Lemma 2). The two results will be combined in Theorem 2 to show the same lower bound for EU under the non-word-local protocol. Consider the input data organized as an array, with each word constituting a row and with the bit positions aligned as columns. We begin by partitioning the input array as X = [M,D], where M (the matching part) and D (the data) are blocks of logn-1 and h columns:



The bits of M will be used to enforce an appropriate matching of the input words, which will be specified later. Subsequently, we will be concerned only with the information flow induced by D, and all bisection arguments will be based on the bits of D.

Lemma 1: Under the word-local protocol assumption (A3), $\phi_{EU}(n,h) = \Omega(nh)$.

Proof: The proof is based on a restriction of element uniqueness to pair-wise element uniqueness. Without loss of generality, we assume that d_i enters P_L for $0 \leqslant i < n/2$ and it enters P_R for $n/2 \leqslant i < n$. We will prove a lower bound on the flow by considering the restricted class of input assignments such that:

$$m_i = i, \text{ for } 0 \leqslant i < \frac{n}{2}, \text{ and}$$

$$m_i = i - \frac{n}{2}, \text{ for } \frac{n}{2} \leqslant i < n.$$

In essence, we have partitioned the inputs into n/2 pairs, where each pair contains d_i and $d_{i+n/2}$ for $0 \leqslant i < n/2$. The two members of each pair are in a different processor (one in P_L, the other in P_R). Thus, the elements are not unique (output = 0) if $d_i = d_{i+n/2}$ for any $0 \leqslant i < n/2$. It can be shown by a generalization of the argument in [MS] that the result matrix has full rank ($2^{nh/2}$), and thus the flow has a bound of $\Omega(nh)$[GLTWZ]. □

Under the word-local assumption, each bit of a given input word enters the same processor (P_L or P_R). Now, we consider the "opposite case," where half of the input bit positions are assigned to each processor.

Lemma 2: Under the bit-local protocol assumption (A4), $\phi_{EU}(n,h) = \Omega(nh)$, for $h = O(\log n)$.

Proof: The fragment of d_i in P_L (P_R) is denoted by d_i^L (d_i^R). As before, we restrict our attention to a particular class of inputs, one that forces a pairing of a word fragment in P_L with one in P_R.

We partition D into $\lfloor n/H \rfloor$ groups of H elements each, where $H = 2^{h/2+1}$. In the jth group $(0 \leqslant j < \lfloor n/H \rfloor - 1)$, let:

$$m_i = j, \text{ for } 0 \leqslant i < H,$$

$$d^L_{jH+i} = d^R_{jH+i} = i, \text{ for } 0 \leqslant i < H/2, \text{ and}$$

$$d^L_{jH+i} = \pi_j(i-H/2),\ d^R_{jH+i} = \sigma_j(i-H/2), \text{ for } H/2 \leqslant i < H,$$

where π_j is an arbitrary permutation of $\{0,\ldots,H/2-1\}$ and σ_j is an arbitrary mapping $\{0,\ldots,H/2-1\} \rightarrow \{0,\ldots,H/2-1\}$:

	D^L	D^R	M
	h/2 bits	h/2 bits	logn-1 bits
d_{jH}:	0	0	j
d_{jH+1}:	1	1	j
	. . .	. . .	
$d_{jH+H/2-1}$:	$2^{h/2}-1$	$2^{h/2}-1$	
$d_{jH+H/2}$:	$\pi_j(0)$	$\sigma_j(0)$	
	. . .	. . .	
d_{jH+H-1}:	$\pi_j(2^{h/2}-1)$	$\sigma_j(2^{h/2}-1)$	j

In this setting, the elements are not unique (output=0) if $\pi_j(i) = \sigma_j(i)$ for $0 \leqslant i < H/2$ and any j. There are (H/2)! permutations of π_j for any j. The result matrix for any one group is the submatrix obtained by deleting certain rows from the matrix introduced in Lemma 1. Thus, it also has full rank, i.e., (H/2)!. The overall result matrix is the Kronecker product of the group matrices, and its rank is therefore the product of the ranks of these matrices:

$$\prod_{i=1}^{\lfloor n/H \rfloor} (H/2)!\ .$$

From Equation 5, we can establish the desired bound on the flow:

$$\phi = \log \prod_{i=1}^{\lfloor n/H \rfloor} (H/2)! = \sum_{i=1}^{\lfloor n/H \rfloor} \log(H/2)! = \lfloor n/H \rfloor (H/2 \log H/2) = \Omega(n \log H),$$

and, because $H = 2^{h/2+1}$, $\phi = \Omega(nh)$. □

Now we will extend the results of Lemmas 1 and 2 to the non-word-local protocol assumption. In this situation, any bit of any word may enter either of the two processors.

Theorem 2: Under the non-word-local protocol assumption, $\phi_{EU}(n,h) = \Omega(nh)$, for $h = O(\log n)$.

Proof: Our strategy is to show that, for an arbitrary (but fixed) partition of the input bits, a large portion of the input words must all be either "substantially" word-local or "substantially" bit-local. The set of input words is partitioned into two sets, the set B of *biased* words and the set U of *unbiased* words. Intuitively, a biased word is one with most of its bits in one processor (P_L or P_R), and an unbiased word is one with almost the same number of bits in each processor. More formally,

$$B = \{d_i \mid |d_i^L| > \frac{3h}{4} \text{ or } |d_i^R| > \frac{3h}{4}\}, \text{ and } U = \{d_i \mid \frac{h}{4} \leqslant |d_i^L| \leqslant \frac{3h}{4}\}.$$

Note that $B \bigcup U = D$ and $B \bigcap U = \varnothing$. We analyze the distribution of input bits in each processor according to the size of the sets B and U.

Case 1) $|B| \geqslant 3n/4$ (thus $|U| \leqslant n/4$):

We partition the biased words further into the *left-biased* B_L and the *right-biased* B_R, namely,

$$B_L = \{d_i \in B \mid |d_i^L| > \frac{3h}{4}\}, \text{ and } B_R = \{d_i \in B \mid |d_i^R| > \frac{3h}{4}\}$$

The total number of bits in P_L are nh/2. At most $n/4(3h/4) = 3nh/16$ of these bits belong to the set U, and at most $n(h/4) = nh/4$ of these bits belong to the set B_R. Thus, at least $nh/2 - (3nh/16 + nh/4) = nh/16$ bits in P_L belong to the words in B_L. A symmetric argument verifies that at least nh/16 bits in P_R belong to the words in B_R; thus, $|B_L|, |B_R| \geqslant n/16$. Consider two input words $d_i \in B_L$ and $d_j \in B_R$. Clearly, d_i^L has at least h/2 bit positions that correspond with positions in d_j^R.

The remaining h/2 bits (of both d_i and d_j) may be set to any arbitrary value. The result matrix has rank $2^{nh/32}$, and thus $\phi = \Omega(nh)$.

Case 2) $|B| < 3n/4$ (thus $|U| > n/4$):

Consider an arbitrary pairing of the elements of U and let (d_i,d_j) be one such pair. By the definition of unbiased, d_i must have at least h/4 bits in each processor. Each bit position in d_i^L (d_i^R) corresponds to a bit position in either d_j^L or d_j^R. We can distinguish two subcases:

Case 2a) Either h/8 positions in d_i^L correspond to positions in d_j^R or h/8 positions in d_i^R correspond to positions in d_j^L. These pairs are referred to as *word-type*.

Case 2b) In the event that (d_i,d_j) does not satisfy the conditions of Case 2a, then, by the pigeonhole principle, h/8 positions in d_i^L correspond to positions in d_j^L and h/8 positions in d_i^R correspond to positions in d_j^R. These pairs are referred to as *bit-type*.

Clearly, each of the n/8 pair of unbiased inputs is either word-type or bit-type. By another application of the pigeonhole principle, either there are at least n/16 word-type pairs or there are at least n/16 bit-type pairs. Thus, we either have a word-local setting or a bit-local setting. In either case, from Lemmas 1 and 2, we conclude $\phi = \Omega(nh)$. □

This implies, by virtue of Equation 1, that any chip with area A that solves EU(n,ϵlogn) in time T satisfies $AT^2 = \Omega(n^2\log^2 n)$ under the non-word-local assumption, and, by virtue of Equation 4, $A = \Omega(n\log n)$ under the non-word-serial assumption.

4. Applications

In this section, we demonstrate the application of the previous results to the establishment of AT^2 and A lower bounds for related problems. First, however, we prove the following lemma, which facilitates problem transformation by allowing us to relax the unilocal (semellective) assumption, A1 (B1). Instead, we now assume A1' (B1').

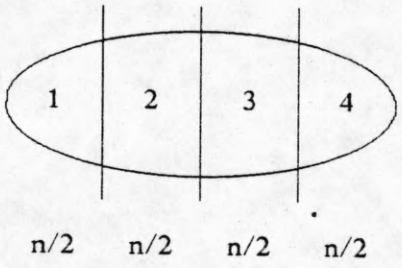
(A1') Bilocal: Each input/output bit is available at no more than two ports (but perhaps at several time instances).

(B1') Bilective: Each input/output bit is available at no more than two time instances (but perhaps at several ports).

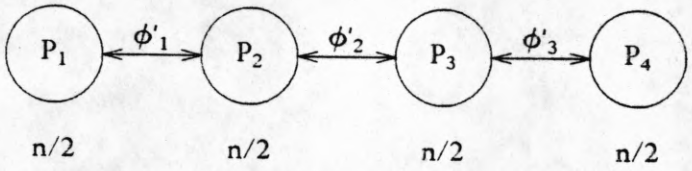
Let $\phi'_{EU}(n,h)$ denote the bisection flow under A1', A2, and A3. Obviously, the traditional bisection technique fails to establish any bound on $\phi'_{EU}(n,h)$ because each input may enter both processors (P_L and P_R). Nevertheless, we can still obtain a lower bound on $\phi'_{EU}(n,h)$ by employing a method similar to the bisection technique.

Lemma 3: $\phi'_{EU}(n,h) = \Omega(nh)$ for $h = O(\log n)$.

Proof: Consider any (convex) chip C'_{EU} that solves EU(n,h) under A1', A2, and A3. Let us partition the chip into four sections, by means of lines parallel to the shorter side of the minimum-area enclosing rectangle, such that each section contains n/2 input words (recall that there are now 2n input words: $\{x_0,x_0,x_1,x_1,\ldots,x_{n-1},x_{n-1}\}$). The general framework is one in which the four processors (P_1,P_2,P_3,P_4) associated with the four sections of C'_{EU} cooperate to solve EU(n,h) (see Figure 6).



a. C'_{EU} to solve EU(n,h)



b. a four-processor system

Figure 6

A straightforward modification of Equation 2 implies:

$$AT^2 = \Omega((\phi'_{EU}(n,h))^2),$$

where $\phi'_{EU}(n,h) = \max(\phi'_1,\phi'_2,\phi'_3)$. A lower bound on any one of the ϕ'_is cannot be established independent of the others, for it may be the case that processors to the left or right of the link associated with ϕ'_i have access to the entire input set and thus do not need to send or receive any infor-

mation to or from the other processors. In fact, this situation occurs when P_1 and P_3 each contain a copy of $(x_0, \ldots, x_{n/2-1})$, and P_2 and P_4 each contain a copy of $(x_{n/2}, \ldots, x_{n-1})$.

Our strategy is to partition the four processors into two sets, P_L and P_R, such that each set contains both copies of (at least) n/16 input words. These inputs can then be revealed to the other set only by information flow through the links connecting P_L and P_R. Since there are a total of n/2 inputs in P_1, and each input is repeated twice then there must be at least n/4 distinct inputs in P_1. The other copies of these n/4 input words are in P_1, P_2, P_3, or P_4. By the pigeonhole principle, P_1 and P_i (for some $1 \leq i \leq 4$) must contain both copies of at least n/16 input words. Let $P_L = \{P_1\} \bigcup \{P_i\}$ and $P_R = \{P_2, P_3, P_4\} - \{P_i\}$. We can view P_L–P_R as a two processor system with a flow ϕ_{LR} of $\Omega(nh/16)$ bits between P_L and P_R (see Lemma 1). Clearly, $\phi_{LR} \leq \phi'_1 + \phi'_2 + \phi'_3$, and thus, $\phi'_{EU}(n,h) = \max(\phi'_1,\phi'_2,\phi'_3) = \Omega(nh)$. □

From the discussion of section 2, it is clear that the temporal analog of Lemma 3 also holds under assumptions B1', B2, B3. Furthermore, A3 (B3) may be replaced by A4 (B4) while maintaining the same flow bound. (The roles of n and h are simply reversed in Lemma 3.)

Now we demonstrate the problem transformation methodology by means of two examples. Specifically, two fundamental problems are shown to be at least as hard (in either the AT^2 or the A sense) as element uniqueness. We conclude with a brief catalog of related problems, together with lower bounds on their AT^2 and A complexity, as obtained via problem transformation.

The first problem is a fundamental one in computational geometry, namely, closest pair.

CP(n,h): Given a set of n points $p_i = (a_i,b_i)$ for $0 \leq i < n$, where each coordinate is represented with h+logn-1 bits, find the closest pair of points.

We want to show $EU(n,h) \overset{o(nh)}{\rightarrow} CP(n,h)$. Assume there is a two-processor system $P_{CP}(n,h)$ that solves CP(n,h). This system can then be used to solve EU(n,h) under the non-word-local assumption (A1 and A2) in the following manner.

1) The coordinates of each point are set as $p_i = (x_i,0)$, which is a trivial transformation.

2) P_{CP} is used to solve this (restricted) closest pair problem. (The chip is bisected in such a way that each processor inputs half of the "meaningful" data, that is, x_is.)

3) Once the closest pair of points is determined, P_L sends all of its output bits (O(logn)) to P_R. P_R then computes the distance between the two points and outputs a 0 if the distance is equal to 0 and a 1 otherwise. It is clear that the output is 1 if and only if the elements are unique.

By Theorem 1, $\phi_{EU}(n,h) = \Omega(nh)$. Since steps 1 and 3 above require the transmission of $O(\log n) = o(nh)$ bits, $EU(n,h) \overset{O(\log n)}{\rightarrow} CP(n,h)$. Theorem 3 follows immediately from the proposition.

Theorem 3: Under the non-word-local protocol assumption, $\phi_{CP}(n,h) = \Omega(nh)$ for $h = O(\log n)$.

Thus, any chip with area A that solves CP(n,elogn) in time T satisfies $AT^2 = \Omega(n^2\log^2 n)$ under the non-word-local assumption, and $A = \Omega(n\log n)$ under the non-word-serial assumption.

Now we establish a ϕ lower bound on the problem of finding the size of the maximum clique in an interval graph (MCIG) by showing that EU is transformable to it. This serves as an excellent illustration of the utility of the previous results.

MCIG(n,h): Given a collection of intervals $I_i=(l_i,r_i)$ for $1 \leqslant i \leqslant n$, where l_i and r_i are respectively the left and right endpoints of interval I_i, we can define a graph $G = (V,E)$, where $V=\{I_i \mid 1 \leqslant i \leqslant n\}$ and $E=\{(I_i,I_j) \mid I_i \cap I_j \neq \phi,\ 1 \leqslant i,j \leqslant n\}$. Such a graph is called an *interval graph*. Let h+logn-1 be the length of the integers used to represent the l_is and r_is, i.e., $0 \leqslant l_i,r_i, \leqslant n2^{h-1}-1$ for $1 \leqslant i \leqslant n$. The problem is to find the size of the maximum clique in this graph.

We want to show $EU(n,h) \overset{o(nh)}{\rightarrow} MCIG(n,h)$. Assume there is a two-processor system $P_{MCIG}(n,h)$ that solves MCIG(n,h). This system can then be used to solve EU(n,h) under the bilocal assumption (A1', A2, and A3) in the following manner.

1) Each interval is set as $I_i = (x_i,x_i)$, which is a trivial transformation due to the bilocality.

2) P_{MCIG} is used to solve this (restricted) maximum clique problem.

3) Ignoring the least significant bit, P_L and P_R form the logical OR of their output bits. P_R then sends its result to P_L (or vice versa), and P_L outputs the NOR of its result and that of P_R. This requires exactly one bit of additional communication. It is clear that the output is 1 if and only if the elements are unique.

By Lemma 3, $\phi'_{EU}(n,h) = \Omega(nh)$. Since steps 1 and 3 above require the transmission of one bit, $EU(n,h) \overset{O(1)}{\rightarrow} MCIG(n,h)$. Theorem 4 follows immediately from the proposition.

Theorem 4: Under the word-local protocol assumption, $\phi_{MCIG}(n,h) = \Omega(nh)$ for $k = O(\log n)$.

Thus, any chip with area A that solves $MCIG(n,\epsilon\log n)$ in time T satisfies $AT^2 = \Omega(n^2\log^2 n)$ under the word-local or bit-local assumption, and, $A = \Omega(n\log n)$ under the word-serial or word-parallel assumption.

Element uniqueness can be transformed to a large number of related problems. Here, we list a few. All of these problems have $AT^2 = \Omega(n^2\log n^2)$ and $A = n\log n$.

1) Visibility problem: Given a collection of vertical segments $S_i = (b_i,t_i)$, where b_i and t_i are respectively the bottom and the top points of S_i, for $1 \leqslant i \leqslant n$, find all pairs of segment that "see" each other -- two segments S_i and S_j "see" each other if and only if there exist a horizontal segments that crosses only S_i and S_j. Element uniqueness is o(1) transformable to this problem even if only one such a pair is desired.

2) Interval graph problems: maximum independent set, minimum clique cover, and minimum dominating set in interval graphs.

3) Proximity problems: all closest pairs, euclidean minimum spanning tree, Delaunay triangulation, convex hull.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Franco Preparata for his guidance during this research. We also acknowledge many helpful and enlightening discussions with Gianfranco Bilardi, Prasoon Tiwari, and Doug West.

REFERENCES

[AA] Abelson, H. and Andreae, P., "Information Transfer and Area-Time Trade-offs for VLSI Multiplication," *Communications of the ACM*, vol. 23, no. 1, 1980, pp. 20-22.

[BP] Bilardi, G. and Preparata, F. P., "Tessellation Techniques for Area-Time Lower Bounds with Applications to Sorting," to appear in *Algorithmica*, Mar. 1986.

[BK] Brent, R. P. and Kung, H. T., "The Chip Complexity of Binary Arithmetic," *Journal of the ACM*, vol. 28, 1981, pp. 521-534.

[DSVT] Ďuriš, P., Sýkora, O., Vrťo, I., and Thompson, C. D., "Tight Chip Area Lower Bounds for Discrete Fourier and Walsh-Hadamard Transformations," *Information Processing Letters*, vol. 21, no. 5, Nov. 1985, pp. 245-247.

[GJ] Garey, M. R. and Johnson, D. S., *Computers and Intractability*, W. H. Freeman and Co., 1979.

[GLTWZ] Gafni, E., Loui, M. C., Tiwari, P., West D. B., and Zaks, S., "Lower Bounds on Common Knowledge in Distributed Algorithms," technical report ACT-50, Coordinated Science Laboratory, University of Illinois, 1984.

[L] Leighton, F. T., "Tight Bounds on the Complexity of Parallel Sorting," *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing*, Washington D.C., Apr. 1984, pp. 71-80.

[MS] Mehlhorn, K. and Schmidt, E. M., "Las Vegas is Better than Determinism in VLSI and Distributed Computing," *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing*, San Francisco, May 1982, pp. 330-337.

[PS] Preparata, F. P. and Shamos, M., *Computational Geometry*, Springer-Verlag, 1985.

[S] Siegel, A. R., "Minimum Storage Sorting Networks," *IEEE Transactions on Computers*, vol. C-34, no. 4, Apr. 1985, pp. 355-361.

[T] Thompson, C. D., *A Complexity Theory for VLSI*, Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, 1980.

[U] Ullman, J. D., *Computational Aspects of VLSI*, Computer Science Press, 1983.

[V] Vuillemin, J., "A Combinatorial Limit to the Computing Power of VLSI Circuits," *IEEE Transactions on Computers*, vol. C-32, no. 3, Mar. 1983, pp. 294-300.

[Y1] Yao, A. C., "Some Complexity Questions Related to Distributive Computing," *Proceedings of the 11th Annual ACM Symposium on the Theory of Computing*, Atlanta, Apr. 1979, pp. 209-213.

[Y2] Yao, A. C., "The Entropic Limitations on VLSI Computations," *Proceedings of the 13th Annual ACM Symposium on the Theory of Computing*, Milwaukee, May 1981, pp. 308-311.