INFORMATION AND COMPUTATION 90, 178–193 (1991)

Planar Acyclic Computation

W. F. McColl*

Programming Research Group, Oxford University, 11 Keble Road, Oxford OX1 3QD, England

M. S. Paterson*, †

Department of Computer Science, University of Warwick, Coventry CV4 7AL, England

AND

B. H. Bowditch‡

Mathematics Institute, University of Warwick, Coventry CV4 7AL, England

This paper considers the following problem: given a specification consisting of a set of variables X, a multiset of functions F on those variables, and a cyclic ordering on $X \cup F$, determine whether or not there exists a planar acyclic circuit which realizes the specification. An algorithm is given which produces such a circuit whenever one exists. In proving that our algorithm meets this requirement we provide some simple mathematical characterizations of those specifications which are realizable.

1. Introduction

Every Boolean function of n arguments can be computed by an acyclic circuit constructed from input nodes corresponding to the arguments and two-input gates corresponding to binary Boolean functions. Every polynomial over a field F can be computed by an acyclic circuit constructed from input nodes corresponding to the arguments, constant nodes corresponding to the elements of F, and two-input gates corresponding to the binary operations $\{+, -, *, /\}$. Such universality properties have resulted in the acceptance of acyclic circuits as a fundamental model of computation for

* Supported in part by the Esprit Basic Research Action No. 3075, "Algorithms and Complexity."

† This author was supported by an SERC Senior Fellowship.

‡ This author was supported by an SERC Research Fellowship.

0890-5401/91 $3.00

arithmetic and Boolean functions. In recent years there have been a number of major advances in our understanding of the circuit complexity of functions. For an account of the circuit complexity of Boolean functions, see (Dunne, 1988, Wegener, 1987). Results on the circuit complexity of arithmetic functions can be found in (Borodin and Munro, 1975, von zur Gathen, 1987).

Circuit complexity has traditionally been measured in terms of circuit size, i.e., the number of gates. However, recent technological developments have resulted in a much greater emphasis being placed on the design of highly parallel circuits which minimize depth (rather than size) and on the design of circuits which can be directly fabricated as VLSI devices. The circuit depth of functions is discussed in (Borodin and Munro, 1975, von zur Gathen, 1986, Miller, Ramachandran, and Kaltofen, 1986, Reif, 1983, Wegener, 1987). Results on the VLSI complexity of functions can be found in (Lengauer, 1985, Ullman, 1984, Wegener, 1987). Although most of the work on VLSI complexity has assumed a technology which is essentially two-dimensional, with perhaps a fixed number of layers, there has been some research on the problem of embedding circuits in three-dimensional space (Gupta and Hambrusch, 1987, Leighton and Rosenberg, 1986, Preparata, 1983, Rosenberg, 1983), and also on the problem of embedding circuits in "books" (Yannakakis, 1986a, 1986b).

In this paper we consider the computation of functions by acyclic circuits embedded in the plane without crossing edges. The planar circuit model of computation was studied by Lipton and Tarjan (1980) who proved a number of interesting and important lower bounds on the planar circuit size of Boolean functions using their separator theorem. In (Savage, 1981, 1984), Savage showed that planar circuit size was closely related to the AT^2 complexity measure used for VLSI circuits (Ullman, 1984, Wegener, 1987) and, in doing so, simplified and unified a number of earlier lower bounds on VLSI complexity.

DEFINITION. A *planar crossover* is a planar acyclic circuit whose ordered inputs are $\langle x, y \rangle$ and whose ordered outputs are $\langle y, x \rangle$.

EXAMPLE 1. Figure 1 shows a planar Boolean crossover constructed from gates corresponding to the function $\oplus(p, q) = (p + q)$ modulo 2.

The existence of planar crossovers (Lipton and Tarjan, 1980, McColl, 1981, Savage, 1981) is often taken as sufficient justification for the statement that the planar circuit model is universal. However, Aggarwal (1983) made the interesting observation that the conversion of a general acyclic circuit into a planar circuit by the substitution of planar crossovers for crossing edges might introduce a cycle into the graph. For the kinds of

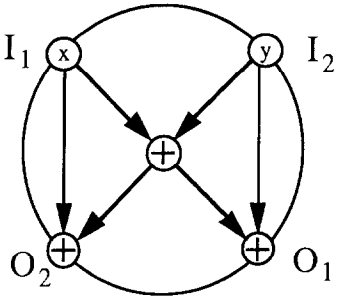

FIGURE 1

planar circuit considered in (Lipton and Tarjan 1980, Savage, 1981), where the inputs and outputs are segregated from each other on the boundary, this problem can easily be resolved by first topologically ordering the gates.

In this paper we consider further the problem raised initially by Aggarwal. We investigate circuits in which the inputs and outputs are interleaved in some way on the boundary of the circuit. Our results show that for many such circuits it is not possible to replace crossing edges by planar crossovers without introducing a cycle. The results are given in terms of an algorithm which, given an input/output specification consisting of

(i) a set of variables X,

(ii) a multiset of functions F on those variables, and

(iii) a cyclic ordering on $X \cup F$ (representing the required sequence of inputs and outputs round the perimeter),

will produce a planar acyclic circuit which realizes the specification whenever one exists. In proving that our algorithm meets this requirement we provide simple mathematical characterizations of those specifications which are realizable. The results established apply to both arithmetic and Boolean circuits.

2. Definitions and Preliminary Results

Let $X_n = \langle x_1, x_2, ..., x_n \rangle$ be a set of n formal arguments and D be a set of values. $D_n = \{f(X_n): D^n \to D\}$ will be used to denote the set of n-argument functions over the domain D.

Let π be a finite region of the plane bounded by a simple closed curve γ. Without loss of generality we may assume the region to be convex. An *I/O specification* consists of a set $I = \{I_1, I_2, ..., I_n\}$ of input ports on γ and a disjoint set $O = \{O_1, O_2, ..., O_m\}$ of output ports on γ, where each I_j is associated with the argument x_j and each O_j is associated with some

$f_j \in D_n$. Note that the cyclic ordering of the ports on γ is part of the I/O specification. We shall see later that it is a very important part.

A *circuit over the basis* Ω, $\Omega \subseteq D_2$, is a directed acyclic graph, where (i) nodes have either in-degree 2 (*gates*) or in-degree 0 (*input nodes*), (ii) each gate has an ordering on its two inputs and corresponds to some function in Ω, and (iii) all output nodes are required to have out-degree 0.

$D_n^{(\Omega)}$ is the set of functions in D_n which can be realized by circuits over the basis Ω.

A *circuit layout for* an I/O specification is a circuit embedded in π as a planar graph with the following properties:

(i) to each I_j, $1 \leqslant j \leqslant n$, is mapped a unique node corresponding to x_j,

(ii) to each O_j, $1 \leqslant j \leqslant m$, is mapped an output node computing f_j.

A planar crossover is then a circuit layout for the specification with $n = m = 2$ given by $f_1 = x_1$, $f_2 = x_2$, and the cyclic order I_1, I_2, O_1, O_2.

In (McColl, 1981), the set of Boolean bases which permit the realization of a planar crossover was characterized. For example, there is a planar crossover for the complete basis $\{\text{NAND}\}$ and for the incomplete basis $\{\wedge, \rightarrow, \leftarrow\}$. On the other hand, there is no planar crossover for the monotone basis $\{\wedge, \vee\}$. The latter result shows that although every (single-output) monotone Boolean function f can be realized by a circuit over the basis $\{\wedge, \vee\}$, it is not necessarily the case that f can be realized by a planar monotone circuit. In (McColl 1985), examples are given of single-output monotone Boolean functions which cannot be realized by any planar monotone circuit. Planar arithmetic crossovers can easily be constructed for many arithmetic bases by using inverse operations, e.g., addition/subtraction or multiplication/division.

An I/O specification is *k-directional* if and only if the boundary γ can be partitioned into $2k$ or fewer segments such that each segment contains either only input ports or only output ports. Note that j-directional implies k-directional for $j < k$.

THEOREM 1. *A specification S has a circuit layout over* Ω *if the following three conditions hold*

(i) $f_1, f_2, \ldots, f_m \in D_n^{(\Omega)}$,

(ii) *there is a planar crossover for* Ω,

(iii) *S is 2-directional.*

Proof. We first prove the result for the case when S is 1-directional. It is straightforward to construct an acyclic (not necessarily planar) circuit over Ω which computes the set of functions $f_1, f_2, \ldots, f_m$. Let

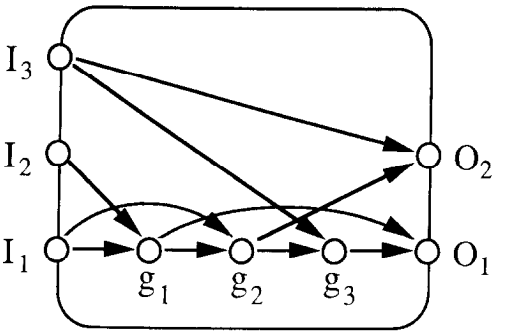

FIGURE 2

$\langle g_1, g_2, ..., g_p \rangle$ be a total ordering of the non-output gates which is consistent with the partial ordering given by the circuit. Let the given cyclic (clockwise) ordering of the ports in the I/O specification be $\langle I_{j_1}, ..., I_{j_n}, O_{k_m}, ..., O_{k_1} \rangle$. Locate I_{j_t} at $(0, t)$ and O_{k_t} at $(p+1, t)$. Locate each g_t at the point $(t, 1)$ and draw each edge as a path increasing monotonically in x as shown in Fig. 2, arranging the edges so that at most two cross at any point.

Replace each such crossing by an instance of the planar acyclic crossover circuit. This yields an acyclic planar layout and establishes the result for the 1-directional case.

Given a 2-directional I/O specification, use planar crossovers to produce two copies of each block of input ports as illustrated in Fig. 3. Now use the

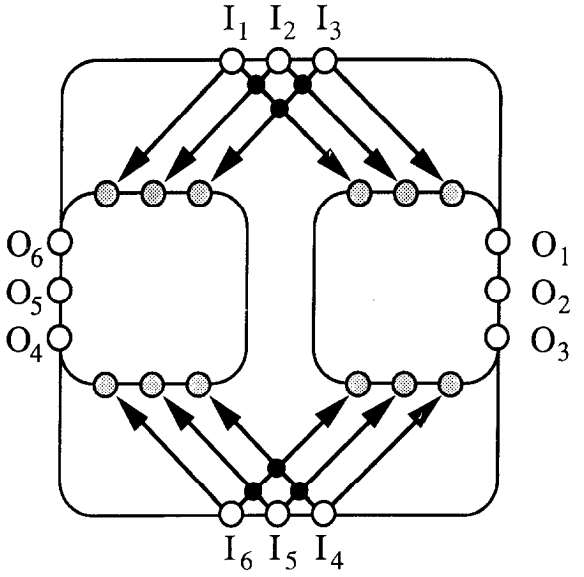

FIGURE 3

construction described above to produce planar acyclic circuits for the resulting pair of 1-directional specifications. ▮

DEFINITION. Given an I/O specification S, the *diagram for* S consists of the disjoint pair I, O, of sets, the cyclic ordering on $I \cup O$, and the *dependency relation* $R_S \subseteq I \times O$ defined by $R_S = \{\langle j, k\rangle \mid \exists a, b \in D, f_{k \mid x_j = a} \neq f_{k \mid x_j = b}\}$, i.e., $\langle j, k\rangle \in R_S$ if and only if f_k depends on x_j.

Our next task is to show that the geometric information necessary to determine whether or not a specification has a circuit layout is captured by its diagram.

It is convenient to extend the definition of k-directional (from specifications) to nodes in the obvious way. A node in a graph embedded in the plane is *k-directional* if its incident arcs in cyclic order can be partitioned into $2k$ or fewer blocks each containing either only incoming or only outgoing arcs.

A *wiring layout for* a diagram is a directed acyclic graph embedded in a simple closed region of the plane, with the following properties:

(i) each input in I (output in O) has a corresponding node with in-degree (out-degree) zero on the boundary of the region,

(ii) all other nodes are in the interior and are *2-directional*,

(iii) for each I/O pair in the diagram there is a directed path between the corresponding nodes in the wiring layout.

Since any 2-directional node can be replaced by a pair of 1-directional nodes as illustrated in Fig. 4, we could restrict the internal nodes of a wiring layout or circuit layout to be 1-directional with no essential loss.

EXAMPLE 2. Figure 5 shows a wiring layout for the diagram of the I/O specification for $m = n = 4$ given by the cyclic order of ports $\langle I_1, O_1, I_2,$

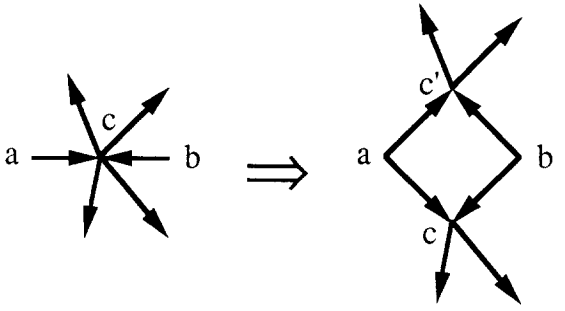

FIGURE 4

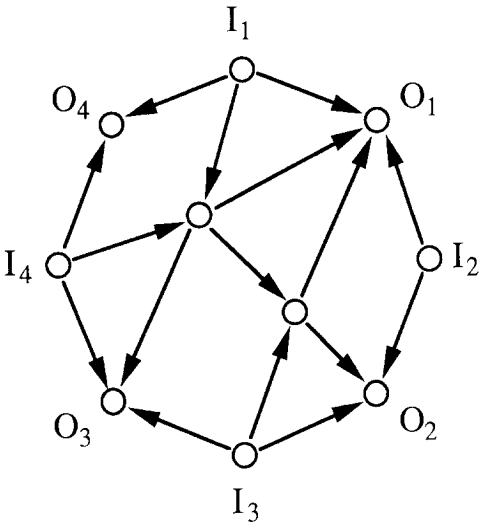

FIGURE 5

O_2, I_3, O_3, I_4, $O_4\rangle$ and the function definitions $f_1 = f_2 = x_1 \wedge x_2 \wedge x_3 \wedge x_4$, $f_3 = x_1 \vee x_3 \vee x_4$, $f_4 = x_1 \wedge x_4$.

LEMMA 1. *Given a specification S and a basis Ω which supports a planar crossover, S has a circuit layout over Ω if and only if the following two conditions hold*:

(i) $f_1, f_2, \ldots, f_m \in D_n^{(\Omega)}$,

(ii) *there exists a wiring layout for the diagram of S.*

Proof. If. Take a wiring layout for the diagram and choose a simple path following edges of the layout to correspond to each pair in the dependency relation R_S. Replace each edge which is traversed by k paths, say, by k parallel edges and replace each output node O_j by the appropriate acyclic circuit for f_j guaranteed by Theorem 1. At an interior node it is necessary to interconnect incoming and outgoing edges to continue the paths. Since such a node is 2-directional, this task corresponds to a 2-directional specification. A local circuit layout at each interior point is therefore also assured by Theorem 1, and merely performs a rearrangement of inputs and outputs using planar crossovers.

Only if. Any circuit layout for S yields a wiring layout simply by ignoring the gate information. We have only to observe that if the function associated with some output port O depends on the variable supplied at some input port I, then there must be a directed path in the circuit layout from I to O. ∎

3. Linear Wirings and Necklaces

Given a diagram D, a brutal approach to constructing a wiring layout would be to position the ports in the appropriate cyclic order on the boundary of a convex region, draw directed straight lines between each input/output pair in the dependency relation, and introduce an internal node wherever a pair of lines crosses. If the inputs and outputs are positioned so that no three of these lines are concurrent at an interior point, the resulting graph is a *linear wiring* for D. It is immediate that a linear wiring is a wiring layout for the diagram if and only if the linear wiring is acyclic. By proving that this acyclicity is equivalent to a combinatorial property of the diagram we shall establish its invariance and eventually show that the brutal approach is adequate.

Definition. For $k > 2$, a *k-necklace* in a diagram is a set of k arcs (i.e., ordered pairs in the relation), $\langle I_1, O_1 \rangle$, $\langle I_2, O_2 \rangle$, ..., $\langle I_k, O_k \rangle$, such that $\langle I_1, O_k, I_2, O_1, I_3, ..., O_{k-2}, I_k, O_{k-1} \rangle$ is in cyclic order.

Lemma 2. *For any diagram, D, the following conditions are equivalent:*

(i) *there is a linear wiring for D which contains a circuit,*

(ii) *in every linear wiring for D there is an elementary region (i.e., a face) for which the boundary is a circuit,*

(iii) *D contains a k-necklace for some k.*

Proof. First we observe that in every linear wiring if there is a circuit then there is a circuit which is the boundary of an elementary region. This is because any nonelementary region is divided into two smaller regions by one of the directed lines and if the boundary were a circuit then one of these subregions would also be a circuit.

Assume (i). By our observation there is some elementary, hence convex, region of the wiring whose boundary is a circuit. Consider now some convex region of maximal area whose boundary is a circuit. The edges of such a region define a necklace, since if any pair of these lines were to intersect outside the region, a larger region with a boundary circuit could be found, contradicting maximality. Hence (i) implies (iii).

Assume (iii). In every linear wiring for D, the lines corresponding to the arcs of the necklace yield a circuit. In the case $k = 3$, the circuit could degenerate to a point were it not for our assumption of non-concurrency. Hence, using our observation again, (iii) implies (ii).

Trivially, (ii) implies (i). ∎

Corollary 1. *For any diagram D, either every linear wiring for D has*

an elementary region for which the boundary is a circuit, or every linear wiring for D is acyclic.

It should be clear that the straightness of the lines in a linear wiring is not critical. Essentially the same proof can be followed for wirings with the properties that any two lines intersect at most once and no line intersects itself. This gives a characterization of a more topological nature.

We will show in a later section that if a diagram contains a necklace then it has no wiring layout. Hence to decide whether a diagram has a wiring layout, it suffices to construct a linear wiring L; if the boundary of each face of L is acyclic then L is a wiring layout, and if not then no wiring layout is possible. A drawback of this approach as a practical algorithm is the exact arithmetic required and we present in the next section an alternative step-by-step construction.

4. Main Construction

In this section we describe a procedure to produce a wiring layout for a diagram by following a sequence of steps each of which corresponds to a simplification of the diagram. Our construction is described in terms of three operations *Extract*, *Prune*, and *Coalesce*. Each of these takes a diagram D and produces a new diagram D' together with a partial wiring layout which relates the two diagrams.

Extract. This operation can be applied whenever there is an adjacent pair of related ports in D. If I_j and O_k are adjacent and related, then we can extract this relationship as follows. D' is obtained from D by eliminating the pair $\langle j, k \rangle$. A wiring layout W for D can then be obtained from a wiring layout W' for D' by adding an edge from I_j to O_k running so close to the perimeter that it does not intersect any edge of the layout W'.

Prune. This operation can be applied to any port P with no related ports. D' is just D with that port removed, and a wiring layout for D is obtained merely by placing a new node for P just outside the perimeter of D' between the neighbours of P and extending the perimeter appropriately.

Coalesce. This operation is applicable where there is a pair of adjacent input (output) ports. We will suppose that I_j and I_k are adjacent. (The case for output ports is analogous.) In D' both ports are removed and a new port I' substituted. I' appears in the cyclic order in the position of the ports it replaces and is related to all output ports which were related to either I_j or I_k. A wiring layout for D can be produced by adding new nodes for I_j and I_k, appropriately ordered, just outside the node corresponding to I' in a layout for D', and drawing directed edges to it from each of the new nodes (see Fig. 6).

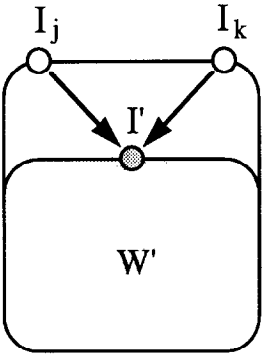

FIGURE 6

DEFINITIONS. A *reduction* is either an extraction, a pruning, or a coalescence. An *irreducible diagram* is a nonempty diagram in which no reductions are possible (so that, in particular, the input and output ports must alternate). Note that a necklace is irreducible.

The construction of a wiring layout is performed by the following recursive procedure, Layout, which makes an arbitrary sequence of reductions and succeeds if the empty diagram is reached. If we reach an irreducible diagram then the procedure fails, and we shall show that in this case no wiring layout was possible.

LAYOUT.

Input: a diagram D.
Output: a wiring layout for D unless the procedure fails.
If D has no nodes **then return** an empty layout,
else if D is irreducible **then** FAIL,
else make an arbitrary reduction, yielding a smaller diagram D' together with a partial wiring layout P, call Layout recursively on D' and **return** the wiring layout formed by combining P and Layout(D).

5. TREE LAYOUTS

We introduce a restricted form of wiring layout and show that this is adequate for wiring. The main theorem then follows easily.

DEFINITION. A *tree layout* for a diagram is a wiring layout with the following extra restrictions:

(i) it is acyclic even as an undirected graph (i.e., it is a tree),

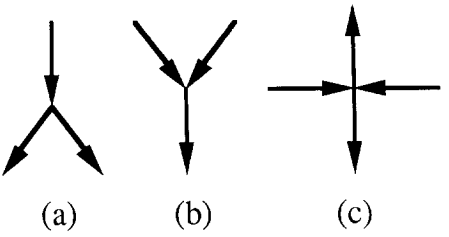

FIGURE 7

(ii) the nodes corresponding to inputs and outputs have degree 1,

(iii) internal nodes have either:

(a) in-degree 2 and out-degree 1,

(b) in-degree 1 and out-degree 2, or

(c) in-degree 2 and out-degree 2 with alternating incoming and outgoing arcs.

These three types are illustrated in Figs. 7a, b, and c.

THEOREM 2. *If diagram D has a wiring layout L then D has a tree layout.*

Proof. We begin by describing a series of transformations sufficient to eliminate all (undirected) circuits from L. (Conditions (ii) and (iii) are easily achieved at the end.) Within the (internal) faces defined by the planar graph L, we first introduce some extra edges in a way that maintains acyclicity. Let F be such a face (see Fig. 8).

(1) If F has more than three edges and $(a \to b)$, $(b \to c)$ are successive edges around F, then insert a new edge $(a \to c)$ in F. Since $(a \to c)$ is adjacent to $(a \to b)$ at a and adjacent to $(b \to c)$ at c, its introduction cannot violate the 2-directionality condition on nodes.

(2) Suppose F has more than five edges and $(a \to b)$, $(b \leftarrow c)$, $(c \to d)$, $(d \leftarrow e)$, $(e \to f)$, $(f \leftarrow g)$ are successive edges, where possibly $a = g$. If there

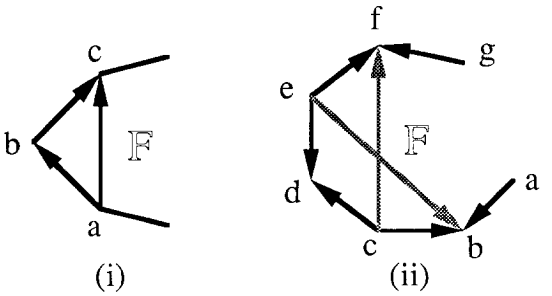

FIGURE 8

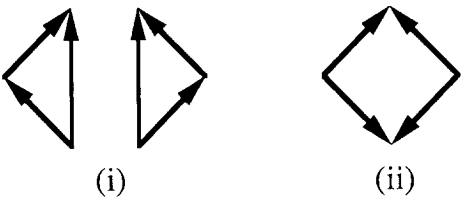

FIGURE 9

is no directed path in L from f to c then introduce a new edge $(c \to f)$ in F, otherwise introduce the edge $(e \to b)$.

As before, neither edge could violate 2-directionality. Acyclicity is maintained since it is impossible for L to have directed paths exterior to F both from b to e and from f to c. (Two such paths would have to meet, at node X say, and then the subpaths from b to X to c, together with the edge $(c \to b)$, would form a directed cycle.)

When these transformations have been applied wherever possible, let the resulting layout be L'. Because of (1), any face of L' with more than three edges has edges with alternating direction and hence an even number of edges. While because of (2), it has fewer than six edges. Since L' is acyclic the only possible forms for F are as shown in Fig. 9. We eliminate any face of type (ii) by inserting a 2-directional node of type (c) as shown in Fig. 10. Note that this transformation cannot introduce cycles or change the directionality of the remaining nodes. In the resulting triangulated graph, for every face F, there are nodes s and t such that the boundary of F consists of two distinct non-empty paths from s to t. This "flow" property is preserved by the following transformation which is used to eliminate faces.

An internal face of the layout with the fewest edges will be called a *minimal* face. Let F be any minimal face, and let s and t be its two nodes distinguished by the flow property. If the two nodes adjacent to s on the boundary of F are u_1, u_2, we can suppose without loss of generality that there is no path from u_2 to u_1, since paths in both directions would form a cycle. Remove the edge $(s \to u_2)$, and introduce the new edge $(u_1 \to u_2)$ if it does not already exist (see Fig. 11). If $(u_1 \to u_2)$ existed before, then F had only three sides and the new graph has one fewer face, otherwise F' has one fewer edge than F had and so is now the (unique) minimal face. The number of edges of a minimal face has been reduced by one.

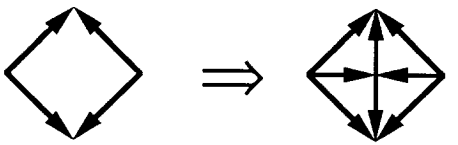

FIGURE 10

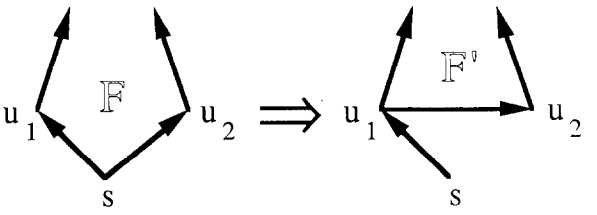

FIGURE 11

Continued repetition of the above transformation must eventually eliminate all faces, leaving a tree. The 1- and 2-directionality of the nodes is preserved. Finally each 1-directional node with in-degree u and out-degree v is replaced by a new edge $(a \to b)$ having a binary tree with u leaves fanning in to a and a binary tree with v leaves fanning out from b (see Fig. 12). The cases $u = 0$ and $v = 0$ correspond to input nodes and output nodes. Each 2-directional node is transformed to the required form by replacing each block of incoming or outgoing arcs by an appropriate binary tree. ∎

6. MAIN THEOREM

We have considered two methods for generating wiring layouts: linear wirings and the procedure "Layout." By Lemma 2, the first method fails only when the diagram contains a necklace, and the second fails only if an irreducible diagram is reached. It remains for us to show that in such cases no layout is possible.

LEMMA 3. *No wiring layout is possible for any diagram D which contains, or is reducible to, an irreducible diagram.*

Proof. Suppose to the contrary that there does exist a wiring layout W for D. Then W is also a layout for any diagram produced by removing arcs from D. If D' results from D by a pruning operation then a wiring layout

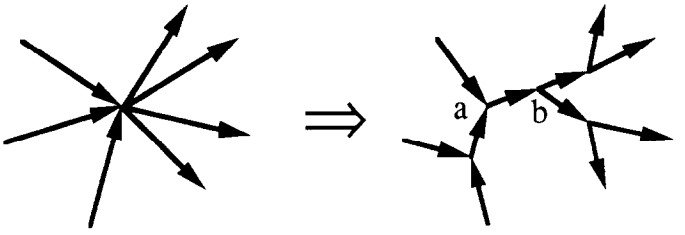

FIGURE 12

for D' can be produced by deleting the corresponding node and its incident arcs from W. If D' results from D by a coalescing operation then a corresponding layout can be produced by coalescing the two corresponding nodes in W.

Hence we may deduce that there is a wiring layout for an irreducible diagram D. By Theorem 2 there is also a tree layout T for D. Any irreducible diagram must have alternating input and output ports and at least six ports. Since the maximum degree of T is four, it has at least two internal nodes. Therefore there is some (extremal) internal node, a, which is adjacent to exactly one internal node, b, and two or three leaf nodes. Without loss of generality we can suppose that the edge between a and b is oriented as $(b \to a)$. Now at least one of the leaves adjacent to a must be an input node, and this has no path in T to a nonadjacent output node, contradicting the irreducibility of D. ▮

COROLLARY 2. *A 3-necklace, or "triple crossover," given by the I/O specification with* $m = n = 3$, *cyclic order of ports* $\langle I_1, O_3, I_2, O_1, I_3, O_2 \rangle$ *and the function definitions* $f_1 = x_1$, $f_2 = x_2$, $f_3 = x_3$, *has no planar acyclic realization.*

Our results are brought together in the following main theorem.

THEOREM 3. (I) *Given an I/O specification S and a basis* Ω, $\Omega \subseteq D_2$, *which supports planar crossovers, S has a circuit layout over* Ω *if and only if the following two conditions hold*:

(i) $f_1, f_2, \ldots, f_m \in D_n^{(\Omega)}$,

(ii) *there exists a wiring layout for* D_S, *the diagram of S.*

(II) *The following conditions are equivalent*:

(i) *the "Layout" procedure always succeeds on* D_S, *whatever sequence of reductions is chosen,*

(ii) *all linear wirings for* D_S *are acyclic,*

(iii) D_S *contains no necklace.*

Proof. Part (I) is just a restatement of Lemma 1. Part (II) follows immediately from Lemmas 2 and 3. ▮

7. CONCLUSION

Our results give a complete solution to the problem of determining whether or not a given I/O specification can be realized as a planar acyclic circuit constructed from two-input gates. The extension to m-input

2-directional gates is straightforward for any $m \geqslant 2$. However. if we allow 3-directional gates then specifications such as the triple crossover in Corollary 2 become immediately realizable.

It is well known that any graph can be embedded in three-dimensional Euclidean space in such a way that each edge corresponds to a straight line segment in R^3. For a proof of this result, see, for example, (White, 1984, Chap. 6). Let $2LG$ be the two layer grid, i.e., the subset of points $\langle x, y, z \rangle \in R^3$, where $0 \leqslant z \leqslant 1$ and at least two of x, y, z are integers. If we allow each edge to be constructed from a set of straight line segments then a simple variant of our construction can be used to show that any I/O specification can be realized as an acyclic circuit embedded in $2LG$ with no two edges meeting at an interior point.

If we restrict our attention to 1-directional specifications then Theorem 1 shows that any such specification can be realized as a planar acyclic circuit. The layout method used in the proof of this result may in some cases produce a circuit which is much larger than is necessary. As a challenge to planar acyclic circuit designers, we offer the following:

OPEN PROBLEM. Let S be a 1-directional I/O specification for which there is an acyclic circuit which can be embedded in the plane with g gates and c crossings. Is there always a planar acyclic circuit for such an S which is of size $O(g + c)$ and has no crossings?

RECEIVED July 13, 1988; FINAL MANUSCRIPT RECEIVED September 20, 1989

REFERENCES

AGGARWAL, A. (1983), On simulation and interrelations between Boolean and VLSI circuits, *in* "Proceedings, 21st Annual Allerton Conf, " pp. 258–265.

BORODIN, A., AND MUNRO, I. (1975), "The Computational Complexity of Algebraic and Numeric Problems," American Elsevier, New York.

P. E. DUNNE, (1988), "The Complexity of Boolean Networks," Academic Press, New York.

VON ZUR GATHEN, J. (1986), Parallel arithmetic computations: A survey, *in* "Proceedings, Mathematical Foundations of Computer Science 1986," Lect. Notes in Comput. Sci., Vol. 233, pp. 93–112, Springer-Verlag, New York/Berlin.

VAN ZUR GATHEN, J. (1987), Feasible arithmetic computations: Valiant's hypothesis, *J. Symbolic Comput.* **4** 137–172.

GUPTA, A. K., AND HAMBRUSCH, S. E. (1987), Optimal three-dimensional layouts of complete binary trees, *Inform. Process. Lett.* **26**, 99–104.

LEIGHTON, F. T., AND ROSENBERG, A. L. (1986), Three-dimensional circuit layouts, *SIAM J. Comput.* **15**, 793–813.

LENGAUER, T. (1985), The communication complexity of VLSI circuits, *in* "Proceedings, AMS Symp. in Applied Mathematics, 31," pp. 59–82.

LIPTON, R. J., AND TARJAN, R. E. (1980), Applications of a planar separator theorem, *SIAM J. Comput.* **9**, 615–627.

MCCOLL, W. F. (1981), Planar crossovers, *IEEE Trans. Comput.* **C-30**, 223–225.

MCCOLL, W. F. (1985), On the planar monotone computation of threshold functions, *in* "Proceedings, 2nd Annual Symp. on Theoretical Aspects of Computer Science," Lect. Notes in Comput. Sci., Vol. 182, pp. 219–230, Springer-Verlag, New York/Berlin.

MILLER, G. L., RAMACHANDRAN, V. AND KALTOFEN, E. (1986), Efficient parallel evaluation of straight-line code and arithmetic circuits, *in* "VLSI Algorithms and Architectures," Lect. Notes in Comput. Sci., Vol. 227, pp. 236–245, Springer-Verlag, New York/Berlin.

PREPARATA, F. P. (1983), Optimal three-dimensional VLSI layouts, *Math. Systems Theory* **16**, 1–8.

REIF, J. (1986), Logarithmic depth circuits for algebraic functions, *SIAM J. Comput.* **15** 231–241.

ROSENBERG, A. L. (1983), Three-dimensional VLSI: A case study, *J. Assoc. Comput. Mach.* **30**, 397–416.

SAVAGE, J. E. Planar circuit complexity and the performance of VLSI algorithms, *in* "VLSI Systems and Computations" (H. T. Kung, B. Sproull, and G. Steele, Eds.), pp. 61–68, Comput. Sci. Press, Rockville, MD, Expanded version appears as INRIA Report No. 77, INRIA, Rocquencourt, France, 1981.

SAVAGE, J. E. (1984), The performance of multilective VLSI algorithms, *J. Comput. System Sci.* **29**, 243–273.

ULLMAN, J. D. (1984), "Computational Aspects of VLSI," Comput. Sci. Press, Rockville, MD.

WEGENER, I. (1987), "The Complexity of Boolean Functions," Wiley–Teubner, New York/Stuttgart.

WHITE, A. T. (1984), "Graphs, Groups and Surfaces," North-Holland, Amsterdam.

YANNAKAKIS, M. (1986a), Four pages are necessary and sufficient for planar graphs, *in* "Proceedings, 18th ACM Symp. on Theory of Computing," pp. 104–108.

YANNAKAKIS, M. (1986b), Linear and book embeddings of graphs, *in* "VLSI Algorithms and Architectures," Lect. Notes in Comput. Sci., Vol. 227, pp. 226–235, Springer-Verlag, New York/Berlin.