LINEAR TIME ALGORITHMS FOR OPTIMAL CMOS LAYOUT

Ravi Nair, Anni Bruss
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

and John Reif
Aiken Computational Laboratory
Harvard University, Cambridge, MA 02138

We consider the problem of efficient CMOS circuit layout which has been formulated into an interesting graph-theoretical problem. A linear-time algorithm is described for optimal layout of a graph when the circuit topology is fixed. A further linear-time algorithm is provided to determine an optimal layout (i.e., having no diffusion gaps) when such a layout exists in some topology for the circuit. The key to our solution is a finite set of representative graphs which concisely describe topologically distinct paths in planar embedded series-parallel graphs.

I. Introduction

Considerable work has been done recently on the efficient implementation of MOS digital circuits. For a complete list of references see (Muroga82). However not much has appeared in the literature about the automatic generation of layout for such circuits. A few articles have been reported in the literature on the physical implementation of MOS complex cells (Schweikert71, Larsen71 and Gibson77). Uehara and vanCleemput (Uehara81) formulated the problem for CMOS complex cells. We present here an efficient (linear-time) and general algorithm for implementing CMOS complex functional cells. Previously no such algorithm has existed. Our technique is also suitable for implementation by a computer program.

Our paper is organized as follows: In the next section we describe the technological assumptions underlying the CMOS layout problem which is formulated in graph-theoretical terms in section III. A finite family of representative graphs is defined in section IV which is used to specify topologically distinct paths in planar embeddings of series-parallel graphs. In section V we describe how these representative graphs can be composed such that paths in a series-parallel graph can be recursively determined from the representative graphs of its components. Finally in section VI we present two linear-time algorithms: one to determine an optimal CMOS layout for a circuit with fixed topology, the other to determine an optimal layout (if it exists) in some topology. Note that all the proofs of the theorems, corollaries and propositions may be found in (Nair83).

II. Technology assumptions

The basic assumptions in this paper are the same as those in (Uehara81). Only those functions that are negative unate in their inputs can be implemented by a single CMOS complex cell (Muroga79). This implies that implementation of functions which depend on both the true and complemented forms of some input variable must have both forms as inputs. Further, only those circuits which can be logically represented by a series-parallel network of transistors will be considered.

***Preprint of paper appearing* in *International Workshop on Parallel Computing and VLSI*, Amalfi, Italy, May 1984; *VLSI: Algorithms and Architectures*, North-Holland Publishers, pp. 327-338.**

The basic layout of a complex cell is an array of transistors. A row of pMOS transistors is laid out adjacent to a row of nMOS transistors. As mentioned in (Uehara81), because of the requirement that the pMOS and nMOS sides are each other's dual, the number of transistors is the same in both rows. This allows the transistors on the two sides to be aligned vertically with common inputs as shown in Figure 1. Transistors which are physically adjacent and have electrically common diffusion regions may be collapsed as shown in Figure 2, making the cell width smaller. Note that in the case of CMOS circuits the adjacent transistors may be butted only if their diffusion regions can be connected on both the p side and the n side. The same layout in nMOS would require no separations. (A discussion on the nMOS case may be found in (Nair77).)

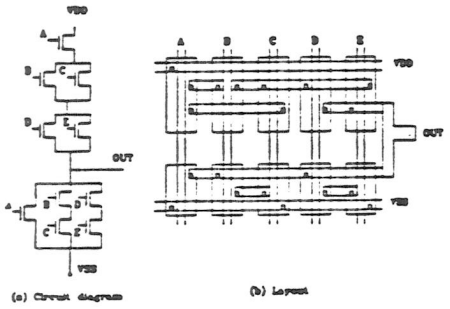


Figure 1: Basic layout of complex cell

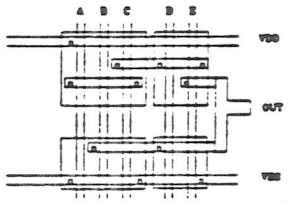


Figure 2: Layout after collapsing adjacent diffusion regions

In general a circuit realizes a boolean function. As an example one might want to realize the function $\overline{A + B.C + D.E}$, where $+$ stands for the Boolean OR function . stands for the Boolean AND function and the vertical bar stands for negation. Note that there are many circuits which realize this function as for instance $\overline{A + B.C + D.E} = \overline{B.C + A + D.E}$. Although these two circuits are logically equivalent, their layout may vary in size. Hence we address the following problem: Given a series-parallel network of transistors, find a CMOS layout implementation which has the least width. Essentially then, it is necessary to find a permutation of the transistors such that a maximum number of adjacent transistor pairs can share a diffusion area. (Equivalently, the number of separations between adjacent transistors needs to be minimized.)

III. Graph-theoretic formulation of problem

The elegant graph-theoretic formulation, originally presented in (Uehara81) will be repeated here for completeness. Some graph-theoretical terms are stated informally at first to motivate the precise definitions given later in this section. With any given circuit we associate a p-graph, $G_p = (V_p, E)$ and an n-graph, $G_n = (V_n, E^*)$ which define the structure for the p and n sides of the layout respectively. The sets of edges, E and E^* represent the inputs to the gates of the transistors in the circuit on the p and n-side respectively. Each source and each drain is represented by a vertex in the graph. However a vertex in either graph represents all the sources or drains which are electrically common. Any circuit which logically realizes the dual function of the n-side is sufficient for the p-side. But we shall assume that G_p is the dual of G_n. (In the rest of the paper reference will be made to the dual of a graph, when the plane-embedding of the graph is unambiguous.) It is a common practice among designers to assume this duality condition even though it is not necessary. This is so that the same inputs serve both sides of the circuit in such a model. Besides, a single representation can define both graphs simultaneously. Note that the set of edges E^* in G_n is dual to the set of edges E in G_p.

Recall that we are considering only series-parallel networks - hence the graphs G_n and G_p are series-parallel graphs. Associated with a series-parallel graph are two distinguished vertices which determine the points at which other series-parallel graphs may be connected. Let s_1 and t_1 represent these vertices in a graph G_1. If another series-parallel graph G_2 with distinguished vertices s_2 and t_2 is connected in parallel to G_1, the resulting series-parallel graph will have two distinguished vertices, which are simply obtained by (i) merging vertices s_1 with s_2 and t_1 with t_2 or (ii) merging vertices s_1 with t_2 and t_1 with s_2. If the two graphs were combined in series then the distinguished vertices of the combined graph would consist of one vertex of one graph and one of the other. For example, if the series combination were formed by merging t_1 with s_2 then s_1 and t_2 would form the distinguished vertices of the combined graph.

Both G_p and G_n are series-parallel graphs which are known to be *planar*. Planar embeddings have well defined duals (see (Lawler76), for example).

As mentioned in the previous section, a minimum width for the layout can be obtained by a maximal sharing of adjacent diffusion regions. In the case when there are no separations between adjacent transistors in the layout, this implies the discovery of a path through the graph which covers each edge exactly once. This is readily recognized as the problem of finding an Euler path in the graph.

In order to be able to vertically align the inputs for the *p* and the *n* sides in a CMOS implementation it is necessary that the same path must be Eulerian for both G_p as well as G_n.

Definition: Let $G = (V,E)$ be an undirected planar connected graph with a fixed planar embedding with vertex set V and edge set E. A *dual-Euler path* or *d-Euler path* is an Euler path in a plane-embedded graph G such that the corresponding sequence of edges form an Euler path in $G_d = (V_d,E)$, the dual of G. A planar embedded graph which has a d-Euler path will be referred to as *d-Eulerian*.

It is obvious that not all graphs have a d-Euler path since not all graphs are Eulerian. Uehara and vanCleemput (Uehara81) show that if both the number of transistors in every series set and every parallel set is odd then the graph of the resulting network is d-Eulerian.

Definitions: A *complete set of paths*, P^G, for a graph $G = (V,E)$ is defined as a set of paths which covers every edge in E exactly once. (In Figure 3, (ABD,CE) forms a complete set of paths in G_p.) A *complete set of paths*, $\mathbf{P}^G$, in (G,G_d), where G_d is the planar embedded dual graph of G, will be defined as one which is a complete set for both G as well as G_d. (A,BC,DE) is an example of a complete set of paths for (G_n,G_p) in Figure 3.

There may not be a unique complete set of paths. Thus $\mathbf{P}^G \in \{P^G\} \cap \{P^{G_d}\}$, where $\{P^G\}$ and $\{P^{G_d}\}$ represent the set of complete sets of paths for the graphs G and G_d respectively.

Definition: A *minimal complete set of paths*, $\mathbf{P}^G_m$, is a complete set of paths in (G,G_d) having the least cardinality, i.e., the minimum number of paths.

The cardinality of a set $\mathbf{P}^G_m$ is 1 if the graph G is d-Eulerian. (Referring to Figure 3, (ABC,DE) forms the minimal complete set of paths. Again the minimal set is not unique as exemplified by $(A,BCED)$ which also has a cardinality of 2.)

Let P^G_m and $P^{G_d}_m$ represent complete sets of paths in G and its dual G_d each having minimal cardinality. Then, by duality, $\{\mathbf{P}^G_m\} = \{\mathbf{P}^{G_d}_m\}$, and $|\mathbf{P}^G_m| \geq \max(|P^G_m|, |P^{G_d}_m|)$. The first condition follows from the fact that a planar graph is itself the unique dual of its dual graph. The second condition follows directly from $\{\mathbf{P}^G\} = \{P^G\} \cap \{P^{G_d}\}$.

A minimal complete set of paths for a CMOS circuit determines a minimal width implementation for the circuit. However, it is rather expensive to compute the minimal complete set by enumerating the complete sets for G and G_d. In fact, even in the apparently simple case where the graph is known to be d-Eulerian, it is computationally expensive to list all Euler paths in one graph to determine one which is also an Euler path in the dual, since the number of such paths may be exponential. The problem is further complicated by the fact that permutation of the inputs in a series set or parallel set of say G_n while not affecting $|P_m^G|$ or $|P_m^{G_d}|$ individually, could affect the minimal complete set of paths $|P_m^G|$ for the combined circuit.

These concepts are illustrated by Figures 3 and 4. Figure 3a shows a circuit for the function $\overline{A + B.C + D.E}$. The graph model for the circuit is shown in Figure 3b. Each of the graphs G_p and G_n individually have Euler paths, e.g. $ABDEC$ in G_p and $ABCED$ in G_n however the graphs are not d-Eulerian. A minimal complete set of paths for (G_n, G_p) is (ABC, DE). Figure 4a shows a logically equivalent circuit for the same function, the only change from Figure 3 being the permutation of the input variables to restate the original expression as $\overline{B.C + A + D.E}$. The new graphs G_p' and G_n' are shown in Figure 4b. These graphs are d-Eulerian, as evidenced by the path $BCADE$.

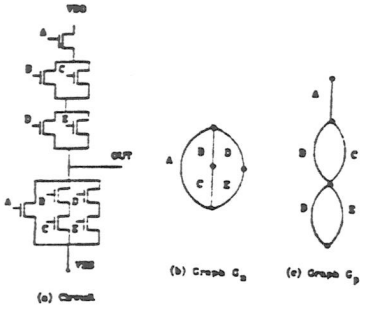


Figure 3: Circuit and graph model for $\overline{A + B.C + D.E}$

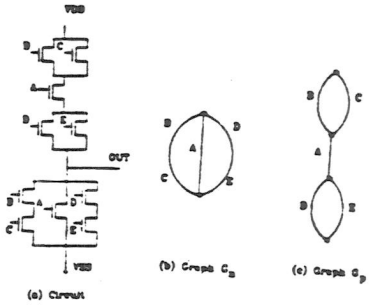


Figure 4: Circuit and graph model for $\overline{B.C + A + D.E}$

The above example shows that to obtain an optimal CMOS layout of a series-parallel circuit the possible permutations of transistors in the circuit have to be taken into account. Towards this end we define logically equivalent graphs.

Definition: Two series-parallel graphs G and G' are said to be logically equivalent if G' can be derived by reordering subgraphs of G appearing in series.

Hence, if C and C' are two logically equivalent series-parallel circuits where C' can be obtained by permuting the transistors in C, then the graphs G and G' associated with C and C' are logically equivalent. (It should be obvious what we mean by a graph associated with a circuit.)

It then becomes more interesting to restate the problem given at the end of the previous section: Given a Boolean expression minimized with respect to the number of literals find the smallest complete set of paths for all series-parallel circuits representing the expression. Equivalently we can state the problem: Given a Boolean expression minimized with respect to the number of literals and its associated graph G, find a logically equivalent graph G' with embedding G'_n and dual G'_p which has the smallest complete set of paths. The following sections show how to solve this problem. It will be seen that when the smallest complete set of paths has cardinality 1, the problem can be solved in linear time.

IV. Composite model

Let us consider a planar embedded series-parallel graph G_n with distinguished vertices (s_n,t_n) and its dual G_p with distinguished vertices (s_p,t_p). Let $e(s_n)$ represent the set of edges connected to the vertex s_n. We define similarly the sets $e(s_p)$, $e(t_n)$ and $e(t_p)$. Let us also define e_{ss}, e_{st}, e_{ts} and e_{tt} as $e_{ss} = e(s_n) \cap e(s_p)$, $e_{st} = e(s_n) \cap e(t_p)$, $e_{ts} = e(t_n) \cap e(s_p)$ and $e_{tt} = e(t_n) \cap e(t_p)$. Since edges that are in series in one graph must be in parallel in the dual graph, it is not difficult to see that each of e_{ss}, e_{st}, e_{ts} and e_{tt} consists of a single (though not necessarily distinct) edge in G_n or G_p. For example, for the graphs in Figure 5, $e_{ss} = A$, $e_{st} = A$, $e_{ts} = C$, and $e_{tt} = E$.

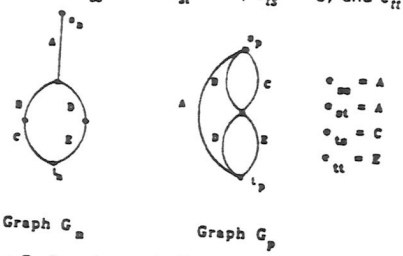


Figure 5: Sample graph illustrating distinguished edges

Consider a path ρ in G_p which is also a path in G_n. We call ρ a *0-terminal path* if ρ has either no terminal at the distinguished vertices of G_n or no terminal at the distinguished vertices of G_p. We call ρ a *1-terminal path* if ρ has exactly one terminal at a distinguished vertex of one of the graphs (G_n,G_p) and at least one terminal at a distinguished vertex of the other. Finally, we call ρ a *2-terminal path* if the terminals of ρ are distinguished vertices of both G_n and G_p. A 0-terminal path will be represented by a *representative graph* which has no vertices or edges but has an *index* 1. The representative graph for a 1-terminal path will be a single vertex v_{ss}, v_{st}, v_{ts} or v_{tt}. The first letter of the subscript refers to the graph G_n and the second to its dual graph G_p. Thus a path which originates (or terminates) at t_n in G_n and originates (or terminates) at s_p in G_p will be represented by v_{ts}. A 2-terminal path will be represented by an edge joining two vertices v_{xy} and $v_{x'y'}$ where $x,y,x',y' \in \{s,t\}$. For example, the two vertices would be v_{ss} and v_{tt} for some path ρ if ρ has terminals at s_n and t_n in graph G_n and terminals at s_p and t_p in G_p. In addition, the end edges for the path must be e_{ss} and e_{tt}. (The latter condition is imposed only to distinguish it from the case where the vertices of the representative graph are v_{st} and v_{ts}). Both 1- and 2-terminal paths are associated with an index 0. Examples of representative graphs for 0-, 1- and 2-terminal paths are shown in Figure 6. As illustrated by the trivial case of Figure 7, it may be possible to have a path which has more than one representative graph.

Let us represent a series-parallel graph G_n and its dual graph G_p with a complete set of paths $\mathbf{P}$ by a *representative graph* $H = (V,E)$ having an *index* γ, formed by the union of the representative graphs for all paths $\rho \in \mathbf{P}$ where γ is the sum of the indices of all paths $\rho \in \mathbf{P}$.

Proposition 1: A representative graph $H = (V,E)$ has the following properties:

(i) $V \subseteq \{v_{ss},v_{st},v_{ts},v_{tt}\}$

(ii) For all $v \in V$, degree$(v) \leq 1$.

(iii) For all $v_{xy} \in V$, $x \in \{s_n,t_n\}$, $y \in \{s_p,t_p\}$, there exists a path in G_n which is also a path in G_p. Further this path has a terminal in distinguished vertex x of G_n and y of G_p.

(iv) For all $(v_{xy},v_{x'y'}) \in E$, $x,x' \in \{s_n,t_n\}$, $y,y' \in \{s_p,t_p\}$, there exists a path in G_n which is also a path in G_p. Further this path has both its ends terminating at distinguished vertices in both graphs.

(v) The number of paths (as represented by H) in G_n which are also valid paths in G_p is exactly $\gamma + |V| - |E|$.

(vi) Both edges (v_{ss},v_{tt}) as well as (v_{st},v_{ts}) do not exist in a representative graph.

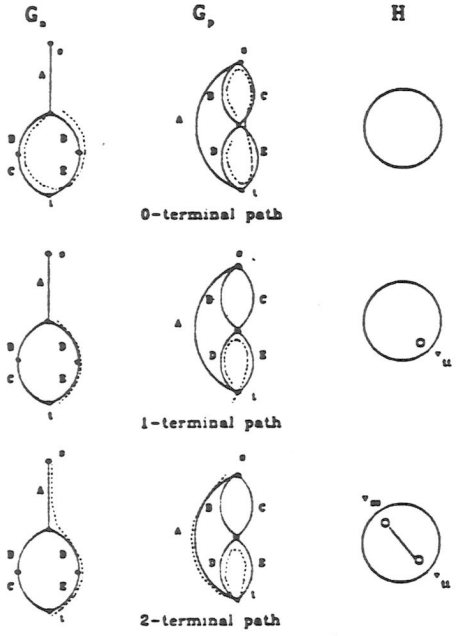


Figure 6: Representative graph for various path types

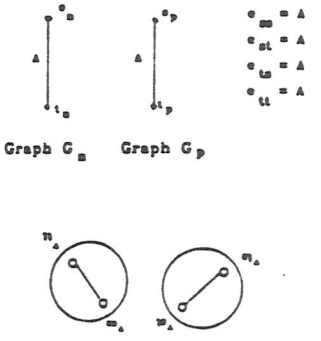


Figure 7: Trivial graph and its possible representative graphs

V. Series and parallel composition of representative graphs

The forms that a representative graph (ignoring the index) can take are limited by the properties stated in Proposition 1. Figure 8 illustrates all the 18 forms. A given circuit can have many different representative graphs. Even if the path set is restricted to the minimal complete set of paths, a circuit could have more than one representative graph. This section will examine how the representative graphs for a given circuit may be determined. We will use a recursive technique.

Let $H = (V,E)$ with index γ, $V \subseteq \{v_{ss}, v_{st}, v_{ts}, v_{tt}\}$ and
$\overline{H} = (\overline{V},\overline{E})$ with index $\overline{\gamma}$, $\overline{V} \subseteq \{v_{\overline{s}\overline{s}}, v_{\overline{s}\overline{t}}, v_{\overline{t}\overline{s}}, v_{\overline{t}\overline{t}}\}$

be the representative graphs for two graphs G and $\overline{G}$ with distinguished vertices (s,t) and $(\overline{s},\overline{t})$ respectively. Consider the series composition, $\tilde{G}$, of G with $\overline{G}$, connecting t with $\overline{s}$. Also assume that this corresponds to a parallel composition of the duals G^d and $\overline{G}^d$ to form $\tilde{G}^d$ where distinguished vertices s^d and $\overline{s}^d$ are connected together as are distinguished vertices t^d and $\overline{t}^d$. Construct the graph

$H' = (V \cup \overline{V}, E \cup \overline{E} \cup E')$, where $E' = \{(v_{ts}, v_{\overline{s}\overline{s}}) \cup (v_{tt}, v_{\overline{s}\overline{t}})\}$.

An example is shown in Figure 9.

Now consider $\tilde{H} = (\tilde{V},\tilde{E})$ with index $\tilde{\gamma}$, where

$\tilde{V} = \{\{v_{ss}, v_{st}\} \subseteq V\} \cup \{v_{\overline{t}\overline{s}}, v_{\overline{t}\overline{t}} \subseteq \overline{V}\}\}$,

$\tilde{E} = \{(x,y) \mid x,y \in \tilde{V}$, if and only if there exists a path from x to y in $H'\}$, and

$\tilde{\gamma} = \gamma + \overline{\gamma} +$ (the number of paths and isolated vertices in H' that are not in $\tilde{H}$).

Figure 10 shows $\tilde{H}$ for the example in Figure 9.

Theorem 1: $\tilde{H}$ is a representative graph for $\tilde{G}$.

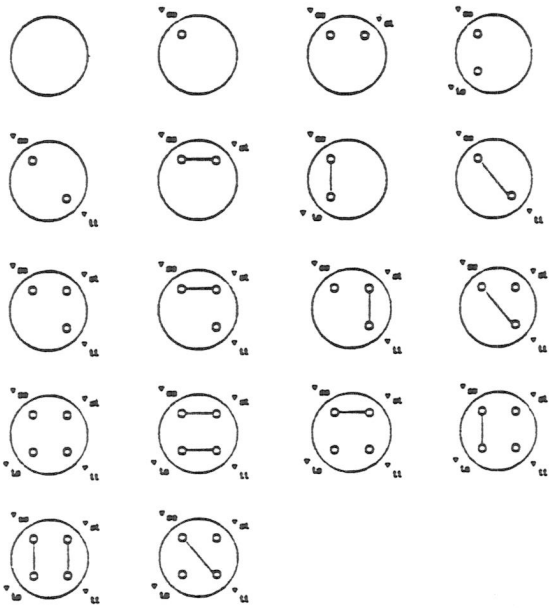
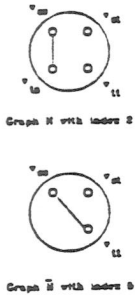
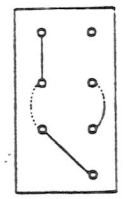


Figure 8: The 18 canonical forms for a representative graph



Graph $\tilde{H}$ with index 2

Graph $\tilde{H}$ with index 8



Graph H'
(Dotted lines belong to edge set E')

Graph $\tilde{H}$ with index 8

Figure 10: Representative graph of series composed components of Figure 10

Figure 9: Illustration of series composition

In general there may be multiple representative graphs for the component graphs which result in multiple representative graphs for the graph formed by the series or parallel composition. However, if the indices are ignored, there can be no more than 18 possible representative graphs even for the composite graph. In fact, in practical cases, the number generally is far fewer than 18. Further, since one is interested in producing a realization corresponding to a minimal complete set of paths, it is never necessary to carry around two representative graphs which have the same vertex set and edge set, but differ only in the index γ. In fact, it is expedient to retain from representative graphs having the same vertex set and edge set only that which has the least index. We will refer to this as the minimal index representative form.

Consider the graph $\dot{H}$ formed exactly as $\tilde{H}$ with the exception that edges in E' are ignored.

Corollary 1: $\dot{H}$ is a representative graph for $\tilde{G}$ having the same vertex set as $\tilde{H}$.

We will refer to the set of paths represented by $\dot{H}$ as a *trivial* complete set of paths. Every path in a trivial complete set of paths represented by $\dot{H}$ is contained in a complete set of paths represented by $\tilde{H}$ (both compositions being performed on the same set of component representative graphs).

Lemma 1: The set of paths represented by the composition of $\dot{H}$ with any other representative graph must be a trivial set.

Theorem 2: If $\{H\}$ and $\{\overline{H}\}$ are the sets of representative graphs for all non-trivial complete sets in G and $\overline{G}$ respectively, then the set of representative graphs $\{\tilde{H}\}$ formed by using the composition rules above contains the representative graphs for all non-trivial complete sets in $\tilde{G}$.

VI. Existence and determination of d-Euler path

Given a set of graphs, $\{G_1, G_2, \ldots, G_n\}$ we wish to find whether there exists a series composition of these graphs which produces a graph that is d-Eulerian. Let us first consider a simplified version of this problem where the ordering in the series composition is already given.

Algorithm 1: To determine a minimal complete set of paths for the series composition of a given ordered sequence of graphs $\{G_1, G_2, \ldots, G_n\}$.

Let $\{H_i\}$ be the complete set of representative graphs for G_i allowing for lateral flippings. (The lateral flip of a representative graph is obtained by interchanging the distinguished terminals in the dual graph.) Let the ordering be such that distinguished vertex s_i of G_i is connected to distinguished vertex t_{i-1} of G_{i-1} for $i = 2,3,\ldots,n$. Let G_{ij} with representative graph set $\{H_{ij}\}$ be the series composition of G_i with G_j. $\{H_{ij}\}$ is obtained by applying the composition rules of Section V on each element of $\{H_i\}$ with each element of $\{H_j\}$ and retaining only the minimal index representations. Thus the series composition of G_1 with G_2 gives G_{12}, the series composition of G_{12} with G_3 gives G_{123}, and so on.

Figure 11 illustrates the algorithm. The numbers on the representative graphs denote indices. Note that it is more convenient to carry around only one form of the representative graph, when the other can be obtained by dual-flipping.

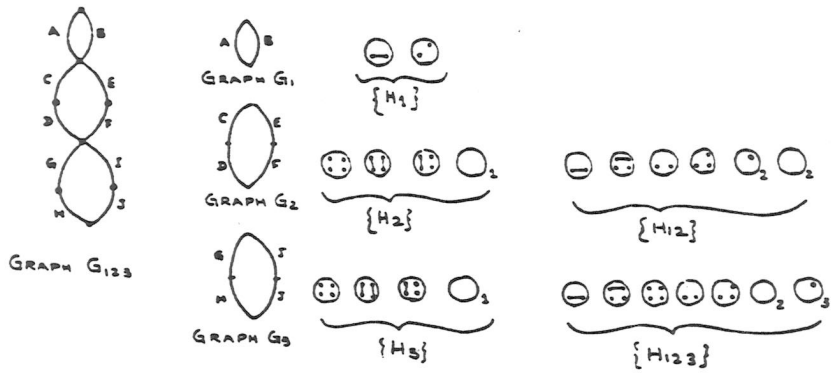


Figure 11: Illustration of series composition

Let $\pi = |V| - |E| + \gamma$ be the number of paths in G that are also represent valid paths in its dual.

Theorem 3: If π_{min} is the minimum of values of π for all representative graphs in $\{H_{123...n}\}$, then there cannot exist a complete set of paths for $G_{123...n}$, with cardinality less than π_{min}.

Corollary 2: The ordered series composition of the graphs $G_1, G_2, ..., G_n$ is d-Eulerian if and only if the set of representative graphs $\{H_{123...n}\}$ obtained as in Algorithm 1 for $G_{123...n}$ contains at least one representative graph $H = (V,E)$ with index γ which satisfies $|V| - |E| + \gamma = 1$.

Theorem 4: Algorithm 1 takes time that is linear in the number of components being connected in series.

If the distinguished vertices can be flipped in the component graphs during series composition the solution may be obtained by simply trying out either of the two cases for each component graph. This is often satisfactory when the number of components is small, as is the case in practice. But when the number of series components becomes large, the number of possibilities to be considered grows exponentially and exhaustive techniques become inefficient. This is also true when the order of the component graphs is not specified. In such cases, we present another algorithm which, while being much more complex than the approach outlined above, retains the characteristic that it is linear in the number of components to be connected.

We model the composition process by a set of finite-state machines, each machine associated with a specific form of representative graph, and each machine indicating some of the various ways in which component representative graphs can be composed to generate that representative graph. The machine model is a restricted form of the general Moore model for a finite-state machine.

A *C-machine* M is a quadruple $M = (I,S,\delta,\lambda)$ where I and S are finite, nonempty set of inputs and states respectively;

δ: $I \times S \rightarrow S$ is the state transition function;

λ: $S \rightarrow 1$ is the acceptor function.

We distinguish one of the states as the *starting state* which has no transitions into it.

The set of inputs to our algorithm is the set of distinct non-equivalent representative graphs. The C-machine for the representative graph of Figure 12(a) is shown in Figure 12(b). (Here, and in the rest of this section, we restrict ourselves to compositions which do not change the value of the index γ. This suffices for the case when the problem is to determine whether the composition can produce a d-Eulerian graph. We will also discuss how the method may be extended to the case when a complete set of paths with minimum cardinality is required.) A *possible* set of representative graphs for a set of component graphs will be understood to be one which contains exactly one representative graph form for each of the component graphs. A set of component graphs is said to be *accepted* by the machine if there exists some permutation of a possible set of the representative graphs for the components which leads the machine from the starting state to one of the acceptor states. For example, the set of component graphs shown in Figure 13(a) is accepted by the C-machine of Figure 12(b), while the set of Figure 13(b) cannot be accepted by the machine.

For simplification of the algorithm we will break down the C-machines for a representative graph into a set of *simple* C-machines. The simple C-machines for the machine of Figure 12(b) are shown in Figure 14.

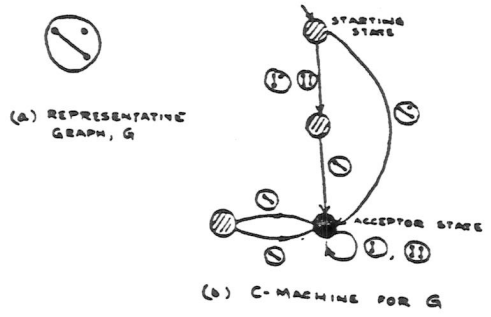


Figure 12: C-machine example

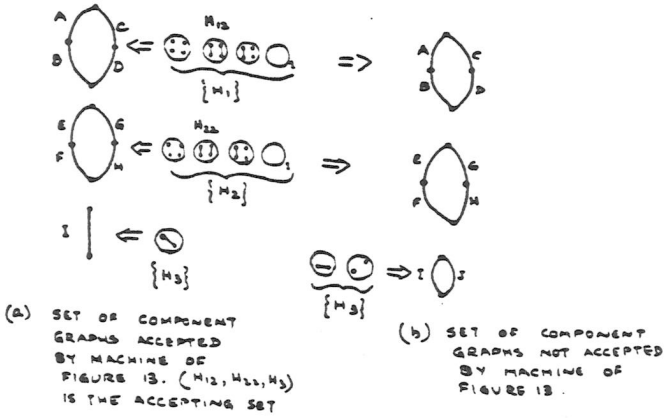


Figure 13: Illustration of accepting component graph representation

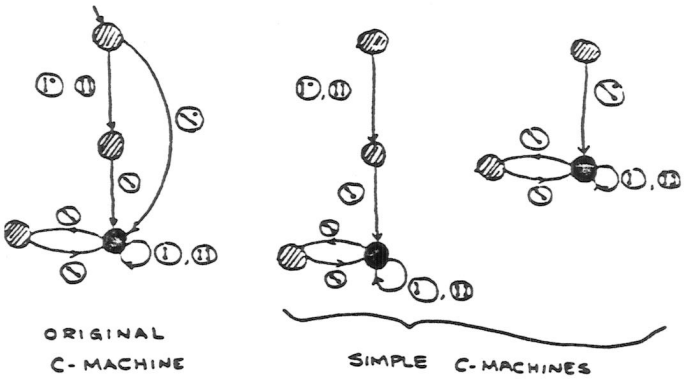


Figure 14: Illustration of simple C-machines

In general, a simple C-machine has the following properties:

(a) There is exactly one acceptor state.
(b) Every transition is associated with exactly one input.
(c) There is exactly one set of essential transitions from the starting state to the acceptor state, i.e., all input sets accepted by the machine must undergo these transitions.
(d) All other transitions form loops from a state on the essential transition path back to the same state.

A lists all the simple C-machines for all possible representative graphs can be found in (Nair83). It may be seen that the loops of property (d) above fall into one of two categories: (i) a single transition from a state back to itself, or (ii) a loop between one state and an intermediate state with both transitions involving identical inputs. The first category corresponds to a machine which can accept any number of inputs of a given type in addition to the inputs implied by the essential transition. The second category corresponds to a machine which needs an even number of inputs of a given type in addition to the essential inputs.

For any given simple C-machine M, we can divide its set of inputs I into 3 disjoint sets I^0, I^1 and I^2, where I^0 is the set of inputs that appear only at the essential transitions, I^1 is the set of inputs that appear in the self loop transitions of category (i) above, and I^2 is the set of inputs that appear in the loop transitions of category (ii).

Algorithm 2: To determine the existence of a d-Euler path in the series composition of a set of component graphs.

Retain only those representative graphs for each component graph which have an index of 0. If one of the component graphs has no representative graph with index 0, then the algorithm is complete: there does not exist a d-Euler path in the series composition. Classify the set of graphs to be composed serially into a set of equivalence classes, $Q = \{Q_1, Q_2, \ldots, Q_{|Q|}\}$, where each class Q_i is associated with a unique set of representative graphs, $R(Q_i)$. Further let n_i be the number of component graphs falling in class Q_i. For each C-machine, M, in the complete set of simple C-machines for all representative graphs we do the following:

Decompose the set of inputs, I, associated with the machine M into the three subsets, I^0, I^1 and I^2 as mentioned above. For each input r_j in I let m_j be the number of times that input appears in the essential path of the machine. We denote as x_{ij} the number of representative graphs of the form r_j that need to be taken from equivalence class Q_i, i taking on $|Q|$ values and j taking on $|R(Q) \cup I|$ values. A *possible* set of representative graphs can be found only if $I^0 \subseteq R(Q)$. We now write the following set of inequalities:

$$\begin{aligned}
\Sigma_j x_{ij} &= n_i \text{ for each equivalence class } Q_i \\
\Sigma_i x_{ij} &= m_j \text{ for all inputs } r_j \in I^0 \\
\Sigma_i x_{ij} &\geq m_j \text{ for all inputs } r_j \in I^1 \\
(\Sigma_i x_{ij}) - 2k_j &\geq m_j \text{ for all inputs } r_j \in I^2 \\
k_j &\geq 0 \text{ for all inputs } r_j \in I^2 \\
x_{ij} &\geq 0 \text{ for all } (i,j),\ r_j \in R(Q_i) \\
x_{ij} &= 0 \text{ for all } (i,j),\ r_j \notin R(Q_i)
\end{aligned}$$

We now determine a feasible solution such that all x_{ij} and all k_j take on integer values.

Theorem 5: The series composition of the set of component graphs is d-Eulerian if and only if a feasible solution exists for the above set of equations for some machine M which corresponds to a representative graph $H = (V,E)$ with index γ, such that $|V| - |E| + \gamma = 1$.

Theorem 6: Algorithm 2 takes time that is linear in the number of components being connected in series.

A more efficient solution (i.e., with a smaller constant factor) may often be obtained by using a linear programming approximation. As mentioned earlier, in practical cases where the number of components is less than about 5, it may be more efficient to enumerate the permutations and proceed as suggested for the ordered graph.

When the minimal complete set of paths is to be determined, it is necessary to keep all the representative graphs forms of minimum index for the component graphs, rather than only those with index 0. Further, the set of simple C-machines must be extended to accommodate those that could increase the index due to composition. It is not known at present whether a polynomial time algorithm exists in this case.

We have not described the parallel composition process. It should be clear though that the procedure is analogous to the series composition in all its details. Specifically, one needs to simply turn the representative graphs over on their sides and proceed with the series composition in order to get a parallel composition of the component graphs. In fact, the only reason why the complete set of machines for all possible representative graph forms was considered in Algorithm 2 as described above (rather than just those which will lead to representative graphs for d-Eulerian graphs) was that they may be needed in a later parallel composition of the series composed components themselves.

By repeated steps of series compositions and parallel compositions, we get a recursive algorithm to determine the existence of a d-Euler path in a general series-parallel circuit, or to determine the minimal complete set of paths for the entire circuit.

Theorem 7: The time needed for determining whether a given series-parallel circuit has a d-Eulerian graph grows linearly as the number of transistors in the circuit.

References

(Gibson77) D. Gibson and S. Nance, 'Symbolic system for circuit layout and checking,' in *Proc. IEEE Int. Symp. Circuits and Syst.*, 1977, pp. 436-440.

(Larsen71) R. P. Larsen, 'Computer-aided preliminary layout of customized MOS arrays,' *IEEE Trans. Comput.*, vol. C-20, pp.512-523, June 1971.

(Lawler76) E. Lawler, *Combinatorial Optimization: Networks and Matroids* New York, Holt, Rinehart and Winston, 1976.

(Muroga79) S. Muroga, *Logic Design and Switching Theory* New York, John Wiley, 1979.

(Muroga82) S. Muroga, *VLSI System Design,* New York, John Wiley, 1982.

(Nair77) R. Nair and G. Metze, 'An algebra for the realization of switching functions using a certain type of MOS package,' Coordinated Science Laboratory Report UILU-ENG 77-2222, University of Illinois, Urbana, Illinois, 1977.

(Nair83) R. Nair, A. Bruss and J. Reif, 'Linear time algorithms for optimal CMOS layout,' *Research Report* RC-10279 IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1983.

(Schweikert71) D. G. Schweikert, 'Computer-generated IGFET layout using vertically packed Weinberger arrangement,' in *Dig. IEEE Int. Solid-State Circuits Conf.*, Philadelphia, PA, 1971, pp. 118-119.

(Uehara81) T. Uehara and W. M. vanCleemput, 'Optimal Layout of CMOS Functional Arrays,' *IEEE Trans. Comput.*, pp. 305-312, May 1981.